SIMULATING DESIGN THEORY USING LLM AGENTS: A CASE STUDY OF C-K THEORY

Kevin Ma
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: kevinma1515@berkeley.edu

Eric Reynolds Brubaker*
Engineering Integration Branch
NASA Langley Research Center
Hampton, VA 23666
Email: eric.r.brubaker@nasa.gov

Kosa Goucher-Lambert†
Department of Mechanical Engineering
University of California
Berkeley, California 94720
Email: kosa@berkeley.edu

ABSTRACT

In the majority of computational simulations developed for engineering design research, the focus is on simulation for the purpose of analysis, such as simulating stresses to identify yield or fracture points in structures. However, what about simulating design theory for the purpose of developing theory? In fields like organizational psychology, simulations have proven valuable in predicting behaviors and understanding decision-making processes. This work draws inspiration from those fields to investigate its applicability in design theory. This work use C-K design theory as a representative case study to demonstrate this approach. We designed a simulation using computational agents fueled by large language models. The simulation was designed to adhere to the C-K theory methodology in both wording and framework. The results of the simulation were evaluated utilizing a mix of both qualitative and quantitative methods. Findings from the results reveal that the concept to concept transition was the predominant operation and that diversity trended downwards across multiple experimental conditions. These findings from the simulation highlight gaps in C-K theory and suggest directions for future theoretical development. Ultimately, this case study demonstrates that design theories can be effectively simulated computationally, enabling the design research community to better understand and develop improved design theories.

1 Introduction

A significant proportion of work in the engineering design research field is to develop computer-aided engineering systems to simulate design processes. Examples include new methods for simulating structural stress and analysis through finite element analysis, as well as fluid flow and aerodynamics via computational fluid dynamic models [1, 2, 3, 4, 5]. But what about the simulation of design theory? Recent studies focus on whether simulating certain design processes or methods can lead to better design outcomes or can replicate similar outcomes to human case studies [6, 7]. Going further back, authors like *Newell and Simon* simulated the problem space to understand how different heuristics might navigate it and extending those findings to speculate about human decision-making processes [8]. This built on *Simon's* seminal work, *The Sciences of the Artificial* whereby he proposed that systematic procedures and logical processes are core components of all design processes. However, subsequent work has critiqued these views as unrealistic, noting that design is not entirely rationalistic and frequently influenced by context and situational factors [9, 10].

Nevertheless, in fields like organizational psychology, simulations have proven valuable in exploring decision-making processes, understanding how different variables might impact outcomes, and predicting behaviors across different scenarios [11, 12]. For example, *Davis et. al.* proposed that computational simulations can be useful for developing theory and can be employed to understand and further develop existing theories [12]. They argued that simulations are most effective for capturing theories that unfold over time and are best used when the theoretical framework follows a logical structure [12]. Similarly, we suspect that simulating design theories could potentially help us capture underlying mechanisms that lead to successful design outcomes, such as increased innovation and higher creativity.

The goal of this paper is to explore whether computationally simulating a design theory can generate new insights and advancements for the theory based on the simulation's findings. In this study, we use C-K theory as our representative case study

*This work was conducted in the second author's personal capacity as an independent researcher, rather than as part of his official responsibilities with NASA.
†Address all correspondence to this author.

because (a) the theory is time-dependent, as pathways between or within concept and knowledge spaces occur over time, and (b) C-K theory follows a structured methodology, making it easier to simulate than other design theories [13, 14, 15].

We develop a simulation using pre-trained large language models (LLMs). The simulation is designed to follow the C-K theory methodology in both wording and framework. We want to emphasize that the focus of this paper is not on C-K theory itself. Rather, our work aims to highlight the utility of simulation in design theory and explore whether there are nuances to consider when using simulations to enhance our understanding of the design theories developed by the community.

2 Background

2.1 C-K Theory

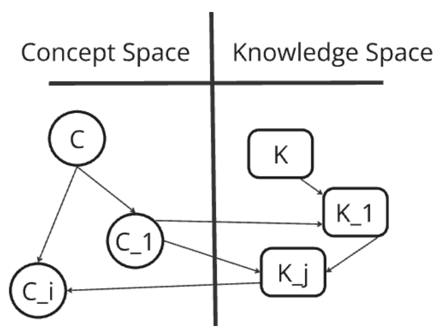

FIGURE 1: Figure representation of the C-K theory as proposed by [13].

C-K theory delineates two distinct spaces: the concept space (C) and the knowledge space (K) (see Figure 2) [15, 13]. The knowledge space contains propositions with a logical status, meaning elements in this space can be evaluated as true or false [13,15]. Thus, it can be viewed as embodying what is known and accepted by a designer. There is the caveat that the logical status can be based on standard or non-standard logical systems, but for the purpose of our paper, we simplify the logic to the classic true or false logic. As mentioned earlier, the knowledge space is expandable, allowing for the addition of new knowledge over time. However, it is important to note that conflicting views and uncertainties can also be a part of K provided they are considered established knowledge available to the designer [15].

On the other hand, the concept space consists of propositions or groups of propositions that lack a logical status in the knowledge space [13, 15]. Elements in this space are called concepts, and they are essentially ideas or properties that cannot be proven true or false within the current knowledge space. They serve as the starting point for design by representing new possibilities that are not yet integrated into the existing knowledge space [13, 15]. Like the knowledge space, the concept space is also expandable, permitting the generation of new concepts during the design process.

The key distinction appears to lie in how the information is treated [14]. If the information serves as a basis for further exploration and is open to modification or rejection, it likely aligns more with the concept space. However, if the information is accepted as a foundational element, even if the information contains some uncertainty, it likely aligns more with the knowledge space. This approach to distinguishing how C and K are treated has been discussed in prior literature that talked about the importance of situating the theory around the current environmental context [16].

The process of C-K theory begins with an initial concept C_0, a proposition such as *"there exists an entity x with attributes A_0 that is neither true nor false in K"* [15]. This concept represents the initial idea. As the design process unfolds, the initial attributes A_0 are modified, leading to new sets of attributes A_i and design parameters D_i. This results in new propositions C_i whereby *"there exists an entity x with attributes A_i and design parameters D_i"* [15]. Each new proposition C_i is evaluated against the current knowledge space K [15]. Three possible logical outcomes exist for C_i: C_i is false in K, C_i is true in K, or C_i is neither false nor true in K [15]. If C_i is false in K, the design process must alter some attributes or design parameters, indicating the concept will likely require further change [15]. If C_i is true in K and is a candidate for a solution for X, the attributes A_i and design parameters D_i together form a potentially viable solution to the design problem [15]. If C_i is neither true nor false in K, it becomes a new concept and the design process must continue [15].

This logic implies that new concepts are generated as design processes explore possibilities, and new knowledge is added as concepts are tested and validated or invalidated. This process is iterative, with continuous feedback between C and K. Thus, the C-K theory defines *"design as a reasoning activity that starts with a concept (an undecidable proposition regarding existing knowledge) about a partially unknown object x and an attempt to expand it into other concepts and/or new knowledge"* [15]. Among the knowledge generated by this expansion, certain new propositions can be selected as new definitions or new objects arise.

Thus, within this framework, we eventually notice there exists four possible operators for C-K theory: $C \to C$, $C \to K$, $K \to K$, and $K \to C$. Each operator facilitates the expansion of spaces C and K. $C \to C$ operates to expand concepts by partitioning and exploring new attributes. $C \to K$ transforms concepts into knowledge by validating propositions (e.g., C_i). $K \to K$ expands knowledge by adding new validated propositions, and $K \to C$ generates new concepts from existing knowledge.

We note, however, there exists little explicit guidelines on

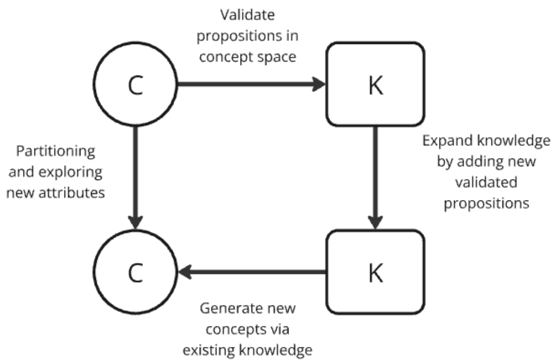

FIGURE 2: The four possible operators for C-K theory, which is also known as the design square in the original literature [13].

when to perform one operation over another (e.g., $C \rightarrow C$ vs $C \rightarrow K$ or $K \rightarrow K$ vs $K \rightarrow C$. Thus, we implicitly inferred the guidelines from prior literature that discussed the industrial application of C-K theory [14]. Thereby, we note that if one is deciding between $C \rightarrow K$ vs $C \rightarrow C$, you perform $C \rightarrow K$ by comparing the concept against existing parameters or knowledge to decide whether the concept is ready to be validated against exciting knowledge [14]. If the answer is no, perform a $C \rightarrow C$ operation. Similarly, if one is deciding between $K \rightarrow C$ vs $K \rightarrow K$, $K \rightarrow K$ is done when new knowledge can be generated based on existing propositions or data [14]. If the current knowledge instead suggests potential new concepts, a $K \rightarrow C$ operation would be conducted to generate those concepts.

2.2 Large Language Models

Large language models (LLMs) are machine learning models capable of reasoning and generating creative output through natural language generation [17]. These models are primarily trained on large text datasets to predict subsequent words or sequences in a sentence, which at the time of writing this paper, are trained primarily based on transformer architectures that enable efficient training on large datasets [18]. These models are able to decipher patterns, nuances, and context within the human language, and recently, there has been an increased development of more catered pre-trained language models that have increased capabilities in creative reasoning (e.g., GPT-4.5) [19].

With these increased capabilities in LLMs, they have been applied across a range of design tasks, including design concept generation, simulating artificial human empathy, and specific design processes such as TRIZ and FBS [20,21,6,7,22]. They have also been utilized within the framework of C-K theory, where *Chen et. al.* established a human-AI approach in which the LLM assists designers in retrieving knowledge and uncovering new concepts using C-K theory as a guideline [23]. Thus, we note the potential capability of LLMs to perform design reasoning activities, rendering them valuable tools for simulating C-K theory. Using LLMs, an LLM-driven agent can make decisions about which operation to conduct (e.g., $C \rightarrow C$ vs. $C \rightarrow K$) and perform the subsequent execution of those operations.

2.3 Using Computational Simulation to Build Theory

One of the earliest applications of simulations for theory development was Norbert Wiener's work in cybernetics [24]. Wiener proposed that systems, whether mechanical or biological, primarily function through feedback processes. He observed that systems self-regulate via feedback and suggested that human behavior could similarly be understood in terms of input and output. He argued that cybernetics theory could help further our understanding of human behavior through the principles of feedback, regulation, and information exchange. Later on, Simon and Newell explored theories related to problem-solving [8]. They proposed a theory of problem-solving whereby a problem space comprises of an initial state, a goal state, and all possible intermediate states of a problem. They then relied on heuristics to guide the navigation of this problem space. Using computational simulations, they developed and tested theories about human problem-solving to understand how different strategies and heuristics might interact and function. This work was broadly influenced by Simon's earlier work, *The Sciences of the Artificial*, where he argued that design is central to all human-made constructs, with the core goal being to create solutions that meet specific objectives within given constraints [25].

These examples illustrate that computational simulation can be a significant methodological approach to theory development. However, using simulations to build theory have also faced criticisms, with some noting that simulations are just toy models that either replicate the obvious or are so detached from realism that they fail to yield any useful and real-life applicable results [26]. Such criticisms often focus on the underlying assumptions that may operate within a simulation, which are frequently noted to be unrealistic. In a work by *Davis et. al.*, the authors argue that simulation can, instead, be valuable as the intermediate step between theory creation and theory testing [12]. They argue that simulation can help develop simple or initial theories into more precise and comprehensive ones and is particularly useful for studying processes over time. In our work, we build a simulation for C-K theory. This theory involves operations over time, which may be difficult to capture in real-world data, as many individuals may not precisely follow the logic and operations outlined in C-K theory. Thus, we use a simulation to further understand the theory and potentially offer directions for further refinement.

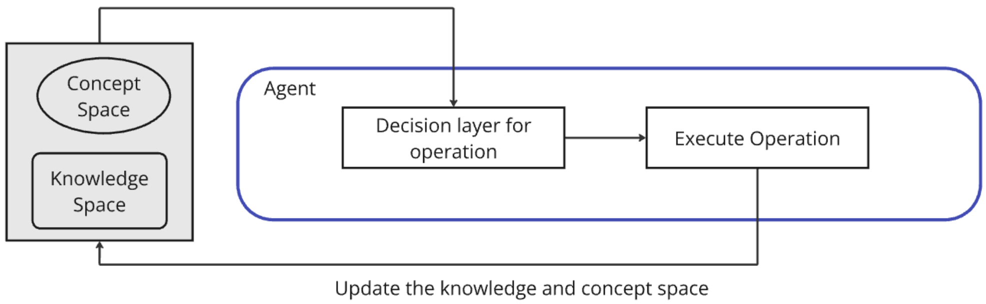

FIGURE 3: The agent takes in the concept space and the knowledge space, which contains the decision layer for the operation and the execute operation layer. The decision layer decides whether the agent should execute a $C \to C$, $C \to K$, $K \to C$, or a $K \to K$ operation. Afterwards, the agent executes the respective operation and updates the concept or knowledge space.

3 Methods

3.1 Constructing the Simulation

We encoded the C-K theory into an LLM agent, attempting to replicate the logic and operations of the C-K theory as accurately as possible according to the literature [13, 15]. The simulation follows the structure as shown in Figure 3. The agent takes in the concept space and the knowledge space to decide between four possible actions: $C \to K$, $C \to C$, $K \to C$, $K \to K$. Note that the agent can only perform a $C \to C$ or $C \to K$ operation when the current element fed into the agent is a concept, and similarly, $K \to K$ or $K \to C$ when the current element is knowledge.

The design topic we chose was one taken from prior literature: *design a creative nail holder for use while hammering a nail* [14]. We initialized the simulation by constructing a knowledge space for our design topic. We developed this knowledge space by generating design requirements using GPT-4.5 [19], the latest reasoning model by OpenAI, and validating them with design experts experienced in product design. We had a total of 20 design parameters in the knowledge space. The knowledge space we used is available in our GitHub repository[1].

The first primary component of this simulation model involves deciding whether to go to $C \to C$ or $C \to K$ when in a concept state or $K \to K$ or $K \to C$ when in a knowledge state. The second primary component is the operation the LLM conducts when it is at $C \to C$, $C \to K$, $K \to K$, or $K \to C$. To simplify our simulation, we chose to have the agent generated the output in a *Concept/Knowledge Title:* and *Concept/Knowledge Description:* format.

The first decision-making logic the agent must perform is to determine, when in a concept space, whether to proceed with concept or knowledge based on the prompt logic shown in Appendix A. The function takes in the current concept title, concept description, past operation transformations (which are updated in each iteration), and existing knowledge base. The agent then decides whether to perform a $C \to C$ or a $C \to K$ operator based on the following logic: choose $C \to C$ if the concept is ambiguous or novel and needs further exploration, or $C \to K$ if the concept is refined, aligns with existing knowledge, or has the ability to enrich the knowledge base.

Similarly, when in the knowledge space, the decision to proceed with concept or knowledge is based on the prompt logic shown in Appendix B. Here, the function takes in the current knowledge title, knowledge description, past transformations, and the existing knowledge space. The agent then decides whether to perform a $K \to C$ or $K \to K$ operator based on the following logic: if existing knowledge suggests new concepts or insights, initiate $K \to C$ for concept expansion; if new information is validated and can be integrated into the knowledge space, initiate $K \to K$ for knowledge expansion.

The agent will then execute one of the four operations after the agent has decided whether to conduct either $C \to C$, $C \to K$, $K \to K$, or $K \to C$. When the agent executes the operation, the agent will consider the context of the existing knowledge space, the current concept description and title, and past transitions. However, the level in which this occurs depends on the experimental conditions. As we have referenced in Section 2, the agent was developed to execute each operation according to the C-K theory literature: $K \to C$ is to use existing knowledge to generate new concepts, $C \to K$ is to validate or transform the concepts into knowledge, $C \to C$ is to expand or modify the concepts within the conceptual space, and $K \to K$ is to refine or expand the existing knowledge. The specific prompt we wrote for the LLM varies a bit depending on the experimental condition, but for reference some examples are provided in Appendix C. For more information, refer to the GitHub repository.

3.2 Experimental Conditions

During the simulation of the initial condition, where the agent considers only the knowledge and concept space as the state, we noticed that the agent almost always opts for a $C \to C$ operation. Thus, we implemented a penalizing mechanism that

[1] GitHub Repository

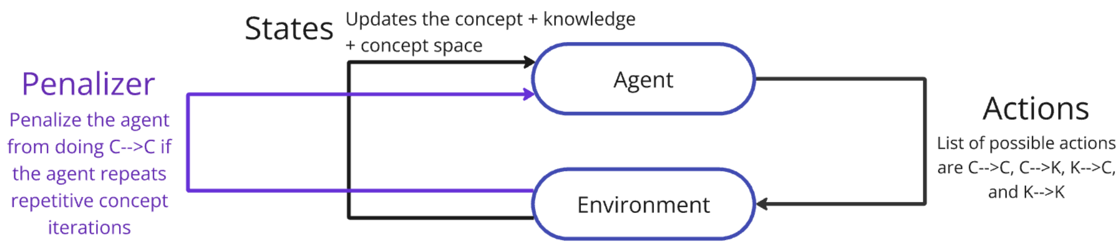

FIGURE 4: The purple indicates the penalizer mechanism for the diagram. The agent now takes in this penalizer mechanism along with the concept and knowledge space to make a decision on which operation to execute.

penalizes the agent if it repeats similar concepts excessively. This forces the agent to perform a $C \rightarrow K$ operation if similar concepts are repeated. This penalization mechanism includes an additional LLM layer to evaluate the novelty and potential of each concept that determines whether future $C \rightarrow C$ operations are justified based on unexplored attributes or additional propositions. The prompt for this penalization mechanism is shown in Appendix D, and the experimental condition for penalization is represented diagrammatically in Figure 4. In the results section, we labeled this as *With/Without FB Loop*.

We also had an additional condition of whether or not to allow the LLM agent to have the concept space as additional context when performing the operations. We initially provided the concept space to each agent when they are executing an operation (e.g., executing $C \rightarrow C$ or $K \rightarrow C$) as additional context, thereby the LLM agent would receive both the knowledge and concept space during the simulation. However, during our experiments, we observed that the LLM agent's outputs varied significantly when it did not receive the existing concept space during the simulation. This experimental condition explores how the agent's memory of concepts might impact the performance and outcome. In the results section, we labeled this as *With/Without Concept Memory*.

Finally, we introduced another experimental condition by testing different LLM models. As mentioned in Section 2, there has been an increase in the types of pre-trained LLMs available, including the releases of GPT-4o and GPT-4.5 by OpenAI [17, 19]. OpenAI reported that GPT-4.5 is a general purpose model that is much smarter than GPT-4o and outperforms all previous GPT models in complex task automation. However, it is also a much larger and more expensive model than GPT-4o. For reference, the cost of running GPT-4o for this experiment was roughly 30 times cheaper than running GPT-4.5. Our testing across different models aims to understand whether there is a substantial performance difference between pre-trained language models when simulating C-K theory. Thus, in our study, we experimented with both of OpenAI's latest creativity reasoning models: *gpt-4.5-preview* and *gpt-4o*.

3.3 Computational Evaluation

In each iteration, we generated a concept comprising various concept titles and descriptions. We converted these titles and descriptions into embeddings using OpenAI's embedding model named *text-embedding-3-small* and set the embedding dimension to 512. These embeddings were then used to measure computational diversity across each time step iteration. To measure diversity, we used both DPP and the average distance to the centroid. This method of calculating diversity has been utilized in prior work [21, 27].

The DPP framework we used to calculate diversity is based on the properties of the eigenvalues of the cosine similarity matrix. All embeddings are first normalized so that the cosine similarity between any two vectors lies in the range $[-1, 1]$. Since DPPs require a positive semi-definite kernel matrix, the cosine similarity matrix is transformed to non-negative values, and the resultant matrix is used to compute its eigenvalues. We then logarithmically scale the eigenvalues and take the average of all the values. Therefore, a more negative output from our DPP framework is due to a smaller eigenvalue, which implies that a more negative DPP output in our graph indicates a less diverse embedding space.

The average distance to centroid framework we used assesses how the embeddings are distributed around the centroid at each time step. We calculated average distance to centroid by finding the average distance of all the embeddings from the centroid. Thus, a smaller average distance indicates that the points are closely clustered, while a larger average distance would suggest that the points are more spread out. We had to reduce the dimensionality of the text embeddings to 9 dimensions to make the computation of the average distance to centroid feasible. The complete algorithm and code for this process are available in our GitHub repository.

4 Results

We now present our results in all four conditions for both GPT-4o and GPT-4.5 models. Recall, according to C-K theory, there are only two possible spaces: the concept space and the

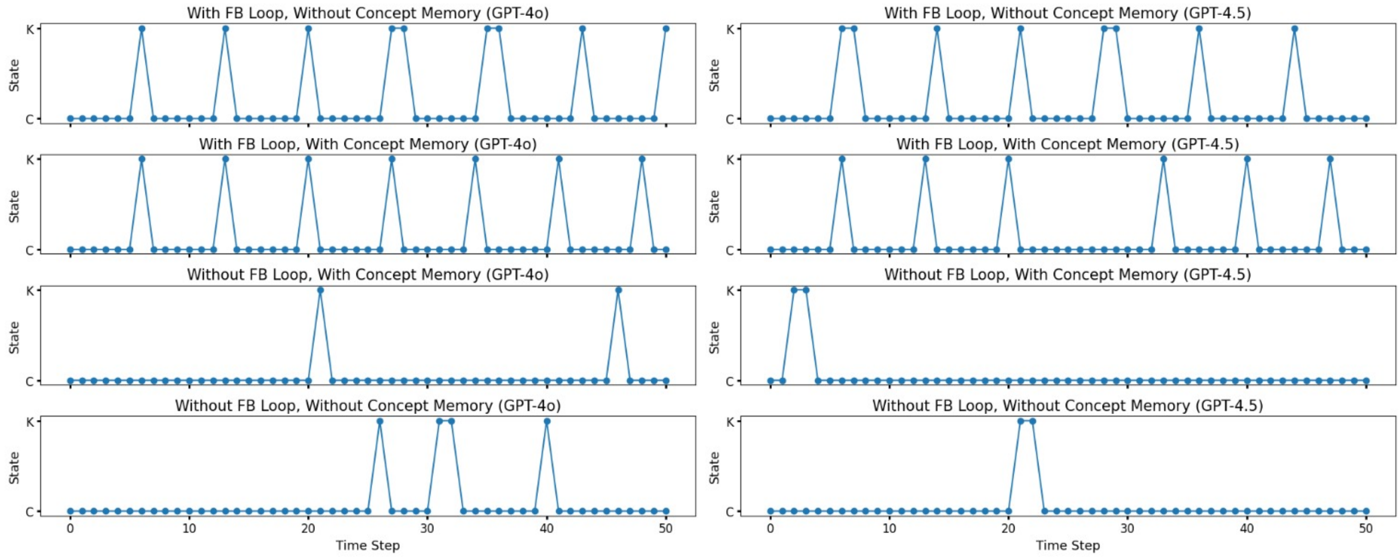

FIGURE 5: There are eight different graphs. The group on the left features those with GPT-4o, while the group on the right comprises those with GPT-4.5. From top to bottom, there are four different experimental conditions. The title of each graph contains the abbreviation *FB loop* and *With or Without Concept Memory*. *FB loop* indicates whether there is a penalizer feedback loop, and *With or Without Concept Memory* indicates whether the simulation has memory of the concept space. On the x-axis are the time steps from 1 to 50. On the y-axis are labels with C and K, where C indicates concept and K indicates knowledge. Note, that all iterations start with a concept, and this graph labels the decision the LLM agent made in its transformation.

knowledge space [13]. In our simulation, we tested four scenarios for our study: with the penalizer feedback loop and with concept memory, with the penalizer feedback loop and without concept memory, without the penalizer feedback loop and with concept memory, and without the penalizer feedback loop and without concept memory.

4.1 Transition History

Our findings across all conditions and models shown in Figure 5 reveal that $C \to C$ is the most dominant operation whereas the $C \to K$ operations are almost non-existent in scenarios without the penalizer. Recall from Section 3, the penalizer case may force a $C \to K$ operation after 5 time step if the penalizer agent determines that the past concepts are considered repetitive. As a result, we noticed that in almost all cases, there was a $C \to K$ jump after 5 time steps in both the GPT-4o and GPT-4.5 models. This indicates that, regardless of the LLM model, the penalizer agent has determined that both GPT-4o and GPT-4.5 models are generating repetitive concepts. We also noticed that there seems to be a short period where the penalizer agent does not force a $C \to K$ operation in the *with FB loop with Concept Memory GPT-4.5* case. However, this may just be an outlier.

4.2 Diversity Evaluation

In Figure 6, we evaluated the diversity of the concept space over time. Our results showed that DPP gets more negative and the average distance to centroid decreased as the time steps progressed (starting at 10 iterations). This indicates that over time, concepts became increasingly similar, resulting in a more negative DPP value. Similarly, this also indicates that newly generated concepts are increasingly clustering closer to the centroid, resulting in a decreased average distance to centroid. There is, however, a noticeable anomaly in the *with FB loop, without Con Mem* case in the average distance to centroid calculation between the time steps 25 and 35, whereby there was a slight increase in the average distance to the centroid. After analyzing the qualitative outputs of the concepts in those time steps, we determined that this is likely due to the way the average distance to the centroid is calculated. This increase is likely because the embeddings are clustering at the edges of the distance to the centroid rather than closer to the centroid. This does not indicate an increase in diversity; rather, it may just indicate that the embeddings are stagnating at a certain distance from the centroid.

In addition, there was a near consistent decrease in diversity across all conditions. We implemented these three different conditions to explore ways to increase the diversity of the concept outputs. We initially suspected that perhaps the LLM agent just needed to be forced into $C \to K$ to achieve more diverse outputs.

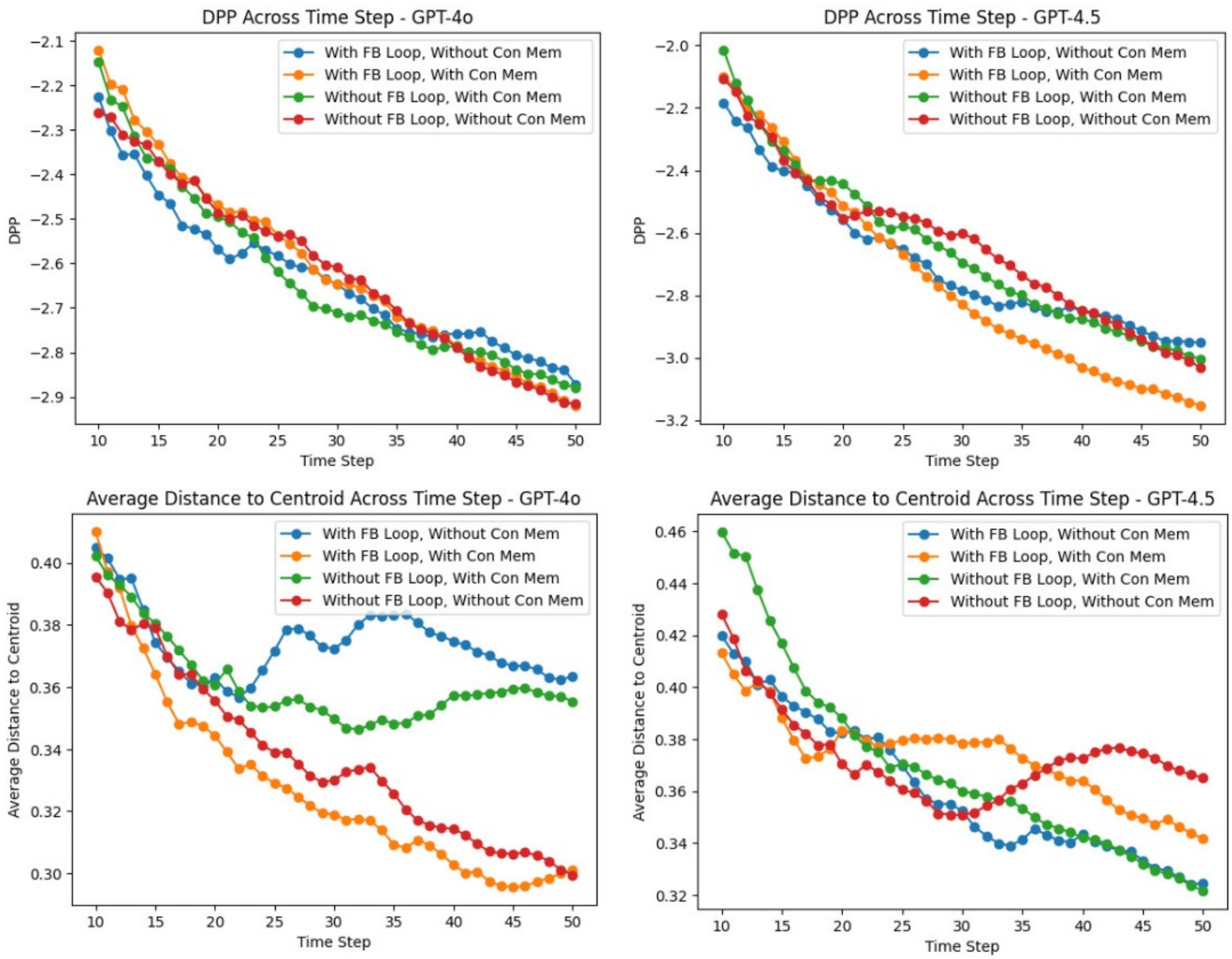

FIGURE 6: The top two graphs show the DPP values over time after 10 time steps and the bottom one displays the average distance to centroid values over time after 10 time steps. The first time step includes one concept, with each subsequent iteration adding a new concept. Diversity is measured after 10 time steps to ensure there are a sufficient number of concepts for calculating diversity. The legends indicate the four different conditions, which include the abbreviations *FB Loop* and *With/Without Con Mem*. *FB Loop* denotes whether there is a penalizer feedback loop, and *With/Without Con Mem* indicates whether the simulation retains memory of the concept space.

When that did not work, we speculated that it might be an issue with the fact that we were using a less powerful and creative reasoning model like GPT-4o instead of GPT-4.5. Or perhaps it was simply because the LLM agent received all the old concepts and was repeating them due to the agent drawing from prior examples too much. However, we noticed that regardless of these conditions, diversity continued to decrease, and the outputs became increasingly repetitive.

One example we examined to help our understanding is the qualitative output of the *With FB Loop Without Concept Memory* dataset from the GPT-4o model (see Table 1). In this case, the LLM agent has a penalizer loop to force it to go from $C \to K$ if the outputs are repetitive, so a new idea should be generated after it exits the knowledge space by performing the $K \to C$ operation. It also has no memory of the old concepts, so it should not fall into the trap of repeating things that it has already produced or "knows". However, if we look at the concepts and knowledge titles from time steps 23 to 30 in Table 1, it first introduces the idea of quantum for the first three concepts. Then, the LLM agent performs the $C \to K$, $K \to K$, and $K \to C$ operations. At time step

TABLE 1: Example concept or knowledge: *With FB Loop Without Concept Memory - GPT4o condition*

Time	C or K	Title of the Concept or Knowledge
23	C	Quantum Resonance Nail Holder: Integrating Quantum Physics and Resonance for Precision and Safety
24	C	Quantum-Biomechanical Harmony Nail Holder: Integrating Quantum Dynamics and Biomechanics for Enhanced Precision and User Interaction
25	K	Quantum Eco-Resonant Nail Holder: Merging Quantum Mechanics and Eco-Conscious Design for Enhanced Functionality and Sustainability
26	K	Quantum-Aware Eco-Materials Dynamics Nail Holder (QEMDNH)
27	C	Systematic Validation Pathways for Transitioning the Adaptive Biometric Nail Holder from Concept to Knowledge Space
28	C	Expanding Nail Holder Innovation: Integrating AR, Gamification, and Thermal Materials with Advanced Validation and Ethical Frameworks
29	C	Innovative Expansion of Nail Holder Design: Integrating Haptic Feedback, Modular Customization, AI Analytics, and Sustainable Materials
30	C	Quantum-Inspired Adaptive Nail Holder (QIANH)

28, the concept is no longer talking about quantum related concepts. However, by time step 30, it starts bringing up quantum related concepts even though it has no memory of the old concepts.

5 Discussion

We begin by describing how the present study contributes to further developing C-K theory. This is followed by a discussion on the broader potential of LLMs to be used to simulate and develop additional design theories (Function Behavior Structure theory, etc.).

5.1 Contributions to C-K Theory

Our findings offer two main contributions to C-K theory. The first is the identification of concept to concept loops, including insight into the mechanisms by which LLMs can get stuck in such loops. This finding points to a potential gap within C-K theory–a need for greater clarity on the theoretical specification of $C \to K$ transitions. The second contribution we offer is a revised theoretical specification of $K \to C$ transitions. We show that it matters not just whether and when but how $K \to C$ transitions occur. This latter contribution draws connections between C-K theory and theories of fixation and design-by-analogy to suggest a more robust C-K theory.

5.1.1 Understanding concept to concept loops: Addressing sparse $C \to K$ transitions. In the literature about C-K theory, we note that the authors stated that if C_i is true in K and is a candidate for a solution for X (X being the problem space at hand), the attributes A_i and design parameters D_i can form a potentially viable solution to the design problem [13, 15]. Thus, we created an initial knowledge space with some design parameters, so when the LLM agent is deciding whether to go $C \to C$ versus $C \to K$, the LLM agent is cross-checking whether the concept being generated is ready to be validated against the knowledge space.

However, findings from our paper reveal that the $C \to C$ operation was by far the most dominant operation while the $C \to K$ operation occurred only very rarely. We suspect that this issue stems from the fact that once the simulation performs a $C \to C$ operation, it encourages the addition of novel partitions to the concept. Thus, the underlying idea of each concept still remains the same across each time step, but with each $C \to C$ operation, new elements are added to the concept. This has the downside of making the concept more challenging to validate with the existing knowledge space, as it becomes increasingly impractical and disconnected from the established work with each time step.

Even though we tried our best to follow the literature as strictly as possible, this could be an issue with the way we interpreted the decision logic between doing $C \to C$ versus $C \to K$. One possible issue is that the LLM agent may require different wording in its prompt that is not as exact as the wording from the design literature to get the simulation to perform better. On the other hand, our findings might also suggest that there exists a gap in the theory's formal logic on when to perform a $C \rightarrow C$ operation versus a $C \rightarrow K$ operation, which suggests a need for greater clarity in the theoretical specification of C-K transitions

5.1.2 Addressing decreasing concept diversity: It matters not just whether and when but how $K \rightarrow C$ transitions occur.

In our simulation, we added a penalizer feedback loop to check if the past five or more concepts are getting repetitive. This loop penalizes the LLM agent for performing too many $C \rightarrow C$ operations that lead to repetitive or non-valuable concept additions, forcing the agent to perform a $C \rightarrow K$ operation. Note that once a $C \rightarrow K$ operation is executed, the agent must eventually transition to $K \rightarrow C$ to exit the knowledge space. During this $K \rightarrow C$ operation, the LLM agent is instructed to use existing knowledge to suggest new concepts and avoid repeating existing ones, which should lead to the emergence of different concepts.

However, across all four conditions – regardless of concept memory, feedback loop presence, or whether the simulation was using GPT-4.5 or GPT-4o – the diversity values trended downward. This raises the question: if new concepts are generated through the $K \rightarrow C$ operation, why are they not leading to higher diversity? The issue is that the LLM agent appears to always revert to previous ideas (see Table 1 for an example). This suggests that merely performing a $K \rightarrow C$ operation may not suffice for generating innovative and creative ideas, indicating a potential need to revisit what $K \rightarrow C$ operation entails in C-K theory. This implies that merely transitioning into the knowledge space and exiting it is not what matters, but rather the manner in which this transition is executed.

Overall, our findings indicate that the logic for $K \rightarrow C$ transitions in C-K theory is not sufficiently specified to ensure the generation of diverse or innovative concepts when implemented algorithmically. The persistent fixation and declining diversity observed in our LLM-based simulations suggest that, as currently formulated, C-K theory may lack the explicit logic or description to help designers to avoid concept stagnation especially when applied systematically at scale. This suggests that future theoretical development of C-K theory would benefit from more precise definitions of transition logic and from integrating other relevant design theory into C-K theory. For example, consider established and emerging design theory research related to design fixation, e.g., [28, 29, 30, 31], inspirational stimuli [32], and design-by-analogy with near and far analogical concepts [33]. For these applications, once in the knowledge space, it matters what knowledge is drawn upon and how that knowledge (e.g., as a near or far analogy) might inspire new ideas in the concept space. This has been a well studied phenomenon in the design-by-analogy literature but, although quite relevant, has not been integrated into the formal logic of $K \rightarrow C$ transitions in C-K theory.

5.2 Developing Design Theory through LLM-based Simulation

Inspired by the longstanding tradition of developing theory through simulation, e.g., [12] and the emerging promise of using LLMs to simulate social processes [34], the present study set out to investigate whether LLMs can be useful in developing design theory. By simulating C-K theory as a case example, our paper concluded that LLMs can indeed be useful for theory development. In particular, we found that LLM-driven simulations can reveal gaps in a design theory by systemizing its logic at scale and observing the emergent behaviors.

A unique capability of LLMs in this context is their ability to run simulations at a scale and speed that is unattainable with human participants. Thus, researchers can use LLM-based simulations to rapidly iterate through hundreds of design transitions to identify any theoretical gaps or patterns that may only emerge after many cycles. Additionally, LLMs are able to strictly follow the reasoning prescribed by a design theory, which is in contrast to human designers who might improvise or stray from the established logic. This strict adherence can be a double-edged sword in that, on one hand, it can enable rigorous testing of the theory's internal logic; however, it may also lead to behaviors that are less likely to occur in real-world practices with human designers.

Looking forward, the methodology we propose in this paper is not limited to C-K theory. We suspect that it could be applied to both established design theories rooted in design, such as FBS (Function-Behavior-Structure), or situations where theories from other domains are being explored in design contexts (e.g., dual process theory applied to early stage design cognition [6, 35]). Both of these theories are systematic in their processes, and could, in principle, be operationalized and simulated using LLMs. Such simulations could similarly expose ambiguities or gaps in the theories as our work did for C-K theory. For example, simulating FBS transitions or dual process reasoning with LLMs could reveal whether these frameworks are sufficiently specified for computational reasoning and if additional mechanisms are needed to avoid specific issues like fixation. More broadly, LLMs are a new tool in the tool belt of simulation approaches for theory development. Future work is needed to identify which design theories might benefit most from further development via simulation; which simulation approach (LLM-driven, agent-based model, system dynamics, etc.) might best fit a given research question, set of assumptions, and theoretical logic; and how LLMs might be used for effective simulation experiments and, ultimately, be validated with empirical data.

6 Limitations

There are several limitations to the present study. The first is that how we prompted the LLMs may be contested, and the results may differ with different prompts. While our goal was to adhere to established C-K theory as closely as possible when wording the prompts, some deviations from the original wording from the C-K theory literature was necessary. On the other hand, we may have adhered too closely to the original wording when prompting the LLM, as LLMs may require different means of prompting than a human. Future work could explore building similar simulations using prompts that are more suited for an LLM while staying true to the original C-K theoretical framework.

Secondly, there may be questions regarding whether our method of simulating C-K theory is accurate and truly representative of how C-K theory is actually applied by human designers. The simulation could be viewed as too much of a "toy model" of the C-K approach to be useful–that it strips away so much human realism or is polluted with too much LLM-specific tendencies or bias that it is simply too inaccurate to yield valid theoretical insights. This is a common criticism of simulation approaches and one that needs to be addressed further in future research using LLM-driven simulations. Future work could expand on variations in how C-K theory is approached by simulating directed C-K transitions to examine how different variations of the C-K theories are associated with various design outcomes. These variations could be tested using a statistically significant sample of simulation runs to better cover all the different possibilities of how the C-K design theory operates. Additionally, sensitivity analysis could be applied to variations in prompts.

Lastly, our argument that LLM-driven simulations have utility for developing design theory is limited by the fact that this work examined only a single design theory and a limited number of experimental conditions. While we hope our paper makes a case for this new potential approach to developing design theory, much more work is needed to examine larger, more comprehensive sets of experimental conditions across different design theories. Only then will it become clear how useful LLM-driven simulations can be for advancing design theory.

7 Conclusion

This paper examined whether LLM-driven simulations can support the development of design theory. This work used C-K theory as a case study design theory to demonstrate the utility of the simulation approach. The simulation was created using pre-trained LLMs, and the findings were evaluated both qualitatively and quantitatively. The results revealed that the simulation was consistently stuck in a $C \to C$ loop and that, regardless of the experimental conditions, the diversity of the concept space decreased over time. These findings suggest gaps in C-K theory. We identified a need for greater clarity on the theoretical specification of $C \to K$ transitions, and we suggest a strengthened theoretical formulation of $K \to C$ transitions through connections to research and theory on design-by-analogy, among others. Both contributions to C-K theory could inform future testing and development of the theory. Taken together, this case study demonstrates the potential utility of LLM-driven simulations for developing design theory.

ACKNOWLEDGMENT

Niharika Sharma, an undergraduate at the University of California, Berkeley and affiliate of the CoDesign Lab, aided in the development of preliminary prototypes of the C-K theory simulation. This work helped inform future development of the prototype, and the authors give ample thanks to her for her contribution and hard work. In addition, the authors wish to thank the members of the Berkeley Institute of Design for their valuable feedback. This work was supported through funding from the Keith L. Markolf and Cristina T. Ruland Fellowship.

REFERENCES

[1] Hazelrigg, G. A., 1999. "On the role and use of mathematical models in engineering design". *Journal of Mechanical Design,* ***121***(3), September, pp. 336–341.

[2] Weber, R. C., and Davis, R. L., 1982. "A mathematical model used in conjunction with a finite element analysis to aid in the design and analysis of bourdon tubes". *ASME. J. Mech. Des.,* ***104***(3), July, pp. 540–543.

[3] Tangsali, K., Krishnamurthy, V. R., and Hasnain, Z., 2021. "Generalizability of convolutional encoder–decoder networks for aerodynamic flow-field prediction across geometric and physical-fluidic variations". *Journal of Mechanical Design,* ***143***(5), p. 051704. Paper No: MD-20-1208.

[4] Song, B., Yuan, C., Permenter, F., Arechiga, N., and Ahmed, F., 2024. "Data-driven car drag prediction with depth and normal renderings". *Journal of Mechanical Design,* ***146***(5), p. 051714. Paper No: MD-23-1544.

[5] Marchesse, Y., Changenet, C., Ville, F., and Velex, P., 2011. "Investigations on cfd simulations for predicting windage power losses in spur gears". *Journal of Mechanical Design,* ***133***(2), p. 024501.

[6] Chen, L., Zuo, H., Cai, Z., Yin, Y., Zhang, Y., Sun, L., Childs, P., and Wang, B., 2024. "Toward controllable generative design: A conceptual design generation approach leveraging the function–behavior–structure ontology and large language models". *ASME. J. Mech. Des.,* ***146***(12), December, p. 121401.

[7] Jiang, S., and Luo, J. "Autotriz: Artificial ideation with triz and large language models". In Proceedings of the ASME 2024 International Design Engineering Technical Conferences and Computers and Information in Engineering Con-

ference. Volume 3B: 50th Design Automation Conference (DAC), ASME, p. V03BT03A055.

[8] Newell, A., and Simon, H. A., 1972. *Human Problem Solving*. Prentice-Hall.

[9] Winograd, T., and Flores, F., 1987. *Understanding Computers and Cognition: A New Foundation for Design*. Addison-Wesley.

[10] Suchman, L. A., 1987. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press.

[11] Wilson, R. C., and Collins, A. G. E., 2019. "Ten simple rules for the computational modeling of behavioral data". *eLife,* ***8***, p. e49547.

[12] Davis, J. P., Eisenhardt, K. M., and Bingham, C. B., 2007. "Developing theory through simulation methods". *Academy of Management Review,* ***32***(2), pp. 480–499.

[13] Hatchuel, A., and Weil, B., 2003. "A new approach of innovative design: An introduction to c-k theory". In DS 31: Proceedings of ICED 03, the 14th International Conference on Engineering Design, A. Folkeson, K. Gralen, M. Norell, and U. Sellgren, eds., ICED, pp. 109–110 (exec. summ.), full paper no. DS31_1794FPC.

[14] Hatchuel, A., Masson, P. L., and Weil, B., 2004. "C-k theory in practice: Lessons from industrial applications". In DS 32: Proceedings of DESIGN 2004, the 8th International Design Conference, Dubrovnik, Croatia, M. D., ed., pp. 245–258.

[15] Hatchuel, A., and Weil, B., 2009. "C-k design theory: an advanced formulation". *Research in Engineering Design,* ***19***, pp. 181–192.

[16] Kazakci, A. O., and Tsoukiàs, A., 2005. "Extending the c–k design theory: A theoretical background for personal design assistants". *Journal of Engineering Design,* ***16***(4), pp. 399–411.

[17] OpenAI, 2024. Gpt-4 technical report.

[18] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I., 2017. "Attention is all you need". In Advances in Neural Information Processing Systems, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds., Vol. 30, Curran Associates, Inc.

[19] OpenAI, 2025. Openai gpt-4.5 system card. Publication-Safety, February 27. This is an OpenAI publication.

[20] Zhu, Q., and Luo, J., 2024. "Toward artificial empathy for human-centered design". *ASME Journal of Mechanical Design,* ***146***(6), June, p. 061401.

[21] Ma, K., Grandi, D., McComb, C., and Goucher-Lambert, K., 2025. "Do large language models produce diverse design concepts? a comparative study with human-crowdsourced solutions". *ASME. J. Comput. Inf. Sci. Eng.,* ***25***(2), February, p. 024501. Published December 23, 2024.

[22] Baudoux, G., and Goucher-Lambert, K., 2024. "Automating analogical reasoning: A wizard of oz study on the benefits and pitfalls of a sketch-based ai image generator for design". In *Design Computing and Cognition'24. DCC 2024*, J. S. Gero, ed. Springer, Cham.

[23] Chen, L., Xia, D., Jiang, Z., Tan, X., Sun, L., and Zhang, L., 2025. "A conceptual design method based on concept–knowledge theory and large language models". *ASME. J. Comput. Inf. Sci. Eng.,* ***25***(2), February, p. 021001.

[24] Wiener, N., 1948. *Cybernetics: Or, Control and Communication in the Animal and the Machine*. MIT University Press, Cambridge, MA.

[25] Simon, H. A., 1996. *The Sciences of the Artificial, 3rd Edition*, 1 ed., Vol. 1. The MIT Press, December.

[26] Chattoe, E., 1998. "Starting artificial societies: A methodological introduction". *Journal of Artificial Societies and Social Simulation,* ***1***(3), p. http://jasss.soc.surrey.ac.uk/1/3/2.html.

[27] Regenwetter, L., Srivastava, A., Gutfreund, D., and Ahmed, F., 2023. "Beyond statistical similarity: Rethinking metrics for deep generative models in engineering design". *Computer-Aided Design,* ***165***, December, p. 103609.

[28] Jia, M., Jiang, S., Hu, J., and Qi, J. "Toward understanding sources and influences of design fixation: A focus on example stimuli and background of novice designers".

[29] Leahy, K., Daly, S. R., McKilligan, S., and Seifert, C. M. "Design fixation from initial examples: Provided versus self-generated ideas".

[30] Crilly, N., and Cardoso, C. "Where next for research on fixation, inspiration and creativity in design?". pp. 1–38.

[31] Vasconcelos, L. A., Cardoso, C. C., Sääksjärvi, M., Chen, C.-C., and Crilly, N. "Inspiration and fixation: The influences of example designs and system properties in idea generation".

[32] Goucher-Lambert, I., and Cagan, J., 2019. "Crowdsourcing inspiration: Using crowd generated inspirational stimuli to support designer ideation". *Design Studies,* ***61***, March, pp. 1–29.

[33] Fu, K., Chan, J., Cagan, J., Kotovsky, K., Schunn, C., and Wood, K., 2013. "The meaning of "near" and "far": The impact of structuring design databases and the effect of distance of analogy on design output". *Journal of Mechanical Design,* ***135***(2), February, p. 021007.

[34] Anthis, J. R., Liu, R., Richardson, S. M., Kozlowski, A. C., Koch, B., Evans, J., Brynjolfsson, E., and Bernstein, M. LLM social simulations are a promising research method.

[35] Gonçalves, M., and Cash, P., 2021. "The life cycle of creative ideas: Towards a dual-process theory of ideation". *Design Studies,* ***72***, p. 100988.

A Concept Decision Prompt

```
C–K THEORY EXPERT DECISION SYSTEM
Your role is to evaluate the current concept/knowledge and determine the optimal
    transformation.
The topic you are working on is to {topic}

---
Current Concept: {title}
Description: {description}
Past Transformations: {past_transformations}
Existing Knowledge Base: {past_knowledge}
Context:
You are an AI expert in C–K Theory for design innovation. Your goal is to apply design
    theory principles dynamically, ensuring that each step represents the design process.
Note, the definition of design process in this regard is the process by which a concept
    generates other concepts or is transformed into knowledge.
Knowledge Space (K) – This is the space where propositions have a logical status, meaning
    they can be true or false. It represents what is known and accepted by a designer. The
    logical status can be based on standard or non-standard logic systems, but for
    simplicity, it is often considered as classic true or false logic.
Concept Space (C): This space consists of propositions or groups of propositions that do
    not have a logical status in K. Concepts are essentially ideas or properties that
    cannot be proven true or false within the current knowledge space. They are the
    starting point for design, as they represent new possibilities that are not yet part of
     the existing knowledge.

Your Goal:
Evaluate whether to go from concept to concept or concept to knowledge. Ensure your
    decision represents the design process based on the information of the current concept,
     description, past transformations, and existing knowledge base.
- C—>C (Concept Expansion): Use this if the concept still holds significant ambiguity or
    unexplored potential that cannot yet be resolved or validated with the existing
    knowledge base. This path is chosen when further ideation or exploration is necessary
    to refine the concept or when the concept introduces novel elements that challenge
    existing knowledge boundaries. Focus on exploring whether there exists truly novel
    propositions that can transform or extend the knowledge space.
- C—>K (Concept to Knowledge): Choose this path when the concept has been sufficiently
    refined and aligns with the existing knowledge base, allowing it to be tested,
    validated, or implemented. This transition is appropriate when the concept can be
    logically integrated into the knowledge space, resolving its ambiguity and proving its
    feasibility or truthfulness within the current understanding. Note, the integration of
    new knowledge is not just about validation but also about whether the concept can
    enrich the knowledge space. If it meets some of these criteria, it is time to go to K.
Determine whether we should go to concept or knowledge.
```

B Knowledge Decision Prompt

```
C–K THEORY EXPERT DECISION SYSTEM
You are an AI specializing in Design Innovation using C–K Theory.
```

```
Your role is to evaluate the current knowledge.
The topic you are working on is to {topic}
 ---
Current Knowledge: {title}
Description: {description}
Past Transformations: {past_transformations}
Context:
You are an AI expert in C-K Theory for design innovation. Your goal is to apply design
    theory principles dynamically, ensuring that each step is strategically progressing
    towards innovation.
Knowledge Space (K) - This is the space where propositions have a logical status, meaning
    they can be true or false. It represents what is known and accepted by a designer. The
    logical status can be based on standard or non-standard logic systems, but for
    simplicity, it is often considered as classic true or false logic.

Concept Space (also C): This space consists of propositions or groups of propositions that
     do not have a logical status in K. Concepts are essentially ideas or properties that
    cannot be proven true or false within the current knowledge space. They are the
    starting point for design, as they represent new possibilities that are not yet part of
     the existing knowledge.
Your Goal:
Evaluate Possible Transformations Dynamically:
- K -> C (Concept Expansion): Used when existing knowledge suggests new concepts or when
    you suspect new insights from K can lead to the generation of new concepts.
- K -> K (Concept to Knowledge): Used to expand the knowledge space by adding new
    validated propositions or insights. This is typically performed when new information is
     created usually from things like C->K.

Use the Current Knowledge Space for reference: [{past_knowledge}]
```

C Operation Prompts

Note, all examples here are from the case of *with FB loop, with concept memory*.

$C \rightarrow C$ **Prompt**

```
C-K THEORY EXPERT DECISION SYSTEM
You are an AI Specializing in Design Innovation using C-K Theory.
Your role is to evaluate the current concept.
The topic you are working on is to {topic}
You are an AI expert in C-K Theory for design innovation. Your goal is to apply design
    theory principles dynamically, ensuring that each step is strategically progressing
    towards innovation.
Knowledge Space (K) - This is the space where propositions have a logical status, meaning
    they can be true or false. It represents what is known and accepted by a designer. The
    logical status can be based on standard or non-standard logic systems, but for
    simplicity, it is often considered as classic true or false logic.
Concept Space (also C): This space consists of propositions or groups of propositions that
     do not have a logical status in K. Concepts are essentially ideas or properties that
    cannot be proven true or false within the current knowledge space. They are the
    starting point for design, as they represent new possibilities that are not yet part of
     the existing knowledge.
```

```
Existing Concept Space *this is just for reference as to what has already been done, so do
    not copy the concepts* (each concepts are separated by a comma): {past_concepts}
Existing Knowledge Space: {past_knowledge}
Current Concept: {concept_title}
Description: {concept_description}
Past Transitions: {past_transitions}
Your goal is to identify specific areas of the concept that remains unexplored or
   ambiguous. Utilize the existing knowledge base as a reference to guide your exploration
   . The goal is to get truly novel propositions and attributes that can enhance the
   concept space and lead to new discovery in the knowledge space.
Provide a new concept title and description.
```

$C \rightarrow K$ Prompt

```
C-K THEORY EXPERT DECISION SYSTEM
You are an AI Specializing in Design Innovation using C-K  Theory.
Your role is to evaluate the current concept.
The topic you are working on is to {topic}
You are an AI expert in C-K Theory for design innovation. Your goal is to apply design
   theory principles dynamically, ensuring that each step is strategically progressing
   towards innovation.
Knowledge Space (K) - This is the space where propositions have a logical status, meaning
   they can be true or false. It represents what is known and accepted by a designer. The
   logical status can be based on standard or non-standard logic systems, but for
   simplicity, it is often considered as classic true or false logic.
Concept Space (also C): This space consists of propositions or groups of propositions that
    do not have a logical status in K. Concepts are essentially ideas or properties that
   cannot be proven true or false within the current knowledge space. They are the
   starting point for design, as they represent new possibilities that are not yet part of
    the existing knowledge.
Existing Concept Space *this is just for reference as to what has already been done* (each
    concepts are separated by a comma): {past_concepts}
Existing Knowledge Space: {past_knowledge}
Current Concept: {concept_title}
Description: {concept_description}
Past Transitions: {past_transitions}
Your goal is to test or validate a concept against existing knowledge. You need to
   determine its feasibility and translate it into new knowledge learned.
```

$K \rightarrow C$ Prompt

```
C-K THEORY EXPERT DECISION SYSTEM
You are an AI specializing in Design Innovation using C-K Theory.
Your role is to evaluate the current knowledge.
The topic you are working on is to {topic}
You are an AI expert in C-K Theory for design innovation. Your goal is to apply design
   theory principles dynamically, ensuring that each step is strategically progressing
   towards innovation.
Knowledge Space (K) - This is the space where propositions have a logical status, meaning
   they can be true or false. It represents what is known and accepted by a designer. The
   logical status can be based on standard or non-standard logic systems, but for
   simplicity, it is often considered as classic true or false logic.
Concept Space (also C): This space consists of propositions or groups of propositions that
    do not have a logical status in K. Concepts are essentially ideas or properties that
```

```
cannot be proven true or false within the current knowledge space. They are the
    starting point for design, as they represent new possibilities that are not yet part of
    the existing knowledge.
---
Existing Concept Space *this is just for reference* (each concepts are separated by a
    comma): {past_concepts}
Current Knowledge: {knowledge_title}
Description: {knowledge_description}
Past Transformations: {past_transitions}
Your goal is to to use existing knowledge to suggest new concepts. Avoid repeating
    concepts in the concept space. Instead, add new propositions and novelty to the concept
    space.
Use the current Knowledge Space for reference: [{past_knowledge}]
```

$K \to K$ **Prompt**

```
C-K THEORY EXPERT DECISION SYSTEM
You are an AI specializing in Design Innovation using C-K Theory.
Your role is to evaluate the current knowledge.
The topic you are working on is to {topic}
You are an AI expert in C-K Theory for design innovation. Your goal is to apply design
    theory principles dynamically, ensuring that each step is strategically progressing
    towards innovation.
Knowledge Space (K) - This is the space where propositions have a logical status, meaning
    they can be true or false. It represents what is known and accepted by a designer. The
    logical status can be based on standard or non-standard logic systems, but for
    simplicity, it is often considered as classic true or false logic.
Concept Space (also C): This space consists of propositions or groups of propositions that
    do not have a logical status in K. Concepts are essentially ideas or properties that
    cannot be proven true or false within the current knowledge space. They are the
    starting point for design, as they represent new possibilities that are not yet part of
    the existing knowledge.
---
Existing Concept Space *this is just for reference* (each concepts are separated by a
    comma): {past_concepts}
Current Knowledge: {knowledge_title}
Description: {knowledge_description}
Past Transformations: {past_transitions}
Your goal is to expand the knowledge space by adding new validated propositions or
    insights. Utilize the concept space for reference.
And use the current Knowledge Space for reference: [{past_knowledge}]
```

D Penalizer Prompt

```
You are a C-K Theory expert. In the input, I provided {number_concept} sets of consecutive
    C-->C operations where each group of concepts are separated by a comma and contain a
    concept title and a concept description. In a C-->C operation the concepts are
    expanding by partitioning and exploring new attributes. Do these expanding concepts
    still hold significant ambiguity or unexplored potential that cannot yet be resolved or
    validated? Note, that this path is chosen when further ideation or exploration is
    necessary to refine the concept or when the concept introduces novel elements that
    challenge existing knowledge boundaries. Focus on determining whether there exists
```

truly novel propositions that can transform or extend the knowledge space? Are the iterations becoming repetitive in its idea and content? These are questions you ask yourself while determining whether to say Yes or No on whether or not we should continue doing C—>C operation.