Generalist Generative Agent

Open-ended design exploration with large language models

Anton Savov[1], Angela Yoo[2], CheWei Lin[3] and Benjamin Dillenburger[4]
[1,2,3,4]*Digital Building Technologies, ETH Zurich.*
[1]*Center for Augmented Computational Design in Architecture, Engineering and Construction (Design++), ETH Zurich.*
[1]*savov@arch.ethz.ch, ORCID: 0000-0002-5244-5285*
[2]*yoo@arch.ethz.ch, ORCID: 0009-0003-8864-9751*
[3]*weilin@arch.ethz.ch, ORCID: 0009-0009-9775-4960*
[4]*dillenburger@arch.ethz.ch, ORCID: 0000-0002-5153-2985*

Abstract. Architects often navigate ambiguity in early-stage design by using metaphors and conceptual models to transform abstract ideas into architectural forms . However, current computational tools struggle with such exploratory processes due to narrowly defined design spaces. This paper investigates whether Large Language Models (LLMs) can offer an alternative generative paradigm by interpreting human intent and translating it into actionable design logic. We propose an Agentic AI framework in which LLM agents interpret metaphors, formulate design tasks, and generate procedural 3D models. Using this framework, we produced 1,000 procedural designs and 4,000 images based on 20 metaphors to demonstrate the emergent capabilities of LLMs for creating architecturally relevant conceptual models. Our findings suggest that LLMs effectively engage with ambiguity, delivering diverse, meaningful outputs with notable potential for early-phase design. We discuss the strengths and shortcomings of the AI agents within the framework and suggest ways to extend their capacity for tackling open-ended design challenges, thereby enhancing their relevance in architectural practice.

Keywords. agentic AI, large language models, generative architectural design, multi-agent framework, design synthesis

1. Introduction

Architects are skilled generalists who excel at using abstract concepts to synthesise a project's value system into architectural form. In early-phase design, they often draw on metaphors and conceptual models to navigate the ambiguity and complexity inherent in open-ended challenges. While existing computational design tools tackle tasks ranging from simulation-based form-finding to data-driven form synthesis, current methods often operate within narrowly encoded goals. Could there be an

ARCHITECTURAL INFORMATICS, Proceedings of the 30th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2025, Volume 1, 193-202.

alternative generative design paradigm that can engage with ambiguity and the intuitive, exploratory nature of architectural ideation?

Large Language Models (LLMs) are making an impact across various fields by enabling systems capable of interpreting human intent and transforming it into domain-specific actionable logic. This suggests significant potential for LLMs to augment early-phase design exploration. In this paper, we investigate this hypothesis by:

- Demonstrating the emergent capabilities of LLMs in interpreting design intent and generating architecturally relevant conceptual models (Fig. 1)
- Proposing an Agentic AI framework for conceptual architectural design with components resembling steps in the design process. LLM agents are tasked with interpreting metaphors, formulating design tasks, and generating procedural 3D models for further refinement (Fig. 2 and Sec. 4).
- Offer a discussion of the strengths and shortcomings across the framework's AI agents (Sec. 5) and offer directions on how LLMs can be extended to enhance their relevance to architectural practice (Sec. 6).

By bridging conceptual ideas with procedural geometry, a 'generalist generative agent' transforms computational tools into versatile collaborators for open-ended design challenges while making them more accessible through natural language interaction.

2. Motivation

Architects often begin designing amid ambiguity, using metaphors and conceptual models to frame ideas and guide decisions. Metaphors can act as key design drivers or 'primary generators,' offering a conceptual framework that directs spatial exploration (Caballero-Rodriguez, 2013). Design drivers help reduce complexity by focusing on core value judgments rather than exhaustive requirements (Darke, 1979).

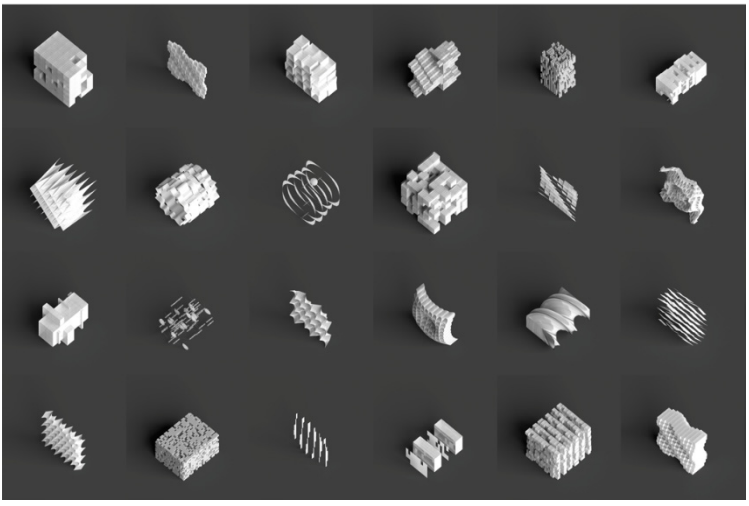

Figure 1. Concept 3D models generated by the LLM-enabled agentic framework showcasing diverse formal interpretations of design tasks derived from metaphors. Top 24 selected by a human architect.

Conceptual models complement this by translating abstract ideas into tangible forms. Unlike scale models, which prioritise accuracy, conceptual models are quick, gestural, and abstract, highlighting themes such as structure, space, light, or movement while sparking associations and imagination (Holtrop et al., 2011; Morris, 2006). We aim to explore how LLMs can emulate the interpretive power of metaphors and conceptual models in early-phase design, bridging the gap between open-ended thinking and computational design synthesis.

3. State of the Art

3.1. DESIGN SYNTHESIS

Existing computational techniques improve efficiency in layout generation (Weber et al., 2022), structural and environmental optimisation (Stieler et al., 2022), and probabilistic 3D model generation (Dai, 2023). However, these methods lack generalisability: they optimise a single predefined concept rather than generating diverse alternatives (Bolan, 2018). This shortfall limits their use in early conceptual stages when architects explore multiple diverse ideas (Dorst & Dijkhuis, 1995).

3.2. LARGE LANGUAGE MODELS AS GENERALIST AGENTS

LLMs have demonstrated remarkable capabilities as generalist tools, excelling in tasks like text and code generation, spatial reasoning, and object arrangement (Bubeck et al., 2023; B. Chen et al., 2024; Sharma, 2023). Agentic AI harnesses these strengths by using LLMs to interpret high-level intent and act in complex, unstructured scenarios (Park et al., 2023; Schick et al., 2023; Wang et al., 2023) — mirroring how designers work with partial intent. However, in architecture, LLMs are primarily deployed to translate task-specific instructions (e.g., “add a box” or “shift the building 10 meters”) into layouts, visualisations, 3D models, and BIM data (Q. Chen et al., 2020; Galanos et al., 2023; Leng et al., 2023). Most approaches treat LLMs as interfaces rather than true generators (Makatura et al., 2023). We propose exploring their capacity to transform high-level design concepts into architecturally relevant procedural models, unleashing their generative power for design synthesis.

4. Methodology

To investigate the generative capabilities of LLMs in the early stages of architectural ideation, we propose a structured multi-agent framework (Fig. 2) and use it to generate about 1000 concept models, which we then review and discuss. A hand-picked selection by an architect is used to control and validate the framework stages. Our framework comprises four agents—Metaphor, Interpretation, Modelling, and Evaluation—each addressing a distinct aspect of design ideation.

First, the Metaphor Agent generates a metaphor and key descriptive traits that serve as a design driver. It is instructed to use participle adjectives (“rippled”) and nouns (“grid”) to evoke spatial or formal qualities.

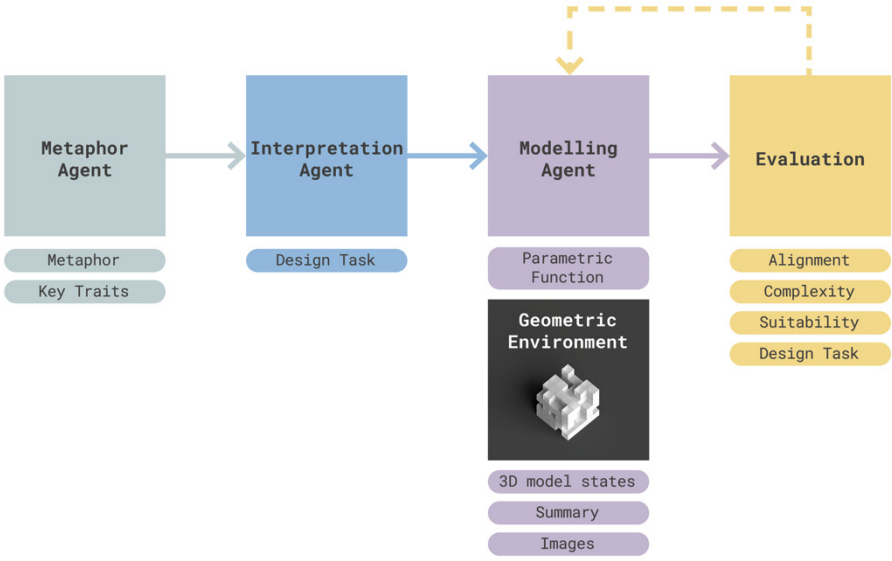

Figure 2. Overview of the proposed Agentic AI framework for conceptual architectural design, showing how the four main agents—Metaphor, Interpretation, Modelling, and Evaluation—enable open-ended design exploration. The Metaphor Agent generates a design driver, the Interpretation Agent formulates a task from it, and the Modelling Agent creates procedural 3D models. Finally, the Evaluation Agent assesses each model's alignment with the original intent.

Second, the Interpretation Agent takes a metaphor, outlines the requirements for an architectural concept model, and generates a concise design task. It iterates five times, producing diverse outputs while keeping history to maintain context across iterations.

Third, the Modelling Agent receives the metaphor, its key traits, and, optionally, the design task. It then generates procedural models as Python functions for RhinoCommon/Grasshopper, each accompanied by five sample calls that illustrate parameter variations. An automated loop runs these scripts in Rhino/Grasshopper, instantiating and saving the resulting models. For each metaphor, the agent operates in three contexts: (1) zero-shot (metaphor + traits), (2) zero-shot (metaphor + traits + design task), and (3) few-shot (metaphor + traits + design task + previously generated code). All model instances are exported as OBJ files and rendered in Blender in axonometric projection with neutral colours.

Finally, the Evaluation Agent uses a Vision Transformer (ViT) to assess each rendered model on four criteria — (1) Metaphor Alignment, (2) Conceptual Strength, (3) Geometric Complexity, and (4) Adherence to the Design Task—all rated on a 1–5 scale. It processes PNG renders and corresponding JSON metadata describing the design driver and task. It then outputs these ratings as float values in a CSV file.

To facilitate exchange, each agent's outputs are stored as JSON files, while procedural functions are saved as Grasshopper-compatible .py files, accompanied by an LLM-generated Markdown summary of what it does. All agents are implemented in Python using the LangChain (Chase, 2024) and OpenAI (OpenAI, 2024) libraries, with GPT-4o (via OpenAI's API) as the primary LLM and ViT.

5. Results and Discussion

The experiment produced 20 metaphors and 100 design tasks, i.e. five per metaphor. A total of 1,100 procedural designs for concept models were attempted, with 992 successfully generated models, resulting in a 90% success rate. This number comes from 55 attempts per metaphor across the three contexts: 5 for zero-shot with metaphor only, and 25 each for zero-shot with design task and few-shot. The procedural models produced 3,992 valid OBJ files, which were rendered (out of an expected 4,960—five per design), achieving an 80% success rate at this stage. Data generation took approximately one minute per design, totalling 18.5 hours. Evaluation required about 10 seconds per image, amounting to approximately 11 hours. Additionally, we asked an architect to manually select images for their conceptual strength and architectural relevance, giving us a total of 137 selections.

5.1. METAPHOR AGENT

All 20 metaphors used in this experiment are listed in Fig. 7. The Metaphor Agent outputs revealed that LLMs struggle to generate architecturally potent metaphors, emphasising the need for human input from architects and domain experts. Generated models highlight the importance of strong design drivers in achieving architectural relevance. Formally evocative metaphors like "rippled grid" produce diverse, compelling outputs even without additional context, while weak metaphors such as "split void" lead to repetitive or irrelevant results across all three contexts. These findings reaffirm that the selection of effective design drivers, rich in formal interpretation, is critical and ultimately dependent on the architect's agency.

5.2. INTERPRETATION AGENT

The Interpretation Agent translates a metaphor into a succinct design task, guiding the creation of an architectural concept model. By turning intent into actionable logic, LLMs excel at generating diverse tasks (Fig. 3). Including the design task significantly improves success rates, procedural diversity, and architectural suitability (Fig. 4). Consequently, design-task generation proves an essential step in the framework, and LLMs handle it effectively.

5.3. MODELLING AGENT

The Modelling Agent plays a pivotal role in producing quick, imaginative architectural models, effectively opening avenues for creative exploration. Its ability to generate diversity across different runs with the same context is notable, even without incorporating prior examples (Fig. 4). A typical generated parametric function is 45-80 lines of code and the agent benefits from the RhinoCommon knowledge learned by the LLM in training. Some models stand out for their architectural relevance, while others intrigue due to their unique procedural approach.

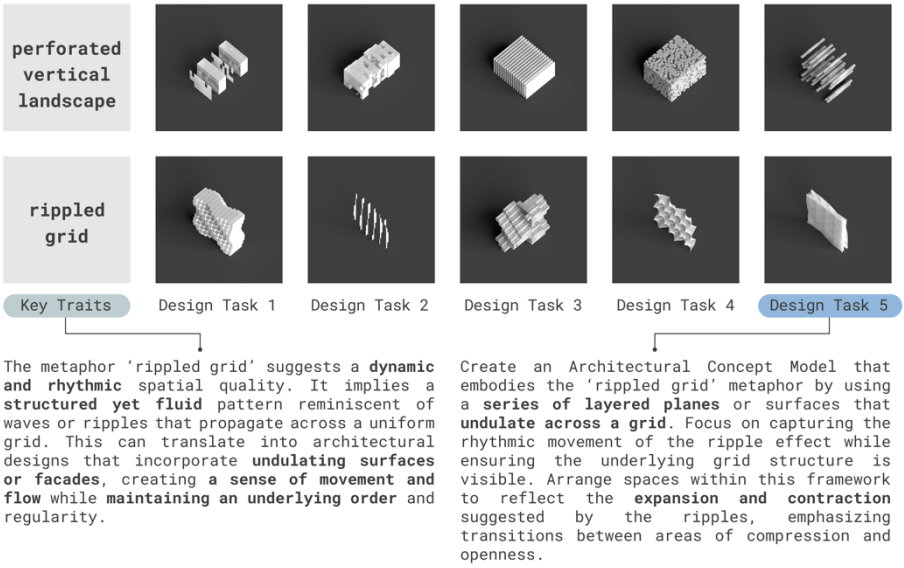

Figure 3. 3D model outputs across five design tasks for two metaphors, generated from key traits and the corresponding design tasks. The LLM translates concepts like "rippled grid," described as a "structured yet fluid pattern," into instructions such as "a series of layered planes" exhibiting "expansion and contraction," demonstrating its ability to align geometric entities with design drivers.

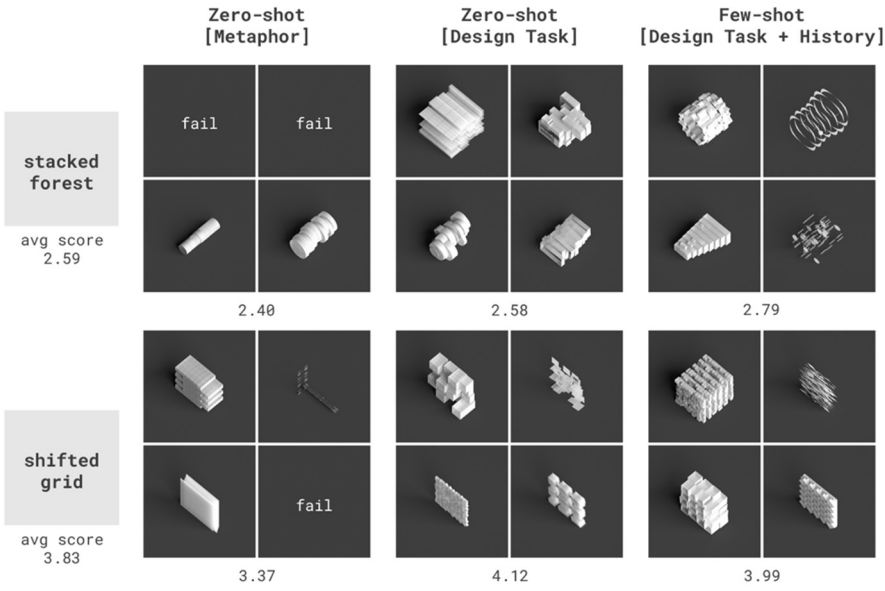

Figure 4. The middle and right columns (design task provided) demonstrate higher success rates, greater procedural diversity, and improved architectural suitability compared to the left column (no design task). These improvements, particularly evident for weak (above) and medium-strength (below) metaphors, underscore the importance of incorporating a design task in guiding generation.

	Zero Shot [Only Metaphor]	Zero Shot [With Design Task]	Few Shot [With History]
Attempted Designs	5	25 (5 per task)	25 (5 per task)
Success Rate (%)	87%	86%	95%

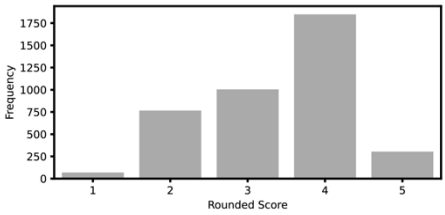

Figure 5. Left: Success rates for models generated under three contexts. Right: The average scores assigned by the Vision Transformer (ViT) to all 3,992 images follow a roughly normal distribution, indicating that the Evaluation Agent effectively distinguishes different levels of quality.

Context proved to be quite influential, with the overall success rate improving from 86% in zero-shot scenarios to 95% with few-shot examples (Fig. 5 Left). Notably, medium-strength metaphors like "box in a cloud" and "stacked forests" improve on success rates, architectural relevance and diversity when combined with design tasks and few-shot examples.

Improving the geometric awareness of the LLM can be explored with neuro-symbolic approaches, which combine the quick associative capabilities of LLMs with domain-specific symbolic engines. These hybrid systems enhance the functionality of LLMs across various applications, from solving geometric math problems and robot path planning to physics calculations and explorations of simulated worlds (Liu et al., 2022; Ma et al., 2023; Trinh et al., 2024; Wang et al., 2023).

5.4. EVALUATION AGENT

In our agentic framework, the Evaluation Agent provides feedback to the Modelling Agent on how well the generated models meet the objectives set by the Metaphor and Interpretation Agents—or by the human user. A core question is whether a Vision Transformer (ViT) can reliably evaluate these concept models in alignment with human judgment. As shown in Fig. 5, the roughly normal distribution of ViT scores indicates good discrimination across varying levels of concept quality. Fig. 6 illustrates representative low- and high-scoring examples. When scores are grouped by metaphor (Fig. 7 top), the ViT results mirror our observations of weak, medium, and strong meta-

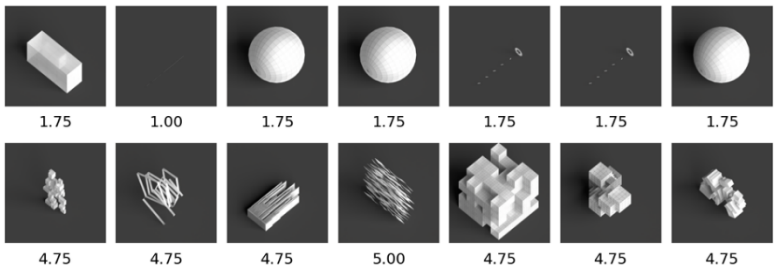

Figure 6. Representative random samples of the lowest- and highest-scoring concept images. Lower-scoring images (top row) typically show weaker designs, while those with the highest scores (bottom row) correlate to higher-quality concepts. A visual comparison between the higher-scoring images and the manually selected images in Fig. 1 reveals similar diversity, complexity, and formal traits.

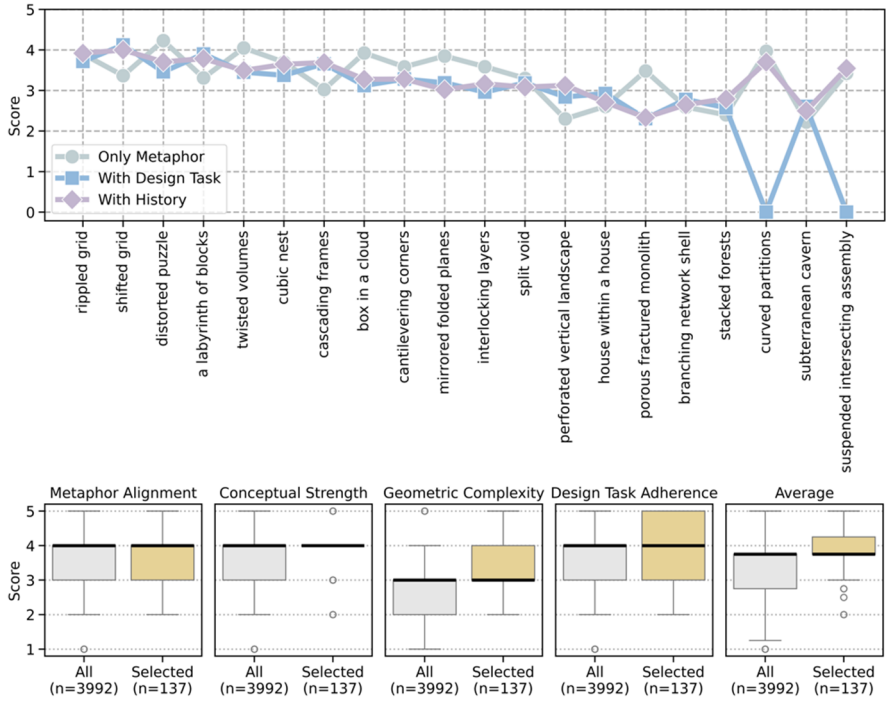

Figure 7. Top: Context-specific average scores by metaphor, reflecting the differences in formal potential among these conceptual drivers—stronger metaphors on the left, weaker on the right. Each metaphor score comprises around 200 evaluated images. Bottom: Boxplot comparison of the ViT scores for each criterion across all 3,992 images versus a curated set of 137 images selected by architects. The human-chosen images uniformly exhibit higher conceptual strength and overall scores, underscoring the alignment between expert judgment and the Evaluation Agent's ratings.

phors, while higher ViT scores correlate with the images manually selected by architects (Fig. 7 bottom).

These findings suggest the ViT-based approach effectively distinguishes between weaker and stronger architectural concepts, aligning closely with human judgment. However, it cannot offer quantitative assessments or fully replace the nuanced intuition of experienced designers. Future enhancements may include fine-tuning the ViT on human-rated images and integrating metrics based on structured geometric representations and computational analyses to move beyond purely visual assessment.

6. Outlook & Conclusion

In this work, we:

- Proposed and prototyped an LLM-enabled agentic framework for generating 3D architectural concept models;
- Demonstrated and discussed the emergent capabilities of LLMs to generate architectural models using 1,000 generated procedural designs and 4,000 images.

A key finding in our exploration is the diversity produced by the "Generalist Generative Agent" framework, particularly through the Modelling Agent. While this diversity enhances creativity and design possibilities, it poses a challenge: how can designers navigate the multitude of variations? Future work may include:

- Extending the framework with Retrieval-Augmented Generation (RAG), enabling designers to visually select previously generated models that align with their sensibilities without specifying explicit analytical criteria.
- Integrating iterative loops with history and evaluation to enhance the framework's effectiveness and extending support for decentralised structures with shared memory to enable agents to refine outputs iteratively.
- Incorporating human-in-the-loop methodologies to refine the models further and address qualitative aspects that LLMs cannot fully capture.

The proposed "Generalist Generative Agent" framework bridges open-ended design exploration with procedural modelling, paving the way for intuitive and adaptable AI-enabled design workflows. By integrating human intuition into the selection process, the vast diversity generated by AI agents can be directed in ways that are most relevant and inspiring to the designer.

Acknowledgements

This research is supported by an ETH Career Seed Award and a Hasler Stiftung Project Grant and is embedded within the Center for Augmented Computational Design in Architecture, Engineering, and Construction (Design++), ETH Zurich. ChatGPT (OpenAI, 2024) was used to improve the text flow of this paper, and Grammarly (Grammarly Inc., 2024) to correct spelling and grammar.

References

Bolan, R. S. (2018). Urban Planning's Philosophical Entanglements: The Rugged, Dialectical Path from Knowledge to Action. Routledge. https://doi.org/10.4324/9781315309217

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M. T., & Zhang, Y. (2023). Sparks of Artificial General Intelligence: Early experiments with GPT-4 (No. arXiv:2303.12712). arXiv. https://doi.org/10.48550/arXiv.2303.12712

Caballero-Rodriguez, R. (2013). From Design Generator to Rhetorical Device: Metaphor in Architectural Discourse. In A. Gerber & B. Patterson (Eds.), Metaphors in Architecture and Urbanism (pp. 89–104). transcript Verlag. https://doi.org/10.14361/transcript.9783839423721.89

Chase, H. (2024). LangChain [Python]. https://github.com/langchain-ai/langchain

Chen, B., Xu, Z., Kirmani, S., Ichter, B., Driess, D., Florence, P., Sadigh, D., Guibas, L., & Xia, F. (2024). SpatialVLM: Endowing Vision-Language Models with Spatial Reasoning Capabilities (No. arXiv:2401.12168). arXiv. https://doi.org/10.48550/arXiv.2401.12168

Chen, Q., Wu, Q., Tang, R., Wang, Y., Wang, S., & Tan, M. (2020). Intelligent Home 3D: Automatic 3D-House Design from Linguistic Descriptions Only (No. arXiv:2003.00397). arXiv. https://doi.org/10.48550/arXiv.2003.00397

Dai, A. (2023). Co-creation: Space Reconfiguration by Architect and Agent Simulation Based Machine Learning. In P. F. Yuan, H. Chai, C. Yan, K. Li, & T. Sun (Eds.), Hybrid

Intelligence (pp. 304–313). Springer Nature. https://doi.org/10.1007/978-981-19-8637-6_27
Darke, J. (1979). The primary generator and the design process. Design Studies, 1(1), 36–44. https://doi.org/10.1016/0142-694X(79)90027-9
Dorst, K., & Dijkhuis, J. (1995). Comparing paradigms for describing design activity. Design Studies, 16(2), 261–274. https://doi.org/10.1016/0142-694X(94)00012-3
Galanos, T., Liapis, A., & Yannakakis, G. N. (2023). Architext: Language-Driven Generative Architecture Design (No. arXiv:2303.07519). arXiv. https://doi.org/10.48550/arXiv.2303.07519
Holtrop, A., Princen, B., Teerds, H., Floris, J., & de Koning, K. (2011). Editorial. Models. The idea, the representation and the visionary. In Models. The Idea, the Representation and the Visionary (Vol. 84, pp. 20–23). https://oasejournal.nl/en/Issues/84/Editorial
Leng, S., Zhou, Y., Dupty, M. H., Lee, W. S., Joyce, S. C., & Lu, W. (2023). Tell2Design: A Dataset for Language-Guided Floor Plan Generation (No. arXiv:2311.15941). arXiv. https://doi.org/10.48550/arXiv.2311.15941
Liu, R., Wei, J., Gu, S. S., Wu, T.-Y., Vosoughi, S., Cui, C., Zhou, D., & Dai, A. M. (2022). Mind's Eye: Grounded Language Model Reasoning through Simulation (No. arXiv:2210.05359). arXiv. https://doi.org/10.48550/arXiv.2210.05359
Ma, Y. J., Liang, W., Wang, G., Huang, D.-A., Bastani, O., Jayaraman, D., Zhu, Y., Fan, L., & Anandkumar, A. (2023). Eureka: Human-Level Reward Design via Coding Large Language Models (No. arXiv:2310.12931). arXiv. https://doi.org/10.48550/arXiv.2310.12931
Makatura, L., Foshey, M., Wang, B., HähnLein, F., Ma, P., Deng, B., Tjandrasuwita, M., Spielberg, A., Owens, C. E., Chen, P. Y., Zhao, A., Zhu, A., Norton, W. J., Gu, E., Jacob, J., Li, Y., Schulz, A., & Matusik, W. (2023). How Can Large Language Models Help Humans in Design and Manufacturing? (No. arXiv:2307.14377). arXiv. https://doi.org/10.48550/arXiv.2307.14377
Morris, M. (2006). Models: Architecture and the miniature. Wiley-Academy.
OpenAI. (2024). Openai/openai-python [Python]. OpenAI. https://github.com/openai/openai-python
Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior (No. arXiv:2304.03442). arXiv. https://doi.org/10.48550/arXiv.2304.03442
Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., & Scialom, T. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools (No. arXiv:2302.04761). arXiv. https://doi.org/10.48550/arXiv.2302.04761
Sharma, M. (2023). Exploring and Improving the Spatial Reasoning Abilities of Large Language Models (No. arXiv:2312.01054). arXiv. https://doi.org/10.48550/arXiv.2312.01054
Stieler, D., Schwinn, T., Leder, S., Maierhofer, M., Kannenberg, F., & Menges, A. (2022). Agent-based modeling and simulation in architecture. Automation in Construction, 141, 104426. https://doi.org/10.1016/j.autcon.2022.104426
Trinh, T. H., Wu, Y., Le, Q. V., He, H., & Luong, T. (2024). Solving olympiad geometry without human demonstrations. Nature, 625(7995), 476–482. https://doi.org/10.1038/s41586-023-06747-5
Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., & Anandkumar, A. (2023). Voyager: An Open-Ended Embodied Agent with Large Language Models (No. arXiv:2305.16291). arXiv. https://doi.org/10.48550/arXiv.2305.16291
Weber, R. E., Mueller, C., & Reinhart, C. (2022). Automated floorplan generation in architectural design: A review of methods and applications. Automation in Construction, 140, 104385. https://doi.org/10.1016/j.autcon.2022.104385