Retrieval-Augmented Generation in Engineering Design

Preprint · August 2024

DOI: 10.13140/RG.2.2.18595.26406

1 author:

Debiprasad Ghosh
Larsen & Toubro (L&T) Construction

Retrieval-Augmented Generation in Engineering Design

Debi Prasad Ghosh, debiprasadghosh@lntecc.com

Design Automation Team

Engineering Design & Research Center, Larsen & Toubro Construction (M&M) Kolkata, India

Abstract: This paper explores the application of Retrieval-Augmented Generation (RAG) in engineering design, examining its potential to revolutionize the design process through advanced computational techniques and artificial intelligence. We investigate the core components of the RAG framework—Retrieve, Analyze, and Generate—and their contribution to enhanced efficiency, improved accuracy, and increased innovation in engineering design. The study discusses advanced techniques that augment RAG capabilities, including Physics-Informed Geometry-Aware Neural Operators (PI-GANO), Geometric Operators (GOs), and Multi-Fidelity Cross-Validation (MFCV). These methods significantly improve design simulation accuracy, reduce surrogate model overfitting, and optimize computational resource utilization. We explore emerging technologies facilitating RAG implementation, such as Azure Intelligent Document, Azure OpenAI, LlamaIndex, and LangChain, demonstrating their role in automating data extraction, enhancing natural language processing, and enabling efficient information retrieval in engineering contexts. The paper addresses challenges in RAG adoption, including data quality, computational requirements, and the need for interpretability in AI-driven design decisions.

This comprehensive exploration of RAG in engineering design contributes to the intersection of artificial intelligence and engineering, offering insights for researchers and practitioners. As RAG evolves, it promises to reshape engineering design, enabling faster, more accurate, and more creative solutions to complex challenges.

1. Introduction

Engineering design has always been at the forefront of technological innovation, constantly evolving to meet the challenges of an increasingly complex world. As the demands for more efficient, sustainable, and innovative solutions grow, so does the need for advanced methodologies that can leverage the vast amounts of available data and knowledge. In recent years, the integration of artificial intelligence (AI) and advanced computational techniques has opened new frontiers in design methodologies. Among these emerging approaches, Retrieval-Augmented Generation (RAG) stands out as a particularly promising framework for enhancing the design process.

RAG combines the power of large-scale information retrieval with sophisticated analysis and generative capabilities. This synergy allows engineers to leverage vast amounts of existing knowledge while simultaneously pushing the boundaries of innovation. By automating and optimizing various aspects of the design process, RAG has the potential to significantly reduce development time, improve design quality, and foster creativity in ways previously unattainable.

The core components of the RAG framework—Retrieve, Analyze, and Generate—work in harmony to create a powerful tool for engineering design:

1. The Retrieve stage gathers relevant data from diverse sources, including historical design data, simulation results, and existing literature.
2. The Analyze phase evaluates this data to extract insights and identify patterns that inform the design process.
3. The Generate stage produces innovative design solutions based on the analysis of the retrieved data.

This paper aims to provide a comprehensive overview of RAG in the context of engineering design. We will explore the theoretical foundations of the RAG framework, examine its practical applications across different engineering domains, and discuss the tools and technologies that enable its implementation. Furthermore, we will analyze the benefits and challenges associated with RAG adoption and consider future directions for research and development in this rapidly evolving field.

By investigating emerging technologies such as Azure Intelligent Document, Azure OpenAI, LlamaIndex, and LangChain, we will demonstrate how these tools facilitate the implementation of RAG approaches in real-world engineering scenarios. Through a comprehensive analysis of current methodologies and case studies, we will showcase the significant impact of RAG on reducing project costs and improving design outcomes across various engineering domains, including structural engineering, mechanical design, aerospace engineering, and robotics.

As we delve into the intricacies of RAG and its applications, this paper seeks to illuminate the transformative potential of this approach in revolutionizing engineering design practices. By the end of this exploration, readers will gain a deep understanding of how RAG is shaping the future of engineering innovation and paving the way for more efficient, accurate, and creative design solutions.

1.1 Literature Survey: Retrieval-Augmented Generation in Engineering Design

Retrieval-Augmented Generation (RAG) has emerged as a powerful approach in engineering design, combining the strengths of information retrieval, data analysis, and generative techniques. This literature survey provides an overview of recent advancements in RAG and related technologies that are shaping the future of engineering design.

1.1.2 Advanced Techniques in RAG

1.1.2.1 Physics-Informed Neural Operators

Zhao et al. [1] introduced the Physics-Informed Geometry-Aware Neural Operator (PI-GANO), a method for solving parametric Partial Differential Equations (PDEs) with varying domain geometries. PI-GANO combines physics-informed learning with a geometry encoder, offering improved accuracy and efficiency without requiring large datasets.

Building on this, Zhao et al. [32] developed the Diffeomorphism Neural Operator (DNO), which maps different shapes onto a standard domain and then learns a neural operator to solve PDEs on this standard domain. This approach improves generalization to different shapes and parameters while reducing computational costs.

1.1.2.2 Geometric Operators and Surrogate Models

Khan et al. [3] introduced Physics-Informed Geometric Operators (GOs) to enhance the performance of surrogate, dimension reduction, and generative models in engineering design. GOs extract high-level geometric information and physics from shapes, improving model performance and reducing overfitting.

Renganathan and Carlson [4] proposed Multi-fidelity Cross-validation (MFCV), a method for improving surrogate models by combining multiple fidelity models into a single surrogate model. This approach offers more efficient use of computational resources and improved accuracy.

1.1.2.3 Bayesian Optimization and Uncertainty Quantification

Ahmadianshalchi et al. [86] introduced PAC-MOO, a Bayesian optimization method for handling constrained multi-objective optimization problems. This approach uses surrogate models for objectives and constraints to efficiently explore the design space.

Nemani et al. [80] provided a comprehensive overview of uncertainty quantification (UQ) for machine learning models, focusing on neural networks. The paper discusses various UQ techniques and their applications in engineering design and health prognostics.

1.1.3. AI-Driven Design and Optimization

1.1.3.1 Generative Models for Engineering Design

Chong et al. [2] proposed a method to improve the practicality of using text-to-image models for engineering design by incorporating CAD images into model prompts. This approach enhances design feasibility while maintaining creativity.

Fan et al. [66] introduced SA-ALAE, a deep generative model for creating realistic engineering designs. The model incorporates self-attention to capture long-range dependencies in engineering designs, enabling the generation of realistic and diverse engineering blueprints.

1.1.3.2 Optimization Algorithms

Lou et al. [24] introduced the Competitive Game Optimizer (CGO), a new optimization algorithm inspired by competitive game dynamics. CGO incorporates exploration and exploitation phases and has demonstrated effectiveness in UAV path planning and other engineering design problems.

Jiang and Luo [30] developed AutoTRIZ, a tool that combines the Theory of Inventive Problem Solving (TRIZ) with large language models to automate the design ideation process. AutoTRIZ offers a more accessible and efficient approach to innovation in engineering design.

1.1.3.3 Machine Learning in Design Processes

Wong et al. [7] introduced Prompt Evolution Design Optimization (PEDO), a method that uses natural language prompts to generate 3D designs and then evaluates and improves these designs using physics-based simulations and a vision-language model.

Park and Kang [9] proposed a Bayesian graph neural network (GNN) model for predicting engineering performance directly from mesh representations of CAD models. This approach improves accuracy in predicting engineering performance compared to other 3D representation methods.

1.1.4 Data-Driven Approaches and Datasets

1.1.4.1 Engineering Design Datasets

Hong et al. [8] introduced the DeepJEB dataset, a large-scale collection of 3D jet engine bracket designs and their corresponding structural analysis data. This dataset supports improved surrogate model performance and is compatible with complex models like graph neural networks.

Cobb et al. [75] created AircraftVerse, a massive dataset containing diverse aerial vehicle designs and their corresponding performance metrics. This dataset serves as a valuable resource for researchers developing AI-driven aircraft design tools and optimization techniques.

1.1.4.2 Data-Driven Design Methodologies

Yang et al. [96] proposed a framework for Data-Driven Intelligent Computational Design (DICD), utilizing deep learning to extract design features from historical data and generate novel designs. The framework provides a systematic roadmap for DICD implementation, including dataset building, feature engineering, and design optimization.

Picard et al. [77] presented a systematic approach to creating synthetic datasets for engineering design applications. The research provides guidelines for generating, annotating, and validating synthetic datasets, addressing the critical challenge of data availability in AI-driven engineering design.

1.1.5 Human-AI Collaboration in Engineering Design

1.1.5.1 Evaluating AI Performance in Design Tasks

Ege et al. [23] compared ChatGPT 4.0 to human engineering students in a design challenge. The study found that ChatGPT demonstrated strong concept generation abilities but struggled with problem-solving and decision-making. The research provides recommendations for effective human-AI collaboration in design.

Doris et al. [25] introduced DesignQA, a benchmark to evaluate how well large language models understand and apply engineering requirements from technical documents. The study establishes a foundation for improving AI-assisted engineering design processes.

1.1.5.2 Collaborative Design Tools

Zhu et al. [16] investigated the effectiveness of human-AI collaboration in feature engineering. The study developed a prototype tool to provide AI and human-generated feature suggestions to data scientists and conducted a user study to understand how users interact with and perceive AI and human-generated features.

Gmeiner et al. [91] investigated how engineers learn to use AI-based design tools, focusing on understanding the challenges designers face when collaborating with AI tools in a design process. The study highlights the need for improved AI tool design to facilitate effective human-AI collaboration.

1.1.6 Applications in Specific Engineering Domains

1.1.6.1 Aerospace Engineering

Yu et al. [26] introduced PFN4sBO, a method for optimizing complex engineering problems using Bayesian Optimization. The approach uses pre-trained transformers to quickly and accurately find optimal solutions, with potential applications in various engineering fields, including aerospace.

1.1.6.2 Mechanical Engineering

Nobari et al. [10] developed LInK, a method for designing mechanisms that combines deep learning (contrastive learning) with traditional optimization techniques. LInK offers significantly faster and more accurate optimization than previous methods, especially for complex problems.

1.1.6.3 Robotics and Control Systems

Tong et al. [5] focused on determining the initial shape of flexible rods for desired deformation under gravity. The research combines physics-based modeling with machine learning techniques, with potential applications in soft robotics and animation.

Kevian et al. [27] evaluated the capabilities of large language models in solving control engineering problems. The study developed a benchmark dataset (ControlBench) and compared various LLMs on undergraduate-level control problems.

1.1.7 Future Directions and Challenges

As the field of RAG in engineering design continues to evolve, several challenges and opportunities emerge:

1. Data Quality and Availability: Ensuring comprehensive and accurate datasets across all relevant engineering domains remains a challenge, as highlighted by Picard et al. [77].
2. Computational Resources: Advanced RAG techniques often require significant computational power, which may be a limitation for smaller organizations or resource-constrained projects.
3. Integration with Existing Workflows: Incorporating RAG methodologies into established engineering processes may require substantial changes to existing workflows and training for personnel.
4. Interpretability and Explainability: As RAG systems become more complex, ensuring the interpretability and explainability of design decisions becomes increasingly important, particularly in safety-critical applications.
5. Human-AI Collaboration: Improving the interface between AI systems and human engineers to leverage the strengths of both is an ongoing area of research, as demonstrated by studies like Gmeiner et al. [91].
6. Ethical Considerations: As AI plays an increasingly significant role in engineering design, addressing ethical considerations and ensuring responsible AI development becomes crucial, as discussed by Constantinides et al. [65].

1.1.8 Conclusion: Retrieval-Augmented Generation in Engineering Design

Retrieval-Augmented Generation is revolutionizing engineering design by enabling more efficient, accurate, and innovative design processes. The integration of physics-informed methods, advanced optimization techniques, and large-scale datasets is pushing the boundaries of what's possible in engineering design. As the field continues to evolve, addressing challenges related to data quality, computational resources, and human-AI collaboration will be crucial for realizing the full potential of RAG in engineering design.

1.2 Literature Review: Emerging Technologies in RAG for Engineering Design

1.2.1. Introduction: Emerging Technologies

Retrieval-Augmented Generation (RAG) has emerged as a powerful approach in engineering design, combining the strengths of information retrieval, data analysis, and generative techniques. This literature review focuses on four key technologies that are facilitating the implementation of RAG approaches in real-world engineering scenarios: Azure Intelligent Document [138], Azure OpenAI [139], LlamaIndex [140], and LangChain [141].

1.2.2. Azure Intelligent Document

Azure Intelligent Document is a cloud-based service that automates the extraction of relevant data from various document formats, streamlining the information retrieval process crucial for RAG systems.

1.2.2.1 Key Features and Capabilities

- Optical Character Recognition (OCR): Azure Intelligent Document incorporates advanced OCR capabilities to convert images and scanned documents into machine-readable text. This is particularly valuable in engineering contexts where historical design documents or hand-drawn sketches need to be digitized and analyzed.
- Form Recognition: The service can automatically identify and extract key-value pairs, tables, and structured data from forms and documents. This capability is essential for processing engineering specifications, bills of materials, and other structured documents commonly used in design processes.
- Custom Model Training: Azure Intelligent Document allows users to train custom models on domain-specific documents, enabling more accurate extraction of engineering-specific information.

1.2.2.2 Applications in Engineering Design

Knaster et al. [134] highlight the importance of efficient document processing in large-scale engineering projects like the International Fusion Materials Irradiation Facility (IFMIF). While not explicitly mentioning Azure Intelligent Document, their work underscores the need for tools that can efficiently process and extract information from complex engineering reports and design documents.

Harish and Prasad [136] present a method to convert 2D engineering drawings into 3D models using computer vision techniques. Although their work focuses on OpenCV, it demonstrates the potential for services like Azure Intelligent Document to automate the digitization and analysis of engineering drawings, a crucial step in implementing RAG systems for legacy design data.

1.2.3. Azure OpenAI

Azure OpenAI provides access to powerful language models for natural language processing and design ideation, enabling sophisticated text generation and completion tasks within RAG systems.

1.2.3.1 Key Features and Capabilities

- Text Generation and Completion: Azure OpenAI's language models can generate coherent and contextually relevant text, which is valuable for creating design descriptions, specifications, and even code snippets.
- Semantic Similarity Analysis: The service can perform semantic similarity comparisons, enabling the identification of related design concepts or principles across different documents or projects.
- Fine-tuning Capabilities: Azure OpenAI allows for fine-tuning of models on domain-specific data, enabling more accurate and relevant outputs for engineering applications.

1.2.3.2 Applications in Engineering Design

Ege et al. [23] compared ChatGPT 4.0 to human engineering students in a design challenge. While their study focused on ChatGPT specifically, it demonstrates the potential of large language models like those available through Azure OpenAI in engineering design tasks. The research highlights the strengths of AI models in concept generation and their current limitations in problem-solving and decision-making.

Wong et al. [7] introduced Prompt Evolution Design Optimization (PEDO), a method that uses natural language prompts to generate 3D designs. While not specifically using Azure OpenAI, their work illustrates how large language models can be integrated into the design process, suggesting potential applications for Azure OpenAI in engineering design ideation and optimization.

1.2.4. LlamaIndex

LlamaIndex is a data framework for large language models that enables efficient retrieval of relevant information from large datasets, making it particularly useful for implementing RAG systems in engineering design contexts.

1.2.4.1 Key Features and Capabilities

- Advanced Indexing Techniques: LlamaIndex employs sophisticated indexing methods to enable quick and accurate information retrieval from large, unstructured datasets.
- Query Processing: The framework offers customizable query processing capabilities, allowing users to tailor information retrieval to specific engineering domains.
- Integration with Various Data Sources: LlamaIndex can seamlessly integrate with multiple data sources and formats commonly used in engineering design, including text documents, CAD files, and simulation results.

1.2.4.2 Applications in Engineering Design

While LlamaIndex is a relatively new tool, its potential applications in engineering design are significant. Siddharth and Luo [48] investigated the underlying structure of engineering design knowledge as expressed in patent documents. Their work in creating knowledge graphs from patent text aligns well with LlamaIndex's capabilities in processing and indexing large volumes of technical documentation.

Hong et al. [8] introduced the DeepJEB dataset, a large-scale collection of 3D jet engine bracket designs and their corresponding structural analysis data. While not directly using LlamaIndex, their work highlights the need for efficient data management and retrieval systems in engineering design, a need that LlamaIndex is well-positioned to address.

1.2.5. LangChain

LangChain is a framework for developing applications powered by language models, facilitating the creation of interactive design tools and complex RAG systems.

1.2.5.1 Key Features and Capabilities

- Modular Architecture: LangChain provides a modular architecture for combining different language model capabilities, enabling the creation of sophisticated RAG systems tailored to engineering design needs.
- Context and Memory Management: The framework offers tools for managing context and memory in language model interactions, crucial for maintaining coherence in complex design tasks that may span multiple sessions or involve iterative refinement.
- Integration with External Tools: LangChain can integrate with external tools and data sources, enhancing the versatility of RAG applications in engineering contexts.

1.2.5.2 Applications in Engineering Design

Göpfert et al. [73] explored the potential of large language models to revolutionize the engineering design process. While not specifically mentioning LangChain, their work aligns with the framework's capabilities in integrating language models into complex workflows. They propose using LLMs to automate creative and reasoning aspects of the design process, which could be implemented using LangChain's modular architecture.

Rios et al. [68] investigated the integration of large language models and text-to-3D models for engineering design optimization. Their work demonstrates the potential for frameworks like LangChain to bridge the gap between natural language processing and geometric modeling in engineering design.

1.2.6. Conclusion: Emerging Technology

The emerging technologies of Azure Intelligent Document, Azure OpenAI, LlamaIndex, and LangChain offer powerful capabilities for implementing Retrieval-Augmented Generation systems in engineering design. While direct applications of these specific tools in engineering literature are still emerging, the reviewed studies demonstrate clear potential and need for such technologies.

Azure Intelligent Document addresses the crucial task of digitizing and extracting information from engineering documents. Azure OpenAI provides sophisticated language modeling capabilities that can enhance design ideation and natural language processing in engineering contexts. LlamaIndex offers efficient data management and retrieval, critical for handling the large datasets typical in engineering design. LangChain facilitates the integration of these capabilities into coherent, interactive systems tailored to engineering design workflows.

As these technologies continue to evolve and find wider adoption in the engineering community, we can expect to see more direct applications and case studies demonstrating their impact on reducing project costs, improving design outcomes, and fostering innovation across various engineering domains.

2. Methodology

This section details the methodological approach of Retrieval-Augmented Generation (RAG) in engineering design. We explore the core components of the RAG framework, discuss advanced techniques that enhance its capabilities, and examine the tools and technologies that facilitate its implementation.

2.1 The RAG Framework

The RAG framework consists of three primary components: Retrieve, Analyze, and Generate. Each component plays a crucial role in leveraging existing knowledge and creating innovative design solutions.

2.1.1 Retrieve

The Retrieve stage focuses on gathering relevant data and information from various sources. This process involves:

- Data Sources: Historical design data, design guidelines, design code provisions, simulation results, academic literature, patents, and industry standards are among the diverse sources of information utilized.
- Advanced Retrieval Techniques: Knowledge graphs and machine learning algorithms are employed to efficiently identify and retrieve pertinent information. These techniques can understand complex

relationships between different pieces of information and extract relevant data even from unstructured sources.
- Semantic Search: Unlike traditional keyword-based searches, semantic search understands the intent and contextual meaning of the query, allowing for more accurate and relevant information retrieval.

2.1.2 Analyze

In the Analyze phase, the retrieved data is evaluated to extract insights and identify patterns that inform the design process. This stage involves:

- Data Preprocessing: Cleaning, normalizing, and structuring the retrieved data to ensure consistency and compatibility for analysis.
- Feature Extraction: Identifying key features and parameters that are most relevant to the design problem at hand.
- Pattern Recognition: Utilizing machine learning algorithms to detect patterns, trends, and correlations within the data that may not be immediately apparent to human designers.
- Performance Metric Assessment: Evaluating design parameters against predefined performance metrics to gauge the potential effectiveness of different design approaches.

2.1.3 Generate

The Generate stage focuses on producing innovative design solutions based on the analysis of the retrieved data. This process includes:

- Generative Design Algorithms: Implementing algorithms such as Prompt Evolution Design Optimization (PEDO) and AutoTRIZ to create diverse and feasible design options.
- Constraint Satisfaction: Ensuring that generated designs meet all specified constraints and requirements of the project.
- Optimization: Refining generated designs to maximize performance across various metrics such as efficiency, cost-effectiveness, and sustainability.
- Design Iteration: Continuously improving designs through feedback loops and iterative refinement based on analysis results.

2.2 Advanced Techniques in RAG

Several advanced techniques have been developed to enhance the capabilities of the RAG framework:

2.2.1 Physics-Informed Geometry-Aware Neural Operator (PI-GANO)

PI-GANO combines physics-informed learning with a geometry encoder to handle both varying shapes and PDE parameters. This method:

- Improves the accuracy and efficiency of design simulations by incorporating physical laws directly into the learning process.
- Enables the model to understand and adapt to different geometric configurations, making it particularly useful for complex engineering designs.
- Reduces the reliance on extensive datasets by leveraging physical principles to guide the learning process.

2.2.2 Physics-Informed Geometric Operators (GOs)

GOs are designed to extract high-level geometric information and physics from shapes, enhancing the performance of surrogate models. Key features include:

- Efficient encoding of complex geometric information, allowing for faster and more accurate analysis of design variations.
- Reduction of overfitting by focusing on physically relevant geometric features.
- Improved generalization to new design configurations, enabling more robust design exploration.

2.2.3 Multi-Fidelity Cross-validation (MFCV)

MFCV is a technique that combines multiple fidelity models into a single surrogate model. This approach:

- Enables more efficient use of computational resources by strategically balancing high-fidelity and low-fidelity simulations.
- Improves overall accuracy by leveraging the strengths of different model fidelities.
- Facilitates faster design iterations while maintaining a high level of confidence in the results.

2.3 Tools and Technologies

The implementation of RAG in engineering design is facilitated by several cutting-edge tools and technologies:

2.3.1 Azure Intelligent Document

This cloud-based service automates the extraction of relevant data from various document formats, streamlining the information retrieval process. Features include:

- Optical Character Recognition (OCR) for converting images and scanned documents into machine-readable text.
- Natural Language Processing (NLP) capabilities for understanding document context and extracting key information.
- Integration with other Azure services for seamless data processing and analysis.

2.3.2 Azure OpenAI

Azure OpenAI provides access to powerful language models for natural language processing and design ideation. Key capabilities include:

- Text generation and completion for creating design descriptions and specifications.
- Semantic similarity analysis for comparing different design concepts.
- Integration with other Azure services for end-to-end AI-powered design workflows.

2.3.3 LlamaIndex

LlamaIndex is a data management tool that enables efficient retrieval of relevant information from large datasets. It offers:

- Advanced indexing techniques for quick and accurate information retrieval.
- Customizable query processing to tailor information retrieval to specific engineering domains.
- Seamless integration with various data sources and formats commonly used in engineering design.

2.3.4 LangChain

LangChain is a framework for building applications that utilize language models, facilitating the creation of interactive design tools. It provides:

- A modular architecture for combining different language model capabilities.

- Tools for managing context and memory in language model interactions, crucial for maintaining coherence in complex design tasks.
- Integration capabilities with external tools and data sources, enhancing the versatility of RAG applications.

2.3.5 Gradio

Gradio is an open-source library for creating user-friendly interfaces for machine learning models, enhancing the accessibility of RAG-based design tools. It offers:

- Rapid prototyping of interactive interfaces for design tools.
- Easy integration with various machine learning models and frameworks.
- Customizable components for visualizing design outputs and facilitating user interaction.

By leveraging these advanced techniques and tools, the RAG framework provides a powerful methodology for enhancing the engineering design process. The synergy between sophisticated data retrieval, analysis, and generation capabilities enables engineers to explore design spaces more efficiently and innovatively than ever before.

3. Results

4. Discussion

5. Conclusion

Retrieval-augmented generation represents a paradigm shift in engineering design, offering unprecedented opportunities for enhancing efficiency, accuracy, and innovation. By leveraging advanced AI techniques and computational tools, RAG enables engineers to tackle complex design challenges with greater speed and precision than ever before.

As the field continues to evolve, future research should focus on addressing the challenges identified, particularly in areas of data quality, computational efficiency, and seamless integration with existing engineering workflows. Additionally, the development of more intuitive user interfaces and the expansion of RAG techniques into emerging engineering domains present exciting opportunities for further innovation.

In conclusion, Retrieval-Augmented Generation stands poised to revolutionize the field of engineering design, driving innovation and efficiency across industries. As researchers and practitioners continue to refine and expand upon these techniques, the future of engineering design looks brighter and more innovative than ever before.

References

1. Zhong, W., & Meidani, H. (2024). Physics-Informed Geometry-Aware Neural Operator. Https://arxiv.org/abs/2408.01600
2. Chong, L., Rayan, J., Dow, S., Lykourentzou, I., & Ahmed, F. (2024). CAD-Prompted Generative Models: A Pathway to Feasible and Novel Engineering Designs. Https://arxiv.org/abs/2407.08675
3. Khan, S., Masood, Z., Usama, M., Kostas, K., & Kaklis, P. (2024). Physics-Informed Geometric Operators to Support Surrogate, Dimension Reduction and Generative Models for Engineering Design. Https://arxiv.org/abs/2407.07611
4. Renganathan, S. A., & Carlson, K. (2024). Multifidelity Cross-validation. Https://arxiv.org/abs/2407.01495
5. Tong, D., Hao, Z., & Huang, W. (2024). Inverse Design of Planar Clamped-Free Elastic Rods from Noisy Data. Https://arxiv.org/abs/2406.15166
6. Zhou, R., Yuan, C., Permenter, F., Zhang, Y., Arechiga, N., Klenk, M., & Ahmed, F. (2024). Bridging Design Gaps: A Parametric Data Completion Approach With Graph Guided Diffusion Models. Https://arxiv.org/abs/2406.11934
7. Wong, M., Rios, T., Menzel, S., & Ong, Y. S. (2024). Generative AI-based Prompt Evolution Engineering Design Optimization With Vision-Language Model. Https://arxiv.org/abs/2406.09143
8. Hong, S., Kwon, Y., Shin, D., Park, J., & Kang, N. (2024). DeepJEB: 3D Deep Learning-based Synthetic Jet Engine Bracket Dataset. Https://arxiv.org/abs/2406.09047
9. Park, J., & Kang, N. (2024). Bayesian Mesh Optimization for Graph Neural Networks to Enhance Engineering Performance Prediction. Https://arxiv.org/abs/2406.01996

10. Nobari, A. H., Srivastava, A., Gutfreund, D., Xu, K., & Ahmed, F. (2024). LInK: Learning Joint Representations of Design and Performance Spaces through Contrastive Learning for Mechanism Synthesis. Https://arxiv.org/abs/2405.20592
11. Abueidda, D. W., Pantidis, P., & Mobasher, M. E. (2024). DeepOKAN: Deep Operator Network Based on Kolmogorov Arnold Networks for Mechanics Problems. Https://arxiv.org/abs/2405.19143
12. Zhang, Y. (2024). Non-Driving-Related Tasks Influencing Drivers' Takeover Time: A Meta-Analysis. Https://arxiv.org/abs/2405.18667
13. Cheng, Z., Hao, Z., Wang, X., Huang, J., Wu, Y., Liu, X., Zhao, Y., Liu, S., & Su, H. (2024). Reference Neural Operators: Learning the Smooth Dependence of Solutions of PDEs on Geometric Deformations. Https://arxiv.org/abs/2405.17509
14. Chu, M., & Qian, W. (2024). Physics Constrained Deep Learning For Turbulence Model Uncertainty Quantification. Https://arxiv.org/abs/2405.16554
15. Patel, Y., Rayan, S., & Tewari, A. (2024). Conformal Robust Control of Linear Systems. Https://arxiv.org/abs/2405.16250
16. Zhu, Q., Wang, D., Ma, S., Wang, A. Y., Chen, Z., Khurana, U., & Ma, X. (2024). Towards Feature Engineering with Human and AI's Knowledge: Understanding Data Science Practitioners' Perceptions in Human&AI-Assisted Feature Engineering Design. https://doi.org/10.1145/3643834.3661517
17. Serani, A., & Diez, M. (2024). A Survey on Design-space Dimensionality Reduction Methods for Shape Optimization. Https://arxiv.org/abs/2405.13944
18. Mayeur, J., Cheng, J., Lee, Y., Knipe, K., & Kadau, K. (2024). Self-supervised feature distillation and design of experiments for efficient training of micromechanical deep learning surrogates. Https://arxiv.org/abs/2405.10135
19. Chu, M., & Qian, W. (2024). A Deep Learning Approach For Epistemic Uncertainty Quantification Of Turbulent Flow Simulations. Https://arxiv.org/abs/2405.08148
20. Okpala, E., Vishwamitra, N., Guo, K., Liao, S., Cheng, L., Hu, H., Wu, Y., Yuan, X., Wade, J., & Khorsandroo, S. (2024). AI-Cybersecurity Education Through Designing AI-based Cyberharassment Detection Lab. Https://arxiv.org/abs/2405.08125

21. Hamade, K., Sen, S., Kleinberg, J., & Anderson, A. (2024). Designing Skill-Compatible AI: Methodologies and Frameworks in Chess. [*Https://arxiv.org/abs/*2405.05066](https://arxiv.org/abs/2405.05066)
22. Arisaka, S., & Li, Q. (2024). Accelerating Legacy Numerical Solvers by Non-intrusive Gradient-based Meta-solving. [*Https://arxiv.org/abs/*2405.02952](https://arxiv.org/abs/2405.02952)
23. Ege, D. N., Øvrebø, H. H., Stubberud, V., Berg, M. F., Elverum, C., Steinert, M., & Vestad, H. (2024). ChatGPT as an inventor: Eliciting the strengths and weaknesses of current large language models against humans in engineering design. [*Https://arxiv.org/abs/*2404.18479](https://arxiv.org/abs/2404.18479)
24. Lou, T., Guan, G., Yue, Z., Wang, Y., Qi, R., & Tong, S. (2024). A competitive game optimization algorithm for Unmanned Aerial Vehicle path planning. [*Https://arxiv.org/abs/*2404.09567](https://arxiv.org/abs/2404.09567)
25. Doris, A. C., Grandi, D., Tomich, R., Alam, M. F., Cheong, H., & Ahmed, F. (2024). DesignQA: A Multimodal Benchmark for Evaluating Large Language Models' Understanding of Engineering Documentation. [Https://arxiv.org/abs/2404.07917](https://arxiv.org/abs/2404.07917)
26. Picard, C., & Ahmed, F. (2024). Fast and Accurate Bayesian Optimization with Pre-trained Transformers for Constrained Engineering Problems. [*Https://arxiv.org/abs/*2404.04495](https://arxiv.org/abs/2404.04495)
27. Kevian, D., Syed, U., Guo, X., Havens, A., Dullerud, G., Seiler, P., Qin, L., & Hu, B. (2024). Capabilities of Large Language Models in Control Engineering: A Benchmark Study on GPT-4, Claude 3 Opus, and Gemini 1.0 Ultra. [*Https://arxiv.org/abs/*2404.03647](https://arxiv.org/abs/2404.03647)
28. He, J., Koric, S., Abueidda, D., Najafi, A., & Jasiuk, I. (2024). Geom-DeepONet: A Point-cloud-based Deep Operator Network for Field Predictions on 3D Parameterized Geometries. https://doi.org/10.1016/j.cma.2024.117130
29. Wang, J., Petra, C. G., & Peterson, J. L. (2024). Constrained Bayesian optimization with merit functions. [*Https://arxiv.org/abs/*2403.13140](https://arxiv.org/abs/2403.13140)
30. Jiang, S., & Luo, J. (2024). AutoTRIZ: Artificial Ideation with TRIZ and Large Language Models. [*Https://arxiv.org/abs/*2403.13002](https://arxiv.org/abs/2403.13002)
31. Liu, P., Hong, Y., Escobar, L. A., & Meeker, W. Q. (2024). On Equivalence of Likelihood-Based Confidence Bands for Fatigue-Life and Fatigue-Strength Distributions. [*Https://arxiv.org/abs/*2403.12757](https://arxiv.org/abs/2403.12757)

32. Zhao, Z., Liu, C., Li, Y., Chen, Z., & Liu, X. (2024). Diffeomorphism Neural Operator for various domains and parameters of partial differential equations. *Https://arxiv.org/abs/*2402.12475
33. Li, K., & Li, F. (2024). Multi-Fidelity Methods for Optimization: A Survey. *Https://arxiv.org/abs/*2402.09638
34. Usama, M., Masood, Z., Khan, S., Kostas, K., & Kaklis, P. (2024). Generative VS non-Generative Models in Engineering Shape Optimization. *Https://arxiv.org/abs/*2402.08540
35. Cabral, R., Kalinowski, M., Baldassarre, M. T., Villamizar, H., Escovedo, T., & Lopes, H. (2024). Investigating the Impact of SOLID Design Principles on Machine Learning Code Understanding. *Https://arxiv.org/abs/*2402.05337
36. Nobari, A. H., Giannone, G., Regenwetter, L., & Ahmed, F. (2024). NITO: Neural Implicit Fields for Resolution-free Topology Optimization. *Https://arxiv.org/abs/*2402.05073
37. Kim, H. (2024). Enhancing Gaussian Process Surrogates for Optimization and Posterior Approximation via Random Exploration. *Https://arxiv.org/abs/*2401.17037
38. Lou, Y., Sun, S., & Nocedal, J. (2024). Noise-Tolerant Optimization Methods for the Solution of a Robust Design Problem. *Https://arxiv.org/abs/*2401.15007
39. Porter, J. H., & Brake, M. R. (2024). Tracking Superharmonic Resonances for Nonlinear Vibration of Conservative and Hysteretic Single Degree of Freedom Systems. https://doi.org/10.1016/j.ymssp.2024.111410
40. Naeini, S. S., & Snaiki, R. (2024). A Physics-informed machine learning model for time-dependent wave runup prediction. *Https://arxiv.org/abs/*2401.08684
41. Peng, Y., Niloy, I., Kam, M., Celli, P., & Plucinsky, P. (2024). Programming bistability in geometrically perturbed mechanical metamaterials. *Https://arxiv.org/abs/*2401.07881
42. Picard, C., & Ahmed, F. (2024). Fast and Accurate Zero-Training Classification for Tabular Engineering Data. *Https://arxiv.org/abs/*2401.06948
43. Study Group, T. C. (2023). CEPC Technical Design Report -- Accelerator (v2). *Https://arxiv.org/abs/*2312.14363

44. Yang, L., Zhang, J., Li, H., Ren, L., Yang, C., Wang, J., & Shi, D. (2023). A Comprehensive End-to-End Computer Vision Framework for Restoration and Recognition of Low-Quality Engineering Drawings. Https://arxiv.org/abs/2312.13620
45. Emuna, H., Borenstein, N., Qian, X., Kang, H., Chan, J., Kittur, A., & Shahaf, D. (2023). Imitation of Life: A Search Engine for Biologically Inspired Design. *Https://arxiv.org/abs/*2312.12681
46. Namura, N. (2023). Single and Multi-Objective Optimization Benchmark Problems Focusing on Human-Powered Aircraft Design. *Https://arxiv.org/abs/*2312.08953
47. Zhang, J., Chen, X., Wang, Y., Wang, C., Liu, Y., Li, X., Yang, M., & Tao, D. (2023). Exploring Plain ViT Reconstruction for Multi-class Unsupervised Anomaly Detection. *Https://arxiv.org/abs/*2312.07495
48. Siddharth, L., & Luo, J. (2023). Linguistic and Structural Basis of Engineering Design Knowledge. *Https://arxiv.org/abs/*2312.06355
49. Wang, Z., Xiong, J., Liu, X., Liu, L., & Lai, X. (2023). An Updated Lagrangian Particle Hydrodynamics (ULPH)-NOSBPD Coupling Approach for Modeling Fluid-Structure Interaction Problems. *Https://arxiv.org/abs/*2312.05016
50. Hawthorne, F., Cleuren, B., & Fiore, C. E. (2023). Thermodynamics of a minimal collective heat engine: Comparison between engine designs. *Https://arxiv.org/abs/*2311.18629
51. Jessica, L. S., Arafat, N. A., Lim, W. X., Chan, W. L., & Kong, A. W. (2023). Finite Volume Features, Global Geometry Representations, and Residual Training for Deep Learning-based CFD Simulation. *Https://arxiv.org/abs/*2311.14464
52. Do, B., & Zhang, R. (2023). Multi-fidelity Bayesian Optimization in Engineering Design. *Https://arxiv.org/abs/*2311.13050
53. Picard, C., Edwards, K. M., Doris, A. C., Man, B., Giannone, G., Alam, M. F., & Ahmed, F. (2023). From Concept to Manufacturing: Evaluating Vision-Language Models for Engineering Design. Https://arxiv.org/abs/2311.12668
54. Wang, Z., Mantey, S. K., & Zhang, X. (2023). A numerical tool for efficient analysis and optimization of offshore wind turbine jacket substructure considering realistic boundary and loading conditions. *Https://arxiv.org/abs/*2310.19388

55. Lie, H., Studer, K., Zhao, Z., Thomson, B., Turakhia, D. G., & Liu, J. (2023). Training for Open-Ended Drilling through a Virtual Reality Simulation. [Https://arxiv.org/abs/2310.17417](https://arxiv.org/abs/2310.17417)
56. Azizzadenesheli, K., Kovachki, N., Li, Z., Kossaifi, J., & Anandkumar, A. (2023). Neural Operators for Accelerating Scientific Simulations and Design. [Https://arxiv.org/abs/2309.15325](https://arxiv.org/abs/2309.15325)
57. Kang, R., Mu, T., Liatsis, P., & Kyritsis, D. C. (2023). Physics-Driven ML-Based Modelling for Correcting Inverse Estimation. [Https://arxiv.org/abs/2309.13985](https://arxiv.org/abs/2309.13985)
58. Han, T., Li, J., Guo, Z., & Jin, Y. (2023). A Region-Shrinking-Based Acceleration for Classification-Based Derivative-Free Optimization. [Https://arxiv.org/abs/2309.11036](https://arxiv.org/abs/2309.11036)
59. Zhao, Y., Liu, Y., & Xu, Z. (2023). Predicting Fatigue Crack Growth via Path Slicing and Re-Weighting. https://doi.org/10.1016/j.taml.2023.100477
60. Sáez, A., & Lecampion, B. (2023). Fluid-driven slow slip and earthquake nucleation on a slip-weakening circular fault. [Https://arxiv.org/abs/2309.04567](https://arxiv.org/abs/2309.04567)
61. Li, C., Chen, H., Yan, M., Shen, W., Xu, H., Wu, Z., Zhang, Z., Zhou, W., Chen, Y., Cheng, C., Shi, H., Zhang, J., Huang, F., & Zhou, J. (2023). ModelScope-Agent: Building Your Customizable Agent System with Open-source Large Language Models. [Https://arxiv.org/abs/2309.00986](https://arxiv.org/abs/2309.00986)
62. Goyal, P., Duff, I. P., & Benner, P. (2023). Guaranteed Stable Quadratic Models and their applications in SINDy and Operator Inference. [Https://arxiv.org/abs/2308.13819](https://arxiv.org/abs/2308.13819)
63. Jwa, M., Kim, J., Shin, S., Jin, A., Shin, D., & Kang, N. (2023). Performance Comparison of Design Optimization and Deep Learning-based Inverse Design. [Https://arxiv.org/abs/2308.13000](https://arxiv.org/abs/2308.13000)
64. Girija, A. P. (2023). Planetary Entry Probe Dataset: Analysis and Rules of Thumb for Future Missions. [Https://arxiv.org/abs/2308.07005](https://arxiv.org/abs/2308.07005)
65. Constantinides, M., Bogucka, E., Quercia, D., Kallio, S., & Tahaei, M. (2023). RAI Guidelines: Method for Generating Responsible AI Guidelines Grounded in Regulations and Usable by (Non-)Technical Roles. [Https://arxiv.org/abs/2307.15158](https://arxiv.org/abs/2307.15158)

66. Fan, J., Vuaille, L., Wang, H., & Bäck, T. (2023). Adversarial Latent Autoencoder with Self-Attention for Structural Image Synthesis. https://doi.org/10.1109/CAI59869.2024.00030
67. Siddharth, L., & Luo, J. (2023). Retrieval Augmented Generation using Engineering Design Knowledge. *Https://arxiv.org/abs/*2307.06985
68. Rios, T., Menzel, S., & Sendhoff, B. (2023). Large Language and Text-to-3D Models for Engineering Design Optimization. *Https://arxiv.org/abs/*2307.01230
69. Giannone, G., Regenwetter, L., Srivastava, A., Gutfreund, D., & Ahmed, F. (2023). Learning from Invalid Data: On Constraint Satisfaction in Generative Models. *Https://arxiv.org/abs/*2306.15166
70. Szaksz, B., Stepan, G., & Habib, G. (2023). Dynamical integrity estimation in time delayed systems: A rapid iterative algorithm. *Https://arxiv.org/abs/*2306.14821
71. Saxena, N., Jain, M., Liu, C., & Bengio, Y. (2023). Multi-Fidelity Active Learning with GFlowNets. *Https://arxiv.org/abs/*2306.11715
72. Arechiga, N., Permenter, F., Song, B., & Yuan, C. (2023). Drag-guided diffusion models for vehicle image generation. *Https://arxiv.org/abs/*2306.09935
73. Göpfert, J., Weinand, J. M., Kuckertz, P., & Stolten, D. (2023). Opportunities for Large Language Models and Discourse in Engineering Design. *Https://arxiv.org/abs/*2306.09169
74. Song, B., Yuan, C., Permenter, F., Arechiga, N., & Ahmed, F. (2023). Surrogate Modeling of Car Drag Coefficient with Depth and Normal Renderings. *Https://arxiv.org/abs/*2306.06110
75. Cobb, A. D., Roy, A., Elenius, D., Heim, F. M., Swenson, B., Whittington, S., Walker, J. D., Bapty, T., Hite, J., Ramani, K., McComb, C., & Jha, S. (2023). AircraftVerse: A Large-Scale Multimodal Dataset of Aerial Vehicle Designs. *Https://arxiv.org/abs/*2306.05562
76. Ansari, N., Seidel, H., & Babaei, V. (2023). Large-Batch, Iteration-Efficient Neural Bayesian Design Optimization. *Https://arxiv.org/abs/*2306.01095
77. Picard, C., Schiffmann, J., & Ahmed, F. (2023). DATED: Guidelines for Creating Synthetic Datasets for Engineering Design Applications. *Https://arxiv.org/abs/*2305.09018

78. Wang, R., Shapiro, V., & Behandish, M. (2023). Model Consistency for Mechanical Design: Bridging Lumped and Distributed Parameter Models with A Priori Guarantees. https://doi.org/10.1115/1.4064810
79. Seurin, P., & Shirvan, K. (2023). Assessment of Reinforcement Learning Algorithms for Nuclear Power Plant Fuel Optimization. *Https://arxiv.org/abs/*2305.05812
80. Nemani, V., Biggio, L., Huan, X., Hu, Z., Fink, O., Tran, A., Wang, Y., Zhang, X., & Hu, C. (2023). Uncertainty Quantification in Machine Learning for Engineering Design and Health Prognostics: A Tutorial. https://doi.org/10.1016/j.ymssp.2023.110796
81. Bollapragada, R., Karamanli, C., Keith, B., Lazarov, B., Petrides, S., & Wang, J. (2023). An Adaptive Sampling Augmented Lagrangian Method for Stochastic Optimization with Deterministic Constraints. *Https://arxiv.org/abs/*2305.01018
82. Alpat, A. B., Coban, A., Kaya, H., & Bartolini, G. (2023). MRADSIM-Converter: A new software for STEP to GDML conversion. https://doi.org/10.1016/j.cpc.2023.108688
83. Sharma, G., Mun, S., Lee, N., Peterson, L., Tellkamp, D., & Nellippallil, A. B. (2023). Machine Learning-Based Multi-Objective Design Exploration Of Flexible Disc Elements. *Https://arxiv.org/abs/*2304.07245
84. Chen, L., Yao, X., Liu, K., Tan, C., & Moon, S. K. (2023). Multisensor fusion-based digital twin in additive manufacturing for in-situ quality monitoring and defect correction. *Https://arxiv.org/abs/*2304.05685
85. Mallon, C. N., Thornton, A. W., Hill, M. R., & Badia, S. (2023). Neural Level Set Topology Optimization Using Unfitted Finite Elements. *Https://arxiv.org/abs/*2303.13672
86. Ahmadianshalchi, A., Belakaria, S., & Doppa, J. R. (2023). Preference-Aware Constrained Multi-Objective Bayesian Optimization. *Https://arxiv.org/abs/*2303.13034
87. Kochiyama, S., & Kesari, H. (2023). Functional significance of lamellar architecture in marine sponge fibers: Conditions for when splitting a cylindrical tube into an assembly of tubes will decrease its bending stiffness. *Https://arxiv.org/abs/*2303.12189-

88. Giannone, G., & Ahmed, F. (2023). Diffusing the Optimal Topology: A Generative Optimization Approach. *Https://arxiv.org/abs/*2303.09760
89. Kumar, S., & Atre, S. V. (2023). Design and optimization of brake disc using Multi-Objective genetic algorithm. https://doi.org/10.46720/F2020-ADM-071
90. Han, R., Kramer, B., Lee, D., Narayan, A., & Xu, Y. (2023). An approximate control variates approach to multifidelity distribution estimation. *Https://arxiv.org/abs/*2303.06422
91. Gmeiner, F., Yang, H., Yao, L., Holstein, K., & Martelaro, N. (2023). Exploring Challenges and Opportunities to Support Designers in Learning to Co-create with AI-based Manufacturing Design Tools. https://doi.org/10.1145/3544548.3580999
92. Vardhan, H., Volgyesi, P., & Sztipanovits, J. (2023). Fusion of ML with numerical simulation for optimized propeller design. *Https://arxiv.org/abs/*2302.14740
93. Toklu, N. E., Atkinson, T., Micka, V., Liskowski, P., & Srivastava, R. K. (2023). EvoTorch: Scalable Evolutionary Computation in Python. *Https://arxiv.org/abs/*2302.12600
94. Song, B., Zhou, R., & Ahmed, F. (2023). Multi-modal Machine Learning in Engineering Design: A Review and Future Directions. *Https://arxiv.org/abs/*2302.10909
95. Regenwetter, L., Srivastava, A., Gutfreund, D., & Ahmed, F. (2023). Beyond Statistical Similarity: Rethinking Metrics for Deep Generative Models in Engineering Design. *Https://arxiv.org/abs/*2302.02913
96. Yang, M., Jiang, P., Zang, T., & Liu, Y. (2023). Data-driven intelligent computational design for products: Method, techniques, and applications. *Https://arxiv.org/abs/*2301.12382
97. Goyal, P., Duff, I. P., & Benner, P. (2023). Inference of Continuous Linear Systems from Data with Guaranteed Stability. *Https://arxiv.org/abs/*2301.10060
98. Segarra, S., Paoletti, J., Zajac, S., Beier, M. E., Sabharwal, A., Wettergreen, M., & Salas, E. (2023). Conversational Turn-taking as a Stochastic Process on Networks. *Https://arxiv.org/abs/*2301.04030

99. Okada, A., Yoshida, H., Kidono, K., Matsumori, T., Takeno, T., & Kadowaki, T. (2023). Design Optimization of Noise Filter using Quantum Annealer. https://doi.org/10.1109/ACCESS.2023.3271969
100. Zhang, H., Zhou, Y., & Li, F. (2023). An Automatic Method for Generating Symbolic Expressions of Zernike Circular Polynomials. https://doi.org/10.1109/ACCESS.2023.3283028
101. Huang, J. (2023). Digital Engineering Transformation with Trustworthy AI towards Industry 4.0: Emerging Paradigm Shifts. *Https://arxiv.org/abs/*2301.00951
102. Arora, R., & Shrivastava, A. (2022). Spatio-Temporal Super-Resolution of Dynamical Systems using Physics-Informed Deep-Learning. *Https://arxiv.org/abs/*2212.04457
103. Bond, R. B., Ren, P., Hajjar, J. F., & Sun, H. (2022). An Unsupervised Machine Learning Approach for Ground-Motion Spectra Clustering and Selection. https://doi.org/10.1002/eqe.4062
104. Fan, J., Vuaille, L., Wang, H., & Bäck, T. (2023). Adversarial Latent Autoencoder with Self-Attention for Structural Image Synthesis. https://doi.org/10.1109/CAI59869.2024.00030
105. Zhang, W., Joseph, J., Yin, Y., Xie, L., Furuhata, T., Yamakawa, S., Shimada, K., & Kara, L. B. (2022). Component Segmentation of Engineering Drawings Using Graph Convolutional Networks. https://doi.org/10.1016/j.compind.2023.103885
106. Raina, A., Cagan, J., & McComb, C. (2022). Learning to design without prior data: Discovering generalizable design strategies using deep learning and tree search. https://doi.org/10.1115/1.4056221
107. Vardhan, H., Timalsina, U., Volgyesi, P., & Sztipanovits, J. (2022). Data efficient surrogate modeling for engineering design: Ensemble-free batch mode deep active learning for regression. *Https://arxiv.org/abs/*2211.10360
108. Hao, Z., Liu, S., Zhang, Y., Ying, C., Feng, Y., Su, H., & Zhu, J. (2022). Physics-Informed Machine Learning: A Survey on Problems, Methods and Applications. *Https://arxiv.org/abs/*2211.08064
109. Ardalani, N., Pal, S., & Gupta, P. (2022). DeepFlow: A Cross-Stack Pathfinding Framework for Distributed AI Systems. *Https://arxiv.org/abs/*2211.03309

110. Kumar, D., Ahmed, F., Usman, S., Alajo, A., & Alam, S. (2022). Recent Advances in Uncertainty Quantification Methods for Engineering Problems. *Https://arxiv.org/abs/2211.03012*
111. Siddharth, L., Li, G., & Luo, J. (2022). Enhancing Patent Retrieval using Text and Knowledge Graph Embeddings: A Technical Note. *Https://arxiv.org/abs/2211.01976*
112. Casey, S. (2022). Quantifying Complexity: An Object-Relations Approach to Complex Systems. *Https://arxiv.org/abs/2210.12347*
113. Wei, H., Lyu, D., Hu, W., & Wu, C. T. (2022). RVE Analysis in LS-DYNA for High-fidelity Multiscale Material Modeling. *Https://arxiv.org/abs/2210.11761*
114. Filanova, Y., Duff, I. P., Goyal, P., & Benner, P. (2022). An Operator Inference Oriented Approach for Mechanical Systems. https://doi.org/10.1016/j.ymssp.2023.110620
115. Sadeghi, J., Mueller, R., & Redford, J. (2022). An Active Learning Reliability Method for Systems with Partially Defined Performance Functions. *Https://arxiv.org/abs/2210.02168*
116. Mattioli, J., Delaborde, A., Khalfaoui, S., Lecue, F., Sohier, H., & Jurie, F. (2022). Empowering the trustworthiness of ML-based critical systems through engineering activities. *Https://arxiv.org/abs/2209.15438*
117. Song, Z., Yao, H., Tian, D., & Zhan, G. (2022). Cyclegan Network for Sheet Metal Welding Drawing Translation. *Https://arxiv.org/abs/2209.14106*
118. Song, Z., & Yao, H. (2022). Segmentation method of U-net sheet metal engineering drawing based on CBAM attention mechanism. *Https://arxiv.org/abs/2209.14102*
119. Ghosh, A., Deb, K., Goodman, E., & Averill, R. (2022). An Interactive Knowledge-based Multi-objective Evolutionary Algorithm Framework for Practical Optimization Problems. *Https://arxiv.org/abs/2209.08604*
120. Vani, N., Escudier, S., & Sauret, A. (2022). Influence of the solid fraction on the clogging by bridging of suspensions in constricted channels. *Https://arxiv.org/abs/2208.11643*
121. Mazé, F., & Ahmed, F. (2022). Diffusion Models Beat GANs on Topology Optimization. *Https://arxiv.org/abs/2208.09591*

122. Arora, R. (2022). PhySRNet: Physics informed super-resolution network for application in computational solid mechanics. *Https://arxiv.org/abs/*2206.15457
123. Vardhan, H., & Sztipanovits, J. (2022). Deep Active Learning for Regression Using ϵ-weighted Hybrid Query Strategy. *Https://arxiv.org/abs/*2206.13298
124. Chen, Z., Li, Q., & Zhang, Z. (2022). Self-Healing Robust Neural Networks via Closed-Loop Control. *Https://arxiv.org/abs/*2206.12963
125. Thomas, S. S., Palandri, J., Chakravarty, P., & Blaschko, M. B. (2022). Designing MacPherson Suspension Architectures using Bayesian Optimization. *Https://arxiv.org/abs/*2206.09022
126. Regenwetter, L., & Ahmed, F. (2022). Towards Goal, Feasibility, and Diversity-Oriented Deep Generative Models in Design. *Https://arxiv.org/abs/*2206.07170
127. Khoshoo, B., Blank, J., Pham, T. Q., Deb, K., & Foster, S. N. (2022). Optimal Design of Electric Machine with Efficient Handling of Constraints and Surrogate Assistance. *Https://arxiv.org/abs/*2206.01695
128. Sarkar, S., Pandey, P., & Kar, S. (2022). Automatic Detection and Classification of Symbols in Engineering Drawings. *Https://arxiv.org/abs/*2204.13277
129. Andersen, B., Delipei, G., Kropaczek, D., & Hou, J. (2022). MOF: A Modular Framework for Rapid Application of Optimization Methodologies to General Engineering Design Problems. *Https://arxiv.org/abs/*2204.00141
130. Li, Y., & Dankowicz, H. (2022). Model-free Continuation of Periodic Orbits in Certain Nonlinear Systems Using Continuous-Time Adaptive Control. https://doi.org/10.1007/s11071-022-08059-1
131. Regenwetter, L., Weaver, C., & Ahmed, F. (2022). FRAMED: An AutoML Approach for Structural Performance Prediction of Bicycle Frames. *Https://arxiv.org/abs/*2201.10459
132. Karcher, C., & Haimes, R. (2022). A Method of Sequential Log-Convex Programming for Engineering Design. *Https://arxiv.org/abs/*2201.08436
133. Astudillo, R., & Frazier, P. I. (2021). Bayesian Optimization of Function Networks. *Https://arxiv.org/abs/*2112.15311

134. Knaster, J., Ibarra, A., Abal, J., Sena, A. A., Arbeiter, F., Arranz, F., Arroyo, J. M., Bargallo, E., Beauvais, P. Y., Bernardi, D., Casal, N., Carmona, J. M., Chauvin, N., Comunian, M., Delferriere, O., Delgado, A., Arocas, P. D., Fischer, U., Frisoni, M., . . . Yokomine, T. (2021). The accomplishment of the Engineering Design Activities of IFMIF/EVEDA: The European Japanese project towards a Li(d,xn) fusion relevant neutron source. https://doi.org/10.1088/0029-5515/55/8/086003
135. Lambert, J., Liu, Z., Sener, O., Hays, J., & Koltun, V. (2021). MSeg: A Composite Dataset for Multi-domain Semantic Segmentation. *Https://arxiv.org/abs/*2112.13762
136. Harish, A. B., & Prasad, A. R. (2021). Photo2CAD: Automated 3D solid reconstruction from 2D drawings using OpenCV. *Https://arxiv.org/abs/*2101.04248
137. Chandrasekhar, A., Sridhara, S., & Suresh, K. (2021). Integrating Material Selection with Design Optimization via Neural Networks. *Https://arxiv.org/abs/*2112.12566
138. https://azure.microsoft.com/en-in/products/ai-services/ai-document-intelligence
139. https://learn.microsoft.com/en-us/azure/ai-services/openai/overview
140. https://www.llamaindex.ai/
141. https://www.langchain.com/

Appendix A: Physics-Informed Geometry-Aware Neural Operator (PI-GANO)

This appendix provides a detailed overview of the Physics-Informed Geometry-Aware Neural Operator (PI-GANO), including its architecture, key components, and training methodology.

A.1 Overview

PI-GANO is designed to solve parametric Partial Differential Equations (PDEs) with varying domain geometries and PDE parameters. It combines the strengths of Physics-Informed Deep Compositional Operator Network (PI-DCON) and Physics-Informed PointNet (PI-PointNet) to create a neural operator capable of generalizing across both PDE parameters and domain geometries without requiring training data.

A.2 Problem Setting

The goal is to develop an efficient machine learning-based solver for parametric PDEs formulated as:

$$N_x[u(x), k(x)] = 0, x \in \Omega,$$

$$B_x[u(x)] = g(x), x \in \partial\Omega,$$

where:

- Ω is a physical domain in R^d
- x is a d-dimensional vector of spatial coordinates
- N_x is a general differential operator
- B_x is a boundary condition operator acting on the domain boundary $\partial\Omega$
- $k(x)$ refers to the parameters of the PDE
- $g(x)$ denotes the boundary conditions
- $u(x)$ is the solution of the PDE

The aim is to approximate the operator M defined by:

$$M: \{k_i(x), g_i(x), \Omega_i\} \rightarrow u_i(x), x \in \Omega_i, \Omega_i \in \Omega$$

A.3 Model Architecture

PI-GANO's architecture consists of several key components:

A.3.1 Geometry Encoder

The geometry encoder captures the domain geometry features:

1. Represent domain geometry Ω_i using a set of collocation points $\{x_j^i\}_{j=1}^{s_i}$.
2. Compute high-dimensional features using an MLP mapping $U_G(\cdot, \theta_G)$.
3. Apply average pooling to extract global geometry features:

$$G_i = \mathrm{AvgPool}(\{h_j\}_{j=1}^{s_i}),$$

where $\{h_j\}_{j=1}^{s_i} = \{U_G(x_j, \theta_G)\}_{j=1}^{s_i}$

A.3.2 Parameter Encoder

The parameter encoder processes the PDE parameters:

1. Represent PDE parameters in discrete form by evaluating functions at sampled coordinates.

2. Apply a multi-layer perceptron $U_b(\cdot, \theta_b)$ with a Max-pooling layer:

$$b = \mathrm{Maxpool}(\{U_b(g(x'_j), \theta_b)\}_{j=1}^m)$$

A.3.3 Local Coordinate Feature Computation

Compute local coordinate features for each collocation point:

$$h_i = \sigma(W_t x_i + B_t)$$

A.3.4 Feature Combination

Concatenate the geometry global feature G_i with each local coordinate feature h_i:

$$H_i = [h_i \| G_i]$$

A.3.5 Operator Layers

Apply a series of operator layers to produce the PDE solution:

$$u_i(x_j^i) \approx M_\theta(x_j^i, g_i(x), \Omega_i) = \mathrm{sum}\{b \odot \ldots(b \odot (W_O^2 \sigma(b \odot (W_O^1 [\sigma(W_t x_i + B_t) \| G_i] + B_O^1)) + B_O^2))\}$$

A.4 Training Methodology

PI-GANO is trained using a physics-informed approach, which incorporates the governing PDEs directly into the loss function:

$$L(\theta) = L_{PDE}(\theta) + \alpha L_{BC}(\theta)$$

where:

- $L_{PDE}(\theta)$ is the PDE residual loss
- $L_{BC}(\theta)$ is the boundary condition loss
- α is a weighting hyperparameter

The specific formulations of $L_{PDE}(\theta)$ and $L_{BC}(\theta)$ depend on the problem being solved. For example, in the Darcy flow problem:

$$L_{PDE}(\theta) = (1/N) \sum_{i=1}^N (N_x[U_\theta(x, k_i(x'), g_i(x'')), k_i(x')])^2, \; x \in \Omega$$

$$L_{BC}(\theta) = (1/N) \sum_{i=1}^N (B_x[U_\theta(x, k_i(x'), g_i(x''))] - g_i(x''))^2, \; x \in \partial\Omega$$

A.5 Key Advantages

1. Generalization: PI-GANO can handle both varying PDE parameters and domain geometries.
2. Data-free: No need for training data obtained from high-fidelity simulations.

3. Efficiency: Faster training compared to data-driven methods, especially for fine meshes.
4. Accuracy: Improved performance compared to existing physics-informed neural operators.

A.6 Limitations and Future Work

1. Current focus on steady-state solutions; need to adapt for time-dependent PDEs.
2. Potential instability in physics-informed training, requiring more training epochs.
3. Opportunity to incorporate more sophisticated architectures, such as attention mechanisms.

By addressing these limitations and building upon the strengths of PI-GANO, future research can further enhance the capabilities of neural operators in solving complex engineering and scientific problems involving PDEs on varying geometries.

Appendix B: Physics-Informed Geometric Operators (GOs)

This appendix provides a detailed overview of Physics-Informed Geometric Operators (GOs), including their mathematical formulation, computation methods, and their relation to physics in engineering design problems.

B.1 Overview

Physics-Informed Geometric Operators (GOs) are proposed to enrich geometric data provided to surrogate models, dimension reduction models, and generative models in engineering design. GOs leverage the shape's differential and integral properties to capture varying characteristics of the shape related to its volume distribution, complexity of the bounding surface, and overall surface smoothness.

B.2 Components of Geometric Operators

The GOs are defined as:

$$GO(G) = (P(G), M(G), K(G), F(G))$$

Where:

- G is a geometric object representing a baseline design
- P(G) is a vector function providing a suitable representation of G
- M(G) represents geometric moments

- $K(G)$ represents curvature integrals
- $F(G)$ represents Fourier descriptors

B.3 Geometric Moments

B.3.1 Definition

An s-order geometric moment $M_s = M_{p,q,r}$ of a geometric object G is defined as:

$$M_{p,q,r}(G) = \iiint_G x^p y^q z^r \, dx \, dy \, dz, \text{ with } p, q, r \in \{0, 1, 2, \ldots\}$$

B.3.2 Computation

Geometric moments are typically computed using Gauss's divergence theorem, which converts volume integrals to surface integrals.

B.3.3 Relation to Physics

Geometric moments are strongly connected to physical quantities that determine design performance characteristics. For example, in ship hull design, the moments of the Sectional Area Curve (SAC) are directly related to the ship's hydrodynamic properties.

B.4 Curvature

B.4.1 Gaussian Curvature

For a surface $P(u,v)$, the Gaussian curvature $K(u,v)$ is defined as:

$$K(u,v) = \kappa_1 \kappa_2 = (LN - M^2) / (EG - F^2)$$

Where L, M, N, E, F, G are coefficients of the first and second fundamental forms of the surface.

B.4.2 Total Curvature

Total curvature is the integral of Gaussian curvature over the entire surface. It's connected to the Euler characteristic through the Gauss-Bonnet Theorem.

B.4.3 Relation to Physics

Curvature is linked to fluid flow behavior around objects. For example, in aerofoil design, curvature distribution affects aerodynamic performance and heat transfer characteristics.

B.5 Three-dimensional Fourier Descriptor

B.5.1 Definition

Fourier Descriptors (FDs) are particularly sensitive to the geometry of the shape's boundary. They provide an enhanced description of the object's boundary surface.

B.5.2 Computation

The computation of 3D FDs is based on the work of Park and Lee, which involves:

1. Voxelization of the 3D object
2. Computation of the 3D Fourier transform
3. Extraction of Fourier coefficients

B.5.3 Relation to Physics

FDs are particularly useful in capturing the boundary characteristics of functional surfaces like wings, blades, and ship hulls, which are critical for their performance.

B.6 Application in Machine Learning Models

GOs can be used to augment training data for various machine learning models:

1. Surrogate Models: GOs improve the accuracy of performance prediction models.
2. Dimension Reduction Models: GOs enable the extraction of more efficient and physically meaningful latent spaces.
3. Generative Models: GOs help in generating more diverse and valid designs and can be used as a substitute for computationally expensive physics-based metrics.

B.7 Benefits of GOs

1. Capture both global and local shape features
2. Embed physical information in geometric representations
3. Improve model accuracy and generalization capability
4. Reduce the dimensionality of design spaces while maintaining diversity and validity
5. Provide a computationally efficient substitute for physics-based metrics in some applications

By incorporating these physics-informed geometric operators, machine learning models in engineering design can achieve better performance, improved generalization, and more physically meaningful results.

Appendix C: CAD-Prompted Generative Models

This appendix provides a detailed overview of CAD-Prompted Generative Models, including their methodology, implementation, and implications for engineering design.

C.1 Overview

CAD-Prompted Generative Models are an innovative approach to improving the feasibility of designs generated by text-to-image (T2I) models in engineering design contexts. This method addresses the limitation of T2I models in producing feasible design concepts by incorporating Computer-Aided Design (CAD) images as prompts alongside text inputs.

C.2 Methodology

C.2.1 Workflow

1. Text Prompt Input: The designer provides a text description of the desired design.
2. CAD Image Selection: Using OpenAI's CLIP model, the system identifies a CAD image from a dataset that best matches the text prompt semantically.
3. Image Generation: Both the text prompt and the selected CAD image are used as inputs for the T2I model (e.g., Stable Diffusion 2.1).
4. Output: The T2I model generates new design images, blending characteristics from both the text and CAD image prompts.

C.2.2 CAD Image Dataset

The method relies on a dataset of CAD images relevant to the product being designed. For example, in a bike design task, the dataset would consist of diverse bike designs in CAD image format.

C.2.3 Prompt Weighting

The influence of the CAD image on the generated design can be adjusted through a weighting system. Weights typically range from 0.35 to 1, where higher weights give more prominence to the CAD image characteristics in the final output.

C.3 Implementation

C.3.1 Tools and Models

- T2I Model: Stable Diffusion 2.1 (via Leonardo.AI platform)
- Image Selection: OpenAI's CLIP model
- CAD Image Dataset: e.g., BIKED dataset for bike designs

C.3.2 Generation Settings

Multiple settings can be used to explore different balances between text and CAD image prompts:

1. Stable Diffusion 2.1 (SD)
2. SD + Prompt Magic (SD+PM)
3. SD+PM + CAD Image Prompting (CIP) with varying weights (e.g., 0.35, 0.51, 0.67, 0.83, 1)

C.4 Evaluation Metrics

1. Perceived Feasibility: How manufacturable and realistic the generated design appears.
2. Novelty: The uniqueness and creativity of the generated design.
3. CLIP Similarity Score: Measure of similarity between designs generated under different settings.

C.5 Key Findings

1. Improved Feasibility: CAD image prompting significantly enhances the perceived feasibility of generated designs.
2. Feasibility-Novelty Tradeoff: Higher CAD image prompt weights tend to increase feasibility but decrease novelty.
3. Optimal Weighting: A CAD image prompt weight around 0.35 can improve feasibility without significantly compromising novelty.
4. Weight Threshold: Weights above 0.83 may lead to decreased feasibility, possibly due to the model's difficulty in balancing text and image inputs.

C.6 Implications for Engineering Design

1. Expanded Applicability: Enables T2I models to generate more realistic and potentially manufacturable designs.
2. Design Process Integration: Facilitates the use of T2I models beyond mere inspiration, potentially streamlining the concept-to-model process.

3. Flexible Design Exploration: Allows designers to adjust the balance between feasibility and novelty based on the design stage and requirements.
4. CAD Synergy: Potential for integration with image-to-CAD methods, enabling seamless transitions between concept generation and detailed design.

C.7 Limitations and Future Directions

1. Dataset Dependency: The quality of generated designs depends on the CAD image dataset used.
2. Prompt Balance: Further research on optimizing the balance between text and CAD image prompts is needed.
3. Validation: Additional case studies across various engineering domains must validate the method's effectiveness.
4. User Studies: Empirical investigations are needed to understand how engineers and designers can most effectively use this method in their workflows.

By addressing these limitations and building upon the strengths of CAD-Prompted Generative Models, future research can further enhance the capabilities of T2I models in engineering design, potentially revolutionizing the concept generation and design refinement processes.