

A DOMAIN ADAPTATION OF LARGE LANGUAGE MODELS FOR CLASSIFYING MECHANICAL ASSEMBLY COMPONENTS

Fatemeh Elhambakhsh¹, Daniele Grandi², Hyunwoong Ko^{3,*}

¹School of Computing and Augmented Intelligence, Arizona State University, Tempe, AZ

²Autodesk, San Francisco, CA

³School of Manufacturing Systems and Networks, Arizona State University, Mesa, AZ

ABSTRACT

The conceptual design phase represents a critical early stage in the product development process, where designers generate potential solutions that meet predefined design specifications based on functional requirements. Functional modeling, a foundational aspect of this phase, enables designers to reason about product functions before specific structural details are determined. A widely adopted approach to functional modeling is the Function-Behavior-Structure (FBS) framework, which supports the transformation of functional intent into behavioral and structural descriptions. However, the effectiveness of function-based design is often hindered by the lack of well-structured and comprehensive functional data. This scarcity can negatively impact early design decision-making and hinder the development of accurate behavioral models. Recent advances in Large Language Models (LLMs), such as those based on GPT architectures, offer a promising avenue to address this gap. LLMs have demonstrated significant capabilities in language understanding and natural language processing (NLP), making them suitable for automated classification tasks. This study proposes a novel LLM-based domain adaptation (DA) framework using fine-tuning for the automated classification of mechanical assembly parts' functions. By fine-tuning LLMs on domain-specific datasets, the traditionally manual and subjective process of function annotation can be improved in both accuracy and consistency. A case study demonstrates fine-tuning GPT-3.5 Turbo on data from the Oregon State Design Repository (OSDR), and evaluation on the A Big CAD (ABC) dataset shows that the domain-adapted LLM can generate high-quality functional data, enhancing the semantic representation of mechanical parts and supporting more effective design exploration in early-phase engineering.

Keywords: Large Language Models, Domain adaptation, Mechanical product design, Functional modeling

*Corresponding author: Hyunwoong.Ko@asu.edu

1. INTRODUCTION

The process of translating ideas to developing new products or enhancing an existing product to meet specific objectives that can address various aspects, such as usability, functionality, sustainability, and aesthetics, is known as product design [1]. Thanks to computer-aided product tools, such as computer-aided design (CAD) and computer-aided manufacturing (CAM), product design processes have become more digitalized [2]. The product design phase consists of three main stages: (1) Specification, (2) Conceptual design, and (3) Detailed design [3]. The conceptual design stage is arguably the critical stage, for it involves functional formulation and concept generation, evaluation, and improvement to generate physical solutions that meet design specifications [4, 5]. One way to achieve this goal is by utilizing the Function-Behavior-Structure (FBS) framework [6].

The FBS framework refers to conceptualizing objects based on their function, structure, and behavior, which mainly transforms presumed functions into behavioral descriptions [6]. A function is defined as a set of tasks that enable a product to fulfill its design purpose [7]. Functional modeling supports and offers decision guidance for a product design to understand an overall product function during the conceptual design phase, where design decisions are yet to be made [7, 8]. Although systematic, function-based design can be subjective due to how designers interpret the intended function. The human interpretation of function has often generated unstructured and unbalanced repositories, thereby affecting the accuracy of behavior predictions and subsequent structural decisions. Therefore, a significant challenge arises from the limited availability of well-structured function data, necessitating a more accurate, automated way of generating function data.

With the advent of large language models (LLMs), such as GPTs and Llama, language understanding and generation have been significantly advanced by training billions of parameters on vast amounts of text [9]. LLMs can identify patterns and relationships within language and apply these to generating responses, solving problems, and writing. LLMs have the ability to

perform a wide range of natural language processing tasks without the need for task-specific training because they learn broader language patterns and relationships rather than being confined to specific tasks [10]. This versatility makes LLMs ideally suited for various language-related tasks.

However, directly applying LLMs to complex domain-specific contextual applications faces many challenges caused by heterogeneity, uniqueness, sophistication, and constraints of domain data [11]. Consequently, efforts have shifted toward adapting foundational, pre-trained, general-purpose LLMs to domain-specific applications. Design and manufacturing are particularly critical application areas, as the knowledge within these fields contains complex structures due to the intricate nature of a product manufacturing lifecycle from design to build. This necessitates domain adaptation (DA) to bridge the knowledge gap between foundational LLM capabilities and the distinct characteristics of design and manufacturing datasets.

Given the capabilities of LLMs in learning and generating textual data, as well as the challenge of the lack of functional data in mechanical design, this study explores two research questions: (1) Are LLMs sufficiently capable of generating accurate functional annotations for mechanical components? 2) How effectively can customizing LLMs infer the functions of new mechanical components based on textual data? To address the research questions, the study proposes an LLM DA framework for the function classification of mechanical assembly parts. The case study utilizes the Oregon State Design Repository (OSDR) as the domain knowledge LLM customization. Later, the study applied the domain-adapted LLM to A Big Cad (ABC) data set to generate novel classified functional data.

The remainder of the paper is as follows: Section 2 reviews the literature. Section 3 introduces the problem statement. Section 4 proposes the methodology. Section 5 presents a case study. Section 6 discusses the advantages and limitations. Section 7 concludes the study.

2. LITERATURE REVIEW

This section reviews the latest studies of functional modeling and the role of LLMs in mechanical design.

2.1. Function Annotation in Mechanical Design

Function refers to a system's general input and output relationship that performs a task [12]. In mechanical design, function refers to the set of tasks that enable a product to perform its intended function [7]. Function-based tools and modeling offer decision guidance for product design to understand an overall product function during the conceptual design phase, where design decisions are yet to be made [7, 8].

In this regard, Ferrero et al. proposed a graph neural network (GNN)-based framework to classify mechanical component functions to address the scarcity of function-based design data [7]. Han et al. [13] developed a three-dimensional (3D) product-based methodology to construct a semantic functional ontology for semantic annotation of functions for mechanical parts in conceptual design. The goal was to bridge the gap to capture the correlation of function and structure and retrieve 3D models based on their specific functions at the early design stage.

2.2. Large Language Models in Mechanical Design

Language is the way humans communicate, self-express, and interact with machines [14]. The need for language-based models arises from the challenges of handling complex language-based tasks, such as translation, summarization, information retrieval, and conversations with the machines [14]. Recent advances in Natural Language Processing (NLP), driven by transformers, enabled the development of Large Language Models (LLMs) that achieve near-human performance across various domains [15].

LLMs demonstrate remarkable capabilities and advantages in performing complex NLP tasks, such as capturing the underlying data distribution of sequential, contextual data and generating content that closely mimics the patterns of their training data [16–18]. However, directly applying LLMs to complex domain-specific contextual applications faces many challenges caused by heterogeneity, uniqueness, sophistication, and constraints of domain data [11]. Consequently, efforts have shifted toward adapting foundational, pre-trained, general-purpose LLMs to domain-specific applications. In general, there are three approaches for LLM DA in literature : (1) External augmentation, (2) Prompt crafting, and (3) Model fine-tuning, utilizing them heavily depends on the level of accessibility of LLMs [11].

Several studies utilized the capabilities of LLMs in conceptual design. For instance, Chen et al. proposed an LLM-guided framework to generate design concepts using the FBS method [19]. Li et al. developed a method, LLM4CAD, to enable multimodal LLMs in 3D CAD generation utilizing different input modalities, including the text-only, text+sketch, text+image, and text+ sketch+image data [20]. Ataei et al. developed an LLM-based framework, Elicatron, that utilizes to automate and improve requirements elicitation in product design [21]. The study utilized LLM agents to simulate a wide range of user personas, which can explore diverse and latent needs that traditional methods overlooked.

While these studies showcase the promising role of LLMs in conceptual design, they still face limitations, such as not fully utilizing domain-specific knowledge through DA techniques. Design and manufacturing are particularly critical application areas, as the knowledge within these fields contains complex structures due to the intricate nature of a product manufacturing lifecycle from design to build. This highlights the need for LLM customization techniques to fully capture domain-specific knowledge.

3. PROBLEM STATEMENT

Functional modeling supports and offers decision guidance for a product design to understand an overall product function during the conceptual design phase, where design decisions are yet to be made [7, 8]. However, function-based design is subjective due to how designers interpret the intended function. The human interpretation of function has often generated unstructured and unbalanced repositories, thereby affecting the accuracy of behavior predictions and subsequent structural decisions. Therefore, a significant challenge arises from the limited availability of well-structured function data, necessitating a more accurate, automated way of generating function data.

The traditional ways of function annotations for mechanical assembly components are mainly based on manually curated data or domain-specific rules. These methods are not only time-consuming and inaccurate, but they cannot capture the complex correlation between structure and function. These challenges limit traditional methods’ scalability and adaptability for conceptual design generation. Recent studies have explored the capabilities of Machine Learning (ML), such as GNNs, to automatically classify mechanical parts’ function [7]. Although graph models have demonstrated strong capabilities across various domains [22–25], these methods lack the mechanism to capture the complex distribution of sequential, contextual data.

To address the challenge of lack of function data and the limited capability of traditional ML methods, this study proposed a supervised DA method for LLMs to generate synthesized function-classified data [26].

4. METHODOLOGY

This section introduces a supervised DA framework utilizing a fine-tuning approach. The method aims to automatically classify and annotate mechanical assembly components to generate synthesized functional labels in the conceptual design of a product design. Figure 1 indicates the overview of the proposed framework.

DA relies on two distinct domains: a source environment (general domain) and a target domain (assembly design domain). In this study, a generative language model, pre-trained on the source domain to capture the general population distribution of the source data, adapts its knowledge of assembly design labels from the source domain to improve performance in the target domain [27]. The goal is to fine-tune the model’s distribution toward assembly data.

To achieve this, the study employs function-labeled contextual design data as the target domain for supervised fine-tuning, serving as the DA method. The function labels are derived from the training dataset’s conceptual design phases, while the assembly data represent the mechanical parts and their assembly name of the outputs of the corresponding detailed design phases. Once fine-tuned, the language model generates synthesized label samples of assembly designs based on detailed design phases for *the target domain’s different design cases/scenarios*. These synthesized samples, guided by the fine-tuned distribution, enhance the function-structure database used in the target domain’s design processes, ultimately supporting the transition from conceptual to detailed design phases.

Function labels play a crucial role in guiding the generative model’s learning process by mapping design features to their intended functionalities. These labels categorize design elements based on their purpose, enabling the model to generate meaningful assembly design samples aligned with conceptual design requirements. For example, in an assembly design domain, function labels may include “support,” “fasten,” “transmit motion,” or “seal,” each defining a specific role in the overall design structure.

The function-labeled contextual design data ($D_{T_{\text{labeled}}}$) indicates the relationships between design components and their intended functionalities. $D_{T_{\text{labeled}}}$ consists of n columns, where the first $n - 1$ columns represent design features, and the final column

contains function labels (classes) that guide the training process by demonstrating the desired behavior for the model. This structured data format is designed to align with the model’s learning process, where separating features from function labels allows the transformer-based architecture to establish correlations between design characteristics and functional intent. By structuring the dataset in this way, the model can effectively capture latent relationships between design features and their corresponding functionalities, improving generalization and predictive accuracy in generating assembly designs.

The function labels data (D_L) comprises a set of m function classes that define the labels of interest in the conceptual design phase for the training process. Additionally, the function definition dataset ($D_{L_{\text{dis}}}$) provides textual descriptions or structured representations of these m function classes, serving as a reference for interpreting and validating the assigned labels[28]. Unlike traditional design methods, which rely on manual classification and domain expertise to assign functions, the incorporation of textual descriptions uniquely enhances labeling and design processes through the transformer model’s ability to understand contextual information [29].

The self-attention mechanism in transformers enables the model to extract nuanced relationships between words and concepts, allowing it to recognize functional similarities even when design elements vary in structure or composition. This capability empowers the model to automate and refine the labeling process, reducing human effort and improving consistency in function classification. By leveraging structured function labels and contextual descriptions, the model learns a more comprehensive representation of assembly design principles, ultimately enhancing its ability to generate functionally coherent synthesized label data that support the conceptual design phase.

We defined three hyper-parameters, $H = [E, B, LR]$, where E , B , and LR , are the number of epochs, batch size, and learning rate multiplier, respectively, to enhance LLM’s performance. During each tuning iteration, we randomly select and evaluate combinations of hyperparameters from a specified range to explore a large parameter space as written in Equation 1:

$$H = \{(E, B, LR) \mid E \in \mathcal{E}, B \in \mathcal{B}, LR \in \mathcal{L}\} \quad (1)$$

The next step is the fine-tuning step, which is the main step of the proposed pipeline. Utilizing the randomly chosen hyperparameters, the LLM receives $D_{T_{\text{labeled}}}$, D_L , $D_{L_{\text{dis}}}$ as the input with a prompt and predicts the function labels from D_L . The fine tuning process is represented in Equation 2 [30]:

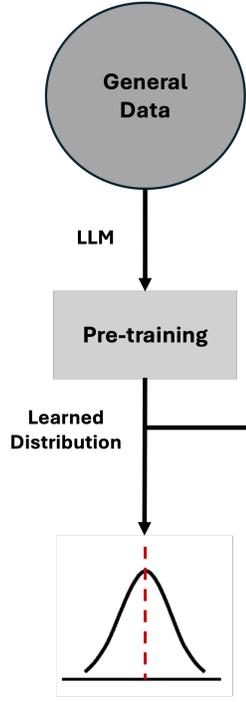
$$LLM_{\text{pre-trained}}(\{D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}}\}) \rightarrow LLM_{\text{fine-tuned}} \quad (2)$$

More specifically, the LLM utilizes the prompt to receive the tokenized input (d_1, \dots, d_s) and predict label y from D_L . Equation 3 indicates the output for the final layer of the transformer in the LLM which predicts y :

$$P(y \mid d_1, \dots, d_s) = \text{softmax}(h_s^l W_y) \quad (3)$$

, where h_s^l and W_y are the hidden state for the final layer and weight matrix for the classification task, respectively. During each fine-tuning iteration, The objective function, which is based

Source Domain



Target Domain

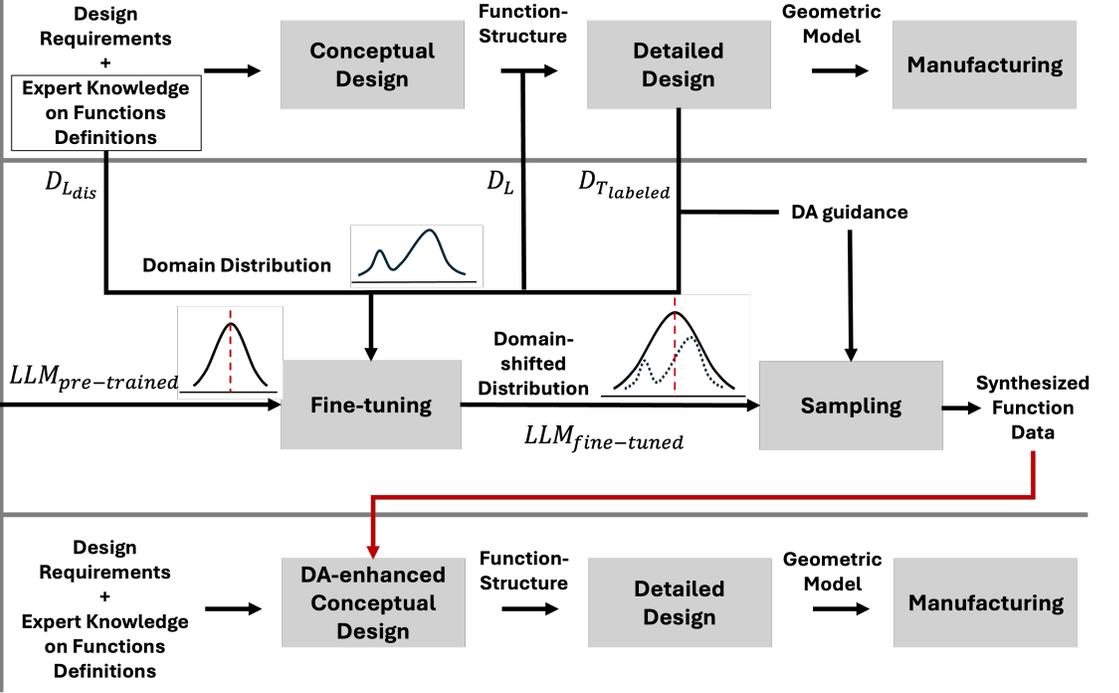


FIGURE 1: AN OVERVIEW OF A SUPERVISED DA VIA FINE-TUNING

on the minimization of the cross entropy function is shown in Equation 4:

$$L(D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}}) = - \sum_{(d,y)} \log P(y | d_1, \dots, d_s) \quad (4)$$

The final objective function and updating hyperparameters are shown in Equations 5, and 6:

$$L(D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}}) = - \sum_{e=1}^E \sum_{i=1}^{\frac{N}{B}} \sum_{(d,y) \in \mathcal{B}_i} \log P(y | d_1, \dots, d_s) \quad (5)$$

$$W_y^{t+1} = W_y^t - LR \cdot \nabla_{W_y} L(D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}}) \quad (6)$$

, where N is the total number of samples, i is the i -th batch, and $\nabla_{W_y} L(D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}})$ is the gradient of the loss function.

We applied the fine-tuned LLM to a test function labeled mechanical parts data (D_{test}) to predict the labels and evaluate the performance of the LLM during the evaluation step. For evaluation, we utilized accuracy, precision, recall, and F1 metrics as shown in Equation 7:

$$M = \frac{1}{S} \sum_{i=1}^S \mathcal{F}(\hat{y}_i, y_i) \quad (7)$$

, where M is an evaluation metric, S indicates the number of samples for $D_{T_{\text{labeled}}}$, \mathcal{F} is the evaluation function, and \hat{y} is the predicted function class for each mechanical part.

The evaluation step is necessary to choose a list of optimized hyperparameters as in Equation 8:

$$H' = \{(E', B', LR') | E' \in \mathcal{E}, B' \in \mathcal{B}, LR' \in \mathcal{L}\} \quad (8)$$

We selected the fine-tuned LLM with the highest values of evaluation metrics to choose H' . Algorithm 1 indicates the fine-tuning steps.

Finally, the design domain-adapted classifier with the optimized hyperparameters values is applied for the function annotation of unlabeled design data ($D_{\text{unlabeled}}$) for a different design scenario. The classification task is represented in Equation 9:

$$LLM_{\text{fine-tuned}}(D_{\text{unlabeled}}, D_L, D_{L_{\text{dis}}}) \rightarrow \hat{y}(D_{\text{unlabeled}}) \quad (9)$$

, where $\hat{y} \in D_L$ is the predicted function for a mechanical part in the unlabeled design data. The study applied the proposed domain-adapted LLM classifier to a conceptual design case study to generate synthesized function annotations for mechanical assembly components.

Algorithm 1 SUPERVISED DA VIA FINE-TUNING

```
1: INPUT:  $LLM_{\text{pre-trained}}, D_{T_{\text{labeled}}}, D_L, D_{L_{\text{dis}}}$ , and  $H = [E, B, LR]$ 
2: OUTPUT:  $H' = [E', B', LR']$ , and  $LLM_{\text{fine-tuned}}$ 
3: Initialize hyperparameter combinations:  $H = \{(E, B, LR) | E \in \mathcal{E}, B \in \mathcal{B}, LR \in \mathcal{LR}\}$ 
4: for each hyperparameter set  $(E, B, LR) \in H$  do
5:   Initialize model:  $LLM_{\text{fine-tuned}} \leftarrow LLM_{\text{pre-trained}}$ 
6:   for epoch  $e = 1, 2, \dots, E$  do
7:     for each batch  $(d, y) \in D_{T_{\text{labeled}}}$  do
8:       Compute loss using Equation (5)
9:       Compute gradient using Equation (6)
10:      Update parameters using Equation (6)
11:      Compute predictions using Equation (3)
12:     end for
13:   end for
14:   Evaluate model using Equation (7)
15: end for
16: Select optimal hyperparameters  $(E', B', LR')$  using Equation (8)
17: return  $H' = [E', B', LR']$ , and  $LLM_{\text{fine-tuned}}$ 
```

5. CASE STUDY

The case study section indicates the capability of the proposed method for a function annotation task for mechanical assembly parts. The section, initially, explains the datasets used for DA and classification steps. Then, the experimental details of fine-tuning are highlighted. Finally, we discuss the results. Figure 2 provides an overview of the case study.

5.1. Data

5.1.1. Oregon State Design Repository The OSDR includes information on various mechanical assembly parts, their basis, system type, function, sub-function, flow, and assembly relationships [31, 32]. The OSDR contains 15636 samples. We removed the duplicate product values and the ones with a lack of information regarding their basis, system name, and function. The pre-processed OSDR contains 7568 unique parts. Table 1 is an illustration of the pre-processed OSDR dataset. The OSDR includes the products' functions and sub-functions in a three-tier hierarchy. We extracted the function hierarchy from the data as shown in Figure 3. The first-tier functions are the labels of interest in this study for the classification task, containing eight function classes. Table 2 indicates the definition of function classes [28].

5.1.2. Geometric Model Data The ABC data set consists of one million high-quality CAD geometric models of mechanical parts [33]. The models are defined by parametric surfaces and are accompanied by accurate ground truth information regarding patch decomposition, sharp feature annotations, and differential properties. Figure 4 indicates samples of CAD models for mechanical parts in the ABC data. The dataset is split into 100 chunks, each of which contains 10000 CAD models of mechanical parts. We matched the samples' names in each chunk with those in the OSDR dataset. The matched dataset contains 27950 matched samples. We removed the duplicate samples. Finally,

the matched dataset contains 6786 unique ABC mechanical parts with their names and their assembly names.

5.2. Fine-Tuning

We used GPT 3.5 Turbo as the pre-trained LLM for the fine-tuning task. The OSDR, as the function-labeled data, the eight function classes, and the class definitions are the inputs of the LLM. We split the OSDR data to %90 and %10 for training and testing, respectively. Figure 5 indicates the training and test sets of the OSDR data. Additionally, a prompt is used to guide the LLM for the classification task, utilizing the function classes and their definitions:

Given the following function classes and their definitions, {function classes: definitions}, what is the function of a part {part name} in the system {assembly name}?

We set different values for hyperparameters during the fine-tuning step. Additionally, to improve the fine-tuning results, we repeated the same experiment with different sample sizes for the training set.

5.3. Results

Table 3 illustrates the evaluation results during the fine-tuning step for different hyperparameter values. Accuracy, F1-score, precision, and recall are calculated for both training and test sets to choose the optimum hyperparameter values.

To improve the fine-tuning results, we trained the model with different training sample sizes for the optimized hyperparameter values. Table 4 indicates the results of metrics for different training sample sizes. As can be seen, decreasing the data size improves the training and test set prediction results. More specifically, by utilizing only 681 samples of the OSDR data set, we were able to significantly improve the classification results from %4 to %90. Figure 6 indicates the confusion matrices for pre-trained LLM with different training set sizes results for the test set.

As a comparative study, we compared the prediction accuracy of the fine-tuned models, containing different training sample sizes, with that of the latest pre-trained GPT models, as can be seen in Figure 7. Table 5 indicates the detailed results of the pre-trained GPTs. The fine-tuned LLM with different training sample sizes all outperformed the pre-trained GPT models, indicating the importance of fine-tuning LLMs for classifying mechanical assembly parts' functions.

We applied the fine-tuned model with the highest accuracy to the matched ABC dataset for function classification. Figure 8 indicates the results of function annotation for the ABC data set. As shown, %99 of the classified labels fall within the pre-defined function classes, with only %1 of the samples being misclassified. Additionally, 'channel' and 'support' are the most frequently classified labels, consistent with the function class distribution in the OSDR data. Table 6 indicates samples of newly generated function-labeled ABC data.

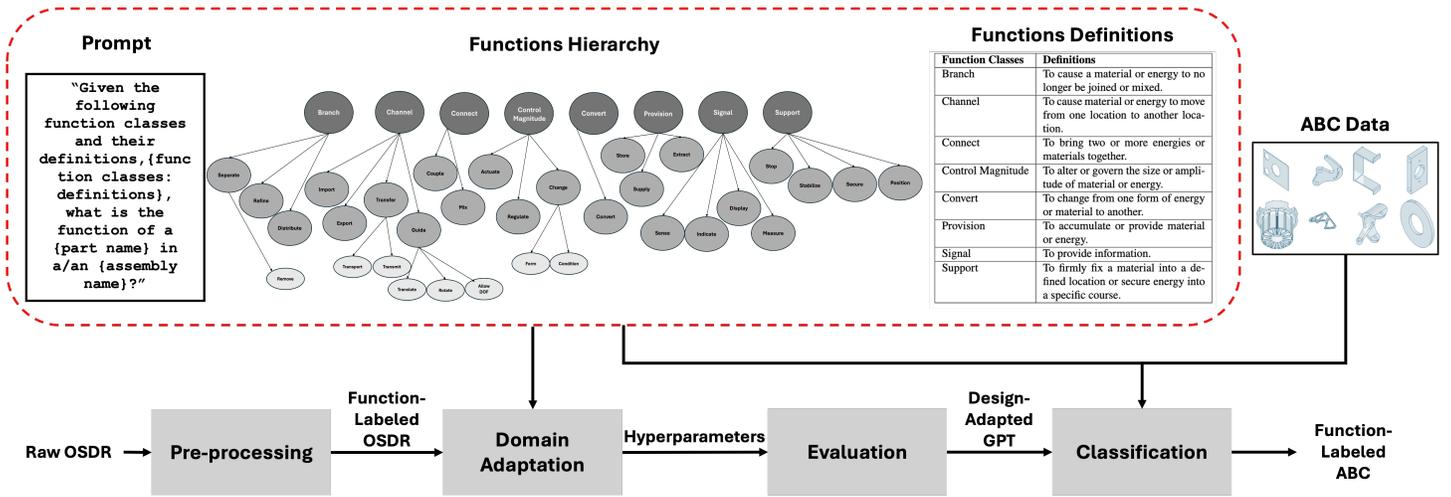


FIGURE 2: FUNCTION CLASSIFICATION OF MECHANICAL ASSEMBLY PARTS

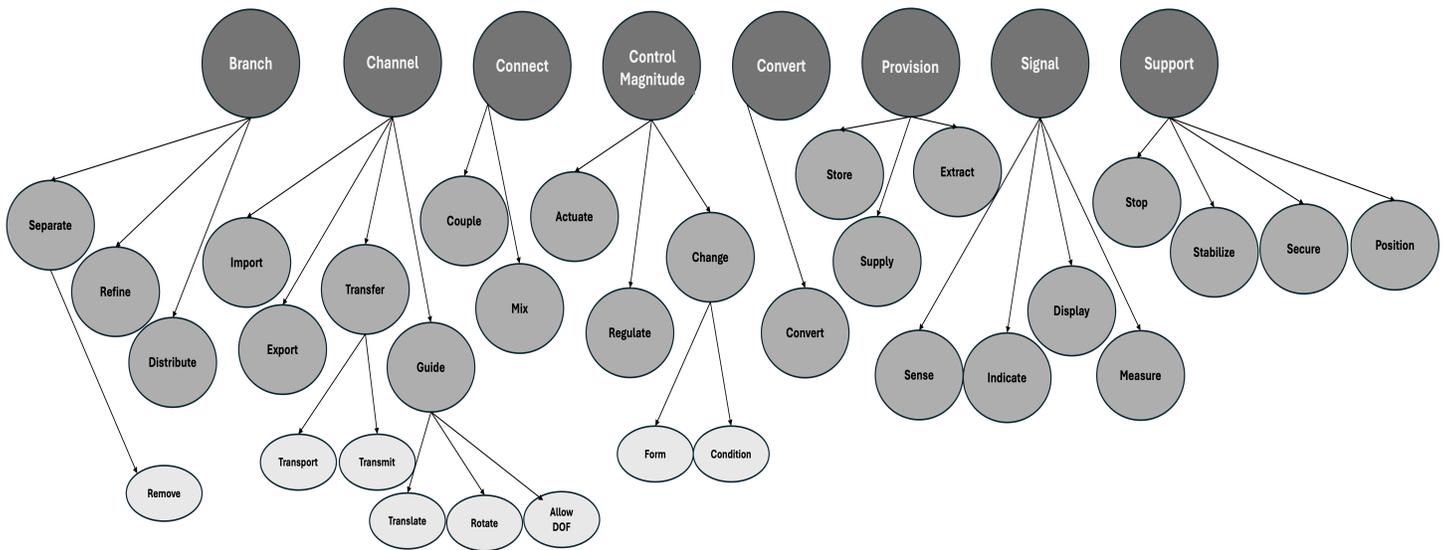


FIGURE 3: A THREE-TIER FUNCTION HIERARCHY [28]

TABLE 1: PRE-PROCESSED FUNCTION-LABELLED DESIGN DATA

name	component_basis	sys_name	subfunction_basis	parent_subfunction
master cylinder	hydraulic pump	brake system	import	channel
drum	container	brake system	import	channel
caliper	clamp	brake system	import	channel
caliper	clamp	brake system	export	channel
caliper	clamp	brake system	regulate	control magnitude
pedal	lever	brake system	guide	channel
fin	airfoil	brake system	distribute	branch
fin	airfoil	brake system	export	channel
brake cylinder	hydraulic pump	brake system	export	channel
rotor	wheel	brake system	import	channel
battery tray	housing	digger dog	transfer	channel

6. DISCUSSION

DA of LLMs has significantly improved the function classification results of mechanical assembly parts as compared to general-purpose LLMs, as shown in Figure 7. Moreover, LLM's

unique capabilities allowed the incorporation of function definitions as an extra input for both DA and classification steps. This integration, which was not feasible by traditional ML methods, enabled the model to better understand the context and re-

TABLE 2: FUNCTION CLASSES AND THEIR DEFINITIONS [28]

Function Classes	Definitions
Branch	To cause a material or energy to no longer be joined or mixed.
Channel	To cause material or energy to move from one location to another location.
Connect	To bring two or more energies or materials together.
Control Magnitude	To alter or govern the size or amplitude of material or energy.
Convert	To change from one form of energy or material to another.
Provision	To accumulate or provide material or energy.
Signal	To provide information.
Support	To firmly fix a material into a defined location or secure energy into a specific course.

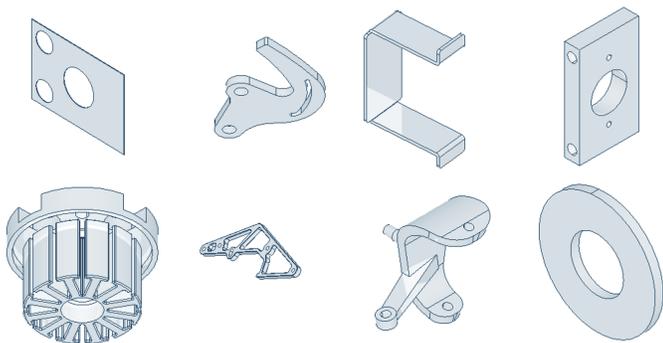


FIGURE 4: SAMPLES OF MECHANICAL PARTS IN ABC DATA [33].

relationships between component descriptions and their associated functions.

Despite the noticeable capability of LLMs DA in generating synthesized function-labeled conceptual design data, several limitations and challenges were observed. First, the prediction accuracy during fine-tuning improved when the number of training samples decreased. One possible reason could be that LLMs, specifically GPTs, for this specific domain, struggle to handle larger datasets, resulting in overfitting.

Second, the OSDR data used for the DA step is imbalanced, with certain function classes, such as channel and support, being heavily overrepresented, as shown in Figure 5. The model’s strong performance in these classes is likely a reflection of their dominance in the training data. In contrast, the accuracy for less common function classes was significantly lower, revealing a bias towards more frequently occurring function classes as shown in Figure 8.

Third, the DA prediction results heavily depend on the pre-defined function labels, limiting the domain-adapted classifier’s ability to distinguish other possible function classes. As shown

in Figure 8 %99 of the classified labels fit within the existing classes.

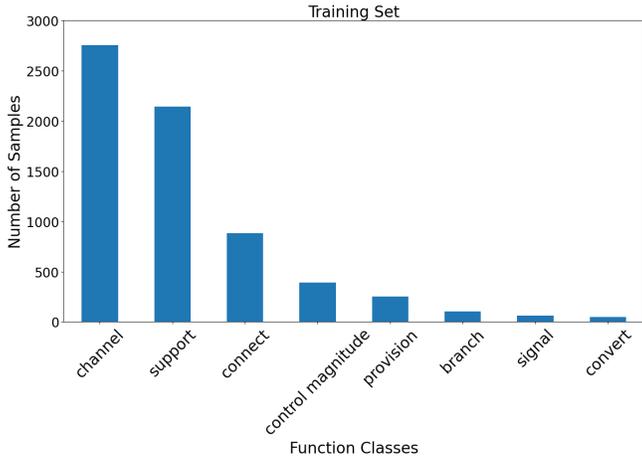
7. CONCLUDING REMARKS AND FUTURE WORKS

The study developed a supervised DA framework for the function classification of mechanical assembly parts. The methodology aimed to fine-tune a foundational LLM model, GPT 3.5 Turbo, utilizing the OSDR data. The DA approach successfully generated synthesized functional labels for the ABC data set to address the lack of function data in conceptual design.

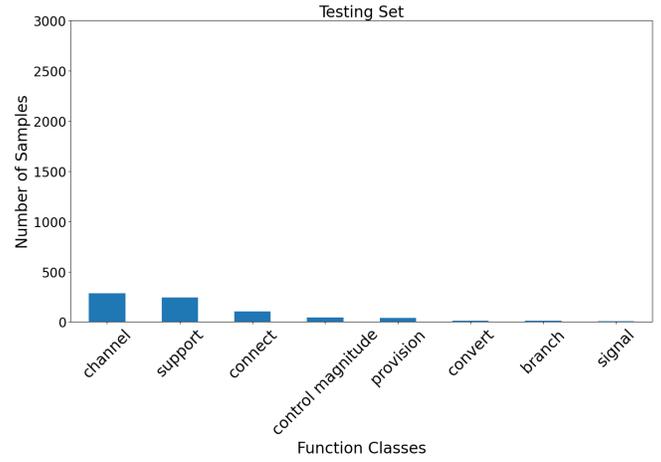
Future work will focus on improving the model performance, utilizing reasoning-enhanced and chain-of-thoughts prompting approaches. Additionally, DA will be applied to Vision-Language models utilizing both textual and vision design data to predict the functions of mechanical assembly components.

REFERENCES

- [1] M. Baxter, *Product Design*, CRC Press, 2018.
- [2] M. Zhang, F. Sui, A. Liu, F. Tao, and A. Y. C. Nee, “Digital twin driven smart product design framework,” in *Digital Twin Driven Smart Design*, Elsevier, 2020, pp. 3–32.
- [3] S. M. Ibbotson and S. Kara, “An approach to identify the factors that affect a product’s life time energy consumption during the concept design stage,” *Procedia CIRP*, vol. 70, pp. 223–228, 2018.
- [4] Y. Wang, A. Liu, F. Tao, and A. Y. C. Nee, “Digital twin driven conceptual design,” in *Digital Twin Driven Smart Design*, Elsevier, 2020, pp. 33–66.
- [5] W. Hsu and I. M. Y. Woon, “Current research in the conceptual design of mechanical products,” *Computer-Aided Design*, vol. 30, no. 5, pp. 377–389, 1998.
- [6] S. Al-Fedaghi, “Function-behavior-structure model of design: An alternative approach,” *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 7, 2016.
- [7] V. Ferrero, B. DuPont, K. Hassani, and D. Grandi, “Classifying component function in product assemblies with graph neural networks,” *Journal of Mechanical Design*, vol. 144, no. 2, p. 021406, 2022.
- [8] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, “A functional basis for engineering design: reconciling and evolving previous efforts,” *Research in Engineering Design*, vol. 13, pp. 65–82, 2002.
- [9] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib, M. M. J. Mim, J. Ahmad, M. E. Ali, and S. Azam, “A review on large language models: Architectures, applications, taxonomies, open issues and challenges,” *IEEE Access*, vol. 12, pp. 26839–26874, 2024.
- [10] P. P. Ray, “ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope,” *Internet of Things and Cyber-Physical Systems*, vol. 3, pp. 121–154, 2023.
- [11] C. Ling, X. Zhao, J. Lu, C. Deng, C. Zheng, J. Wang, T. Chowdhury, Y. Li, H. Cui, X. Zhang, et al., “Domain specialization as the key to make large language models disruptive: A comprehensive survey,” *arXiv preprint arXiv:2305.18703*, 2023.



(a) Train set.



(b) Test set.

FIGURE 5: THE DISTRIBUTION OF TRAINING AND TEST SET DATA FOR FINE-TUNING STEP.

TABLE 3: EXPERIMENTAL RESULTS

Hyperparameters			Train Accuracy	Train (Weighted)			Train (Macro)			Train (Micro)			Test Accuracy	Test (Weighted)			Test (Macro)			Test (Micro)				
Number of Epochs	Batch Size	Learning Rate Multiplier		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall		
12	20	20	0.64	0.64	0.64	0.64	0.64	0.64	0.31	0.31	0.31	0.63	0.63	0.63	0.42	0.43	0.42	0.42	0.29	0.30	0.28	0.4	0.4	0.4
20	20	20	0.65	0.65	0.65	0.65	0.65	0.65	0.28	0.28	0.28	0.65	0.65	0.65	0.37	0.37	0.37	0.37	0.18	0.18	0.18	0.37	0.37	0.37
10	48	30	0.63	0.63	0.63	0.63	0.63	0.63	0.17	0.17	0.17	0.63	0.63	0.63	0.41	0.41	0.41	0.41	0.22	0.24	0.22	0.41	0.41	0.41
30	48	30	0.66	0.66	0.66	0.66	0.66	0.66	0.51	0.51	0.51	0.66	0.66	0.66	0.37	0.37	0.37	0.37	0.20	0.21	0.20	0.37	0.37	0.37
8	24	0.5	0.40	0.40	0.43	0.40	0.40	0.40	0.10	0.11	0.10	0.40	0.40	0.40	0.39	0.40	0.43	0.39	0.20	0.20	0.20	0.39	0.39	0.39
15	24	20	0.64	0.64	0.65	0.64	0.64	0.64	0.27	0.27	0.27	0.64	0.64	0.64	0.41	0.41	0.41	0.41	0.25	0.25	0.24	0.41	0.41	0.41
12	100	20	0.63	0.63	0.63	0.63	0.63	0.63	0.29	0.29	0.30	0.63	0.63	0.63	0.42	0.42	0.42	0.42	0.29	0.30	0.28	0.42	0.42	0.42
12	20	0.6	0.42	0.43	0.45	0.42	0.42	0.42	0.15	0.14	0.16	0.42	0.42	0.42	0.35	0.36	0.38	0.35	0.16	0.16	0.18	0.35	0.35	0.35
12	20	40	0.33	0.33	0.33	0.33	0.33	0.33	0.14	0.15	0.14	0.33	0.33	0.33	0.32	0.31	0.31	0.32	0.14	0.15	0.14	0.32	0.32	0.32
1	20	20	0.41	0.42	0.42	0.41	0.41	0.41	0.05	0.05	0.06	0.41	0.41	0.41	0.39	0.39	0.40	0.39	0.14	0.14	0.14	0.39	0.39	0.39

TABLE 4: PERFORMANCE METRICS FOR DIFFERENT TRAINING SET SIZES

Training Set Size (%)	Train Accuracy	Train (Weighted)			Train (Macro)			Train (Micro)			Test Accuracy	Test (Weighted)			Test (Macro)			Test (Micro)		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
10% (681 samples)	0.93	0.93	0.94	0.93	0.82	0.80	0.84	0.93	0.93	0.93	0.89	0.89	0.90	0.89	0.64	0.63	0.66	0.89	0.89	0.89
30% (2043 samples)	0.83	0.83	0.84	0.83	0.64	0.64	0.64	0.83	0.83	0.83	0.84	0.84	0.85	0.84	0.84	0.85	0.83	0.84	0.84	0.84
40% (2724 samples)	0.78	0.78	0.78	0.78	0.66	0.64	0.67	0.78	0.78	0.78	0.79	0.79	0.79	0.79	0.56	0.56	0.56	0.79	0.79	0.79
50% (3406 samples)	0.75	0.75	0.75	0.75	0.39	0.39	0.40	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.72	0.71	0.72	0.75	0.75	0.75
75% (5109 samples)	0.68	0.69	0.69	0.68	0.31	0.30	0.31	0.68	0.68	0.68	0.68	0.68	0.69	0.68	0.55	0.56	0.55	0.68	0.68	0.68
90% (6130 samples)	0.64	0.64	0.64	0.64	0.34	0.34	0.34	0.64	0.64	0.64	0.63	0.63	0.64	0.63	0.56	0.57	0.56	0.63	0.63	0.63

TABLE 5: PERFORMANCE EVALUATION OF PRE-TRAINED GPT MODELS ON TEST DATA

Model	Accuracy	Weighted			Macro			Micro		
		F1 Score	Precision	Recall	F1 Score	Precision	Recall	F1 Score	Precision	Recall
GPT-3.5 Turbo	0.04	0.06	0.15	0.04	0.02	0.04	0.01	0.04	0.04	0.04
GPT-4 Turbo	0.27	0.28	0.39	0.27	0.06	0.07	0.07	0.27	0.27	0.27
GPT-4o Mini	0.20	0.22	0.43	0.20	0.07	0.09	0.09	0.20	0.20	0.20

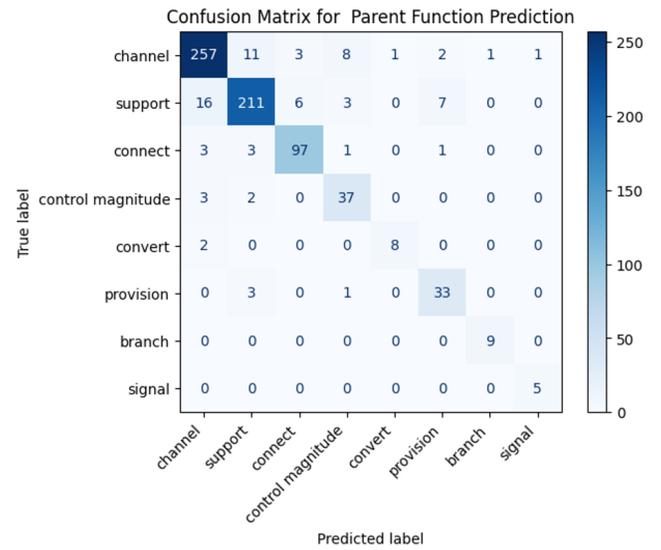
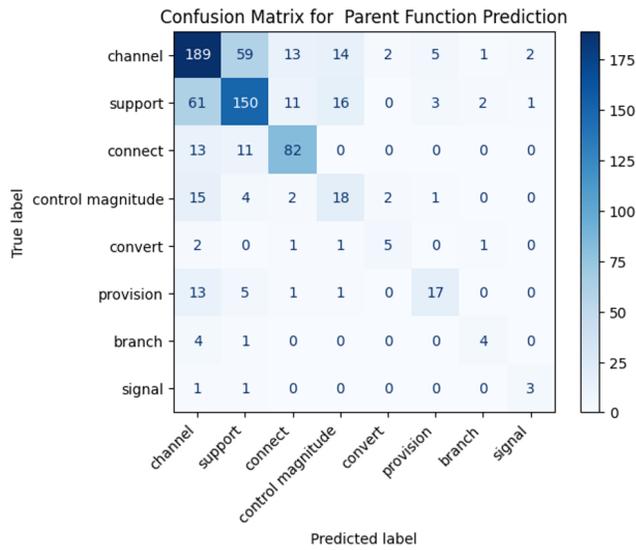
[12] C. F. Kirschman and G. M. Fadel, “Classifying functions for mechanical design,” 1998. [Online]. Available: [Publisher not specified]

[13] Z. Han, H. Zhang, W. He, L. Ba, and Q. Yuan, “Automatic annotation of functional semantics for 3D product model based on latent functional semantics,” *Scientific Programming*, vol. 2023, no. 1, p. 9885859, 2023.

[14] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M.

Usman, N. Akhtar, N. Barnes, and A. Mian, “A comprehensive overview of large language models,” *arXiv preprint arXiv:2307.06435*, 2023.

[15] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, “Transformers: ‘The end of history’ for natural language processing?,” in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2021, pp. 677–693.

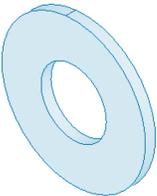


(a) Confusion Matrix for Testing Set for the Fine-Tuned Model with %90 of Training Samples

(b) Confusion matrix for Testing Set for the Fine-Tuned Model with %10 of Training Samples

FIGURE 6: THE CONFUSION MATRICES FOR THE TEST SET WITH DIFFERENT TRAINING SAMPLE SIZES.

TABLE 6: SAMPLES OF GENERATED SYNTHESIZED FUNCTION LABELS FOR THE ABC MECHANICAL PARTS

Chunk	Assembly id	Assembly name	Part id	Part name	Function	Image
0	390	Tablet Stand	39	Washer	Support	
0	253	Independent Front Suspension	18	Spindle	Channel	

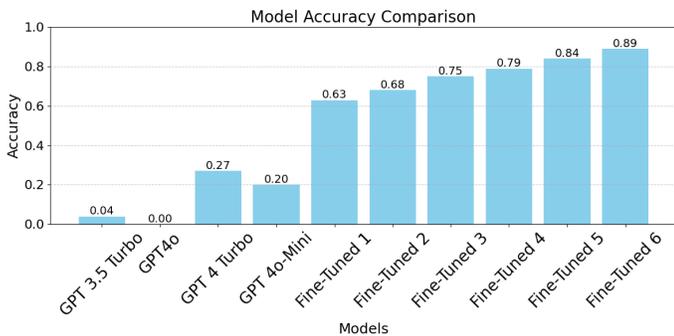


FIGURE 7: THE COMPARISON OF ACCURACY FOR PRE-TRAINED GPT MODELS AND FINE-TUNED MODELS

[16] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili, et al., "A survey on large language models: Applications, challenges, limitations, and practical usage," *Authorea Preprints*, vol. 3, 2023.

[17] E. Farahmand, R. R. Azghan, N. T. Chatrudi, E. Kim, G. K. Gudur, E. Thomaz, G. Pedrielli, P. Turaga, and H. Ghasemzadeh, "Attengluc: Multimodal transformer-based blood glucose forecasting on AI-READI dataset," *arXiv preprint arXiv:2502.09919*, 2025.

[18] S. M. Alsadat, J.-R. Gaglione, D. Neider, U. Topcu, and Z. Xu, "Using large language models to automate and expedite reinforcement learning with reward machine," *arXiv preprint arXiv:2402.07069*, 2024.

[19] L. Chen, H. Zuo, Z. Cai, Y. Yin, Y. Zhang, L. Sun, P. Childs, and B. Wang, "Toward controllable generative de-

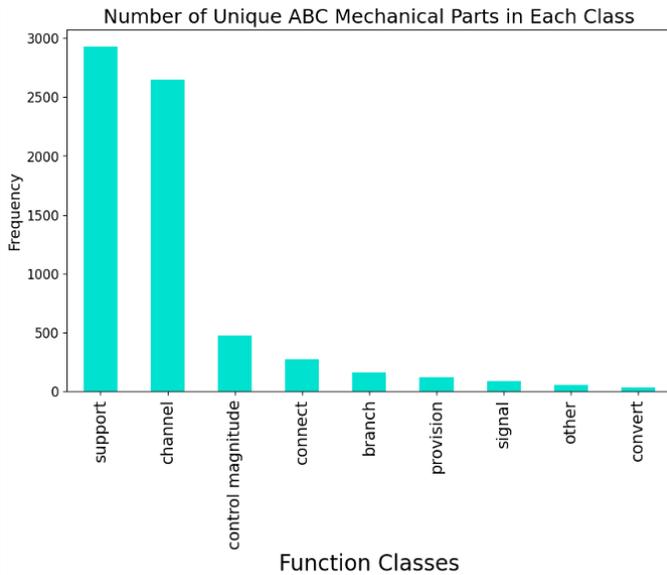


FIGURE 8: GENERATED SYNTHESIZED FUNCTIONS OF ABC MECHANICAL ASSEMBLY PARTS

sign: A conceptual design generation approach leveraging the Function–Behavior–Structure ontology and large language models,” *Journal of Mechanical Design*, vol. 146, no. 12, 2024.

- [20] X. Li, Y. Sun, and Z. Sha, “LLM4CAD: Multi-modal large language models for 3D computer-aided design generation,” in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 88407, pp. V006T06A015, 2024.
- [21] M. Ataei, H. Cheong, D. Grandi, Y. Wang, N. Morris, and A. Tessier, “Elicitron: A large language model agent-based simulation framework for design requirements elicitation,” *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, 2025.
- [22] K. Sabri-Laghaie, S. Jafarzadeh Ghouschi, F. Elhambakhsh, and A. Mardani, “Monitoring blockchain cryptocurrency transactions to improve the trustworthiness of the fourth industrial revolution (Industry 4.0),” *Algorithms*, vol. 13, no. 12, p. 312, 2020.
- [23] F. Elhambakhsh and M. Saidi-Mehrabad, “Developing a method for modeling and monitoring of dynamic networks using latent variables,” *International Journal of Industrial Engineering & Production Research*, vol. 32, no. 1, pp. 29–36, 2021.
- [24] F. Elhambakhsh, K. Sabri-Laghaie, and R. Noorossana, “A latent space model and Hotelling’s T^2 control chart to monitor the networks of Covid-19 symptoms,” *Journal of Applied Statistics*, vol. 50, no. 11–12, pp. 2450–2472, 2023.
- [25] F. Elhambakhsh and K. Sabri-Laghaie, “The SCAN method to monitor cryptocurrency transactions,” *International Journal of Industrial Engineering*, vol. 33, no. 1, pp. 1–14, 2022.
- [26] V. B. Parthasarathy, A. Zafar, A. Khan, and A. Shahid, “The ultimate guide to fine-tuning LLMs from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities,” *arXiv preprint arXiv:2408.13296*, 2024.
- [27] S. Goodman, H. Greenspan, and J. Goldberger, “Supervised domain adaptation by transferring both the parameter set and its gradient,” *Neurocomputing*, vol. 560, p. 126828, 2023.
- [28] R. B. Stone and K. L. Wood, “Development of a functional basis for design,” in *Int. Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 19739, pp. 261–275, 1999.
- [29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [30] Y. Li, H. Ko, and F. Ameri, “Integrating graph retrieval-augmented generation with large language models for supplier discovery,” *Journal of Computing and Information Science in Engineering*, vol. 25, no. 2, 2025.
- [31] M. R. Bohm, R. B. Stone, T. W. Simpson, and E. D. Steva, “Introduction of a data schema to support a design repository,” *Computer-Aided Design*, vol. 40, no. 7, pp. 801–811, 2008.
- [32] M. R. Bohm and R. B. Stone, “Product design support: Exploring a design repository system,” in *ASME Int. Mech. Eng. Congress and Exposition*, vol. 47047, pp. 55–65, 2004.
- [33] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, and D. Panozzo, “ABC: A big CAD model dataset for geometric deep learning,” in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9601–9611.