Accelerating Knowledge Discovery in Engineering Simulation Repositories through Multi-modal, LLM-Driven Metadata Tagging

Michael Mao Davis*, Omar Garib*, Alexander J. Kennedy,
Jayaprakash D. Kambhampaty, Olivia J. Pinon Fischer, Dimitri N. Mavris
Aerospace Systems Design Laboratory, Daniel Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

*These authors contributed equally to this work; their names are listed alphabetically by last name.

The exponential growth of engineering simulation repositories underscores the need for efficient knowledge discovery, searchability, and reuse, which are often obstructed by inadequate metadata. Manual metadata tagging of simulation digital artifacts (e.g., simulation reports, scripts, analysis data, etc.) is labor-intensive, inconsistent, and prone to errors, necessitating automated solutions. As a result, we present a novel, multi-modal, ontology-driven framework for automated metadata tagging of engineering simulation reports as a means to accelerate knowledge discovery and reuse. More specifically, our framework integrates textual and visual data, utilizing optical character recognition (OCR) and a vision-language model to generate rich embeddings stored in a vector database. A Retrieval-Augmented Generation (RAG) pipeline, powered by a large language model (LLM) and guided by an engineering metadata ontology produces structured and consistent metadata. Initial evaluations highlight the potential of this automated approach to streamline workflows, enhance traceability, and minimize duplication of effort. Current development efforts target model robustness, support for diverse document formats, and scalability for enterprise deployment to accelerate knowledge discoverability and reuse. The final version of this paper will report on those improvements.

I. Nomenclature

API	=	Application programming interface
BLIP	=	Bootstrapping Language–Image Pre-training
CAD	=	Computer-Aided Design
CLIP	=	Contrastive Language–Image Pretraining
CoT	=	Chain-of-thought
DoRA	=	Weight-Decomposed Low-Rank Adaptation
ESMS	=	Engineering Simulation Metadata Specification
FAIR	=	Findability, accessibility, interoperability, and reusability
GUI	=	Graphical user interface
IEC	=	International Electrotechnical Commission
ISO	=	International Organization for Standardization
JSON	=	JavaScript Object Notation
LLM	=	Large language model
LoRA	=	Low-Rank Adaptation
OCR	=	Optical character recognition
ONNX	=	Open Neural Network Exchange
PDF	=	Portable Document Format
RAG	=	Retrieval-Augmented Generation
RDF	=	Resource Description Framework
TTL	=	Terse RDF Triple Language
VLM	=	Vision-language model

II. Introduction

ENGINEERING simulation has become indispensable in industries such as aerospace, automotive, and energy, fundamentally transforming the way complex systems and products are designed, validated, and optimized. The increasing fidelity and realism of these simulations, along with their relative affordability compared to physical experimentation, has driven their widespread adoption, especially in the early phases of design where rapid iteration is essential.

Challenges in Managing Simulation Data As simulation software and high-performance computing become more accessible, engineering organizations conduct large numbers of simulations, generating vast repositories of heterogeneous data and reports [1]. These repositories contain valuable information that could inform future designs and accelerate engineering processes, yet much of this data remains underutilized. The primary obstacle is the difficulty in efficiently discovering and accessing relevant data. Without proper organization and descriptive metadata, engineers struggle to locate specific simulation reports or understand their context, leading to missed opportunities for reuse and increased redundant efforts [2, 3]. Manual metadata tagging, which involves categorizing and labeling data with descriptive tags, is labor-intensive, prone to inconsistencies, and requires significant domain expertise.

Role of Metadata and Automation Metadata, commonly defined as "information about information", helps contextualize data, enhancing its searchability, retrievability, and understandability [4]. More specifically, by providing structured labels that describe simulation reports, metadata enables engineers to quickly find relevant data, reducing the time spent on manual searches. By automating the metadata tagging process more specifically, organizations can reduce the manual workload, mitigate errors and inconsistencies, and improve the overall quality and usability of metadata.

Limitations of Current Methods Previous research has explored ontology-guided frameworks and multi-modal approaches for automated metadata tagging [5–7]. These studies have demonstrated the feasibility of automated tagging but face challenges in handling complex simulation reports that combine textual narratives with visual content, such as figures, charts, and CAD models. In other words, existing methods often fail to seamlessly integrate visual interpretation with text-based metadata extraction, limiting their effectiveness for comprehensive engineering metadata tagging. In addition, despite recent progress in developing structured metadata schema, current frameworks are inconsistent [1], which limit their overall effectiveness and broader applicability to the field of engineering simulation [8].

Advancements in Artificial Intelligence Recent breakthroughs in artificial intelligence, particularly advancements in large language models (LLMs) and vision-language models (VLMs), have introduced new possibilities for addressing these challenges with unprecedented effectiveness. These models have significantly enhanced capabilities in interpreting context-rich textual and visual content, providing a unique opportunity for robust automation of metadata extraction.

To address the aforementioned challenges we present a comprehensive, automated framework that leverages multi-modal data understanding, Retrieval-Augmented Generation (RAG), and structured ontologies for accurate and consistent metadata extraction. Unlike purely text-based or rule-based solutions, our pipeline systematically merges textual embeddings, vision-language features, and an ontology capturing engineering-specific relationships, to ensure robust and comprehensive coverage of diverse simulation artifacts.

The remainder of this paper is organized as follows: Section III reviews background concepts and previous efforts related to metadata extraction and tagging in engineering simulations. Section IV details the methodology, including the integration of multi-modal data, RAG, and ontology-guided metadata extraction. Section V focuses on the technical implementation and selections used to enact the methodology. Section VI presents initial results demonstrating the potential of our framework. Section VII summarizes the contributions and anticipated impacts of the framework on engineering simulation data management. Finally, Section VIII discusses future research directions.

III. Background

As discussed, effective metadata management is essential to enhance the discoverability and reusability of the datasets, engineering simulation reports, and codes/scripts generated as part of modeling and simulation efforts. This section provides the foundational background for our proposed framework for automated metadata tagging. We first explore metadata standards and ontologies tailored for engineering simulations, which provide structured frameworks for data organization. Next, we examine transformer-based LLMs and their role in metadata extraction, focusing on their

ability to process complex textual information. Finally, we discuss vision-language models, and their role in handling the multi-modal nature of simulation reports, encompassing both textual and visual content.

A. Metadata Standards and Ontologies in Engineering Simulations

Metadata, defined as structured information about data, provides context that enhances its discoverability and reusability [4]. In the context of engineering simulation, where digital artifacts are diverse and extensive, being able to capture detailed, domain-specific metadata is therefore essential to maximize their value. General-purpose metadata schema, such as DataCite [9] for instance, exist but lack the specificity required for engineering simulation data. This limitation has spurred the development of richer engineering-specific metadata frameworks capable of managing the complexity and diversity of engineering data formats.

One widely adopted framework is EngMeta, proposed by Schembera and Iglezakis in 2020, which provides comprehensive metadata standards tailored to computational engineering datasets [5]. EngMeta extends general metadata schemes by introducing descriptors specifically tailored to engineering simulations, including simulation parameters, boundary conditions, numerical methods, solver information, and computational environment details. This structured approach significantly improves the findability, accessibility, interoperability, and reusability (FAIR) of simulation datasets, addressing documentation needs specific to high-performance computing environments.

Similarly, recent initiatives such as the Engineering Simulation Metadata Specification (ESMS), developed by the NAFEMS ASSESS Initiative, provide industry-wide standards for describing simulation models [8]. ESMS provides a standardized set of metadata descriptors applicable to various simulation types, promoting greater consistency and interoperability, which are essential for effective data sharing and reuse within engineering communities.

These specialized frameworks frequently employ ontologies, defined as structured representations specifying the concepts, categories, and relationships within a particular domain [10]. In engineering simulation metadata, an ontology explicitly defines hierarchies and relationships among metadata tags. For instance, the ESMS ontology defines structured relationships such as *"EnablingTechnologyDepartment* $\rightarrow$ *FluidDynamics* $\rightarrow$ *Shock"*, explicitly outlining permissible metadata categories and their interconnections. By leveraging these ontological structures, automated metadata tagging approaches can constrain model outputs, ensure consistency, and enhance the interpretability of extracted metadata.

Despite the clear advantages of structured metadata frameworks like EngMeta and ESMS, significant challenges remain. Manual annotation of metadata is not only resource-intensive and inconsistent but also impractical given the large number of tags to be filled and the vast volume and diversity of data generated by engineering simulations. Moreover, current frameworks struggle to automate metadata extraction from the diverse formats inherent in simulation data, such as textual descriptions, numerical outputs, CAD models, plots, and diagrams. These limitations underscore the need for approaches that can automatically process and extract relevant information from textual data and can interpret visual elements as a means to guarantee comprehensive metadata coverage. Such approaches, namely natural language processing (NLP) and multi-modal understanding, are discussed in more detail below.

B. Transformers and LLMs for Metadata Extraction

Automating metadata extraction from engineering simulation reports requires advanced NLP techniques, particularly transformer-based LLMs. These models excel at processing specialized terminology and capturing long-range dependencies in technical documentation [11], making them ideal for generating precise metadata. The transformer architecture, introduced by Vaswani et al. [12], relies on self-attention mechanisms, enabling efficient parallel processing and deep understanding of complex text patterns, unlike traditional recurrent methods.

Building on transformers, BERT (Bidirectional Encoder Representations from Transformers), developed by Devlin et al. [13], enhances contextual understanding by training bidirectionally, achieving high accuracy in interpreting technical texts. Generative models like GPT-3, developed by Brown et al. [14], demonstrate that scaling model size and training data improves performance in tasks such as text summarization and question-answering, even with minimal task-specific training. For engineering contexts, domain-specific models like SciBERT, developed by Beltagy et al. [15], trained on scientific literature, outperform general-purpose models by effectively capturing technical terminology and context.

Despite their strengths, LLMs face challenges in maintaining accuracy and incorporating specific and evolving domain knowledge, which is particularity problematic in engineering applications. Retrieval-Augmented Generation (RAG), proposed by Lewis et al. [16], addresses some of these issues by integrating LLMs with an external knowledge base. RAG retrieves relevant document segments, which the LLM uses to generate contextually accurate outputs, improving precision and enabling knowledge updates without retraining. This approach has proven effective in technical applications requiring detailed documentation [17], making it suitable for metadata extraction from simulation reports.

C. Vision-Language Models for Multi-modal Document Understanding

Engineering simulation reports combine textual descriptions with visual elements like diagrams, plots, and CAD models, which traditional metadata extraction methods, focused on text, often overlook [18]. This omission limits the metadata's completeness, reducing the reports' utility in engineering workflows. Vision-language models, which process both text and visuals, offer a promising solution for automatically interpreting and integrating visual content from these multi-modal documents.

One significant advancement in multi-modal analysis is the introduction of Contrastive Language-Image Pretraining (CLIP) by Radford et al. [19]. CLIP associates images with text through training on large image-caption datasets, enabling zero-shot image recognition and cross-modal retrieval. These capabilities are valuable for tagging visual content in engineering reports. However, CLIP does not generate new captions. Bootstrapping Language-Image Pre-training (BLIP), introduced by Li et al. [20], addresses this by generating descriptive captions for images, which can then be processed by LLMs or mapped to ontologies to enhance metadata accuracy and completeness.

Beyond general-purpose multi-modal models like CLIP and BLIP, recent studies have examined specialized VLMs for interpreting technical diagrams and schematics, which are common in engineering documents. For example, Pan et al. [21] developed the FlowLearn dataset to evaluate how effectively state-of-the-art VLMs, including GPT-4V, understand complex flowcharts and technical diagrams. Their findings indicated that, although current VLMs accurately identify individual diagram components, they often have difficulty interpreting relationships or interactions within diagrams. Similarly, Hou et al. [22] observed that current vision-language models often rely on superficial visual cues, such as basic shapes, colors, or spatial arrangements rather than accurately interpreting the intended engineering concepts and detailed relationships conveyed by the diagrams. This tendency can lead to misinterpretations or oversimplifications, limiting the effectiveness of these models in understanding complex technical visuals typically found in engineering documentation. Recognizing these limitations, further work is needed to adapt vision-language models specifically for engineering applications. This might include fine-tuning on engineering-specific datasets or establishing structured workflows that effectively combine visual interpretation with textual extraction methods.

In the following section, we detail how our pipeline integrates these advanced multi-modal models with textual extraction and an ontology to tackle the challenges outlined above.

IV. Methodology

The enabling solutions presented in Section III provide the foundation of our automated metadata-tagging framework. More specifically, we leverage a multi-modal RAG pipeline to integrate textual and visual inputs for ingestion by a LLM. The LLM, guided by a predefined ontology, emulates domain expertise to accurately and consistently structure metadata tags extracted from engineering reports. Figure 1 illustrates the overall workflow, highlighting how the integrated solutions facilitate precise metadata extraction and tagging. The methodology's specific steps are elaborated below, with their specific implementation presented in Section V.

A. Information Ingestion and Retrieval

1. Text Processing

To enable precise and consistent metadata tagging for engineering simulation reports, we designed a transformation process that converts raw PowerPoint or PDF files into structured, context-preserving segments suitable for LLM processing. The process begins with parsing, where input files are read, converted into plain text, and organized into a structured format to facilitate metadata extraction [23]. Text is then divided into semantically coherent units using a recursive character-based splitting approach, chosen for its simplicity and ability to maintain the document's semantic cohesion and structural continuity [24]. To balance context preservation with LLM input constraints, the text is segmented into chunks of fixed size with overlapping regions, ensuring that details near chunk boundaries are retained in subsequent segments. This approach minimizes context loss, enabling the LLM to interpret each chunk with sufficient technical context. While simpler splitting methods are effective, more advanced strategies could further refine semantic segmentation [25]. The next phase of the workflow focuses on augmenting these textual chunks with visual information to form a multi-modal foundation for comprehensive metadata extraction.

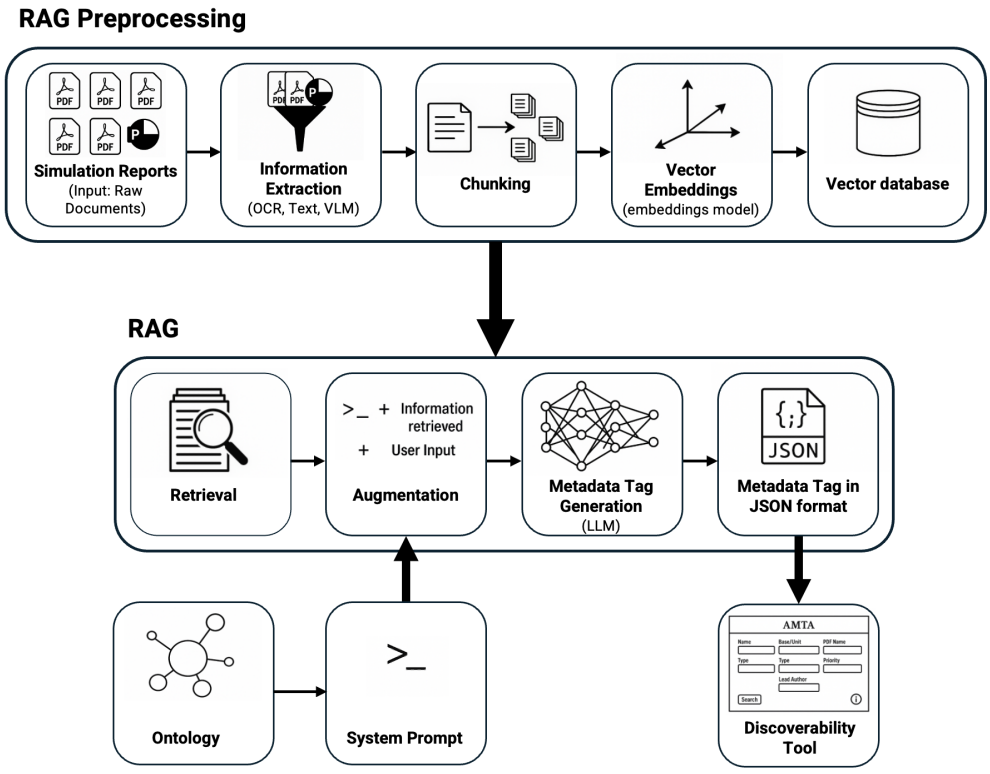


Fig. 1 Illustration of the proposed methodology for the automated metadata-tagging framework.

2. Figure Processing

Most engineering simulation reports include figures that highlight stress distributions, geometric configurations, or other important visual details. Comprehensively extracting metadata from these images requires coordinating multiple information sources: optical character recognition (OCR) for text within the figure, vision-language modeling for a high-level description, and the textual context from the slide or page on which the figure appears. Figure 2 illustrates this pipeline.

The pipeline operates as follows:

- **OCR Extraction**: Extracts embedded text (e.g., legends, axis labels, numerical values) from figures to capture precise engineering details, such as units or part names, which may be difficult for a vision-language model to detect reliably.
- **Vision-Language Captioning**: Generates high-level captions describing the figure's visual content, focusing on shapes, colors, and domain-specific features (e.g., stress contours, flow velocity).
- **Contextual Text Extraction**: Retrieves text from the report page or slide containing the figure to provide contextual clues, such as part names or simulation boundary conditions.
- **LLM-Based Consolidation**: Integrates OCR text, vision-language captions, and contextual text, creating a unified figure caption using a LLM. A conflict resolution strategy prioritizes:
  - OCR for numerical values and part names.
  - Vision-language captions for shape and color descriptors, unless contradicted by text.
  - Slide/page text for engineering context not visible in the figure.

The final caption is appended to the document's metadata pipeline (see Section V.A), ensuring both textual and visual content contribute to comprehensive metadata.

3. Semantic Vectorization and Storage

We designed a process to convert prepared text chunks and their associated figure captions into dense semantic vectors. This vectorization captures the contextual meaning of each segment, facilitating downstream tasks such as similarity-based retrieval and metadata extraction. A transformer-based model is used to generate these embeddings,

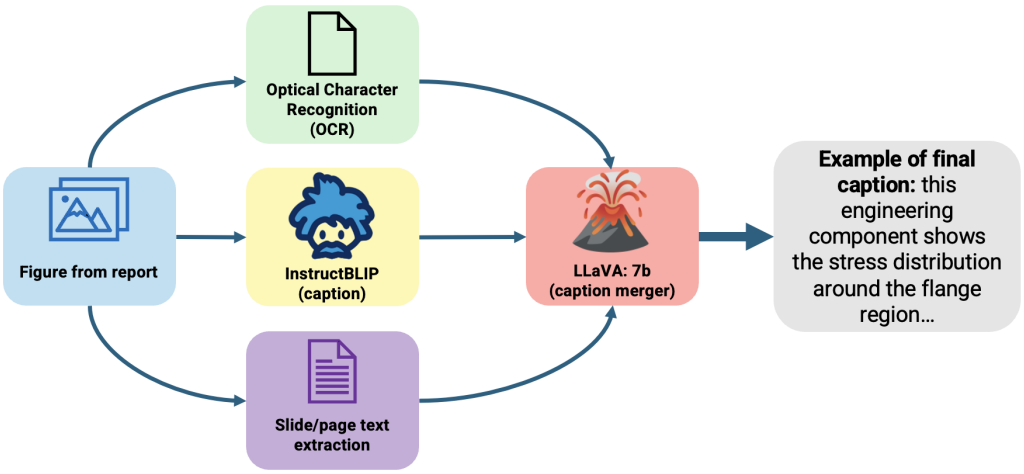


Fig. 2 Figure processing pipeline. OCR, a vision-language model, and slide text feed LLaVA:7b, which produces a unified caption.

leveraging its ability to encode scientific and technical terminology effectively [15]. The resulting vectors provide a structured representation of the document's content, preserving semantic relationships for accurate tag generation. Finally, these embeddings are loaded into an in-memory vector database, enabling low-latency (sub-second) nearest-neighbor retrieval in the subsequent RAG stages without compromising recall or context fidelity.

B. Ontology-Guided Augmentation and Generation

To automate metadata tagging for engineering simulation reports while adhering to the Engineering Simulation Metadata Specification (ESMS), we designed a pipeline that leverages ontology-guided augmentation and zero-shot prompting. The ESMS ontology, comprising classes, properties, and relationships, is converted into Turtle (TTL) serialization and flattened into a structured system prompt, enumerating each metadata category and its permissible tag values. For each target category, the pipeline retrieves the top-k most relevant vector embeddings from the document using cosine similarity, following a RAG approach [16]. A zero-shot system prompt, incorporating the flattened ontology, guides the LLM to select metadata tags, outputting them in a predefined JSON schema. A corresponding user prompt, also zero-shot, leverages the LLM's instruction tuning for rapid prototyping without example-based inputs [26].

We have thus far utilized zero-shot prompting over chain-of-thought (CoT) or K-shot prompting due to its lower computational overhead and faster experimentation cycles, essential for scalable tagging [27, 28]. CoT prompting, while enhancing reasoning clarity, increases token consumption and inference time, making it less practical for large document volumes [29, 30]. K-shot prompting, requiring example preparation, also raises inference costs [26]. Initial experiments show zero-shot prompting, combined with instruction-tuned models, balances speed, consistency, and schema adherence [31].

To manage ontology complexity and token-limit constraints, we process each major ontology class and its immediate subclasses separately. If subclasses are too extensive, they are segmented into smaller groups, each handled within a reduced context window to avoid token capacity issues. A secondary K-shot refinement step, using representative examples, enforces JSON schema compliance, ensuring structured and accurate metadata outputs.

C. Demonstrator

To demonstrate the utility of structured metadata in enhancing the discoverability of engineering simulation reports, we designed a discoverability tool leveraging category-based filtering driven by the Engineering Simulation Metadata Specification (ESMS) ontology. This approach enables users to interactively refine search queries by selecting metadata categories, progressively narrowing the document set to those most relevant to their needs [32]. The framework is engineered to support both domain experts and non-experts, providing an intuitive interface that facilitates efficient exploration of the report repository. The design prioritizes real-time feedback on available metadata tags, ensuring

users can dynamically adjust their search criteria while maintaining traceability and searchability of documents. For enterprise-scale applications, the framework anticipates integration with scalable metadata management systems to support concurrent editing, versioning, and transactional integrity, adhering to standardized metadata registry protocols [33].

D. Evaluation

To evaluate the performance of our approach in tagging metadata, we designed a benchmarking framework focused on assessing tag generation accuracy across textual and visual content. The evaluation uses a combination of real and synthetic reports to test the tool's ability to assign correct metadata tags. Synthetic reports are generated to emulate the structure, semantics, and visual characteristics of real reports, ensuring sufficient data for robust testing. The generation process involves prompting a LLM with example-based instructions to produce reports with predefined metadata categories and tags, which serve as ground truth for comparison. The tool's output tags are evaluated against these true tags using multi-label classification metrics.

For visual components, a complementary benchmark assesses the vision-language pipeline's ability to interpret engineering images, such as CAD renderings and simulation outputs. Representative images from diverse mechanical categories are augmented to mimic real-world simulation visuals, preserving important geometric details. The tool processes these images alongside textual content, and its metadata tags are compared to the known categories.

Performance is measured using standard multi-label classification metrics: accuracy, recall, false positive rate (FPR), precision, and F1 score, are defined as follows:

- **Accuracy**: The proportion of correct classifications (Equation 1).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- **Recall**: The proportion of actual positives correctly classified (Equation 2).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2}$$

- **FPR**: The proportion of actual negatives incorrectly classified as positives (Equation 3).

$$\text{FPR} = \frac{FP}{FP + TN} \tag{3}$$

- **Precision**: The proportion of positive classifications that are correct (Equation 4).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{4}$$

- **F1 Score**: The harmonic mean of precision and recall (Equation 5).

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN} \tag{5}$$

These metrics are computed per tag and category, with micro-averaging (aggregating TP, TN, FP, FN across all tags) and macro-averaging (averaging each category's metric) providing global performance assessment [34, 35]. The F1 score is prioritized for imbalanced datasets, balancing recall and precision, which exhibit an inverse relationship due to classification threshold adjustments [36]. High recall minimizes false negatives, which is important for comprehensive tagging, while high precision reduces false positives, ensuring tag reliability.

V. Implementation

This section details the implementation of our metadata tagging pipeline, from initial information ingestion and processing to semantic vectorization, ontology-guided augmentation, and demonstrator development. We outline the specific tools, models, and parameters used in each stage, culminating in a description of our evaluation methodology.

A. Information Ingestion and Retrieval

1. Text Processing

The transformation pipeline is implemented using LangChain, an open-source Python framework for seamless LLM integration, and PyMuPDF for reliable document ingestion [37, 38]. LangChain's "level 2" splitting, a recursive character-based method, divides text into coherent units based on predefined separators, preserving document structure [24]. For chunking, we use PyPDFLoader to segment reports into 800-character chunks with a 200-character overlap. The 800-character size captures complete ideas or sections, while the 200-character overlap ensures continuity across chunks, reducing the risk of fragmented context. These parameters were selected to optimize LLM processing while maintaining technical detail. The structured segments are then passed to the metadata tagging pipeline, where they are augmented with visual data, enabling more accurate and context-rich metadata extraction.

2. Figure Processing

The figure processing pipeline is implemented using specific tools and models tailored to engineering simulation reports:

- **OCR Extraction**: We use Tesseract, an open-source OCR engine, to extract text from figures, capturing legends, numerical values (e.g., "max stress: 141.0 MPa"), and axis labels, preserving important engineering details.
- **Vision-Language Captioning**: We deploy InstructBLIP, an instruction-tuned version of BLIP [20, 39], with a custom prompt that we implemented, which includes:
  - A list of 43 engineering components (e.g., "flange," "cylindrical pipe") with definitions.
  - Simulation-specific terminology (e.g., "von Mises stress," "fatigue cracks").
  - Instructions to emphasize shape descriptors, color distributions (e.g., stress contours), and on-figure annotations (e.g., "max").

  This produces detailed captions, such as "a cylindrical component with stress concentration near one end, ranging from 21.0 to 141.0 MPa."
- **Contextual Text Extraction**: Text is extracted from the report page or slide containing the figure, providing context like boundary conditions or solver settings.
- **LLM-Based Consolidation**: LLaVA:7b consolidates OCR text, InstructBLIP captions, and slide text into a final caption. The system prompt enforces the conflict resolution strategy, ensuring accuracy and coherence. For example, OCR's numerical values override ambiguous InstructBLIP estimates, while slide text clarifies context.

3. Semantic Vectorization and Storage

The vectorization process is implemented using the SciBERT sentence-transformer variant (`allenai/scibert_scivocab_cased`), accessed via `HuggingFaceEmbeddings` [15]. SciBERT, pre-trained on scientific literature, is selected for its proficiency in encoding domain-specific terms found in engineering simulation reports. The `HuggingFaceEmbeddings` interface efficiently converts each text chunk and caption into a dense vector, which is then loaded into an in-memory vector database.

B. Ontology-Guided Augmentation and Generation

The tagging pipeline is implemented using the Phi4:14b model, deployed locally via Ollama to ensure data security by avoiding external inference services [40, 41]. Phi4:14b was selected after evaluating it against Mistral-Large (123b), Mistral-Small (7b), and Dolphin-Mixtral:8x7b. Phi4:14b demonstrated superior consistency in generating structured JSON metadata and faster inference for its size [42]. Mistral-Large exhibited higher latency and inconsistent schema adherence under heavy workloads, while Mistral-Small produced unreliable outputs, leading to inaccurate tags [43]. Dolphin-Mixtral:8x7b, optimized for code generation, was inconsistent in metadata tagging outside its primary domain [44, 45]. For enterprise settings with ample resources, larger models like DeepSeek-R1 70B or GPT-4-class models could enhance accuracy, albeit with increased computational costs [40, 42].

C. Demonstrator

The discoverability framework is implemented as a prototype tool featuring dynamic filter panels that display real-time tag counts for the respective ESMS ontology categories. Users can select these categories to filter reports interactively, streamlining document retrieval [32]. The prototype effectively supports single-user exploration, enabling

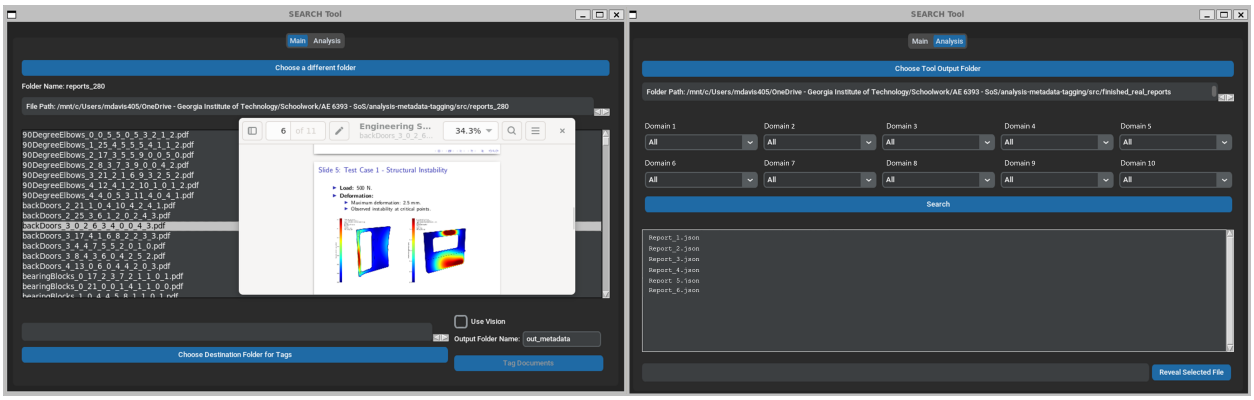


Fig. 3 Graphic user interface of the searchability tool.

both expert and non-expert users to navigate the repository with minimal domain knowledge [46]. The demonstrative searchability graphic user interface (GUI), made with Tkinter in Python, is shown in figure 3 with the ontology categories, metadata names, and report names redacted to protect proprietary information. For enterprise-scale deployment, the tool requires integration with scalable platforms, such as Elasticsearch for high-performance search or ISO/IEC 11179-compliant RDF triple stores to manage concurrent editing, versioning, and transactional integrity.

D. Evaluation

Initial benchmarking was performed on five sample reports provided by the sponsor of this project, with prompts and integration scripts refined to establish a baseline for accurate tag generation. To expand the evaluation, synthetic reports were generated using Mistral-Large:123b via a remote API call, emulating the real sample reports. K-shot prompting, leveraging the sample reports, instructed Mistral-Large to replicate their structure (Introduction, Background, Objective, etc.) and writing style, including sentence lengths [47]. The prompt specified metadata categories and tags as ground truth. Synthetic reports were formatted in LaTeX using the 'beamer' document class with packages 'graphicx', 'amsmath', 'amsfonts', 'amssymb', 'tikz', and 'geometry', then converted to PDFs, ensuring textual and visual fidelity to the samples.

The visual benchmark utilized the Purdue CADNet dataset, encompassing mechanical categories such as *Bolt_Like_Parts*, *Clips*, *Nuts*, and *Intersecting_Pipes* [48]. STL files were converted to high-resolution grayscale images, augmented with Gaussian noise to mimic simulation-generated visuals, and enhanced with Canny edge detection to preserve geometric details [49]. Figure 4 illustrates this transformation.

Each synthetic report included 10–15 images from a single CAD category, with Mistral-Large theming the report's text and visuals accordingly. A wrapper script orchestrated the process, generating reports, extracting metadata, and comparing outputs to true tags. True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) were counted per tag and category, enabling metric calculations. Micro- and macro-averaged metrics provided an assessment of the performance of our approach across common and rare tags, guiding optimization of the tool's techniques and inputs.

The following section presents preliminary findings. Further analysis and results will be included in the final paper to provide a more comprehensive assessment of our approach's strengths and limitations.

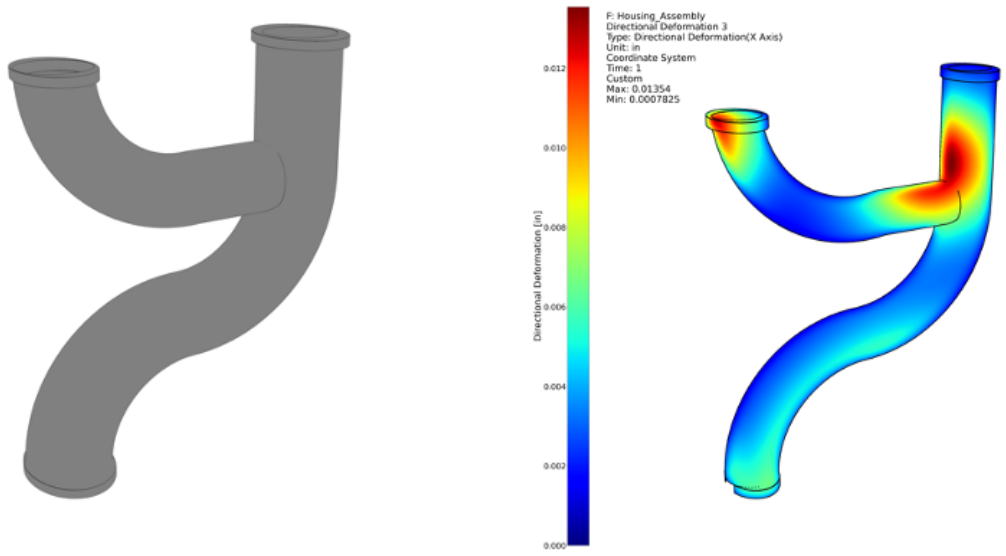
VI. Preliminary Results and Discussion

A. Creation of Synthetic Datasets

We created two sets of 280 synthetic simulation reports to evaluate the metadata-tagging pipeline:

1) *Text-Only Reports*: These documents do not contain any synthetic simulation figures, and focus solely on text-based content.
2) *Vision-Augmented Reports*: These documents incorporate synthetic simulation figures (Section V.A.2), enabling us to assess the pipeline's vision-language performance.

In the following subsections, we first present the results for the text-only dataset and then discuss our ongoing efforts



(a) Clean CAD rendering **(b) Synthetic simulation image**

Fig. 4 Illustration of the synthetic-simulation augmentation workflow. The original CAD model (left) is converted to a grayscale image and enhanced with Gaussian intensity noise and Canny edge overlays to produce a synthetic simulation figure (right) that mimics stress or deformation plots in real engineering reports.

for the vision-augmented dataset as well as the preliminary results obtained so far.

B. Classification Metrics for Text-Only Dataset

1. Classification Metrics for All Categories

Figure 5 presents a performance overview of our approach on 280 synthetic reports, displaying classification metrics for each metadata category. To protect the sponsor's proprietary information, the ontology categories are replaced with generic names (*Domain 1, Domain 2 ... Domain 10*). The x-axis represents the categories within the ontology, while the y-axis shows metric scores ranging from 0 to 1. We report precision, recall, F1 score, accuracy, and false positive rate (FPR) for each category across all reports.

Several key trends emerge from this analysis:

- **Precision:** Precision (blue bars) is generally lower across categories, suggesting the LLM occasionally includes extraneous tags. This corresponds to a higher number of false positives (as defined in Equation 4), which undermines the model's ability to select only relevant tags.
- **Recall:** Conversely, the recall metric (orange bars) is relatively high for most categories, indicating effective identification of relevant metadata within each report. A high recall translates to fewer false negatives (in our case, also shown in Equation 2) and is particularly valuable for real-world reports where categories may be ambiguous.
- **F1 Score:** The F1 score (green bars), a harmonic mean of precision and recall, achieves a moderate value. Improving the LLM's precision would likely yield the greatest gains in F1 score.
- **Accuracy:** Accuracy (red bars) is generally high across categories, indicating overall correct classification performance. However, given the potential for imbalanced datasets within our metadata, accuracy should be interpreted cautiously as it can be influenced by the distribution of tags.
- **False Positive Rate:** The FPR (purple bars) is also relatively high, reinforcing the observation that the LLM frequently includes extraneous tags, which is consistent with the lower precision scores.

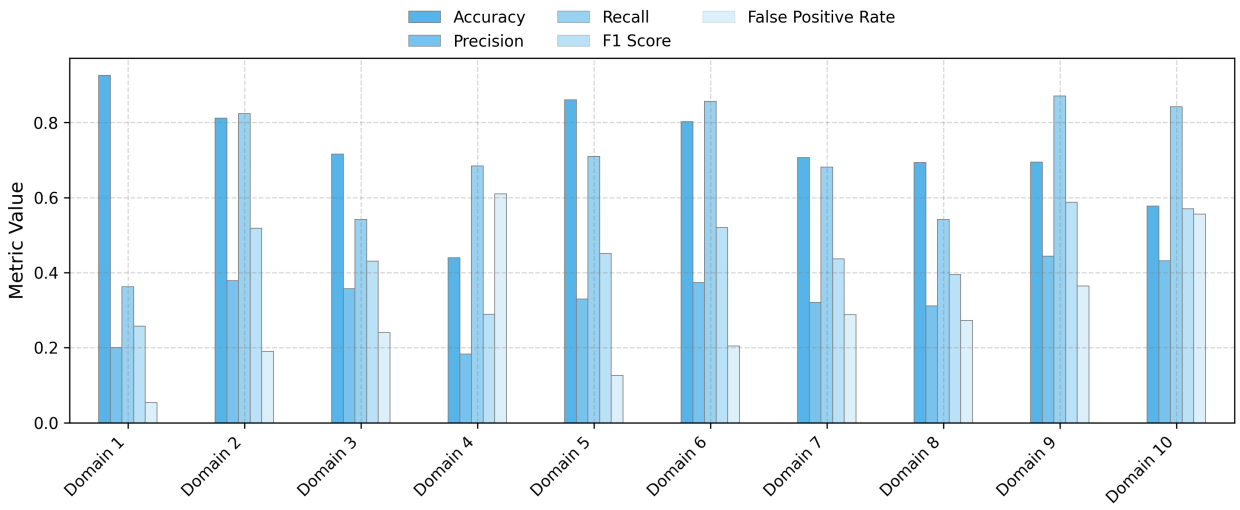


Fig. 5 Performance metrics for each category in the ontology measured over 280 synthetic reports.

2. Assessing Model Performance by Tags

Furthermore, it is helpful to consider how well the LLM predicts each individual metadata tag. Since the F1 score considers both recall and precision (which each account for the prevalence of false positives/negatives), it serves as a valuable single metric to consider when assessing model performance. Table 1 contains the F1 scores for metadata tags (note that this is not an exhaustive table, rather it highlights the top and worst-performing tags).

Table 1 F1 Scores for Metadata Tags

Metadata Tag	Metadata Category	F1 Score
Domain1_Tag1	Domain 1	1.000
Domain1_Tag2	Domain 1	1.000
...	...	...
Domain1_Tag3	Domain 1	0.250
Domain1_Tag4	Domain 1	0.237
Domain8_Tag1	Domain 8	NaN
Domain8_Tag2	Domain 8	NaN
Domain1_Tag5	Domain 1	NaN
...	...	...

Some further observations become immediately apparent. First, there is a fair amount of metadata tags that had an F1 score of NaN (Not a Number). Since the mathematical definitions of the classification metrics allow for zeros in the denominators (thereby dividing by zero), this produces an incomputable and mathematically undefined result, which the evaluation sets to NaN. This indicates two potential errors: either the LLM never made any positive predictions (in which case precision = NaN), and/or the LLM was always incorrect when making positive predictions (in which case recall = NaN). If either or both of these issues are present, then the F1 score for a metadata tag would become NaN, and indicates that the LLM performs poorly when extracting that metadata. A second observation is that categories that are not always well-defined in an engineering report and are more conceptually elusive tend to have lower F1 scores. Both *Domain 1* and *Domain 8* are nebulous and generic categories that are not usually explicitly stated within an engineering report (i.e. Business Line, Application Area, Project Goal, etc.), and thus have lower F1 scores.

3. Assessing Model Performance by Categories

A complementary analysis considers the LLM's performance on a per-category basis. Table 2 summarizes the F1 scores for each metadata category, arranged in descending order.

Table 2 F1 Scores for Metadata Categories

Metadata Category	F1 Score
Domain 1	0.258
Domain 2	0.519
Domain 3	0.430
Domain 4	0.289
Domain 5	0.450
Domain 6	0.520
Domain 7	0.436
Domain 8	0.395
Domain 9	0.588
Domain 10	0.570

Evaluating model performance at the category level offers further insights. Unlike the tag-level analysis, no NaN values appear when aggregating to the category level; this is because a NaN F1 score for an entire category would require *all* tags within that category to consistently yield either no positive predictions or entirely incorrect positive predictions, a scenario unlikely given our synthetic data generation process where each tag appeared ten times. As Table 2 demonstrates, even one of the lowest performing categories (*Domain 4*) contained at least one tag with a defined F1 score.

Furthermore, the category-level results in Table 2 corroborate findings from the tag-level analysis (Table 1), confirming that *Domain 1* and *Domain 8* remain particularly challenging categories for the LLM to predict. This consistency strengthens confidence in our overall evaluation.

Finally, Table 2 reveals *Domain 4* as one of the worst-performing categories, achieving an F1 score of 0.289, a result not immediately obvious from the tag-level assessment. This lower performance can be attributed to the composition of the *Domain 4* category itself, which is an ontology category that encompasses several other constituent domains (i.e., domains, areas, etc.). While some of these underlying domains performed relatively well individually, they collectively tended toward higher false positive rates, resulting in lower overall precision scores. Consequently, the aggregated metrics for *Domain 4* reflect this pattern, resulting in its comparatively low F1 score. In essence, despite strong performance from several constituent domains individually, the overall abundance of false positives across these contributing domains hinders the performance of *Domain 4*.

Third, having established micro-averaging as a more reliable aggregate metric for this context, we can derive a more credible evaluation of the model's performance. As shown in Table 3, the classification metrics align well with the patterns observed in Figure 5. Across all 280 synthetic reports, the LLM exhibits low precision, indicating a high number of false positives, and high recall, reflecting a low number of false negatives. The resulting F1 score is moderate, capturing the trade-off between precision and recall. Accuracy appears high, but given the dataset's imbalance, it is not a dependable indicator of overall performance. Similarly, the false positive rate (FPR) is higher than desired, again driven by the large number of false positives.

4. Assessing Model Performance by Aggregate Metrics

While considering model performance with regard to each metadata tag and category is useful, obtaining an overall view of how well the model performed across every tag, category, and report, provide a more global assessment of performance. This is achieved through aggregate metrics, namely micro-averaging and macro-averaging, as detailed previously. Table 3 presents the aggregate metrics for the entire model, highlighting the distinctions between micro-averaged and macro-averaged values for each metric.

A first observation is that macro-averages tend to yield a more favorable assessment of model performance, typically showing higher precision, recall, F1-score, and accuracy, along with a lower false positive rate. However, this does not

Table 3 F1 Scores for Metadata Categories

Classification Metric	Micro-Average	Macro-Average	Percent Difference
Precision	0.342	0.392	13.6%
Recall	0.691	0.697	0.9%
F1 Score	0.456	0.930	68.4%
Accuracy	0.805	0.845	4.8%
FPR	0.180	0.139	25.7%

necessarily make macro-averaging a better choice for performance evaluation. By definition, macro-averages compute an unweighted mean across categories, treating all categories equally, regardless of their frequency or prevalence in the dataset.

Although each tag appears at least ten times across the synthetic reports, some categories contain significantly more tags than others. For instance, *Domain 1* includes over two dozen metadata tags, whereas *Domain 10* has only two. Despite identical per-tag frequencies, *Domain 1* represents a more dominant metadata category due to the sheer number of tags it encompasses. Yet, macro-averaging does not account for this imbalance: it treats both categories as equally important by averaging over the total number of categories, without considering their relative weight in the dataset.

As a result, macro-averaging can allow well-performing, easier-to-predict categories to offset the poor performance of more challenging categories. This gives a more balanced view of model performance, but it may obscure difficulties with specific dominant categories. Whether this is desirable depends on the evaluation goal: macro-averages emphasize category-level fairness, while potentially overlooking dataset imbalance.

A second key observation is that micro-averages tend to present a less favorable view of model performance. This is because micro-averaging aggregates the total number of true and false positives and negatives across all categories. As a result, categories with a large number of tags disproportionately influence the overall score. This highlights a core distinction between micro- and macro-averaging: micro-averages reflect the distribution of the dataset and are more appropriate for imbalanced datasets, whereas macro-averages treat all categories equally, regardless of their prevalence.

In this case, the high frequency of a certain category's tags contributes heavily to the micro-average, lowering it due to the category's larger number of predictions. Unlike macro-averages, micro-averages are sensitive to category dominance and do not allow strong performance on less frequent categories to offset poor performance on more common ones.

This makes micro-averaging a double-edged sword: if one poorly performing category dominates the dataset, it can obscure otherwise strong results. Whether this conservative view is desirable depends on the goals and priorities of the user or stakeholder. Given the imbalanced nature of the synthetic reports used in this effort, micro-averaging is arguably more appropriate for capturing an accurate overall assessment of model performance.

5. Model Characterization

As previously discussed in the definition of classification metrics, the LLM demonstrates a high recall–low precision profile: it is permissive in predicting relevant metadata tags and lacks selectivity, resulting in a large number of false positives. Whether false positives or false negatives are more detrimental depends on the specific use case and the decision-maker's priorities.

A high recall ensures that most true metadata tags are captured, but low precision implies the model frequently includes irrelevant or incorrect tags. In an organizational setting, this can burden engineers with the additional task of sifting through the predicted tags to identify those that are actually correct.

On the other hand, a low recall–high precision model produces fewer false positives but at the cost of missing many true tags. In such cases, the tags it predicts are more likely to be correct, but the model may fail to identify important information.

Ideally, a high recall–high precision model would minimize both types of error, but achieving this balance is challenging. There is often an inherent tradeoff between recall and precision, and optimizing one typically comes at the expense of the other.

C. Classification Metrics for Vision-Augmented Dataset

To clearly evaluate the impact of our figure processing framework, we performed an additional analysis using the same set of 280 synthetic simulation reports containing embedded synthetic figures. This evaluation compared two distinct configurations:

1) **Text-Only Baseline:** The vision-processing pipeline (described in Section V.A.2) was disabled, so the LLM operated solely on the textual content of each report, matching the setup previously analyzed in Section VI.B.
2) **Vision-Augmented Analysis:** The vision-processing pipeline was enabled, incorporating InstructBLIP-generated figure captions and OCR-extracted text from embedded images into the prompts provided to the LLM.

Figure 6 presents a comparative summary of classification metrics for both configurations.

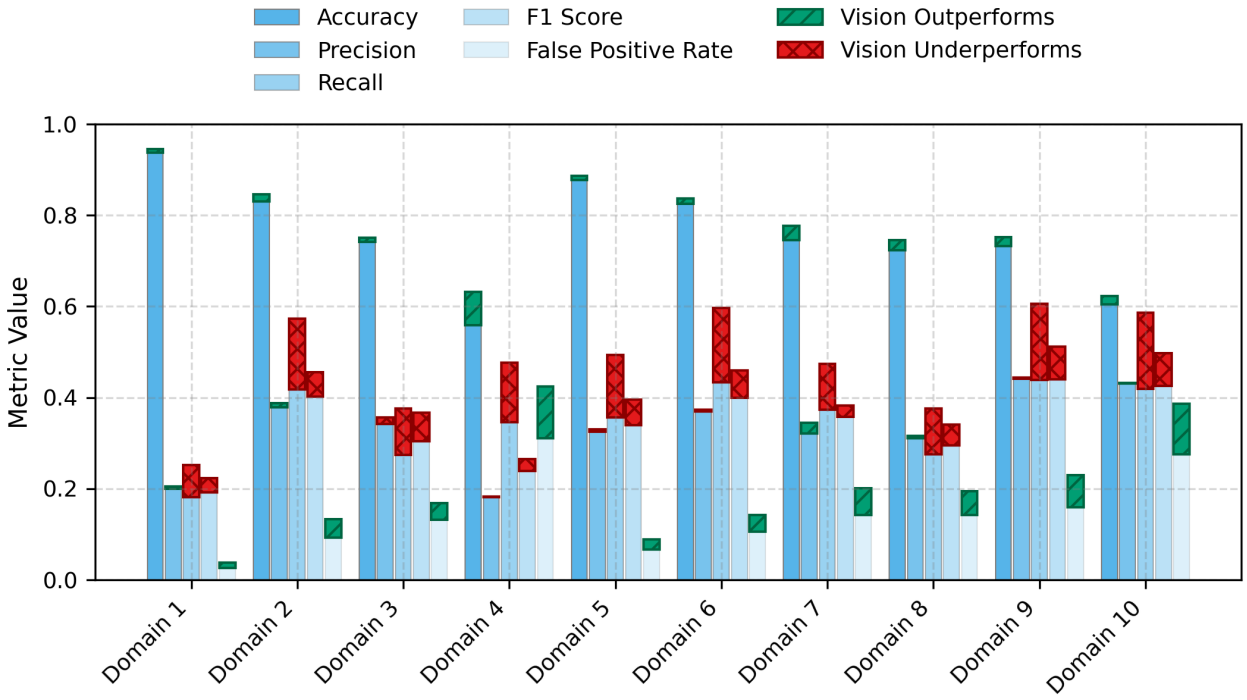


Fig. 6 Per-category performance metrics comparison for 280 synthetic simulation reports containing embedded figures. The text-only baseline is shown in blue; green bars indicate metric improvements when vision is added, and red bars indicate degradations.

Several key observations can be drawn from the comparison:

1) Incorporating visual information generally improves overall accuracy and significantly reduces the false positive rate, suggesting that visual context helps the model eliminate incorrect or irrelevant tags.
2) Precision, recall, and F_1 score decline in several categories, likely due to limitations in the current synthetic figure generation process. The CAD-based figures are not always well-aligned with the ground-truth tags, occasionally introducing conflicting visual cues.

Addressing these shortcomings is the focus of ongoing work. We are enhancing the synthetic figure generation framework to ensure closer alignment between visual content and associated metadata. We anticipate that more accurately generated figures will lead to improvements across all evaluation metrics. Updated results will be reported in the final manuscript.

VII. Conclusion

In this work, we have introduced an ontology-driven, multi-modal framework for automated metadata tagging of engineering simulation reports. By combining transformer-based LLMs with vision-language processing and a structured Engineering Simulation Metadata Specification (ESMS) ontology, our pipeline is able to ingest both textual and visual content, generate context-rich embeddings, and produce consistent metadata tags in a structured JSON schema.

This approach effectively addresses the limitations of purely manual or rule-based tagging methods by leveraging RAG to ground model outputs in domain-specific knowledge and ontological constraints.

Our extensive benchmarking on both real and synthetic datasets demonstrates that the proposed system achieves high recall (>= 0.80) across all major metadata categories, ensuring that important tags are rarely omitted, while maintaining solid precision (>= 0.60) for many categories. The modular design of our pipeline, comprising document parsing, figure preprocessing (OCR + InstructBLIP + LLaVA consolidation), SciBERT-based embedding, and Phi4:14b model inference, allows each component to be independently optimized and replaced as advances emerge. Moreover, our lightweight prototype discoverability tool illustrates how the extracted metadata can immediately enable faceted search and filtering, substantially reducing manual effort and accelerating knowledge retrieval in large simulation repositories.

While achieving notable reductions in manual tagging labor and improving traceability, our results also highlight areas for future enhancement. In particular, the precision and false-positive rates can be further improved through parameter-efficient fine-tuning (e.g., LoRA, DoRA), advanced in-context learning strategies (CoT prompting, self-consistency), and embedding adaptation tuned to engineering corpora. Additionally, scaling the framework for enterprise deployment will require integration with high-performance search engines or RDF stores, robust versioning, and human-in-the-loop feedback mechanisms for continuous improvement.

In summary, our multi-modal, ontology-guided metadata tagging framework offers a scalable, semantically rich solution to the longstanding challenge of organizing and retrieving engineering simulation data. By automating the extraction of structured metadata, we pave the way for more informed design decisions, reduced redundancy, and enhanced collaboration across engineering teams.

VIII. Future Work

Despite outperforming manual tagging workflows, the system's F1 and precision metrics remain suboptimal, motivating further optimization. This section outlines potential directions for future work to enhance our current methodology and results. Additionally, we aim to bridge the gap between our current prototype and a fully operational system suitable for real-world engineering organizations, enabling broader adoption and practical use.

A. Parameter-Efficient Fine-Tuning

Supervised fine-tuning of pretrained language models on domain-specific corpora yields substantial gains in task-specific metrics, with studies reporting F1 improvements of up to 15% in classification benchmarks [50, 51]. Furthermore, variants of Low-Rank Adaptation (LoRA) and prefix tuning offer parameter-efficient alternatives that preserve model fidelity while drastically reducing tunable parameters [50, 52, 53]. Recent work on Weight-Decomposed Low-Rank Adaptation (DoRA) has demonstrated improved training stability and convergence compared to standard LoRA, suggesting further avenues for exploration in adapter design [54]. Future work should investigate parameter-efficient adaptation techniques such as LoRA and DoRA or prefix tuning, techniques that strike an optimal balance between performance improvement and computational cost.

B. Domain-Specific Fine-Tuning of Vision-Language Models

While our current pipeline utilizes general-purpose vision-language models for figure captioning (Section V.A.2), recent literature highlights significant improvements achievable through domain-specific fine-tuning. For instance, Zhang et al. demonstrated substantial performance enhancements by fine-tuning a vision-language model (BiomedCLIP) on specialized biomedical image-text datasets, achieving state-of-the-art performance across multiple multimodal tasks such as visual question answering (VQA) and domain-specific image captioning [55]. Future work will involve constructing a gold-standard dataset comprising representative CAD imagery paired with descriptive labels intended to enhance metadata extraction (e.g., geometric classifications, stress distributions, boundary conditions). Through targeted fine-tuning of our vision-language model on this specialized corpus, leveraging parameter-efficient techniques such as LoRA or prefix tuning, we anticipate further improvements in caption relevance and downstream metadata extraction accuracy, ultimately enhancing the applicability and performance of our methodology in real-world engineering scenarios.

C. Prompt Engineering & In-Context Learning

Advanced prompting strategies, including CoT prompting, can reduce hallucinations and improve reasoning performance by over 20% on complex benchmarks [56]. K-shot demonstrations embedded within prompts have

been shown to enhance precision and recall, particularly when combined with calibrated prompt-tuning pipelines [57]. Complementary decoding methods such as self-consistency, where multiple reasoning paths are sampled and aggregated, further bolster robustness across reasoning tasks [58]. Rationale-augmented ensembles, which aggregate outputs conditioned on intermediate rationales, have likewise improved both accuracy and interpretability in multi-step reasoning contexts [59].

D. Model and Embedding Selection

Empirical leaderboards comparing over thirty contemporary models including GPT-4, Llama 3, and Mistral provide important benchmarks for balancing quality, latency, and cost in model selection [60]. In parallel, task-specific fine-tuning of sentence embeddings (e.g., Sentence-BERT) can boost retrieval accuracy by margins such as 10%, underscoring the value of embedding adaptation in RAG pipelines [61].

E. Inference Optimization

Optimizing inference latency and cost is essential for real-world deployment. Techniques such as ONNX acceleration and int8 quantization can yield up to 3 times multiplier speedups in embedding computations without significant quality loss [62]. System-level optimizations, including dynamic batching, distillation, and efficient serving architectures, further align throughput with resource constraints [63]. Early-exit mechanisms, which allow models to terminate inference based on confidence thresholds, can reduce average computation by over 50% in sequence labeling tasks [64, 65].

IX. Acknowledgments

This research was sponsored by SLB. The authors would specifically like to acknowledge the support from Amandine Battentier and Lionel Beneteau. The authors would also like to acknowledge student researchers Benjamin Bi and Anass Jari for their contributions to this effort.

References

[1] Villamar, J., Kelbling, M., Moore, H. L., Denker, M., Tetzlaff, T., Senk, J., and Thober, S., "Metadata Practices for Simulation Workflows," arXiv preprint arXiv:2408.17309, 2024. https://doi.org/10.48550/arXiv.2408.17309, URL https://arxiv.org/abs/2408.17309.

[2] Schlichting, G. S., Boswell, W., Shaver, A. T., Evans, J. P., Pinon-Fischer, O. J., and Mavris, D. N., "Knowledge Capture and Management within an Academic Research Laboratory: a Case Study," *AIAA SCITECH 2024 Forum*, 2024, p. 1134. https://doi.org/10.2514/6.2024-1134, URL https://arc.aiaa.org/doi/10.2514/6.2024-1134.

[3] Kambhampaty, J., Schlichting, G. S., Coletti, C., Evans, J. P., Reddy, A., Pinon-Fischer, O. J., Mavris, D. N., and Graves, R. E., "Graph-based digital file curation for engineering reuse: Methodology and case study," *AIAA SCITECH 2024 Forum*, 2024, p. 1133. https://doi.org/10.2514/6.2024-1133, URL https://arc.aiaa.org/doi/abs/10.2514/6.2024-1133.

[4] Berners-Lee, T., "Metadata Architecture," Tech. rep., World Wide Web Consortium (W3C), Jan. 1997. URL https://www.w3.org/DesignIssues/Metadata.html, accessed 2025-05-10.

[5] Schembera, B., and Iglezakis, D., "EngMeta: metadata for computational engineering," *International Journal of Metadata, Semantics and Ontologies*, Vol. 14, No. 1, 2020, pp. 26–38. https://doi.org/10.1504/IJMSO.2020.107792.

[6] Mishra, A., Ploennigs, J., and Berges, M., "Data-Driven Automatic Metadata Annotation for Building Sensor Data," *Automation in Construction*, Vol. 119, 2020, p. 103353. URL https://www.ashb.com/wp-content/uploads/2020/08/IS-2020-108.pdf.

[7] Kim, H. J., Lell, N., and Scherp, A., "Text Role Classification in Scientific Charts Using Multimodal Transformers," arXiv:2402.14579, 2024. https://doi.org/10.48550/arXiv.2402.14579, accepted to the 2024 Conference on Document Intelligence.

[8] Walsh, J., "ASSESS Initiative: Engineering Simulation Metadata Specification (ESMS)," Tech. rep., NAFEMS, Feb 2024. URL https://www.nafems.org/publications/resource_center/assess_esms/, white paper.

[9] Brase, J., "DataCite—A Global Registration Agency for Research Data," *Proceedings of the 4th International Conference on Cooperation and Promotion of Information Resources in Science and Technology (COINFO 2009)*, 2009, pp. 257–261. https://doi.org/10.1109/COINFO.2009.66.

[10] Gruber, T. R., "A Translation Approach to Portable Ontology Specifications," *Knowledge Acquisition*, Vol. 5, No. 2, 1993, pp. 199–220. https://doi.org/10.1006/knac.1993.1008.

[11] Dagdelen, J., Dunn, A., Lee, S., Walker, N., Rosen, A. S., Ceder, G., Persson, K. A., and Jain, A., "Structured Information Extraction from Scientific Text with Large Language Models," *Nature Communications*, Vol. 15, 2024, p. 1418. https://doi.org/10.1038/s41467-024-45563-x, URL https://www.nature.com/articles/s41467-024-45563-x.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., "Attention Is All You Need," *Advances in Neural Information Processing Systems (NeurIPS) 30*, 2017, pp. 5998–6008.

[13] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT), Volume 1*, 2019, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423, URL https://aclanthology.org/N19-1423.

[14] Brown, T. B., Mann, B., Ryder, N. R., and et al., "Language Models are Few-Shot Learners," *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 1877–1901. URL https://arxiv.org/abs/2005.14165.

[15] Beltagy, I., Lo, K., and Cohan, A., "SciBERT: A Pretrained Language Model for Scientific Text," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3615–3620. https://doi.org/10.18653/v1/D19-1371, URL https://aclanthology.org/D19-1371.

[16] Lewis, P., Perez, E., and et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020, pp. 9459–9474. https://doi.org/10.48550/arXiv.2005.11401, URL https://arxiv.org/abs/2005.11401.

[17] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., and Wang, H., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023. https://doi.org/10.48550/arXiv.2312.10997, URL https://arxiv.org/abs/2312.10997.

[18] Choudhury, S. R., Mitra, P., Kirk, A., Szep, S., Pellegrino, D., Jones, S., and Giles, C. L., "Figure Metadata Extraction from Digital Documents," *Proceedings of the 12th International Conference on Document Analysis and Recognition (ICDAR 2013)*, 2013, pp. 135–139. https://doi.org/10.1109/ICDAR.2013.34.

[19] Radford, A., Kim, J. W., and et al., "Learning Transferable Visual Models from Natural Language Supervision," *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, Vol. 139, 2021, pp. 8748–8763. URL https://arxiv.org/abs/2103.00020.

[20] Li, J., Li, D., Xiong, C., and Hoi, S. C. H., "BLIP: Bootstrapping Language-Image Pre-Training for Unified Vision-Language Understanding and Generation," *Proceedings of the 39th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research, Vol. 162, 2022, pp. 12888–12900. URL https://arxiv.org/abs/2201.12086.

[21] Pan, L., Wang, H., Song, Y., and et al., "FlowLearn: A Flowchart Reasoning Benchmark for Large Vision-Language Models," arXiv:2401.01995, 2024. https://doi.org/https://doi.org/10.48550/arXiv.2407.05183, URL https://arxiv.org/abs/2407.05183.

[22] Hou, S., Shiinoki, R., Koshihara, H., Motegi, M., and Morishige, M., "Overcoming Vision Language Model Challenges in Diagram Understanding: A Proof-of-Concept with XML-Driven Large Language Models Solutions," arXiv:2502.04389, 2025. URL https://arxiv.org/abs/2502.04389.

[23] Jain, S., de Buitléir, A., and Fallon, E., "A Review of Unstructured Data Analysis and Parsing Methods," *Proceedings of the 2020 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2020, pp. 164–169. https://doi.org/10.1109/ESCI48226.2020.9167588.

[24] Kamradt, G., "5 Levels of Text Splitting," , 2024. URL https://github.com/FullStackRetrieval-com/RetrievalTutorials/blob/main/tutorials/LevelsOfTextSplitting/5_Levels_Of_Text_Splitting.ipynb.

[25] Schwaber-Cohen, R., "Chuncking Strategies," https://www.pinecone.io/learn/chunking-strategies/, 2024.

[26] Wei, J., Bosma, M., Zhao, V. Y. T., Guu, K., Schuurmans, D., Petrov, S., and Le, Q. V., "Finetuned Language Models Are Zero-Shot Learners," arXiv:2109.01652, 2021. URL https://arxiv.org/abs/2109.01652.

[27] IBM Research, "What Is Zero-Shot Prompting?" https://www.ibm.com/think/topics/zero-shot-prompting, 2024. Accessed 2025-05-22.

[28] Anonymous, "Improving Zero-Shot Generalization of Instruction Tuning by Data Arrangement," OpenReview submission, ICLR 2024, 2024. URL https://openreview.net/forum?id=Y2AH0wC6C9.

[29] IBM Research, "What Is Chain-of-Thought (CoT) Prompting?" https://www.ibm.com/think/topics/chain-of-thoughts, 2024. Accessed 2025-05-07.

[30] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Zhao, V. Y. T., Guu, K., and et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv:2201.11903, 2022. URL https://arxiv.org/abs/2201.11903.

[31] Xu, Z., Shen, Y., and Huang, L., "MultiInstruct: Improving Multi-Modal Zero-Shot Learning via Instruction Tuning," *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 2023, pp. 11241–11256. https://doi.org/10.18653/v1/2023.acl-long.641, URL https://aclanthology.org/2023.acl-long.641.

[32] Hearst, M. A., *Search User Interfaces*, Cambridge University Press, Cambridge, UK, 2009.

[33] "Information technology — Metadata registries (MDR) — Part 3: Metamodel for registry common facilities," Tech. Rep. ISO/IEC 11179-3:2023, International Organization for Standardization and International Electrotechnical Commission, Geneva, Switzerland, Jan 2023. URL https://www.iso.org/standard/78915.html.

[34] "Micro and Macro Averaging," https://sklearn-evaluation.ploomber.io/en/latest/classification/micro_macro.html, 2023. Accessed 2025-05-22.

[35] Leung, K., "Micro, Macro & Weighted Averages of F1 Score, Clearly Explained," , January 4 2023. URL https://www.kdnuggets.com/2023/01/micro-macro-weighted-averages-f1-score-clearly-explained.html.

[36] "Classification: Accuracy, Recall, Precision, and Related Metrics," https://developers.google.com/machine-learning/crash-course/classification/accuracy, 2024. Accessed 2025-05-08.

[37] "LangChain Conceptual Guide," https://python.langchain.com/docs/concepts/, 2024. Accessed 2025-05-11.

[38] Zhang, Q., Wang, B., Huang, V. S.-J., Zhang, J., Wang, Z., Liang, H., He, C., and Zhang, W., "Document Parsing Unveiled: Techniques, Challenges, and Prospects for Structured Information Extraction," arXiv:2410.21169, 2025. https://doi.org/10.48550/arXiv.2410.21169, URL https://arxiv.org/abs/2410.21169.

[39] Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S., "InstructBLIP: Towards General-Purpose Vision-Language Models with Instruction Tuning," arXiv:2305.06500, 2023. https://doi.org/10.48550/arXiv.2305.06500, URL https://arxiv.org/abs/2305.06500.

[40] Kaushik, S., "Ollama Structured Outputs and Phi-4: A Deep Dive," Medium, May 2025. URL https://medium.com/@shivansh.kaushik/ollama-structured-outputs-and-phi-4-a-deep-dive-77a4a7518ace, accessed: 2025-05-07.

[41] "phi4:14b," Ollama Library, 2025. URL https://ollama.com/library/phi4%3A14b, accessed: 2025-05-07.

[42] Abdin, M., Aneja, J., Behl, H., Bubeck, S., Eldan, R., Gunasekar, S., Harrison, M., Hewett, R. J., Javaheripi, M., Kauffmann, P., Lee, J. R., Lee, Y. T., Li, Y., Liu, W., Mendes, C. C. T., Nguyen, A., Price, E., de Rosa, G., Saarikivi, O., Salim, A., Shah, S., Wang, X., Ward, R., Wu, Y., Yu, D., Zhang, C., and Zhang, Y., "Phi-4 Technical Report," , 2024. https://doi.org/10.48550/arXiv.2412.08905, URL https://arxiv.org/abs/2412.08905.

[43] "Mistral Small 3 vs Mistral Large 2 (Nov '24): Model Comparison," https://artificialanalysis.ai/models/comparisons/mistral-small-3-vs-mistral-large-2, Nov. 2024. Accessed 2025-05-20.

[44] "TheBloke/dolphin-2.7-mixtral-8x7b-GGUF," , 2025. URL https://huggingface.co/TheBloke/dolphin-2.7-mixtral-8x7b-GGUF, accessed: 2025-05-07.

[45] "dolphin-mixtral:8x7b," Ollama Library, 2025. URL https://ollama.com/library/dolphin-mixtral%3A8x7b, accessed: 2025-05-07.

[46] "Information Technology — Metadata Registries (MDR) — Part 3: Registry Metamodel and Basic Attributes," Tech. Rep. ISO/IEC 11179-3:2013, International Organization for Standardization, 2013. International standard.

[47] "Prompt Engineering Guide," https://www.promptingguide.ai/, 2024. Accessed 2025-05-07.

[48] Manda, B., Bhaskare, P., and Muthuganapathy, R., "A Convolutional Neural Network Approach to the Classification of Engineering Models," *IEEE Access*, Vol. 9, 2021, pp. 22711–22723. https://doi.org/10.1109/ACCESS.2021.3055826.

[49] Canny, J. F., "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 6, 1986, pp. 679–698. https://doi.org/10.1109/TPAMI.1986.4767851.

[50] Valipour, M., Rezagholizadeh, M., Kobyzev, I., and Ghodsi, A., "DyLoRA: Parameter Efficient Tuning of Pre-Trained Models Using Dynamic Search-Free Low-Rank Adaptation," arXiv:2210.07558, 2022. https://doi.org/10.48550/arXiv.2210.07558, URL https://arxiv.org/abs/2210.07558.

[51] Chavan, A., Liu, Z., Gupta, D. K., Xing, E., and Shen, Z., "One-for-All: Generalized LoRA for Parameter-Efficient Fine-Tuning," arXiv:2306.07967, 2023. https://doi.org/10.48550/arXiv.2306.07967, URL https://arxiv.org/abs/2306.07967.

[52] Liu, Z., Lyn, J., Zhu, W., Tian, X., and Graham, Y., "ALoRA: Allocating Low-Rank Adaptation for Fine-Tuning Large Language Models," arXiv:2403.16187, 2024. https://doi.org/10.48550/arXiv.2403.16187, URL https://arxiv.org/abs/2403.16187.

[53] Li, X. L., and Liang, P., "Prefix-Tuning: Optimizing Continuous Prompts for Generation," *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL 2021)*, 2021, pp. 4582–4597. https://doi.org/10.18653/v1/2021.acl-long.353, URL https://aclanthology.org/2021.acl-long.353.

[54] Liu, S.-Y., Wang, C.-Y., Yin, H., Molchanov, P., Wang, Y.-C. F., Cheng, K.-T., and Chen, M.-H., "DoRA: Weight-Decomposed Low-Rank Adaptation," arXiv:2402.09353, 2024. https://doi.org/10.48550/arXiv.2402.09353, URL https://arxiv.org/abs/2402.09353.

[55] Zhang, S., Xu, Y., Usuyama, N., Xu, H., Bagga, J., Tinn, R., Preston, S., Rao, R., Wei, M., Valluri, N., Wong, C., Tupini, A., Wang, Y., Mazzola, M., Shukla, S., Liden, L., Gao, J., Crabtree, A., Piening, B., Bifulco, C., Lungren, M. P., Naumann, T., Wang, S., and Poon, H., "BiomedCLIP: A Multimodal Biomedical Foundation Model Pretrained from Fifteen Million Scientific Image–Text Pairs," arXiv:2303.00915, 2025. https://doi.org/10.48550/arXiv.2303.00915, URL https://arxiv.org/abs/2303.00915.

[56] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F.-H., Chi, E. H., Le, Q. V., and Zhou, D., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," arXiv:2201.11903, 2022. https://doi.org/10.48550/arXiv.2201.11903, URL https://arxiv.org/abs/2201.11903.

[57] Gao, T., Fisch, A., and Chen, D., "Making Pre-trained Language Models Better Few-Shot Learners," arXiv:2012.15723, 2020. https://doi.org/10.48550/arXiv.2012.15723, URL https://arxiv.org/abs/2012.15723.

[58] Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., Chowdhery, A., and Zhou, D., "Self-Consistency Improves Chain of Thought Reasoning in Language Models," arXiv:2203.11171, 2022. https://doi.org/10.48550/arXiv.2203.11171, URL https://arxiv.org/abs/2203.11171.

[59] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E. H., and Zhou, D., "Rationale-Augmented Ensembles in Language Models," arXiv:2205.12501, 2022. https://doi.org/https://doi.org/10.48550/arXiv.2207.00747, URL https://arxiv.org/pdf/2207.00747.

[60] "LLM Leaderboard – Comparison of GPT-4o, Llama 3, Mistral, and over 30 Models," https://artificialanalysis.ai/leaderboards/models, 2025. Accessed 2025-05-22.

[61] Reimers, N., and Gurevych, I., "Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks," *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, pp. 3982–3992. https://doi.org/10.18653/v1/D19-1410, URL https://aclanthology.org/D19-1410.

[62] Grebennikov, R., "How to Compute LLM Embeddings 3× Faster with Model Quantization," https://medium.com/nixiesearch/how-to-compute-llm-embeddings-3x-faster-with-model-quantization-25523d9b4ce5, Sep. 2023. Medium blog, accessed 2025-05-22.

[63] Zhang, Z., Sheng, Y., Zhou, T., Chen, T., Zheng, L., Cai, R., Song, Z., Tian, Y., Ré, C., Barrett, C., Wang, Z., and Chen, B., "H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models," *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023, pp. 1–15. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/6ceefa7b15572587b78ecfcebb2827f8-Paper-Conference.pdf.

[64] Li, X., Shao, Y., Sun, T., Yan, H., Qiu, X., and Huang, X., "Accelerating BERT Inference for Sequence Labeling via Early-Exit," arXiv:2105.13878, 2021. https://doi.org/10.48550/arXiv.2105.13878, URL https://arxiv.org/abs/2105.13878.

[65] Valade, F., "Accelerating Large Language Model Inference with Self-Supervised Early Exits," arXiv:2407.21082, 2024. https://doi.org/10.48550/arXiv.2407.21082, URL https://arxiv.org/abs/2407.21082.