CLEAR

Comprehensive Learning for Enhanced AI Responsiveness

D3.1 State-of-the-art and practices in multi-modal fine-tuning

Submission date of deliverable: May 31, 2026

Edited by: Abbas K. (RISE), Mehrdad S. (RISE), Hakan K. (Orion Innovation), Yuriy Y. (Vaadin), Abdul R. (Test Scouts), Nicolás G. (SIRRIS), Sreeraj R. (SIRRIS), Pehlivan T. S. (U3CM), Juan G. (U3CM)

Project start date	Dec 1, 2025
Project duration	36 months
Project coordinator	Mehrdad Saadatmand, RISE Research Institutes of Sweden
Project number & call	24026 - ITEA Call 2024
Project website	https://itea4.org/project/clear.html & https://rprj.net/p/clear

Contributing partners	RISE, SIRRIS, VTT, Wapice, Vaadin, Materialise, Alstom, FormalTrust AB, Defne, Panel Sistemas, Test Scouts, UC3M, Ekkono, eAgronom, STACC and ONIZEA.
Version number	V1.0
Work package	WP3
Work package leader	Abbas Khan, RISE Research Institutes of Sweden
Dissemination level	Public
Description	This deliverable presents a state-of-the-art analysis of multimodal learning and fine-tuning approaches relevant to the CLEAR project application domains. The study combines a limited scoping review with a cursory survey of existing multimodal large language models, frameworks, and various practices to identify current trends, capabilities, challenges, and limitations.

Executive Summary

Modern systems generate and manage information across multiple modalities, including source code, models, requirements, logs, documentation, time-series data, and three-dimensional representations. These heterogeneous data sources collectively capture important architectural decisions, operational, and domain knowledge. Multimodal understanding enables systems to process, integrate, and reason across such diverse modalities, supporting more informed analysis, decision-making, and automation in industrial environments.

Recent advances in multimodal learning and large language models have demonstrated significant capabilities in combining information from different modalities and adapting pretrained models to domain-specific tasks through finetuning techniques. Such approaches enable the development of intelligent systems capable of capturing both general multimodal knowledge and specialized domain expertise. As a result, multimodal approaches have become increasingly relevant for industrial decision support.

Within the CLEAR project, multimodal understanding and reasoning are of paramount importance across the project use-cases, particularly in domains involving software engineering artefacts, time-series data, and three-dimensional models. CLEAR use cases require domain-specific multimodal pipelines capable of integrating heterogeneous information sources while also leveraging the broader reasoning capabilities of general-purpose multimodal foundation models. Therefore, the adaptation and finetuning of existing pretrained multimodal models play a critical role in enabling CLEAR-specific intelligence and decision-support capabilities.

This deliverable presents a state-of-the-art investigation of multimodal learning and finetuning approaches relevant to the CLEAR project domains. The study adopts a mixed-method research methodology that combines a limited scoping literature review with a cursory survey of existing open-source multimodal large language models. Through this methodology, the deliverable identifies, analyses, and synthesizes current multimodal learning and finetuning methods covering approaches for software engineering artefacts, time-series data, and three-dimensional data representations, while also considering emerging multimodal foundation models and open-source ecosystems. In addition to identifying current capabilities and promising approaches, the deliverable analyses gaps and limitations associated with the current approaches.

The findings of this deliverable provide the CLEAR consortium with a consolidated overview of current multimodal landscape, supporting future design decisions, technology selection, and the development of domain-specific multimodal pipelines within the project. Furthermore, the identified limitations and research gaps highlight opportunities for future research and innovation in multimodal learning and industrial decision support systems.

Table of contents

1. Introduction

Modern industrial and software-intensive systems generate large volumes of heterogeneous data across multiple modalities, including execution logs, error reports, User Interface (UI) alerts, logs, telemetry streams, images, three-dimensional (3D) representations, and textual documentation. In addition, processes to develop and maintain such systems produces additional multimodal information like requirements, code, architectural diagram, release logs and refactoring reports [1]. Further, such system might deal with multimodal input and output themselves [2]. In practice, these modalities are often fragmented across tools, repositories, and operational environments, making it difficult for engineers and decision-makers to obtain a unified understanding of the system or process state. As a result, important architectural and operational decisions frequently require time-consuming manual analysis, cross-referencing of multiple information sources, and coordination among domain experts. This fragmentation of information increases as the complexity of decision-making processes increases, and it limits the ability to efficiently derive actionable insights from available data.

Recent advances in multimodal artificial intelligence (AI) offer promising opportunities to address these challenges [3]. Multimodal AI systems are designed to process, integrate, and reason across multiple heterogeneous modalities simultaneously, enabling richer contextual understanding than traditional unimodal approaches. By combining information from text, images, structured data, time-series signals, 3D representations, and other modalities, multimodal systems can support more comprehensive reasoning, contextual interpretation, and intelligent decision support [4]. Such capabilities are increasingly relevant for complex industrial environments where critical knowledge is distributed across different data sources and representations.

The rapid development of large language models (LLMs) and multimodal foundation models has further accelerated progress in this area [5]. Modern LLMs and vision-language models (VLMs) demonstrate strong general reasoning capabilities and can be adapted to a wide variety of downstream tasks through prompting, fine-tuning, retrieval augmentation, and parameter-efficient finetuning (PEFT) [6] and adaptation methods. These models can handle multimodal inputs, generate structured outputs, and support complex workflows involving reasoning, summarization, retrieval, generation, and interaction. Their growing use in software engineering [4], industrial analytics, robotics [7], computer-aided design (CAD) [8], and time-series analysis [9] highlights their potential to transform industrial decision-support systems and engineering workflows.

Within the CLEAR project, multimodal understanding and reasoning are essential for addressing the needs of the project use cases. The project involves application scenarios where relevant information is distributed across software engineering artefacts, time-series data, and 3D models. The CLEAR project would require domain-specific multimodal pipelines capable of integrating heterogeneous data sources while leveraging the broader reasoning and knowledge capabilities of modern multimodal foundation models. To support the design and development of such pipelines, it is necessary to understand the current State-of-the-Art (SoTA) in multimodal learning, and fine-tuning approaches relevant to the project domains.

Several existing review studies have investigated aspects of multimodal AI and have mapped the SoTA. However, the rapidly evolving landscape of multimodal learning requires updating the mapping of SoTA. Furthermore, the rapid pace of development in multimodal foundation models means that many emerging systems and open-source implementations may not yet comprehensively covered in traditional survey literature. This deliverable presents a SoTA review of multimodal learning and fine-tuning approaches relevant to the CLEAR project. The review combines a limited scoping literature review [10] with a cursory survey [11] of existing open-source multimodal large language models and frameworks to achieve the goals of the review, as follows:

- identify and map multimodal learning and finetuning approaches relevant to the CLEAR application domains;
- analyze modality combinations, fusion mechanisms, and fine-tuning strategies;
- investigate opensource multimodal foundation models and multimodal LLMs;
- identify limitations, research gaps, challenges in SoTA.

The remainder of this deliverable is structured as follows. Section 2 presents the research methodology, including the review process, objectives, search strategy, study selection, and data extraction procedures. Section 3 presents the findings of the review, including an overview of the SoTA, analysis of modality combinations, fusion mechanisms, fine-tuning strategies, foundation models, and identified limitations and research gaps. Section 4 presents the threats to validity and Section 5 discusses the results and concludes the deliverable.

2. Methodology and Scope

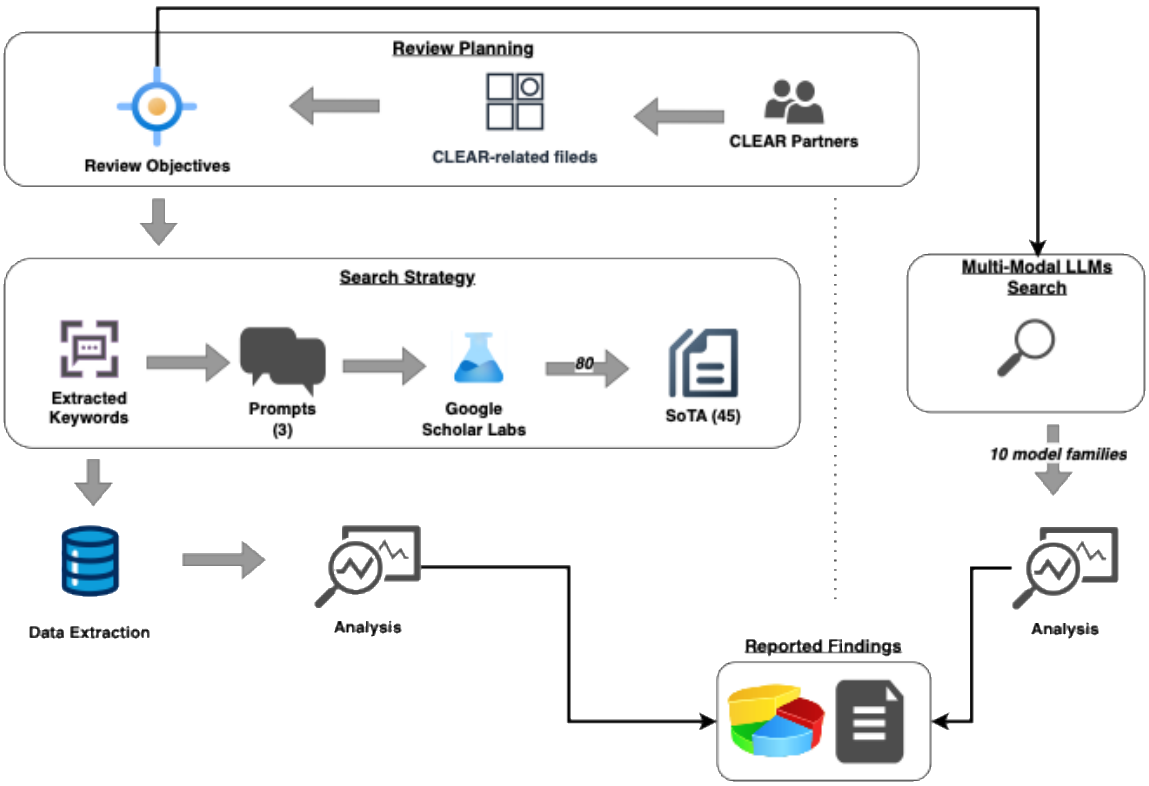

Figure 1: SoTA Review Process

This section describes the methodology adopted for conducting the SoTA review within the CLEAR project. The methodology defines the overall research design, review objectives, literature collection strategy, multimodal LLM identification process, data extraction procedure, and the methods used to synthesize the findings. As illustrated in Figure 1, the review follows a mixed-method research methodology that combines a structured scoping literature review [10] with a targeted survey [11] of currently available LLMs frameworks, and tools.

The adopted methodology is designed to capture both academic literature and available tools. While scientific publications provide validated and peer-reviewed knowledge, the rapidly evolving nature of multimodal LLM technologies means that many relevant systems, frameworks, and applications are first released through open-source repositories, technical documentation, industrial reports, and online communities before appearing in formal academic literature. Consequently, the methodology

integrates both scholarly and grey literature sources to provide a comprehensive overview of the current technological landscape.

2.1. Review Planning

As shown in Figure 1 (Review Planning), the CLEAR consortium first identified and consolidated relevant application areas of multimodal AI through discussions and exploratory analysis of the CLEAR use cases. These identified application domains, together with the initial scope and objectives of Deliverable D3.1, served as the foundation for formulating three overarching research questions to guide the review.

To ensure a structured and rigorous review process, GQM approach was adopted. The high-level research questions were systematically decomposed into a set of sub-goals, sub-research questions, and associated data extraction metrics. This process enabled us to establish clear traceability between the review objectives, the evidence collected from the literature, and the analytical dimensions used during synthesis. The resulting GQM structure provides a consistent framework for analyzing and reporting multimodal fine-tuning approaches across different domains (software engineering, three-dimensional software and timeseries), modalities, model architectures, and application scenarios.

2.2. Objectives and Research Questions

Recent advances in foundation models, LLMs, and multimodal learning have significantly expanded the capabilities of AI systems to process and reason across heterogeneous data sources [2]. Instead of relying on a single modality, modern AI systems increasingly combine text, source code, images, logs, time-series signals, telemetry, 3D representations, and user interaction data to support complex analytical and decision-making tasks. This shift has accelerated research on multimodal fine-tuning approaches, where pre-trained models are adapted to domain-specific tasks through the integration and alignment of multiple modalities.

Within the context of the CLEAR project, multimodal AI is particularly relevant for domains such as software engineering, industrial Internet of Things (IoT), time-series analysis, and 3D engineering data understanding. These domains often involve highly heterogeneous data environments where meaningful insights, often drawn from many modalities, emerge only through the combination of multiple information sources. At the same time, the rapid pace of development in multimodal AI has produced a fragmented research landscape spanning peer-reviewed publications, preprints, and industrial grey literature. Therefore, there is a need for an analysis of the current state of the art to better understand existing approaches, their effectiveness, and their limitations for the CLEAR project to guide future development in the project.

The objective of this SoTA deliverable is therefore to identify, document, and analyze the state of the art in multimodal fine-tuning approaches relevant to the CLEAR project. The review focuses on understanding how multimodal systems are designed, fine-tuned, and evaluated across different model architectures, data modalities, and application domains. In addition to identifying current technical trends, the review also aims to assess the maturity of existing solutions, highlight open research challenges, and identify opportunities for future research and development within CLEAR.

To guide the SoTA review process, the following three main research questions (RQs) were defined:

- **RQ1:** How is multimodal fine-tuning implemented across different data modalities and model architectures?
- **RQ2:** How do multimodal approaches improve (or fail to improve) performance in CLEAR-related tasks?

- **RQ3:** What are the current limitations, challenges, and research gaps in existing multimodal systems?

Within the scope of this review, CLEAR-related tasks include:

- *Software engineering tasks* involving source code, logs, telemetry, requirements, tests and software models
- *Time-series analysis* and IoT-related applications
- *3D and structured data* understanding, including CAD models and point clouds

To ensure a rigorous review methodology, the GQM approach was adopted. The high-level research questions were decomposed into a set of operational goals, sub-questions, and associated data extraction metrics. This structure enables consistent evidence collection and analysis across heterogeneous studies while maintaining traceability between the review objectives, extracted evidence, and the final synthesis of findings. Below, we present the operational goals and their metrics.

Goal 1: Characterize the State of the Art in Multimodal Fine-Tuning
Goal 1 establishes a structured understanding of how multimodal fine-tuning approaches are implemented across different modalities, architectures, and learning paradigms. *Related Research Questions:* **Q1.1:** What types of modalities are combined in current multimodal approaches? Examples include code and text, time-series and text, point clouds and text, and UI interaction data combined with logs or telemetry. **Q1.2:** What fine-tuning strategies are used? This includes full model fine-tuning, parameter-efficient fine-tuning approaches (e.g., LoRA or adapters), and modality-specific versus joint optimization strategies. **Q1.3:** How is multimodal fusion implemented? Examples include early fusion, late fusion, hybrid architectures, and tool-based integration mechanisms. **Q1.4:** What types of foundation models and backbone architectures are used? Examples include LLMs, VLMs, hybrid multimodal architectures, and time-series foundation models. *Metrics and Data Extraction* • Modalities used • Fusion strategy and implementation type • Fine-tuning method • Model backbone architecture • Training data composition

Goal 2: Assess Multimodal Approaches for CLEAR-Related Software Engineering Tasks
Goal 2 evaluates how multimodal methods are applied to software engineering tasks relevant to CLEAR and assess their effectiveness in comparison. *Research Questions* **Q2.1:** Which software engineering tasks are addressed by multimodal systems?

Goal 2: Assess Multimodal Approaches for CLEAR-Related Software Engineering Tasks

Examples include code generation, debugging, UI generation, DevOps support, and requirements analysis.

Q2.2: How is multimodality leveraged within these tasks?

For example, combining logs and source code, UI representations and code, or telemetry with natural language descriptions.

Q2.3: Do multimodal approaches outperform unimodal baselines?

The review investigates whether multimodal integration provides measurable improvements in performance or task completion quality (if reported).

Q2.4: What is the level of automation supported by current systems?

This includes static pipelines, semi-autonomous workflows, and autonomous agent-based systems.

Metrics and Data Extraction

- Software engineering task category
- Performance improvements compared to baselines
- Use of agents or autonomous components
- Degree of automation

Goal 3: Analyse Time-Series Analysis in Multimodal Contexts

Goal 3 investigate how time-series data is integrated into multimodal learning systems, particularly within IoT and industrial contexts relevant to CLEAR.

Research Questions

Q3.1: What types of time-series models are used?

Including native time-series foundation models and LLM-adapted time-series approaches.

Q3.2: How is time-series data integrated with other modalities?

Examples include combining time-series data with text, logs, or user interaction signals.

Q3.3: Are generative time-series models used, and for what purpose?

Such as forecasting, anomaly detection, simulation, or synthetic data generation.

Q3.4: How does fine-tuning for time-series data differ from fine-tuning approaches used for other modalities?

Metrics and Data Extraction

- Time-series model type
- Input-output modality combinations
- Target task (forecasting, anomaly detection, control, etc.)
- Fine-tuning approach
- Dataset characteristics and domain

Goal 4: Analyse Multimodal Learning for 3D and Structured Data Understanding

Goal 4 examine how multimodal systems process and integrate 3D and structured engineering data, including CAD models and point clouds.

Research Questions

Q4.1: What representations are used for 3D data?

Examples include point clouds, meshes, and CAD graphs.

Q4.2: How are 3D modalities fused with text or code?

Including instruction tuning, embedding alignment, and cross-attention mechanisms.

Q4.3: What application tasks are targeted?

Examples include design validation, simulation, and instruction-following systems.

Q4.4: What limitations exist in current multimodal fusion techniques for 3D data?

Metrics and Data Extraction

- 3D representation type
- Fusion mechanism
- Task category
- Evaluation metrics and reported performance

Goal 5: Analyze Open-Source Multimodal Foundation Models and LLM Ecosystems

Goal 5 investigate the current landscape of open-source multimodal large language models and foundation models, with a focus on their architectures, multimodal capabilities, and limitations.

Open-source multimodal LLMs are increasingly important for research reproducibility, customization, transparency, and deployment in privacy-sensitive or domain-specific environments (such as CLEAR's industrial partners' setup). In addition, the analyzed literature might miss mentioning or using these open-source models. Therefore, this goal is achieved with a dedicated cursory survey that searches and analyze a variety of open-source multi-modal LLMs.

Research Questions

Q5.1: How has the architecture of open-source multimodal large language models evolved? How has this evolution impacted the models?

Examples include modular bridge, and native unified architecture types.

Q5.2: What modalities do multimodal LLMs support? How do they shape real-world use cases?

Examples include bi-modal such as text and image, and omni-modal types such as text, image, audio, video, speech.

Metrics and Data Extraction

- Evaluation progress
- Architecture types
- Supported modalities
- Key features and primary use cases

Goal 6: Identify Gaps, Limitations, and Research Opportunities
Goal 6 synthesizes the limitations of current multimodal systems and identify underexplored research directions relevant to CLEAR. *Research Questions* **Q6.1:** Where do current multimodal approaches fail? Examples include temporal reasoning, cross-modal alignment, scalability, and robustness. **Q6.2:** Which modality combinations or application areas remain underexplored? **Q6.3:** What evaluation and benchmarking gaps exist in the literature? **Q6.4:** What limitations are reported in existing multimodal systems?
Metrics and Data Extraction • Reported limitations and recurring themes • Missing modality combinations • Evaluation weaknesses • Maturity level of the proposed systems (prototype, benchmarked, or production-ready)

2.3. Search & Study Selection

As shown in Figure 1, a structured search strategy was designed based on the identified CLEAR application domains and the review objectives. The objective of the search process was to identify relevant SoTA, technical reports, and emerging multimodal AI approaches applicable to the CLEAR project.

The search strategy was developed collaboratively within the CLEAR consortium. Consortium partners were first asked to provide search phrases, technical topics, and domain-specific concepts related to the identified CLEAR application areas. These inputs were subsequently abstracted into a set of high-level keywords representing the core research themes of the review.

The extracted keywords were then used to formulate prompts for the AI-assisted literature search engine Google Labs. Google Labs was selected because it enables semantic and prompt-driven literature retrieval, allowing searches to capture conceptually related studies beyond exact keyword matching. Furthermore, Google Labs retrieves publications indexed through Google Scholar, which includes peer-reviewed scientific literature as well as selected grey literature sources such as technical reports, white papers, preprints, and industrial publications. This capability makes it suitable for identifying recent developments in the rapidly evolving domain of multimodal AI.

To ensure the review focused on recent advances in multimodal learning and foundation models, the search was restricted to studies and technical resources published between January 2022 and April 2026. This timeframe was selected because major advances in multimodal large language models and foundation model architectures have largely emerged recently.

Three search prompts were created to target the primary application domains relevant to the CLEAR project:

- **Google Labs Prompt 1:**

  Multimodal approaches for 3D tasks, including 3D data, point clouds, and CAD data, between 2022 and 2026.

- **Google Labs Prompt 2:**
  Multimodal approaches for software engineering dealing with various data modalities such as user interfaces, models, source code, software releases, and system logs, between 2022 and 2026.
- **Google Labs Prompt 3:**
  Multimodal approaches for time-series data between 2022 and 2026.

The search process initially retrieved approximately 80 candidate studies and technical resources across the three prompts in multiple executions. The retrieved results included peer-reviewed conference papers, journal articles, arXiv preprints, technical reports, and other forms of relevant grey literature. To maintain the feasibility of the review while ensuring adequate coverage of the domain, the retrieved studies were ranked according to their relevance to the prompt and the top 15 most relevant studies for each prompt were selected, resulting in a final set of 45 primary studies for detailed review and analysis.

The study selection process was conducted in multiple stages. Initially, titles and abstracts were screened to determine relevance to the review objectives. This was followed by an assessment of Google Labs generated summary of the studies and finally a full-text analysis. During this process, inclusion and exclusion criteria were applied to ensure consistency across the selected studies.

Inclusion Criteria

Studies were included in the review if they satisfied all of the following criteria:

(I1) The study is written in English. (I2) The study addresses multimodal learning, multimodal foundation models, or multimodal data processing. (I3) The study presents a method, approach, framework, technique, empirical evaluation, or opinion relevant to multimodal AI. (I4) The study was published between January 2022 and May 2026.

Exclusion Criteria

Studies were excluded from the review if they met any of the following conditions:

(E1) The study does not satisfy the inclusion criteria. (E2) The study is exclusively a survey or secondary review paper focusing on a narrowly specialized topic (e.g., multimodal fine-tuning techniques) without presenting original approaches or opinions relevant to the review objectives. (E3) The study does not explicitly address multimodal learning and only processes unimodal data. (E4) The study is a duplicate version of an already selected publication.

The selected studies were used for structured data extraction and comparative analysis, as described in the following sections.

For the survey study on open-source multimodal large language models, the search was conducted using a two-phase approach. In the first phase, a search was conducted on platforms hosting open-source models, such as Hugging Face and Ollama, to identify current representative model families. In the second phase, information such as the architectural structures and supported modalities of these models was identified and analyzed based on real-world use cases. Within this scope, a total of 10 model families, representing different architectural approaches and use cases, and 15 to 20 open-source multimodal LLMs belonging to these families were reached in this study.

2.4. Study Execution, Data Extraction and Synthesis

The search and study selection process was executed in April 2026 following the search strategy and selection criteria described in the previous section. The execution process was conducted by the CLEAR partners with efforts in WP3. Initially, one researcher performed the primary search execution using the three predefined Google Labs prompts. The retrieved studies were screened by applying the inclusion and exclusion criteria to identify publications relevant to the CLEAR objectives and application domains. During this phase, duplicate entries, unrelated studies, and studies lacking sufficient relevance to multimodal AI were removed. To maintain a manageable yet representative set of primary studies, a maximum of 15 studies per topic area were selected, resulting in a total of 45/80 primary studies for detailed review.

Following the initial filtering phase, the selected studies were subjected to a secondary screening process. Both the AI-generated summaries provided by Google Labs and the full-text versions of the publications were screened to validate their relevance and ensure alignment with the review objectives. This additional validation step helped identify studies that, despite appearing relevant during the initial screening, did not sufficiently address multimodal learning, multimodal foundation models, or CLEAR-related application scenarios upon closer inspection. After the final set of studies was confirmed, the data extraction phase was initiated. To distribute the review effort across the consortium and leverage domain expertise from different partners, each CLEAR partner with efforts in WP3 were assigned three studies for detailed analysis and data extraction. A standardized extraction template based on our review objectives was used to ensure consistency across papers and reviewers. The extraction process captured both quantitative and qualitative information from each study, including publication metadata, addressed application domain, supported modalities, model architectural approaches, datasets and benchmarks, evaluation methods, model capabilities, fusion strategies, limitations and challenges, deployment and validation levels. In addition to structured quantitative metrics, reviewers also extracted open-ended observations and qualitative insights related to emerging research trends, practical implementation challenges, and opportunities for future development.

Following the extraction phase, the collected data was synthesized and analyzed by seven CLEAR partners. The analysis process involved synthesizing demographic information, technical characteristics, qualitative observations, and comparative findings across the selected studies. The partners identified recurring themes, research gaps, and other insights relevant to the CLEAR project. The synthesized findings from this process and the comparative analysis of multimodal LLMs are reported and presented in the following sections of this deliverable.

The data extraction and synthesis process for open-source multimodal LLMs (MLLMs) was structured in accordance with the defined research methodology. For the selected models, information such as release date, architectural structure, supported modalities, key features, and use-cases and other relevant metrics were extracted using a template. As a result of this analysis, comparative and comprehensive insights regarding the development trends and application potential of MLLMs were derived.

3. Findings

This section presents the findings of the limited scoping review and cursory survey conducted as part of this deliverable. The findings synthesize the current SoTA (Section 3.1) in multimodal learning, multimodal reasoning, and fine-tuning approaches relevant to the CLEAR project domains, including software engineering artefacts (Section 3.2), time-series data (Section 3.3), and three-dimensional models (Section 3.4). In addition, the section analyses existing open-source multimodal large language models (Section 3.5), and emerging research directions, while highlighting current limitations, challenges, and opportunities for future development (Section 3.6).

3.1. Overview of state-of-the-art

This section presents a cross-domain overview of how multimodal fine-tuning is currently approached in the research literature, drawing on all 45 papers covered in the review. The review spans three primary application domains: Software Engineering (SE), 3D Computer-Aided Design (3D/CAD), and Time-Series Analysis (TS). Collectively, these papers characterize the state of the art with respect to the modalities that are combined, the fine-tuning strategies that are employed, the fusion mechanisms used to integrate heterogeneous inputs, and the backbone architectures on which models are built. The four sub-sections below address the four research questions that structure Goal 1 of this deliverable.

Goal 1: Characterize the State of the Art in Multimodal Fine-Tuning
Goal 1 establishes a structured understanding of how multimodal fine-tuning approaches are implemented across different modalities, architectures, and learning paradigms. *Related Research Questions:* **Q1.1:** What types of modalities are combined in current multimodal approaches? Examples include code and text, time-series and text, point clouds and text, and UI interaction data combined with logs or telemetry. **Q1.2:** What fine-tuning strategies are used? This includes full model fine-tuning, parameter-efficient fine-tuning approaches (e.g., LoRA or adapters), and modality-specific versus joint optimization strategies. **Q1.3:** How is multimodal fusion implemented? Examples include early fusion, late fusion, hybrid architectures, and tool-based integration mechanisms. **Q1.4:** What types of foundation models and backbone architectures are used? Examples include LLMs, VLMs, hybrid multimodal architectures, and time-series foundation models. *Metrics and Data Extraction* • Modalities used • Fusion strategy and implementation type • Fine-tuning method • Model backbone architecture • Training data composition

Modalities

Q1.1 asks what types of modalities are combined in current multimodal approaches. Across the 45 reviewed papers, a wide and heterogeneous set of input modality combinations is observed, reflecting the different representational demands of each application domain.

In the 3D/CAD domain (15 papers, 3D1 - 3D15 in the bibliography), the dominant input modality is the 3D point cloud, which appears either alone or in combination with mesh data, images, text descriptions, sketches, or CAD construction sequences. Papers such as CAD-Recode [3D5], TransCAD [3D7], and CAD-SIGNet [3D6] take a raw point cloud as their sole input and produce structured CAD command sequences. Others extend this to richer multimodal inputs: Img2CAD [3D10] operates from RGB images supplemented by semantic structural information predicted by a vision-language model; SldprtNet [3D11] combines multi-view rendered images with parametric CAD scripts; and the multi-system work [3D1] spans sketches, text descriptions, images, and UI

screen sequences across three sub-systems. The most modality-rich 3D paper is Req2CAD [3D15], which integrates text, UI elements, CAD geometry, images, point clouds, and B-rep representations in a single agentic pipeline.

In the Software Engineering domain (15 papers, SE1 - SE15 in the bibliography), text is the universal modality, appearing in every paper, but it is regularly combined with code, images, UI artefacts, logs, and design diagrams. Code appears alongside text in papers on bug triaging [SE11], code search [SE15], issue classification [SE6, SE10], and code generation from visual inputs [SE5, SE13]. UI and image modalities feature in mobile app bug verification [SE2], UI element grouping [SE8], and multimodal code generation from diagrams [SE1, SE14]. System-level telemetry data, including metrics, logs, and distributed traces, forms the multimodal input for observability and performance testing work [SE4, SE9]. The proposed defect-detection framework [SE3] is the most comprehensive on the SE side, enumerating source code, metrics, requirements documents, UML models, version-control history, and execution logs as inputs.

In the Time-Series Analysis domain (15 papers, TS1 - TS15 in the bibliography), the combination of numerical time-series data with natural language text is by far the most prevalent pattern, appearing in eleven of the fifteen papers. The textual companion may be news articles, weather reports, financial summaries, or domain metadata. A smaller set of papers converts time-series signals into visual form before feeding them to vision-capable models: VisionTS++ [TS4] encodes each variate as a colored subplot in a composite image, while MLLM4TS [TS7] renders channels as color-coded line plots. Aurora [TS5] is the most modality-diverse TS paper, combining raw numerical signals, text, and synthetically generated 2D images. Di Martino et al. [TS6] represents a distinct sub-class, combining physiological and inertial measurement unit (IMU) signals such as ECG, EMG, EDA, accelerometer, gyroscope, and magnetometer, rather than text, as the second modality.

Figure 2 summarizes the primary input modality combination for each of the 45 papers, with each paper assigned to one category and each domain contributing exactly 15 papers. In the 3D/CAD domain, point cloud or 3D mesh inputs account for 6 papers and image combined with text for the remaining 9, reflecting the shift towards vision-language pipelines for CAD generation. In the Software Engineering domain, image combined with text is the largest category (11 papers), covering UI screenshots, design diagrams, and engineering documentation, while code or log data combined with text accounts for 4 papers. In the Time-Series domain, text paired with numerical time-series data dominates at 12 papers, with the remaining 3 split between time-series rendered as images (2 papers) and multi-signal sensor fusion (1 paper). The figure makes clear that each domain has a distinct primary modality profile, with very little overlap across domains.

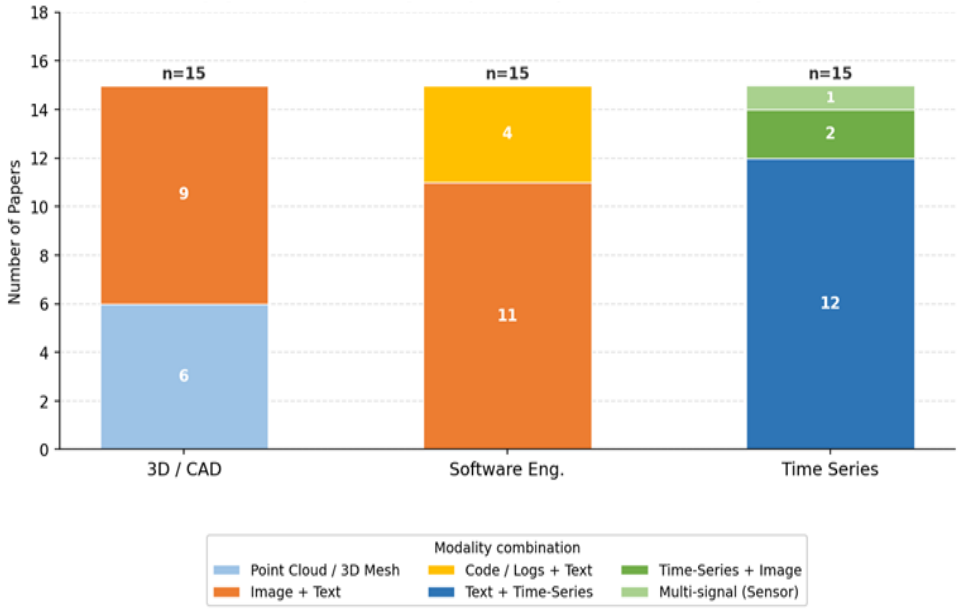

Figure 2: Primary input modality combinations across the 45 reviewed papers, grouped by application domain (n=15 per domain). Each paper is assigned to one modality category. Stacked segments show the breakdown within each domain; segment labels indicate paper counts. Image + Text is the only category that appears across all three domains

On the output side, the reviewed papers are similarly varied. Generated CAD command sequences and executable CAD code dominate in the 3D/CAD domain classification labels, defect flags, and textual summaries dominate in Software Engineering and numerical forecast values or structured temporal predictions dominate in Time-Series Analysis. A smaller but notable set of papers produces mixed outputs: Kapoor et al. [SE4] output both a severity classification and a root-cause explanation, while Parker et al. [TS8] produce forecasts, anomaly labels, question-answering responses, and textual reports from the same model. This breadth of output types underlines that multimodal fine-tuning in the reviewed literature spans generation, classification, retrieval, and reasoning. Notably, Image + Text is the only modality combination that appears across all three domains, appearing in 9 of the 15 3D/CAD papers, 11 of the 15 SE papers, and 1 TS paper, which reflects its generality as a fusion substrate across very different tasks.

Fine-Tuning Strategies

Q1.2 concerns what fine-tuning strategies are used across the 45 reviewed papers. For the purposes of this analysis, each paper was assigned to one of four meaningful training strategy categories: Full Fine-Tuning (all or most model parameters updated on the target dataset), Zero-Shot / No Training (the model is used as-is without any parameter updates), PEFT / Partial Fine-Tuning (only a small subset of parameters is updated, for example via LoRA, adapters, or frozen backbone strategies), and Mixed / Multi-Stage (the paper combines two or more of the above strategies in sequence). A separate count tracks papers where the fine-tuning approach was not described with sufficient detail to allow categorization. Figure 3 presents the distribution.

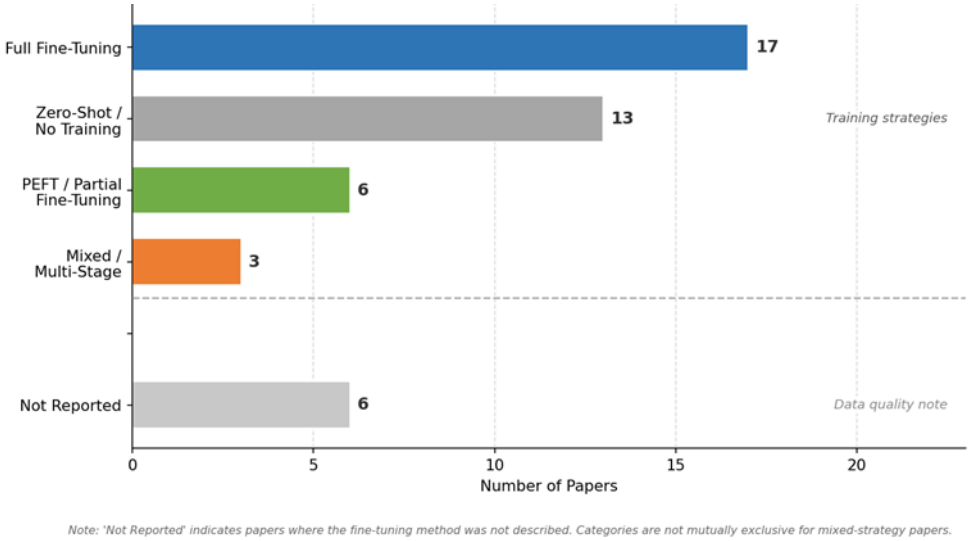

Figure 3: Distribution of fine-tuning strategies across all 45 reviewed papers. The four coloured bars represent distinct training strategies; the grey bar at the top is a data quality note indicating papers where the fine-tuning method was not reported. A dashed line separates the two groups. Full fine-tuning is the most common strategy (17 papers), followed by zero-shot or no-training approaches (13 papers) Distribution of fine-tuning strategies across all 45 reviewed papers. Full fine-tuning is the most common single strategy; zero-shot or no-training approaches are the second largest group.

Full fine-tuning, which involves updating all or the majority of model parameters on a target dataset, is the most frequently reported strategy, appearing in 17 papers. In the 3D/CAD domain this is the norm: CAD-Recode [3D5], RenCAD [3D4] GenCAD-3D [3D9], CAD-Coder [3D12], GenCAD [3D14], and Img2CAD [3D10] all perform full-parameter fine-tuning. In the TS domain, VisionTS++ [TS4] applies full continual pre-training of a vision backbone on the LOTSA corpus, while MLLM4TS [TS7] and TsLLM [TS8] fine-tune the entire LLM. Ansari et al. [TS13] performs full pre-training on time-

series data using a T5-style architecture. In the SE domain, Kapoor and Bhardwaj [SE4] and Arafah et al. [SE6] apply full model training on their respective tasks.

Zero-shot or no-training approaches constitute the second largest group, at 13 papers. This is particularly prevalent in papers that propose benchmarks rather than new models: OmniGIRL [SE10], HumanEval-V [SE13], DesignQA [SE7], MTBench [TS11], and LLM4CAD [3D3] all evaluate existing models without any fine-tuning. Massenon et al. [SE2] and Tzanettis et al. [SE9] also operate entirely through prompt-based inference and rule-based schema fusion respectively. The prevalence of zero-shot approaches reflects how capable modern foundation models have become out of the box, and positions zero-shot capability as the de facto baseline against which fine-tuned models must demonstrate improvement.

PEFT is represented in 6 papers. ChatTime [TS2] applies LoRA to adapt an LLM for unified time-series and text processing with minimal additional parameters. GPT4MTS [TS3] uses a selective freeze strategy, keeping attention and feed-forward layers fixed while fine-tuning only positional embeddings and layer normalizations, treating text embeddings as trainable soft prompts. GMM-TS [TS15] employs an MLP adapter as its only learned component. EGFE [SE8] freezes a pre-trained ResNet-50 and BERT encoder, training only downstream layers. VoT [TS14] and VisCodex [SE5] adopt partial tuning strategies, leaving large portions of the backbone frozen. The comparatively limited adoption of LoRA and adapters in the 3D and SE domains suggests that the more specialised representational requirements of point clouds, CAD sequences, and system logs may demand more substantial adaptation than PEFT typically affords.

Mixed or multi-stage strategies appear in 3 papers. Chai et al. [SE14] use a two-stage curriculum: full-parameter fine-tuning of all components in stage one, followed by LLM-only fine-tuning with the vision tower frozen in stage two. Yu et al. [3D9] train end-to-end on a synthetic balanced dataset, then apply full-parameter fine-tuning on a real distribution subset. Man et al. [3D1] combines zero-shot inference, training from scratch, and partial fine-tuning across three sub-systems. Six papers do not report their fine-tuning approach with sufficient clarity to allow categorization.

Multimodal Fusion Implementation

Q1.3 addresses how multimodal fusion is implemented, covering early fusion, late fusion, hybrid architectures, tool-based integration, and novel emerging mechanisms. Fusion strategy is the dimension along which the reviewed papers show the greatest architectural diversity. Figure 4 shows the verified breakdown across all 45 papers.

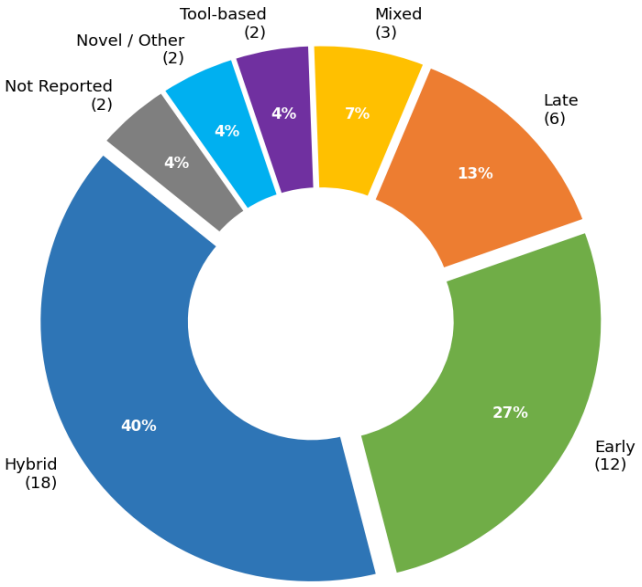

Figure 4: Distribution of fusion strategies across the 45 reviewed papers. Hybrid fusion is the most prevalent approach (18 papers); early fusion is the second most common (12 papers).

Hybrid fusion, which combines feature-level and decision-level integration in a single architecture, is the most common strategy, found in 18 papers. In the 3D domain, hybrid fusion typically involves a modality-specific encoder whose output tokens are cross-attended against language tokens within a transformer decoder. CAD-SIGNet [3D6] exemplifies this with layer-wise cross-attention between point-cloud embeddings and CAD language tokens, augmented by a sketch-instance guided attention module. GenCAD-3D [3D9] aligns modality-specific encoder outputs to a shared latent space through contrastive loss, then uses the aligned embedding as a conditioning signal to a latent diffusion model. In the TS domain, hybrid approaches are equally prevalent: TiMi [TS9] embeds a multimodal Mixture-of-Experts module inside a transformer backbone, routing computation through text-informed or series-aware expert pathways. ContexTST [TS10] decomposes time-series signals using frequency-domain analysis and anchors them with variable-level and global textual context through a context-informed Mixture-of-Experts (MoE) gating mechanism.

Early fusion, which involves concatenating or summing modality representations before a shared processing model, is the second most common strategy, found in 12 papers. In its simplest form, as in GPT4MTS [TS3] text embeddings are prepended as soft prompts to time-series patch embeddings before a frozen GPT-2 backbone. EGFE [SE8] sums five independent embeddings (vision, text, color, bounding box, and element class) into a single 256-dimensional token per UI element before a transformer encoder. In the 3D domain, CAD-Coder [3D12] and SldprtNet [3D11] fuse visual features from an image encoder with language tokens inside a vision-language transformer. MTBench [TS11] illustrates the simplest possible early fusion: time-series values are serialized as a text list and concatenated directly with the news or report text in the prompt context window.

Late fusion, which trains separate modality-specific models and combines their outputs at the decision level, appears in 6 papers. GMM-TS [TS15] uses a learnable gating architecture to aggregate predictions from a numerical time-series model and an LLM-based text model. Arafah et al. [SE6] apply CLIP-style late-fusion similarity scoring by comparing text and image embeddings and classifying based on the highest cosine similarity. Pisal et al. [SE11] use graph neural networks to integrate multi-source software data at the decision level. MultiCAD [3D2] aligns point-cloud and CAD-sequence embeddings into a shared space through contrastive learning, which is structurally a late-fusion alignment strategy.

Tool-based integration appears in 2 papers. LLM4CAD [3D3] uses GPT-4V to generate CAD code via prompting, with CAD execution handled by an external tool; the rendered image is fed back as a visual correction loop. Req2CAD [3D15] orchestrates multiple foundation models and external retrieval and execution systems through an agentic pipeline. Novel fusion mechanisms appear in 2 further papers: VisCodex [SE5] merges pre-trained vision-language and coding model parameters arithmetically using task vectors, avoiding any explicit fusion layer; ChatTime [TS2] achieves fusion by extending the LLM tokenizer vocabulary to include discretized time-series tokens, treating numerical and textual data as a single unified token stream.

Foundation Models and Backbone Architectures

Q1.4 asks what types of foundation models and backbone architectures are used. Figure 5 summarizes the distribution by domain, grouped into three categories: LLM/VLM-based backbones, hybrid architectures combining multiple model types, and custom or task-specific architectures.

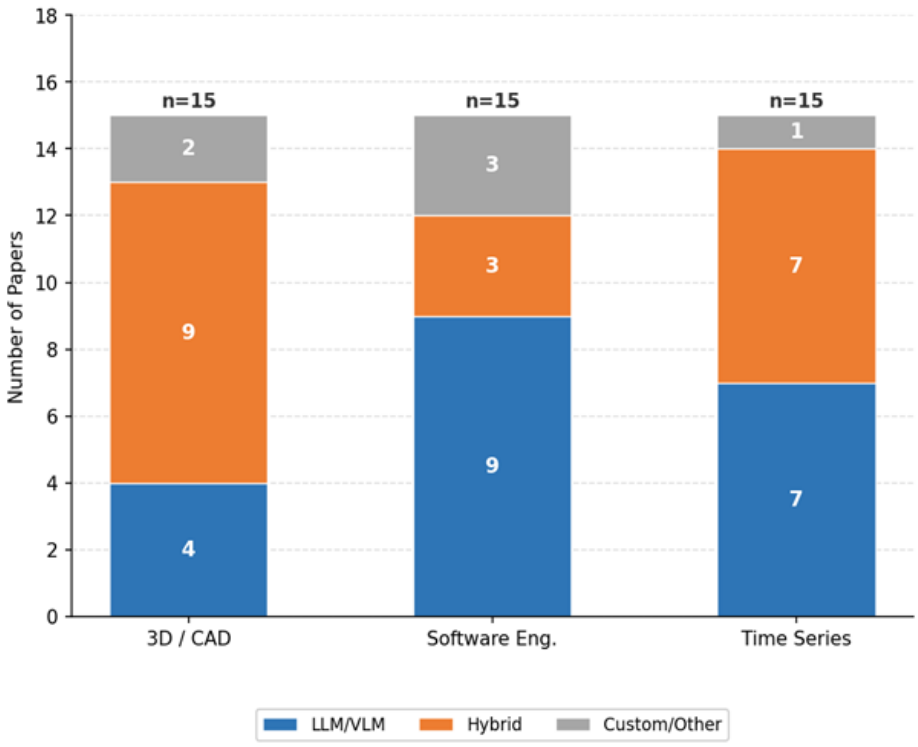

Figure 5: Backbone architecture types by domain. In 3D/CAD, hybrid architectures are most common (9 papers); in SE, LLM/VLM backbones dominate (9 papers); in TS, LLM/VLM and hybrid architectures are evenly split (7 each).

LLM/VLM-based backbones appear in 20 of the 45 papers. In the SE domain they account for 9 of the 15 papers, reflecting the natural affinity between language models and software artefacts. Models used include CLIP, Qwen2-VL-7B, GPT-4o, and various GPT-4 variants for zero-shot evaluation. In the 3D domain, 4 papers use a VLM backbone: GPT-4V, Qwen2.5-VL-7B, and a fine-tuned Llama 3.2 in Img2CAD [3D10]. In the TS domain, 7 papers rely on LLM or VLM backbones: GPT-2 as a frozen backbone, LLaMA-3, LLaVA-1.5 with CLIP, and LLM-based models for ChatTime [TS2], GMM-TS [TS15], and multimodal benchmark evaluation [SE10].

Hybrid architectures, which combine a specialized domain-specific model with one or more foundation model components, account for 19 papers. This category is dominant in the 3D domain (9 papers), where geometry-specific encoders such as DGCNN, PointNet++, or PVCNN are paired with transformer decoders or diffusion priors, as in GenCAD-3D [3D9], RenCAD [3D4], and CAD-SIGNet [3D6]. In the TS domain, 7 papers use hybrid architectures, typically pairing a time-series-specific processing component with a frozen or lightly adapted LLM: TiMi [TS9] embeds a text-conditioned MoE inside a transformer TS backbone; ContexTST [TS10] integrates LLM-generated context into a frequency-domain transformer; VoT [TS14] chains PatchTST with DeepSeek and LLaMA-3 components for event-driven reasoning. Aurora [TS5] builds an ensemble of separate time-series, text, and image foundation models aligned through distillation.

Custom or task-specific architectures account for 6 papers. In the 3D domain, MultiCAD [3D2] introduces a contrastive representation learning model based on PointNet++; TransCAD [3D7] is a custom hierarchical transformer for point-cloud-to-CAD inference. In the SE domain, Tzanettis et al. [SE9] operate through rule-based schema fusion with no learned backbone, Sehring et al. [SE12] proposes a conceptual framework without a concrete model, and Zhang, X., et al. [SE13] use a Siamese UniXcoder architecture. In the TS domain, Chronos [TS13] uses a T5-style transformer pre-trained entirely on time-series data, treated here as a custom TSFM since it operates on no natural language.

Across all three domains, the center of gravity is shifting away from task-specific architectures trained from scratch and towards the adaptation of large pre-trained foundation models. This shift

is most advanced in the SE domain, where the majority of recent papers leverage a pre-trained LLM or VLM. It is progressing in the 3D domain, where the emergence of capable vision-language models has enabled image-conditioned CAD generation that previously required bespoke architectures. In the TS domain the shift is underway but more contested, with purpose-built TS foundational model architectures continuing to compete with LLM-based adaptations on standard benchmarks, and the hybrid category (pairing domain-specific TS components with LLM reasoning) currently representing the most active architectural direction.

3.2. Multimodal Approaches for Software Engineering

This section relates to goal 2 of the review, as follows.

Goal 2: Assess Multimodal Approaches for CLEAR-Related Software Engineering Tasks
Goal 2 evaluates how multimodal methods are applied to software engineering tasks relevant to CLEAR and assess their effectiveness in comparison. *Research Questions* **Q2.1:** Which software engineering tasks are addressed by multimodal systems? Examples include code generation, debugging, UI generation, DevOps support, and requirements analysis. **Q2.2:** How is multimodality leveraged within these tasks? For example, combining logs and source code, UI representations and code, or telemetry with natural language descriptions. **Q2.3:** Do multimodal approaches outperform unimodal baselines? The review investigates whether multimodal integration provides measurable improvements in performance or task completion quality (if reported). **Q2.4:** What is the level of automation supported by current systems? This includes static pipelines, semi-autonomous workflows, and autonomous agent-based systems.
Metrics and Data Extraction • Software engineering task category • Performance improvements compared to baselines • Use of agents or autonomous components • Degree of automation

Q2.1: Software Engineering Tasks

Multimodal systems target a range of software engineering tasks, including code generation from visual artifacts; debugging, defect detection, and issue resolution; runtime monitoring, documentation, and UI prototyping; and code search, as shown in Figure 6.

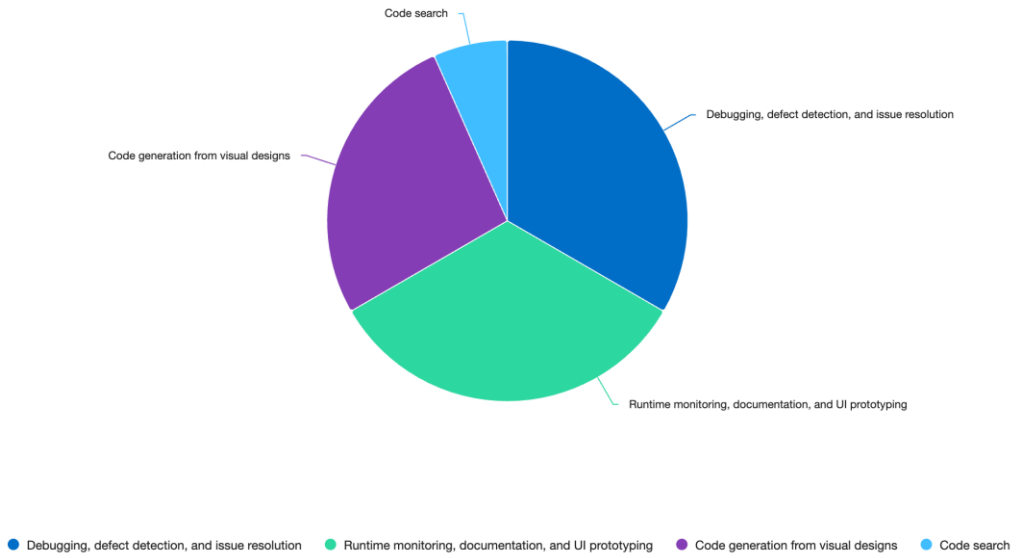

Figure 6: Distribution of software engineering tasks addressed by multimodal systems

Code generation from visual designs. Several systems generate source code from images combined with text. M²-CODER jointly uses textual instructions and software-design diagrams so the model can infer software structure, workflow, and implementation requirements [SE1]. VisCodex generates HTML, Python, and UI-to-code outputs from image and text inputs [SE5]. EGFE groups fragmented elements in UI design prototypes, improving downstream GUI-to-code quality [SE8]. HumanEval-V is a benchmark of 108 entry-level Python coding tasks in which the model must produce code from a visual context together with a predefined function signature [SE13].

Debugging, defect detection, and issue resolution. Quality-assurance tasks are covered at several stages of the bug lifecycle. BUGFixChecker verifies mobile-app bug fixes after deployment by comparing before/after UI screenshots against user reports, developer claims, and changelogs [SE2]. Kundu et al.'s defect-detection framework combines source code, design documents, execution logs, and test results across the software development lifecycle (SDLC) [S3]. For issue resolution, OmniGIRL provides a benchmark over GitHub issues that include screenshots and error images alongside text [SE20], and Pisal et al.'s bug-triaging pipeline operates over code, logs, traces, commits, and issue descriptions [SE11]. Arafah et al.'s fine-tuned CLIP classifier for incoming issue reports as bug or feature classification using text, code snippets, and attached images [SE6].

Runtime monitoring, documentation, and UI prototyping. A further strand covers software at runtime and engineering documentation. On the runtime side, Kapoor and Bhardwaj's performance-testing framework detects performance degradations from system metrics, logs, user interactions, and UI captures [SE4], while Tzanettis et al.'s observability framework combines metrics, logs, and distributed traces to support DevOps (running and maintaining software in production) for microservice deployments [SE9]. On the documentation side, DesignQA is a benchmark that asks MLLMs to answer questions about engineering documentation combining text, 2D images, and CAD data from the Formula SAE rulebook [SE7], and Petrovic et al.'s automotive case study summarizes UML and EMF diagrams in a model-based engineering toolchain [SE14]. Sehring's approach connects informal visual artifacts such as sketches, wireframes, and graphical design prototypes to formal model elements so the two remain consistent [SE12].

Code search. I2R aligns natural-language queries with source code by treating natural language and code as two distinct modalities and aligning them through contrastive training [SE15].

Q2.2 Leveraging of Modalities

Fusion in the reviewed work takes several forms: joint prompting of vision-language models, learned cross-modal embeddings, fusion of operational signals, and model-driven formalization. The machine-learning-based approaches combine modalities inside the model, while the model-driven approach formalizes informal artifacts through model elements without a learned model. Figure 7 shows the distribution of various modalities across the SE papers.

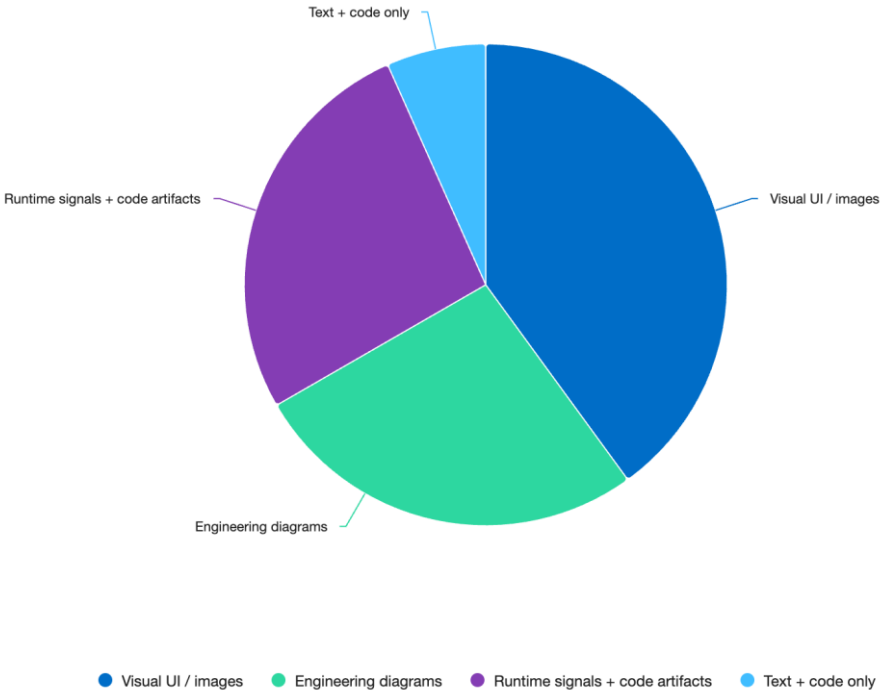

Figure 7:Distribution of input-modality types across SE papers

Joint prompting of vision-language models. A common approach combines images and text inside a single prompt to a vision-language model or MLLM and lets the model handle fusion internally. M²-CODER, for example, feeds design diagrams and instructions jointly to a Qwen2-VL backbone after two-stage supervised fine-tuning [SE1]. BUGFixChecker takes a similar joint-prompting route but enriches the visual side: screenshots are augmented with view-hierarchy parsing, optical character recognition (OCR), and bounding-box annotations alongside textual evidence, and GPT-4V then performs a comparative chain-of-thought comparison [SE2]. VisCodex extends the idea at training time, merging a vision-language model and a coding language model arithmetically by adding their task vectors so a single model inherits both skill sets while the vision encoder remains intact [SE5] DesignQA, by contrast, compares two prompting variants over text and CAD imagery: full document in context versus retrieval-augmented generation (RAG, the technique of retrieving relevant document chunks before generation) [SE7]. OmniGIRL and HumanEval-V follow the same prompt-based pattern, evaluating many MLLMs without task-specific fine-tuning [SE10, SE13], and Petrovic et al.'s case study likewise prompts MLLMs with diagram images and questions, guided by chain-of-thought, before deploying InternVL2-8B as a web service [SE14].

Learned cross-modal embeddings. Another approach trains encoders that bring different modalities into a shared vector space. Arafah et al.'s fine-tuned CLIP classifier, scoring how well an image matches a text caption, classifies issues by selecting the prompt with the highest cosine similarity to the screenshot [SE6]. In EGFE, five embeddings per UI element (ResNet image features, BERT text features, RGBA color, bounding-box coordinates, and an element-class label)

are summed into one 256-dimensional token and then passed through a six-layer Transformer [SE8]. In the non-visual setting, I2R trains a Siamese network on top of the UniXcoder code encoder, using contrastive learning to align natural-language and code representations and an additional consistency loss within each modality [SE15].

Fusion of runtime signals and code artifacts. A separate line of work fuses different kinds of engineering data instead of vision and text. Kapoor and Bhardwaj's framework evaluate three fusion modes over system metrics, log files, user interactions, and UI captures: early (feature concatenation), late (one model per modality, combined at the decision level), and hybrid [SE4]. Tzanettis et al.'s framework follows a tool-based route instead: metrics from Prometheus, logs from Fluentd, and traces from Zipkin are correlated through shared trace identifiers under a common JSON schema, with no learned model [SE9]. Kundu et al.'s framework routes each modality to a specialised network feeding a central defect-detection module: convolutional networks for UML imagery, graph neural networks (GNNs, which operate over graph-shaped data) for code and design graphs, and transformers for text [SE3]. Pisal et al.'s pipeline likewise uses a graph-based approach, integrating code components, microservices, logs, and traces through a GNN and using reinforcement learning (RL) to refine the bug-resolution loop from developer feedback [SE11].

Model-driven formalization. Sehring's approach is non-learning-based: informal visual artifacts (sketches, wireframes) are formalized by attaching formal model elements to them, and new artifacts are generated from formal descriptions, keeping the two consistent through traceability and model-checking [SE12].

Q2.3 Multimodal approaches vs. unimodal baselines

Among the reviewed SE work, direct comparisons between a multimodal system and a text-only or single-modal version of the same task consistently favor the multimodal approach. Comparisons among multimodal systems, by contrast, show wide gaps that track model scale, and several studies report no quantitative baseline at all.

Clear gains over unimodal baselines. Several systems show measurable improvements over unimodal baselines. BUGFixChecker, for example, reaches 0.830 accuracy and 0.805 macro F1 against a text-only baseline at F1 = 0.610, an improvement of about 19.5 F1 points from adding vision [SE2]. Kapoor and Bhardwaj's framework report a similar pattern, with 92% accuracy against 81% for the best single-modal baseline, detection roughly 15% faster, and false positives at about 7% versus 12-18% [SE4]. Arafah et al.'s fine-tuned CLIP classifier (ViT-B/32) gains 18.8 accuracy points and 19.6 F1 points over its zero-shot baseline [SE6]. EGFE improves precision, recall, and F1 by roughly 30 percentage points over the best baseline (UILM) at edit distance ≤ 4, and an ablation confirms that removing the image feature costs 18.93 macro-F1 points while removing the text description costs 7.54 [SE8]. Pisal et al.'s pipeline reports a reduction in mean time to resolution of more than 50%, a root-cause accuracy of 84.7%, and a test-case effectiveness of 83.2% against traditional machine-learning and graph-based baselines [SE11]. In the code-search setting, I2R improves Mean Reciprocal Rank (MRR, a standard retrieval metric) by 2.5 percentage points on the CSN benchmark and 0.8 on AdvTest over the underlying UniXcoder encoder it extends [SE15].

Comparisons among multimodal systems show wide gaps and dependence on model scale. When the comparison is between multimodal systems, the picture is mixed and tracks model scale closely. M²-CODER-7B reaches 25.3 average Pass@1 against 12.0 for Qwen2-VL-Instruct-7B and 24.7 for Qwen2.5-VL-Instruct-72B but stays below GPT-4o at 49.7 [SE1]. VisCodex shows a similar size effect: VisCodex-8B outperforms open-source models in the 7–15B parameter range and GPT-

4o-mini, while VisCodex-33B matches GPT-4o [SE5]. DesignQA reports GPT-4o supplied with the full ruleset at 0.881 F1 (bag-of-words on retrieval), against 0.185 for RAG-based variants and 0.082 for naive baselines, suggesting that the long-context variant outperforms the retrieval variant in this setting [SE7]. OmniGIRL shows that even the best model (GPT-4o) resolves only 8.6% of issues overall, and Claude-3.5-Sonnet leads on image-containing issues at 10.5%, suggesting that current LLMs are still weak on realistic GitHub issue resolution [SE10]. HumanEval-V points in the same direction: GPT-4o reaches 13% pass@1, 70B open-weight models stay below 4%, and multimodal training degrades coding ability relative to the same model's text-only counterpart [SE13].

No quantitative comparison. Some studies report no quantitative baseline. Kundu et al.'s framework is a proposal with no experimental results [SE3], while Tzanettis et al.'s framework reports only qualitative improvements in latency root-cause identification [SE9]. Sehring's approach similarly reports qualitative measures of coherence between informal designs and formal specifications [SE12]. Petrovic et al.'s case study scores five hand-written questions per model as correct, partially correct, or incorrect, with no aggregate metric []SE14.

Validation maturity also varies across the reviewed work, from industry deployment through lab benchmarks to early-stage proposals.

Q2.4 Level of automation

Automation in the reviewed systems ranges from reactive single-shot pipelines, through pipelines that depend on human input during operation, to workflows that act proactively. One pipeline combines a large language model with graph-based modelling and reinforcement learning into a self-enhancing loop that refines its strategy from developer feedback.

Reactive single-shot pipelines. Most systems run only when triggered, produce a single output, and have no human in the loop at runtime. BUGFixChecker, for instance, runs fully automated; human annotators are used only to build the reference dataset for evaluation, not to operate the system [SE2]. Petrovic et al.'s summarizer, similarly, is exposed as a web service to a toolchain, but it still answers incoming requests rather than initiating actions [SE14].

Reactive pipelines that depend on humans during operation. Some studies stay reactive but rely on human input during operation. EGFE is trained on labels produced by eighteen front-end developers and evaluated through a user study with ten senior developers [SE8]. Tzanettis et al.'s framework is rule-based and presents fused signals through Grafana dashboards for human interpretation [SE9]. In a more design-oriented setting, Sehring's approach frames the workflow around designers, test users, and developers who maintain coherence between informal artifacts and formal models [SE12].

Proactive workflows. Other systems act ahead of an explicit request. Kundu et al.'s framework is intended to act proactively, raising defect-likelihood flags across SDLC phases, though it remains a proposal without an implemented pipeline [SE3]. Kapoor and Bhardwaj's framework predict degradations and supports root-cause workflows that quality-assurance engineers act on [SE4]. Pisal et al.'s pipeline goes furthest in this direction: it combines an LLM with graph-based modelling and reinforcement learning into a self-enhancing loop covering triaging, root-cause analysis, and test-case synthesis, with developer feedback refining its strategy [SE11].

Across the reviewed work, most systems are reactive single-shot pipelines. Proactive behavior or human-in-the-loop operation appears in the remaining systems, with one pipeline additionally using reinforcement learning.

3.3. Multimodal approaches for timeseries

This sub-section presents the findings related to goal 3 of the review, as follows. Figure 8 visualizes an overview to this sub-section.

Goal 3: Analyze Time-Series Analysis in Multimodal Contexts
Goal 3 investigate how time-series data is integrated into multimodal learning systems, particularly within IoT and industrial contexts relevant to CLEAR. *Research Questions* **Q3.1:** What types of time-series models are used? Including native time-series foundation models and LLM-adapted time-series approaches. **Q3.2:** How is time-series data integrated with other modalities? Examples include combining time-series data with text, logs, or user interaction signals. **Q3.3:** Are generative time-series models used, and for what purpose? Such as forecasting, anomaly detection, simulation, or synthetic data generation. **Q3.4:** How does fine-tuning for time-series data differ from fine-tuning approaches used for other modalities? *Metrics and Data Extraction* • Time-series model type • Input-output modality combinations • Target task (forecasting, anomaly detection, control, etc.) • Fine-tuning approach • Dataset characteristics and domain

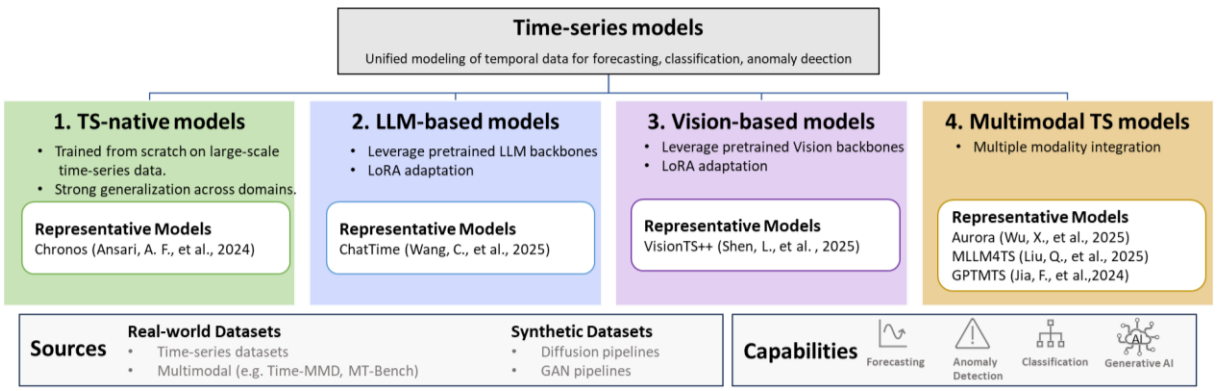

Figure 8: Time-series overview

Q3.1 Categorization of time-series models

Time series foundation models and related frameworks have emerged as a rapidly growing research direction for modelling univariate and multivariate temporal data, supporting a wide range of downstream tasks such as forecasting, imputation, classification, and anomaly detection. While

some models can be considered foundation level approaches with strong generalization capabilities across time-series tasks, others are better characterized as frameworks that leverage or adapt existing pretrained large models without performing full time-series pretraining.

Among the reviewed papers, we categorize time-series models into three primary groups: TS-native models, LLM-based models, and vision-based models. TS-native models are trained from scratch on large-scale time-series data and are either designed specifically for temporal representation or inherited from existing LLM architectures (e.g., [TS13]). LLM-based models leverage pretrained language backbones for time-series representation learning, reasoning, and cross-modal interaction (e.g., [TS2]). Meanwhile, vision-based models transfer knowledge from pretrained vision backbones to time-series representation learning (e.g., [TS4]).

TS-native models such as Chronos (based on the T5 family of language models) are designed for time-series forecasting via tokenized sequence modelling. They capture sequential dependencies, seasonality, and long-range correlations from data, enabling transferable forecasting capabilities without relying on external language-model pretraining.

ChatTime, as an example of LLM-based time-series models, enables pretrained LLM models (e.g., LLaMa) to process temporal data through vocabulary expansion, framing time series as a foreign language. The model employs LoRA to enable efficient adaptation across tasks. Reported results in the study suggest competitive zero-shot prediction accuracy compared to Chronos while using only 4% of the training data. On the other hand, VisionTS++, as an example of vision-based time-series models, converts multivariate time-series into color-coded image representations and processes them using a pretrained MAE-based vision backbone. The framework reformulates time-series forecasting as an image reconstruction task, enabling continual pretraining on large-scale time-series data and improving multivariate and probabilistic forecasting performance. VisionTS++ achieves a lower mean-absolute error than Chronos on LTSF benchmark, improving performance from 0.379 to 0.326, which corresponds to a ~14% relative reduction.

In addition, there exist multimodal and generative time-series frameworks that reformulate or integrate large-scale time-series models across different modality designs. Within this category, Aurora [TS5] is characterized as a generative multimodal time-series model that leverages large-scale pretraining across heterogeneous time-series domains to enable zero-shot forecasting with multimodal design elements. It incorporates separate modality encoders for time, text, and images, along with token-level alignment and distillation into a unified representation. In contrast, MLLM4TS [TS7] does not rely on time-series aligned data; instead, it reformulates multivariate time-series modelling using two modalities: time-series numerical inputs tokenized by a time-series tokenizer and color-coded line-plot images. It operates as a VLM-based adaptation framework that converts time-series data into visual representations (i.e., line plots) and processes them using pretrained multimodal backbones.

Q3.2 Time-series integration with other modalities

Time-series data can be reformulated into alternative representations, such as visual encodings or structured embeddings, and integrated with complementary modalities to enhance representation learning and downstream performance. Based on how information is exchanged across modalities, we categorize fusion strategies into three types: early fusion, where multimodal inputs are combined at the input or embedding level before model processing (e.g., GPT4MTS [TS3]); late fusion, where modality-specific models are independently processed and their outputs are aggregated at the decision level (e.g., GMM-TS [TS15]); and MoE-based fusion, where multiple modality-specific

experts are dynamically weighted via a learned gating mechanism to enable adaptive and context-dependent integration (e.g., MoME [TS12]).

Within the early fusion family, different designs vary in how tightly modalities are coupled during representation learning. ContextTST [TS10] introduces a context-aware Transformer for cross-domain time-series forecasting under the Unify and Anchor paradigm, where FFT-based decomposition constructs a unified frequency representation capturing shared cross-domain structures, while domain-specific contexts act as anchors that guide adaptation and constrain the prediction space through lightweight modulation. TsLLM [TS8] further extends early fusion by adopting an interleaved modeling strategy, where text and time-series tokens are jointly processed using a patch-based encoder-decoder architecture under a unified autoregressive framework. Similarly, GPT4MTS [TS3] employs a soft-prompting mechanism to jointly encode numerical and textual inputs within a pretrained GPT-2 decoder, enabling multimodal reasoning without modifying the backbone. VoT [TS14] incorporates textual event information as a conditioning signal and performs event-driven reasoning with multi-level alignment between textual and numerical representations, enabling improved forecasting under event-sensitive conditions.

In contrast, late fusion approaches decouple modality-specific processing before aggregation. GMM-TS [TS3] is representative of this paradigm by learning a Transformer-based gating network that dynamically weights and combines predictions from multiple uni-modal experts. This design supports flexible expert configurations, enables interpretable per-expert contributions, and achieves strong performance across diverse domains and forecasting settings. Finally, MoE-based fusion methods perform dynamic routing at the expert level to enable fine-grained multimodal integration. TiMi [TS9] incorporates LLM-guided semantic signals within an MoE-enhanced Transformer, while MoME [TS12] adopts a fully modular expert-based design that routes multimodal inputs through specialized experts via learned gating functions, improving adaptability and forecasting performance across heterogeneous settings.

Q3.3 Generative time-series models

Generative time-series models are increasingly used to model the underlying distribution of temporal data rather than producing single-point predictions. By learning probabilistic or reconstruction-based representations, these models support a range of downstream applications including forecasting, imputation, uncertainty-aware prediction, and synthetic data generation. Autoregressive and LLM-based models (e.g., Chronos and ChatTime) leverage large-scale pretraining to generate coherent future trajectories, while multimodal generative systems (e.g., Aurora) extend this capability by incorporating cross-modal signals such as text. In parallel, GAN-based approaches aim to synthesize realistic time-series data for domains (e.g., Multi-agent GAN pipeline [TS6]) with limited or privacy-sensitive data, such as healthcare and human sensing. Despite their effectiveness, generative time-series models often face challenges in preserving long-term temporal structure and ensuring downstream performance.

Q3.4 Real-world and synthetic multimodal time-series data

Multimodal time-series datasets and benchmarks can be broadly categorized into real-world and synthetic data settings. Real-world datasets, such as Time-MMD [TS9] and MTBench [TS11], are recent multimodal time-series resources that integrate additional modalities, such as contextual or textual information, to support forecasting, classification, and temporal reasoning tasks. Both datasets operate in cross-domain settings rather than single specific domains, enabling evaluation of generalization across heterogeneous temporal environments. Time-MMD focuses on heterogeneous real-world signals across multiple domains for robust representation learning. The dataset integrates numerical temporal signals with aligned textual or contextual information,

enabling the use of external knowledge beyond raw time-series values. In contrast, MTBench provides a benchmark for evaluating temporal reasoning and question-answering capabilities over time-series data.

However, domain-specific multimodal time-series datasets remain limited, which motivates the exploration of synthetic data generation approaches. In contrast, synthetic data settings are typically introduced through generative modelling approaches aimed at data augmentation, rare-event simulation, or robustness under distribution shift. A representative domain-specific example is a Multi-agent GAN pipeline [TS6] for multimodal time-series generation in human sensing and mobile health applications, which revisits multi-Agent GAN architectures to generate realistic and diverse synthetic multimodal sensor data through an adversarial multi-agent framework. However, empirical findings show that current multimodal GAN-based approaches often fail to produce high-quality synthetic time-series data, leading to significant degradation in downstream classification performance and indicating limitations in preserving task-relevant temporal structure.

3.4. Multimodal approaches for three-dimensional data

This sub-section relates to the goal 4 of our review, as follows.

Goal 4: Analyze Multimodal Learning for 3D and Structured Data Understanding
Goal 4 examine how multimodal systems process and integrate 3D and structured engineering data, including CAD models and point clouds. *Research Questions* **Q4.1:** What representations are used for 3D data? Examples include point clouds, meshes, and CAD graphs. **Q4.2:** How are 3D modalities fused with text or code? Including instruction tuning, embedding alignment, and cross-attention mechanisms. **Q4.3:** What application tasks are targeted? Examples include design validation, simulation, and instruction-following systems. **Q4.4:** What limitations exist in current multimodal fusion techniques for 3D data?
Metrics and Data Extraction • 3D representation type • Fusion mechanism • Task category • Evaluation metrics and reported performance

Q4.1 Modalities

Figure 9 visualize the distribution of modalities in papers with main focus on 3D data. Regarding input modalities, 4 of the considered papers consider text-based descriptions [3D1, 3D3, 3D13, 3D15] possibly accompanied by supporting sketches, images or even a CAD or point cloud representation. Most papers (7; [3D2, 3D4, 3D5, 3D6, 3D7, 3D8, 3D9]), consider a point cloud as input representation. Three papers [3D10, 3D12, 3D14] consider colored or grayscale images of rendered CAD objects or real images, possibly incorporating sketches or a natural language-like semantic structure. The remaining paper [3D11] considers multi-view rendered images, parametric CAD modelling scripts (as structured text), and 3D CAD files.

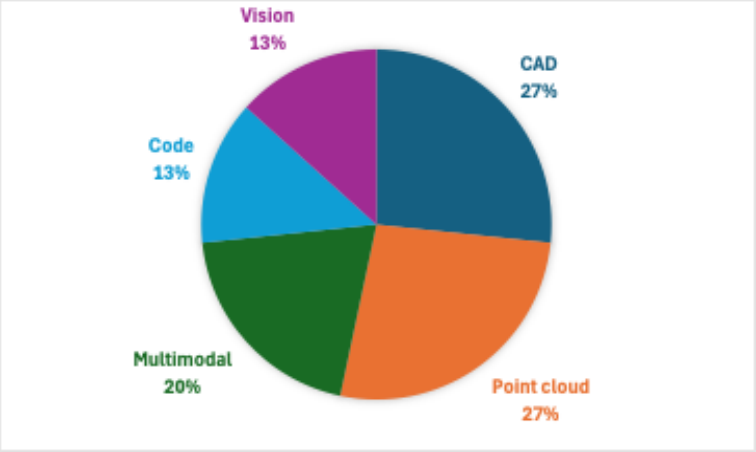

Figure 9. Primary modality for 3D data

When considering the primary modality, 4 papers are classified use CAD [3D4, 3D5, 3D9, 3D15], 4 papers works with point cloud [3D2, 3D6, 3D7, 3D8], 3 papers work with multimodal (combining images, 3D representations, code and text) [3D1, 3D3, 3D11], and two paper use code scripts [3D13, 3D14]. Finally, two papers primarily use vision (hence relying primarily on images and sketches) [3D10, 3D12].

Regarding output modalities, we observe that all papers generate CAD-based artifacts of different nature. These can be CAD models, CAD sequences and CAD programs and scripts. Among these, most (9; [3D3, 3D4, 3D5, 3D6, 3D7, 3D9, 3D10, 3D13, 3D14]) focus on one such type of artifact. The other papers (6) provide either CAD sequences or CAD programs and scripts, combined with CAD models. In 2 cases [3D1, 3D11], text-based descriptions and images are provided along the aforementioned artifacts.

Q4.2 Data fusion

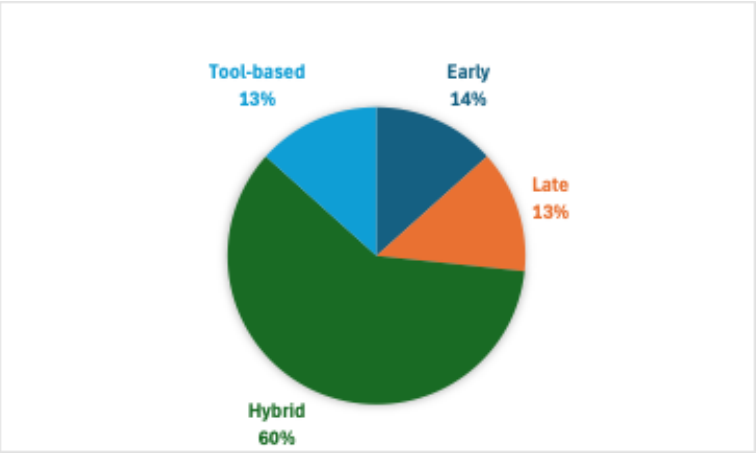

Figure 10. Data fusion for 3D data

Figure 10 visualize the different fusion approaches used in the analyzed papers working with 3D data. Early fusion, taking place when input modalities are combined before they are applied to a model, is considered in 2 papers [3D11, 3D12]. In both cases, visual features are combined with text-based features via a multimodal, vision-language, model. Late fusion, taking place when input modalities are combined at later stages, after having been (initially) processed by a model, is also considered in 2 papers [3D2, 3D13]. One paper [3D2] uses contrastive learning to align embeddings across different modalities into a shared representation space. Another paper [3D13] creates CAD scripting code from textual descriptions and images. The textual descriptions are given to a vision-language model to generate questions for subsequent validation, and CAD code, which is

subsequently executed to render a 3D object. Both the image and the questions are fed back to the VLM, which answers the questions and corrects the code based on this visual feedback.

In most cases (9 papers), input modalities are combined through both early and late fusion mechanisms, depending on the underlying task. A common pattern involves encoding each modality with a dedicated encoder and aligning the resulting representations through contrastive learning. Further, a generative module such as a diffusion model or autoregressive decoder conditions on the aligned embeddings to produce the final output. Cross-attention between point cloud embeddings and CAD sequence tokens within transformer layers is another recurring mechanism, enabling geometric and parametric information to interact progressively during decoding. In a few cases, hybrid fusion arises because the overall framework encompasses multiple sub-systems, each employing a different fusion strategy for its specific sub-task.

Finally, two papers [3D3, 3D15] fully rely on external tools and APIs for handling input modalities. For instance, LLMs such as GPT are used to combine text and (possibly) image data and generate CAD code; CADQuery[1] is used to execute the CAD code and render a 3D object. Note that CADQuery is used also in some papers with late and hybrid fusion strategies (e.g. [3D5]) for executing CAD code.

Q4.3 Targeted tasks

Figure 11. Targeted tasks for 3D data

Figure 11 shows the targeted 3D task for the analyzed papers related to the 3D domain. Many papers (8; [3D2, 3D4, 3D5, 3D6, 3D7, 3D8, 3D9, 3D10]) concentrate on reconstructing (or reverse engineering) a CAD representation of an object from a point cloud, a 3D scan, geometric inputs or images. The second most covered task (6 papers; [3D3, 3D11, 3D12, 3D13, 3D14, 3D15]) is the generation of a CAD representation from a textual description or instructions, sketches and images. Some papers (e.g., [3D12, 3D13]) concentrate more specifically on solid geometry reconstruction, on syntactically valid CAD code synthesis and on iterative visual verification. One paper [3D1] covers multiple tasks: 3D mesh generation from sketches, CAD model construction from videos, and CAD code generation from point clouds and images.

[1] https://github.com/CadQuery/cadquery

Q4.4 Limitations

Several different types of limitations are identified in the reviewed papers. We list in the following those appearing across two of more papers:

- Tasks under scope: several works (7; [3D1, 3D2, 3D4, 3D5, 3D6, 3D9, 3D14]) focus on a restricted set of (relatively simpler) operations, such as sketch-extrude operations, and do not address (and may not generalize to) more complex operations required in real-world designs, such as fillets, sweeps, lofts, revolutions, and chamfers.
- Evaluation datasets: many papers (7; [3D1, 3D3, 3D4, 3D5, 3D6, 3D7, 3D9]) evaluate results against small-scale synthetic datasets or benchmarks, rather on real-world industrial data. It is therefore unclear how the proposed methods would perform on real-world data including noise, missing parts, or objects outside the training distribution.
- Performance: greatly as a result of the previous point, the performance of several papers (5; [3D5, 3D6, 3D10, 3D12, 3D14]) depend on the evaluation datasets used and may degrade as complexity increases and challenges present in real-world industrial data become tougher.
- No editing capabilities: some results (4; [3D1, 3D2, 3D3, 3D14]) lack editing capabilities; they generate a model but offer limited to no control to further adapt that model or incorporate corrections.
- Evaluation metrics: for two papers [3D1, 3D13], limitations at the level of the evaluation metrics are identified. For one, metrics such as Intersection over Union (IoU) and Valid Syntax Rate (VSR) fail to capture design intent or editability. For the other, point cloud and Hausdorff distance are deemed to be noisy metrics that do not capture logical structural errors such as gaps between parts. This might indicate a not-met-yet need for new metrics able to capture and reason upon the specific challenges related to 3D modelling.
- Lack of repeatability: in 2 cases [3D1, 3D13], it is identified that results are very sensitive to the specific prompt used and that using the same prompt might yield different results. Such a lack of reproducibility might have an impact on reliability if not taken care of properly.

3.5. Open-Source Multimodal Large Language Models

As discussed, the rapidly evolving nature of multimodal learning paradigms makes the field fragmented over grey literature and press releases in addition to the scientific publications. This sub-section is related to goal 5 of the study that is focused on identifying and analyzing open-source Multimodal large language models (MLLMs) with a cursory survey.

Goal 5: Analyze Open-Source Multimodal Foundation Models and LLM Ecosystems
Goal 5 investigate the current landscape of open-source multimodal large language models and foundation models, with a focus on their architectures, multimodal capabilities, and limitations. Open-source multimodal LLMs are increasingly important for research reproducibility, customization, transparency, and deployment in privacy-sensitive or domain-specific environments (such as CLEAR's industrial partners' setup). In addition, the analyzed literature might miss mentioning or using these open-source models. Therefore, this goal is achieved with a dedicated cursory survey that searches and analyze a variety of open-source multi-modal LLMs. *Research Questions* **Q5.1:** How has the architecture of open-source multimodal large language models evolved? How has this evolution impacted the models? Examples include modular bridge, and native unified architecture types.

Goal 5: Analyze Open-Source Multimodal Foundation Models and LLM Ecosystems
Q5.2: What modalities do multimodal LLMs support? How do they shape real-world use cases? Examples include bi-modal such as text and image, and omni-modal types such as text, image, audio, video, speech.
Metrics and Data Extraction • Evaluation progress • Architecture types • Supported modalities • Key features and primary use cases

MLLMs are transforming artificial intelligence into a system capable of perceiving the real world and interacting with it, thanks to their ability to interpret text, images, audio, and video together. This study provides a comprehensive overview of the applications of open-source multimodal LLMs.

Open-source multimodal LLMs distinguish themselves from traditional LLMs by processing different data types, such as text, images, audio, and video, simultaneously [MLLM1]. These models stand out for their ability to interpret multiple data types with a single model, as well as their support for long context, robust visual interpretation, multi-model integration, and omni-modal interactions [MLLM2]. They are actively used in various domains, including intelligent document analysis (such as financial report analysis and legal contract review), visual understanding and interpretation (such as quality control, medical image analysis, and camera analysis), video analysis and understanding (such as sports position analysis), multimodal assistants, AI agent systems, and many others.

According to a report by Grand View Research[2], the global multimodal AI market is projected to grow at a compound annual growth rate (CAGR) of 36.8%.

According to Zebracat's research[3], multimodal AI solutions show a distinct distribution in terms of both technology segmentation and use cases. On the technology side, vision-based systems are the most common segment at 35%, while natural language processing (NLP) solutions integrated with visual data have reached a 28% share, particularly in customer service. It has been observed that such combined approaches provide 25% higher user engagement compared to single-modal systems. Systems that use both audio and visual are utilized at an 18% rate in security and surveillance, while this sector accounts for a 10% share of general use cases. Sensor fusion applications, prominent in industrial automation, account for 12%, while the use of multimodal AI in robotic systems has reached 14% in production and healthcare support processes. In the retail sector, usage stands at 15% in areas such as inventory management and personalized marketing. In contrast, gesture recognition technologies have a more limited scope of application, accounting for 7% specifically in gaming and virtual reality. Overall, the adoption rate of vision-based solutions is approximately twice that of audio-based systems.

This cursory survey examines the architectures, capabilities, performance metrics, development visions, practical applications in networks such as the IoT and 6G, and the public datasets used by open-source MLLM solutions. The findings of this cursory study can be summarized as follows:

- We present an analysis and comparison of xx open-source LLM models.

[2] Grand View Research. Multimodal ai market (2025 - 2030). Accessed: 30-03-2026. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/multimodal-artificial-intelligence-ai-market-report

[3] Michael Baumgartner. 50+ multimodal ai market size insights and growth projection. Accessed: 30-03-2026. [Online]. Available: https://www.zebracat.ai/post/multimodal-ai-market

- We examine and compare the applications and use cases of these models in the studies where they are used.

Models & Highlighted Features

This sub-section will examine the architectures, performance metrics, and practical applications of open-source multimodal language models. This will enable us to present the capabilities, development visions, limitations, and real-world use cases of prominent architectures in the literature within an objective framework.

With the emergence of open-source architectures as robust alternatives to closed architectures, the open-source ecosystem can be divided into legacy modular bridge architectures and advanced native unified architectures in Table 4.1.1. The methodology focuses on the transition from modular bridges to native unified and the integration of these models into critical infrastructure in compliance with legal requirements. The terms “Bi-Modal” and “Omni-Modal” describe the evolution of LLM models from processing a single type of input to understanding and generating content across multiple integrated modes. Sparse MoE [MLLM3] is a machine learning architecture that reduces the computational cost of LLMs while enhancing their capacity and performance. This is achieved by processing each input only through the relevant expert network, rather than running the entire model for every input.

Table 4.1.1: Comparison of Open-Source Multimodal LLMs

Model Name	Year	Architecture Type	MoE Status	Supported Modalities	Key Feature / Integration	Primary Use Case
LLaVA, Series	2023	Modular (Bridge)	No	Text, Image	Cost-effective visual alignment	Healthcare and sensor analysis
LLaVA-OneVision Series	2024	Modular (Bridge)	No	Text, Image, Video	Cross-layer injection to close modality gap	High-res spatial reasoning, medical analysis
InternVL Series	2023	Modular (Bridge)	Yes	Text, Image, (Video)	Dynamic high-resolution architecture	Advanced visual R&D
GLM-V series	2025	Modular (Bridge)	Yes	Text, Image, Video	Autonomous computer control simulation	Computer control & GUI navigation
Qwen-VL series	2023	Modular (Bridge)	No	Text, Image, Video	High-resolution visual parsing capabilities	GUI navigation, document parsing, coding
Qwen-Omni Series	2025	Native (Unified)	No	Text, Image, Video, Audio, Speech	End-to-end simultaneous processing	Real-time audio-visual assistants
Mistral Family (Pixtral & Mistral 3.x)	2024	Native (Unified)	Yes	Text, Image	Energy efficiency, scalable RL pipeline	Edge deployment, mobile IoT application
Gemma Family	2024	Hybrid	Yes	Text, Image, Video	Advanced reasoning &	AI agents, edge AI

(PaliGemma, Gemma 3, Gemma 4)					agentic workflows	
EuroLLM Series	2024	Native (Unified)	No	Text (Multilingual)	24 EU languages and 11 additional languages	Cultural and linguistic autonomy
Transfuser	2022	CNN & Transformer	No	Camera, LIDAR	Feature-level spatial data fusion	Autonomous driving, collision avoidance

The Large Language and Vision Assistant (**LLaVA**), released in 2023 [MLLM4], the next-generation **LLaVA 1.5**, released in the same year, and **LLaVA-NEXT (LLaVA 1.6),** released in 2024 [MLLM4] are open-source, end-to-end trained multimodal AI models capable of understanding and processing both visual and text data. As the versions advance, image resolution, reasoning capabilities, speed, and efficiency continue to improve.

The LLaVA series has continued with **LLaVA-OneVision** [MLLM5] in 2024 and **LLaVA-OneVision-1.5** [MLLM6] in 2025, which support omni-modal models that demonstrate improved performance in understanding complex diagrams, tables, and graphical data.

The **InternVL** series consists of vision-language models developed to integrate visual and linguistic understanding. The first **InternVL 1.0** [MLLM7] model in the series was released in 2023, and the latest model is **InternVL-U** [MLLM8], which was released in early 2026. The model's core feature is the vision encoder known as InternViT-6B. Its primary applications include image and video classification, image and video-text retrieval, and multimodal dialogue systems.

The **GLM-V series (GLM-4.1V-Thinking, GLM-4.5V, and GLM-4.6V)** [MLLM9] consists of multimodal vision-language (VLM) models that support text, image, and video modalities, featuring a Mixture-of-Experts-based architecture with a total of 106 billion parameters and 12 billion active parameters. These models can mimic human-like mouse and keyboard movements and can be used for tasks such as STEM problem-solving, video understanding, content recognition, coding, GUI-based agents, and long-document summarization.

The **Qwen-VL series** (**Qwen-VL, Qwen-VL-Chat, Qwen2-VL, Qwen2.5-VL, Qwen3-VL**) [MLLM10] consists of large-scale open-source vision-language models that support text, images, and video; feature a vision encoder for images and a modular architecture; and possess capabilities for multimodal reasoning, document analysis, and image-based dialogue.

The **Qwen-Omni series** (**Qwen-2.5-Omni, Qwen-3-Omni, Qwen-3.5-Omni**) [MLLM11, MLLM12] is a family of multimodal large language models (LLMs) that adopts an "omni-modal" approach, capable of processing text, images, audio, and video from end to end within a single model rather than separate modules. Unlike the Qwen-VL series, it supports not only vision-language but also audio, speech, and video, and is capable of performing perception, reasoning, and generation within a unified system.

The **Mistral**[4] AI team first began their work on multimodal LLMs in 2024 with the **Pixtral** [MLLM13] model which had 12 billion parameters and a modular bridge architecture, but they have since discontinued this model. Subsequently, they continued with the new-generation **Mistral 3 series**, which is closer to a native unified architecture. As part of this effort, the visual language-focused **Mistral Large 3, Mistral Medium 3.1 / 3.5, Mistral Small 3.x, and Ministral** models [MLLM14] have been released.

Within the **Gemma family**, released as open source by Google DeepMind, the **PaliGemma** [MLLM15] models are multimodal large language models that integrate visual encoders with a

[4] M. A. Team. Mistral models. Accessed: 21-05-2026. [Online]. Available: https://huggingface.co/mistralai

language model in a modular architecture. Additionally, while **Gemma 3** [MLLM16] offers lightweight multimodal capabilities via a modular vision-language extension, **Gemma 4**[5] is steering the family toward agent-based and hybrid MoE-based multimodal architectures with enhanced context length, greater modularity, and reasoning capabilities.

EuroLLM [MLLM17] is a series of open-source, multilingual large-scale language models developed to represent European languages in linguistic learning ecosystems. The models have been trained on large-scale multilingual datasets covering all 24 official EU languages and 11 additional global languages. While EuroLLM is not yet an MLLM, it is evolving toward multimodal expansion, and work is underway on **EuroVLM** [MLLM18].

Although **TransFuser** [MLLM19] is not an LLM, it is a multimodal model focused on the autonomous driving domain that supports camera, and LiDAR modalities.

3.6. Key Challenges and Open Problems

This sub-section relates to goal 6 of our SoTA study that focuses on identifying limitations, gaps and research opportunities in the field.

Goal 6: Identify Gaps, Limitations, and Research Opportunities
Goal 6 synthesizes the limitations of current multimodal systems and identify underexplored research directions relevant to CLEAR. *Research Questions* **Q6.1:** Where do current multimodal approaches fail? Examples include temporal reasoning, cross-modal alignment, scalability, and robustness. **Q6.2:** Which modality combinations or application areas remain underexplored? **Q6.3:** What evaluation and benchmarking gaps exist in the literature? **Q6.4:** What limitations are reported in existing multimodal systems? *Metrics and Data Extraction* • Reported limitations and recurring themes • Missing modality combinations • Evaluation weaknesses • Maturity level of the proposed systems (prototype, benchmarked, or production-ready)

The reviewed literature shows clear progress in multimodal AI, but most systems still perform best in controlled settings. Across 3D/CAD, software engineering, and time-series studies, the main gaps are not only technical accuracy, but also reliability, context awareness, scalability, and readiness for real industrial use.

Q 6.1 Where do current multimodal approaches fail?

Current multimodal systems mainly fail when they must combine heterogeneous signals under realistic conditions. The most common weaknesses are cross-modal alignment, temporal reasoning, robustness, interpretability, uncertainty handling, and scalability.

- **Cross-modal alignment and grounding:** Systems often struggle to map text, images, video, audio, point clouds, meshes, CAD code, and time-series data into a shared semantic representation. This can lead to shallow correlations rather than genuine multimodal

[5] G. A. for Developers. Gemma 4 model overview. Accessed:21-05-2026. [Online]. Available: https://ai.google.dev/gemma/docs/core

understanding. In the reviewed papers, CAD systems struggled to preserve design intent across text, image, point cloud, mesh, and CAD-code representations [3D1, 3D13, 3D14], while time-series systems depended strongly on the quality and availability of textual signals [TS9, TS12, TS14].

- **Temporal reasoning and synchronization:** Many approaches are weak at handling time lags, irregular sampling, variable-length inputs, and long-horizon causal relationships. Time-series systems often assume fixed-length, regularly sampled data [TS10], and some benchmarks cover only limited combinations such as univariate text and structured time-series inputs [TS11].
- **Robustness to missing or degraded modalities:** Performance often drops when one modality is noisy, incomplete, or unavailable. Many systems assume clean, synchronized inputs and lack fallback mechanisms. CAD systems also produced invalid, incomplete, non-repeatable, or non-manufacturable outputs in several cases [3D1, 3D7, 3D12, 3D13, 3D14].
- **Uncertainty and interpretability:** Fusion mechanisms often operate as black boxes. Users cannot always identify which modality influenced a decision, and models rarely treat inputs according to their reliability. This is problematic for high-stakes or industrial settings, where decisions must be explainable and auditable.
- **Scalability and computational cost:** Several systems require high GPU resources, expensive preprocessing, complex synchronization, or large curated datasets. Examples include multimodal software testing pipelines [SE4], vision-augmented time-series models [TS2, TS7], text-scraping pipelines [TS3], and point-cloud CAD methods that scale poorly with large inputs [3D6].

Q 6.2 Which modality combinations or application areas remain underexplored?

The literature is concentrated around three clusters: 3D/CAD, software engineering, and time-series forecasting or analysis. Important industrial combinations remain weakly represented.

- **Time-series + vision + language:** Few systems jointly reason over rich visual evidence, textual context, and temporal sensor streams. This gap is important for monitoring, diagnostics, and operational decision-making.
- **3D/CAD + time-series + operational data:** Most CAD work focuses on geometry, sketches, meshes, B-Reps, or CAD code, but rarely connects design representations with live telemetry, maintenance records, or time-dependent industrial behavior.
- **Geospatial/satellite + temporal data + language:** Remote sensing, geospatial data, and satellite imagery are not strongly covered in the reviewed set. One paper [TS11] explicitly notes that satellite imagery, graphs, and tabular data are excluded.
- **Human-in-the-loop multimodal systems:** Most systems are reactive and non-interactive. Explicit human feedback, correction, or validation is limited, although some papers include partial human involvement [3D1, 3D15, SE7, SE8, SE12].
- **Industrial deployment areas:** Transportation, agriculture, manufacturing operations, telecommunications, infrastructure monitoring, and emergency response remain underexplored compared with benchmark-oriented research tasks.
- **Q 6.3 What evaluation and benchmarking gaps exist?**
- Evaluation practices are a major bottleneck. Many reported results are difficult to compare and may not reflect real deployment performance.
- **Synthetic and simplified data:** CAD studies often rely on synthetic CAD programs, synthetic point clouds, or simplified sketch-extrude workflows [3D1, 3D6, 3D9, 3D14]. Such settings may not transfer to noisy scans, complex assemblies, or real industrial workflows.
- **Small or narrow benchmarks:** Several evaluations in the SE domain are limited in scale or scope, such as the 53 cases used in one paper [SE2], 108 entry-level Python tasks [SE13], one UML class diagram and five questions [SE14], 15 repositories [SE10], or one Formula rule document [SE7].

- **Fragmented benchmark landscape:** Many papers introduce custom datasets or benchmarks such as CADPrompt, OmniGIRL, HumanEval-V, MTBench, or domain-specific CAD/UI datasets. This limits fair comparison across approaches.
- **Metrics that miss practical value:** Traditional metrics such as IoU, VSR, Hausdorff distance, point-cloud distance, syntax accuracy, or reconstruction similarity do not fully capture design intent, editability, manufacturability, usability, safety, trustworthiness, or logical structural correctness [3D1, 3D13].
- **Limited real-world validation:** Most systems are validated in lab or benchmark settings. Only one entry in the reviewed sheet is explicitly marked as industry validated [SE8], so production readiness remains largely unproven.

Q 6.4 What limitations are reported in existing multimodal systems?

The reported limitations are consistent across the reviewed literature. The most frequent issues are:

- limited modality and task coverage, with many systems focused on narrow combinations such as point cloud + CAD, image + CAD, text + time series, or UI + code;
- synthetic or benchmark-only validation, with limited evidence from noisy real-world environments;
- poor generalization to complex inputs, missing modalities, rare structures, noisy text, or industrial-scale data;
- weak cross-modal grounding and alignment, especially when modalities are incomplete or only loosely related;
- insufficient evaluation metrics, which often measure accuracy or reconstruction quality but not usefulness, trust, safety, or decision quality;
- high computational cost and dependence on expensive pipelines, proprietary APIs, or specialized models;
- invalid, incomplete, inconsistent, or hallucinated outputs; and
- limited human-in-the-loop support and very little evidence of production-ready deployment.

Overall synthesis and research directions

Overall, the reviewed systems are mostly at the prototype or benchmarked-research stage rather than production-ready maturity. They demonstrate promising capabilities, but their reliability under real-world, multimodal, and industrial conditions remains limited.

Future research should therefore prioritize context-aware and uncertainty-aware multimodal systems that can handle noisy, missing, and asynchronous inputs; combine time-series, visual, textual, geospatial, and operational data; include human verification and feedback; and use evaluation frameworks that measure robustness, interpretability, practical utility, and readiness for deployment.

4. Threats and Limitations

As common for review studies, several threats to validity may affect the reliability, completeness, and interpretation of the findings and conclusions. In our review, we analyze threats related to conclusion validity, internal validity, construct validity, and external validity, and discuss the measures taken to mitigate their potential impact, following the guidelines proposed by Runeson and Höst [12].

Conclusion validity concerns the extent to which meaningful and reliable relationships can be established between the reviewed evidence and the conclusions drawn from the study. One potential threat to conclusion validity in our review is the use of Google Labs as the primary search platform for retrieving relevant studies. Although Google Labs provides AI-assisted semantic retrieval capabilities and broad access to both academic and grey literature through Google Scholar

indexing, the ranking and retrieval mechanisms may introduce bias by prioritizing certain publications over others. Further, some relevant studies may not have been retrieved or ranked highly enough for selection. Another potential threat relates to bias during the data extraction and interpretation process. Since the extraction of qualitative and open-ended information inherently involves human judgment and differences in interpretation among reviewers may affect the consistency of the extracted data and metrics.

To mitigate these threats, several measures were adopted. First, despite using Google Labs for retrieval, a predefined inclusion and exclusion criteria was systematically applied to all candidate studies. This helped ensure that study selection was driven by the review objectives rather than solely by AI-generated relevance rankings. Second, the data extraction process was distributed across multiple CLEAR partners, allowing different perspectives and expertise to contribute to the review. The involvement of multiple reviewers helped reduce individual researcher bias and encouraged cross-validation of extracted information. Furthermore, a standardized data extraction template based on the review goals was used to improve consistency and comparability across the reviewed studies.

Internal validity refers to the extent to which the study execution support reliable relationships between the collected evidence and the resulting findings. In the context of this review, a potential threat to internal validity is the rapidly evolving nature of multimodal AI technologies. Many state-of-the-art multimodal LLMs and tools are released through open-source platforms, technical blogs, or industrial announcements before appearing in peer-reviewed scientific literature. Relying exclusively on academic databases could therefore result in an incomplete representation of the current technological landscape.

To strengthen internal validity, the review explicitly incorporated both peer-reviewed literature and grey literature sources. In addition, a dedicated cursory survey of open-source multimodal LLMs and publicly available frameworks was conducted alongside the literature review. This complementary approach helped capture emerging technologies, practical implementations, and industrial developments that may not yet be represented in traditional scientific publications.

Construct validity concerns the extent to which the selected measures, indicators, and reviewed studies accurately represent the concepts being investigated. A possible threat to construct validity in this study is the exclusion of potentially relevant studies due to the limited scope of the review and the selected search prompts. Since the review focused specifically on multimodal approaches relevant to CLEAR application domains, some adjacent research areas or alternative terminologies may not have been fully captured.

To address these concerns, the search strategy incorporated multiple application domains and different types of modalities relevant to CLEAR. Furthermore, the selected studies collectively provide substantial breadth and depth across the identified research areas, enabling a representative overview of the current SoTA within the scope of this deliverable.

External validity concerns the degree to which the findings of the review can be generalized beyond the analyzed studies and applied to broader contexts. In review studies, external validity can closely relate to the representativeness and relevance of the selected literature with respect to the review objectives. One potential limitation to external validity is that the study selection was intentionally focused on application domains relevant to the CLEAR project i.e., multimodal approaches for 3D data, software engineering tasks, and time-series data. As a result, the findings may not fully generalize to all multimodal AI domains or application scenarios outside the CLEAR project scope. Further, the rapidly evolving nature of multimodal AI and LLMs means that new

models may emerge shortly after the review. Therefore, some findings may become outdated over time as the field continues to evolve.

Despite these limitations, the review includes a diverse collection of peer-reviewed studies, grey literature sources, and open-source multimodal LLMs covering multiple modalities, architectures, and application domains. This diversity improves the overall representativeness of the analyzed SoTA and supports the broader applicability of the findings to multimodal AI research and practice.

5. Discussion and Conclusion

This deliverable investigated the current SoTA in multimodal learning and fine-tuning approaches across three application domains relevant to the CLEAR project: software engineering, time-series analysis and 3D data. Through the analysis of 45 primary studies and a complementary survey of open-source multimodal large language models the review provides a consolidated overview of current multimodal architectures, modality combinations, fusion mechanisms, base models, finetuning strategies, and emerging research directions.

The findings show that across all three domains, there is a clear transition from narrowly specialized architectures toward the adaptation and orchestration of large pretrained multimodal foundation models. Vision-language models and LLM-based architectures are becoming central components for multimodal reasoning, generation, and decision support. This trend is most mature in software engineering applications, increasingly visible in 3D/CAD systems, and actively emerging in time-series analysis through hybrid architectures that combine domain-specific processing with LLM reasoning capabilities.

The findings also highlights that multimodal systems differ significantly across application domains in terms of modality combinations, fusion mechanisms, and output representations. While image and text combinations emerge as the only modality pair consistently present across all domains, each application area shows other distinct modality profiles and domain-specific architectural requirements. Hybrid fusion architectures are the dominant integration strategy overall, highlighting the need to balance feature-level and decision-level reasoning when handling heterogeneous modality input.

From a training perspective, full fine-tuning remains the most adopted strategy, particularly in domains requiring specialized geometric or temporal representations. At the same time, the growing use of zero-shot inference and parameter-efficient fine-tuning demonstrates the increasing capability of foundation models to generalize across tasks with minimal adaptation. Nevertheless, the relatively limited adoption of PEFT strategies in 3D and software engineering domains suggests that current multimodal foundation models still struggle to fully capture highly specialized industrial representations without substantial retraining.

Despite the substantial progress demonstrated in the reviewed literature, the findings also reveal that most multimodal systems remain at the prototype or benchmark-oriented research stage rather than being production-ready solutions. Across domains, the most significant limitations relate to robustness, cross-modal grounding, temporal reasoning, interpretability, scalability, and handling of noisy or incomplete modalities. Many systems continue to rely on clean, synchronized, and highly curated datasets, while real-world industrial environments typically involve heterogeneous, asynchronous, uncertain, and partially missing information sources.

The findings further reveal important research and evaluation gaps. Current benchmark ecosystems remain fragmented, with many studies introducing domain-specific datasets and evaluation protocols that hinder reproducibility and fair comparison across approaches. Existing evaluation metrics frequently focus on technical reconstruction or prediction accuracy while overlooking practical relevance, unitality, trustworthiness, interpretability, and operational decision quality.

Furthermore, very limited evidence of large-scale industrial deployment or long-term operational validation was identified across the SoTA.

Several underexplored modality combinations and application areas were also identified. In particular, the integration of time-series, visual, textual, geospatial, and operational data remains insufficiently investigated despite its importance for industrial monitoring, diagnostics, predictive maintenance, and decision support. Similarly, human-in-the-loop multimodal systems remain relatively rare, with most current approaches relying on static inference pipelines rather than interactive and collaborative AI workflows.

Overall, the findings suggest that future research should move beyond benchmark-centric performance optimization toward robust, context-aware, explainable, and deployment-oriented multimodal AI systems. Future multimodal pipelines should be capable of reasoning across heterogeneous and imperfect industrial data sources, handling inconsistent and missing modalities, incorporating human feedback and validation, and providing transparent and explainable outputs suitable for high-stakes operational environments.

For the CLEAR project, the review provides important insights into the capabilities, limitations, and maturity of current multimodal AI approaches. The identified architectural trends, finetuning strategies, and multimodal reasoning approaches provide a strong foundation for the development of CLEAR-specific multimodal pipelines. At the same time, the identified research gaps and limitations highlight important opportunities for innovation, particularly in domain-specific multimodal reasoning, trustworthy AI, industrial scalability, and multimodal decision support across heterogeneous engineering and operational data.

References

[1] Robillard, P. N. (1999). The role of knowledge in software development. *Communications of the ACM*, *42*(1), 87-92.

[2] Turk, M. (2014). Multimodal interaction: A review. *Pattern recognition letters*, *36*, 189-195.

[3] Liang, P. P., Zadeh, A., & Morency, L. P. (2024). Foundations & trends in multimodal machine learning: Principles, challenges, and open questions. *ACM computing surveys*, *56*(10), 1-42.

[4] Boumedra, I., & Adeshola, I. (2025). The Emergence of Multimodal AI in Software Development: What to Expect in the Future?.

[5] Huang, D., Yan, C., Li, Q., & Peng, X. (2024). From large language models to large multimodal models: A literature review. *Applied Sciences*, *14*(12), 5068.

[6] Han, Z., Gao, C., Liu, J., Zhang, J., & Zhang, S. Q. (2024). Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

[7] Zhao, W., Gangaraju, K., & Yuan, F. (2025). Multimodal perception-driven decision-making for human-robot interaction: a survey. *Frontiers in Robotics and AI*, *12*, 1604472.

[8] Song, B., Zhou, R., & Ahmed, F. (2024). Multi-modal machine learning in engineering design: A review and future directions. *Journal of Computing and Information Science in Engineering*, *24*(1), 010801.

[9] Jiang, Y., Ning, K., Pan, Z., Shen, X., Ni, J., Yu, W., ... & Song, D. (2025, August). Multi-modal time series analysis: A tutorial and survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2* (pp. 6043-6053).

[10] Arksey, H., & O'malley, L. (2005). Scoping studies: towards a methodological framework. *International journal of social research methodology*, *8*(1), 19-32.

[11] Hutchinson, S. R. (2003). Survey research. In *Foundations for research* (pp. 299-318). Routledge.

[12] Runeson, P., & Höst, M. (2009). Guidelines for conducting and reporting case study research in software engineering. *Empirical software engineering*, *14*(2), 131-164.

[3D1] Man, K. Y. B. (2025). *From sketch to CAD code: Multimodal AI for controllable design generation* [Master's thesis, MIT]. MIT DSpace. https://dspace.mit.edu/handle/1721.1/165165

[3D2] Ma, W., Xu, M., Li, X., & Zhou, X. (2023). *MultiCAD: Contrastive representation learning for multi-modal 3D computer-aided design models*. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. https://doi.org/10.1145/3583780.3614982

[3D3] Li, X., Sun, Y., & Sha, Z. (2025). *LLM4CAD: Multimodal large language models for three-dimensional computer-aided design generation*. Journal of Computing and Information Science in Engineering. https://doi.org/10.1115/1.4067085

[3D4] Lu, J., Wang, Y., Wu, Y., Li, H., Shi, Y., & Ning, F. (2026). *An autoregressive framework for reconstructing editable parametric computer-aided design models from point clouds*. Engineering Applications of Artificial Intelligence. https://doi.org/10.1016/j.engappai.2025.113107

[3D5] Rukhovich, D., Dupont, E., & Mallis, D. (2025). *CAD-Recode: Reverse engineering CAD code from point clouds*. Proceedings of ICCV 2025. https://openaccess.thecvf.com/content/ICCV2025/html/Rukhovich_CAD-Recode_Reverse_Engineering_CAD_Code_from_Point_Clouds_ICCV_2025_paper.html

[3D6] Khan, M. S., Dupont, E., & Ali, S. A. (2024). *CAD-SIGNet: CAD language inference from point clouds using layer-wise sketch instance guided attention*. Proceedings of CVPR 2024. https://openaccess.thecvf.com/content/CVPR2024/html/Khan_CAD-SIGNet_CAD_Language_Inference_from_Point_Clouds_using_Layer-wise_Sketch_CVPR_2024_paper.html

[3D7] Dupont, E., Cherenkova, K., Mallis, D., & Gusev, G. (2024). *TransCAD: A hierarchical transformer for CAD sequence inference from point clouds*. Proceedings of ECCV 2024. https://link.springer.com/chapter/10.1007/978-3-031-73030-6_2

[3D8] Mallis, D., Aziz, A. S., Dupont, E., et al. (2023). *SHARP Challenge 2023: Solving CAD history and parameters recovery from point clouds and 3D scans*. Proceedings of ICCVW 2023. https://openaccess.thecvf.com/content/ICCV2023W/SHARP/html/Mallis_SHARP_Challenge_2023_Solving_CAD_History_and_pArameters_Recovery_from_ICCVW_2023_paper.html

[3D9] Yu, N., Ferdous Alam, M., & Hart, A. J. (2026). *GenCAD-3D: Computer-aided design program generation using multimodal latent space alignment and synthetic dataset balancing*. Journal of Mechanical Design. https://doi.org/10.1115/1.4069276

[3D10] You, Y., Uy, M. A., Han, J., Thomas, R., & Zhang, H. (2025). *Img2CAD: Reverse engineering 3D CAD models from images through VLM-assisted conditional factorization*. Proceedings of the ACM SIGGRAPH 2025 Conference. https://doi.org/10.1145/3757377.3763891

[3D11] Li, R., Li, S., Mu, Y., & Ding, M. (2026). *SldprtNet: A large-scale multimodal dataset for CAD generation in language-driven 3D design*. arXiv preprint arXiv:2603.13098. https://arxiv.org/pdf/2603.13098

[3D12] Doris, A. C., & Alam, F. (2026). *CAD-Coder: An open-source vision-language model for computer-aided design code generation*. Journal of Mechanical Design, 148(7), 071702. https://doi.org/10.1115/md-25-1707

[3D13] Alrashedy, K., Tambwekar, P., & Zaidi, Z. (2025). *Generating CAD code with vision-language models for 3D designs*. arXiv preprint arXiv:2410.05340. https://arxiv.org/pdf/2410.05340

[3D14] Alam, M. F., & Ahmed, F. (2024). *GenCAD: Image-conditioned computer-aided design generation with transformer-based contrastive representation and diffusion priors*. arXiv preprint arXiv:2409.16294. https://arxiv.org/pdf/2409.16294

[3D15] Jing, Q., Lu, H., Huang, S., Childs, P., & Chen, L. (2026). *Req2CAD: Bridging functional requirements and parametric CAD models to support conceptual 3D design*. Proceedings of the ACM/IEEE International Conference on Human Factors in Computing Systems. https://dl.acm.org/doi/pdf/10.1145/3772318.3791949

[SE1] Chai, L., Yang, J., Liu, S., Zhang, W., Wang, L., & Jin, K. (2025). *Multilingual multimodal software developer for code generation*. arXiv preprint arXiv:2507.08719. https://arxiv.org/pdf/2507.08719

[SE2] Massenon, R., Gambo, I., & Khan, J. A. (2025). *Toward an automated cross-multimodal verification of mobile app bug fixes integrating user feedback, developer responses, changelogs, and*

UI visual analysis. Information and Software Technology, 191. https://doi.org/10.1016/j.infsof.2025.107996

[SE3] Kundu, S., Mishra, D., & Mishra, A. (2025). *A proposal on an AI-based framework for software defect detection using multimodality in software industries*. Proceedings of the ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM 2025). https://ieeexplore.ieee.org/abstract/document/11323337

[SE4] Kapoor, S., & Bhardwaj, S. (2025). *Multi-modal learning for robust and reliable performance testing*. Proceedings of the IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC 2025). https://ieeexplore.ieee.org/abstract/document/10903838

[SE5] Jiang, L., Huang, S., Wu, X., Li, Y., & Zhang, D. (2025). *VisCodex: Unified multimodal code generation via merging vision and coding models*. arXiv preprint arXiv:2508.09945. https://doi.org/10.48550/arXiv.2508.09945

[SE6] Arafah, Y., Shejin, S., & Akhter, S. (2025). *Multimodal classification of software issue reports using CLIP: A fine-tuned approach for bug and feature detection*. Proceedings of the International Conference on Electrical, Computer and Communication Engineering (ECCE 2025). https://ieeexplore.ieee.org/abstract/document/11012920

[SE7] Doris, A. C., Grandi, D., & Tomich, R. (2025). *DesignQA: A multimodal benchmark for evaluating large language models' understanding of engineering documentation*. Journal of Computing and Information Science in Engineering, 25(2), 021009. https://doi.org/10.1115/jcise_25_2_021009

[SE8] Chen, L., Chen, Y., Xiao, S., Song, Y., & Sun, L. (2024). *EGFE: End-to-end grouping of fragmented elements in UI designs with multimodal learning*. Proceedings of the 46th International Conference on Software Engineering (ICSE 2024). https://dl.acm.org/doi/abs/10.1145/3597503.3623313

[SE9] Tzanettis, I., Androna, C. M., & Zafeiropoulos, A. (2022). *Data fusion of observability signals for assisting orchestration of distributed applications*. Sensors, 22(5), 2061. https://www.mdpi.com/1424-8220/22/5/2061

[SE10] Guo, L., Tao, W., Jiang, R., Wang, Y., Chen, J., & Liu, X. (2025). *OmniGIRL: A multilingual and multimodal benchmark for GitHub issue resolution*. ACM Transactions on Software Engineering and Methodology. https://doi.org/10.1145/3728871

[SE11] Pisal, P., Jalan, P., & Chigurupati, S. R. (2025). *Integrating LLMs for automated bug triaging and root cause localization in software systems*. Proceedings of the 3rd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings 2025). https://ieeexplore.ieee.org/abstract/document/11296163

[SE12] Sehring, H. W. (2024). *Visual artifacts in software engineering processes*. Proceedings of The Sixteenth International Conference on Creative Content Technologies (CONTENT 2024). https://personales.upv.es/thinkmind/dl/conferences/content/content_2024/content_2024_1_10_60002.pdf

[SE13] Zhang, F., Wu, L., Lin, G., Li, X., Yu, X., Wang, Y., & Chen, B. (2024a). *HumanEval-V: Evaluating visual understanding and reasoning abilities of large multimodal models through coding tasks*. arXiv preprint arXiv:2410.12381. https://openreview.net/forum?id=KRdiRGSNc9

[SE14] Petrovic, N., Zhang, Y., Maaroufi, M., & Chao, K. Y. (2025). *Multi-modal summarization in model-based engineering: Automotive software development case study*. arXiv preprint arXiv:2503.04506. https://link.springer.com/chapter/10.1007/978-3-032-00071-2_10

[SE15] Zhang, X., Xiang, Y., Liu, Z., & Hu, X. (2024b). *I2R: Intra and inter-modal representation learning for code search*. Intelligent Data Analysis, 28(3). https://doi.org/10.3233/IDA-230082

[TS1] Liu, H., Xu, S., Zhao, Z., & Kong, L. (2024). *Time-MMD: Multi-domain multimodal dataset for time series analysis*. Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024). https://doi.org/10.52202/079017-2476

[TS2] Wang, C., Qi, Q., Wang, J., Sun, H., & Zhuang, Z. (2025). *ChatTime: A unified multimodal time series foundation model bridging numerical and textual data*. Proceedings of the AAAI Conference on Artificial Intelligence, 39(12). https://ojs.aaai.org/index.php/AAAI/article/view/33384

[TS3] Jia, F., Wang, K., Zheng, Y., Cao, D., & Liu, Y. (2024). *GPT4MTS: Prompt-based large language model for multimodal time-series forecasting*. Proceedings of the AAAI Conference on Artificial Intelligence, 38(21). https://ojs.aaai.org/index.php/AAAI/article/view/30383

[TS4] Shen, L., Chen, M., Liu, X., Fu, H., Ren, X., Sun, J., & Li, Z. (2025). *VisionTS++: Cross-modal time series foundation model with continual pre-trained vision backbones*. arXiv preprint arXiv:2508.04379. https://arxiv.org/pdf/2508.04379

[TS5] Wu, X., Jin, J., Qiu, W., Chen, P., Shu, Y., & Yang, B. (2026). Aurora: Towards universal generative multimodal time series forecasting. Proceedings of ICLR 2026. https://arxiv.org/pdf/2509.22295

[TS6] Di Martino, F., Pal, S. S., & Delmastro, F. (2025). Revisiting multi-agent GAN for multimodal time series generation in human sensing and mHealth applications. Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing. https://dl.acm.org/doi/abs/10.1145/3714394.3756189

[TS7] Liu, Q., Heshmati, S., Mai, Z., & Abraham, Z. (2025). MLLM4TS: Leveraging vision and multimodal language models for general time-series analysis. arXiv preprint arXiv:2510.07513. https://arxiv.org/abs/2510.07513

[TS8] Parker, F., Chan, N., Zhang, C., & Ghobadi, K. (2025). Augmenting LLMs for general time series understanding and prediction. arXiv preprint arXiv:2510.01111. https://arxiv.org/abs/2510.01111

[TS9] Lin, J., Wang, Y., Luo, H., Pei, Z., & Wang, J. (2026). TiMi: Empower time series transformers with multimodal mixture of experts. arXiv preprint arXiv:2602.21693. https://arxiv.org/pdf/2602.21693

[TS10] Hong, X., Zhang, J., Li, W., Lu, S., & Li, J. (2025). Unify and anchor: A context-aware transformer for cross-domain time series forecasting. arXiv preprint arXiv:2503.01157. https://arxiv.org/pdf/2503.01157

[TS11] Chen, J., Feng, A., Zhao, Z., Garza, J., & Nurbek, G. (2025). MTBench: A multimodal time series benchmark for temporal reasoning and question answering. arXiv preprint arXiv:2503.16858. https://arxiv.org/abs/2503.16858

[TS12] Zhang, L., Maatouk, A., Chen, J., & Tassiulas, L. (2026). Multi-modal time series prediction via mixture of modulated experts. arXiv preprint arXiv:2601.21547. https://arxiv.org/pdf/2601.21547

[TS13] Ansari, A. F., Stella, L., Turkmen, C., & Zhang, X. (2024). Chronos: Learning the language of time series. Transactions on Machine Learning Research. https://arxiv.org/pdf/2403.07815

[TS14] Wang, S., Chen, P., Wang, Y., Qiu, W., & Guo, C. (2026). Unlocking the value of text: Event-driven reasoning and multi-level alignment for time series forecasting. arXiv preprint arXiv:2603.15452. https://arxiv.org/pdf/2603.15452

[TS15] Razmadze, K., & Shavit, Y. (2025). GMM-TS: Gating architecture for multi-modal time series forecasting. OpenReview. https://openreview.net/pdf?id=NS22e2Vgnv

[MLLM1] 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, (2024). Available: https://arxiv.org/abs/2307.06435

[MLLM2] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, A survey on multimodal large language models, National Science Review, vol. 11, no. 12, Nov. (2024). Available: http://dx.doi.org/10.1093/nsr/nwae403

[MLLM3] N. Shazeer, A. Mirhoseini, K. Maziarz, A. Davis, Q. V. Le, G. E. Hinton, and J. Dean, (2017) Outrageously large neural networks: The sparsely-gated mixture-of-experts layer," CoRR, vol. abs/1701.06538,. Available: http://arxiv.org/abs/1701.06538

[MLLM4] Liu, C. Li, YuhengLi, and Y. J. Lee, (2024) Improved baselines with visual instruction tuning. Available: https://arxiv.org/abs/2310.03744v2

[MLLM5] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, Y. Li, Z. Liu, and C. Li, (2024) Llava-onevision: Easy visual task transfer, arXiv preprint arXiv:2408.03326.

[MLLM6] X. An, Y. Xie, K. Yang, W. Zhang, X. Zhao, Z. Cheng, Y. Wang, S. Xu, C. Chen, D. Zhu, C. Wu, H. Tan, C. Li, J. Yang, J. Yu, X. Wang, B. Qin, Y. Wang, Z. Yan, Z. Feng, Z. Liu, B. Li, and J. Deng, (2025) Llava-onevision-1.5: Fully open framework for democratized multimodal training,. Available: https://arxiv.org/abs/2509.23661

[MLLM7] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai, “Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks,” 2024. [Online]. Available: https://arxiv.org/abs/2312.14238

[MLLM8] C. Tian, D. Yang, G. Chen, E. Cui, Z. Wang, Y. Duan, P. Yin, S. Chen, G. Yang, M. Liu, Z. Zhu, Z. Fan, L. Gu, H. Wang, Q. Wei, J. Yin, X. Yang, Z. Zhong, Q. Qin, Y. Xin, B. Fu, Y. Liu, J. Ge, Q. Guo, G. Luo, H. Li, Y. Qiao, K. Chen, and H. Zhang, “Internvl-u: Democratizing unified multimodal models for understanding, reasoning, generation and editing,” 2026. [Online]. Available: https://arxiv.org/abs/2603.09877

[MLLM9] Team et al., “Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning,” 2026. [Online]. Available: https://arxiv.org/abs/2507.01006

[MLLM10] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, “Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond,” 2023. [Online]. Available: https://arxiv.org/abs/2308.12966

[MLLM11] J. Xu, Z. Guo, J. He, H. Hu, T. He, S. Bai, K. Chen, J. Wang, Y. Fan, K. Dang, B. Zhang, X. Wang, Y. Chu, and J. Lin, “Qwen2.5-omni technical report,” 2025. [Online]. Available: https://arxiv.org/abs/2503.20215

[MLLM12] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He, Y. Wang, X. Shi, T. He, X. Zhu, Y. Lv, Y. Wang, D. Guo, H. Wang, L. Ma, P. Zhang, X. Zhang, H. Hao, Z. Guo, B. Yang, B. Zhang, Z. Ma, X. Wei, S. Bai, K. Chen, X. Liu, P. Wang, M. Yang, D. Liu, X. Ren, B. Zheng, R. Men, F. Zhou, B. Yu, J. Yang, L. Yu, J. Zhou, and J. Lin, “Qwen3-omni technical report,” 2025. [Online]. Available: https://arxiv.org/abs/2509.17765

[MLLM13] Agrawal et.al, “Pixtral 12b,” 2024. [Online]. Available: https://arxiv.org/abs/2410.07073

[MLLM14] Liu et. al, “Ministral 3,” 2026. [Online]. Available: https://arxiv.org/abs/2601.08584

[MLLM15] Beyer et. al, “Paligemma: A versatile 3b vlm for transfer,” 2024. [Online]. Available: https://arxiv.org/abs/2407.07726

[MLLM16] G. Team, “Gemma 3 technical report,” 2025. [Online]. Available: https://arxiv.org/abs/2503.19786

[MLLM17] M. M. Ramos, D. M. Alves, H. Gisserot-Boukhlef, J. Alves, P. H. Martins, P. Fernandes, J. Pombal, N. M. Guerreiro, R. Rei, N. Boizard, A. Farajian, M. Klimaszewski, J. G. C. de Souza, B. Haddow, F. Yvon, P. Colombo, A. Birch, and A. F. T. Martins, “Eurollm-22b: Technical report,” 2026. [Online]. Available: https://arxiv.org/abs/2602.05879

[MLLM18] G Unbabel, Instituto Superior T´ecnico. Eurovlm-1.7b-preview. Accessed: 21-05-2026. [Online]. Available: https://huggingface.co/utter-project/EuroVLM-1.7B-Preview

[MLLM19] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, “Transfuser: Imitation with transformer-based sensor fusion for autonomous driving,” 2022. [Online]. Available: https://arxiv.org/abs/2205.15997

