HUMAN-CENTRIC AI-ENABLED EXTENDED REALITY APPLICATIONS FOR THE INDUSTRY 5.0 ERA

D4.1 – ADVANCED AI PARADIGMS FOR HUMAN-AI COLLABORATION V1

Lead Beneficiary	ATB
Work Package Ref.	WP4 – AI/XR Symbiosis for Augmented Intelligence
Task Ref.	T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration T4.2 – XR-Enabled Human-AI Collaboration (Neuro-Symbolic AI, Active Learning) T4.3 – XR-Enabled Generative AI
Deliverable Title	D4.1 – Advanced AI Paradigms for Human-AI Collaboration v1
Due Date	2025-01-31
Delivered Date	2025-01-29
Revision Number	3.0
Dissemination Level	Public (PU)
Type	Report (R)
Document Status	Release
Review Status	Internally Reviewed and Quality Assurance Reviewed
Document Acceptance	WP Leader Accepted and Coordinator Accepted
EC Project Officer	Ms. Marta PALAU FRANCO

HORIZON-CL4-2023 Research and Innovation Action

This project has received funding from the European Union's Horizon Europe Research and Innovation Programme under grant agreement no 101135209.

DISCLAIMER

COPYRIGHT MESSAGE

CONTRIBUTING PARTNERS

Partner Acronym	Role[1]	Name Surname[2]
ATB	Lead Beneficiary	Fulya Horozal, Sebastian Scholze
GFT	Internal Reviewer	Matteo Frattini
HOLO	Internal Reviewer	Harsh Manoj Shah, Leesa Joyce
INNOV	Contributor	George Fatouros
UPRC	Contributor	Stergios Chairistanidis
UPRC	Contributor	Dimitrios Dardanis
UPRC	Contributor	George Makridis
UPRC	Quality Assurance	Thanos Kiourtis, Argyro Mavrogiorgou

REVISION HISTORY

Version	Date	Partner(s)	Description
0.1	2024-10-09	ATB, UPRC, INNOV	ToC Version
0.2	2024-12-10	ATB	Inputs
0.3	2024-12-24	UPRC	Inputs
0.4	2024-12-25	INNOV	Inputs
0.5	2025-01-02	UPRC	Inputs
0.6	2025-01-13	ATB, UPRC	Updates
0.7	2025-01-15	INNOV	Updates
1.0	2025-01-16	ATB, INNOV, UPRC	1st Version
1.1	2025-01-17	ATB	Version for Peer Reviews
1.2	2025-01-24	ATB, INNOV, UPRC	Addressed review comments
2.0	2025-01-27	ATB	Version for Quality Assurance
2.1	2025-01-28	UPRC	Quality Assurance revision
2.2	2025-01-29	ATB	Addressed Quality Assurance comments
3.0	2025-01-29	ATB	Version for Submission

[1] Lead Beneficiary, Contributor, Internal Reviewer, Quality Assurance
[2] Can be left void

LIST OF ABBREVIATIONS

Acronym	Definition
AI	Artificial Intelligence
AL	Active Learning
AR	Augmented Reality
BERT	Bidirectional Encoder Representations from Transformers
CCTV	Closed-Circuit Television
CNN	Convolutional Neural Network
DL	Deep Learning
DoA	Description of Action
GenAI	Generative AI
GPT-4	Generative Pre-trained Transformer 4
Grad-CAM	Gradient-weighted Class Activation Mapping
HITL-ML	Human-in-the-loop Machine Learning
LIME	Local Interpretable Model-agnostic Explanations
LlaMA	Large Language Model Meta AI
LLM	Large Language Model
LNN	Logical Neural Network
LTN	Logical Tensor Network
ML	Machine Learning
MPNet	Masked and Permuted Pre-training for Language Understanding
MT	Machine Teaching
NN	Neural network
NSAI	Neurosymbolic Artificial Intelligence
OCR	Optical Character Recognition
RAG	Retrieval Augmented Generation
SHAP	Shapley Additive Explanations
STT	Speech-to-Text
TRL	Technology Readiness Level
VMLC	Visual Machine Learning Control
VR	Virtual Reality
VSA	Vector Symbolic Architecture
XAI	Explainable AI
XR	Extended Reality

EXECUTIVE SUMMARY

Deliverable D4.1, titled *Advanced AI Paradigms for Human-AI Collaboration v1*, is the first deliverable within WP4, *AI/XR Symbiosis for Augmented Intelligence*, which seeks to establish novel synergies between AI and XR technologies for enhanced human-AI collaboration in Industry 5.0 applications. The deliverable documents the initial progress of the development, implementation, and evaluation of innovative AI methodologies aimed at enhancing effective and seamless human-AI collaboration in XR5.0. It outlines the foundational AI frameworks, their initial pilot application scenarios and further refinements designed to address the objectives in WP4.

This deliverable synthesizes efforts across three tasks, Task T4.1 – T4.3, focusing on explainable AI, Neurosymbolic AI, Active Learning and Generative AI models integration into the XR platforms of the project. The approach emphasizes a user-centric and ethical design philosophy, combining technical innovation with the development of tools and infrastructures that enable transparent decision-making, more efficient learning processes and dynamic content creation tailored to the industrial pilots of the project, while ensuring trust, adaptability, and seamless integration of AI technologies into XR environments.

These advancements are aligned with Industry 5.0 goals of human-centric, robust, sustainable and resilient AI systems that are responsive to human input and supported by iterative development cycles and collaborative feedback from project stakeholders.

Deliverable D4.1 serves as a baseline for the refinements and advancements to be detailed in Deliverable D4.2, *Advanced AI Paradigms for Human-AI Collaboration v2*, which represents the next iteration of D4.1.

TABLE OF CONTENTS

List of Figures

List of Tables

1. INTRODUCTION

1.1 Objectives of the Deliverable

The primary objective of D4.1, *Advanced AI Paradigms for Human-AI Collaboration v1*, is to present the progress and advancements achieved in tasks

- T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration,
- T4.2 – XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning),
- T4.3 – XR-Enabled Generative AI

within WP4, *AI/XR Symbiosis for Augmented Intelligence*, of the XR5.0 project. This deliverable consolidates the efforts and outcomes achieved during Months 1–13 (M1–M13) of the project and highlights the advancements in AI methodologies and XR integration for augmented intelligence, including the initial implementation of these innovations in the project pilots. The objectives specific to each task are outlined as follows:

Task T4.1 – AI Models and Infrastructures for Trusted Human-AI Collaboration

T4.1 focuses on the integration of trustworthy AI models into XR solutions for Industry 5.0 applications, ensuring transparent, understandable and user-adaptive AI systems that foster effective human-AI collaboration. Main goals include:

- Developing human-centred explainable AI (XAI) models tailored for industrial end-users that provide personalized, trusted recommendations and ensure ethical compliance and trustworthiness through responsible AI design.
- Incorporating counterfactual methods for "what-if" analysis.
- Building tools and infrastructures for efficient training, inference, and integration of XAI models into XR platforms within XR5.0.

Task T4.2 – XR-Enabled Human-AI Collaboration (Neuro-Symbolic AI, Active Learning)

T4.2 focuses on the development and integration of Neurosymbolic AI (NSAI) and Active Learning (AL) models into XR environments to enhance human-AI collaboration within Industry 5.0 applications, improving accuracy, efficiency and explainability of AI systems. Main goals include:

- Developing Neurosymbolic and Active Learning AI models for XR-enabled human-AI collaboration
- Integrating Neurosymbolic AI into XR-based industrial applications, combining deep learning and symbolic reasoning to create more robust and explainable AI models.
- Integrating XR-enabled active learning techniques that involve humans in the machine learning process for better and faster results with XR interfaces to enable a natural interaction between humans and the AI systems.

Task T4.3 – XR-Enabled Generative-AI

T4.3 focuses on the development and integration of Generative AI models into XR environments to facilitate improved collaboration and enhanced creativity between humans and AI systems. Main goals include:

- Leveraging Generative AI to create novel, context-aware content and solutions for more efficient and effective collaboration in various industrial scenarios.
- Developing solutions based on state-of-the-art generative models, retrieval-augmented generation, agentic workflows and prompt engineering to enable real-time assistance based on proprietary data (e.g., technical manuals) to human users in I5.0 contexts via user-friendly interfaces for (e.g., chat, voice to voice).

1.2 Relation to Other Work Packages and Deliverables

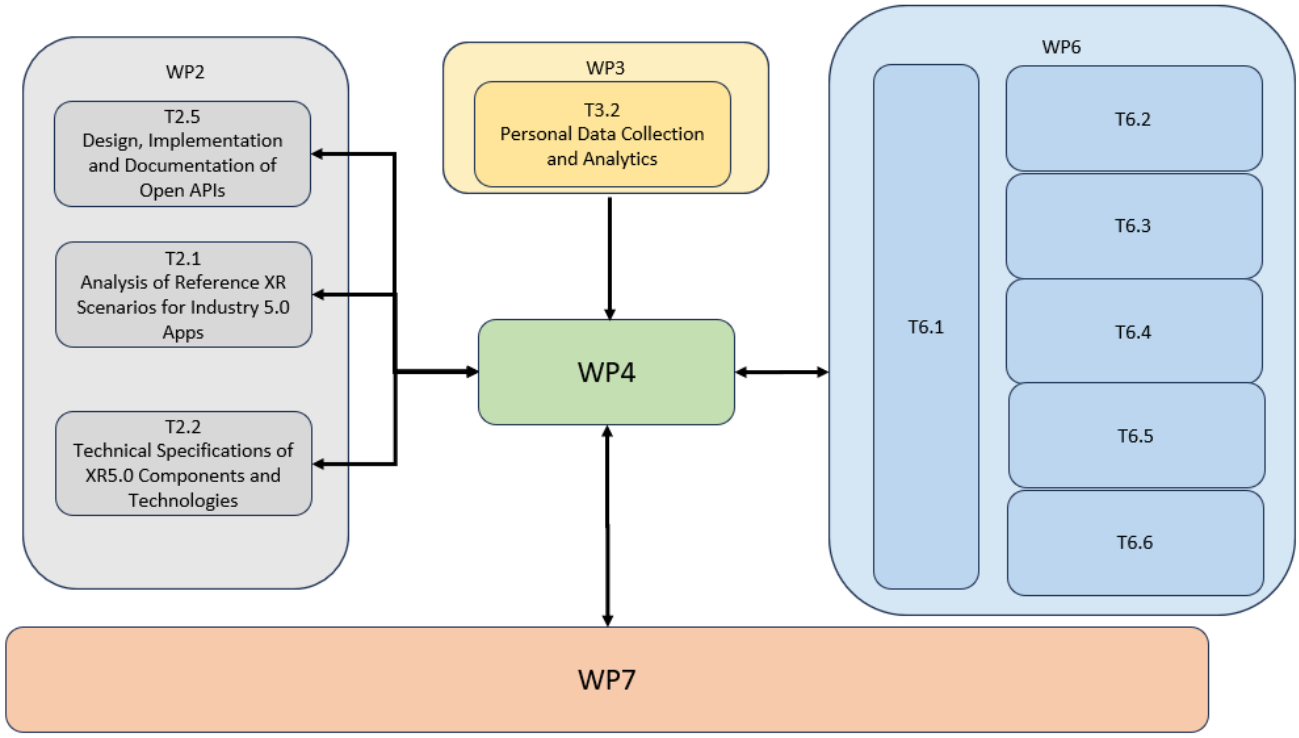

Figure 1 – Relation to Other Work Packages

Figure 1 illustrates the interactions and dependencies between the various WPs and tasks of XR5.0 that are related with WP4. WP2 focuses on the design, implementation, and analysis of XR5.0 technologies, including the documentation of Open APIs (T2.5), the analysis of reference XR scenarios for Industry 5.0 applications (T2.1), and the development of technical specifications for XR5.0 components and technologies (T2.2). These tasks provide inputs to WP4.

WP3 contributes to this integration through the task of personal data collection and analytics (T3.2), ensuring that data-driven insights feed into WP4. WP4, in turn, interacts with WP6, which include various deployment and operational activities, such as system implementation and management across tasks T6.1 through T6.6. Finally, WP7 support the overall project exploitation and dissemination activities, ensuring the seamless interaction and flow of resources between these interconnected components.

The integration of WP4 tools, including XAI, AL, NSAI, GenAI, and AR/VR visualization capabilities, with the XR5.0 reference architecture described in D2.2, for delivering advanced Industry 5.0 functionalities. This integration is depicted in the layered architectural framework, which incorporates the Business, Application, and Technology layers, as detailed in D2.2. The Business Layer emphasizes workflows and services that rely on XR-enhanced training, maintenance, and production monitoring, where tools like XAI provide interpretable insights, and GenAI powers dynamic content generation for operator assistance. For instance, Generative AI, facilitates the transformation of static user manuals into interactive, queryable formats accessible through the XR platform. This aligns with the XR5.0 objective of delivering value-added services such as immersive training and real-time operator guidance, which are essential for Industry 5.0 scenarios.

The XR5.0 platform facilitates the integration of AI models like XAI, NSAI, and GenAI as key components within the system, enabling capabilities such as operational optimization, scenario generation, and decision support. The Container Diagram from D2.2 highlights the orchestration role of the Central XR Hub, which consolidates these AI-driven functionalities while ensuring seamless interaction with AR/VR systems for visualizing real-time analytics and recommendations. The Technology Layer underpins this integration by

providing the necessary virtualization, deployment pipelines, and network connectivity. This setup ensures scalability and secure operation of XR applications across various pilots.

1.3 Relation to Other WP4 Tasks

The integration and workflow of advanced AI methodologies within WP4, highlighting the roles of NSAI, AL and GenAI are illustrated in Figure 2 - Figure 4, emphasizing their contributions to creating trusted human-AI collaboration frameworks, dynamic content generation, and interpretability through XAI. Additionally, it discusses how these technologies connect to downstream tasks for deployment, refinement, and their application in immersive XR5.0 training platform.

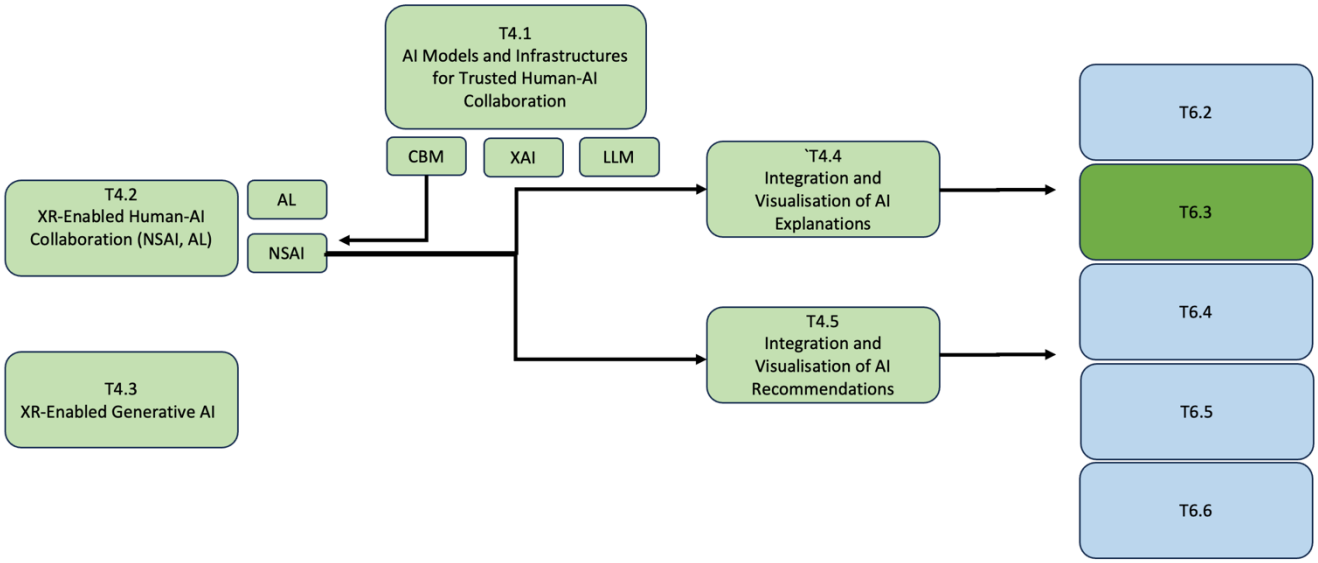

Figure 2 – T4.2, NSAI in relation to Other Tasks

Figure 2 illustrates the workflow of WP4, emphasizing the development and integration of Neurosymbolic AI (NSAI) models (T4.2). Task T4.1 serves as the foundational component, creating AI models and infrastructures for trusted human-AI collaboration. It leverages inputs from key modules, including Context-Based Models (CBM) that, as described in section 4.2, is capable of extracting "concepts" leveraging LLM technologies, in order to make each dataset/AI model self-explainable and NSAI-ready. NSAI (T4.2) is trained on the contextually enhanced data offering a self-explainable model permitting also human-AI interactions. The outcomes from these tasks are directed to tasks T4.4 and T4.5, which facilitate the integration and visualization of AI explanations and recommendations, respectively. Finally, the results are fed into T6.3 for operational deployment and further connect to other WP6 tasks for implementation, testing, and refinement.

Figure 3 illustrates the pipeline showcasing the utilization of Active Learning (AL) models (T4.2) for XR-enabled Human-AI collaboration. In this context, AL is applied to improve the adaptability and efficiency of XR systems by iteratively relabelling data points to refine the learning process. AL operates independently of NSAI in T4.2, though in general, AL can be augmented by NSAI for more complex reasoning tasks. The insights generated by AL within T4.2 feed directly into T4.1, which provides XAI methods for trusted human-AI collaboration. From T4.1 and AL, the outputs are progressing to T4.4 and T4.5, where the focus shifts to the integration and visualization of AI and XAI outputs. T4.4 ensures that AI explanations are interpretable and accessible to end-users, enabling better understanding and trust in the system, while T4.5 focuses on delivering recommendations tailored to specific contexts. These results are then routed to WP6 for deployment and are further refined and implemented across downstream tasks.

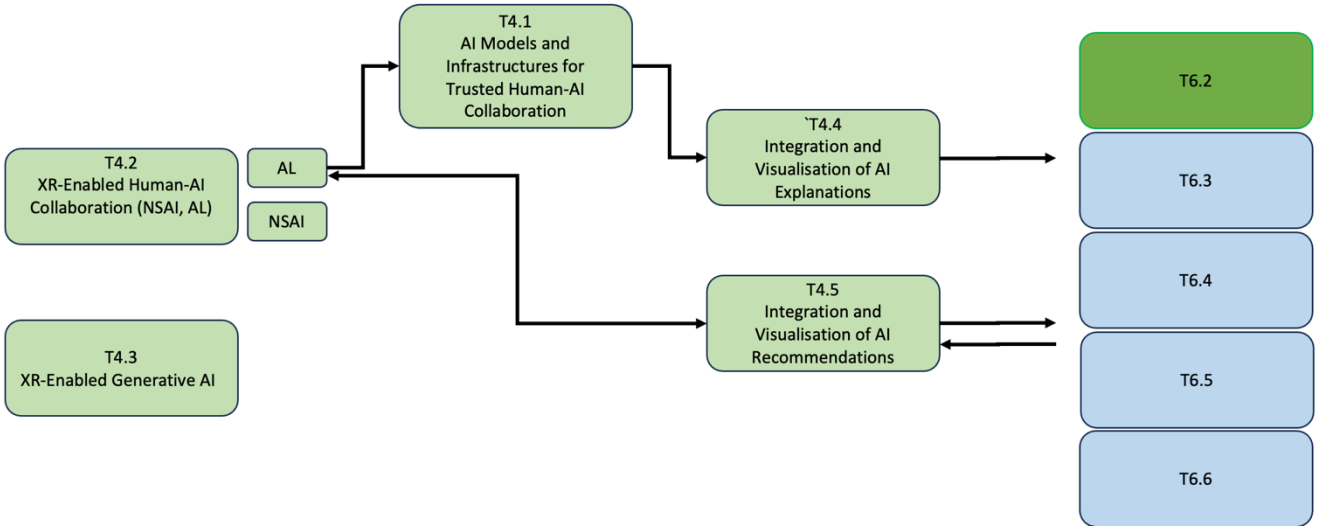

Figure 3 – T4.2, AL in relation to Other Tasks

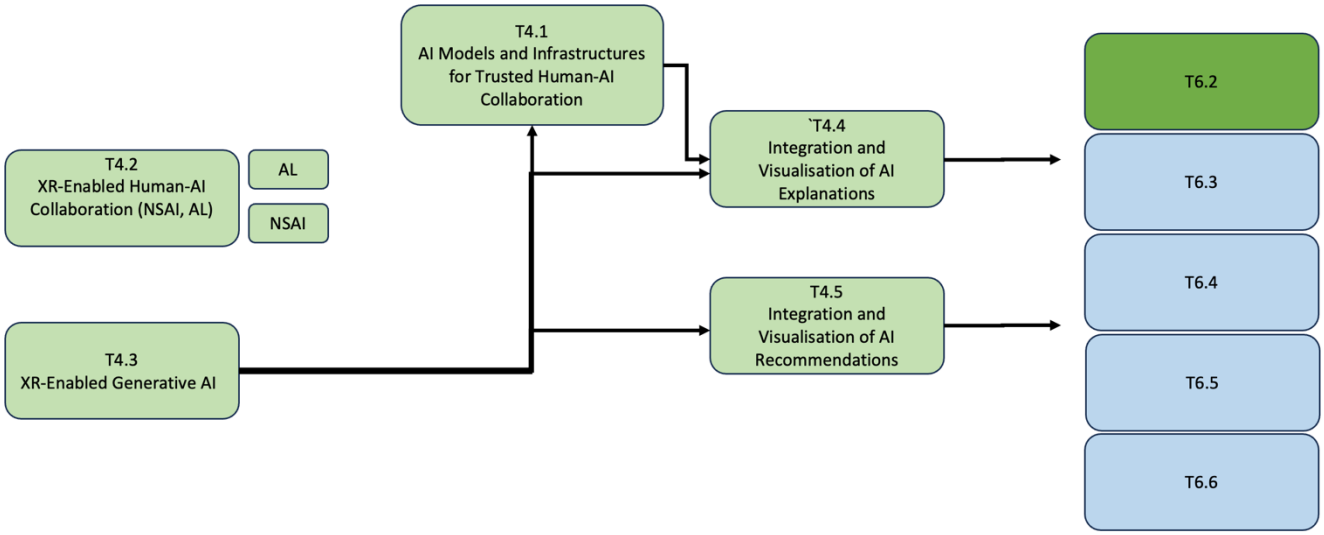

Figure 4 – T4.3, GenAI in relation to Other Tasks

Figure 4 illustrates the pipeline showcasing the role of generative AI (GenAI) models (T4.3) and its broader integration into the XR-enabled framework. T4.3 focuses on employing GenAI to dynamically create content, scenarios, and interactions specifically tailored to enhance XR systems. Beyond its standalone applications in T4.3, GenAI also plays a critical role in T4.1, where GenAI will be utilized as part of Human-Centric XAI, enabling the generation of human-friendly explanations and contextual responses. Additionally, GenAI serves as a key interface for the XR5.0 training platform developed in WP5, facilitating immersive, personalized, and adaptive training experiences tailored to users' needs. The outputs from T4.3 and their integration into T4.1 feed directly into T4.4 and T4.5. Task T4.4 use the combined capabilities to generate actionable and context-aware recommendations for users.

1.4 Structure of the Deliverable

The deliverable is structured systematically to present the progress for each task T4.1, T4.2 and T4.3 in a designated section.

- Section 2, following this introductory section, presents the technologies from Task T4.2, XR-Enabled Human-AI Collaboration (Neurosymbolic AI, Active Learning).
- Section 3 presents the progress on Task T4.3, XR-Enabled Generative AI.
- Section 4 presents the progress on Task T4.1, AI Models and Infrastructures for Trusted Human-AI Collaboration.
- Section 5 concludes the deliverable.

The technologies in each task are presented across several key dimensions addressing:

- *State-of-the-art*, reviewing the current relevant technological developments,
- *Role and functionality*, defining the specific contributions of the respective technology,
- *Component status*, detailing the key components associated with the respective technology,
- *Evaluation*, summarizing the early applications and validation of the technology,
- *Component resource requirements,* specifying the infrastructure needs for applying the technology,
- *Installation and deployment guidelines* to deploy the technology in real-world or pilot environments,
- *Demonstration scenarios and mapping to pilots*, connecting the technology application to specific project pilots,
- *Challenges and limitations* encountered or anticipated during the technology application in the project pilots,
- *Next steps* towards further implementation, suggesting improvements or refinements based on current progress.

2. XR-ENABLED HUMAN-AI COLLABORATION

This section aims to provide an overview of the Artificial Intelligence (AI) models that will be used for the purposes of the XR5.0 project. The European Commission has stated the necessity of the European industry to move towards Industry 5.0 standards, which include the adoption of human-centric activities for digital technologies, specifically AI. Furthermore, the need to up-skill European workers and provide them with digital tools to enhance their productivity and decision-making in their daily tasks is of paramount importance. Therefore, AI technologies applied in industrial environments need to be explainable and understandable by domain experts to enhance their capabilities. Neurosymbolic AI (NSAI) introduces a symbolic reasoning level to traditional AI models allowing domain experts to generate and/or extract semantic rules and reasoning out of "black-box" models. Active Learning (AL) introduces a level of interaction between a human actor and a model which increases algorithmic accuracy but also allows humans to interact directly with the AI. The following segments of this section will introduce the basic principles of NSAI and AL models and showcase how they match with XR5.0 pilots.

2.1 Neurosymbolic AI

In the past years, the advancements in the symbolic reasoning and the neural computation fields have given birth to a new scientific field called Neurosymbolic AI (NSAI). Symbolic AI, which dominated the early stages of artificial intelligence research, is characterized by its reliance on explicit rules, logic and structure knowledge representations. These systems excel in tasks that require reasoning, however they struggle to learn from raw data.

Neural models on the other hand extract patterns from vast amounts of unstructured data and try to generalize and make predictions based on those patterns. However, these models are often criticized based on the lack of explainability. Neurosymbolic AI aims to create a synergistic framework that leverages the complementary strengths of symbolic and neural approaches. By integrating the robust reasoning capabilities of symbolic systems with the adaptive learning capabilities of neural networks, neurosymbolic AI endeavours to build more intelligent, flexible and interpretable AI.

2.1.1 Brief Survey of State-of-the-Art

The NSAI algorithms can be separated into 5 categories, a taxonomy firstly mentioned by Kautz [1].

- Symbolic[Neuro] refers to systems that combine the statistical capabilities of Neural Networks along with symbolic reasoning. Example of this category is Deep Mind's AlphaZero [2], a reinforcement learning approach that given only the rules of the game and with no prior knowledge, achieved superhuman performance in the game of chess leveraging the statistical power of neural networks. The algorithm uses Monte-Carlo Tree Search for its symbolic part and estimates the value of states of the environment using Neural Networks.
- Neuro|Symbolic usually consists of a pipeline where both Symbolic and Neural systems specialize on a specific task and collaborate in order to achieve the final results. Most neurosymbolic algorithms belong to this category. Some widely known algorithms that fall in this category are IBM's neuro-vector-symbolic architecture [3], an architecture that combines Vector Symbolic architecture[3] along with powerful Neural Networks that are able to extract features from images. Utilizing this architecture, the authors were able to achieve state of the art results on the Raven dataset [4]. Another very well-known algorithm of this category is the Neuro-symbolic visual question answering [5] by Yi et al, which is a model applied to the CLEVRER dataset [6], a system capable of recognizing objects on an image and replying to questions regarding those images (how many cylinder objects, which object is the closer one, which is the biggest one etc.).

[3] VSAs are computational models that utilize high-dimensional distributed vectors and the algebraic properties of their robust operations to combine the strengths of connectionist distributed representations with structured symbolic representation.

- Neuro:Symbolic→Neuro consists of systems that use symbolic rules in order to guide the learning process of the neural networks. One of the most relevant works is Logical Neural Networks (LNN) by Riegel et al [7]. The framework combines neural learning with symbolic logic, where neurons represent components of weighted logic formulas for interpretability. It performs omnidirectional inference, including first-order logic reasoning, and uses a contradiction-minimizing loss for robustness to inconsistencies. By bounding truth values, it supports probabilistic semantics and handles incomplete knowledge effectively.
- NeuroSymbolic aims to map symbolic rules onto embeddings in order to be applied to a loss function as soft constraints. Logical Tensor Networks (LTNs), a framework proposed by Badreddine et al [8], is a neurosymbolic system that integrates querying, learning, and reasoning with data and abstract knowledge. It features Real Logic, a fully differentiable logical language that grounds first-order logic elements onto data via neural computational graphs and fuzzy logic.
- Neuro[Symbolic] systems enhance neural networks with the explainability and robustness of symbolic reasoning. Unlike Symbolic[Neuro], which uses symbolic reasoning to guide learning, Neuro[Symbolic] integrates symbolic reasoning directly into neural models, focusing on specific symbols under certain conditions. Neural Logic Machines, a framework proposed by Dong et al [9] combine neural networks for function approximation with logic programming for processing objects, relations, and logical structures. Trained on small tasks, they recover abstract rules and generalize to large-scale problems.

2.1.2 Role and Functionality

The NSAI component aims to provide a pipeline that will help with different tasks in the XR5.0 industry. The component will consist of 2 modules, a Neural Network (NN) part that will be responsible for tasks like object detection/ image recognition and a symbolic part that will be responsible for generating rules that will guide the training of the NN part.

For each pilot where the module will be integrated, separate API will be provided that will receive images or videos as inputs and the outputs will depend on the goals of each pilot along with explanations for its response. Also, the rules extracted from each model will be pilot dependent and there will be a separate end point in the API for adding expert knowledge for the task's rules.

Each model will be trained offline for each specific task and will be uploaded in an online endpoint to produce the process of the image/ video and respond with the corresponding output. In most tasks, based on the implied rules the model has deducted, there will be a detailed explanation of the model's output. To facilitate better knowledge extraction for assisting human experts, the model will use the implied rules to generate detailed explanations regarding its output.

2.1.2.1 Component Status

At its current stage, the component is classified as TRL 4, as it has only been tested in a lab environment.

2.1.2.2 Evaluation

At its current stage, the component uses the dataset (videos) provided by Pilot 3 (T6.4, EKSO pilot) as described in Deliverable D6.1. The goal of this component is to recognize anomalies and accurately identify their types at a satisfactory ratio.

2.1.3 Computational Resource Requirements

The NSAI component requires the following computational resources:

- A minimum of 4 CPUs.
- At least 12 GB of memory.
- A minimum of 20 GB of storage.

2.1.4 Installation and Deployment Guidelines

To run the NSAI component, the following libraries are utilized:

- Python
- Torch
- Scikit-Learn
- OpenAI
- LTN

2.1.5 Demonstration Scenarios and Mapping to Pilots

The NSAI component is currently implemented for Use Case 3 of Pilot 3 (T6.4, EKSO pilot). This use case involves developing an AI-based process for the digital examination of CCTV images and videos, which will also incorporate XR technology.

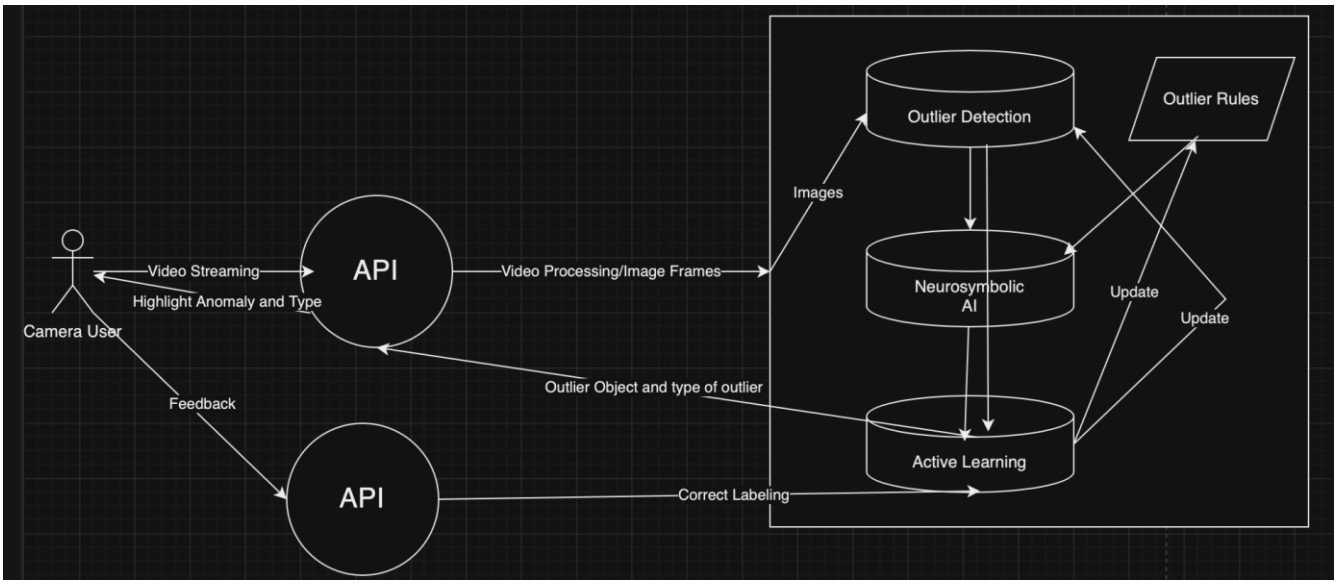

Figure 5 – Pilot 3 NSAI interactions

The role of the NSAI component is to identify the type of outlier detected by the anomaly detection component. This component can be used in Pilot 3 for the task of CCTV inspection. The NSAI component can act synergistically with the outlier detection component.

In this process, the CCTV video would be uploaded and split into multiple frames (images). The outlier detection component would label certain images as outliers. For these outlier images, the NSAI component would aim to identify the type of outlier.

First, it would extract relevant features from the images labelled as outliers. The most relevant features for the task would have been proposed by a large language model (LLM). The model can extract these features as embeddings, which would then serve as rules to categorize the various types of outliers. The results of the model's analysis would be fed back to the user.

Additionally, this process could collaborate with the Active Learning component, allowing an expert user to label the final results as correct or incorrect.

2.1.6 Challenges and Limitations

The biggest challenge faced by this component is extracting distinctive features from the frames (images), as most frames from the CCTV inspection are highly similar. The component must be capable of identifying different features, which will then serve as rules to enable the neurosymbolic part, in collaboration with the outlier detection component, to categorize the type of outlier.

2.1.7 Next Steps

The next steps for the NSAI component are the integration with the Active Learning component that will enable the expert to provide feedback of the model's output.

Also, one of the next steps is to incorporate the Concept Bottleneck Models output from Task T4.1 and their output will act as symbolic rules in the tasks handled by the NSAI component.

2.2 Active Learning

The rapid advancements in the field of Artificial Intelligence (AI) and Machine Learning (ML) have imposed the necessity for development of new ways of interactions between humans and algorithms. Active Learning (AL) algorithms have long been used to label unlabelled datasets or introduce human import(s) when algorithmic uncertainty tends to rise. Due to the recent advancements in the field of AL it is important to separate different approaches based on the notion of who (algorithm or human) manages the learning process of the model. Based on traditional AL concepts the system (e.g. model) has full control of the learning process while interactive machine learning approaches tend to provide a closer collaboration between humans and learning systems. Immersive technologies which tend to create distinct environments between the physical, digital or simulated world can be of great value to interactive machine learning solutions, since they empower users/domain experts to make sophisticated decisions.

2.2.1 Brief Survey of State-of-the-Art

Active Learning (AL) is a subcategory of Machine Learning (ML). It aims to generate interactive questions/queries to label unlabelled datasets with the smallest annotation cost. AL could also be a semi-supervised learning method as it might use both labelled and unlabelled data and request annotations for either existing or new examples only once, through iterative queries, thus increasing model accuracy. Furthermore, [10] AL solutions assume that different data samples of the same dataset tend to have different values in updating a model, thus they try to identify those cases of the highest algorithmic value. Although the identification of the "most important" samples in large datasets is one of the main benefits of AL approaches, [11] their difficulty of handling high dimensional data, urges for the combination of AL with Deep Learning (DL) solutions to achieve superior results. Solutions that combine both DL and AL solutions have been adopted in various fields including image identification [12], [13] and object recognition [14].

While identifying the most relevant samples or combining advanced DL techniques to enhance efficiency are of paramount importance, proper human intervention is of the same significance as well. In traditional ML and AL scenarios human interaction is minimal and is confined in building, modelling and testing algorithms thus creating the risk of generating static models which are difficult to scale and evaluate. Such limitation was highlighted by [15], which also identified the potential loss in identifying causal relationships and logical reasoning. It is worth mentioning that one of the key aspects of AL is the learning process of a given model. In traditional AL solutions [16] the model has full control over the learning process and the human input is limited in annotating unlabelled data or data points of high uncertainty. Interactive Machine Learning (IML) methods [17] provide closer interactions between models and the human factor by requesting information in an incremental way to improve algorithmic parameters and generate useful artifacts. Therefore, the need for new innovative interactions between ML solutions and human/domain experts has generated the field of Human-in-the-loop Machine Learning (HITL-ML) as defined by Munro [18]. HITL-ML approaches diverge from traditional AL solutions whose only focus is to increase algorithmic accuracy, by involving the human factor in such ways that also make humans more efficient and effective while interacting with the model. It should also be noted that part of the HITL-ML umbrella is also the field of [19] Machine Teaching (MT) approaches, which aim to provide "full control" of the learning process to the human expert by carefully defining the information/knowledge that they intend to inherit to the model.

As already mentioned above, a wide variety of AL solutions have been applied for different tasks such as image classification and object recognition. According to [20], HITL-ML and more specifically IML approaches were successfully applied for image and video classification tasks. Interactive image segmentation was particularly useful for domain experts to highlight, mark or annotate the most important

parts of an image that were the most relevant to a given model. Such relevant work can be found in ilastik which was developed by [21] as a tool for (bio)image analysis to assist human users without significant computational expertise and [22] AIDE, which is a tool that assists ecological surveys by leveraging IML concepts.

2.2.2 Role and Functionality

The Active Learning component developed for the purpose of Task T4.2 of the XR5.0 project aims to enhance human-in-the-loop approaches by using immersive technologies to increase algorithmic accuracy and human cognition. Due to the needs and requirements of XR5.0 pilots, the AL component will be provided along with an object detection model that will be used either for detecting sensors in industrial environments or for outlier/anomaly detection tasks.

Due to the different requirements expressed by the project's pilots the AL component will be provided separately for each pilot. It incorporates video and image processing and introduces a semantic layer for the interaction between the human actors and the AI model. At its current state, it is designed to receive either videos and/or images for the image processing task and generate annotated output that is currently stored on a knowledge base and presented on human experts. Furthermore, it measures the algorithmic uncertainty of each output and incorporates human knowledge once the uncertainty levels fall under specific thresholds. Human experts are given the option to also trigger the AL component ad-hoc, in case they identify mislabelled outputs. The AL component is provided as a RESTful service to allow the interaction with other XR5.0 components and more specifically with the XR environments and the NSAI module.

The AL component empowers domain experts to be in the forefront when using AI as it allows them to directly interact with the models in immersive environments, thus enhancing algorithmic accuracy, expert knowledge and human cognition.

2.2.2.1 Component Status

At its current stage, the component is classified as TRL 4, as it has only been tested in a lab environment.

2.2.2.2 Evaluation

At the current stage the component has used initial datasets (videos) provided by Pilot 3 (T6.4, EKSO pilot) to build the initial solutions for the Active Learning and the Outlier Detection components. The initial solution which proves as Proof of Concept was presented to the EKSO representative and was considered satisfactory. Since the original presentation EKSO has provided a larger dataset of CCTV video inspections which are currently being used for the extension of the models. It needs to be mentioned that regarding the AL component, the human input (expert/operator) was performed in a lab environment at this stage.

2.2.3 Computational Resource Requirements

The computational resource requirements for the components are as follows.

Active Learning Component:

- Num. CPUs >= 4
- Memory >= 12GB
- Storage >= 20GB

Outlier Detection Component:

- Num. CPUs >= 4
- Memory >= 12GB
- Storage >= 20GB

2.2.4 Installation and Deployment Guidelines

To build and run both Active Learning and Outlier Detection components the following packages are required:

- Python 3.9.2 (or higher)
- torch 2.5.1
- Opencv-python 4.10
- Docker
- Docker-compose

A docker file is also provided to build all necessary environments and packages.

2.2.5 Demonstration Scenarios and Mapping to Pilots

The Outlier Detection and Active Learning components are currently implemented for the purposes of Use Case 3 of Pilot 3 (T6.4, EKSO pilot). Use Case 3 of Pilot 3 requests the development of an AI based process for digital examination of CCTV images/videos which will also make use of XR technology.

The current purpose of the AL component is to assist in detecting anomalies in an Outlier Detection scenario. The outlier detection objective is to carry digital examinations of CCTV images/videos with AL for EKSO's Pilot 3, to assist human operators in identifying potential flaws in EKSO's pipeline system. It aims to automatically detect outliers/anomalies/defects on the pipelines and alert operators by providing a Human in the loop element of the AI algorithms to assist human actors further improve their efficiency and accuracy. In its current stage, the component receives the CCTV video and splits it into a series of frames (images) for the outlier detection task. Considering that the given data set is unlabelled, the AL component is initially triggered to request from the user/operator the definition of an inlier and an outlier by annotating a series of sampled images. Consequently, the AL component "builds" the proper dataset that will feed the training of the Outlier Detection algorithm. Every time a new video is processed and split into frames/images, the outlier detection component is triggered to determine whether the given image is an outlier or an inlier; at the same time the uncertainty of its decision/prediction is measured and once it raises above a user defined threshold the AL component requests from the expert/operator to label the uncertain image(s). If the label provided by the expert/operator is different from the outlier's detection component prediction, the AL component is triggered again to re-evaluate the algorithmic parameters of the outlier detection task to increase accuracy. At the end of the outlier detection task, the expert/operator is provided with the series of images that have been identified as anomalies and can interact with the system again in case an error is spotted. Therefore, there is a direct handshake between the Outlier Detection algorithm and the Active Learning component which aims to acquire all required human knowledge from the expert/operator to the model in a Human-in-the-loop scenario. An initial validation of the implementation has already been performed by EKSO experts and currently the implementation is moving towards automating all the steps of both Active Learning and Outlier Detection components.

Pilot 1 (T6.2, KUKA pilot), Pilot 5 (T6.6, SPACE pilot) and Pilot 6 (T6.7, LNS pilot) have expressed interest in using the AL component for their activities. Pilot 1 and Pilot 6 have shared small datasets and the UPRC team is currently evaluating the feasibility of the solutions based on the given sets. Pilot 1 and Pilot 5 are mostly interested in using the AL solution in an Object detection scenario either in an offline or streaming manner, while Pilot 6 is evaluating the use of AL solutions to enrich their existing knowledge base.

2.2.6 Challenges and Limitations

The current limitation of the developed approach is twofold. First, the current approach is used in an offline manner, meaning the videos are split into a series of frames/images which are then analysed by the Active Learning and Outlier Detection components. The second limitation lies on the recognition of the outliers themselves. Currently a frame/image is recognised as an outlier (binary classification) or not. Therefore, the nature of the problem in EKSO's pipeline network cannot be specified (e.g. water leakage, pipe corrosion, broken pipes, etc.) to further generate knowledge and assist operators.

2.2.7 Next Steps

As stated above, at its current state the AL component is provided along with an Outlier Detection solution for the purposes of Pilot 3. Therefore, the initial goal is to assist the Outlier Detection component with an Object Detection task that will be used to specify the nature of an outlier (e.g. Broken pipe, etc.). To that end, we aim to use the NSAI component for the "creation" of additional knowledge of the frames/pictures and use them in combination with the Outlier Detection and Active Learning component, since it has been acknowledged by Pilot 3 (EKSO) that any additional information that can be provided by the AI components will be of great value to their experts and operators.

Additionally, Pilot 1 (T6.2, KUKA pilot), Pilot 5 (T6.6, SPACE pilot) and Pilot 6 (T6.7, LNS pilot) are considering utilizing the AL component, particularly for applications in Object Detection scenarios as described in Deliverable D6.1. To that end, data provided by each pilot are currently being evaluated for the feasibility and development of solutions that cover each Pilot's needs and requirements. Once this step is finalized our aim will focus in making the whole solution operational in streaming environments, thus making it real time. Since the interfaces will be built in an XR environment one of the main priorities will be the communication/handshakes between the AI components and the XR environment provided by CYENS.

3. XR-ENABLED GENERATIVE AI

This section explores the integration of Generative AI technologies within XR environments to enhance human-AI collaboration, aligning with Industry 5.0 principles. It outlines the capabilities of the Large Language Model (LLM) Chat Engine, focusing on Retrieval-Augmented Generation (RAG) for delivering intelligent, context-sensitive assistance. The integration aims to revolutionize training and maintenance procedures in industrial settings by providing personalized, real-time AI support. Key components include a vector database for knowledge retrieval and a RESTful API to facilitate seamless interaction between XR devices and AI systems. This section further examines the technical implementation, evaluation, and challenges, concluding with next steps for system enhancement.

3.1 Brief Survey of State-of-the-Art

The landscape of artificial intelligence has been transformed by LLMs and Generative AI (GenAI), fundamentally changing human-computer interaction through natural language interfaces. Modern LLMs based on the Transformer architecture, including GPT-4, LlaMA, Gemini, and Claude, have become the foundation for developing innovative applications, particularly in code generation, chatbot support, and enterprise search and retrieval [23] [24].

Two key techniques have emerged as dominant approaches for leveraging LLMs in practical applications:

1. RAG has become instrumental in enhancing LLM performance within enterprise settings. By combining generative capabilities with external knowledge retrieval, RAG addresses common LLM limitations such as factual inaccuracies and hallucinations [25]. Recent advances include:
    - Development of federated systems integrating diverse information sources.
    - Extensions supporting multi-modal, contextualized, and personalized question-answering.
    - Optimization strategies focusing on content design and modular, model-agnostic approaches [26].
2. LLM Agents represent an evolution in AI development, enabling autonomous execution of complex tasks across domains like robotics, gaming, and API integration [27]. In the agentic workflows, the LLM acts as orchestrator coordinating the tasks required by leveraging tools, functions and external systems. Recent research has focused on improving agents' decision-making capabilities through:
    - Implementation of the Retrieval-Augmented Planning framework
    - Enhanced utilization of contextual memory
    - Integration with both text-only and multimodal environments

In relation to XR, novel LLMs are being researched for 3D object creation. MS2Mesh-XR presents a multi-modal sketch-to-mesh generation pipeline that enables users to create realistic 3D objects in XR environments using hand-drawn sketches and voice inputs [28]. TRELLIS is a large 3D asset generation model which takes in text or image prompts and generates high-quality 3D assets in various formats, such as Radiance Fields, 3D Gaussians, and meshes [29]. Particularly relevant to XR5.0's training focus, applications based on ChatGPT's has been proposed enhancing student learning experiences, specifically in the context of construction hazard recognition and safety training [30].

While GenAI's impact spans numerous fields - from academia and healthcare to agriculture and business management - a significant gap exists in professional development applications, particularly in integrating GenAI with training methodologies [31]. XR5.0 specifically aims to address this gap through innovative approaches to industrial training.

However, implementing LLMs for specialized applications presents distinct challenges. While models excel at broad knowledge tasks, they often struggle with domain-specific contexts crucial for industrial applications. Current research focuses on developing methods and frameworks to enhance LLMs with specialized knowledge for industry and smart manufacturing applications. It's worth noting that the integration of LLMs with XR applications remains in its early stages, with few real-world industrial implementations currently documented.

3.2 Role and Functionality

Task T4.3 introduces a novel approach to industrial training and applications by combining AI-powered conversational capabilities with immersive XR experiences, aligned with Industry 5.0's human-centric vision.

The system architecture centres around two core components:

1. An LLM Chat Engine that delivers intelligent, context-sensitive guidance
2. A Vector database for integrating proprietary or domain specific knowledge

This system can be integrated with XR devices to provide real time assistance to the user while the responses from the chat engine can be adapted to each user's individual needs. Figure 6 illustrates an indicative workflow for training which operates as follows:

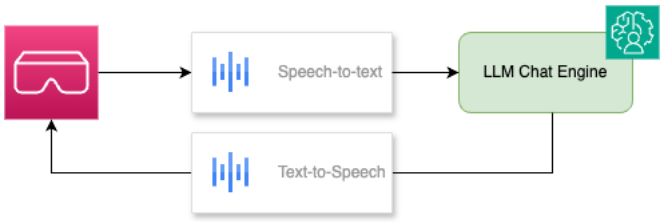

Figure 6 – Indicative Workflow between XR and the LLM Chat Engine

When the user interacts with the system, they can ask questions verbally while wearing a VR headset. These verbal inputs are processed through a Speech-to-Text (STT) service, converting spoken words into text format suitable for AI processing. The VR Application, which manages the immersive environment and user interactions with virtual equipment, forwards this text to the LLM Chat Engine.

The LLM Chat Engine is implemented as a RESTful service that exposes key endpoints for:

- Chat interactions
- Document management and knowledge base updates
- Training session configurations
- Orchestration of predictive maintenance, anomaly detection, and quality control tools

The LLM Chat Engine processes queries using RAG, incorporating relevant information from use case specific technical documentation and manuals to provide contextually appropriate responses. If the user query requires it, the LLM will also employ predictive maintenance tools to support repairing and servicing processes. The generated response is then seamlessly integrated back into the VR environment, where it's presented through visual cues and step-by-step guidance. This creates a natural interaction flow where trainees receive expert-level guidance in real-time while practicing tasks in a virtual space.

Task T4.3 service is also responsible for the uploading, processing and storing of these technical documentation and PDF manuals. Once the PDF is uploaded through the API endpoint, the processing begins where the large document is broken down into smaller, manageable chunks and then converted into numerical embeddings. Then these vectors are organized and stored in a structured index of a Vector database, that facilitates quick access and retrieval during user interactions. The stored data are then accessed by their respective query engines, which are all managed by one OpenAI Agent.

This efficient data processing pipeline, from document ingestion to intelligent retrieval and LLM-powered response generation, forms the backbone of XR5.0's ability to provide end users with highly relevant, contextually aware guidance in real-time.

3.2.1 Component Status

Figure 7 provides a high-level overview of the LLM Chat Engine designed to support interactive, knowledge-enhanced responses to industrial XR-based environments. It demonstrates how various components work together to process user requests, retrieve relevant information, and deliver intelligent responses.

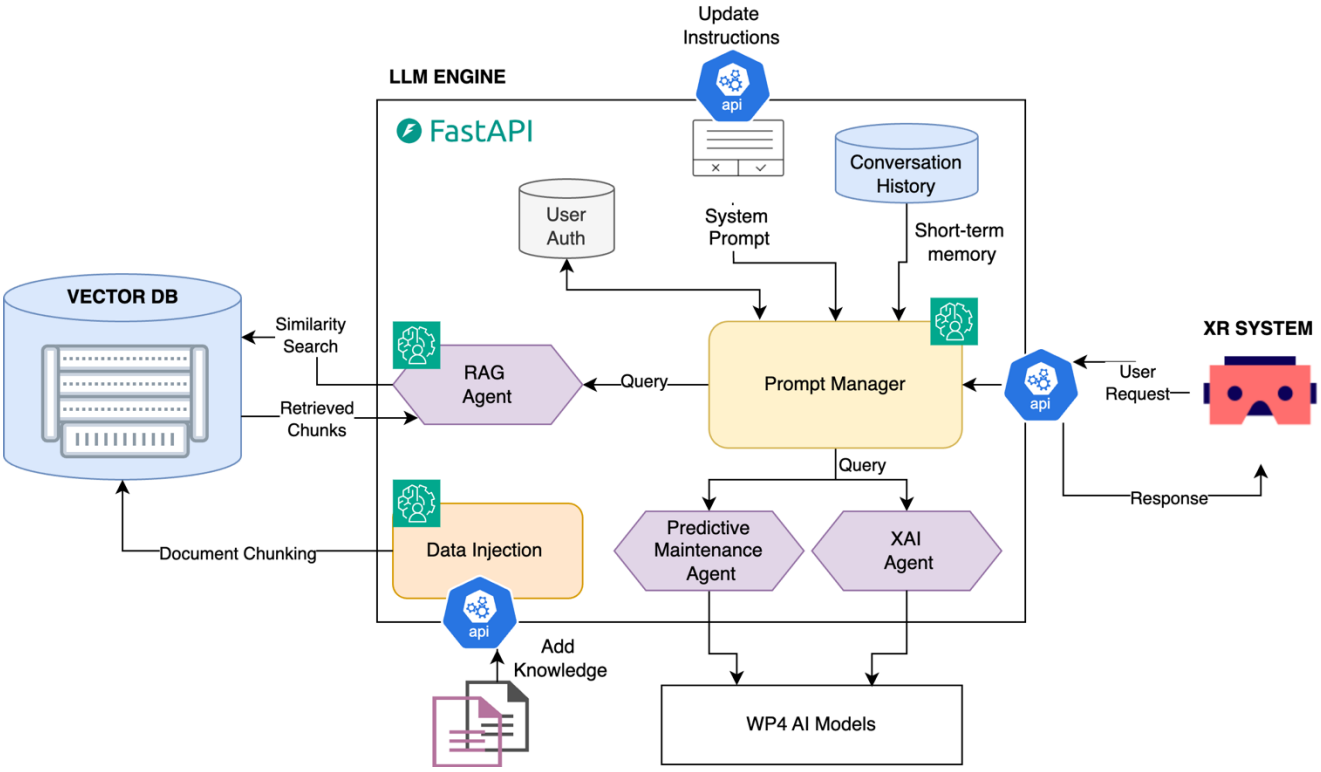

Figure 7 – High-level overview of Task 4.3 LLM Chat Engine

At the core of the system is the LLM Engine, which orchestrates communication between agents, manages prompts, and maintains conversation context. It integrates with a Vector Database for knowledge storage and retrieval, enabling the system to dynamically augment responses with relevant information through RAG. The architecture also incorporates modules for data injection, allowing new knowledge to be added, and agents to provide specialized analysis obtained from the other AI services of WP4 for enhanced decision-making. The system interacts with XR interfaces and applications, enabling immersive experiences by processing user queries and delivering context-aware responses. The current TRL of the LLM Chat Engine is 5-6 with target 7-8 at the end of the project.

The following sections elaborate on the internal operations of the main components including information about their implementation, software, integration mechanisms and deployment.

LLM Engine:

The LLM Chat Engine's core intelligence is driven by the Prompt Manager Agent. It leverages OpenAI's GPT-4o model to orchestrate and process input queries. The agent triggers the required tools or processes to retrieve relevant information and synthesizes the final response. The implementation includes a set of dedicated query engine tools, with each tool specifically designed to handle queries related to technical documents or manuals of a specific use case.

When processing a user query, the agent first evaluates the input and autonomously determines which query engine tools should be invoked. This selection process considers multiple factors, including the query content, document metadata, and user-specific context.

To facilitate accurate information retrieval, each query engine is enhanced with metadata, such as document title, author, and type. For example, a maintenance manual for a robotic arm can be tagged with its version number and manufacturer details, enabling precise matching to queries about specific models. such as document title, author, and type. This metadata enrichment enables the creation of document-specific tools that can be precisely matched to user queries.

The Prompt Manager Agent may also call other specialized agents that assist the user in troubleshooting and problem-solving. These tools can be invoked when users require guidance on equipment maintenance or fault diagnosis, enabling the system to utilize external AI models provided by WP4 that can provide additional context to the user.

The behaviour of the Prompt Manager is dictated by the prompt instructions, which are application-specific and include information about the available tools, guardrails (predefined constraints ensuring responses remain relevant, safe, and within specified guidelines), response format, and the overall objectives of the agent that are application specific and include information of the available tools, guardrails, response format and the overall objectives of the agent.

The response generation process involves combining multiple information sources. Once the relevant text segments are retrieved from the source documents, the agent formulates a comprehensive response by integrating:

- The retrieved document content
- Context from the predefined system prompt
- The user's chat history
- Current training context
- Outputs from other agents when necessary

This orchestration of components enables the LLM Chat Engine to function as an automated reasoning engine that can decompose complex queries into manageable steps and leverage appropriate tools to provide accurate, contextual responses during training sessions.

This service is made available as a REST API built using the FastAPI framework. It includes endpoints to make queries, inject documents, add or update system prompts, and manage user authorization and session handling. Authorization is implemented using API keys for secure access control, while session management supports context persistence across interactions, ensuring seamless user experiences during training sessions. The several required agents and tools were built with LlamaIndex and LangChain frameworks [32].

Knowledge Integration:

The Knowledge Integration Component is comprised of four subcomponents that are described in the following subsections that create a data processing pipeline shown in Figure 8.

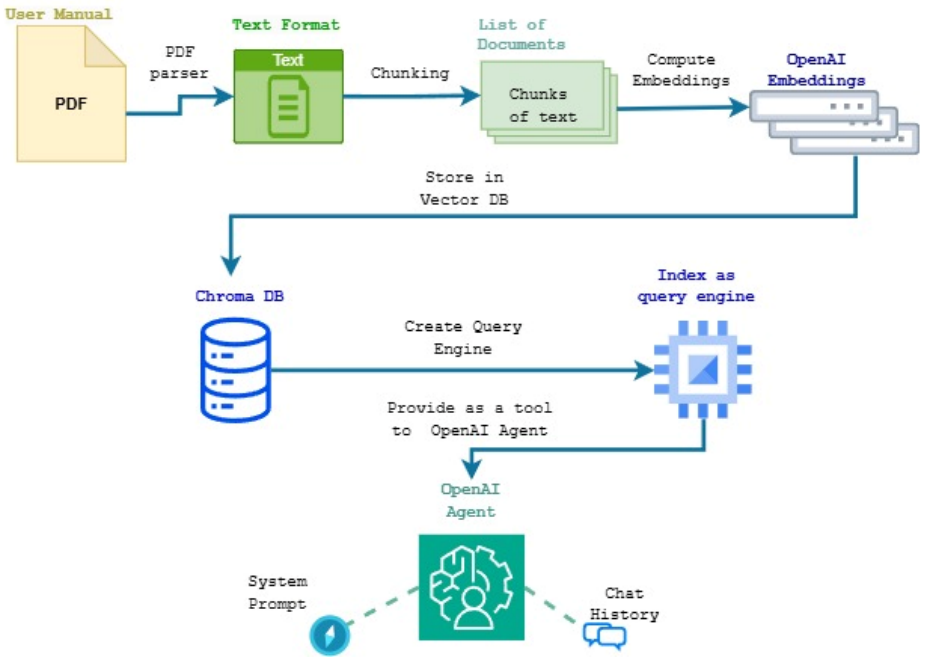

Figure 8 – Data and Knowledge Injection

Document Parsing and Chunking: When a document is retrieved through the PDF uploading API, it is first converted to text. In the context of XR5.0, the system primarily deals with technical documents related to industrial equipment, such as user manuals, technical datasheets, maintenance reports and standard operating procedures. These technical PDFs can vary in length, complexity and quality, necessitating the use of different libraries for optimal text extraction. The choice of library depends on the specific characteristics of the technical document, such as the presence of images, tables, graphs or scanned content.

For technical documents that are primarily text-based, like user manuals or standard operating procedures, the PyPDF2 library is used. PyPDF2 is a pure-Python library that can split, merge, crop, and transform PDF pages. It works well with standard PDFs and provides a straightforward method for extracting text content from technical documents with a simple layout.

When processing more complex technical documents that contain a mix of text, images, tables, and graphs, like technical datasheets or product catalogs, the PyMuPDF library (also known as fitz) is employed. PyMuPDF is a Python binding for the MuPDF library, which is a lightweight and fast PDF, XPS, and E-book viewer. It can handle a wide range of PDF types and is particularly effective for extracting text from scanned technical documents using Optical Character Recognition (OCR) techniques. For example, a technical datasheet for a new industrial sensor, containing detailed product specifications, performance graphs, and comparison tables, would be better processed using PyMuPDF to ensure accurate text extraction while preserving the document's structure.

For challenging technical documents, such as legacy maintenance reports or low-quality scanned images, the Tesseract OCR engine is utilized. Tesseract is an open-source OCR engine that supports a wide range of languages and can handle difficult-to-read text in images. It is useful for extracting text from poor-quality scanned technical documents or images with complex backgrounds.

By leveraging these different libraries based on the specific characteristics of the technical PDF documents, XR5.0 ensures that the text extraction process is optimized for each case. This approach maximizes the accuracy and completeness of the extracted text, which is crucial for the subsequent chunking and embedding steps in the LLM Chat Engine pipeline.

In addition to the extraction of the text, since these documents can be lengthy, spanning hundreds of pages, the text is divided into smaller segments or chunks. This chunking process is crucial for several reasons:

1. It enhances the relevance of the content retrieved from the vector database by ensuring that the embedded content contains minimal noise.
2. It ensures that the text fits within the context sent to external model providers like OpenAI, considering the limitations on the number of tokens that can be sent per request.
3. The choice of chunking technique, such as fixed-size chunking, can significantly impact the accuracy of the RAG process.

In the context of XR5.0's use cases, the input documents are typically well-structured official manuals for industrial equipment. These manuals are organized into distinct sections and subsections, with answers to user queries often found in specific parts of the document. Consequently, using fixed-length chunking could result in incomplete responses.

To address this issue, XR5.0's LLM Chat Engine employs the Unstructured framework for chunking. This framework partitions documents into semantic units called document elements. The system only resorts to text splitting when a single element exceeds the maximum supported chunk size. By doing so, most chunks contain one or more complete sections, preserving the coherence of the semantic units established during partitioning.

Specifically, XR5.0 uses a “by title” chunking strategy, where the content under each title or heading is treated as a separate chunk. This method groups relevant information, such as process steps or tables with technical specifications, while maintaining the logical structure of the manual.

By implementing this intelligent chunking approach, XR5.0 ensures that the information provided to users is comprehensive, accurate, and relevant to their queries. This contributes to the overall effectiveness of the AI-powered conversational training experience within the immersive VR environment.

Embeddings Creation: After the technical documents have undergone the pre-processing stage, where they are parsed and chunked into smaller, more manageable segments, the next crucial step in XR5.0's LLM Chat Engine pipeline is to convert these text chunks into text embeddings. Text embeddings are a powerful technique that transforms textual data into high-dimensional dense vectors of real numbers, effectively capturing the semantic meaning and context of the text.

The choice of embedding model plays a significant role in determining the quality and effectiveness of the text embeddings. Each embedding model works with a specific number of dimensions, where each dimension represents a unique semantic aspect of the text. In the context of XR5.0, several embedding models were considered, including OpenAI Embeddings, BERT, and MPnet. After evaluation, OpenAI Embeddings were selected due to their better performance and higher dimensionality. OpenAI Embeddings typically have 1536 dimensions, compared to 768 dimensions in other models like BERT and MPnet. The higher dimensionality allows for a more detailed capture of the semantic information within the text chunks.

To generate the text embeddings, XR5.0 utilizes the OpenAI’s API, specifically the *text-embedding-3-small* model. This model demonstrated excellent performance in capturing semantic similarities and relationships between text fragments.

By capturing the semantic meaning of the text chunks, the embeddings allow the LLM to identify and retrieve the most relevant information from the document database, even when the user's query does not contain exact matches to the document text.

Vector Database: For storing these embeddings, a Vector Database is preferable over a traditional database. These specialized databases excel at performing rapid and precise similarity searches, in comparison to traditional databases. When a user query is converted to text embeddings, the vector database becomes the most suitable option for comparing the input against stored embeddings, enabling quick data retrieval and improving result relevance.

For Task T4.3 system development several different vector database systems, (Pinecone, Chroma and Faiss) were evaluated in terms of cost effectiveness and accuracy. Pinecone emerged as the most suitable option after the assessment, with detailed benchmarking results presented in a subsequent section.

Beyond database selection, another significant part in the development of the system is metadata management, which is crucial for efficient embedding vector retrieval. By structuring metadata strategically, the system can significantly reduce the search space before applying vector similarity check.

The similarity search within Pinecone uses the approximate nearest neighbours search algorithm. Evaluations showed that the most accurate results were consistently obtained by selecting the top three chunks most similar to the query.

Query Engine Tools: The next phase of the document ingestion pipeline involves implementing a querying functionality depicted in Figure 9. For this purpose, the LlamaIndex and LangChain frameworks were chosen. Recognizing that the user typically references a small number of manuals within the documents uploaded by the organisation they work for, the system was designed to create dedicated query engines for each document. Each query engine is enriched with metadata including the document's title, author, and type, which helps define its specific scope and applicability. The LlamaIndex framework provides a flexible Query Engine interface that enables comprehensive querying of embedded document content. By creating document-specific query engine tools, the system can more effectively navigate and extract relevant information from specialized technical manuals. These query engines are managed by the Prompt Manager Agent.

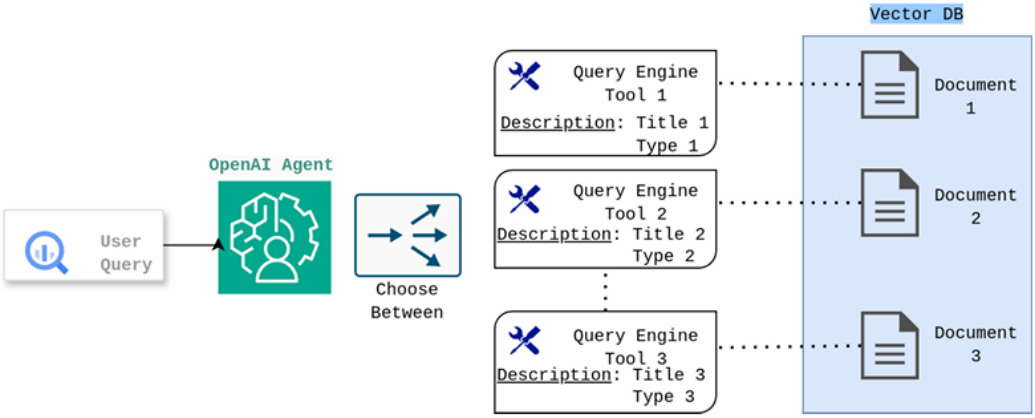

Figure 9 – Retrieval Augmented Generation (RAG) for Query Answering

3.2.2 Evaluation

To ensure the scalability and optimal performance of the retrieval system across diverse documents and question types, evaluation and benchmarking of various chunking strategies, embedding models, and vector databases was performed. The evaluation process and final retrieval outcomes were assessed using RAGChecker [33], an evaluation framework specifically designed for RAG systems.

The experiment focuses on testing three recognized strategies for each core function of the LLM Chat Engine—chunking, embeddings, and vector databases—to identify the most effective combination. The selection of parameters for testing was guided by both practical considerations and current industry standards in RAG implementations. For chunking methods, we tested fixed-length approaches at 1024 and 2048 tokens, representing common practice in production systems, along with context-based chunking to evaluate potential improvements in semantic coherence. The embedding models selected included OpenAI's *text-embedding-ada-002* and *text-embedding-3-small*, chosen for their widespread adoption and documented performance, complemented by HuggingFace's *all-mpnet-base-v2* as a leading open-source alternative. For vector databases, we evaluated FAISS, Pinecone, and Chroma based on their production readiness and established adoption in enterprise environments. This parameter selection ensures that our

evaluation results remain relevant for real-world implementations while exploring key alternatives in each component category.

This assessment aimed to further evaluate the retrieval system's performance across various types of user manuals by utilizing three PDFs representing different equipment from distinct manufacturers. Additionally, the PDFs differed in size. This approach ensured that the structural layout of the documents varied and added complexity to the evaluation.

After establishing the combinations of chunking, embedding models, and vector databases, the query engine was tested using 20 queries for each combination across all PDFs. These queries, which were potential questions users might ask during training, covered various sections of each document and were selected by the developer. Additionally, RAGChecker requires ground truth answers for all queries to enable later evaluation. The ground truth answers comprised complete, unmodified text segments from the source documents that addressed each query. This approach enabled precise validation: RAGChecker could verify that every claim in a response was directly supported by the retrieved document segments, thereby detecting any information not present in the source material as hallucination.

In total, 1,620 queries were evaluated across the 3 PDFs, with each query tested using 3 different chunking strategies, 3 embedding models, and 3 vector databases. For each query, two retriever metrics—Claim Recall and Context Precision—and two generator metrics—Hallucination and Faithfulness—were recorded.

Retriever Metrics

- Claim Recall: The proportion of ground truth claims covered by the retrieved chunks.
- Context Precision: The proportion of retrieved chunks that are relevant.

Generator Metrics

- Hallucination: The proportion of incorrect claims not found in any retrieved chunks.
- Faithfulness: How closely the generator's response aligns with the retrieved chunks.

For each combination of chunking, embedding model, and vector database in each PDF, the 20 queries were averaged to obtain a single value for each metric. The evaluation was conducted through a complete grid search, testing all possible combinations simultaneously rather than keeping components fixed. The results from these comprehensive combinations were then aggregated, which enabled the calculation of average metrics for each individual component (chunking strategy, embedding model, and vector database) while accounting for their performance across all possible combinations with the other components. This processing enabled the identification of which combinations yielded the desired results and provided the average metrics for each chunking strategy, embedding model, and vector database across all queries, regardless of the combinations used. The results are presented in Table 1 - Table 3 below.

Table 1 – Chunking strategy evaluation

Chunking Strategy	Claim Recall	Context Precision	Hallucination	Faithfulness	Rank
Fixed length=2028	**97.12**	99.05	3.59	91.14	2
Semantic Context	93.05	**99.13**	**0.21**	**99.07**	1
Fixed length=1024	85.41	95.37	4.51	95.49	3

Table 2 – Embedding model evaluation

Embedding Strategy	Claim Recall	Context Precision	Hallucination	Faithfulness	Rank
OpenAI - ada	93.7	98.6	3.56	96.49	3
Mpnet	**95.28**	**99.4**	5.37	90.28	1
OpenAI - small	86.61	96.47	**1.06**	**98.97**	2

Table 3 – Vector Database evaluation

Vector Store	Claim Recall	Context Precision	Hallucination	Faithfulness	Rank
Pinecone	**93.37**	97.22	2.54	**96.5**	1
Chroma	92.11	**98.05**	**2.01**	95.18	2
Faiss	90.11	98.05	3.21	94.48	3

3.3 Computational Resource Requirements

The computational infrastructure required for the XR-enabled Generative AI system is designed to ensure scalability, performance, and integration flexibility. The system utilizes Docker containers for deployment, ensuring consistent performance across environments. The LLM Chat Engine leverages OpenAI's API, which offloads intensive computational tasks to cloud-based infrastructure, minimizing on-premises hardware demands. A minimum of 2 vCPUs and 4 GB of RAM is recommended for managing API requests, processing user queries, and enabling real-time interactions, though resource requirements may scale based on the number of concurrent users.

For knowledge retrieval, Pinecone is employed as the vector database, offering scalable and high-performance embedding storage and similarity search capabilities. Storage needs are determined by the size of uploaded documents and embeddings, with a suggested baseline of 50 GB SSD for indexing and retrieval. Furthermore, a database such as PostgreSQL is necessary for user and session management to handle authentication, session tracking, and user-specific data storage effectively.

For knowledge retrieval, Pinecone serves as the vector database, providing scalable and high-performance embedding storage and similarity search capabilities. Storage requirements depend on the size of uploaded documents and embeddings, with an estimated baseline of 50 GB SSD for indexing and retrieval. Additionally, a database for user and session management, such as PostgreSQL is essential to handle authentication, session tracking, and user-specific data storage.

3.4 Installation and Deployment Guidelines

The LLM Chat Engine is containerized using Docker to ensure consistent deployment across different environments. The deployment process follows these straightforward steps:

1. Repository Setup:

- Clone the repository (https://github.com/giorgosfatouros/XR5.0-LLM-ENGINE) to your local environment
- Navigate to the project's root directory

2. Configuration:
    - Configure the OpenAI API key in the following files:
        - Dockerfile
        - docker-compose.yml
3. Build Process:
    - Build the Docker image by executing:
        - docker build -t llm-engine .
4. Deployment:
    - Launch the service using Docker Compose:
        - docker-compose up
5. Test the API with Curl:
    - curl --location 'http://0.0.0.0:8000/api/chat' \

      --header 'Content-Type: application/json' \

      --data '{“query”: “How can you help me?”}'

This containerized approach ensures that all dependencies and configurations are properly managed, making the deployment process reliable and reproducible across different environments.

3.5 Demonstration Scenarios and Mapping to Pilots

Some Demonstration scenarios aligned with XR5.0's pilot implementations include:

1. **Visual Inspection and Anomaly Detection Training:**
    - Supports EKSO's smart pipes pilot by providing real-time AI guidance for inspection procedures
    - Assists trainees in identifying potential anomalies by leveraging knowledge from technical documentation
    - Provides context-aware responses based on the trainee's position in the VR environment and current inspection task
2. **Condition-Based Maintenance Training:**
    - Provides AI-powered assistance for interpreting maintenance indicators
    - Offers step-by-step guidance for predictive maintenance procedures
3. **Remote Support Integration:**
    - Enables real-time support during VR-based maintenance simulations
    - Provides context-aware responses considering the trainee's current task and environment
    - Delivers personalized guidance based on trainee expertise level and task complexity

As presented in D6.1, Task T4.3’s functionalities are relevant to various pilots of the project especially in use cases requiring technical training. For example, Pilot 1 (T6.2, KUKA pilot) is currently planning to integrate LLM functionalities into their use cases. Additionally, there are ongoing discussions to extend these integrations to Pilot 4 (T6.5, TAP pilot) and Pilot 5 (T6.6, SPACE pilot) to enable Generative AI’s capabilities for industrial training through the XR5.0 Training Platform. These efforts aim to ensure that the task supports real-time assistance and training in an agnostic manner across all pilots.

The LLM Engine's flexible architecture allows it to effectively support these diverse pilots while maintaining consistency in knowledge delivery and personalization capabilities across different pilot implementations. Currently, Task 4.3 is being validated in Pilot 3 (T6.4, EKSO pilot) and it is planned to empower the training platform of WP5.

3.6 Challenges and Limitations

Testing the RAG performance revealed promising results in delivering contextually relevant and accurate responses. However, several limitations were identified. First, the evaluation primarily focused on technical feasibility rather than comprehensive end-user testing. While integration with an AR app (provided by Task 4.5) validated the technical setup, it did not measure usability and user experience. The AR app successfully demonstrated query submission from Unity and responses via the LLM Chat Engine using proprietary knowledge retrieval, but the interface was optimized solely for validating integration rather than full-scale deployment. Future testing will emphasize in user feedback from pilot users to refine performance and optimise the system's logic. Addressing these constraints is essential to achieve higher TRLs.

3.7 Next Steps

To address the identified limitations, future efforts will focus on conducting user-centric evaluations to assess usability, response accuracy, and contextual relevance within XR applications. Iterative updates to the chat engine will be guided by user feedback. Furthermore, Task T4.3 will specify and define more precisely how to adapt the system for each pilot's unique requirements. While Pilot 3 (T6.3, EKSO pilot) documentation has been successfully incorporated into T4.3, the architectural design for Pilot 1 (T6.2, KUKA pilot) demands specialized knowledge modelling to handle complex robotic system documentation and maintenance procedures. The expansion to Pilot 4 (T6.5, TAP pilot) and Pilot 5 (T6.6, SPACE pilot) will require custom indexing strategies to effectively manage aerospace maintenance protocols and SPACE's technical specifications. These pilot-specific implementations will be coordinated through the XR5.0 Training Platform, ensuring a standardized yet flexible approach to industrial training.

Performance optimization will target the implementation of personalization features to adapt responses based on user progress, training contexts, and positioning in the VR/AR space, improving relevance and engagement. A monitoring framework, based on LLM observability tools, will be developed to track user interactions and refine the system continuously based on real-world usage data. Additionally, the system will incorporate external APIs and services (e.g., other AI systems developed in WP4) through LLM-orchestrated knowledge connectors, enhancing domain-specific capabilities across various industrial scenarios. Each pilot's unique data structure will inform the development of custom retrieval techniques, ensuring optimal performance for their specific use cases. By focusing on these enhancements, the system will progress towards higher TRLs, delivering robust AI-powered solutions that align with XR5.0 objectives.

4. AI MODELS FOR TRUSTED HUMAN-AI COLLABORATION

This section outlines the development and application of AI models tailored to achieve trusted human-AI collaboration within the XR5.0 framework. It outlines the explainable AI (XAI) methodologies and ontology-driven semantic approaches to address the integration of trustworthy and transparent AI outputs within the XR environments. The integration aims to enhance human understanding, build user trust and enable dynamic, context-aware decision-making by providing clear and interpretable AI outputs that align with user expectations and domain-specific requirements. This approach ensures that AI systems within XR environments are not only technically robust but also intuitive and adaptable, supporting seamless collaboration between humans and AI. The following parts of this section introduce an XAI suite and an ontology-driven semantic methodology for XAI, and how they can be demonstrated within the XR5.0 pilots.

4.1 Explainable AI Models

4.1.1 Brief Survey of State-of-the-Art

The manufacturing sector is experiencing a paradigm shift driven by the adoption of AI technologies, with image classification and Large Language Models (LLMs) emerging as main technologies. AI applications in manufacturing span a wide range of areas, including predictive maintenance, quality control, process optimization, supply chain management, and intelligent decision support systems. As these AI applications become more sophisticated, the need for Explainable AI (XAI) has grown.

Image classification has become a cornerstone of quality control and process optimization in manufacturing:

- Visual Quality Control: AI-powered image classification systems are being deployed to detect defects and anomalies in real-time, significantly improving product quality [34].
- Statistical Process Control: Advanced approaches like Visual Machine Learning Control (VMLC) combine classification with anomaly detection to enhance robustness in high-throughput manufacturing lines [35].
- Defect Classification: Machine learning algorithms are being used to classify defects, enabling more efficient screening of potentially faulty parts in manufacturing applications [36].

LLMs are transforming communication and decision-making processes in industrial settings. Fine-tuned LLMs, such as GPT-3.5 Turbo and Gemini 1.5 Pro, are being developed to assist human operators by providing targeted, accurate responses in real-time for product management and production line operations [37] [38].

Specific XAI Models and Techniques:

- LIME (Local Interpretable Model-agnostic Explanations) [39]: This technique has been applied to enhance the robustness of datasets against gradient evasion attacks in manufacturing image classification tasks.
- Grad-CAM (Gradient-weighted Class Activation Mapping) [40]: Used for creating visual explanations for decisions made by convolutional neural networks in manufacturing image classification [41].
- Integrated Gradients: This method attributes the prediction of a deep network to its input features, providing insights into the model's decision-making process [41].
- DeepDream Representations: Used in conjunction with multiple classifiers to enhance the interpretability of convolutional neural networks in histological image classification [42].

These XAI techniques are being integrated into AI enhanced manufacturing environments to create more transparent, adaptive, and user-centric systems that can seamlessly integrate with existing industrial processes while addressing the unique challenges of the manufacturing sector.

4.1.2 Role and Functionality

The XAI components developed for this project build upon solutions designed under the STAR (H2020) and HumAIne (HEU) initiatives, and they include complementary approaches to explainability:

- Glass-Box Models: Designed for inherently interpretable outputs through human-centric, concept-driven models that embed interpretability directly into the AI process.
- Post-Hoc Methods: Advanced tools that provide explanations for black-box models, allowing integration into existing workflows without retraining.
- Human-Centric GPT-Based Interaction: A conversational interface leveraging large language models (LLMs) for intuitive and interactive AI explanations.

These components ensure a balance of transparency, usability, and flexibility, catering to various industrial requirements and levels of domain expertise. Each component has been validated in controlled environments and early-stage pilot implementations, demonstrating their readiness for integration into diverse applications.

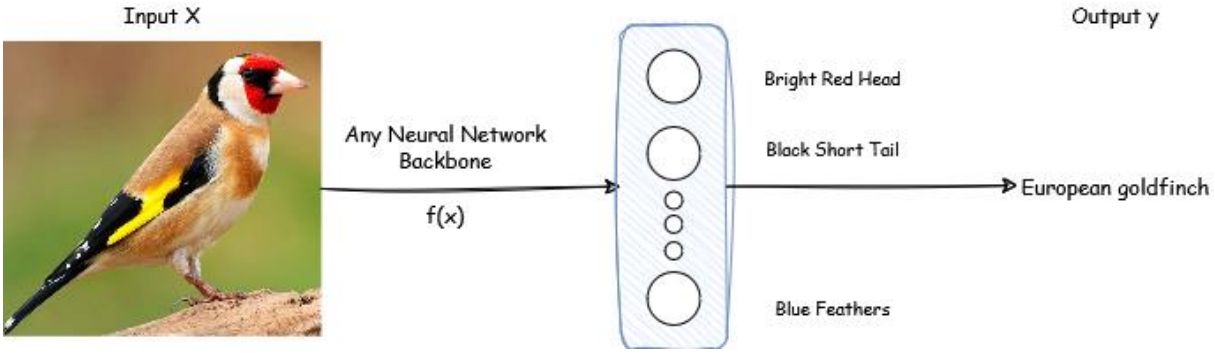

Figure 10 – CBM intermediate layer of neural network components

Glass-Box Models:

Concept Bottleneck Models (CBMs): CBMs enhance interpretability by aligning neural network components with human-understandable concepts. They achieve this using an intermediate layer, as shown in Figure 10, where neurons represent interpretable concepts guiding final predictions. This approach provides transparency while allowing users to intervene directly, incorporating their expertise to influence model predictions. The initial testing has been applied with CUB dataset[4], image datasets like CUB, are widely used in AI research, provide a reliable medium for showcasing the tool's versatility and its potential to handle both image and video-based applications within XR5.0.

However, traditional CBMs face challenges such as reliance on labelled data and a trade-off between accuracy and interpretability in complex datasets. These limitations hinder scalability and adoption. Data collection also risks inconsistency and bias, especially when human annotations define concepts. Advances with LLMs like GPT-3/GPT-4 address these issues by automating concept generation through dataset-specific prompts, eliminating manual labelling. When coupled with NSAI, CBMs go beyond their traditional role by enabling context-driven reasoning and rule generation. NSAI can leverage the extracted concepts from CBMs to transform them into formalized rules that represent the underlying logic of the data. These rules are then utilized within NSAI's symbolic reasoning framework,


```
{
    "Black_footed_Albatross": [
        "Dark plumage",
        "Long, narrow wings",
        "Hooked beak, often dark in color",
        "Large size with a wingspan",
        "Pale facial markings, especially around the eyes and base of the bill"
    ],
    "Laysan_Albatross": [
        "White head and body",
        "Dark gray or black wings",
        "Long, slender wingspan",
        "Pale pinkish bill with a hooked tip",
        "Dark eye with a light eye-ring"
    ],
    "Sooty_Albatross": [
        "Item: \"Sooty Albatross\"",
        "Dark gray or sooty brown plumage",
        "Long, narrow wings",
        "Slender, hooked beak",
        "Conspicuous white eye-ring",
        "Streamlined body",
        "Extensive wingspan"
    ],
    "Groove_billed_Ani": [
        "Long, slender body",
        "Glossy black feathers",
        "Prominent, thick, and curved groove-billed beak",
        "Long tail",
        "Dark eyes"
    ],
    "Crested_Auklet": [
        "Item: \"Crested Auklet\"",
        "Distinctive forward-curving crest on head",
        "Dark, slate-gray plumage",
        "Bright orange bill",
```


Figure 11 – JSON of raw concepts extracted using LLM from class-names

[4] Wah, Catherine, et al. "The caltech-ucsd birds-200-2011 dataset." (2011).

allowing for advanced decision-making processes that combine data-driven insights with structured, logical reasoning. For instance, in XR5.0 applications such as image and video recognition, CBMs can extract high-level features (e.g., “long beak,” “bright plumage”) from datasets using context provided by LLMs. NSAI can then transform these features into symbolic rules (e.g., “IF bright plumage AND long beak, THEN likely category: tropical bird”).

LLM-powered CBMs enable dynamic user interactions during inference, allowing users to adjust concept activations to refine predictions (human-in-the-loop approach). This transforms CBMs into collaborative platforms integrating domain expertise in real-time, bridging automated intelligence and human insight while fostering trust. This attribute of CBMs is highly correlated with AL component (Section 4.3).

In XR5.0, image recognition tasks are a key focus, and the inclusion of video data further underscores the tool's relevance. Video data, often pre-processed into frames treated as individual images, aligns seamlessly with the tool's capabilities. These frames contain intricate patterns such as shapes, colors, and textures, making them ideal for testing and evaluation.

Once data is uploaded, the tool uses class names from the dataset as targets for classification tasks. It queries a large language model (LLM) to extract key features for each class name, organizing these features in a structured JSON format (Figure 11). Since all classes belong to the bird superclass, some concepts appear across multiple classes. A multi-step filtering process refines the concept set by removing overly long, redundant, or irrelevant concepts and ensuring quality through a similarity threshold evaluation (Figure 12).

The refined concepts train CBM networks and produce predictions with concept importance attribution (Figure 13). Concepts negatively impacting predictions are marked with “NOT” to indicate their absence or inverse contribution, while positive contributors are listed with their impact. For instance, if “NOT a rosy breast” influences a decision, it reflects the a2bsence of this feature in the model's reasoning. Users can refine predictions by adjusting or nullifying a concept’s influence, promoting trust and control. This task will be coupled with NSAI (T4.2) offerings.

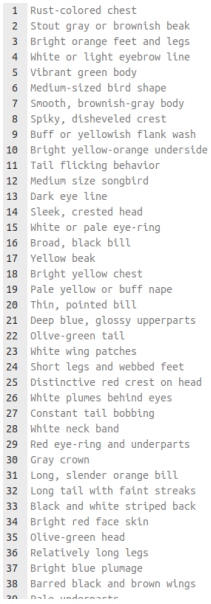

Figure 12 – Filtered concepts through iterative filtering process

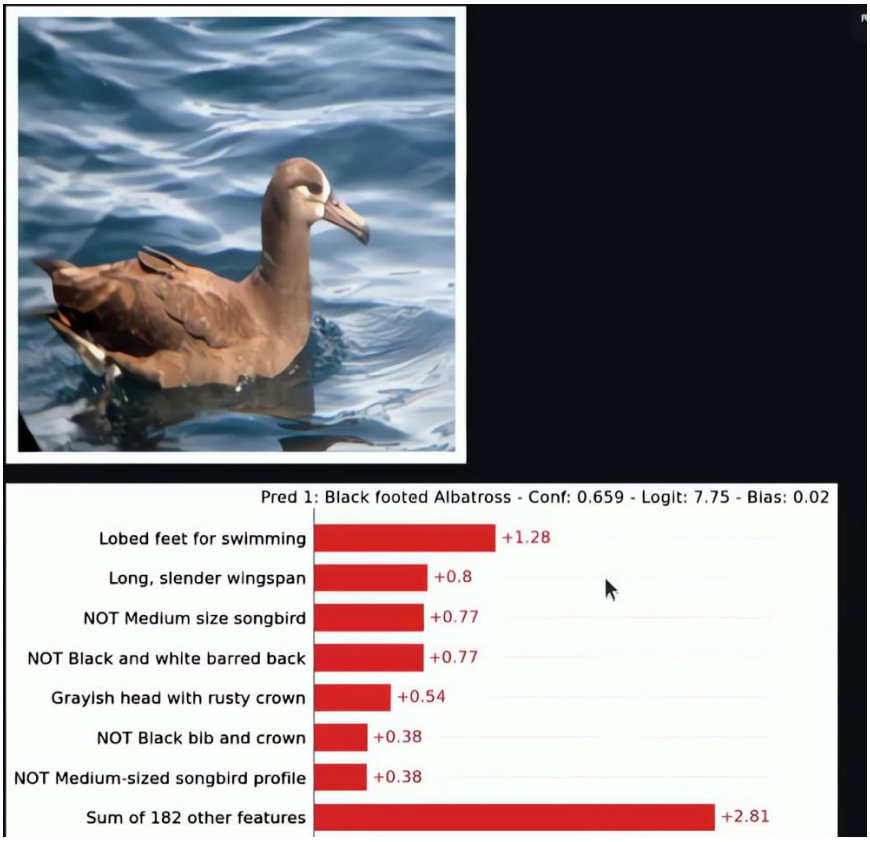

Figure 13 – CBM prediction for a single image with concept importance attribution

Post-Hoc XAI Methods:

XAI can significantly enhance industrial visual inspection systems by making their machine-learning components more transparent and interpretable. These systems utilize cameras or other sensor to capture images of products or processes, analyze them with machine learning algorithms and identify defects or abnormalities. However, machine-learning models often lack interpretability, making it challenging for operators to understand their reasoning, especially when errors occur. XAI techniques address this limitation by providing clear and interpretable insights into how decisions are made. The primary aim of incorporating XAI into visual inspection systems is to reduce the cost and time associated with manual inspections while allowing workers to focus on more meaningful, less repetitive tasks. By bridging the gap between AI and human operators, XAI will explain predictions made by CNNs that analyze image data for identifying defects in manufactured parts. These explanations will increase confidence in AI-driven decisions and assist workers in handling complex inspection tasks that require human judgment. The XAI component will provide explanations in terms of attribution scores, offering insights into the importance of each input feature for a particular prediction. For image data, this importance can be visualized as heatmaps where pixel contributions are displayed in varying colors, helping operators locate defects efficiently during manual revisions.

To implement this component, we are experimenting with various off-the-shelf XAI libraries, either as part of the final system or as benchmarks for comparison. Key libraries include:

- **LIME (Local Interpretable Model-agnostic Explanations):** Accessible via pip, LIME supports multiple data modalities, including vectors, images, and text. It is model-agnostic and interacts directly with the model's predict() method. While lightweight and not reliant on GPUs for standard use, it may face limitations with highly complex models or large datasets.
- **SHAP (Shapley Additive Explanations):** SHAP offers a model-agnostic framework to quantify feature contributions. While primarily designed for tabular and scalar data, it can be extended to

images through random masking techniques. However, its application to image classification can be less effective, as interpreting pixel-level SHAP values for complex visual patterns poses challenges. Nonetheless, SHAP serves as a useful baseline for explainability in timeseries and other modalities.

- **Grad-CAM (Gradient-weighted Class Activation Mapping):** Specifically designed for CNNs, Grad-CAM generates visual explanations by leveraging gradients flowing through the network. It highlights image regions that contribute most to the model's predictions (Figure 14). The pip package supports PyTorch and GPU-accelerated implementations, making it a powerful tool for deep learning models when access to weights and architectures is available.

These Python-based libraries integrate seamlessly with XR5.0's use cases, offering scalable and interpretable solutions for the project. By using these tools, Task T4.1 aims to create an XAI suite that not only enhances AI-driven visual inspection systems but also aligns with the transparency and trustworthiness requirements of Industry 5.0 applications.

Figure 14 – Example of Grad-CAM model heatmap output

Human-Centric XAI with LLMs:

The XAI suite for XR5.0 represents a significant advancement in human-centric explainability for black-box AI models, providing a multi-layered approach to interpretability. Designed to address the inherent opacity of complex AI systems, the suite combines advanced post-hoc methods with user-tailored explanations.

At its core, the XAI suite integrates widely used interpretability tools such as SHAP, LIME and Grad-CAM to decompose model predictions into feature-level contributions. These tools have been implemented within a LLM capabilities enhancing usability and accessibility. A distinguishing feature of the suite is its emphasis on personalization. By dynamically tailoring explanations to align with the user's domain knowledge, the suite ensures that insights are relevant and comprehensible.

In technical demonstrations, the XAI suite has successfully aggregated outputs from methods like SHAP and LIME, transforming them into actionable natural language insights. For instance, it can summarize key features influencing model predictions, visualizing their impacts as heatmaps or graphs, and providing an overarching explanation tailored to user queries (Figure 15).

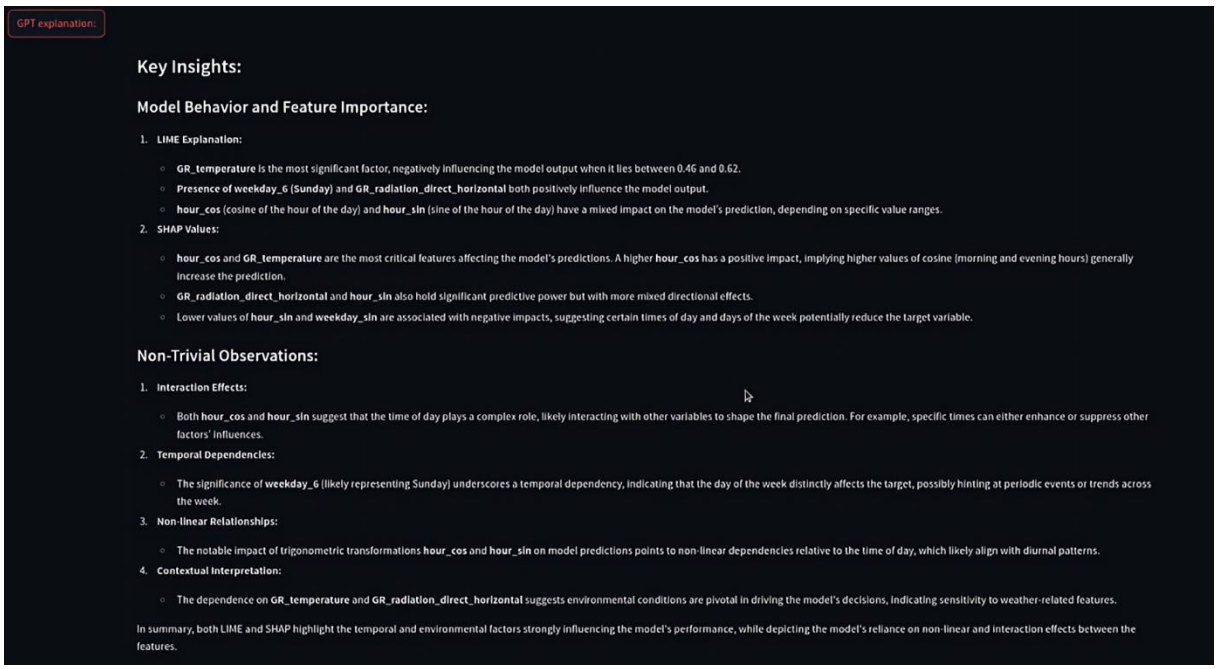

Figure 15 – Natural Language explanations based on output of post-hoc XAI (SHAP & LIME)

4.1.2.1 Component Status

The XAI suite, implemented as a stand-alone module for a technical demonstration, is currently at Technology Readiness Level (TRL) 4-5, indicating that it has been validated in controlled environments and through prior applications in other projects. The suite is designed to define and train explainable models, emphasizing versatility and broad applicability across diverse datasets and use cases. By being data-agnostic and use-case-agnostic, the XAI suite is adaptable for seamless integration into various XR5.0 scenarios, supporting the project's objectives of transparency and trust in AI-driven decisions.

The suite comprises four main components: a user interface, a data ingestion and storage service, a model training service, and an inference service. These components are implemented as Dockerized microservices that communicate through REST APIs, ensuring platform-agnostic and interoperable communication. The use of containerization encapsulates dependencies and runtime configurations, enabling consistent deployment, efficient scalability, and straightforward updates. At its current TRL, the suite's user interface, developed with Streamlit, provides an intuitive platform for uploading datasets, selecting models, and initiating training tasks. The data ingestion and storage service, implemented as a Flask microservice, manages dataset validation, cleaning, and secure storage, ensuring maintainability and scalability. The model training service, also Flask-based, handles resource-intensive training processes in an isolated environment, enabling experimentation and updates without impacting other system components. The inference service provides a scalable mechanism for generating predictions via REST endpoints, with the flexibility to adapt to increased demands or enhancements independently. The XAI suite's TRL4-5 status reflects its readiness for integration and deployment in XR5.0 pilot environments. Initial demonstrations will focus on Pilot 3 (predictive maintenance for smart water pipes), where it will analyze video data processed frame-by-frame to detect anomalies and provide actionable insights through augmented reality interfaces.

4.1.2.2 Evaluation

During the initial development of CBMs, the focus was on creating a general-purpose data/model agnostic glass-box solution adaptable to diverse domains. As a result, no formal evaluation was conducted at this stage. However, plans are in place to implement a comprehensive evaluation framework in future iterations to thoroughly assess the models' performance, interpretability, and practical utility using both quantitative and qualitative methods. Possibly the evaluation of CBMs will be covers by the NSAI evaluation. On the quantitative side, the evaluation will center on measurable improvements delivered by CBMs. This includes

examining predictive performance to ensure that the models maintain high accuracy while enhancing interpretability. Key metrics will include prediction accuracy, concept alignment scores, and model calibration, all of which measure how well the models align with predefined human-understandable concepts and maintain consistent, reliable predictions. Additionally, reductions in misclassified instances will be analysed, particularly in tasks where concept-based reasoning plays a critical role. Qualitative evaluation will focus on the user experience and the practical application of CBMs in real-world scenarios. Focus groups (comprising experts or the end-users) and questionnaires can play an essential role in understanding and assessing how well the models align with domain-specific knowledge and whether they capture (accurately enough) understandable/truthful concepts. Besides that, surveys can gather feedback on the intuitiveness of the system and areas for potential enhancements, setting an emphasis on human-centered aspects of CBM.

To assess the practicality and impact of the GPT-based XAI explainer, a survey was conducted with 30 professionals, featuring 12 questions designed to explore familiarity with AI, Machine Learning (ML), and Deep Learning (DL), and perceptions of XAI techniques. This effort provided critical insights into user needs and preferences, enabling refinement of the explainer's features. Participants evaluated explanations from traditional XAI methods like LIME, SHAP, and Grad-CAM alongside descriptions generated by the GPT explainer. Over 80% preferred the GPT-generated explanations for their clarity and accessibility, particularly in decision-making and image analysis tasks. The survey also revealed a gap in AI literacy, with over 70% of respondents rating their understanding of AI-based models below 60%, and only 30% actively using XAI techniques. Figures Figure *16* - Figure *18* provide further insights into preferences, showing variations between end-users and AI experts. While end-users favoured simpler explanations tailored to their use cases, experts preferred consistent, detailed outputs aligned with their technical understanding.

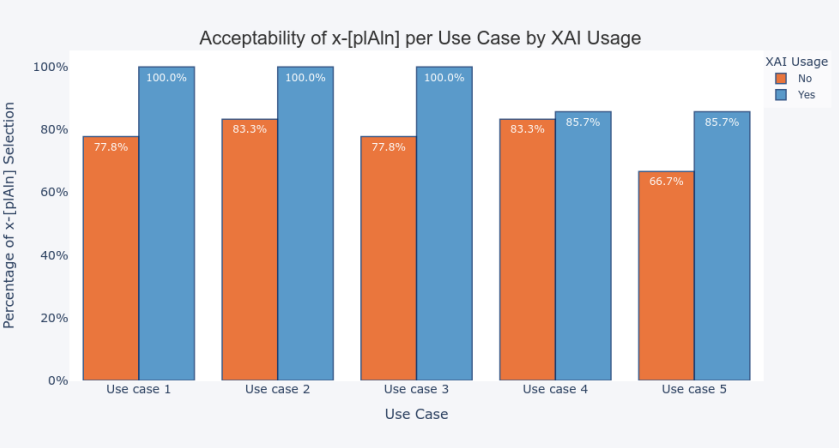

Figure 16 – Acceptability correlated with use case by XAI usage

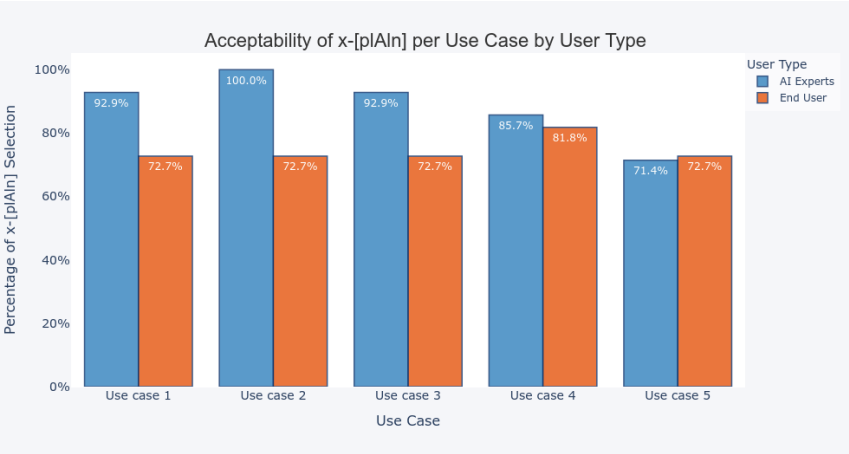

Figure 17 – Acceptability correlated with use cases by user type

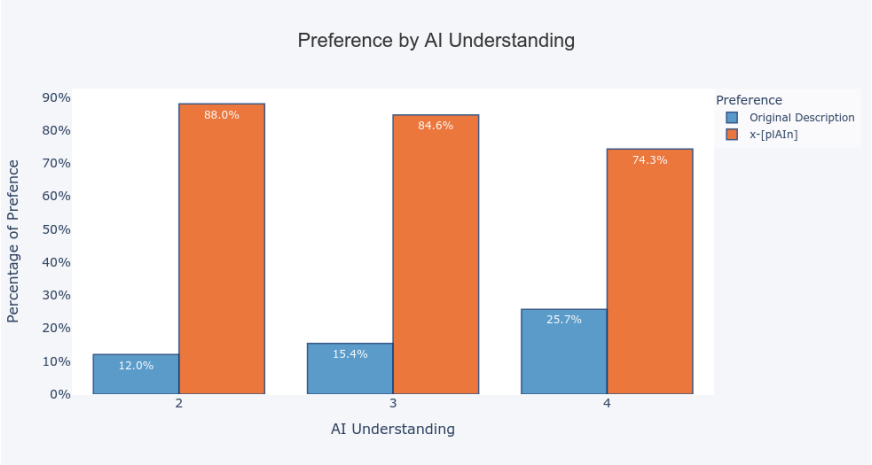

Figure 18 – Preferences correlated with self-reported comprehension of AI model outputs

4.1.3 Computational Resource Requirements

The computational requirements presented here represent typical configurations and are subject to change based on the size and complexity of the datasets, as well as specific application needs. For smaller datasets or simpler tasks, resource requirements can be significantly reduced, while large-scale deployments may demand additional resources. This balanced approach ensures resource efficiency while maintaining scalability and adaptability to the diverse needs of the XR5.0 project. The following estimations are based on image-based datasets and applications.

Glass-Box Models:

- Training: Requires a mid-range GPU (e.g., NVIDIA RTX 3060, 12 GB VRAM), 64 GB RAM, and 500 GB SSD. Training typically takes 12–24 hours for datasets with up to 50,000 samples.
- Inference: Operates efficiently on a CPU (16 GB RAM) or a low-end GPU (e.g., NVIDIA T4) for real-time tasks, with prediction times of <300 ms per instance.

Post-Hoc Methods:

- Training: Minimal resource requirements with a CPU-only setup (32 GB RAM); for visualization tasks, a low-end GPU (e.g., NVIDIA T4) is optional. Storage of 250 GB SSD is recommended. Explanation generation takes 1–2 hours for 10,000 samples.
- Inference: Runs on a CPU (16 GB RAM) with explanations generated in 0.5–1 second per instance.

Custom Explainability Tools:

- Training: Typically, CPU-based (32 GB RAM, 500 GB SSD) with no GPU required. Training time is around 4–8 hours, depending on dataset size and slicing complexity.
- Inference: Operates on a CPU (16 GB RAM) with prediction and analysis times of <300 ms per instance, supporting 100–200 predictions per second in batch mode.

4.1.4 Installation and Deployment Guidelines

Based on interactions with potential stakeholders during the co-creation meetings, the adoption and adaptation of tools like the XAI suite often require significant customization to align with existing systems and workflows. As a result, the suite's architecture has been redesigned to provide its functionality "as a service" through a modular, microservices-based API. This approach ensures the XAI suite can integrate flexibly with pilot-specific systems and tools without altering its core logic or methodology. Each API represents an independent microservice responsible for a specific function or set of functions. These microservices communicate as needed to complete their tasks. This architecture enhances flexibility and scalability, allowing each microservice to be developed, deployed, and managed independently. Additionally, this design accelerates development timelines, as microservices can be built in parallel and integrated later.

4.1.5 Demonstration Scenarios and Mapping to Pilots

Task T4.1 is focused on providing XAI models, primarily for image recognition tasks, for XR5.0's industrial and operational applications. The initial demonstrations will prioritize Pilot 3 (Task 6.4, EKSO pilot) on predictive maintenance for smart water pipes at EKSO, Pilot 1 (T6.2, KUKA pilot) on recognizing sensor models from images or videos, and Pilot 5 (T6.6, SPACE pilot) for enhancing edge device assembly and repair.

Pilot 1 will demonstrate the application of XAI in recognizing machine states and sensors from image or video data, allowing technicians to identify and assess machine states and components efficiently. This capability will streamline assembly line workflows by providing clear, interpretable visualizations of machine diagnostics and recommendations through AR-enabled tools.

In Pilot 3, XAI models will process video data converted into frame-by-frame images to detect anomalies in smart water pipes. Maintenance technicians will utilize AR interfaces to visualize malfunction indications and receive real-time insights on the infrastructure's condition, enabling precise interventions.

For Pilot 5 XAI will enhance training and assembly workflows by analysing image data to generate tailored instructions, ensuring technicians of varying skill levels receive adaptive and interpretable guidance.

Additionally, XAI will be integrated with human-digital twins under WP3 to analyse physiological data and predict human stress and fatigue levels, crucial for improving operator well-being. This predictive capability, coupled with interpretable insights, ensures that both human and operational factors are addressed.

4.1.6 Challenges and Limitations

The deployment and integration of the XAI suite within the XR5.0 pilots face several challenges and limitations, which could impact the demonstration scenarios and pilot outcomes:

Data Availability and Quality: A significant challenge lies in obtaining sufficient and representative datasets for training and evaluation. Many pilots require specific types of data (e.g., video frames for predictive maintenance in Pilot 3 or sensor recognition data for Pilot 1), which may not be readily available or accessible. Even when data is available, ensuring its relevance, completeness, and quality is a persistent issue.

Varying TRL Levels Across Components: An end-to-end solution for a pilot includes components with varying TRLs, ranging from TRL 4 to TRL 6-7. While higher-TRL components have been validated in near-operational environments, lower-TRL components require further development and testing. This discrepancy presents challenges:

- The integration of TRL 4 components with more mature parts of the system can delay adoption and deployment, as additional validation and refinement may be needed.
- Users and stakeholders may hesitate to trust or adopt solutions perceived as less mature, impacting the overall adoption of XR5.0 components.

4.1.7 Next Steps

The next phase of development for the XAI suite focuses on addressing these challenges while advancing its integration and applicability across XR5.0's pilots and components. Key next steps include:

Implementation and Validation in Pilots:

- Pilot 1 (Machine States and Sensor Recognition): Test the suite's application in identifying and diagnosing machine states and sensors from image and video data, streamlining workflows with clear, interpretable visualizations.
- Pilot 3 (Predictive Maintenance): Deploy and evaluate the suite's ability to process video data for anomaly detection, providing actionable insights through augmented reality interfaces for maintenance technicians.
- Pilot 5 (User-specific Instructions for Edge Device Assembly): Enhance training and edge device assembly by generating tailored, interpretable instructions, ensuring effective guidance for technicians of varying expertise levels.

Integration with WP3 Digital Twins:

Collaborate with WP3 to incorporate the XAI suite into human-digital twins, enabling the analysis of physiological data to predict stress and fatigue levels.

Collaboration with Related Tasks:

- Task T4.2 (NSAI): Leverage the context-driven reasoning capabilities of NSAI to transform extracted concepts from CBMs into symbolic rules, enhancing decision-making processes.
- Task T4.3 (GenAI): Utilize generative AI to dynamically generate content and scenarios, improving XAI explanations with richer, context-specific insights.
- Task T4.4 (Explainability Visualization): Refine the visualization of XAI outputs, ensuring they are intuitive and accessible for diverse user groups in XR5.0.

User-Centric Enhancements:

Further personalize XAI explanations to align with varying levels of domain knowledge, addressing gaps in AI literacy and fostering user engagement through the conversational GPT-based explainer.

Finaly, implement a rigorous evaluation framework combining quantitative metrics (e.g., accuracy, concept alignment, and calibration) and qualitative methods (e.g., user feedback and focus groups) to validate the suite's performance, interpretability, and usability in real-world applications.

4.2 Semantic Methods for Explainable AI

4.2.1 Brief Survey of State-of-the-Art

Ontologies:

An ontology [43] is a structured representation of knowledge, defining a set of concepts within a specific domain and the relationships between them. It includes *i)* classes (or concepts) that represent the general categories in that domain, *ii)* instances (or individuals) that are specific examples of the classes, *iii)* properties (or attributes) that define characteristics or data associated with classes or instances, *iv)* relationships that define how classes and instances interact with or relate to one another, *v)* rules and constraints that specify logical conditions or constraints to ensure consistency and correctness.

Ontologies can play a significant role in enhancing XAI models by offering structured frameworks that enable AI models to provide transparent, interpretable, and domain-relevant explanations, contributing in various aspects:

1. **Knowledge Representation**: Ontologies define domain concepts, relationships, and constraints, enabling AI systems to operate within a structured and interpretable framework [44] [45]. By mapping decisions to well-defined knowledge structures, ontologies ensure that the logic behind AI operations can be readily understood and explained [46].
2. **Reasoning and Inference**: Ontologies support reasoning engines that provide step-by-step explanations of the logical inferences made by the system. They embed domain rules and constraints, enabling AI systems to detect and explain contradictions, ambiguities, or unexpected outcomes with logical precision [47].
3. **Traceability**: Ontologies enhance the traceability of AI decisions by recording the sources and justifications for data and conclusions [48] [49]. They can enable step-by-step explanations of how inputs are transformed into outputs, allowing users to follow the decision-making process in a clear and structured manner.
4. **User-Centric Explanations**: Ontologies can enable explanations to be customized to the user's level of expertise by using domain-specific terminology and examples [50]. This capability allows AI systems to deliver interactive and contextualized explanations that are tailored to the needs of different users, enhancing their understanding and trust [51].

4.2.2 Role and Functionality

Context Awareness Framework:

Context Awareness Framework [52] is a structured system designed to collect, process, and utilize contextual data to enable intelligent decision-making and adaptive services. It operates by integrating data from various sources, interpreting its meaning within a defined model, and delivering contextual information to applications or services in real time. Figure 19 illustrates the components of the framework. It integrates a structured approach for managing contextual data, grounded in a formal *Context Ontology*. The framework processes data in a pipeline that includes *Context Monitoring*, *Context Extraction*, and *Context Provision*, with outputs feeding into dynamic services and rule-based decision-making systems.

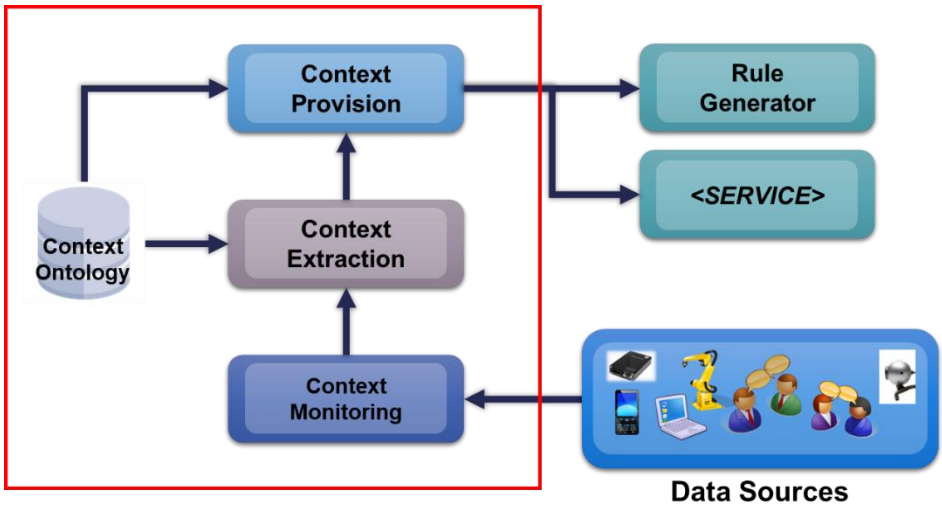

Figure 19 – Context Awareness Framework components

Context Monitoring: The objective of the Context Monitoring service is to receive raw data and provide aggregated monitored data. To achieve this, the Context Monitoring service allows monitoring of systems via different interfaces. It is therefore able to "standardise" and correlate the data from distinct systems (e.g. map actions from file systems and Web-Services) which later serves as a basis for identification and extraction of contexts.

The main component of the Context Monitoring service is the modular monitoring process, used for all monitoring services with an extendable and configurable standardised process. The process is three-parted and contains the following modules:

- *Monitoring* module, which contains all services to monitor systems and devices in enterprises, via the Data Access Layer. The distributed monitoring services also call back this module with their gathered information. The monitoring services can be extended and configured for different systems and do not need to comply with other modules.
- *Parser* module, which contains content parser for the different possible data captured by the monitoring services. The parser offers access to the diverse data possible to be interacted with and therefore monitored. It provides the access for the analyser and may parse available environmental properties.
- *Analyser / Monitoring Data* builder module, which correlates the monitored content (and maybe environmental properties) and constructs the standardised monitoring data to be stored and handed over to the Context Monitoring / Extraction service, or any other service that needs this information.
- *Monitoring user behaviour* module, which allows monitoring of non-structured sources of information (that are not collected by a system/sensor), from the user environment, via different interfaces from the Data Access Layer. It is therefore able to standardise and correlate the (e.g. usage) data from distinct systems (e.g. map actions from file systems and Web-Services) which later serves as a basis for identification and Extraction of Contexts.

Context Extraction: The objective of the Context Extraction service is to extract and identify high-level contexts from the monitored data in the Context Monitoring service (systems and sensor raw data) and this information is used to help being more productive and/or economic within (semi-) automated environments, as well as for further knowledge enhancement.

The process identifies the context from the monitoring data provided by the Context Monitoring Service (Context Identification), manipulates it through different types of reasoning techniques (Context Reasoning), and provides the refined identified context to further services (Context Provision).

The purpose of Context Reasoning is to produce more accurate and meaningful context out of the identified context. Three types of context reasoning are provided, namely *Ontological Context Reasoning*, *Rule-Based Context Reasoning* and *Statistical Context Reasoning*.

- Ontological Context Reasoning explores the semantics of the OWL ontology language and the definitions in the context model to infer deductive results out of the identified context.
- Rule-Based Context Reasoning applies user defined domain specific rules to infer new contextual knowledge from the existing contextual knowledge. Thereby the Jena rule engine is used[5].
- The purpose of Statistical Context Reasoning is to determine the current context based on the available contextual information and historical contexts. Statistic Context Reasoning does not rely on strict logical rules but instead tries to correlate information into possible relations, as suggested by the empirical data.

Context Provision: The extracted context is shared with downstream systems, such as rule generators or adaptive services, to enable dynamic responses and intelligent behaviour. Rule generators use the

[5] https://jena.apache.org/documentation/inference/#rules

provisioned context to create operational rules or triggers. Services leverage these rules to personalize experiences, optimize processes, or automate actions based on the current environment.

4.2.2.1 Component Status

The Context Awareness Framework is based on the services developed within the EU H2020 project SmartClide[6] and corresponds to TRL 6. The baseline component - smartclide-context[7] - is available as open-source from the Eclipse OpenSmartCLIDE[8] project.

4.2.2.2 Evaluation

The Context Awareness Framework extracts and interprets environmental, operational, user-related or situational context. This context can be used to generate tailored, situation-specific explanations for AI decisions. Furthermore, by monitoring and extracting real-time context, the framework can identify relevant contextual features influencing an AI decision, helping XAI systems explain why certain features were considered important in a decision and emphasizing the relationship between the input context and the output.

The work of the Context Awareness Framework originated in the H2020 SAFIRE project[9]. Within SAFIRE, the initial framework was developed. In the project it was used to monitor situations within manufacturing processes, such as electrical discharge machining and process industry. The knowledge generated by the Context Awareness Framework was used in other SAFIRE components to better support the reconfiguration of production lines and/or machines.

In the SmartCLIDE project the Context Awareness Framework was used for gathering knowledge about a software development lifecycle within the SmartCLIDE IDE. The SmartCLIDE project was successfully finalised in 2023 and validated in three pilot cases. The results from the project were transferred into an Eclipse open-source project – Eclipse OpenSmartCLIDE.

In the AI4Work HE project[10], the Context-Awareness Framework is used for the monitoring of AI systems and/or robots in order to provide information to make better decisions for sharing work between humans and AI/robots. The AI4Work services are currently being underdeveloped and the Early Prototype results are expected in mid-2025.

4.2.3 Computational Resource Requirements

The computational resource requirements for the system will vary depending on the specific pilot implementation. At this stage, only approximate estimates are available, with detailed specifications to be provided in subsequent iterations of the deliverable. The preliminary resource requirements are as follows:

- Num. CPUs >=1
- Memory >= 1GB
- Storage >= 1GB

These baseline values are subject to refinement based on the complexity and scale of the final implementation.

4.2.4 Installation and Deployment Guidelines

Preconditions to build and run Context Monitoring and Extraction Services:

To build and run Context Monitoring and Extraction, the following software is required:

- Java (at least version 11)
- Apache Maven (at least version 3.5.4)

[6] https://cordis.europa.eu/project/id/871177
[7] https://github.com/eclipse-opensmartclide/smartclide-context
[8] https://projects.eclipse.org/projects/ecd.opensmartclide
[9] https://cordis.europa.eu/project/id/723634
[10] https://ai4work.eu

- Docker (for running tests and deploying Context Handling on the SmartCLIDE cluster)
- docker-compose (for running local sample instance only)

How to build Context Monitoring and Extraction:
Context Monitoring and Extraction can be built using maven with the following command:

```
mvn install
```

In order to build and push a container image that can be deployed, the following command can be used:

```
mvn install
mvn jib:build -pl context -Djib.to.image="${IMAGE_NAME:IMAGE_TAG}" -
Djib.to.auth.username="${CONTAINER_REGISTRY_USERNAME}"              -
Djib.to.auth.password="${CONTAINER_REGISTRY_TOKEN}"
```

How to run Context Monitoring and Extraction:
A sample configuration and docker-compose file can be found in the samples folder.

Run the sample with the following command:

```
docker-compose -f samples/docker-compose.yml up
```

4.2.5 Demonstration Scenarios and Mapping to Pilots

Within the scope of XR5.0, it is planned that the Context Awareness Framework will support the XAI suite with context monitoring, context extraction and rule generation, providing the project pilots with ontology-based semantic alignment between pilot specific data and the respective XAI models.

The Context Awareness Framework extracts and interprets environmental, operational, user-related or situational context. This context can be used in particularly to generate tailored, situation-specific explanations for AI decisions. By monitoring and extracting real-time context, the framework can identify relevant contextual features influencing an AI decision, emphasizing the relationship between the input context and the XAI output. The Context Monitoring component can support that explanations adapt to changes in the environment in real-time. The XAI suite can explain their decisions relative to current conditions, enhancing trust in dynamic and uncertain environments.

The Rule Generator component in the framework can be used to create rules that explain AI behaviour in terms of context-sensitive actions. The outputs of the XAI suite can be linked to pre-defined or generated rules within a specific context and make the reasoning process traceable and explainable. The generated rules can provide structured reasoning for the XAI suite, and enable it to explain its outputs more effectively.

More specifically, the Context Awareness Framework can support the XAI applications in the XR5.0 pilots described in Section 4.1.5 by enhancing XAI models with context-specific information:

- **Pilot 1 (Machine states and sensor recognition at KUKA):** The Context Awareness Framework can enhance machine states and sensor recognition by identifying the context in which they are being used, such as the specific operation of the robot or machine-specific workflows. Such contexts can improve the recognition accuracy by pre-filtering irrelevant scenarios or parameters.
- **Pilot 3 (XAI for predictive maintenance of Smart Water Pipes at EKSO):** The Context Awareness Framework can enhance anomaly detection by enabling the XAI models to explain the anomalies in terms of the context in which they occurred, providing a more comprehensive insight for the technicians. This can potentially be used to assess the severity of anomalies by combining contextual factors with the detected anomaly, and thus to prioritize responses to critical issues.
- **Pilot 5 (Edge device assembly and repair at SPACE):** The Context Awareness Framework can potentially be used to extract task-specific information, such as the complexity of assembling certain components, indicating the number of steps, precision requirements, and potential pitfalls. The XAI model can use this context to adjust the level of detail in the instructions it provides. The extracted

context on task complexity can be useful for the XAI models to diagnose and explain the root cause of these errors.

4.2.6 Challenges and Limitations

Several challenges and limitations should be acknowledged and addressed when integrating the Context Awareness Framework with the XAI suite within the scope of the XR5.0 pilots:

Ontology Development and Semantic Alignment: Developing and maintaining a robust ontology for semantic alignment between pilot-specific data and XAI models is time-consuming and requires domain-specific expertise. Overly complex ontologies may introduce inefficiencies in processing, while simplistic ones may fail to capture critical nuances. Ensuring semantic alignment across diverse pilot scenarios (e.g., smart water pipes, sensor recognition, and edge device assembly) is challenging due to the diversity of contexts. Poorly designed ontologies could result in misaligned data, reducing the effectiveness of context-driven explanations and rule generation.

Data Quality and Context Accuracy: The accuracy of the extracted context depends heavily on the quality and reliability of input data from sensors, devices or user interactions. Noisy, incomplete or unreliable data can lead to incorrect context extraction and rule generation. Erroneous context extraction can result in misleading rules and explanations, which reduce trust in the system.

Scalability in Dynamic Environments: In dynamic systems, context is constantly changing, requiring the framework to scale and adapt quickly to process large amounts of contextual data in real time. Processing delays may arise due to the complexity of monitoring, extracting, and reasoning about context. Scalability issues can occur in large-scale systems with numerous sensors or data sources. Delays in real-time explanations can frustrate users or render the explanations obsolete in fast-changing scenarios.

4.2.7 Next Steps

Implement Pilot-Specific Ontologies: The first step is to engage experts in each pilot, sensor systems (Pilot 1, T6.2, KUKA), smart water infrastructure (Pilot 3, T6.4, EKSO) and edge device assembly (Pilot 5, T6.6, SPACE), to ensure that ontologies are tailored to domain-specific requirements, using a modular approach and ensuring each pilot has a dedicated and manageable ontology that addresses pilot-specific needs.

Implement Pilot-Specific Context Monitors: After the definition of pilot-specific ontologies, the next step is to design and deploy monitoring systems to capture relevant contextual data from sensors, devices, and user interactions. This process requires that the implemented monitors can handle dynamic environments and provide real-time inputs for further processing.

Implement Pilot-Specific Context Extraction: After the implementation of pilot-specific context monitor, the next step is to process and extract meaningful context from the monitored data.

Implement Pilot-Specific Rule Generation: Using the pilot-specific contexts that are extracted, context-sensitive rules will be generated. These rules should link contextual data to explanations that are inferenced based on the pilot-specific contexts, ensuring relevance and precision for the XAI suite's recommendations and explanations.

5. CONCLUSIONS

Deliverable D4.1 has demonstrated significant advancements in innovative AI methodologies for XR-based human-AI collaboration to achieve the vision of XR5.0, emphasizing trustworthiness and efficiency across industrial applications. Through the outcomes of Tasks T4.1, T4.2, and T4.3, the deliverable presented different advanced AI paradigms to enhance operational processes and industrial workforces in digitally augmented environments.

Task T4.1 has presented an Explainable AI (XAI) suite that ensures transparency and interpretability in industrial settings. By leveraging state-of-the-art methodologies, such as Concept Bottleneck Models, SHAP, and Grad-CAM, the XAI models provide user-centric, context-aware insights to foster trust and improve decision-making. The inclusion of the Context Awareness Framework has the potential to further strengthen the suite by enabling real-time, context-driven explanations. XAI applications are still being explored and are in the early stages of pilot implementation, including predictive maintenance, sensor recognition, and instructions for edge device assembly and repair.

Task T4.2 has demonstrated substantial progress in advancing Neurosymbolic AI (NSAI) and Active Learning (AL) components for XR-enabled human-AI collaboration. By synergizing neural networks and symbolic reasoning, NSAI enhances explainability and adaptability, while AL introduces a human-in-the-loop approach to improve algorithmic accuracy and user understanding. Both components have demonstrated use cases for early pilot implementations, such as object detection and anomaly classification.

Task T4.3 has demonstrated Generative AI capabilities for XR environments focusing on the creation of context-aware, domain-specific solutions for industrial collaboration and training. The use of large language models (LLMs) and vector databases demonstrate practical applications in visual inspection and anomaly detection training, condition-based maintenance and remote support integration within the XR5.0 pilots, improving training efficiency and operational outcomes.

Across these tasks, the deliverable highlights the integration of explainable, adaptive and Generative AI models within XR platforms, creating a versatile AI toolkit to meet the diverse needs of XR5.0 pilots. This cohesive approach aims to deliver transformative solutions that enhance human-AI collaboration, promote transparency and trust, and drive industrial innovation.

Future work across Tasks T4.1, T4.2 and T4.3 will focus on advancing the XAI suite, NSAI and AL integration, and custom Generative AI-based solutions to address the specific needs of XR5.0 pilots. T4.1 will work on early pilot implementations and focus on extending the XAI suite's applicability across different pilots. This task will strengthen collaboration with WP3 to integrate human-digital twins for physiological data analysis and incorporate insights from NSAI and Generative AI to enrich explanations and decision-making. T4.1 will also refine visualization strategies for robust performance validation. T4.2 will integrate NSAI with AL to enhance object detection and outlier analysis. Future efforts will focus on full pilot implementations and real-time operational capabilities in XR environments to ensure seamless communication between AI and XR components for effective anomaly detection and decision support, and extend their application to other X5.0 pilots.

T4.3 will emphasize user-centric improvements, adapting Generative AI-based solutions to diverse XR5.0 pilot needs through custom indexing and knowledge modelling for specific domains like robotics and industrial training. Performance optimization will include personalized responses, LLM-based monitoring frameworks and API integrations to extend capabilities across XR5.0 applications. Future efforts will prioritize adapting the architecture to meet the unique needs of each pilot, optimize personalization features, and integrate advanced monitoring frameworks in order to enhance system robustness and relevance to achieve higher TRLs and deliver robust, scalable solutions.

The continued focus on pilot-specific refinements, full pilot implementations and pilots validation will ensure the successful deployment of these technologies and bridge the gap between cutting-edge AI and XR

research in real-world applications. The outputs of D4.1 will contribute to further developments and refinements in advanced AI paradigms for human-AI collaboration to be presented in D4.2, the second and final version (v2) of D4.1, which will follow on M27.

REFERENCES

[1] H. Kautz, "The Third AI Summer: AAAI Robert S. Engelmore Memorial Lecture," *AIMag, vol. 43, no. 1,* pp. 105-125, March 2022.

[2] H. Zhang and T. A. Yu, "Deep Reinforcement Learning: Fundamentals, Research and Applications," p. 391–415, 2020.

[3] M. Hersche, M. Zeqiri, L. Benini, A. Sebastian and A. Rahimi, "A neuro-vector-symbolic architecture for solving raven's progressive matrices," *Nature Machine Intelligence,* p. 1–13, 2023.

[4] C. Zhang, F. Gao, B. Jia, Y. Zhu and S.-C. Zhu, "Raven: A dataset for relational and analogical visual reasoning," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,* p. 2019, 5317–5327.

[5] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli and J. Tenenbaum, "Neural-symbolic vqa: Disentangling reasoning from vision and language understanding," *Advances in neural information processing systems,* p. 31, 2018.

[6] K. Yi, C. Gan, Y. Li, P. Kohli, J. Wu, A. Torralba and J. B. Tenenbaum, "Clevrer: Collision events for video representation and reasoning.," *In International Conference on Learning Representations,* 2020.

[7] R. Riegel, A. Gray, F. Luus, N. Khan, N. Makondo, I. Y. Akhalwaya, H. Qian, R. Fagin, F. Barahona, U. Sharma and e. al., "Logical neural networks," 2020. [Online]. Available: https://arxiv.org/abs/2006.13155.

[8] S. Badreddine, A. d. Garcez, L. Serafini and M. Spranger, "Logic tensor networks," *Artificial Intelligence, 303: 103649,* 2022.

[9] H. Dong, J. Mao, T. Lin, C. Wang, L. Li and D. Zhou, "Neural logic machines," *In International Conference on Learning Representations,* 2019.

[10] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen and X. Wang, "A Survey of Deep Active Learning," *ACM Comput. Surv. Vol. 54, Issue 9, Article 180,* p. 40, December 2022.

[11] S. Tong, "Active learning: theory and applications," 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:62018951.

[12] D. A. Gudovskiy, A. Hodgkinson, T. Yamaguchi and S. Tsukizawa, "Deep active learning for biased datasets via fisher kernel self-supervision," *In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE,* p. 9038–9046, 2020.

[13] Y. Huang, Z. Liu, M. Jiang, X. Yu and X. Ding, "Cost-effective vehicle type recognition in surveillance images with deep active learning and web data," *IEEE Transactions on Intelligent Transportation Systems 21, 1,* pp. 79-86, 2020.

[14] D. Feng, X. Wei, L. Rosenbaum, A. Maki and K. Dietmayer, "Deep active learning for efficient training of a LiDAR 3D object detector," *In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium. IEEE,* pp. 667-674, 2019.

[15] L. Holmberg, D. P. and P. Linde, "A Feature Space Focus in Machine Teaching," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Austin, TX, USA, 2020.

[16] B. Settles, "Active Learning Literature Survey," 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:324600.

[17] S. Amershi, M. Cakmak, W. Knox and T. Kulesza, "Power to the People: The Role of Humans in Interactive Machine Learning," *AI Mag. Vol. 35,* pp. 105-120, 2014.

[18] R. Munro, Human-in-the-Loop Machine Learning: Active learning and annotation for human-centered AI, Manning, 2021.

[19] G. Ramos, C. Meek, P. Simard, J. Suh and S. Ghorashi, "Interactive Machine Teaching: A Human-Centered Approach to Building Machine-Learned Models," *Human–Computer Interaction 35 (5–6),* p. 413–51, 2020.

[20] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos and . al., "Human-in-the-loop machine learning: a state of the art," *Artificial Intelligence Review 56,* p. 3005–3054, 2022.

[21] S. Berg, D. Kutra, T. Kroeger and . al. , "ilastik: interactive machine learning for (bio)image analysis," *Nature Methods 16,* p. 1226–1232, 2019.

[22] Kellenberger B., Tuia D. and Morris D. , "AIDE: Accelerating image-based ecological surveys with interactive machine learning. ," *Methods in Ecology and Evolution, Vol. 11, No. 12,* no. Wiley Online Library, pp. 1716-1727, 2020.

[23] M. Shao and e. al., Survey of different large language model architectures: Trends, benchmarks, and challenges, IEEE Access, 2024.

[24] M. Ventures, "The state of generative AI in the enterprise.," 2024. [Online]. Available: https://menlovc.com/2024-the-state-of-generative-ai-in-the-enterprise/.

[25] X. L. Dong, "The Journey to A Knowledgeable Assistant with Retrieval-Augmented Generation (RAG)," *Proceedings of the 17th ACM International Conference on Web Search and Data Mining,* 2024.

[26] S. Packowski and e. al., "Optimizing and Evaluating Enterprise Retrieval-Augmented Generation (RAG): A Content Design Perspective," 2024. [Online]. Available: https://arxiv.org/abs/2410.12812.

[27] T. Kagaya and e. al., "RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents," 2024. [Online]. Available: https://arxiv.org/abs/2402.03610.

[28] Y. Tong and e. al., "MS2Mesh-XR: Multi-modal Sketch-to-Mesh Generation in XR Environments," 2024. [Online]. Available: https://arxiv.org/pdf/2412.09008.

[29] J. Xiang and e. al., "Structured 3D Latents for Scalable and Versatile 3D Generation," 2024. [Online]. Available: https://arxiv.org/html/2412.01506.

[30] S. J. Uddin, A. Alex and T. Mahzabin, "Harnessing the power of ChatGPT to promote Construction Hazard Prevention through Design (CHPtD)," *Engineering, Construction and Architectural Management,* vol. Emerald, 2024.

[31] Z. Bahroun and e. al., "Transforming education: A comprehensive review of generative artificial intelligence in educational settings through bibliometric and content analysis," *Sustainability 15(17), 12983,* 2023.

[32] R. K. Malviya, V. Javalkar and R. Malviya, "Scalability and Performance Benchmarking of LangChain, LlamaIndex, and Haystack for Enterprise AI Customer Support Systems," *IJGIS Fall of 2024 Conference. The New World Foundation,* 2024.

[33] D. Ru and e. al., "Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation," 2024. [Online]. Available: https://arxiv.org/html/2408.08067.

[34] G. Makridis and e. al., "XAI enhancing cyber defence against adversarial attacks in industrial applications," *2022 IEEE 5th International Conference on Image Processing Applications and Systems (IPAS),* pp. 1-8, 2022.

[35] P. Mascha, "VMLC: Statistical Process Control for Image Classification in Manufacturing," in *Proceedings of the 15th Asian Conference on Machine Learning*, PMLR, 2024, pp. 866-881.

[36] L. A. Neves and e. al., "Classification of H&E images via CNN models with XAI approaches, deepdream representations and multiple classifiers," in *International Conference on Enterprise Information Systems - ICEIS*, SciTePress, 2023.

[37] P. Bidyarthi and e. al., "Improving Bangla Regional Dialect Detection Using BERT, LLMs, and XAI," *2024 IEEE International Conference on Computing, Applications and Systems (COMPAS),* pp. 1-6, 2024.

[38] A. Eslaminia and e. al., "FDM-Bench: A Comprehensive Benchmark for Evaluating Large Language Models in Additive Manufacturing Tasks," 2024. [Online]. Available: https://arxiv.org/abs/2412.09819.

[39] M. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier," 2016. [Online]. Available: https://arxiv.org/abs/1602.04938.

[40] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," *International Journal of Computer Vision,* vol. Springer Science and Business Media LLC, no. 128, p. 336–359, 2019.

[41] P. H. Rahmath, K. Chaurasia and A. Gupta, "Unlocking Interpretability: XAI Strategies for Enhanced Insight in GNN-Based Hyperspectral Image Classification," *2024 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI),* 2024.

[42] L. A. Neves and e. al., "Classification of H&E images via CNN models with XAI approaches, DeepDream representations and multiple classifiers," *25th International Conference on Enterprise Information Systems,* 2025.

[43] T. Gruber, "Ontology," in *Encyclopedia of Database Systems*, Boston, MA, Springer US, 2009, pp. 1963-1965.

[44] A. B. Arrieta and e. al., "Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities, and Challenges toward Responsible AI," *Information Fusion, 58,* p. 82–115.

[45] I. H. Sarker and A. Kayes, "A Semantic-Based Framework for Explainable Artificial Intelligence," *Artificial Intelligence Review, 55(4),* p. 3075–3098.

[46] Z. Zhou and M. Chen, "Ontology-Guided Machine Learning for Explainability," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 2019.

[47] C. D'Amato, F. Esposito, N. Fanizzi and T. Lukasiewicz, "Ontological Reasoning for Explainable Artificial Intelligence," *Journal of Applied Ontology, 15(2),* p. 105–124, 2020.

[48] M. Gruninger and M. S. Fox, "Methodology for the Design and Evaluation of Ontologies," *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI),* 1995.

[49] F. Wang and A. Wong, "Traceability and Explainability in AI Systems Using Ontologies," *Artificial Intelligence and Ethics Journal, 9(3),* p. 205–220, 2017.

[50] Y. Gil and B. Selman, "A 20-Year Community Roadmap for Artificial Intelligence Research in the US," *AI Roadmap 2019,* 2019.

[51] E. Brunk, C. Schulte and A. Wichmann, “Towards User-Centric Explanations in AI: Ontologies as a Bridge,” *Journal of Artificial Intelligence Research,* vol. 65, p. 207–228, 2019.

[52] S. Scholze, J. Barata and D. Stokic, “Holistic Context-Sensitivity for Run-Time Optimization of Flexible Manufacturing Systems,” *Sensors,* vol. 17(3), p. 455, 2017.

[53] S. e. a. Packowski, „Optimizing and Evaluating Enterprise Retrieval-Augmented Generation (RAG): A Content Design Perspective,” 2024. [Online]. Available: https://arxiv.org/abs/2410.12812.

[54] T. Kagaya and e. al., “RAP: Retrieval-Augmented Planning with Contextual Memory for Multimodal LLM Agents,” 2024. [Online]. Available: https://arxiv.org/abs/2402.03610.