

MASTER'S THESIS 2026

Context-Aware and Explainable Object Instantiation in XR

Martin Jernstedt, Julia Blomgren

Elektroteknik
Datateknik

ISSN 1650-2884

LU-CS-EX: 2026-49

DEPARTMENT OF COMPUTER SCIENCE

LTH | LUND UNIVERSITY



EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2026-49

**Context-Aware and Explainable Object
Instantiation in XR**

Kontextmedveten och Förklaringsbar
Objektinstansiering i XR

Martin Jernstedt, Julia Blomgren

Context-Aware and Explainable Object Instantiation in XR

Martin Jernstedt
ma0230je-s@student.lu.se

Julia Blomgren
ju6140bl-s@student.lu.se

June 30, 2026

Master's thesis work carried out at Ericsson Research.

Supervisors:

Saeed Bastani, saeed.bastani@ericsson.com
Leire Amenabar, leire.amenabar@ericsson.com
Sule Tekkesinoglu, sule.tekkesinoglu@cs.lth.se

Examiner: Elizabeth Bjarnason, elizabeth.bjarnason@cs.lth.se

Abstract

AI solutions in Extended Reality (XR) have the capability to instantiate 3D objects automatically, enabling seamless integration of virtual objects into real environments. At the same time, existing solutions are limited in their consideration of context. Moreover, explanations for the various reasons that lead to specific 3D object generation and placement decisions have not been studied. We developed a context-aware and explainable 3D object instantiation solution for XR, incorporating scene and user context when generating and placing objects. Our solution employs generative AI, including a vision-language model for context-aware generation and placement, combined with a feature attribution loop that produces fidelity-scored explanations of system decisions. Our system demonstrates context-awareness in both generating and placing objects, as assessed through both quantitative and qualitative evaluations. Users report a preference for explanations, both for trust and reliability reasons. Our solution contributes toward advancing more personalized and transparent XR systems, with potential applications across user-centered domains that rely on interactive world-building.

Keywords: XR, XAI, Explainable AI, Object Placement, Generative AI, Object Detection, Unity, TRELLIS, Cube R-CNN, ChatGPT

Acknowledgements

Firstly, we would like to thank our supervisors. To Sule Tekkesinoglu, thank you for giving us your guidance and feedback, elevating our work to a professional standard. To Saeed Bastani and Leire Amenabar, thank you for giving us the opportunity to explore freely, while giving us the tools and expertise to move forward. To our examiner, Elizabeth Bjarnason, thank you for giving us the opportunity to conduct this research, and giving us your insight along the way.

We would also like to thank everyone at Ericsson Research who helped out along the way, including everyone who took part in user tests, helped out with technical issues, and helped make us feel at home. We would also like to thank the people around us, including friends and family, who gave good advice and were there for us when we needed it.

Contents

1	Introduction	9
1.1	Research Questions	10
1.2	Methodology	10
1.3	Research Company	11
1.4	Contributions	11
1.5	Thesis Outline	11
1.6	Usage of AI	12
2	Background and Related Work	13
2.1	Extended Reality (XR)	13
2.1.1	Context-Awareness	14
2.1.2	3D Object Rendering and Placement	14
2.2	Generative Artificial Intelligence	15
2.2.1	Image and 3D Object Generation Models	15
2.2.2	Large Language Models	16
2.2.3	Visual Language Models	16
2.2.4	Prompt Engineering	17
2.3	Explainable AI (XAI)	17
2.3.1	Generation of Explanations	17
2.3.2	Communication of Explanations	18
2.3.3	Evaluation of Explanations	18
3	Method	21
3.1	Problem Conceptualization	22
3.1.1	Literature Review	22
3.1.2	State-of-the-Art Exploration	23
3.2	Solution Design and Implementation	24
3.2.1	Solution Design	24
3.2.2	Solution Implementation	24
3.3	Solution Evaluation	25

3.3.1	Quantitative Evaluations	25
3.3.2	User Study Evaluations	26
3.4	Context of the Research Company	28
3.5	Threats to Validity	28
3.5.1	External Validity	29
3.5.2	Internal Validity	29
3.5.3	Construct Validity.	30
4	Problem Conceptualization	31
4.1	3D Object Generation	31
4.1.1	3D Object Generation Methods	31
4.1.2	3D Object Generation Models	32
4.2	3D Object Placement	34
4.2.1	3D Object Placement Methods	34
4.2.2	Spatial Understanding with VLMs	35
4.2.3	3D Object Detection Models	35
4.3	Explainability	35
4.3.1	Characteristics of Good Explanations	36
4.3.2	Self-explanation Mechanism	36
4.4	Solution Requirements	37
5	Solution Design and Implementation	39
5.1	Generation Module	39
5.1.1	Instruction Generation	40
5.1.2	Text-to-3D Object Generation	41
5.2	Placement Module	41
5.2.1	Context Specification	42
5.2.2	Placement Specification	42
5.3	Explanation Module	43
5.3.1	Feature Attribution Loop	43
5.3.2	Explanation Prompting	45
6	Solution Evaluation	47
6.1	Quantitative Evaluation	47
6.1.1	Sensitivity Analysis for 3D Object Generation	47
6.1.2	Sensitivity Analysis for 3D Object Placement	49
6.2	Qualitative Evaluation	49
6.2.1	User Study Results	50
7	Discussion	55
7.1	Limitations of Existing Solutions (RQ1)	55
7.1.1	Generation Quality	55
7.1.2	Placement Constraints	56
7.1.3	Limited Context-Awareness	56
7.1.4	Absence of Explainability	56
7.2	Design and Implementation Decisions (RQ2)	57
7.2.1	Addressing Generation Quality	57

7.2.2	Addressing Placement Constraints	57
7.2.3	Addressing Context-Awareness	57
7.2.4	Addressing Explainability	58
7.3	Performance Evaluation (RQ3)	58
7.3.1	Context-Awareness in Generation	58
7.3.2	Context-Awareness in Placement	59
7.3.3	User Perception of Explainability	60
7.4	Limitations and Future Work	60
7.4.1	3D Object Generation	60
7.4.2	3D Object Placement	61
7.4.3	Explainability	62
7.4.4	Performance Evaluation	62
7.4.5	System Ethics and Practical Concerns	63
8	Conclusions	65
	References	67
	Appendix A Prompts	75
	Appendix B User Study Questions	83

Chapter 1

Introduction

Extended Reality (XR) allows users to view, experience, and modify the world around them in a way that feels more real than a traditional screen. Even as XR-environments grow larger and expand into new areas, it is important to maintain the immersive nature of this technology to keep users interested and excited [10]. This immersion is largely dependent on how closely the simulated world corresponds to the real world, including its objects and their instantiation [42].

In an XR environment, objects can be instantiated either manually or through generative models [16]. While manual creation and transformation of objects allows for creative expression, it becomes a daunting task once the worlds become large or otherwise unmanageable. An alternative approach is to use modern generative models to generate virtual objects at a scale which is faster than manual work [28]. The generated objects can then be integrated into automated pipelines with mechanisms such as AI-generated placement [41].

While some AI models can instantiate objects in an environment, the sense of intuition for *where* and *how* an object should be set in a scene remains a challenge [7]. Difficulties in object placement may be both practical and aesthetic, especially when faced with complex environments. Factors such as scaling, translation, rotation, and appearance are all dependent on the information about the surrounding environment, as well as the user. For instantiation to be fully realistic, the surrounding context, user, and their constraints need to be considered by the placement mechanism, such that objects have a natural appearance to the user. This also means that, ideally, the system should consider information that the user gives during runtime, and even be able to instantiate on the request from a user [49].

Users typically seek explanations when the behavior of an AI system deviates from their expectations. In such cases, human-interpretable explanations are essential to help reconcile discrepancies between what the user expects the system to do and what it does [18]. This is particularly pronounced in generative AI, where models often produce complex artifacts through opaque processes [40]. Existing approaches to explaining generative models diverge along different directions. Mechanistic interpretability aims to uncover

a model’s internal representations, offering deep insights for technical users. However, such approaches remain largely inaccessible due to restricted access to model internals (e.g., API-only access). In contrast, self-explaining mechanisms generate explanations as part of the model’s output, for example, through chain-of-thought (CoT) prompting or other forms of rationale generation [40]. These approaches are more feasible in applied settings, such as 3D object instantiation in XR.

In this thesis, we investigate the integration of Generative AI into XR environments, with a focus on context-awareness and explainability. We analyze the limitations of existing methods, design and implement a solution to address identified requirements, and evaluate it using both quantitative and qualitative methods.

1.1 Research Questions

While the use of AI models has accelerated the generation and placement of 3D objects, also called 3D object instantiation, existing frameworks still fail to reconcile environmental and user constraints. Furthermore, the “black-box” nature of these generative systems leaves users unable to discern the underlying logic behind decisions made by these systems. To address the lack of context-awareness and transparency in these systems, this thesis aims to answer the following research questions:

- RQ1:** What are the limitations in current 3D object instantiation solutions for XR environments with respect to context awareness and explainability?
- RQ2:** How can a context-aware and explainable 3D object instantiation solution for XR environments be designed?
- RQ3:** How well does the designed solution perform context-aware and explainable 3D object instantiation with regard to sensitivity analysis and human evaluation?

1.2 Methodology

The research is conducted in alignment with the research questions, following the Design Science Research (DSR) paradigm [37]. It comprises three phases: Problem Conceptualization, Solution Design and Implementation, and Solution Evaluation. The first phase, Problem Conceptualization, aims to understand the scope of current 3D object instantiation solutions and find their limitations in regard to context-awareness and explainability. This is done through literature review, as well as an investigation into state-of-the-art technologies and their performance. The result from the first phase is then used to create requirements for the second phase. In the Solution Design and Implementation phase, the requirements, from the previous phase form an initial plan for tackling the problem. The initial plan creates a pipeline containing three modules: 3D object generation, 3D object placement and explainability. The solution is then implemented iteratively, with testing used to refine the plan. Finally, the Solution Evaluation phase involves rigorous testing to evaluate the created multi-modal pipeline on context-awareness and the explanations.

This is done through a sensitivity analysis and a user study, with results compared against findings from Problem Conceptualization to draw conclusions.

1.3 Research Company

The work forming this Master Thesis is conducted at Ericsson Research. In order to stay at the forefront of technological advancements, Ericsson Research conducts research in devices and technologies which are projected to be a major part of the future landscape of communication. This involves both XR and AI research, which are combined in this thesis. Furthermore, the need for explainable AI is also important to keep in mind as the systems grow larger and more advanced. In other words, building trust for future decision-making systems is crucial for creating technology which serves the broader good.

1.4 Contributions

Both authors took part in the major steps in this work, with specific tasks divided to improve efficiency. In the Problem Conceptualization, the first author mainly researched literature regarding existing instantiation solutions, while the second author mainly researched articles on generative models. Explainability was researched by both authors. Both authors also tested LLM capabilities in state-of-the-art exploration, as well as different object detection models. While implementing, the first author mainly worked on the placement and integration with XR, while the second author mainly worked on 3D object generation and explainability implementation. The first author also worked with the software environment setup. Both authors took part in internal testing during implementation. Quantitative evaluation was planned and performed by the second author, while the user study was planned and analyzed by the first author. Both authors took part in conducting the qualitative evaluation. In writing this report, both authors contributed equally, with the first author focusing on object placement and user study evaluation, the second author focusing on object generation and quantitative evaluation and both authors working on explainability.

1.5 Thesis Outline

The remainder of the thesis is organized as follows. Chapter 2 briefly introduces important concepts that the research builds upon. Following that, we describe our method in Chapter 3. We present the work in the three phases that it was conducted in, beginning with findings from each phase. This includes Problem Conceptualization in Chapter 4, Solution Design and Implementation in Chapter 5, and Solution Evaluation in Chapter 6. In Chapter 7, we discuss the results, limitations, and future work. Chapter 8 summarizes our work and our findings.

1.6 Usage of AI

AI tools have been used as coding assistance to develop the software used in this thesis. AI tools have also been used to help generate ideas, for spell and grammar checks, and assist in finding research articles. These AI tools include ChatGPT, Gemini and Claude.

Chapter 2

Background and Related Work

For the reader to understand the contents of this thesis, there are a few concepts that the reader needs to have knowledge about. The aim of this section is to provide the reader with this knowledge. Hence, the section will cover extended reality (XR), generative artificial intelligence (GenAI) and explainable AI.

2.1 Extended Reality (XR)

Extended reality (XR), commonly used as an umbrella term for virtual reality (VR), augmented reality (AR), and mixed reality (MR), describes how whole or partial realities can be created or altered through technology. However, there is no single, universal definition of XR, its subforms nor what it includes [26, 32]. VR creates an immersive, artificially constructed environment through technology like headsets. Meanwhile, AR blends digital information with the physical world. While AR is commonly accessed via smartphone apps, recent hardware advancements, such as depth sensors and eye tracking, have unlocked novel human-computer interfaces. MR combines virtual and real-world elements, typically utilizing advanced headsets and controllers [32].

Because the similarity in the definitions between AR and MR, there are several papers trying to define their relation. Some claim that AR is a subform of MR, while others do not agree. They argue that VR, AR, MR and the real world are all placed on a spectrum, where MR is placed between AR and the real world [32]. This report will focus mainly on AR and MR and because of the similarity, as well as mixed definitions, moving forward this report will refer to AR and MR as XR.

The following subsections explore the areas deemed most relevant for the user to understand XR in the context of our thesis. First, context-awareness is examined to establish how it can be defined within the scope of our thesis. Second, 3D-object rendering and placement in XR is reviewed to provide the reader with a foundational understanding of the mechanism of object generation and placement in standard XR systems.

2.1.1 Context-Awareness

As context-awareness is a core concept in our thesis, it is crucial to establish the definition of context-awareness within the scope the project. Dey [12] defines a system as context-aware as "if it uses context to provide relevant information and/or services to the user, where the relevancy depends on the user's task" and context is defined as "any type of information that can be used to characterize the situation of an entity" [12]. Within the scope of 3D object instantiation in XR, context-awareness encompasses that the system shall be aware of its environment and place and generate an object in such a way that is natural and safe. Moreover, context-awareness includes adapting the object and placement based on the user attributes. This could be putting a shelf lower on the wall for a shorter user to reach, or generating a toy gun for a kid user and a real gun for a police officer in an XR environment. To conclude, context-awareness in this thesis means adapting 3D objects to both the physical environment and the needs of the specific user.

2.1.2 3D Object Rendering and Placement

As the field of XR has developed, so has the need to create and transform 3D objects, also known as 3D assets, to fit into XR environments. These assets are often intractable and includes humans, props, and scenes. A 3D asset is often complex and consists of vertices, edges, faces, meshes, and a topology [1]. They can be generated through manual 3D modelling, such as Blender and Unity [4, 1]. Manual generation of 3D assets is often time consuming, and therefore, novel solutions have been introduced, such as apps and websites that generate these assets per user request [36]. Examples of these are Tripo AI [45] and Meshy AI [25]. The apps often use 3D object generation models, which will be further explored in Section 2.2.1.

Besides generating an object in XR, the object must also be placed in a natural and safe way within the environment. To be able to do this, there must be a way to understand the environment and where objects can be placed in it. In the physical world, there are physical constraints on the environment and the object, but this is not the case when placing an object in XR. The object and/or the environment are virtual, which means there are no physical constraints. If there are no constraints enforced in an XR environment, a virtual object could be placed anywhere, including non-reality conforming places [38].

A challenge with automatic placement is to determine precise 3D coordinates. In VR, the virtual environment has one center, also called the origo. On the other hand, in AR and MR, the coordinate system changes as the user moves. To get the correct coordinates in XR, there are different proposed solutions. Some use 2D snapshots of the environment, obtain the 2D coordinates, and ray-cast the object into the correct spot in the XR environment [52, 41]. The procedure of ray-casting entails the identification of 2D coordinates and then uses a ray that goes through the camera and the 2D coordinates to find where the ray intersects with a plane in the image. It is at that intersection that the object will be placed [19].

2.2 Generative Artificial Intelligence

Generative AI (GenAI) refers to the creation of novel content in forms of text, images, audio, and code through the use of AI [13]. GenAI is used in many different domains, such as healthcare, education, and business workflows. Some generative models are designed to only be able to do one specific task, while others are designed to be able to generate content for multiple different types of media and formats. Models limited to a single modality for inputs and outputs are known as unimodal, while models that can handle different types of input and produce output in various formats are called multimodal models. Multimodal models assist users with a range of unique tasks, including processing spoken language, answering text-based queries, and identifying objects in images [13].

The following subsections cover the specific generative model types relevant to this thesis, as well as prompt engineering techniques. The models presented below, are instrumental in the area of 3D object instantiation, which is a primary focus of this thesis, while prompt engineering serves as a valuable method to enhance the performance of these models, as detailed in Section 2.2.3

2.2.1 Image and 3D Object Generation Models

The field of image generation has rapidly advanced, with image generation models being able to create and edit images through text and image prompting at a fast speed and high quality. There are plenty of different models to utilize for 2D image generation, including generative adversarial networks (GANs), variational encoders (VAEs), autoregressive models, and diffusion models. GANs pit a generator against a discriminator in adversarial training to produce realistic images, while VAEs encode inputs into a latent distribution and sample from it to generate new instances. Autoregressive models generate images sequentially, token-by-token, whereas diffusion models iteratively denoise a signal from random noise and currently dominate the field due to superior output quality and training stability [20]. Diffusion models underpin most state-of-the-art image generation systems [35].

Although there are several different 2D image generation models, most of them faces similar challenges regarding text-image alignment. When prompted to create an image, they often focus more on certain parts of the prompt, while disregarding other parts of it, leading to the unintended outputs. As the prompts become more complex, the quality of the images decline. This can be shown either through a decrease in the texture quality of the image or the image not fulfilling the prompt [14].

Unlike 2D generation, 3D object generation faces two fundamental challenges. To start, there is a scarce amount of 3D assets available to train the 3D generation models. Moreover, many of the models used to generate 2D images are not adapted to handle higher dimensions [20]. This has created a need for an innovative solution to handle transforming object generation from 2D to 3D [20, 44].

The 3D generation models can be of different kinds and produce content in various formats, for example, meshes, point clouds, and voxel grids. Current 3D generation models typically utilize generative adversarial networks, diffusion processes, or rectified flow frameworks. Rectified flows have been proven effective for large-scale 3D generation [48].

Many 3D object generation models, both open source and proprietary, which refer to models owned by companies, can create 3D objects from different media, including images, text, and speech. Examples of these models are Shape-E and TRELLIS, where Shape-E uses a diffusion model and TRELLIS rectified flows [16, 48].

2.2.2 Large Language Models

The most common type of model for creating generative content is Large Language Models (LLMs), which belong to a type of neural networks that use a transformer architecture [54]. The transformer architecture uses an attention mechanism to process long sequences of words effectively, hence avoiding bottlenecks common in recurrent neural networks [46]. The training of these models can vary both in the data used and the learning process. The models can either be open source or proprietary [54]. Many proprietary models currently available are "black box", which means that the users do not have any insight into the inner workings of the models and their training. This causes issues when the model outputs responses containing misinformation, in the shape of bias or factually incorrect information [40]. Mitigation techniques for avoiding this include implementing LLMs that provide the sources to their statements, human-in-the-loop and prompting the LLM in new ways, for example, through one or few-shot prompting. By including one or more demonstrations of a task within the prompt, one-shot and few-shot prompting guide the model's behavior and ensure more consistent results [54].

2.2.3 Visual Language Models

The further the development of LLMs has made it possible for models to understand and interact with physical spaces in new ways. These models are called Visual Language Models (VLMs), and they combine a vision encoder with a language model to bridge the gap between visual and textual data. There are several models that can reason, to different extents and accuracy, about objects and the relations between them in images and videos. This enables the models to perform tasks such as scene descriptions and retrieval of information from different sources of media [23].

Although development has been rapid, there are still limitations; for instance, the models remain sensitive to prompt phrasing and typically lack precise pixel-level localization [31]. By integrating object detection models, which can identify objects and their location, through bounding boxes or heatmaps, the VLMs' performance can be enhanced [22]. Lüddecke et al. [22] present a model, called ClipSEG, that, given a text prompt and an image, creates a heatmap over the areas in the image where said text prompt holds true. Furthermore, Lui et al. [21] present a model for identifying 2D boundary boxes and Brazil et al. [6] propose one to recognize 3D boundary boxes of objects in an image. The use of an object detection model paired with a VLM has been shown to enhance spatial awareness [41].

To evaluate how well a model understands spatial relations, Visual Question Answering (VQA) can be used. VQAs are designed to give a VLM an image and a question about it, from which the VLM should be able to generate correct answers to the proposed question. This is a way to train models for vision tasks and can also be used to evaluate a model's performance [53].

2.2.4 Prompt Engineering

The thesis will utilize pre-trained, off-the-shelf models whose behavior must be adapted to meet our specific use case. Prompt engineering is an emerging field in the landscape of LLMs and VLMs, where a model receives task-specific instructions, a prompt, with the purpose of enhancing the model's performance of the task. The use of prompt engineering has made it possible for LLMs to excel across various tasks and domains [39].

There are several different ways to use prompting. They can be split into zero-shot prompting, where the model is supposed to handle the prompt without any previous examples on how to answer it, and few-shot prompting, which provides the model with a few examples of the input prompt and how the model should output the answer. Other emerging ways of prompt engineering are chain-of-thought and tree-of-thought prompting, where the model is forced to output the steps taken to reach the answer. This reduces the risk of hallucinations. Lastly, the output can be altered by telling the model to act as a certain role, for example, a senior developer or a teacher. This adapts the model vocabulary and makes it more oriented towards a specific task [39].

2.3 Explainable AI (XAI)

The purpose of explainable AI (XAI) is to make AI systems understandable to humans, thereby supporting appropriate trust and effective use. This is achieved by techniques and principles that make a model's behavior, decisions, or internal mechanisms interpretable. XAI has been widely applied across machine learning tasks such as classification and regression, as well as in application domains including recommender systems, policy learning, and financial analysis. The explanation methods can be categorized into model-specific and model-agnostic methods. Model-specific methods are designed for particular types of machine learning models. Model-agnostic methods can generally be applied to different machine learning models. Moreover, the explanations can be split into global and local explanations, depending on whether a specific decision is being explained or the entire system [24]. In the following, we describe how explanations are generated, communicated, and evaluated. In Section 4.3 we will explore how this is applicable for our system.

2.3.1 Generation of Explanations

Explanation generation techniques developed specifically for GenAI models or classical explainability techniques adjusted for GenAI can be categorized into four categories: feature attribution, sample-based, mechanistic, and probing-based approaches [40]. Feature attribution methods assign relevance or importance scores to individual input features to show their impact on the final output, whereas sample-based methods focus more on the entire sample set to understand the relationship between various inputs and their corresponding outputs. Mechanistic interpretability investigates neurons and their interconnections, aiming to reverse-engineer model components into human-interpretable chunks. Probing-based methods aim at understanding what knowledge an LLM has captured through queries (probes) [40].

Because generative models are largely inaccessible due to restricted access to model internals (e.g., API-only access), explainability methods for these models mostly rely on observations of how variations in the input affect the model output. Another way to generate explanations for these models is through their self-explaining mechanisms as part of the model’s output, for example, through chain-of-thought (CoT) prompting or other forms of rationale generation [40].

2.3.2 Communication of Explanations

Explanations should be communicated in human-understandable terms to be effective and useful [2]. Explanations can be presented in multiple ways through text, visualizations, explanations by example or simplification and feature relevance [2]. Presentation through text uses natural language explanations to explain the behavior of the model. In designing text explanations, cognitive load theory can be considered to minimize the strain on a user [29]. This can include minimizing the amount of unnecessary information, among other things. Visualizations methods visualize the models’ behavior through saliency maps, heatmaps, and/or plots. Explanations by example identify data examples that illustrate the underlying relationships and patterns captured by the model, while explanations by simplification construct a simplified system that mimics the original model’s performance in a more understandable form. Finally, feature relevance methods clarify model behavior by assigning sensitivity scores to variables and thereby measuring each feature’s impact on the final prediction [2].

2.3.3 Evaluation of Explanations

Evaluating explanations can be challenging due to a number of factors. The diversity and dynamic context of the models make it difficult to create a single objective standard. Moreover, evaluating a good explanation can be subjective and vary from person to person. Therefore, the evaluation of explanations can be split into two: objective and subjective assessments [43].

Objective assessments focus on functional and safety measurements, where functional evaluation methods are measured through fidelity, consistency, and stability and safety through accuracy, fairness, and robustness. Fidelity reflects how well the input and output mapping agrees with the inner workings of the model, while consistency addresses how outputs change between different models assigned to similar tasks, and stability refers to how well instances of the same model are able to handle the alterations in input data. Fairness certifies that the model’s explanation is just and does not contain bias. These factors can be assessed through methods such as model understanding, similarity metrics, and sensitivity analysis. Model understanding compares the content of the explanation to the inner workings of the model, similar metrics use certain fitting metrics to evaluate, and sensitivity analysis compares the difference in output when there are slight tweaks to the input data [43].

Subjective assessments focus on human-grounded evaluations, involving either laypersons or domain experts and are assessed through user-satisfaction metrics and trust evaluations. The subjective assessments are often evaluated through comprehensibility, justifiability, and interactivity. Comprehensibility relates to explanation understandability

for the user, justifiability reflects how well a domain expert deems the content of a model accurately reflects the field, and interactivity refers to the model's ability to use previous context to be able to reason and answer questions about the explanation. Evaluation methods for subjective assessments include user surveys and questionnaires, interviews, and post-interaction feedback [43].

Chapter 3

Method

In this study, we followed the Design Science Research (DSR) paradigm to design, implement, and evaluate a 3D instantiation system in XR [37]. As DSR suggests, the research was split into three phases: Problem Conceptualization, Solution Design and Implementation, and Solution Evaluation (see Figure 3.1).

Problem Conceptualization was conducted to form an understanding of the current research landscape within the area. We began by performing a literature review to gain information about previous approaches, including proposed solutions to the problem and the associated limitations. To make informed decisions in designing the system, we also explored state-of-the-art tools. These tools were evaluated on their performance in order to test their suitability to our needs. Both literature review and state-of-the-art exploration were divided into three focus areas, which corresponded to 3D object generation, 3D object placement and explainability.

In the Solution Design and Implementation phase, we designed a solution for 3D object instantiation based on our findings in the previous phase. Methods, techniques, and tools from the Problem Conceptualization that were deemed relevant and effective were included as components in our solution design. In the design phase, we followed the three focus areas identified in the Problem Conceptualization. Once the design had been finalized, the implementation of the system was done in accordance with it.

The final phase, Solution Evaluation, involved evaluating the system in the aspects defined in the third research question. Evaluation of our solution, therefore, combined quantitative and qualitative methods: sensitivity analysis and user study. Sensitivity analysis was used to evaluate the fidelity of the system's explanations, as well as the context-awareness of the object generation and placement. The user study was done to evaluate the provided instantiation and explanation with regard to perceived context-awareness and explainability.

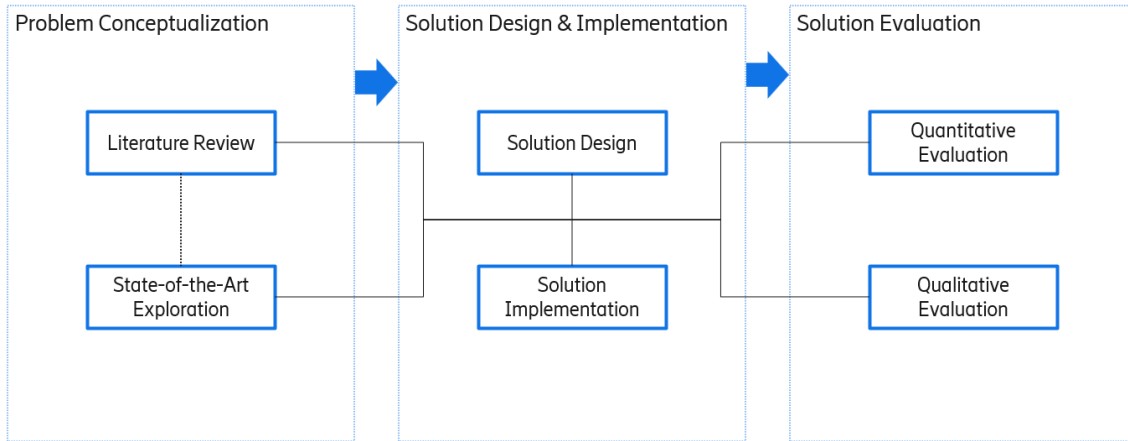


Figure 3.1: Overview of the DSR phases, highlighting the specific steps for our thesis: literature and state-of-the-art reviews, system design and implementation, and final mixed evaluation methods.

3.1 Problem Conceptualization

This phase contained two activities: Literature Review and State-of-the-art Exploration. Literature Review was done in order to create an understanding of the current research landscape and further refine the research scope. Several relevant approaches and tools that were found in the literature review were then tested, subject to their availability as either an open-source or proprietary solution. The goal of this phase was to identify the requirements for our own solution based on the identified gap in the research.

3.1.1 Literature Review

As a first step in conducting our research, we searched for articles to deepen our knowledge in the field. The goal was to explore the current state of the research surrounding context-aware and explainable instantiation systems, identifying what problems, proposed solutions, and evaluation methods were described in the literature. Since the work was done in collaboration with both the university and a company, many different approaches were taken to find relevant literature.

For most research article searches, the university database search engine Finn was used. This tool searches a variety of databases that the university has access to, including IEEE Xplore, ACM Digital Library and Springer. The reason for this choice was that this let us access media that would otherwise be blocked behind a paywall.

Together with Finn, other methods were also used to find relevant research articles. Notably, Google Scholar was used for its ability to find less accredited articles and suggest related ones through the referencing system. The search feature in Google’s Gemini also allowed us to tailor the search terms more narrowly, as well as determine more certainly when we found a research gap. The specific search prompts and keywords employed across these platforms included combinations of: "context-aware artificial intelligence systems," "context-aware extended reality systems," "3D-object instantiation," "3D object placement," "explainable AI," and "explainable XAI." Additional articles recommended

by our supervisors directed the search toward relevant work that may otherwise have been overlooked. We found a total of 97 articles before a selection were made.

Rather than applying a rigid checklist of inclusion and exclusion criteria when deciding which articles were relevant, the selection was based on a qualitative relevance assessment. Each article was screened for its direct alignment with the core objectives of the thesis, specifically focusing on context-aware object instantiation and explainable AI within XR environments. Papers were excluded if they diverged into unrelated machine learning topics or lacked actionable technical insights for 3D instantiation. This screening ensured that the final selection of literature remained tightly focused on the technical requirements of the project.

We categorized the relevant literature into one of several research fields. This included 3D generation, 3D placement, and explainable AI. Sections 4.1.1, 4.2.1 and 4.3 provide the key findings from the literature review.

3.1.2 State-of-the-Art Exploration

While reading research articles was a good way to gain an understanding of a proposed solution, it was hard to see how it could be tailored and applied to our own research problem. Many available solutions were described and evaluated in a way that followed the researchers own goals, which may not align with our own. When the solutions were available openly, we tested them out on our own terms.

A main goal of the exploration was to find an AI model that could reliably reason about a 3D environment based on the data available from an XR headset, as well as a text prompt. Furthermore, the AI model should be able to produce structured outputs in such a way that it can be used in a programmatic manner in a larger system. Since this stage of the system contained the intelligence and decision-making, it was deemed useful to test it detached from later stages, without XR integration. The tools that were explored were a mix of open source and proprietary solutions. While many research articles reveal the methods and data that were used in creating the solution, not all publicly release the solution as a complete, ready-to-use implementation. Due to the limited scope of a Master Thesis, it was not feasible to re-implement or re-build every tool that seemed relevant, so only ones that were openly released were tested. Along with the open source solutions, there were proprietary methods that could serve as alternatives, which were tested using free trials along with university licenses when applicable.

A further goal of the exploration was to identify a 3D generation model which could balance the tradeoff between quality and time. To decide which model was the best match for the task, the models were evaluated qualitatively on metrics such as image quality and, for text-to-3D models, the sensitivity to prompts. Time to produce the 3D object was also considered. The final approach included using a 3D object detection model combined with an LLM to provide the 3D boundary boxes to guide object placement. The model was tested using images from the CA-VQA dataset to test the model.

The findings from this exploration were used to identify the requirements and inform the selection of components and tools to be used in the solution design and implementation phase. Sections 4.1.2, 4.2.2 and 4.2.3 provides further information of the models and the outcome of our state-of-the-art exploration, while Section 4.4 presents the requirements identified.

3.2 Solution Design and Implementation

The second phase of the project consisted of two activities: Solution Design and Solution Implementation. First, a design for the proposed solution was done in line with the requirements that were found in the previous phase. Following this design, the implementation of the system was done to realize the solution and obtain a testable product. Both design and implementation followed an iterative approach, where an idea was designed, implemented, and tested, then revised as needed. For the result of this phase, please see Section 5.

3.2.1 Solution Design

The design of our solution followed a modular approach, divided into three modules, each with its own sub-modules. These modules corresponded to 3D object generation, 3D object placement and explanation.

In designing our solution, we had to consider constraints on the system from the outside. This included mainly the availability of technologies. For software, this meant some alternatives had to be excluded because of their closed-source nature or general unavailability. For hardware, this meant that the solution of the system had to be feasible for the available resources given by the hosting company. These factors all influenced the final design of the solution in one way or another. Another constraint was the type of use case for our solution. In order to limit the scope, the use case was set to indoors, including mainly home and office settings.

Four 3D object generation models were tested on both time and quality of the final object, as well as sensitivity to the details in the prompt. The sensitivity and quality were evaluated qualitatively from how closely they resembled an actual object that matched the prompt. Internal testing for the placement of objects was done in a virtual environment in Unity, to ensure that the proposed coordinates corresponded to the intended placement of the system. Explanations were internally tested for how well they conveyed the context that contributed to the decision, as well as how easily understandable the language was.

3.2.2 Solution Implementation

To execute the generation, placement, and explanation tasks defined in our design, the final solution was implemented as two separate programs in a server-client architecture. This essentially enabled a decoupling between the intelligence of the system and the user interface. Communication was done over HTTP requests to an API on the server, which allowed transferring both text and larger files between the client and the server. A multi-processing approach was taken on the server-side to perform heavy computations while still being responsive on the connection to the user.

The server program was implemented with both software and hardware restrictions in mind. The server program was written in Python. This decision was partially forced by other solutions that we incorporated into our system, which used machine learning frameworks needing this setup. To run the program, we used a Linux server equipped with an RTX 3090 GPU for hardware acceleration.

The user program was implemented in Unity for the Meta Quest 3. Meta development kits were used for general XR functionality.

We used OpenAI’s API with a subscription for generating both text and images. This provided a unified interface for several of our demands for the system. For generating a 3D object from an image, we used Microsoft’s TRELLIS pipeline, hosted locally.

3.3 Solution Evaluation

The final stage of the project involved evaluating our solution in both a quantitative and qualitative way, as discussed in the following sections.

3.3.1 Quantitative Evaluations

Perturbation-based methods, which examine how changes in inputs affect model outputs, are widely used in explainable AI. In our research, we focused on the idea of sensitivity analysis in LLMs as an evaluation method, since it allowed us to analyze large, proprietary models which are restricted in internal model accessibility. For context-awareness, sensitivity analysis was a suitable approach since it provided a way to isolate context sources and view their impact on the system.

The goal of the quantitative evaluation was to evaluate the sensitivity of the system, to determine the degree of context-awareness for all input features. Comparisons between scores were compiled into averages, in order to account for internal variance in the system. Section 3.3.1 provides details of how the evaluation was conducted.

Quantitative Evaluation Setup

For the sensitivity analysis, both the user features and the context of the environment were perturbed. The user features were perturbed by replacing the values of the features feature at a time and observing how the output was affected. The perturbation values were chosen to differ in a major way from the baseline, without using unrealistic values. For the environment, the input image was changed to another room, with alternative colors and style.

We applied these perturbations across four distinct object examples to evaluate whether the system’s sensitivity varied depending on the type of object requested. The objects were: a trash can, a knife, a bike and a wall mounted shelf. We determined the baseline values for each object for both the user data and the environment. Each feature was perturbed in the same way for both generations and placement, and was iterated through five times for each user prompt. Each perturbation result was compared to each baseline result. From this, an average similarity score for each feature was calculated. Considering LLMs are non-deterministic, this accounted for the internal variance, ensuring that any observed changes in the output were due to our perturbations rather than random model fluctuations. In Table 3.1, the values of the baseline and the perturbations of the features are shown. The bold text highlights the feature that is perturbed for each perturbation and its value.

There were two environments utilised during the testing: a baseline environment, Figure 3.2a, and a is perturbed environment, Figure 3.2b. The environment column in Table

3.1 shows which figure is used for each perturbation.

ID	Iterations – Features	Height (cm)	Age	Style	Environment
1	Baseline	160	50	Antique	Figure 3.2a
2	Perturbation 1	200	50	Antique	Figure 3.2a
3	Perturbation 2	160	10	Antique	Figure 3.2a
4	Perturbation 3	160	50	Modern	Figure 3.2a
5	Perturbation 4	160	50	Antique	Figure 3.2b

Table 3.1: The baseline and perturbed values in each iteration.



(a) The original environment.



(b) The alternative environment.

Figure 3.2: Perturbation in environment.

The perturbed output was compared to the reference output, and a similarity score was calculated. The inverse of this similarity score was used as a sensitivity score. The calculations of the score differed based on the module. For the generation module, the DINOv2 embeddings were used. By transforming the images to embedding vectors, DINOv2 compared the style of images and from this generated a similarity score [30]. Placement similarity was measured by calculating the Euclidean distance between the original and perturbed coordinates, integrated with the absolute difference in scaling of the object, see Equation 5.2. The inverse of the similarity score was used to compare the impact of individual features across each prompt and visualized in the grouped bar chart. The details of the analysis can be seen in Sections 6.1.1 and 6.1.2.

3.3.2 User Study Evaluations

In combination with sensitivity analysis, we also found it important to evaluate explainability with tests involving human users, since the end goal of the explanation is to be user-friendly and help the user understand the system. Context-awareness, as defined in our project, is also a characteristic which is best evaluated through a user test, since it is partially a qualitative aspect. Users provided feedback on how they perceived the system from a context-awareness and explainability perspective, both in regards to the user attributes and the surrounding scene.



Figure 3.3: User study setup.

Our user study was performed through a questionnaire-based approach, containing both scale-based and open-ended questions. This approach was chosen because we specifically wanted to determine the context-awareness and explainability of our solution, and could tailor the questions to this goal. By introducing open-ended questions, we could also collect more detailed information without having to conduct an interview. After the questionnaire, users got to choose a preferred explanation for each scenario. This was done to compare user preference to fidelity scores. Section 3.3.2 provides further details of the setup of the user study.

The questionnaire was constructed based on the advice and experience of our supervisors at LTH and Ericsson, whom also reviewed it before the tests started. The questions from the user study are included in Appendix B.

User Study Setup

The user study was conducted in a controlled environment using the headset Meta Quest 3, as shown in Figure 3.3. The environment in question was an XR lab, containing different types of objects and furniture. Before starting each session, a consent form was signed by the participant. After this, they filled in age, height, and personal style into the system. They also provided information about prior experience with XR and 3D graphics.

Initially, each user got to instantiate one object in order to get used to the system. Once this was done, they were instructed to instantiate one object from a set view and prompt. A second object was then instantiated freely by the user, with their own choice of view and prompt. They were, however, instructed to keep the prompt short and have a good, wide view of the scene in front of them. After instantiating the objects and reading the associated explanations, the participants were asked to answer the questionnaire. Finally, they were presented with a set of explanations and were asked to choose their preferred one to evaluate the validity of our explanation improvement.

The study was conducted with a volunteer sample of seven participants ($N = 7$) recruited from within the research organization. Their ages ranged from 24-48, their heights from 170-193, and the decorative styles included minimalist, nature-inspired, casual, and modern. Their answers were compiled into analyzable data by summarizing the quantitative findings into graphs and performing a thematic analysis on the qualitative responses. The results can be viewed in Section 6.2.1.

3.4 Context of the Research Company

In Design Science Research (DSR), the problem context plays a foundational role because it defines why the research matters and what kind of solution is needed. It situates the problem within a real-world environment and guides the entire research process. This was done during the all parts of our thesis. For our thesis, the problem context was explicitly shaped by the operational needs, technical constraints, and resources of the hosting research company, which can be categorized into dimensions of expert-driven guidance, technical limitations, and financial resources.

The company supervisors provided expert-driven guidance, with one offering expertise in the field of AI and the other providing technical direction within the XR context. This input functioned as a design guide that scoped the technical trajectory of the research and directly influenced specific architectural choices. For example, their guidance led to utilizing 3D coordinates from an image rather than relying on standard 2D coordinates or the native plane detection of the XR device, thereby effectively narrowing the design space to suit the target application.

Moreover, the physical and digital infrastructure of the research company imposed specific technical constraints, encompassing designated equipment, hardware instruments, and software configurations. For instance, the company provided a select number of XR headsets that the supervisors steered the project toward utilizing. The software selection was guided by organizational licensing and internal familiarity. Therefore Unity was selected because the company already possessed the required licenses and the internal team was proficient in its environment. Finally, hardware limitations, such as GPU capacity and local memory, constrained the types of models that could be implemented, bounded the scale of testing during the State-of-the-Art exploration, and ultimately restricted the use case to indoor environments.

Conversely, the financial budget allocated by the company acted as a facilitator rather than a constraint. Instead of limiting the project boundaries, this funding expanded the scope of the SOTA exploration by enabling the use and evaluation of both open-source and commercial models. Consequently, this resource broadened the testing and evaluation space.

3.5 Threats to Validity

To ensure the reliability and validity of our research, it is essential to understand the factors that could potentially affect our results. Threats to validity can be sorted into three categories: external validity, internal validity and construct validity. External validity

concerns the generalizability of our research, while internal validity examines if the design of the system introduces errors or bias, and construct validity refers to the accuracy of our chosen measurements. The subsections below present potential threats within these categories.

3.5.1 External Validity

Although our research was extensive for the activities in our problem conceptualization phase, it is possible that some solutions and models were not discovered. Finding other potential solutions may have affected the perceived limitations in the existing literature, and thereby guided the work in a different direction. However, the scope of the search was designed to capture the most prominent solutions, providing a representative baseline of the limitations in current 3D object instantiation solutions.

Moreover, our user variation was strictly limited to the participants found at the research company. This limited user variation creates a sample pool sharing similar technical backgrounds and daily environments, which restricts our ability to generalize the results of the user study. However, while the sample population was constrained to the company, the user features still varied, with a wide range of ages (24–48), physical heights (170–193 cm), and varied interior design preferences.

Finally, the risk of researcher bias is present due to a number of factors. Because of the close proximity between the authors, the research company environment, and the participants, there is a risk they may influence each other and thereby the result. First, the researchers' decisions may have been biased toward framing the project around the pre-existing preferences of the hosting company. Furthermore, due to working in the same environment, the participants in the user study may have possessed prior knowledge of the system's development, compromising the answers from the user study. Similarly, the interpretation of qualitative feedback carries an inherent risk of subjective bias during the evaluation phase, as the researchers designed the underlying system and may unconsciously emphasize positive user remarks over critical ones.

3.5.2 Internal Validity

The design and implementation of our system were done in a modular fashion, in order to separate functionality and enable easier workflows. However, this modularity also separates knowledge, which removes some context-awareness in the system. The placement module is not aware of the generation module's result, and vice versa. In other words, a discrepancy can arise between the semantic assumptions of the placement module and the physical outputs of the generation module. In other words, a discrepancy can arise between the semantic assumptions of the placement module and the physical outputs of the generation module. For instance, if a user requests a shelf, the generation module may create a wall-mounted unit, while the placement mechanism believes that the object is floor-based and therefore positions it on the ground. This mismatch directly degrades the system's runtime validity, which may leave the user confused and unsatisfied with the result.

The architectural separation of the modules extends to the explanation module as well. For example, if the explainability module believes that the object is placed in a specific

place while it is not, it will generate an inaccurate explanation as it lacks the true placement. This may lead to mistrust in the system. Therefore, the system cannot be fully context-aware in regard to its own actions.

3.5.3 Construct Validity.

In our quantitative evaluation, we used sensitivity analysis to evaluate context-awareness in our solution. Although these terms are certainly related, there is no proven equivalence between the two concepts. Indeed, our system may be sensitive to changes in some context sources, but if the decisions are not made in a relevant and correct way, then the system is not truly context-aware in our definition. However, in combination with the qualitative analysis, we believe that our measurements come close to capturing both the perceived and actual context-awareness of the solution.

In our qualitative analysis, we used user evaluation to determine the explainability of our solution. Although user perception of an explanation is of great importance, it cannot fully account for all aspects of an explanation. In our tests, users did not have more information about the system than what the explanation gave, which prevent them from identifying potential missing components. Measuring explainability through only user study is therefore a flawed measurement, which is why it was compared to the fidelity scores from the system.

Chapter 4

Problem Conceptualization

The first phase of the methodology was the Problem Conceptualization. Through our research questions and an initial literature study, we identified key areas as our primary focus. These areas included 3D object generation, 3D object placement, and explainability. From the literature review and state-of-the-art exploration, we deepened our knowledge in these areas and identified existing solutions within 3D object instantiation in XR. The findings from the literature review and our state-of-the-art exploration is presented in this chapter.

4.1 3D Object Generation

For the 3D object generation, we explored existing 3D object generation methods in the literature to obtain a greater understanding of the components needed in 3D object generation to solve our problem. Through a state-of-the-art exploration, we identified the strengths and weakness of object generation models found in the literature. Section 4.1.1 focuses on the methods found in the literature, while Section 4.1.2 examines 3D object generation models with the goal of finding the model most suitable for our use case.

4.1.1 3D Object Generation Methods

The literature review found several studies introducing diverse methods for 3D object generation. We found that available methods varied in solution design, texture quality, and latency. Here, we present the most crucial findings and contributions from each reviewed paper.

Xui et al. [50] propose four distinct pipelines for object generation in XR. The system allows participants to choose from a selection of pre-defined object prompts to generate rather than requiring manual input. The first two pipelines are text-to-3D pipelines that utilize either Shap-E or TRELIS-text for the full pipeline. The remaining pipelines are

text-to-image-to-3D pipelines. They apply FLUX for initial image generation, followed by either TripoSR or TRELLIS for 3D reconstruction. Out of the different pipelines, the quality of the textures from the TRELLIS-image is brought up as favorable. Along with this, the authors also propose results suggesting that users may choose better quality over lower latency in an XR-instantiation system.

Sharma et al. [41] also propose a solution where the object is generated through Shap-E. Similarly, Behravan and Gracanin presents a solution for generating 3D objects from text or speech through the use of Shap-E and the spatial understanding of VLMs [3]. Another framework, proposed by Xing et al. [49] instead uses Google’s 3D object generation model Genie, which has since been made unavailable for the same type of usage. Finally, Lee et al. [17] introduce a system which uses both an OpenAI image model as well as Instant-Mesh for 3D object generation.

While Lee et al. [17], Xing et al. [49] and Sharma et al. [41] offer automatic placement alongside generation, as discussed in Section 4.2, none of the proposed 3D object generation methods have considered user attributes as context during generation, but only account for environmental information in the form of either an image or a scene graph.

4.1.2 3D Object Generation Models

Our literature review identified several commercial and open-source models that support 3D object generation. This included TRELLIS [48] and TRELLIS2 [47] by Microsoft. Importantly, TRELLIS has the capacity to generate content from either text or image data, while TRELLIS2 provides only the latter, with improved quality. TRELLIS2 also provides a range of resolution choices. Moreover, Hunyuan3D-2 [15] by Tencent was another open source model that produces 3D objects from image data. However, we excluded the model from our study because the quality of their generated 3D objects during our initial testing failed to meet the standard we deemed necessary for our thesis. We came to similar conclusion regarding Shap-E by OpenAI [16]. This was concluded by analysing the literature, where other models were shown producing higher quality objects [50].

TRELLIS offered several models with different numbers of parameters and input types, either text or 2D images. Out of these, we tested two different pre-trained models: TRELLIS-text-xlarge and TRELLIS-image-large. The first was a text-to-3D model, and the second was an image-to-3D model. For TRELLIS2, there was only one pre-trained model available, with several resolution settings. Initial testing showed that both models of TRELLIS and TRELLIS2 created high-quality objects, with TRELLIS2 providing the best quality.

Table 4.1 shows the comparison of TRELLIS-text-xlarge, TRELLIS-image-large and TRELLIS2 with resolutions 512 and 1024. As can be seen in the table TRELLIS-image-large were the fastest while TRELLIS-1024 provided the best overall quality. Another thing noted for TRELLIS-text-xlarge was that the model was not sensitive to specific prompting. When asked to create a shelf, it created similar types of shelves, no matter how diverging the prompts were.

Model	TRELLIS-text-xlarge	TRELLIS-image-large	TRELLIS2-512	TRELLIS2-1024
Time (s)	30	21	34	51
Modality	Text-to-3D	Image-to-3D	Image-to-3D	Image-to-3D
Material Quality	Decent	Decent	Poor	Good
Material Shape	Decent	Good	Good	Good
Prompt Sens.	Low	Medium	Medium	Medium
Reference	[48]	[48]	[47]	[47]

Table 4.1: Qualitative evaluation of inference on RTX 3090 GPU.

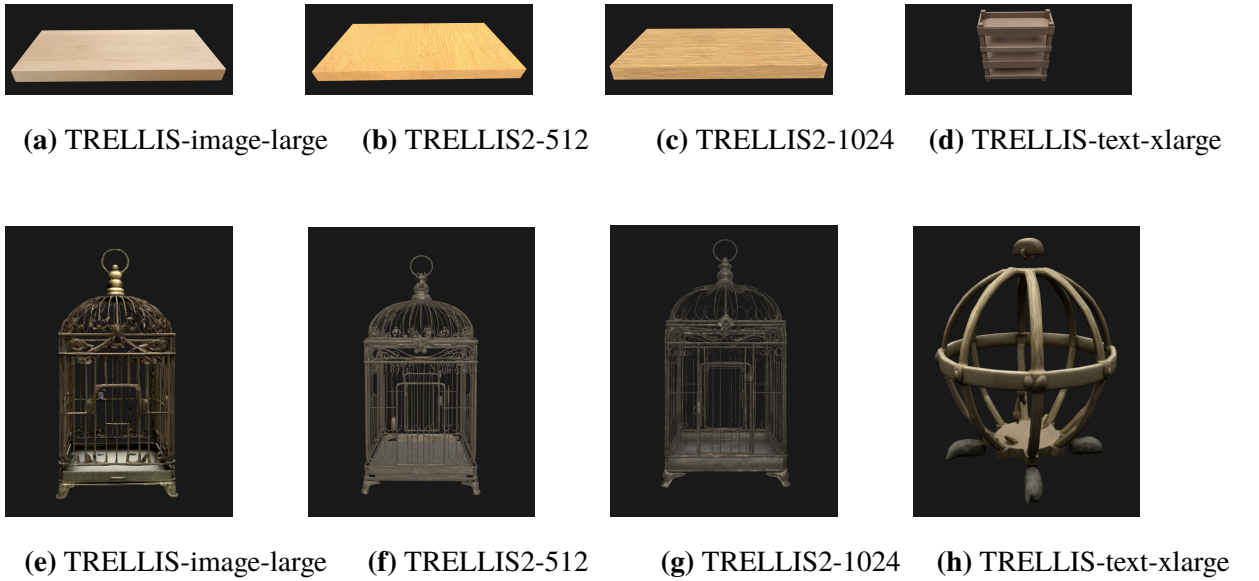


Figure 4.1: Comparison of 3D model quality of TRELLIS and TRELLIS2: shelves (top) and bird cages (bottom).

Figure 4.1 shows the comparison of objects produced by different 3D models, and 4.2 shows the reference image used for TRELLIS-image-large and TRELLIS2. The upper row was generated using the prompt "A single-tier shelf" and the lower row using the prompt "A bird cage."

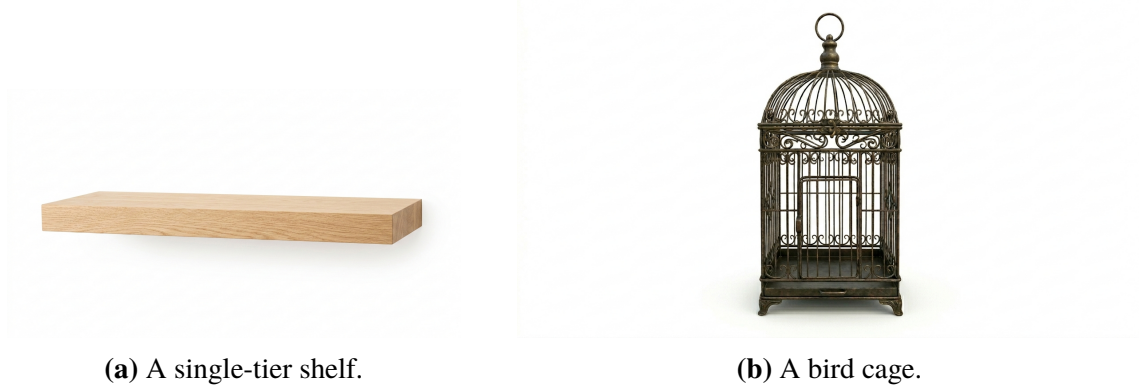


Figure 4.2: Reference images for the bird cage and shelf used to generate the 3D versions.

4.2 3D Object Placement

To address the 3D object placement, our research was performed similarly to the 3D object generation. We reviewed 3D object placement methods to identify the requirements necessary for our proposed solution. Then we used the information obtained from the literature review, as well as the advice from our supervisors, to evaluate different types of models that could be suitable for our 3D object placement problem.

4.2.1 3D Object Placement Methods

During the literature review, we found that solutions often constituted of several components and there were often more than one AI model used for the task. The literature review yielded several studies introducing diverse methods for 3D object generation. Most of the proposed solutions used a combination of 2D object detection models, combined with LLMs and other technologies to place the object in XR. These findings are presented in further detail below.

Sharma et al. [41] present a technique to generate 2D-placement coordinates, and convert them to 3D within the XR environment using ray-casts. This technique does not consider context-aware scaling of the objects. Steps towards 3D placement were taken by Xing et al. [49], who use the detected surfaces from the XR engine to place objects in suitable locations, as well as LLM-generated scaling. Instructing the user to scan their environment can include additional context such as scene graphs, proposed by Lee et al. [17]. Their solution leverages this context for an LLM to place and scale the object. Alternatively, context can be included in the form of Unity objects, proposed by Chen et al. [8]. Here, placement and scaling are also determined by an LLM.

Similar to the systems implementing 3D object generation, none of the identified existing 3D object placement methods consider user information as context while placing or scaling an object.

4.2.2 Spatial Understanding with VLMs

As mentioned, some existing solutions used the spatial understanding of VLMs to place the object. In our own benchmark tests, two state-of-the-art LLMs (Gemini-3 Pro and ChatGPT-5.2) were picked together with an annotated data set, CA-VQA [11]. This dataset contains dictionaries with different images, along with questions and answers related to the spatial relations in the image. We chose to use 10 samples from each dictionary: 2DGrounding, 3DGrounding, Regression, and Multi-Choice. In 2DGrounding and 3DGrounding, the dataset is annotated with questions and answers about 2D and 3D boundary boxes of objects in the image, while Regression covers distances between objects, and Multi Choice includes a variety of multi-choice questions covering relations and measurements of and between objects. Both Gemini-3 Pro and ChatGPT-5.2 performed quite well on the CA-VQA dataset; Gemini outperformed ChatGPT on all dictionary types. However, none of the models was able to provide any accurate 3D boundary boxes on 3DGrounding samples, which was our aim.

When introducing a depth map of the image together with the original image as input to the LLMs, the result did not improve on 3DGrounding samples. While both models (Gemini-3 Pro and Qwen Pro Max 3) provided boundary boxes, they did not correspond to the ground truth. Even when provided with more information, such as the camera intrinsics and pose, the performance did not improve. The conclusion from these tests was that although the spatial reasoning of these models was adequate, the specific 3D task would have to be aided by other technologies, such as 3D object detection models.

4.2.3 3D Object Detection Models

As our our state of the art exploration found that VLMs lacked the knowledge to provide accurate 3D coordinates on their own, this led to the exploration of 3D object detection models. There are a variety of models, presented in the literature, that tackle identifying 3D boxes in a room layout for indoor scenes, such as Factored 3D, 3D-RelNet and Total3D. Brazil et al. [6] introduce the model Cube RCNN, trained for the purpose of identifying 3D boundary boxes of objects in 2D images. This is the 3D object detection model selected for exploration, as it was specifically designed for indoor scenes and aligned with our use case. We found that more accurate 3D placement coordinates could be provided when the output of this model was combined with the VLM.

4.3 Explainability

Our literature study on explainable AI offered different perspectives on key areas of explainability in Generative AI. We were mainly concerned with the explanations for the generated object and its placement, and how it fits the user's request to increase trust and usability. In the identified existing instantiation solutions in XR, there were no attempts to implement explainability into the systems. This was identified as a limitation, leading us instead to explore ways to implement explainability in our own system. Hence, we researched the characteristics of good explanations and the self-explanation mechanism, as discussed below.

4.3.1 Characteristics of Good Explanations

A central part of this research explored how to best present explanations to the user. Schneider notes that LLMs can create interactive, explainable systems when guided by effective prompts [40]. However, the prompts must be carefully designed to ensure that the explanations satisfy the user's needs. To achieve this, we researched the specific qualities that make an explanation successful.

Schneider claims that an explanation is characterized by its faithfulness, plausibility, complexity, and completeness. Faithfulness, also known as explanation fidelity, means that the explanation reflects the inner workings of the model. Plausibility reflects how understanding and compelling the explanation is to the target audience. Completeness ensures that all the relevant factors for the decision are included. Finally, complexity controls that the information conveyed suits the user's knowledge [40].

Miller combines explainable AI with the social sciences, asserting that, as Schneider does, it is impossible to present all causes and that only the most prominent should be presented. Hence, the explanation must be selective and only choose the most relevant cause or causes. Contrastive explanations emphasize that explanations are typically comparative: they answer "Why did P happen instead of Q?" rather than simply "Why did P happen?". The contrastive form is usually more informative and better aligned with what users need to understand how a model reaches a particular decision [27].

Ultimately, these findings suggest that an effective LLM explanation must be faithful to the model's logic while remaining selective by focusing only on the most important factors. Also, it must be contrastive and tailored to the user's specific level of knowledge. To achieve this, the LLM must be prompted to find a balance between fidelity and understandability for the end-user. Since many users of XR solutions are not domain experts and have limited knowledge of AI and XR systems, the LLM must be aware of this knowledge gap to avoid overwhelming them with unnecessary technical details.

4.3.2 Self-explanation Mechanism

For LLMs, much research has been conducted in the field of self-explanations. The idea is that the LLM, which is text-based, should also provide explanations in text along with its output. One major advantage of LLM self-explanations is that they are easily accessible and comprehensible to non-experts, which is highly relevant given the limitations of access to the inner workings of large, proprietary LLMs.

However, explanations given by an LLM may be the product of hallucinations, and they might not accurately represent which input is attributed to the output [40]. To circumvent these issues, we need a way to verify and improve the explanations in order to enforce their fidelity, i.e., how well the explanation reflects the actual process of the model [9].

One way of improving the explanation fidelity, i.e., reducing hallucinated explanations, is to apply an optimization technique like the one proposed by Chuang et al. [9]. This technique gives an explanation for an LLM output, contradicts it in a negative direction, and then produces a new output using the contradiction. The new output shows the shift that the contrary hint causes. If the output remains the same, then the model has likely hallucinated the explanation, indicating low fidelity. If there is a difference in the new output, the explanation demonstrates self-consistency, which serves as a proxy for faithfulness or

high fidelity. Iteratively, contradictory explanations are fed back into the LLM until the difference drops significantly. By doing so, the final explanation is optimized for fidelity. We also note that, in this context, fidelity refers to local or causal fidelity, not mechanistic fidelity. This is because we do not have access to the inner workings of the system to compare how the model actually reaches a final output [34]. However, we found this technique to be highly relevant for reducing hallucinations in self-explanations.

4.4 Solution Requirements

The findings from our literature review and state-of-the-art exploration highlighted the need for a multi-modal pipeline. Current research increasingly employs gen-AI models for 3D-object instantiation, yet the inclusion of user and environment characteristics for true context awareness, as well as the integration of explanations into these machine-generated decisions, has been overlooked. The solution must go beyond simple object generation and placement, and incorporate user attributes and environmental constraints. These insights defined the three modules: generation, placement, and explainability. The solution requirements with the content needed to start the design and implementation phase are listed as follows:

- **Generation Module.** Through the literature review, we identified a lack of context-awareness because of already pre-established prompts and objects and an exclusion of user features when generating the objects. Furthermore, the state-of-the-art exploration showed that there is a trade-off between generation quality and latency, with the literature review highlighting that the quality of the object is more important than latency. Consequently, the solution should be able to take into account a variety of user characteristics, handle prompts for various types of objects and generate an accurate item. Additionally, the quality should be weighed importantly in comparison to the latency.
- **Placement Module.** The solutions from the literature review did not include user features when placing the object and the placement was often constrained, hindering true context-aware placement. Moreover, the state-of-the-art exploration showed that VLMs in combination with a 3D detection model are able to provide free 3D object placement. Hence, in the placement module the object needs to be placed in a natural and safe way based on the environment and the user. The placement should be in 3D coordinates and include scaling of the object.
- **Explainability Module.** None of the solutions from the literature review included explainability, however, for our thesis explainability is crucial. Therefore, in both object generation and placement, the system must provide the user with an explanation of the process that led to the decisions.

Chapter 5

Solution Design and Implementation

In the Solution Design and Implementation phase of our work, we laid out the plan for our solution in alignment with the identified requirements from the Problem Conceptualization phase. The design considered 3D object generation, 3D object placement, explainability, and the ways to tie it together into a unified system. The general solution design is shown in Figure 5.1. The implementation followed this design and considered constraints that were present in hardware and software. In the following, we discuss the generation, placement, and explanation modules in detail.

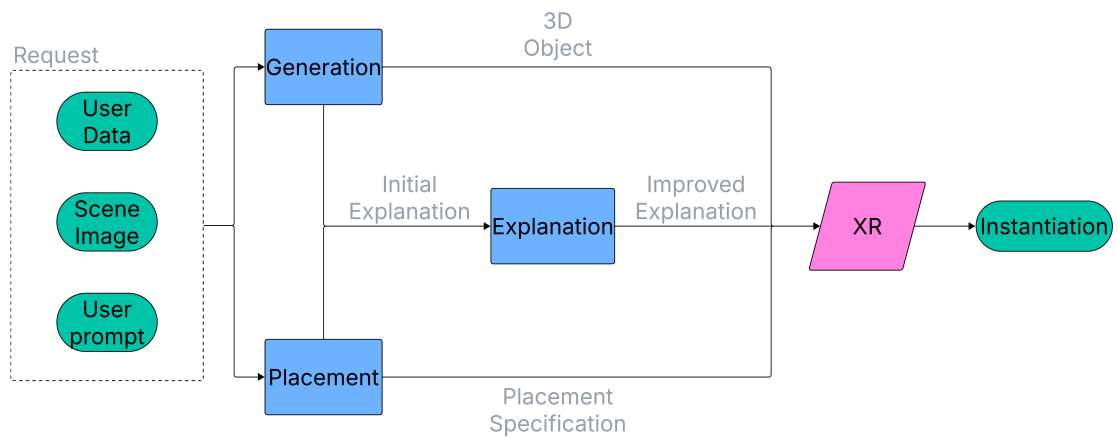


Figure 5.1: General solution design.

5.1 Generation Module

The generation module is responsible for taking a user request for an object, in accordance with the scene and user context, and generating a 3D object, which is then inserted into

the user's view. Figure 5.2 shows an overview of the pipeline needed to generate a 3D object. This module is divided into three sub-modules: instruction generation, text-to-image generation and image-to-3D generation.

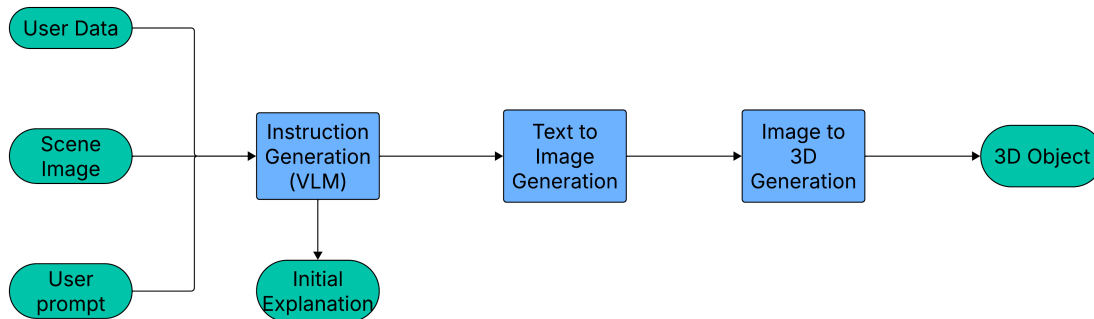


Figure 5.2: 3D object generation module design.

5.1.1 Instruction Generation

To create a context-dependent 3D object, it is crucial to properly instruct the text-to-image generation model. Since these instructions have to consider the context, we found it best to construct them through a VLM (GPT-5.4), utilizing user information and surroundings. Combined with few-shot prompting within our system (Appendix A.1), this provides both ideal problem understanding and consistent output. In other words, the VLM summarizes the context of the object into one or a few sentences, resulting in concise instructions for the text-to-image model. It also outputs a self-explanation as the initial explanation to the explanation module. Figure 5.3 shows an example of the generation prompt created from the given context. The user data consisted of height, age, and style, while the scene context consisted of the captured image and user prompt (requested object by the user).

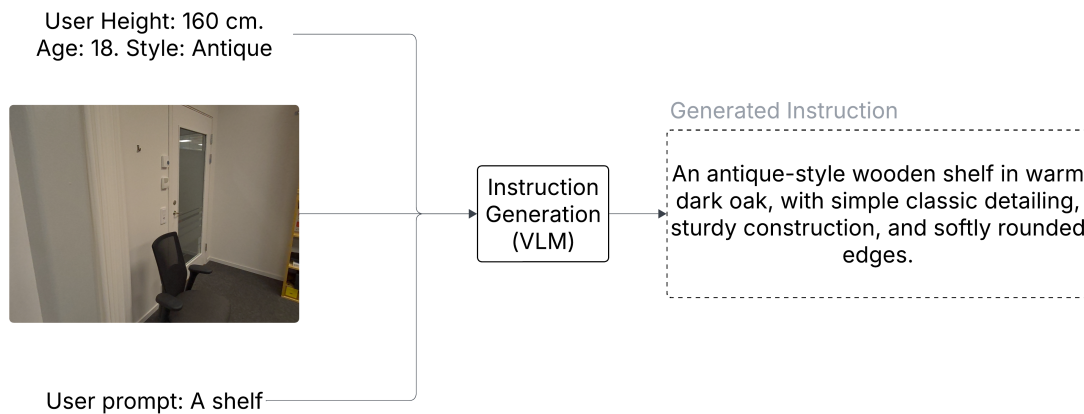


Figure 5.3: VLM-generated instructions from context.

5.1.2 Text-to-3D Object Generation

Once the instruction is created, it is first fed into the text-to-image model (GPT-image-1), along with additional instructions within the system. These include instructions to isolate the object against a plain background, so that only the object is visible in the image fed forward to the 3D object generation model (TRELLIS-image-large). Figure 5.4 shows an example of an image generated from the instructions.

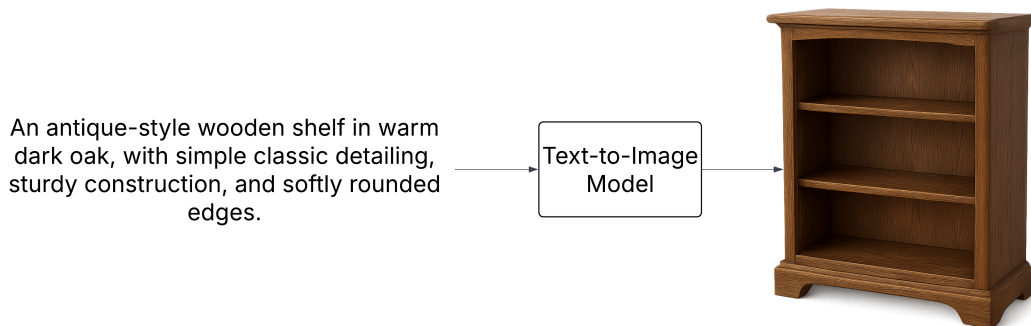


Figure 5.4: Image generated from instruction prompt.

From the generated image, a 3D model is constructed through the Image-to-3D model (TRELLIS-image-large), as can be seen in Figure 5.5. To improve speed, only one image of the object is provided, rather than multiple images from different angles.

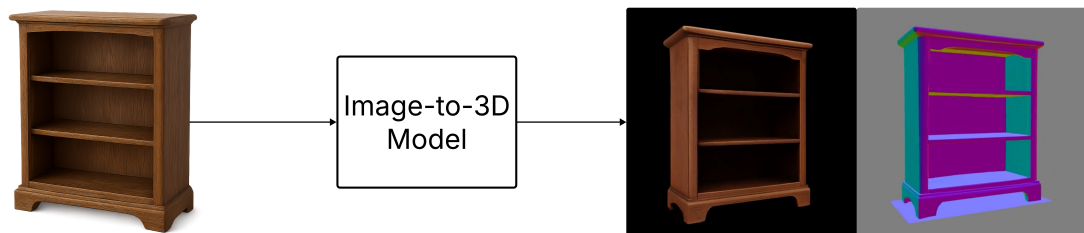


Figure 5.5: 3D model generated from image.

5.2 Placement Module

The placement module is responsible for determining an optimal placement and scaling for the newly requested object, based on the user and scene context. The placement is specified in 3D coordinates, and scaling is represented by a single coefficient. The module design is shown in Figure 5.6.

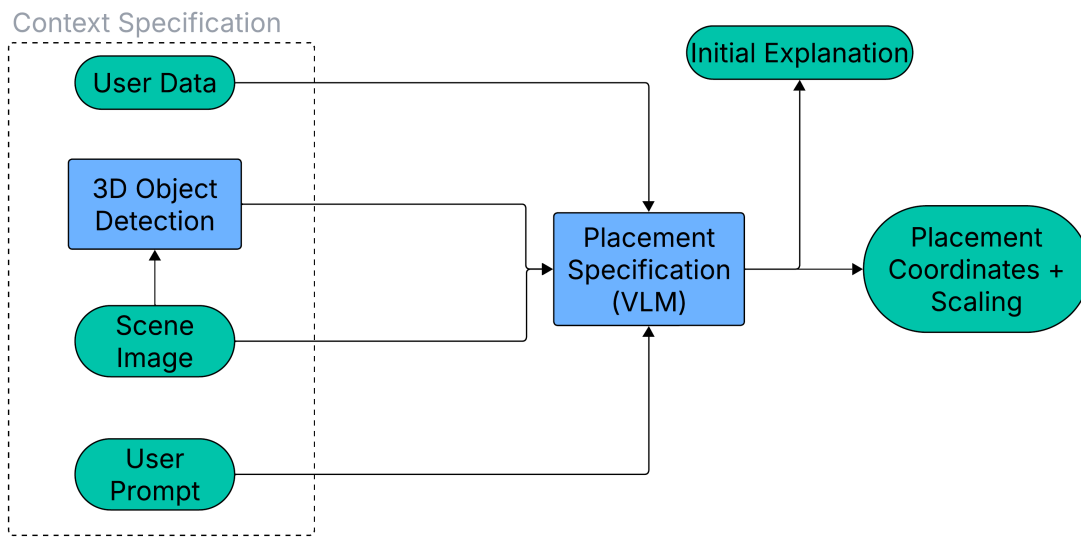


Figure 5.6: Placement module design.

5.2.1 Context Specification

For our solution to process context and reliably produce a matching position for an object, it needed to incorporate some form of semantic understanding and reasoning. The spatial reasoning capabilities of VLMs discussed in Chapter 4 were considered to be a key component of this part of our system. However, spatial reasoning was not enough to provide exact 3D coordinates for the placement of an object in a scene.

In order to ground the VLM (GPT-5.4) with more relevant information about the image, it is also given the 3D coordinates of detected objects in the image. The detected 3D objects that are fed to the VLM are provided by the 3D object detection model (Cube R-CNN). The detected objects belong to a set of classes that the model was trained on, including many common household categories. The detections include both translation (x,y,z) and size along the same axes. Additionally, the rotation of the objects can also be obtained. All this information is used to further assist the VLM in its placement decisions for a new object. Besides producing the placement and scaling, the VLM also outputs a self-explanation as the initial explanation to the explanation module.

Besides being aware of the constraints of the environment, the goal of our solution was to also consider the user-specific information. This is achieved by incorporating user-provided data in the categories height, age, and style. This affects the placement in both translation and scaling, while staying in line with the surrounding constraints of the scene.

5.2.2 Placement Specification

When presented with the Context Specifications, the VLM (GPT-5.4) produces a placement specification (P) for the new object. This is given as a vector of (x, y, z, s) , where x, y, z are cartesian coordinates in camera space, and s is the scaling factor with which to scale the object. Then, these values are passed to the user program to place the object

in the XR view. Since the coordinates are given in camera space, they are transformed to world coordinates to match the XR program origin. This way, the objects stay fixed in place as the user moves around. Our system does not dynamically determine object rotation based on the input features, but always places the object rotated in the Y-axis towards where the user was standing when sending the request.

5.3 Explanation Module

The explanation module is the mechanism behind the system’s transparency, which enables the user’s deeper understanding of the decisions from the system. To optimize the explanations for fidelity and selectivity, we adapted a technique proposed by Chuang et al. [9].

Figure 5.7 illustrates the main components of the explanation module. The VLM’s self-explanation is the initial input to this module, containing all available features. The module then iteratively identifies and evaluates individual features through a feature attribution loop, ultimately selecting the highest-scoring explanation as output. In the following, we discuss this in detail.

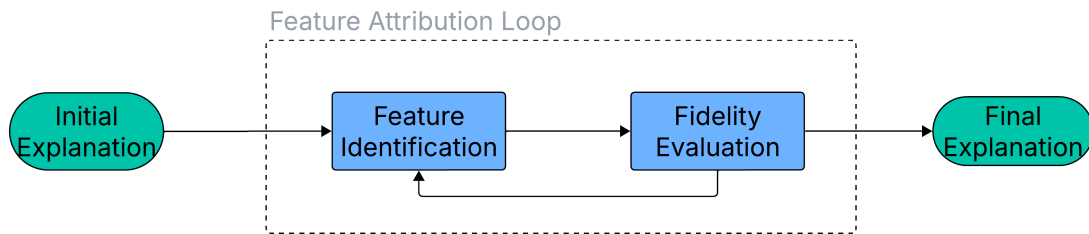


Figure 5.7: Explanation module design.

5.3.1 Feature Attribution Loop

The feature attribution loop contains feature identification and fidelity evaluation. Figure 5.8 shows the detailed steps in the feature attribution loop, and Figure 5.9 shows an example of one iteration. As a first step in our feature attribution, GPT-5.4-mini is prompted to identify a key component of the explanation, where key components are the different context sources, including height, age and style of both user and scene. It then uses this key component to create a concise and selective explanation. Next, GPT-5.4-nano is used to generate a contrary hint to the key component in the explanation. For example, if a key component of the explanation is “the bright color of the basket was chosen, because of the user’s preference of light colors”, a contrary hint could be “the murky color was chosen because of the user’s preference towards dark colors”. Using the contrary hint, a new response to the original prompt is generated using GPT-5.4. Then, the fidelity score is calculated as the difference between the original output and the output generated after the contrary hint. A large difference indicates that the feature strongly influenced the decision, thereby serving as a measure of feature importance.

The new explanation, along with its corresponding fidelity score and contrary hint, is appended to a trajectory list, which contains all explanations from earlier iterations. When the next explanation is constructed, it focuses on a feature which has not yet been evaluated. This process is repeated for a set number of iterations to measure the fidelity of several different features. The number of iterations is equivalent to the number of input context sources. In our case, this is four iterations as the features identified are height, age, style, and the scene. Once finished, the explanation with the best score is chosen as the final explanation. Since we separate the explanations for 3D object generation and 3D placement, the best-scoring ones from each task are combined into the final explanation.

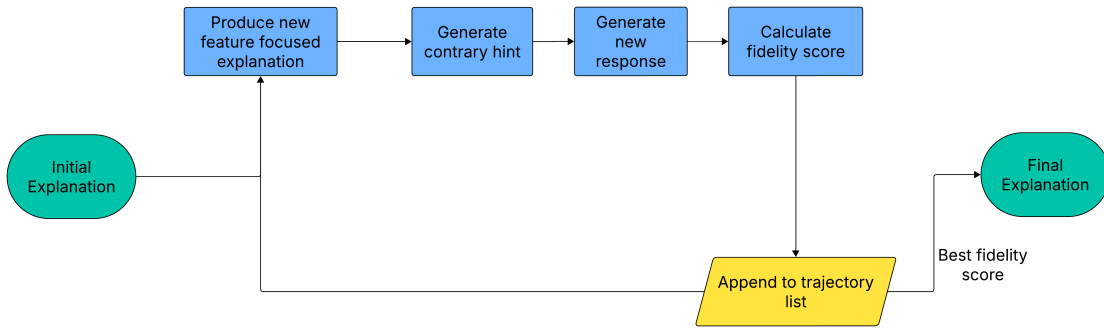


Figure 5.8: Detailed flow of feature attribution loop.

Fidelity score

The explanation was adapted to handle the generation and placement explanations separately, as the calculation of the fidelity score differs based on the type of explanation. In both cases, a larger score means that the explanation considers an important feature for the system.

For the 3D object generation part, the comparison between the different instructions is made using embeddings. Both instructions are converted into high-dimensional vectors through an embedding model, and their difference is measured using Cosine Similarity [33]. An inverse of this similarity is used as a fidelity score, as seen in Equation 5.1.

$$\text{Score}_G(\mathbf{A}, \mathbf{B}) = 1 - \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} \quad (5.1)$$

Where A, B are embedded generation instructions.

The fidelity score of the placement explanation is computed through an inverse of the exponential of the negative Euclidean distance between the points in 3D space. In order to also factor in the scaling of the object, an additional term is added, as seen in Equation 5.2.

$$\text{Score}_P(P_1, P_2) = 1 - \exp(-(\|u_2 - u_1\| + \|s_2 - s_1\|)) \quad (5.2)$$

Where P_i is a placement specification consisting of (u_i, s_i) , and u_i consisting of (x_i, y_i, z_i) . u_i signifies the 3D coordinates and s_i is the scaling factor.

5.3.2 Explanation Prompting

Explanation prompting involves different strategies for different sub-tasks. To make the explanations more clear and appealing, we used different characteristics of explanations from the literature in our prompting. This is discussed in detail as follows.

Prompt for Initial Self-explanations. We instruct VLM to keep its initial explanations contrastive and avoid technical language, to suit the average user. We also instruct it to focus on all relevant objects and features that it has found meaningful for completeness. We defined the instructions for the generation task in Appendix A.1 and for the placement task in Appendix A.2. Besides the tasks being different, the instructions for specific explanation characteristics are the same. In order to improve the consistency and structure, we adopted few-shot prompting in these prompts.

Prompt for Feature Identification. When identifying features and producing new explanations, the focus is to enhance selectivity and fidelity, given in Appendix A.3 and A.4. Appendix A.3 defines the task for the generation explanation, while Appendix A.4 defines the task for the placement explanation. The explanations for both tasks are instructed to contain a maximum of 50 words and only focus on one feature each. The prompts also contain instructions to not include information which does not belong to the task, so that the generation explanation does not discuss placement, or vice versa. The prompts also ensure that only one feature is considered per explanation, in order to ensure selectivity.

Prompt for Contrary Hint Generation. Producing contrary hints for generating or placing an object involves finding essential input features that affect either of the tasks. We defined the logical inversions of such features for generation and for placement in Appendix A.5 and A.6. The definitions make use of few-shot prompting to ensure the consistency of inversions. Contrary hints also involve altering the user’s height or changing the semantics of an important feature in the scene.

Initial generation instruction: A modern adult commuter bicycle with a sleek medium-large frame, matte black finish, clean lines, integrated lights, and subtle metallic details.

Initial explanation for generation: A modern commuter bicycle was chosen instead of a playful BMX or rugged mountain bike, since the user asks for a modern style and the scene shows a clean, neutral, tech-oriented room with screens, metal equipment, and minimal colors. The medium-large adult frame matches a user height of 180 cm better than a compact frame would. Because the user is 48, a refined and practical design fits better than a youthful or highly extreme sports look. The prompt keeps the bicycle streamlined and safe, avoiding unnecessary aggressive or risky features.

New selective explanation: A refined, practical appearance (matte black, clean lines, subtle metallic details) was chosen to suit a 48-year-old user's preference for mature, understated aesthetics.

Contrary Hint: The design should be glossy and vibrant, featuring bright colors, bold lines, and conspicuous metallic accents to create a flashy, youthful look instead of the refined, matte black, understated aesthetic.

New generation instruction: A glossy, vibrant bicycle with bright colors, bold lines, conspicuous metallic accents, a clean modern frame, standard handlebars and pedals, and safe realistic proportions.

Score: 0.2699344860086982

Figure 5.9: Example of explanation module for generation

Chapter 6

Solution Evaluation

This chapter presents the results from our quantitative and qualitative evaluations of the system. The purpose of the evaluations were to test the context-awareness and explainability of our solution. The quantitative evaluation examined if certain user features or the environment had more impact on the output of the system. The qualitative evaluation analysed how the users perceived the system's context-awareness and the provided explanations.

6.1 Quantitative Evaluation

The quantitative evaluation assesses the context-awareness of the system through sensitivity analysis, conducted separately for 3D object generation and 3D object placement. The results for generation and placement, are described in the following subsections.

6.1.1 Sensitivity Analysis for 3D Object Generation

Figure 6.1 showcases how the semantic perturbations affect the output result based on different prompts. In this figure, a higher bar value signifies that a specific input feature had a more pronounced effect on the output.

Sensitivity Analysis for 3D Object Generation

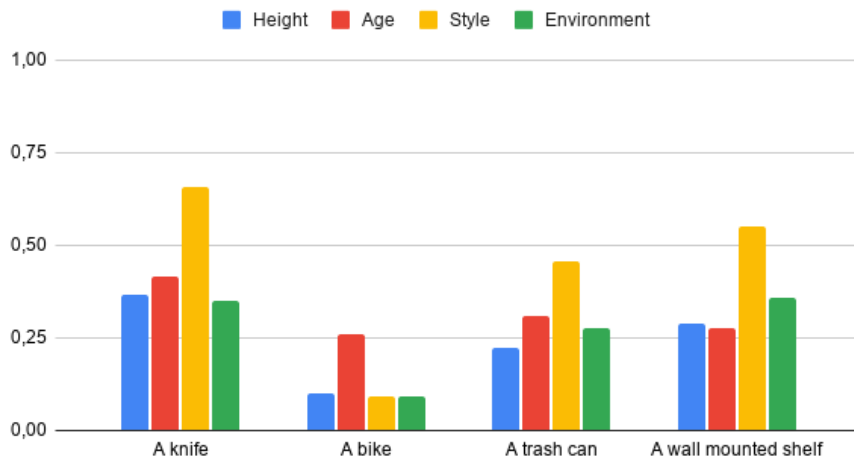


Figure 6.1: Impact of perturbed features on sensitivity scores across four object categories.

As illustrated in Figure 6.1, the model’s sensitivity to specific input features varied across the different prompts. The model demonstrated high sensitivity to the style across three of the four prompts. For bike, however, the most prominent feature is age.

Table 6.1: Examples of the images generated based on the baseline, and perturbed features.

Prompt	Baseline	Age	Height	Style	Image
A bike					
A knife					
A trash can					
A wall-mounted shelf					

Table 6.1 illustrates the baseline images alongside the generations resulting from the feature perturbations detailed in Table 3.1. While prompts like "A knife," varies a lot in

design depending on feature perturbed. "A trash can," and "A wall-mounted shelf" exhibit variations in shape and detail depending on the perturbed feature, "A bike" maintains a consistent appearance independent on features.

6.1.2 Sensitivity Analysis for 3D Object Placement

Figure 6.1 illustrates how various semantic perturbations affect the placement specification of the selected prompts relative to the baseline. Similar to the sensitivity analysis of the generation, a high score indicates a great shift in output, while a low score indicates little or no change in output.

The perturbation of the environment image showcased sensitivity across all object types. The placement of a bike was sensitive to all user feature perturbations, whereas a wall mounted shelf remained largely unaffected by user data changes, responding only to the modified environmental image. A trash can was also only mildly sensitive to user data changes. Age showed an effect on the placement of a knife, accounting for the highest, non-environment feature.

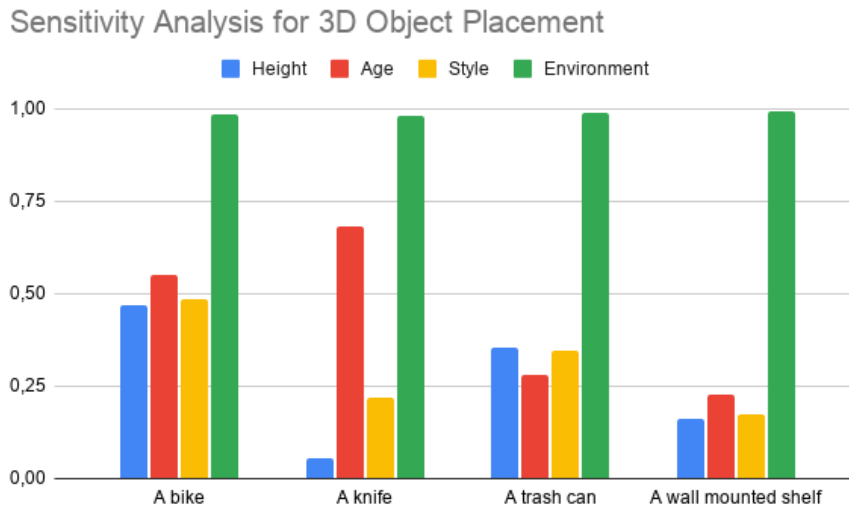


Figure 6.2: Impact of perturbed features on sensitivity scores across four object categories.

6.2 Qualitative Evaluation

The qualitative evaluation assesses the system's context-awareness and explanation through a user study. Unlike the sensitivity analysis, which evaluates system outputs objectively, this phase captures user perception of the instantiation and explanation quality. The user study results are described in the following subsection.

6.2.1 User Study Results

Users reported varied scores for perceived inclusion of different personal features in generation, as can be seen in Figure 6.3a. Age and personal style scored highest, both being above 4. The perceived inclusion of the scene design in generation was generally around the midpoint, as can be seen in Figure 6.3b. Colors of the scene scored higher than both the Materials and Style.

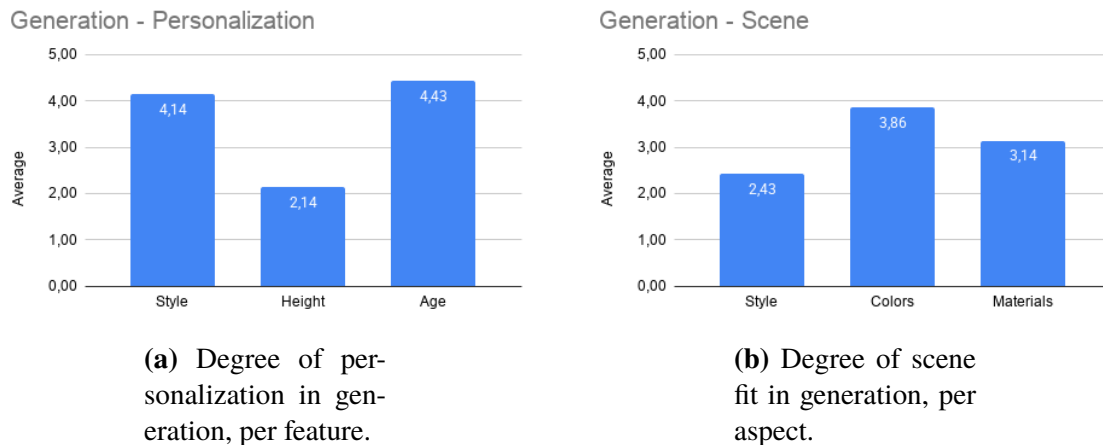


Figure 6.3: Qualitative results for 3D object generation.

The inclusion of personal features in placement was the highest for age, as can be seen in Figure 6.4a. Style and height were reported lower, slightly above 2. Inclusion of personal features for object size was reported above the midpoint for all features, with age scoring the highest around 4 (Figure 6.4b).

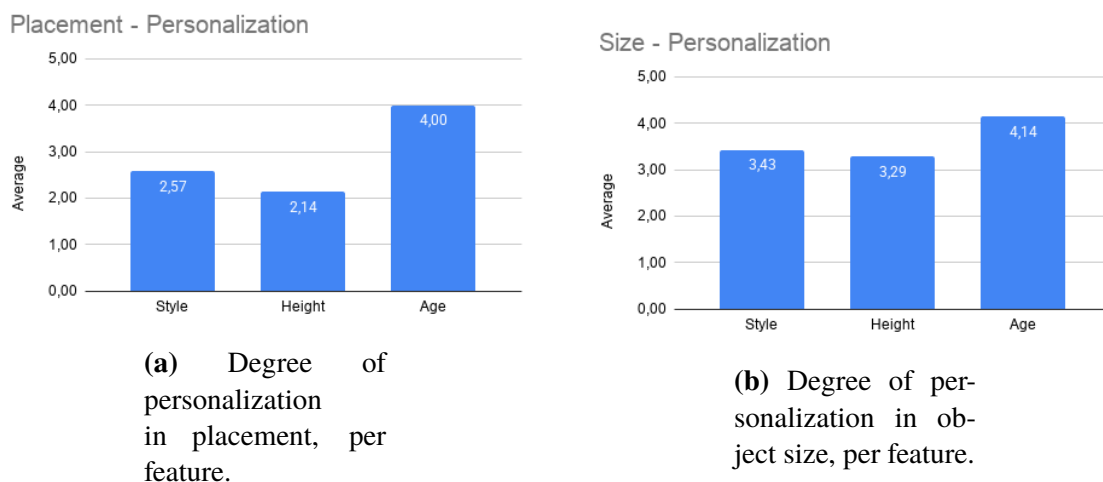


Figure 6.4: Personalization in object placement and size.

Scores for placement and size relative to the scene received different scores. Size in relation to the scene context was rated high, while placement in relation to both context and physical space was rated low (Figure 6.5).

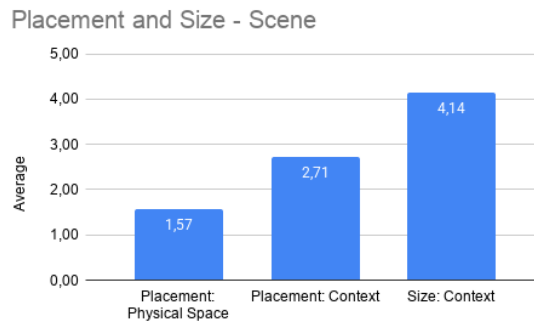


Figure 6.5: Degree of scene fit for placement and size, per aspect.

Users reported high scores on the overall clarity of explanations, as seen in Figure 6.6a. Perceived clarity of personal influence was rated high, while scene influence was rated around the midpoint. Users reported high on trust in generation due to explanations (Figure 6.6b). Trust in placement due to explanations ranked a bit below the midpoint, while overall system reliability due to explanations was a bit above the midpoint.

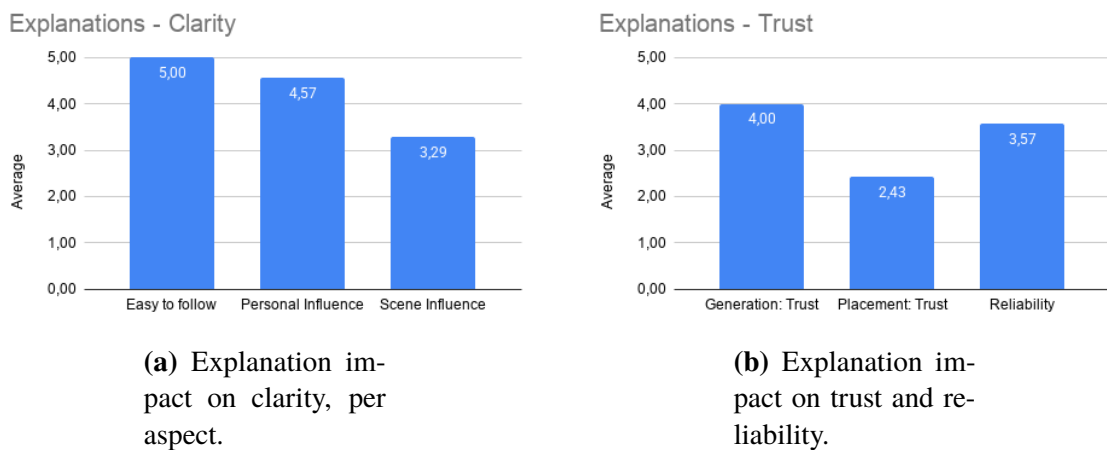


Figure 6.6: Explanation impact.

In general, users voted highly on whether they preferred the system with explanations over one without (Figure 6.7a). For the instantiation time and explanation length, user scores can be seen in Figure 6.7b. The time for instantiation was rated relatively long, while the explanation text length was around the midpoint.

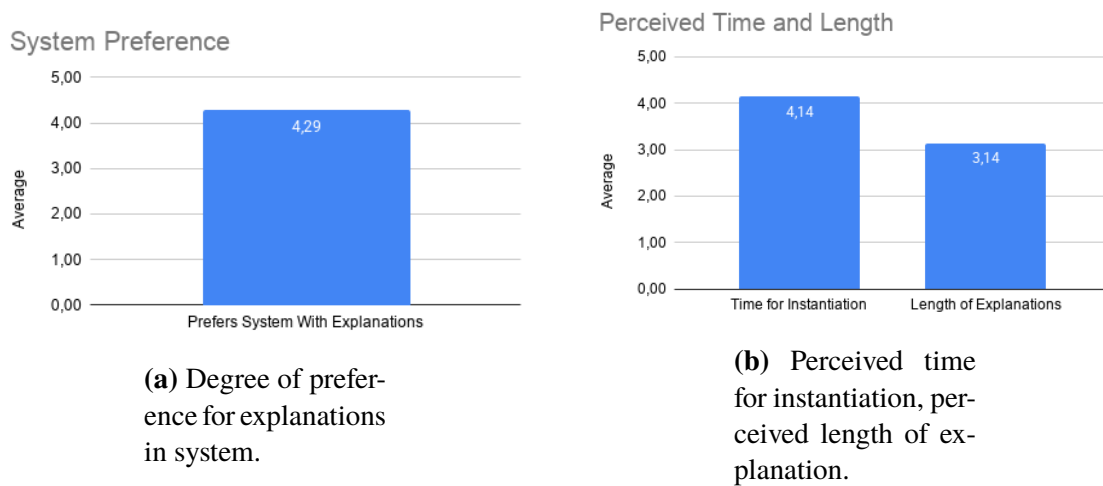


Figure 6.7: Explanation preference, perceived time and length.

Table 6.2 shows identified themes in the open responses. For general comments, almost all had comments about the placement being faulty. Some users expressed irritation with user-centered explanations and wanted more focus on the scene. Also, the contrast between the actual scene and the explanations was something that was mentioned.

Theme	Sample comment
Strange placement	"Sometimes objects weren't placed in a realistic manner, i.e., floating chairs and tilted paintings."
Unrealistic generation	"The painting did not really seem like a painting."
Unfaithful placement explanation	"The explanation was not always correct about placement and size."
Personal focus of explanations	"It just focused on my age and height."

Table 6.2: Themes in general comments about the system.

Figures 6.8a and 6.8b show the relationship between user-chosen explanations and the corresponding fidelity scores, for generation and placement, respectively. The vertical axis is the inverse difference in fidelity score between the user-chosen explanation and the most faithful explanation. The horizontal axis shows the range between the most and least faithful explanation. Blue dots represent each user choice of explanation, while the blue line is the general trend.

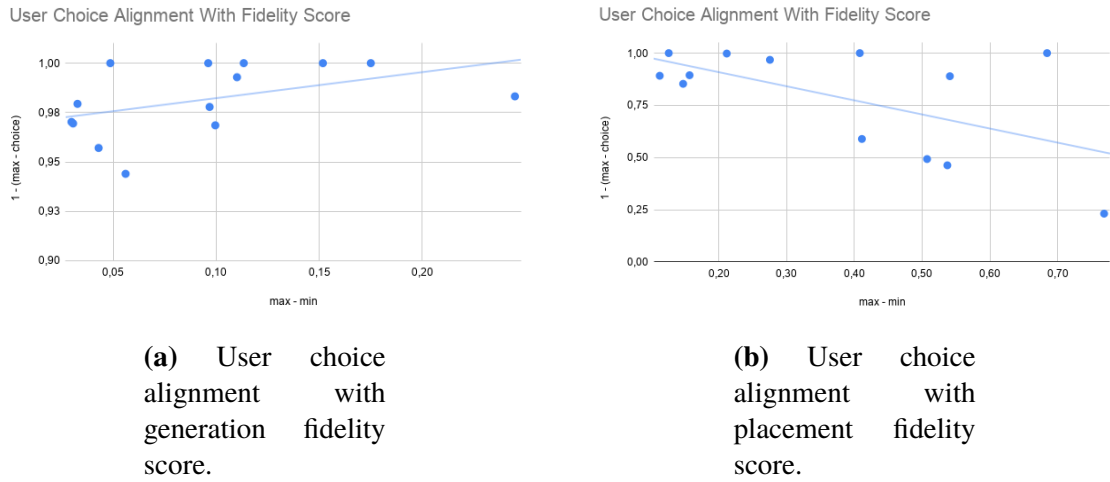


Figure 6.8: User choice alignment with fidelity scores.

Figures 6.9a and 6.9b show the prevalence of input features in the user-chosen explanations. The most chosen input feature for generation was style, followed by scene. For placement, the most chosen input feature was height, followed by scene.

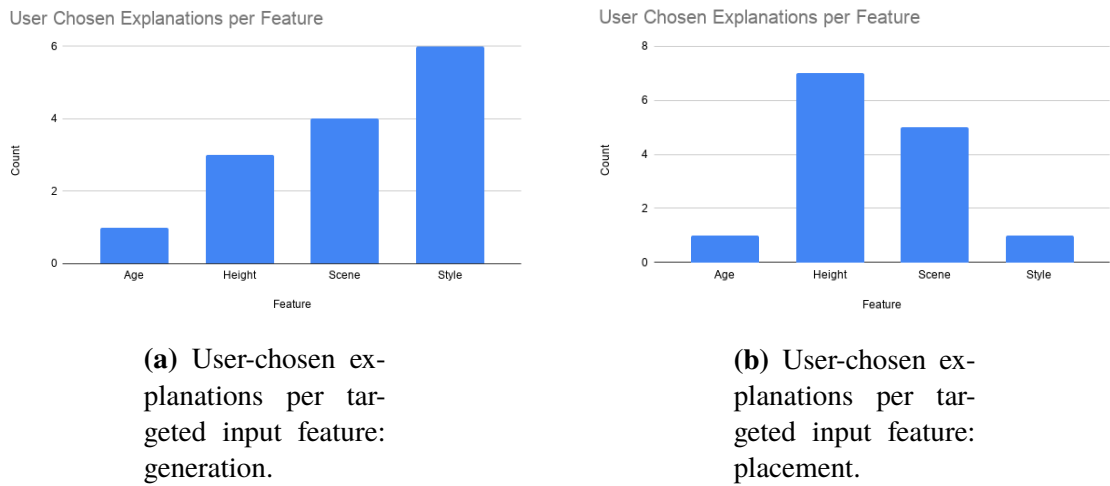


Figure 6.9: User-chosen explanations per feature.

Tables 6.3 and 6.4 show user motivations for choosing their preferred explanation, along with example comments from the study. Explanation qualities that were common to both generation and placement were concreteness, faithfulness, valued features and alignment.

Theme	Sample comment
Concreteness	<i>"Taking into account height and geometry as opposed to some of the other ones which I think are more subjective."</i>
Faithfulness	<i>"This is what I actually thought the model had used to generate it."</i>
Valued features	<i>"Talking about the user is not something I would like to have."</i>
Alignment	<i>"I think the style is the most understandable part of the generated painting"</i>
Sentence Structure	<i>"It is the most to the point"</i>

Table 6.3: Themes in motivations for explanation choice: generation.

Theme	Sample comment
Concreteness	<i>"It feels more concrete and fitted to me personally which I prefer."</i>
Faithfulness	<i>"I think this explanation fits best with how the placement actually was done."</i>
Valued features	<i>"This is the aspect I value the most when I place a painting on my walls."</i>
Alignment	<i>"I think this explanation most closely aligns with the placement that was actually chosen and the explanation makes sense."</i>
Explaining faulty placement	<i>"The artwork was not aligned with my height, and I want to know that XR believes that it is."</i>

Table 6.4: Themes in motivations for explanation choice: placement.

Chapter 7

Discussion

In this chapter, we discuss our research questions based on the findings presented earlier. We answer each question separately, then discuss limitations and future work.

7.1 Limitations of Existing Solutions (RQ1)

During the Problem Conceptualization, we found several limitations related to context-awareness and explainability in the existing instantiation solutions for XR. The limitations were identified mainly through the literature review and state-of-the-art exploration, including generation quality, placement constraints, limited context-awareness and absence of explainability.

7.1.1 Generation Quality

Our research, presented in Sections 4.1.1 and 4.1.2, demonstrates a clear trade-off between latency and generation quality. Faster models, like Shap-E, create blocky shapes and blurry textures compared to higher-quality models such as TRELIS2. While Shap-E offers lower latency, the reduced quality of the generated object compromises user immersion. For example, when a user observes an asset with low geometric or texture quality placed within a physical room, it creates a stark visual contrast that disrupts the user's sense of presence in the environment. This argument is further supported by Xiu et al. [50], who explain that users are willing to wait longer if that means receiving an object of higher quality. Consequently, these insights justify our decision to prioritize high-quality generation pipelines over lower-quality, faster alternatives, ensuring that the generated object integrate with the environment naturally.

7.1.2 Placement Constraints

The literature highlights several constraints regarding object placement, ranging from systems that only place objects on detected surfaces or require lengthy environmental scans, to those that provide no automatic placement at all. Solutions that completely lack direct placement allow users to adapt the layout exactly to their liking. However, this approach can also place an unnecessary workload on the user, and this workload will grow larger as the number of objects increases. This reliance on manual positioning slows down the use of the system and assumes that the user has an abundance of time. Additionally, if the purpose of the system is to automatically organize virtual objects, e.g. in an interior design program, automatic placement is a must.

While ray-casting is a commonly used method to place objects, it does not allow objects to be positioned behind planes. Additionally, the functionality is entirely dependent on the specific XR setup and makes the solution less portable. This limitation means that ideal placement may be disregarded by the system as it cannot identify it. This can degrade the user experience and make the system appear less context-aware. Similarly, placement methods that rely on scene graphs negatively impact the user experience, as they are often time-consuming and highly vulnerable to any changes within the physical environment. Combining these limitations, we argue that prior works treat placement as a hardware or geometric constraint rather than a semantic task. These limitations motivate the use of automatic placement in our solution, as well as free 3D coordinates generated by a VLM, enabling placement without predefined surface constraints.

7.1.3 Limited Context-Awareness

Context-awareness varied considerably across the reviewed solutions, but none achieved full context-awareness with respect to both the user and the environment. As presented in Section 4.1.1, several systems relied on predefined prompts or fixed 3D assets rather than allowing users to dynamically decide what objects to generate, which limits adaptability to individual users. Moreover, other systems relied exclusively on data from the XR environment’s internal tools, resulting in the complete exclusion of a semantic understanding of the physical scene. This exclusion means these systems lack a comprehensive view of the environment. If a system cannot account for the vital components of the environment, it cannot be considered truly context-aware. Furthermore, restricting placement to predefined surfaces limits spatial context-awareness, preventing the layout from fully adapting to the user’s needs. Ultimately, the failure to consider user attributes and environmental semantics highlights the necessity of an approach that uses both user features and environmental context.

7.1.4 Absence of Explainability

None of the solutions for 3D object instantiation systems included Explainable AI in their solutions. This is a critical gap, as users interacting with generative systems in XR have no means of understanding why a particular object was generated or placed in a given way. Without a justification mechanism when the system fails or behaves unexpectedly, users

are likely to develop mistrust, ultimately leading to system abandonment. This motivated the decision to add an explainability module to our solution.

7.2 Design and Implementation Decisions (RQ2)

The Design and Implementation of our 3D object instantiation solution in XR followed the requirements identified in the Problem Conceptualization. Many decisions were taken during the Design and Implementation in order to address the limitations found in existing solutions. These decisions are discussed in this section.

7.2.1 Addressing Generation Quality

Several existing solutions employ lower-quality 3D models in order to decrease latency (Section 4.1.1). This is a limitation in the realism of the scene, thereby decreasing alignment of the solution with the environment, which is one of the context sources it should adhere to. To address this limitation without extreme latency increases, we opted to use TRELLIS-image-large in our solution. This model was chosen due to its superior texture quality over other available options [50], while still retaining a faster inference time than larger models, such as the TRELLIS2 models (see Section 4.1.2). Additionally, it had the advantage of prompt sensitivity over the TRELLIS text models (see Section 4.1.2), which makes custom objects possible for a variety of scenarios. The choice of a high quality, sensitive object generation therefore aimed to increase the context-awareness in the system, since it enables objects to be tailored to both the environment and the user.

7.2.2 Addressing Placement Constraints

A limitation found in existing instantiation solutions for XR is the constrained placement in the scene, usually by detected objects or surfaces (Section 4.2.1). This leads to the solution being unable to fully utilize the context of a 3D scene, and prevents decoupling of the system. We addressed this limitation through 3D coordinate placement in user space, determined from the image of the user's view. This decision aimed to enable placement in all parts of a room, therefore not constraining the semantic understanding that is present in the placement process. We believe that this choice enables the solution to execute on its understanding of the environment and the user, therefore enabling context-awareness.

7.2.3 Addressing Context-Awareness

User context is seldom considered in existing solutions for 3D instantiation in XR (Section 4.1.1, 4.2.1). This is a limitation in appealing to different types of users, as well as considering safety and appropriateness. To address this, we incorporated three different user context features in our system: age, height and style. The choice of these features aimed to expand on the relevant context for both generation and placement. Age was considered

relevant mainly for generating child-friendly objects when appropriate, but also to tailor to the different needs of different adult ages. Height was mainly chosen to tailor placement on walls and some object sizes. Style was incorporated in order to address the user preference for a scene, without the need to include it in every request. We concluded that these three features represented the most impactful dimensions of user context for instantiation, specifically, as they directly shape both what is generated and where it is placed.

7.2.4 Addressing Explainability

A major limitation in existing instantiation solutions for XR is that explainability is not present to the user (Section 4.1.1, 4.2.1). This creates a lack of understanding between the user and the system, and may decrease user trust. We took steps to address this limitation through the explanation module. Self-explanations were chosen as the base of this module, due to the black-box nature of the system. While this approach does not cover every component in the system, it targets the VLM responsible for the core generation and placement decisions, which are the most consequential for the user. In order to mitigate hallucinations in this approach, we chose to measure fidelity for different input features to improve faithfulness. Our explanation method focused on one input feature at a time, producing selective and concise outputs suited to an XR context, aimed at minimizing cognitive load. This approach was considered to be the ideal explanation method for an interactive XR application, aimed at increasing user trust.

7.3 Performance Evaluation (RQ3)

The evaluation of our XR instantiation solution followed both a quantitative and qualitative approach, with the results discussed in this section. We discuss the results from context-awareness in generation and placement, as well as user perception of explanations.

7.3.1 Context-Awareness in Generation

With regard to context-awareness in generation, we found that style had clear impact on three of the tested prompts, while age was only a clear influence in one of them (Figure 6.1). This is also consistent with user evaluation, where age and style were perceived as most impactful for generation (Figure 6.3a), while environment did not have the same impact (Figure 6.3b). These results suggest that the generation module is biased towards personal style, rather than the style of the environment, which would explain some user comments (Table 6.2). The implication of this might be that objects do not align with the environment, and thereby break the realism of the scene. In light of this, context-awareness in the generation module of our system seems to be only partially context-aware.

Over all types of objects and features, it is hard to identify any clear correlation (Figure 6.1). This could indicate that some types of objects contain more inherent variation than others. It could also indicate that some types of objects are more sensitive to the changes of certain features. For example, a bike adjusted for a child might be colorful and have training wheels, but carry cleaner lines and more mature colors for an adult. In contrast, a shelf may be designed similarly for a child and an adult, and here the style might be

the differentiating factor instead. In general, both age and style affect generation in our system, but their effects across different object types are not certain, and all features need to be studied further for conclusive results.

7.3.2 Context-Awareness in Placement

For the placement module, perturbation of the scene shows the most clear influence on the placement specification, while other personal features have a much lower effect (Figure 6.2). This is expected, since the object placement should be mapped to the scene. The smaller contribution of personal features is however less expected, and is likely due to the sensitivity measurement. Both a bike and a trash can are objects that need to be placed on the floor, while a wall mounted shelf needs to be put on a wall. Judging from the available scene, the floor area is larger than the available wall space. This can account for the smaller sensitivity to user features for the shelf, while more variation is possible for a bike and a trash can. Completely changing the scene can also be considered a larger change of context than changing user age, which explains the significance of environment context. In general, the environment seems to impact the placement specification in a major way, while user features have less of an impact.

In the sensitivity analysis, one exception from the diminished user features is the somewhat large effect of age on knife placement (Figure 6.2). This suggests that a knife is moved for safety reasons when younger age is entered, especially since this effect is the largest for a dangerous object. This shows some semantic context-awareness in the solution, and enhances the safety of the user experience.

While age scored high in the perceived impact of personal features on placement, height scored relatively low (Figure 6.4a). This indicates that the placement of objects did not appear tailored to the user's height. This result also aligns with several user comments, which stated that object placement was faulty at several times (Table 6.2). On the other hand, the size of objects scored relatively high on all features (Figure 6.4b). These results can be explained the increased task complexity when increasing the dimension to three coordinates. The task complexity is also noticeable since the system is aware of the user's height in explanations, but often fails to incorporate it effectively in its decision. This suggests a discrepancy between the model knowledge and placement specifications. One reason for this discrepancy is likely the difficulties for VLMs to accurately predict 3D coordinates in an image, even when the coordinates of existing 3D objects are given. Generally, the placement module seems to possess context-awareness regarding personal features, but varies in its ability to execute on them.

In contrast with the quantitative evaluation, scene features had a perceived low impact on placement (Figure 6.5). This, too, aligns with user comments describing placement which broke physical space boundaries (6.2). This includes the rotation of the objects, which were not implemented in a dynamic way, and hence occasionally broke the realism of the scene. The size of the objects was, however, considered appropriate to the scene, which may again be a result of size being a less complex task for the VLM than the placement in three coordinates. Because of this, we argue that the scene features have a lower perceived impact on placement than what quantitative results show, due to the complexity of 3D coordinates for the VLM.

7.3.3 User Perception of Explainability

The clarity of the explanations was rated highly in general (Figure 6.6a). This may be explained through the theory of cognitive load, which indicates that a shorter, more selective explanation may be easier to follow. The length of the explanations was also rated neither very short, nor very long (Figure 6.7b), which is an indication that explanations felt natural to the user. These findings suggest that selectivity, and thereby conciseness, are effective in tailoring an explanation to a user in an XR setting.

User trust was reported to be elevated more by explanations for generation than for placement (Figure 6.6b). This, along with user comments about limitations in the placement mechanism, suggests that explanations cannot raise user trust on their own but need to be accompanied by the decision itself. In other words, trust in the system is more dependent on the perceived functionality, and the impact of explanations on this trust comes second. Nevertheless, the system reliability was slightly improved by the explanations 6.6b, and most users prefer the system with explanations over one without 6.7a. This indicates that explanations may elevate user trust in XR instantiation systems, although they are best when accompanied by satisfactory results.

Chosen explanations by the users varied both by feature (Figures 6.9a and 6.9b) and by fidelity (Figures 6.8a and 6.8b). Nevertheless, different trends can be observed in generation and placement. For explanations of object generation, user choices more often aligned with higher fidelity scores, while they more often aligned with specific features for explanations of placement. The reason behind this may be that the limitations in placement obscure the intent of the system, and therefore also the fidelity of the explanations. It may also be the cause of a preference for height and scene features when reasoning about space, a theme which was present in user comments (Table 6.4). In essence, user choice of explanation may be affected by our fidelity improvement, but also be affected by personal feature preferences.

7.4 Limitations and Future Work

The following presents the limitations and future work of our thesis. This includes both the solution design and implementation, as well as the evaluation of the solution.

7.4.1 3D Object Generation

In this module, the generation pipeline and the characteristics of generated objects present the most notable limitations.

Generation Pipeline. Even though the 3D object generation in general was reliable and produced realistic objects, some limitations of the pipeline appeared during evaluation. One aspect that the users raised was the unrealistic generation of a framed painting (see Section 6.2.1). This issue arose from the generation module, which did not accurately predict the correct shape of the intended object, turning it into a cube. This can be seen as a miscommunication between the image generation model and TRELLIS, as the image model generated a painting facing straight forward, which TRELLIS then interpreted as it being a box instead. To avoid these limitations in the future, a more integrated 3D

object generation pipeline that takes into account the limitations of both the image and 3D generation model is needed.

Instantiation Latency. The total time to instantiate an object with our solution was between 1-2 minutes. This is not feasible in a real-world application for most use cases, and was something that was pointed out by users (Figure 6.7b). In order to shorten the instantiation time, more efficient models for all tasks involved need to be researched. This includes 3D object generation models, but also image generation models and VLMs.

Characteristics of 3D Objects. Another limitation noted with TRELLIS was that it struggled with creating fine, detailed objects, with these objects sometimes becoming a bit patchy, which can be seen in Figure 4.1e. To mitigate this, further research to develop a generation model that can deliver precise, detailed objects, while keeping, or even improving, the latency is needed. TRELLIS is also only able to create the shape and texture of 3D objects, but does not output other characteristics of the objects like lighting, mass or rigidity. Since this thesis focuses on characteristics needed for a context-aware placement, these characteristics were not taken into account. An LLM could potentially be able to provide some of these features from semantic knowledge alone, while, for example, the lighting would have to consider the physical lights in the room in some way. In brief, to enhance the interaction with 3D objects in XR, further research and application will be needed.

7.4.2 3D Object Placement

While the placement module handled many scenarios adequately, limitations were identified in both the object detection component and the accuracy of the resulting placements.

Accuracy of Placement. During the evaluation, the placement of the object was shown to lack accuracy at times. This could be due to the fact that the VLM spatial awareness, although aided by Cube-RCNN, is still lacking when outputting 3D coordinates. For the current, VLM-driven approach of the design, the error in accuracy in placement is hard to fully eliminate. A discriminator pattern for the system might be able to improve on the placement, using a closed-loop to feed images back to the VLM, enabling it to fine-tune the placement until it is satisfied. This, however, would be costly in time and make the user experience less smooth. Supporting this, Google released a paper [5] based on a similar concept, called intrinsic self-critique, where the LLM’s performance was improved on a number of data sets. This kind of solution could enhance the context awareness, as well as the placement, of the system.

Scene understanding. Another concern is that the room in question might not contain any objects whatsoever. This creates a hard situation for the LLM, which can no longer guess which distances apply. One solution to this could be to apply a depth map to the input, to ensure that the model can use this as a fallback for measurements. Another approach could be to use open-vocabulary models instead, to elevate the context absorbed from the environment [51]. Other ways to refine the placement include adding an occlusion functionality to make the objects look more realistically placed, adding manual adjustment for the user to fine-tune placement and/or adding prior virtual objects to the context to avoid collisions.

7.4.3 Explainability

While the explanation module addressed fidelity as a core characteristic, limitations remain in the scope and system-level coverage of the explanations produced.

Characteristics of Explanations. Since fidelity is only one aspect of an explanation, there are naturally others that could have been the focus of our explanations. The basis for the choice of fidelity was that our goal was to ensure that an end user received knowledge about the system’s decisions that could elevate the trust and perceived safety. While completeness and plausibility are important for the user to grasp the contents of the explanations, we found it more important that the explanations did their best to convey the process. Within the field of self-explanations, we also did not want to reward including more aspects than necessary into the explanations to improve completeness or complexity, since this could lead to hallucinations. However, in the future, it would be interesting to explore other characteristics needed to create as good an explanation as possible.

System-Level Explanations. The system focuses on the LLMs explaining their text-based decision making, but leaves the rest of the system opaque. The system does not provide an explanation to the image models or TRELLIS generation or CubeRCNN identification of boundary boxes. Additionally, it can be argued that the fidelity score for both generation and placement, during the Feature Attribution loop, is limited and does not fully encapsulate the decision from our system. For full fidelity, the entire decision, including the final state of the object and position, is needed. Since our explanation module is not aware of this final state of the object in the scene when creating explanations, there is a natural discrepancy between the result and the explanation. For true end-to-end explainability, all parts of the system and their correlations need to be explained.

7.4.4 Performance Evaluation

The performance evaluation revealed several limitations relating to the evaluation of the generation pipeline, similarity score, data diversity, and sample size, each of which affects the robustness of the results.

Limitations in Evaluating Generation. The generation consists of a pipeline, which includes first generating a prompt, then an image and at last a GLB-file of the 3D object. When evaluating the generation, only the image generated by the image model is evaluated, and for a more representative result, every step of the pipeline should be evaluated. An interesting finding was that generation for a specific type of object may have large internal variation (see Table 6.1). For the knife, we can see a variation across all features, indicating that there may be many ways to design a knife adapted to the same user data and environment. This can also be seen in Table 6.1, where the reference knife and the baseline differ qualitatively in design and shape. The reason for this variation may be the inherent randomness of generative AI, which may occasionally overshadow the variation caused by perturbation shifts. In general, this variation makes it harder to determine the context-awareness in generation in an exact way. This needs to be mitigated in future work to obtain a better understanding of the internal variations of the LLMs.

Limitations of the Similarity Score. Discoveries from the quantitative evaluation indicate that the calculations used to evaluate the generation may not completely reflect the behavioral shifts in the generated output. Table 6.1 displays that there are some correla-

tions for how the objects are altered across prompts. For example, when changing the style from antique to modern, the objects acquire cleaner lines and colors. This is displayed in the scores across prompts, such as "A trash can" and "A wall-mounted shelf", but not shown in the "A bike" prompt. In the bike example, age is seen, in Figure 6.1, as having an effect on the output, while style barely has any, even though the human eye can see that there is a stylistic change of the bike. This indicates that the embeddings used are sensitive to changes in certain aspects of an image, for example, big shapes and lines, while less sensitive to color changes and smaller details. Hence, to obtain a more fair evaluation the similarity score needs to be more nuanced. A more inclusive similarity score for the generation will create more nuanced and fair results for how the generation is affected by perturbed features.

Limitations of Evaluation Data Diversity. During the quantitative analysis, the baseline features were chosen to represent an arbitrary user of the system. From this, the height was perturbed by 40 cm to obtain a significant difference while remaining realistic. The age was chosen through similar reasoning, as well as with the reasoning to see if there was a difference in how the system treats an adult user compared to a child. The modern style was chosen to represent a clear stylistic change. The selection of a new environmental context was intended to introduce a different layout with new objects and a style. However, due to specific network requirements, data capture was restricted to the company's office, which limited the variety of available environments. Moreover, the user study is built upon users with relatively similar backgrounds. By expanding the user study, through a variety of diverse users with different features, a more comprehensive and detailed result can be obtained.

Limitations in Sensitivity Analysis Sample Size. The sample size in the sensitivity analysis was limited due to scope and computational cost. More specifically, the perturbations of the features were limited to one perturbation for each feature and were iterated over five times for each prompt. This resulted in 25 data points per prompt and a total of 100 data points for each module. This relatively small sample size poses a challenge for identifying robust statistical trends, as the inherent variance of the models may overshadow the subtle effects of individual perturbations. To mediate this, it would be helpful to create more nuanced examples for this prompt, potentially merging more features from the scene with personal ones in creative ways. Consequently, while the data provides preliminary insights, a larger dataset and budget would be required to confirm these observations with greater certainty.

7.4.5 System Ethics and Practical Concerns

This section addresses ethical and practical considerations related to the system design, including context-awareness limitations, energy and economic costs, and efficiency.

Context-awareness, Privacy, and Bias Concerns. The validity of this work is affected by the definition of the term *context-aware*. In our system, context-aware means that we use the user and scene information to guide the generation and placement in that direction. In this sense, context-awareness and personalization are almost synonymous. This may or may not prove a frustration for users who do not wish to personalize items to their own style or home, but instead discover new ideas and experiences. Due to this, the user may not feel that the system meets their expectations. On the other hand, some users may not

feel that the current solutions take into account all of their most important features, leaving them unsatisfied because of this. To further evaluate the system’s context-awareness and enhance user-centric customization, additional features beyond age, height, and style should be implemented. Examples of these features could be interests, sex, and other biometric measures. However, by providing the system with user-specific features, bias can be introduced, whose occurrences and effects would be of value to study further.

Energy and Economic Concerns. The frequent use of OpenAI models in the explanation improvement, as well as the object instantiation, amounts to both economic and energy concerns. Because of the availability of remotely hosted LLMs and the budget allowed, no local LLMs were used. Even if local LLMs were used, one could argue that it does not take an LLM to produce a contrary hint, for example. This functionality would have to be replaced with a smaller, specialized model, which would have to be trained for the purpose of reversing the information inside some text. For placing an object, a smaller specialized VLM could also be used, which might be similar to MM-spatial [11]. The closed nature of this model, along with the limited ability to implement it ourselves, guided the choice to proprietary choices. However, to save energy and money, it would be worth exploring other smaller task-specific models.

Caching and Efficiency Concerns. In order to save time and energy, common objects for a user could be cached in a database, so that they could be retrieved faster in the future. While this would save energy, the total amount of time might not be affected due to the optimization of explanations being dependent on the scene context. Because of this, all objects are generated on a per-request basis in order to keep the scope of the thesis small.

Chapter 8

Conclusions

In this thesis, we presented a 3D object instantiation solution in XR, with a focus on context-awareness and explainability. Our solution fulfils showcases context-awareness and explainability, improving user trust and reliability.

Answering RQ1, we found several limitations in existing 3D object instantiation solutions for XR. User context for instantiating objects is seldom considered in existing solutions, even though it can be used to tailor the experience to the user. Generation quality is often also rejected for fast instantiation, and placement is done in restrictive ways. Explainable AI is also not present in any of the existing solutions, creating uncertainties for large, advanced systems. These limitations led us to explore tools which could enable context-awareness and explainability, and set requirements for our own solution.

Answering RQ2, we followed our created requirements from RQ1, and designed and implemented a context-aware and explainable 3D object instantiation system in XR. The system considers user age, height, and personal style, as well as the surrounding scene, and compiles the context using a VLM. Using both 3D object generation and 3D object detection, our system is able to generate an object from the user’s text prompt and place it in the setting, following constraints. Additionally, our design utilizes explainability, focusing on selectivity and fidelity.

Answering RQ3, the evaluation of our system highlights the most influential context sources for our system, as well as its strengths and limitations. Age and style impacts the generation of the objects, while the scene is most influential for the placement of the objects. Users report that the generation is more context-aware than the placement, due to physical constraints being broken. Explanations improve user trust in generation, but not for placement, likely due to the discrepancy between explanations and the end result. The faithfulness of explanations seem to have a small impact on user preference for generation, while other factors are more important for placement. To create the best explanation for an end user, system accuracy and full system inclusion in the explanation need to be considered along with fidelity and selectivity. Overall, the system successfully addresses the generation and explainability limitations identified in RQ1, while placement remains

partially resolved due to the inherent difficulty of precise 3D spatial reasoning for VLMs.

This thesis makes three primary contributions to the field of XR instantiation. First, we demonstrate high-quality, context-aware 3D object generation that considers both environmental and user-specific factors, an approach which is absent in existing solutions. Second, we contribute a novel placement method using free 3D coordinates generated by a VLM, guided by object detection and a 2D image, removing the dependency on environment-specific surface scanning or ray-casting. Third, we introduce explainability to XR instantiation systems for the first time, through a self-explanation module with fidelity-improved selectivity, demonstrating that concise, faithful explanations can improve user trust in AI-driven XR environments.

Taken together, our findings suggest that the usage of generative AI is a powerful tool in XR instantiation systems, especially when valuing context-aware decisions. Additionally, explainability is important in maintaining user trust, although it cannot redeem larger user dissatisfaction. As it stands, current generative AI models still face limitations in both 3D generation quality and latency, as well as spatial placement precision, indicating a broader challenge for VLM-based systems. In other words, semantic reasoning does not yet reliably translate into accurate instantiation with our solution design. Addressing these limitations will be key to realising the full potential of context-aware, explainable XR instantiation.

The context-awareness and explainability of our system make it applicable across a range of domains. The sensitivity of generation to user age and style makes the system particularly suited to personalised interior design, where tailoring objects to individual preferences is central. In education, the ability to generate contextually appropriate objects on demand could support immersive, safe, and adaptive learning environments. For game development, the combination of high-quality generation and scene-aware placement offers a foundation for dynamic, user-driven world building. More broadly, the insight that explainability improves trust in AI-driven decisions has implications beyond XR — for any system where generative AI makes consequential choices on behalf of a user.

Future work should aim to improve the placement capabilities of VLMs in 3D space, while retaining the freedom of dynamic, personalized placement. More energy-efficient ways of implementing both instantiation and explanations should also be researched, including other explainability techniques than self-explanations. Furthermore, the solution should be tested on a wide selection of users, including large differences in personal features, in order to determine its full capabilities.

References

- [1] Jesús Balado, Yu Feng, Zhouyan Qiu, W. Gao, and Arttu Julin. 3d as-built environments in extended reality applications: A systematic review. *Photogrammetric Record*, 41(194), 2026.
- [2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Ben-netot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58:82–115, 2020.
- [3] Majid Behravan and Denis Gračanin. From voices to worlds: Developing an ai-powered framework for 3d object generation in augmented reality. In *2025 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 150–155, 2025.
- [4] Blender Foundation. Blender features. <https://www.blender.org/features/>. Accessed: 2026-05-16.
- [5] Bernd Bohnet, Pierre-Alexandre Kamienny, Hanie Sedghi, Dilan Gorur, Pranj-al Awasthi, Aaron Parisi, Kevin Swersky, Rosanne Liu, Azade Nova, and Noah Fiedel. Enhancing llm planning capabilities through intrinsic self-critique. *arXiv:2512.24103*, 2025.
- [6] Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. Omni3d: A large benchmark and model for 3d object detection in the wild. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13154–13164, 2023.
- [7] Iolanda L. Chamusca, Tiago P. Pagano, Rafael B. Loureiro, Erick G. S. Nascimento, Alex Á. B. Santos, Yiyu Cai, Thiago B. Murari, and Ingrid Winkler. Enhancing context awareness in augmented reality environments: A review of opportunities and challenges in object and action recognition through machine learning. *Discover Applied Sciences*, 7(7):750, 2025.

- [8] Jiangong Chen, Xiaoyi Wu, Tian Lan, and Bin Li. Llmer: Crafting interactive extended reality worlds with json data generated by large language models. *IEEE Transactions on Visualization and Computer Graphics*, 31(5):2715–2724, May 2025.
- [9] Yu-Neng Chuang, Guanchu Wang, Chia-Yuan Chang, Ruixiang Tang, Shaochen Zhong, Fan Yang, Mengnan Du, Xuanning Cai, Vladimir Braverman, and Xia Hu. Faithlm: Towards faithful explanations for large language models. *arXiv:2402.04678*, 2025.
- [10] Shakiba Davari and Doug A. Bowman. Towards context-aware adaptation in extended reality: A design space for xr interfaces and an adaptive placement strategy. *arXiv:2411.02607*, 2024.
- [11] Erik Daxberger, Nina Wenzel, David Griffiths, Haiming Gang, Justin Lazarow, Gefen Kohavi, Kai Kang, Marcin Eichner, Yinfei Yang, Afshin Dehghan, and Peter Grasch. Mm-spatial: Exploring 3d spatial understanding in multimodal llms. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7395–7408, 2025.
- [12] Anind K. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, 5(1):4–7, 2001.
- [13] Stefan Feuerriegel, Jochen Hartmann, Christian Janiesch, and Patrick Zschech. Generative ai. *Business & Information Systems Engineering*, 66(1):111–126, September 2023.
- [14] Tham Yik Foong, Shashank Kotyan, Po Yuan Mao, and Danilo Vasconcellos Vargas. The challenges of image generation models in generating multi-component images. *arXiv:2311.13620*, 2023.
- [15] Team Hunyuan3D, Shuhui Yang, Mingxin Yang, Yifei Feng, Xin Huang, Sheng Zhang, Zebin He, Di Luo, Haolin Liu, Yunfei Zhao, Qingxiang Lin, Zeqiang Lai, Xianghui Yang, Huiwen Shi, Zibo Zhao, Bowen Zhang, Hongyu Yan, Lifu Wang, Sicong Liu, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Dongyuan Guo, Junlin Yu, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Shida Wei, Chao Zhang, Yonghao Tan, Yifu Sun, Lin Niu, Shirui Huang, Bojian Zheng, Shu Liu, Shilin Chen, Xiang Yuan, Xiaofeng Yang, Kai Liu, Jianchen Zhu, Peng Chen, Tian Liu, Di Wang, Yuhong Liu, Linus, Jie Jiang, Jingwei Huang, and Chunchao Guo. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. *arXiv:2506.15442*, 2025.
- [16] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv:2305.02463*, 2023.
- [17] Jaewook Lee, Filippo Aleotti, Diego Mazala, Guillermo Garcia-Hernando, Sara Vicente, Oliver James Johnston, Isabel Kraus-Liang, Jakub Powierza, Donghoon Shin, Jon E. Froehlich, Gabriel Brostow, and Jessica Van Brummelen. Imaginear: Ai-assisted in-situ authoring in augmented reality. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*, UIST ’25, page 1–21. ACM, September 2025.

-
- [18] John D. Lee and Katrina A. See. Trust in automation: Designing for appropriate reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 46(1):50–80, 2004.
- [19] Sangyoon Lee, Jinseok Seo, Gerard Kim, and Chan-Mo Park. Evaluation of pointing techniques for ray casting selection in virtual environments. *Proceedings of SPIE - The International Society for Optical Engineering*, 4756, 04 2003.
- [20] Xiaoyu Li, Qi Zhang, Di Kang, Weihao Cheng, Yiming Gao, Jingbo Zhang, Zhihao Liang, Jing Liao, Yan-Pei Cao, and Ying Shan. Advances in 3D Generation: A Survey. *arXiv:2401.17807*, 2024.
- [21] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, pages 38–55, Cham, 2025. Springer Nature Switzerland.
- [22] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7076–7086, 2022.
- [23] Xianzheng Ma, Brandon Smart, Yash Bhalgat, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, Philip H Torr, Marc Pollefeys, Matthias Nießner, Ian D Reid, Angel X. Chang, Iro Laina, and Victor Adrian Prisacariu. When llms step into the 3d world: A survey and meta-analysis of 3d tasks via multi-modal large language models. *arXiv:2405.10255*, 2025.
- [24] Clara Maathuis, Marina Anca Cidota, Dragos Datcu, and Letiția Marin. Integrating explainable artificial intelligence in extended reality environments: A systematic survey. *Mathematics*, 13:290, 01 2025.
- [25] Meshy AI. Meshy: 3d ai generator. <https://www.meshy.ai/>. Accessed: 2026-05-16.
- [26] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, vol. E77-D, no. 12:1321–1329, 12 1994.
- [27] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.
- [28] Xinyu Ning, Yan Zhuo, Xian Wang, Chan-In Devin Sio, and Lik-Hang Lee. When generative artificial intelligence meets extended reality: A systematic review. *International Journal of Human–Computer Interaction*, 42(10):6870–6890, 2026.
- [29] Arif Nurjanah and Endah Retnowati. Augmented reality in the perspective of cognitive load theory. In *FRONTIERS IN INDUSTRIAL AND APPLIED MATHEMATICS*, page 090002, 01 2024.
-

- [30] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *arXiv:2304.07193*, 2024.
- [31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*, 2021.
- [32] Philipp Rauschnabel, Reto Felix, Chris Hinsch, Hamza Shahab, and Florian Alt. What is xr? towards a framework for augmented and virtual reality. *Computers in Human Behavior*, 133:107289, 08 2022.
- [33] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv:1908.10084*, 2019.
- [34] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ”why should i trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [35] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2022.
- [36] Mohsen Rostami, Pratik Pradhan, Niraj Karki, Jeffery Omorodion, Parichehr Milani, Jafer Kamoopuri, Han Yu Liu, and Joon Chung. A comprehensive review of extended reality and its application in aerospace engineering. *Progress in Aerospace Sciences*, 157:101118, 2025.
- [37] Per Runeson, Emelie Engström, and Margaret-Anne Storey. *The Design Science Paradigm as a Frame for Empirical Software Engineering*, pages 127–147. Springer International Publishing, Cham, 2020.
- [38] Jalal Safari Bazargani, Abolghasem Sadeghi-Niaraki, and Soo-Mi Choi. From 2d to xr: enhancing object placement in design simulations through spatial relations. *Virtual Reality*, 30(1):76, feb 2026.
- [39] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv:2402.07927*, 2025.
- [40] Johannes Schneider. Explainable generative ai (genxai): a survey, conceptualization, and research agenda. *Artificial Intelligence Review*, 57(11):289, Sep 2024.

-
- [41] Aditya Sharma, Luke Yoffe, and Tobias Höllerer. Octo+: A suite for automatic open-vocabulary object placement in mixed reality. In *2024 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pages 157–165, 2024.
- [42] Mel Slater. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 364:3549–57, 12 2009.
- [43] Sule Tekkesinoglu. Exploring evaluation methodologies for explainable ai: Guidelines for objective and subjective assessment. *SSRN Electronic Journal*, 2024.
- [44] Yuqi Tong, Yue Qiu, Ruiyang Li, Shi Qiu, and Pheng-Ann Heng. Ms2mesh-xr: Multi-modal sketch-to-mesh generation in xr environments. In *2025 IEEE International Conference on Artificial Intelligence and eXtended and Virtual Reality (AIxVR)*, pages 272–276, 2025.
- [45] Tripo AI. Tripo 3d studio. <https://studio.tripo3d.ai/>. Accessed: 2026-05-16.
- [46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [47] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *arXiv:2512.14692*, 2025.
- [48] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21469–21480, 2025.
- [49] Yunhao Xing, Que Liu, Jingwu Wang, and Diego Gómez-Zarà. smore: Spatial mapping and object rendering environment. In *Companion Proceedings of the 30th International Conference on Intelligent User Interfaces, IUI '25 Companion*, page 115–119, New York, NY, USA, 2025. Association for Computing Machinery.
- [50] Yanming Xiu, Joshua Chilukuri, Shunav Sen, and Maria Gorlatova. Say it, see it: A systematic evaluation on speech-based 3d content generation methods in augmented reality. In *2025 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 320–326, 2025.
- [51] Jin Yao, Hao Gu, Xuweiyi Chen, Jiayun Wang, and Zezhou Cheng. Open vocabulary monocular 3d object detection. In *2026 International Conference on 3D Vision (3DV)*, pages 1801–1811, 2026.
- [52] Luke Yoffe, Aditya Sharma, and Tobias Höllerer. Octopus: Open-vocabulary content tracking and object placement using semantic understanding in mixed reality. In *Proceedings of the 2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 587–588, 2023.
-

- [53] Rufai Yusuf Zakari, Jim Wilson Owusu, Ke Qin, Hailin Wang, Zaharaddeen Karami Lawal, and Tao He. Vqa and visual reasoning: An overview of approaches, datasets, and future direction. *Neurocomputing*, 622:129345, 2025.
- [54] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Yiwen Hu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *Frontiers of Computer Science*, 20(12):2012627, May 2026.

Appendices

Appendix A

Prompts

Prompt A.1: System prompt for generation

Role: You are an expert Spatial AI Coordinator and 3D Scene Designer.

Your task is to safely and logically instantiate a new requested object into a given 3D environment based on visual semantics, spatial data, and user context.

Inputs You Will Receive:

Scene Image: An image capturing the semantic environment.

User Prompt: The specific object the user wants to instantiate.

User Data: Additional context, constraints, or preferences from the user. This may be empty.

Task Instructions: You must process the inputs and output your final result as a strict JSON object following this exact order:

Generate Prompt (GEN): Create a highly contextual, context-aware prompt that will be passed to a generative model to create the requested object.

Ensure this explicitly considers environmental style from both the user and the scene, and safety constraints. Describe only the object that is to be generated, don't mention the surrounding environment here or any information about the placement of the object. Keep it

short.

Explain Generation Decisions (EXPL_GEN): Create a concise and insightful explanation justifying your given GEN answer, avoid it being too technical.
Make use of contrastive explanation, such as "A was chosen instead of B, since B had this weakness".
Thoroughly analyze the relevant given information to explain why this prompt was chosen. Include information about both user style, height and age, as well as the environment from the image. Reason about how these context sources contributed to the output. Avoid introducing outside information or opinions.

Output Format: You must output strictly in valid JSON format.
Do not use markdown formatting blocks (like ```json), or explanations outside of the JSON object itself.
Use the exact schema below:
{ "GEN": "String. The context-aware prompt to generate the object.", "EXPL_GEN": "String. Your concise explanation justifying the generation prompt." }
Example Interaction 1:
Scene Image: <Few-shot Image>
User Prompt: A shelf
User Data: User height: 160 cm. Age: 12. Style: colorful
Output: { "GEN": "A colorful, vibrant shelf with multiple planes and wooden details.",
"EXPL_GEN": "The generation of the object considers the users colorful style, as well as the environment details with wooden components. The object did not include any potentially harmful or sensitive details due to the young age." }

Prompt A.2: System prompt for placement

Role: You are an expert Spatial AI Coordinator and 3D Scene Designer.
Your task is to safely and logically instantiate a new requested object into a given 3D environment based on visual semantics, spatial data, and user context.

Inputs You Will Receive:
Scene Image: An image capturing the semantic environment.
3D Boxes: A list of existing objects in camera coordinates, formatted as [x, y, z, l, h, w] in meters (x = right, y = down, z = forward, l = length, h = height, w =

width).

User Prompt: The specific object the user wants to instantiate.

User Data: Additional context, constraints, or preferences from the user. This may be empty.

Task Instructions: You must process the inputs and output your final result as a strict JSON object following this exact order:

Determine Placement (PLACE): Output the precise ideal location and scaling for the new object as a 4-element numeric array: [x, y, z, s]. x, y, z are the same as the 3D boxes, and s represents the scaling factor with which to scale the largest side of the object. s=1 means that the largest dimension in the resulting object will be 1 meter.

Coordinates are in Camera Space, which means that z is pointing toward the center of the image, not necessarily along the floor. y being local means that a positive y results in a placement in the top half of the 2D image. y being negative means that the placement is in the bottom half of the 2D image. y being near 0 means that the placement will be along horizontal middle line of the 2D image.

You need to thoroughly analyze the depth of the given boxes, and make sure that your final z-coordinate makes sense in relation to the image.

Reason about all the coordinates, and make sure that the numbers align with your reasoning.

The placement must not block pathways or affordances, must align with the <user_data>, and must adhere to physical safety.

Explain Placement Decisions (EXPL_PLACE): Create a concise and insightful explanation justifying your given PLACE answer, avoid it being too technical.

Thoroughly analyze the relevant spatial bounding boxes and visual semantics, as well as the user height and age, to explain why these coordinates and size were chosen.

Do not include the specific coordinates in the answer. Make use of contrastive explanation, such as "A was chosen instead of B, since B had this weakness". Include information about both user height, style and age, as well as the environment from the image and boxes. Reason about how these context sources contributed to the output.

Avoid introducing outside information or opinions.

Output Format: You must output strictly in valid JSON format.

Do not use markdown formatting blocks (like ```json), or explanations outside of the JSON object itself.

Use the exact schema below:

```
{ "PLACE": [x, y, z, s], "EXPL_PLACE": "String. Your concise explanation justifying the placement." }
```

Example Interaction 1:

Scene Image: <Few-shot image>

3D Boxes:

```
[1.11, -0.18, 2.31, 0.49, 0.82, 0.45]
(chair)

[0.08, -0.17, 2.52, 0.58, 0.95, 0.50]
(chair)

[0.35, 0.51, 4.42, 1.29, 1.33, 0.27]
(shelves)

[1.54, -0.23, 2.41, 1.15, 0.76, 0.79]
(table)

[0.36, -0.27, 4.59, 1.22, 0.70, 0.52]
(cabinet)
```

User Prompt>: A shelf

User Data: Height: 160 cm. Age: 12. Style: colorful

Output: { "PLACE": [-2.65, 0, 2.40, 0.40],
"EXPL_PLACE": "The coordinate placement maps the shelf to the left side of the image, on the camera-facing wall and places it at a height appropriate for you to reach. The shelf should be around as far away as the chair, but a tiny bit closer to the camera. The size matches the usual proportions of a multiple story shelf. The position and size was chosen not to block the existing shelf in the middle, and not on the floor since it could block pathways" }

Prompt A.3: Feature identification, generation

System: You are an XAI Evaluation Assistant. Your goal is to ensure every user feature and context attribute is tested and explained. You are not allowed to talk about how the features affect placement or scaling of the object. Only the appearance (color, shape, etc

).

Task:

Identify the existing Target Features: Look at the explanations in the 'Current Trajectory Data'. For each explanation there is a score and a contrary hint. The contrary hint contradicts a certain feature (e.g., Something in the scene, or some user data). Find the specific features (e.g., "User Style") from the contrary hints which have already been tested, and see from the fidelity score how important that feature was.

Find a new Target Feature: Find a target feature that has not yet been tested and given a score (e.g, "User Age"). The new target feature needs to be a part of one of the explanations, and can not be made up from nothing.

Generate a new Explanation: Use the target feature to explain the decision from the model.

Constraint: Do not mention "scores," "trajectories," or "XAI." Write a 1-2 sentence natural justification. Maximum 50 words. You may not include more than one feature in one explanation (only height, age, style or environment from the image)

Output:

Provide ONLY the string explanation centered on that one specific feature.

Inputs:

LLM Output Result: '{Y}'

LLM Self Explanation: '{init_expl}'

Current Trajectory Data: '{trajectory_list}'

Prompt A.4: Feature identification, placement

System: You are an XAI Evaluation Assistant. Your goal is to ensure every user feature and context attribute is tested and explained. You are not allowed to talk about how the features affect appearance (color, shape, etc). Only placement or scaling of the object.

Task:

Identify the existing Target Features: Look at the explanations in the 'Current Trajectory Data'. For each explanation there is a score and a contrary hint. The contrary hint contradicts a certain feature (e.g., Something in the scene, or some user data). Find the specific features (e.g., "User Style") from the contrary hints which have already been tested, and see from the fidelity score how important that feature was.

Find a new Target Feature: Find a target feature that has not yet been tested and given a score (e.g, "User Age"). The new target feature needs to be a part of one of the explanations, and can not be made up from nothing.

Generate a new Explanation: Use the target feature to explain the decision from the model.

Constraint: Do not mention "scores," "trajectories," or "XAI." Write a 1-2 sentence natural justification. Maximum 50 words. You may not include more than one feature in one explanation (only height, age, style or environment from the image)

Output:

Provide ONLY the string explanation centered on that one specific feature.

Inputs:

LLM Output Result: '{Y}'

LLM Self Explanation: '{init_expl}'

Current Trajectory Data: '{trajectory_list}'

Prompt A.5: Creation of contrary hint, generation

Instruction:

Analyze the provided explanation to extract one Core Fact regarding the object's Generation (how it looks)).

Once extracted, invert that fact to create a "Contrary Hint-"an instruction that describes the exact opposite of the intended setup.

Constraints:

Logical Inversion: If the fact says "colorful," the hint must be "grayscale and dull" If the fact says "cute," the hint must be "scary/intimidating ."

Output Format: You must return the result in this exact format:

Your inverted sentence

Example 1:

The Explanation: Matte black framing was chosen to match the 'rooms tech-focused, neutral - paletteparticularly the black -screensso the shelf visually integrates with existing electronics and minimal finishes.

Core Fact: The colors in the room, particularly the TVs, contributed to the decision of a Matte black framing.

RETURNS:

Contrary Hint: The design should be glossy and bright, since the colors in the room are that way .

NOW DO THIS ONE:

Explanation: {explanation}

Prompt A.6: Creation of contrary hint, placement

Instruction:

Analyze the provided explanation to extract one Core Fact regarding the object's Generation (how it looks)).

Once extracted, invert that fact to create a "Contrary Hint"-an instruction that describes the exact opposite of the intended placement or scaling.

Constraints:

Logical Inversion: If the chosen fact is "small," the hint must be "excessively large." If the fact says "centered," the hint must be "offset/obstructive."

Output Format: You must return the result in this exact format:

Your inverted sentence

Example 1:

The Explanation: The shelf is mounted at a height reachable by a 180 cm adult, minimizing bending or stretching for daily use and keeping commonly used items at comfortable standing reach.

Core Fact: The user height being 180 cm contributed to the shelf mount position at a normal height for an adult.

RETURNS:

Contrary Hint: The user height is 100 cm, and the shelf should be mounted at that height to suit their height.

NOW DO THIS ONE:

Explanation: {explanation}

Appendix B

User Study Questions

Question	1	2	3	4	5
1. The visual appearance of the objects reflected my personal style.	☐	☐	☐	☐	☐
2. The visual appearance of the objects felt tailored to my height.	☐	☐	☐	☐	☐
3. The visual appearance of the objects felt tailored to my age.	☐	☐	☐	☐	☐
4. The visual appearance of the objects matched the style of the surrounding physical objects.	☐	☐	☐	☐	☐
5. The visual appearance of the objects suited the colors of the environment.	☐	☐	☐	☐	☐
6. The visual appearance of the objects suited the materials of the environment.	☐	☐	☐	☐	☐
7. The placement of the objects felt tailored to my style.	☐	☐	☐	☐	☐
8. The size of the objects felt tailored to my style.	☐	☐	☐	☐	☐
9. The placement of the objects felt tailored to my height.	☐	☐	☐	☐	☐
10. The size of the objects felt tailored to my height.	☐	☐	☐	☐	☐
11. The placement of the objects felt suitable to my age.	☐	☐	☐	☐	☐
12. The size of the objects felt suitable to my age.	☐	☐	☐	☐	☐

- | | | | | | |
|--|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| 13. The placement of the objects respected the physical space of the environment. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 14. The size of the objects was appropriate relative to the surrounding environment. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 15. The placement of the objects was appropriate for the context of the environment. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 16. The explanation was clear and easy to follow. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 17. The explanation made it clear how my personal characteristics influenced the result. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 18. The explanation made it clear how the environment influenced the result. | ☐ | ☐ | ☐ | ☐ | ☐ |
| *19. The length of the explanations felt | ☐ | ☐ | ☐ | ☐ | ☐ |
| 20. The explanations increased my trust in the system's decisions about object generation. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 21. The explanations increased my trust in the system's decisions about object placement. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 22. The explanation made the system's behavior feel more reliable. | ☐ | ☐ | ☐ | ☐ | ☐ |
| 23. I would prefer this system over one that does not provide explanations. | ☐ | ☐ | ☐ | ☐ | ☐ |
| *24. The time required to instantiate the object felt | ☐ | ☐ | ☐ | ☐ | ☐ |
-

Table B.1: *Note: 1 = Totally Disagree, 5 = Totally Agree.*
** 1 = Very Short, 5 = Very Long*

1. If anything, what did you feel was missing from the explanations?
2. If you have other comments about the system, please write them below.

EXAMENSARBETE Context-aware and Explainable

Object Instantiation in XR

STUDENTER Martin Jernstedt, Julia Blomgren**HANDLEDARE** Sule Tekkesinoglu (LTH),

Saeed Bastani (Ericsson Research), Leire Amenabar (Ericsson Research)

EXAMINATOR Elizabeth Bjarnason (LTH)

Automatisk virtuell inredning, som dessutom förklaras!

POPULÄRVETENSKAPLIG SAMMANFATTNING av **Martin Jernstedt, Julia Blomgren**

Smarta glasögon blir allt vanligare och innehåller inte sällan funktioner för att piffa till din omvärld med virtuella objekt. Men vad händer egentligen innan en låtsas-stol placeras på ditt golv? Med hjälp av vårt nya system kan du få svar på mystiken bakom AI:s beslut, och hur just du och din omgivning togs i beaktning.

AI är på framfart och tekniken tar fler och fler beslut för vår räkning. Smidigt, tycker vissa. Läskigt, tycker andra. Farhågorna uppstår ofta ur mystiken bakom AI, eftersom man sällan får någon riktig förklaring för det som genereras. Användare rapporterar däremot ofta att de litar mer på AI-program när de får ta del av beslutsprocessen. Därför såg vi till att införa transparens när vi utvecklade ett avancerat AI-system för virtuell inredning.

Vi började med att söka igenom litteraturen efter existerande lösningar. Här hittade vi brister i bland annat kontext-medvetenhet och insikt från användarens håll. Vi designade sedan ett system som satte dessa egenskaper först, för att förbättra användarens upplevelse.

Systemet i sig är till för att placera virtuella objekt i ditt rum, som du kan se genom smarta glasögon. Bara säg vad du vill ha, så löser systemet resten! De smarta glasögonen skickar en förfrågan till en server, som sedan genererar, placerar och förklarar objektet med hjälp av olika AI-modeller. Vi utformade systemet för att ta hänsyn till just dina preferenser och förutsättningar, både i objektets utseende och placering. Detta gör att objekten genereras för att passa både din ålder och stil, samt sitter på en passande plats med en

lämplig storlek. När du ser objektet får du också en text som förklarar hur det hela gick till! Fiffigt va? Men hur kan du veta säkert att det du får läsa är sant?

För att se till att vårt system inte ljuger, eller "hallucinerar", testar vi ifall förklaringen faktiskt stämmer. Säger förklaringen att hyllan hamnade högt på väggen eftersom du var lång? Då testar vi vad som händer med en annan längd. Om hyllan flyttas, då stämmer ju faktiskt förklaringen! Om inte, så var det nog något annat som var mer viktigt. Kanske är det ett smalt utrymme som behöver förvaring långt upp för att inte vara i vägen? Den förklaring som är ärligast vinner och får visas som motivering för användaren.

Vi genomförde användartester för att utvärdera vår lösning. Resultaten från testerna visar att användare generellt föredrar förklaringar i denna typ av system. Tilliten till systemet ökar också med förklaringar, så länge objekten inte bryter verklighetsuppfattningen.

Vårt system har kapaciteten att göra utökad verklighet till en förståelig värld, även när många beslut tas automatiskt i bakgrunden. Vår lösning kan användas för att hjälpa till med heminredning, utbildning eller speldesign, där användaren vill ha insikt i systemet.