# ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality

Jaewook Lee*
University of Washington
Seattle, Washington, USA

Filippo Aleotti*
Niantic Spatial, Inc.
London, United Kingdom

Diego Mazala
Niantic Spatial, Inc.
London, United Kingdom

Guillermo Garcia-Hernando
Niantic Spatial, Inc.
London, United Kingdom

Sara Vicente
Niantic Spatial, Inc.
London, United Kingdom

Oliver James Johnston
Niantic Spatial, Inc.
London, United Kingdom

Isabel Kraus-Liang
Niantic Spatial, Inc.
London, United Kingdom

Jakub Powierza
Niantic Spatial, Inc.
London, United Kingdom

Donghoon Shin
University of Washington
Seattle, Washington, USA

Jon E. Froehlich
University of Washington
Seattle, Washington, USA

Gabriel Brostow
University College London
Niantic Spatial, Inc.
London, United Kingdom

Jessica Van Brummelen
Niantic Spatial, Inc.
London, United Kingdom

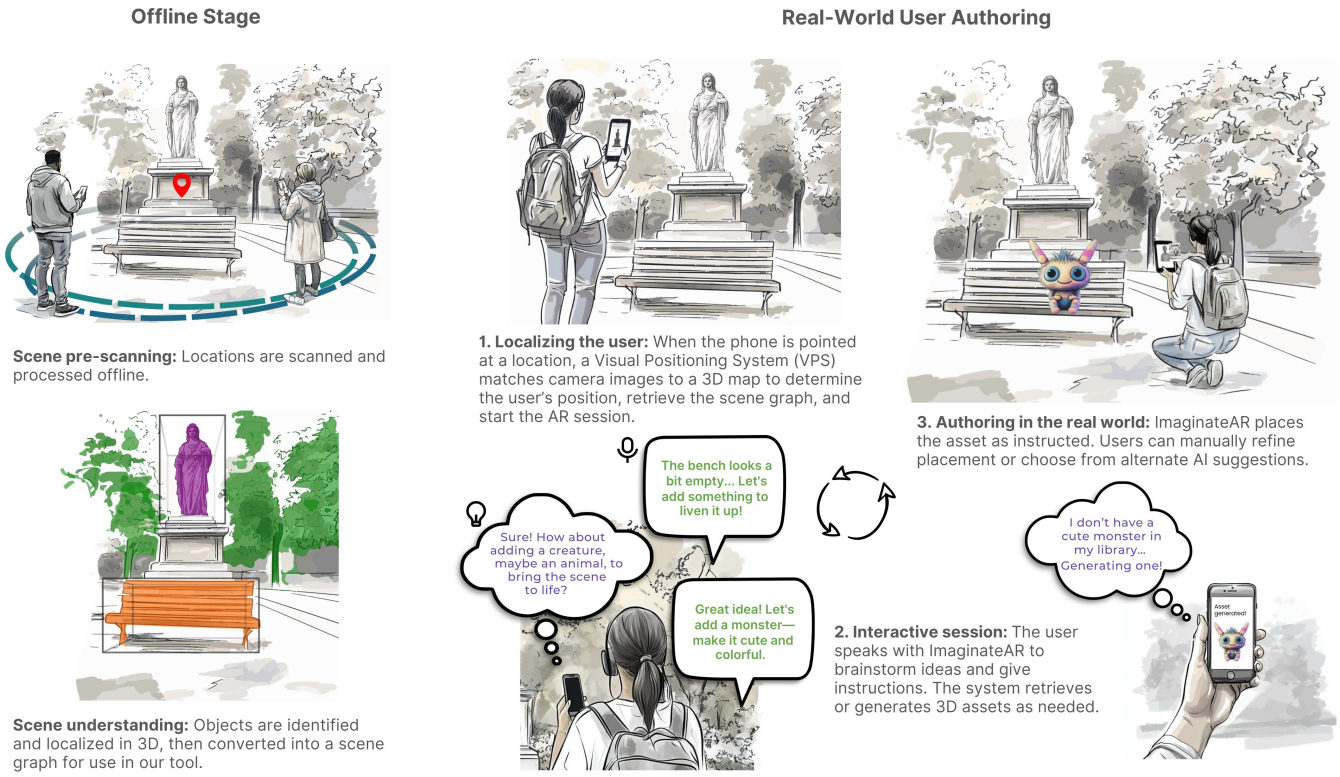



**Figure 1: ImaginateAR enables non-expert users to author personalized AR experiences through natural language interaction. After a location is pre-scanned and processed by our scene understanding pipeline (left), users can brainstorm, generate, and place virtual content on-site with AI assistance (right), and make manual adjustments as needed.**

*Both authors contributed equally to this research.


*UIST '25, September 28-October 1, 2025, Busan, Republic of Korea*

ACM ISBN 979-8-4007-2037-6/2025/09
https://doi.org/10.1145/3746059.3747635

## Abstract

While augmented reality (AR) enables new ways to play, tell stories, and explore ideas rooted in the physical world, authoring personalized AR content remains difficult for non-experts, often requiring professional tools and time. Prior systems have explored AI-driven XR design but typically rely on manually defined VR

environments and fixed asset libraries, limiting creative flexibility and real-world relevance. We introduce *ImaginateAR*, the first mobile tool for AI-assisted AR authoring to combine offline scene understanding, fast 3D asset generation, and LLMs—enabling users to create outdoor scenes through natural language interaction. For example, saying "*a dragon enjoying a campfire*" (P7) prompts the system to generate and arrange relevant assets, which can then be refined manually. Our technical evaluation shows that our custom pipelines produce more accurate outdoor scene graphs and generate 3D meshes faster than prior methods. A three-part user study (N=20) revealed preferred roles for AI, how users create in free-form use, and design implications for future AR authoring tools. ImaginateAR takes a step toward empowering anyone to create AR experiences anywhere—simply by speaking their imagination.

**ACM Reference Format:**
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. 2025. ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality. In *The 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25), September 28-October 1, 2025, Busan, Republic of Korea.* ACM, New York, NY, USA, 21 pages. https://doi.org/10.1145/3746059.3747635

## 1 Introduction

Augmented Reality (AR) can transform everyday spaces into interactive canvases, blending digital content with the physical world. Today, AR is used not just for entertainment, but also to bring people together through games like *Pokémon GO* [69], support location-based education [51], and amplify social causes through public art and storytelling [95]. Yet, most AR content is created using professional tools like *Unity* [105], *Blender* [11], and *Lens Studio* [96], requiring specialized skills and limiting *who* can create and *what* is possible. While this enables highly polished experiences, it leaves everyday users without a way to easily and creatively customize their surroundings with AR. Imagine if everyone had the power to create their own AR worlds—teachers could build interactive history lessons in a schoolyard, artists could install digital murals on city walls, and friends could fill a beach with dancing penguins.

Although some consumer AR applications like *Adobe Aero* [2], *IKEA Place* [46], and *LEGO AR Studio* [56] allow users to create AR content, they rely on predefined assets and manual placement, limiting creative flexibility and expressivity. To address these limitations, recent research has explored generative AI for authoring in extended reality (XR). For instance, systems like *SceneCraft* [43], *3D-GPT* [99], *Ostaad* [3], *VRCopilot* [121], *LLMR* [26], *LLMER* [20], and *Dreamcrafter* [107] integrate large language models (LLMs) for XR scene generation and editing via natural language interaction. While promising, these systems primarily target manually defined environments and lack in-situ authoring, real-world scene understanding, and/or open-ended asset generation, hindering truly personalized AR creation. Furthermore, most scene understanding algorithms are trained on indoor data [21, 36, 50, 79, 101]—so even if prior XR systems sought to incorporate scene understanding, existing models are not readily applicable to outdoor use—despite outdoor AR applications having proven impactful [51, 69, 95].

Suppose anyone could build an AR scene simply by speaking to an AI. A child might turn their backyard into a medieval kingdom by saying, "*Place a pink castle here.*" and "*Add a fire-breathing dragon on the fence!*" An urban planner could preview a structure with, "*Place a five-story apartment building here.*" and "*Make it twice as tall!*" And anyone could build just for fun. This is our vision for AR authoring: enabling users to create immersive scenes grounded in the real world by describing what they imagine. In this paper, we take a step toward that vision by supporting AR authoring in a wide range of static outdoor environments. We introduce *ImaginateAR*, the first mobile tool for AI-assisted AR authoring that generates and arranges virtual assets from speech input, facilitating their seamless integration into the physical world. ImaginateAR achieves this by pushing the boundaries of (1) outdoor scene understanding, (2) fast 3D asset generation, and (3) LLM-driven natural language interaction—each a significant challenge for fully adaptive AR. Together, these advances help bring generative scene authoring—previously confined to VR—into real-world AR.

To address real-world scene understanding, we updated open-vocabulary 3D instance segmentation models—typically trained on indoor data and reliant on user-specified queries—to function autonomously outdoors. Specifically, we enhance *OpenMask3D* [101] with *GPT-4o* [77] for consistent, automatic outdoor semantic labeling and apply *HDBSCAN* [64] clustering to merge redundant object masks. This produces structured scene graphs composed of labeled 3D bounding boxes, enabling spatial reasoning in real-world contexts. To improve usability and ensure a more complete view of the environment, we perform scene understanding offline on pre-scanned environments and retrieve the relevant scene graph at runtime using a *Visual Positioning System (VPS)* [48], rather than requiring users to scan live. For dynamic 3D mesh generation—essential for creativity and personalization—we contribute a pipeline that encourages well-formed AR assets (*i.e.*, complete, volumetric, properly oriented, and scaled), while running significantly faster than prior methods. Our approach expands user input with GPT-4o, synthesizes reference images using *Dall-E 2* [75], segments foreground objects via *DIS* [82], and lifts them into 3D using *InstantMesh* [117]. Finally, a multi-agent LLM pipeline enables speech-driven interaction: a *Brainstorming* agent suggests scene ideas, an *Action Plan* agent determines spatial relationships, and an *Assembly* agent updates the scene graph for coherent placement.

To evaluate ImaginateAR, we conducted a technical assessment of our scene understanding and asset generation pipelines, along with a three-part user study in a public park with 20 participants. Our scene understanding pipeline outperformed the base OpenMask3D [101] model and ablated variants of our pipeline, while our asset generation pipeline achieved comparable quality to state-of-the-art methods but with a significantly faster, sub-minute runtime. As part of our technical evaluation, we also conducted a demonstration-based assessment across varied outdoor scenes, showing that ImaginateAR functions reliably beyond the user study setting. In the user study, participants first explored three authoring modes—*manual*, *AI-assisted*, and *AI-decided*—to evaluate trade-offs between control and automation during different stages of AR authoring. They then used ImaginateAR to freely design their own AR experiences (Part 2), followed by a co-design session to brainstorm future features (Part 3). Participants enjoyed interacting with

ImaginateAR, asking it to "*Put a dancing T-Rex on the grass*" (P1) or "*Make a helicopter hover over the shed*" (P14). Across sessions, users preferred a hybrid approach—leveraging AI for rapid and creative scene generation while retaining manual control for fine-tuned customization. AI assistance accelerated ideation and spatial arrangement, but participants often opted for manual refinement to ensure their creative intent was more precisely reflected in the final scene. We conclude by discussing current limitations and future directions for AI-assisted AR authoring.

In summary, our contributions include: (1) ImaginateAR, a novel AI-assisted AR authoring tool that integrates real-world scene understanding, generative AI, and LLM-based reasoning to streamline content creation; (2) technical innovations in outdoor scene understanding, fast 3D asset generation, and a multi-agent LLM pipeline for speech interaction; and (3) insights into how users engage with AI-assisted AR authoring—including their balance of automation and control, free-form use, and desired future features.

## 2 Related Work

We situate our work at the intersection of HCI and computer vision (CV), drawing from research on AI-powered XR authoring, real-world 3D scene understanding, generative AI for content creation, and AI assistance in creative workflows.

### 2.1 AI-Powered XR Authoring

Because it involves 3D modeling, programming, and spatial design, creating XR content is inherently challenging [7, 66]. To lower this barrier, commercial tools like *Adobe Aero* [2], *Unity MARS* [106], and *Torch* [103] offer direct manipulation interfaces for placing virtual objects, enabling users to manually design scenes, albeit without AI-driven automation or generation. Research prototypes such as *Pronto* [58], *Rapido* [57], and *ARAnimator* [120] simplify AR prototyping through sketches and demonstration-based input, though they primarily support 2D content. Other systems, such as *SemanticAdapt* [22], *ARTiST* [112], and Lindlbauer *et al.* [60], automate content arrangement based on scene semantics but focus on adaptive user interfaces rather than open-ended scene creation. In our work, we explore how generative AI and real-world scene understanding can further lower authoring barriers, taking a step toward enabling anyone to create any AR experience.

Several recent systems have also explored using AI to streamline XR authoring. For instance, *SonifyAR* [98] generates context-aware sound effects in mobile AR by leveraging LLMs. Others, such as *BlenderGPT* [1], *SceneCraft* [43], and *3D-GPT* [99], enable users to generate 3D models via natural language, which can later be arranged into virtual scenes—but they lack fast, in-situ authoring, limiting on-site ideation and iteration. More comprehensive tools like *Ostaad* [3], *DreamCodeVR* [31], *VRCopilot* [121], *Dreamcrafter* [107], and *LLMR* [26] go further by allowing users to iteratively prompt LLMs to build up full XR scenes. While these systems demonstrate the potential of LLM-assisted XR content creation, they primarily target VR and/or rely on predefined asset libraries, limiting expressivity, adaptability, and real-world interaction. Closest to our work, *LLMER* [20] extends LLMR to mixed reality, and Fang *et al.* [29] integrate scene graphs, LLMs, and AR to facilitate robot navigation programming. However, both systems rely on manually constructed scene representations rather than automated scene understanding models. Ultimately, no existing system fully supports in-situ, speech-driven AR authoring with real-world scene understanding and open-ended asset generation. Prior work has also largely overlooked outdoor AR authoring, despite its proven impact in fun, education, public art, and social connection [51, 69, 95].

Building on this foundational work, we explore how outdoor scene understanding, fast 3D mesh generation, and LLM-driven speech interaction can help bring AI-assisted scene authoring—once limited to VR—into real-world AR.

### 2.2 Real-World 3D Scene Understanding

Understanding real-world environments is a fundamental challenge for AR and robotics applications [10, 16]. To seamlessly integrate virtual content into physical spaces, systems must capture both *geometric* and *semantic* properties of a scene. Typically, this is achieved in two steps: first, a 3D map of the environment is built using cameras [67, 91], sometimes augmented with depth or IMU sensors [25, 68]. Next, semantic labels are assigned through CV models trained on 3D datasets [24, 28], enabling object recognition [92, 101]. Beyond individual object detection, some systems structure this information into *scene graphs* [6, 55, 88, 109, 113], where objects are nodes and relationships (*e.g.,* "*a bench is next to a tree*") form edges. This structured representation enables high-level reasoning for context-aware applications, including ours.

Recent advances in multimodal models, such as *CLIP* [83] and vision-language models (VLMs), have enabled open-world object detection [21, 36, 50, 79, 101], allowing models to recognize objects beyond predefined labels. This is critical for real-world use, as environments vary widely—indoor spaces differ from outdoor settings, and even rural and suburban outdoor areas contain distinct objects. Recent efforts in open-vocabulary scene understanding have integrated 3D cues directly into LLMs [45, 63, 118], enabling agents to perform grounding, question-answering, and captioning within 3D environments. While promising, most open-vocabulary segmentation models rely on query-based retrieval [101], identifying scene objects via user prompts or predefined vocabularies. This poses challenges for generating scene graphs: user prompts introduce latency during live graph construction for AR authoring, while defining a single comprehensive vocabulary for arbitrary scenes—needed for offline computation—is difficult. Furthermore, prior work has largely focused on indoor spaces, where object categories are more constrained and fundamentally different from those outdoors.

As such, we explore how existing open-vocabulary 3D instance segmentation models could be updated for outdoor AR—enabling ImaginateAR to generate structured scene graphs of diverse environments through an automatic, offline scene understanding pipeline.

### 2.3 Generative AI for Content Creation

ImaginateAR leverages generative AI for fast, open-ended 3D asset creation, allowing users to verbally generate objects on demand—supporting creative flexibility and expressivity. Traditionally, 3D models are crafted by experts using professional tools, a time-consuming process infeasible for everyday users. While generative models have significantly advanced in 2D content creation, enabling high-quality image generation from text prompts [32, 53, 85–87, 89],

their extension to 3D remains an ongoing challenge. Diffusion-based methods [40] have also improved realism in image synthesis, even supporting controls such as image-based guidance and structured constraints like depth, sketches, and key poses [65, 122]. However, these techniques still focus on 2D outputs rather than 3D assets required for AR.

Generating high-quality 3D content is significantly more complex than image synthesis, requiring solutions that balance efficiency and realism. Early text-to-3D models, such as *DreamFusion* [81], required over 30 minutes on a powerful GPU [39] to generate a single asset, making them impractical for in-situ use. As an alternative, prior systems like LLMR [26] relied on large asset libraries (*e.g., Sketchfab*), which—while expansive—often lack imaginative content such as a "*two-headed giraffe*" (P2), limiting creativity. Today, techniques aim to accelerate 3D asset creation, including zero-shot generation [49] and single-image-to-3D approaches [12, 19, 41, 62, 117]. Among these, *InstantMesh* [117] enables rapid 3D lifting (*i.e.,* reconstructing a 3D shape from a 2D image) and texturing from a single image in seconds. To ensure fast and flexible content generation, ImaginateAR employs *DALL-E 2* [85] to synthesize a 2D image from speech input, then lifts it into 3D using InstantMesh. This pipeline generates a fully textured 3D model in approximately 30 seconds—substantially faster than prior methods in our technical evaluation, and sufficient to support creative iteration in our user study. Generation speed remains a challenge, but 3D generative models are rapidly improving in both speed and fidelity [114–116, 124]. As these models advance, our pipeline can adopt faster or higher-quality components—like replacing InstantMesh—without major system changes.

### 2.4 AI Assistance in Creative Workflows

As a fully functional AI-infused AR authoring tool, ImaginateAR presents a unique opportunity for examining how AI can support creative expression in immersive, real-world environments. While we allow free-form use in our study, we also include a controlled investigation of varying levels of AI involvement to examine trade-offs between automation and human agency—a longstanding concern in HCI [5, 42, 94]. Prior work shows that while AI can enhance expressivity and efficiency, excessive automation may reduce user control or creative ownership [73, 94]. Although this tension has been studied in writing, design, and programming [8, 9, 18, 37, 100], its role in AR authoring remains underexplored. Our study helps fill this gap, uncovering not only what users want to create with ImaginateAR but also how AI can best assist them along the way.

## 3 Design Goals for AI-Infused AR Authoring

Our research is motivated by an overarching belief that AR authoring tools should allow *anyone* to create *anything*, *anywhere*, removing technical barriers and making immersive content creation as effortless as speaking an idea aloud. Imagine a student in their schoolyard curious about ancient civilizations saying, "*Construct a Mayan temple next to the swings.*" and "*Show a person in historical clothing next to it!*". After each request, interactive AR content should quickly appear, blending seamlessly into their surroundings. To pursue this vision, we synthesized the following design goals:

**G1: In-Situ AR Authoring Anywhere.** Users should be able to create, modify, and iterate on AR content directly within their environment, treating their surroundings as a canvas for in-situ authoring. Prior XR authoring systems rely on manually defined and often VR-based environments [3, 20, 26, 29, 31], while scene understanding models typically target indoor spaces and require user queries or predefined vocabularies [50, 79, 101]. Instead, we need to update these models to autonomously interpret a wide range of outdoor scenes.

**G2: Generate High-Quality 3D Assets Quickly.** To support creativity and maintain flow, users need visually compelling AR assets without long waits. Traditional 3D modeling is time-consuming and technically demanding, and while generative models are improving, they often sacrifice either quality or speed (*e.g., ProlificDreamer* [110] takes over 240 minutes on a powerful GPU for a single asset [39]). Achieving in-situ AR authoring requires generating AR-ready 3D assets in seconds—not minutes or hours.

**G3: Simple Speech-Driven Interactions.** AR authoring should feel natural and effortless, letting users create and modify scenes with simple voice commands. For example, in the Mayan temple scenario, a student might say, "*Make the temple bigger*" or "*Remove the person.*" To lower technical barriers, we need LLM-driven speech interactions—enabled by structured scene graphs for spatial context.

**G4: Adjustable AI Assistance.** AI should support—not override—human creativity, offering just the right level of help while keeping users in control. Preferences for AI involvement vary across users and tasks [5, 42, 73, 94]. Additionally, when AI makes mistakes, users need clear ways to recover—such as re-prompting or direct manipulation. To support both flexibility and error recovery, AR authoring systems should let users decide how much AI assistance they want and when, and provide manual tools.

## 4 The ImaginateAR System

Our goal is to explore how AI can help users bring their ideas to life. To support this, we developed *ImaginateAR*, a novel AI-assisted AR authoring tool that lets users create, arrange, and modify virtual content in diverse, static real-world environments using speech.

The ImaginateAR system consists of three key components: (1) an offline scene processing module, (2) a remote asset generation server, and (3) a mobile AR interface. The scene understanding pipeline structures the environment into a scene graph—a compact textual representation of object labels and their 3D bounding box coordinates. When users request content that is not yet available, the server generates 3D assets on demand. The mobile interface lets users issue speech commands, adjust content manually, and visualize their ideas in-situ. At a high level, ImaginateAR retrieves the relevant scene graph, processes voice commands, interprets user intent, fetches or generates 3D assets as needed, updates the scene graph accordingly, and renders changes in the AR scene. We include all LLM and VLM prompts in the Supplementary Materials.

### 4.1 Offline Scene Understanding

To support in-situ AR authoring *nearly* anywhere (*Design Goal 1*), we update an open-vocabulary 3D instance segmentation model to operate autonomously in outdoor environments.

#### 4.1.1 Background Information.

We first introduce *scene graphs* and the *OpenMask3D* model [101], which serve as the foundation of our system.

**What is a Scene Graph?** A scene graph is a structured textual representation of a visual scene, encoding semantic details such as object labels, locations, and spatial extents. This compact format is well-suited for processing by LLMs. Unlike prior approaches like *ConceptGraphs* [36], which include explicit relationship nodes (*e.g.,* "*next to*" or "*on top of*"), our scene graphs focus solely on individual objects and their spatial properties. Modeling inter-object relationships is left as future work.

**What is OpenMask3D?** Our scene understanding method builds on OpenMask3D, a state-of-the-art system for open-vocabulary 3D instance segmentation. Given an input point cloud and an RGB-D video with camera poses, OpenMask3D operates in two stages. First, the *Class-Agnostic Mask Proposal (CAMP)* network generates a pool of 3D binary masks, $S_I$, where each mask represents a potential object instance by marking its corresponding 3D points in the point cloud with a value of 1. Second, a *CLIP* [83] embedding is computed for every mask $M \in S_I$. The system performs a depth-based visibility check to identify frames where $M$ is highly visible. Visible points from these frames are used to prompt the *Segment Anything Model (SAM)* [54] at multiple scales, extracting image regions depicting $M$. These regions are then fed into CLIP to generate embeddings, which are aggregated into a single vector per $M$. At test time, users can query objects via text prompts, which are converted into CLIP embeddings and matched against the precomputed embeddings of all masks to retrieve relevant object instances. Notably, OpenMask3D was trained and evaluated primarily on indoor datasets such as *ScanNet* [24].

However, we identified two main limitations for our use case. First, the CAMP module often produces an excessive number of masks—frequently over 120—making it difficult to construct compact scene graphs that can be efficiently processed by LLMs. Second, relying on user-defined prompts during use introduces latency, as each query must be embedded and compared against the full set of mask embeddings. Precomputing scene graphs with a predefined vocabulary can avoid this cost but requires a comprehensive label set, which is difficult to define given the variability of outdoor scenes. Hence, OpenMask3D needs to be updated for outdoor AR.

#### 4.1.2 Our Process.

We now describe our offline scene understanding pipeline, including how we capture point cloud data and adapt OpenMask3D to address key limitations. We also discuss the scalability of our approach.

**Scene capture.** A key design choice in ImaginateAR is to rely on pre-scans of environments and process them offline, rather than running scene understanding models in real-time as users actively scan their surroundings. We chose this approach for three key reasons: first, it enhances ease of use, as live scene understanding requires users to manually and thoroughly scan their environments, introducing unnecessary friction. Instead, digital twins enable precomputed scene understanding, allowing instant retrieval of scene graph data relevant to the user's location. Second, because users cannot be expected to scan every detail, live scene analysis often results in incomplete context. In contrast, pre-scanned environments can offer a more comprehensive spatial understanding—enabling interactions like placing objects behind the user or real-world structures, even if those areas were never in the camera view. Lastly, real-time scene understanding models typically perform worse than offline methods, especially in complex outdoor environments. That said, relying on pre-scans may limit scalability compared to live methods and may not reflect dynamic scene changes (*e.g.,* a chopped-down tree or moving people), which we discuss later.

To generate a 3D representation of a scene, we capture the environment using a commercial depth-sensing device. In our experiments, we used an iPhone 13 Pro, which has LiDAR, running our custom-built scanning app that records RGB images, depth maps, and camera poses. These data sources are integrated into a 3D point cloud, similar to commercial applications like *Scaniverse* [71] and *Polycam* [80]. Our method is device-agnostic and can be extended to Android devices running *ARCore* [33].

**Pre-Processing.** To ensure accurate scene understanding and protect user privacy, we apply several pre-processing steps to refine captured data. Personally identifiable information (PII), such as faces and license plates, is removed using an off-the-shelf blurring model [84]. We also enhance depth maps by filling holes (*i.e.,* missing values) using a monocular depth model [119]. Because some regions lack depth due to sensor limitations, we infer relative monocular depth and re-scale it with valid LiDAR points to produce dense metric depth maps.

**Initial Mask Prediction.** We use the pre-trained CAMP network from OpenMask3D to generate an initial pool of binary masks, $S_I$, where each mask represents a potential object or object part. However, we observed some masks are small or redundant. Thus, we filter the pool by removing small and duplicate masks, and merging highly overlapping ones, resulting in a refined subset $S_M$.

**Mask Classification.** In this step, we infer a semantic label for each mask in $S_M$. OpenMask3D's CLIP-based strategy requires either generating scene graphs at test time (via user prompts) or using predefined vocabularies. In contrast, we classify each detected object using a vision-language model (VLM) [36]. We modify OpenMask3D's frame selection strategy to select the image with the highest visibility of the object mask, using monocular depth maps to assess point visibility. From this image, we extract two crops: (1) a context crop ($C_k$), which includes surrounding scene details, and (2) an object crop ($O_k$), which isolates the object. These crops are computed only at OpenMask3D's largest scale to better capture contextual information. We leverage *GPT-4o* [77] as the VLM to infer a semantic label from $O_k$ and $C_k$, incorporating a running list of previously predicted labels to enforce consistency. This reduces synonym mismatches (*e.g.,* standardizing "*road*" instead of allowing similar variations like "*road surface*"). We refer to this AI agent as the *Object Classifier*, responsible for generating structured semantic labels across diverse outdoor scenes (Figure 2).

**Semantic Point Cloud and Clustering.** After assigning semantic labels to instance masks, we generate a structured scene representation by storing 3D bounding boxes enclosing each mask in $S_M$. However, $S_M$ may still contain multiple masks for the same object, especially when overlapping masks do not meet the threshold for the prior filter. This redundancy can introduce duplicate instances in the final scene graph. To address this, we compute a final refined set of masks, $S_F$, using semantic information from a

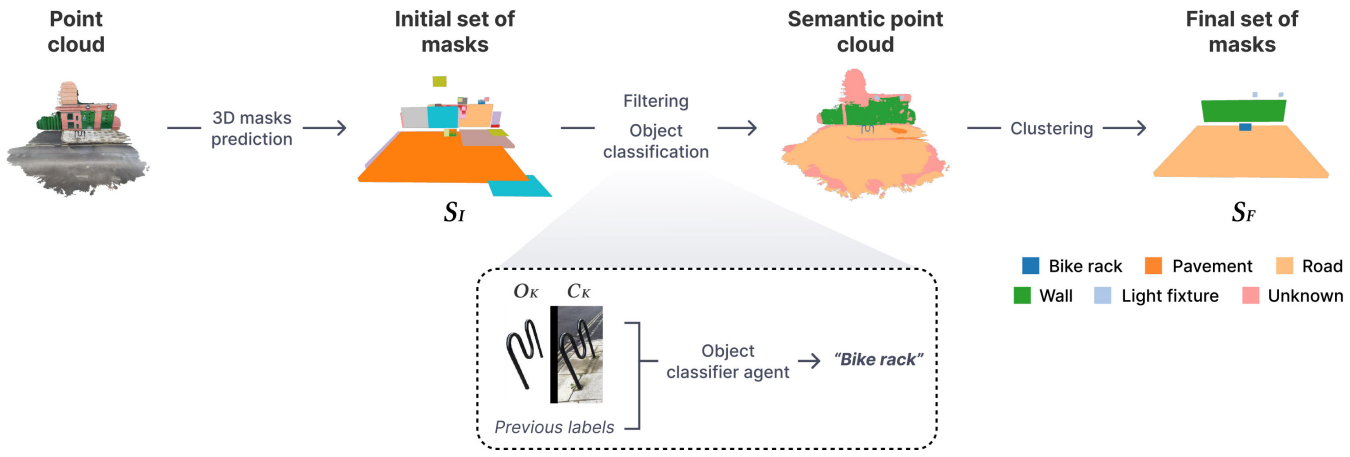



**Figure 2: Diagram of the 3D scene understanding pipeline. Given an input point cloud, we first estimate 3D masks. Next, we assign a semantic label to each mask using a VLM and propagate the label to all points within the mask, producing a semantic point cloud. We then cluster nearby points with the same label to infer the final set of 3D masks, from which we extract 3D bounding boxes. For visualization, we show only the bounding boxes, not the underlying masks. The *Pavement* box is enclosed within the *Road* box and is therefore not visible.**

VLM. For each mask $k$ in $S_M$, we propagate its semantic label to its associated 3D points, producing a semantic point cloud. Points not assigned to any mask are labeled as *unknown* and excluded from the final output. We then apply *HDBSCAN* [64] to cluster nearby points with the same label. This merges spatially close, semantically identical masks (*e.g.*, object parts), producing a more compact set $S_F$ compared to $S_M$. For example, in Figure 2, the number of instances is reduced from 208 ($S_I$) to 15 ($S_M$) and finally to 6 ($S_F$).

**Scene Graph Creation and Deployment.** We construct a scene graph by storing semantic labels along with the minimum and maximum values of the 3D axis-aligned bounding boxes enclosing masks in $S_F$. Since these graphs primarily encode static objects, they remain valid across multiple AR sessions and users, as transient elements (*e.g.*, moving people) are typically absent from traditional point cloud reconstructions. Scene graphs are generated offline using a machine with an NVIDIA L4 GPU; while there is room for optimization, the full pipeline still completes in just a few minutes per scan (Figure 3). During live use, precomputed graphs allow LLM agents to understand the user's surroundings. Tools like *Niantic's Visual Positioning System (VPS)* [48] can estimate a user's precise position relative to the scene graph. For this study, we manually captured all scenes. However, we believe our offline scene understanding pipeline could scale to large pre-scanned datasets already available through platforms like VPS, *Google Street View* [35], and *Geospatial API* [34]. For instance, Niantic VPS currently supports over 1 million scanned locations [72]. Leveraging such resources would enable scalable deployment of ImaginateAR.

## 4.2 Dynamic Asset Generation

Running 3D generation models directly on mobile devices is computationally prohibitive. To enable fast AR asset creation (*Design Goal 2*), we deploy a private web server that generates 3D models remotely based on users' speech commands. For example, a user might say, "*Place a dragon perched on the lamppost*," prompting the server to return a corresponding textured mesh of a dragon.

To generate assets, we first use a text-to-image model to synthesize an initial image, then apply *DIS* [82] to segment the foreground subject from the background. While any text-to-image model can be used, image quality does significantly impact the resulting 3D mesh. Images with complex backgrounds, occlusions, or flat perspectives often produce unrealistic models. To address this, we enhance user prompts using *GPT-4o mini* [76], which expands them with clarifying keywords (*e.g.*, "*white background*") to improve visual clarity and depth. We also provide the model with examples of good and bad images. This step—*prompt boosting*—helps guide the generated images to meet the requirements for reliable 3D reconstruction. To further improve quality, we instruct *Dall-E 2* [85] to edit only the central region rather than generate the full image, encouraging a fully visible, well-defined subject suitable for meshing. We then use *InstantMesh* [117], an efficient single-image-to-3D model, to lift the image into a fully textured mesh. Because asset generation relies on external services, occasional outages may occur. In such cases, we fall back to the original user prompt (without boosting) or switch to *Stable Diffusion Turbo* [90] as a local text-to-image generator. Figure 4 illustrates the full pipeline.

## 4.3 Real-World User Authoring

To support seamless in-situ AR authoring (*Design Goal 3*), we developed a mobile interface that enables speech-driven interactions with advanced AI models. We built it using *Unity* 2022.3.33f1[1], *ARFoundation* 5.1.4 [104], and *Niantic Lightship ARDK* 3.5.0 [47].

We designed ImaginateAR to support five core interactions for authoring an AR scene: *brainstorming*, *model creation*, *placement*, *editing*, and *removal*. For each task, users can choose from three levels of AI involvement (*Design Goal 4*): "*manual*", where they

[1]https://unity.com

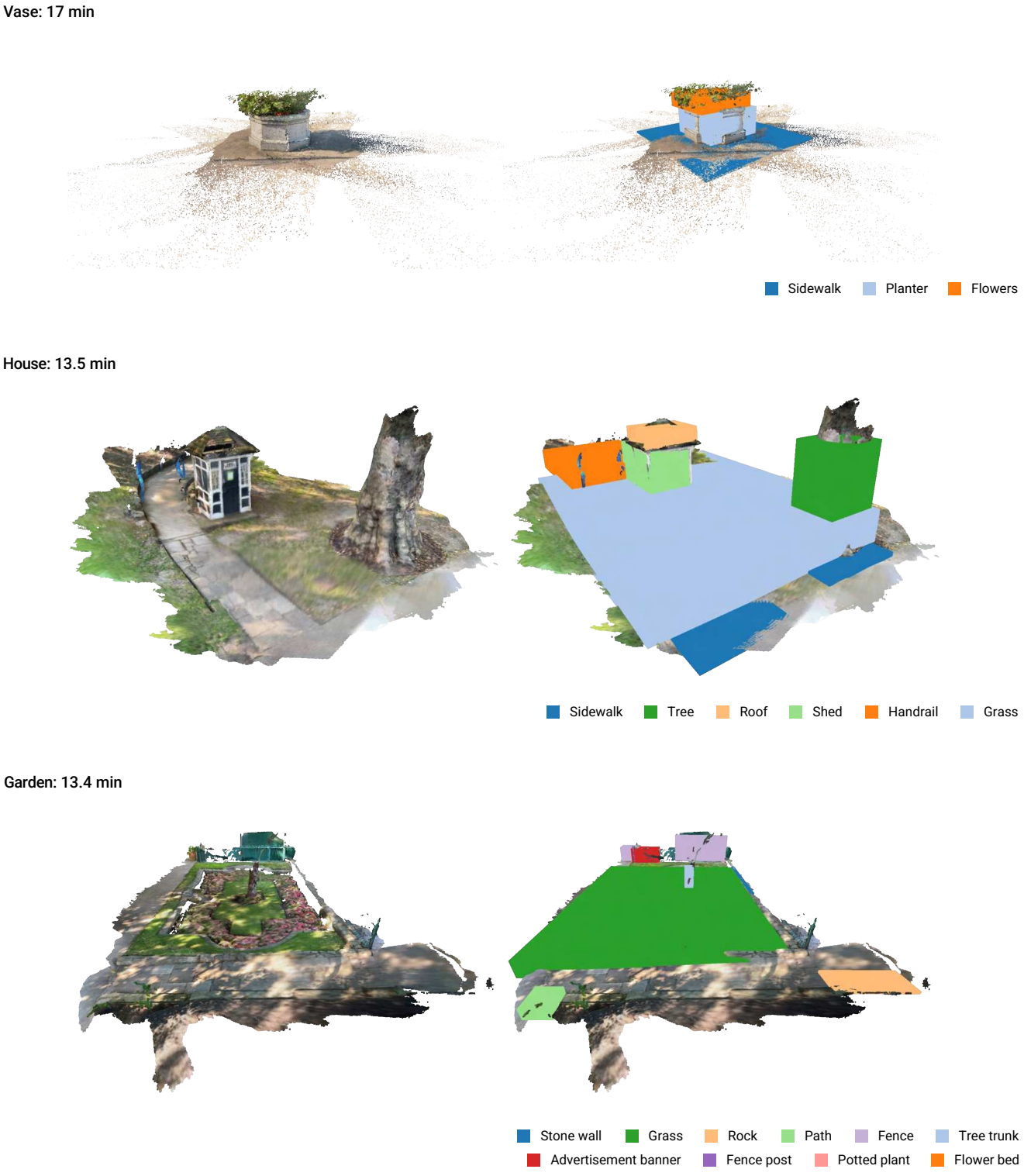



**Figure 3: Results of the 3D scene understanding module. For each of the three scans—*Vase*, *House*, and *Garden*—we visualize the input point cloud (left) and the final set of labeled 3D bounding boxes inferred by our scene understanding pipeline (right). We also report the total time (in minutes) required to estimate the scene graph for each scan. Note that some bounding boxes may be enclosed within others and may therefore be occluded.**

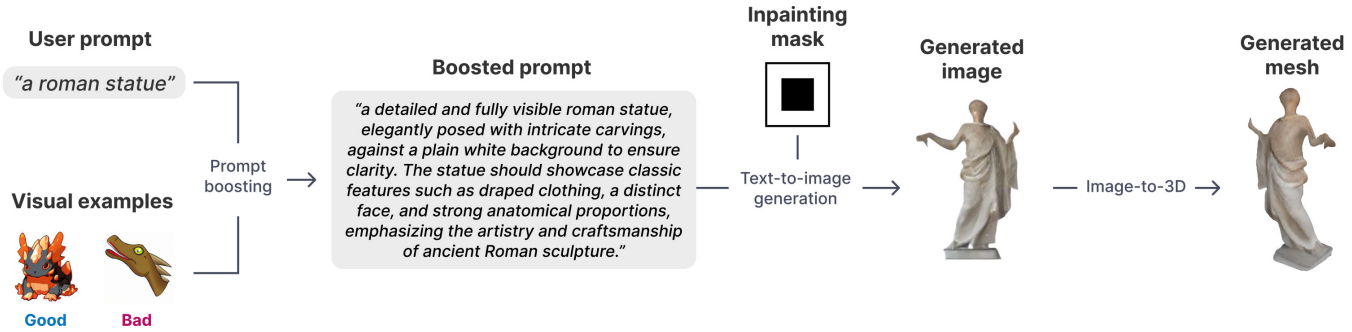



**Figure 4: Example of 3D asset generation. Given a user prompt, we first apply prompt boosting, then use Dall-E 2 [85] to generate a consistent image by editing the center region of a white canvas. The image is then lifted to 3D using InstantMesh [117]. The 'Bad" example (right) illustrates a failure case because it would produce a partial 3D object (*i.e.,* only the dragon's head). Prompt boosting helps avoid such incomplete generations.**

maintain full control; "*AI-assisted*", where the system offers multiple suggestions; and "*AI-decided*", where AI autonomously executes the task and presents a single best option. To facilitate these interactions, ImaginateAR employs three specialized LLM agents: a *Brainstorming* agent for idea generation, an *Action Plan* agent for interpreting user requests and structuring tasks, and an *Assembly* agent for executing actions like asset placement. AI-assisted and AI-decided modes share this LLM pipeline but differ in autonomy and how results are presented to the user. Figure 5 illustrates the interface and supported interactions.

**Localization.** Users begin by pointing their phone around to localize to a nearby *Point of Interest (POI)*—a geotagged location—using Niantic's VPS. Once the system determines the user's position, it retrieves the corresponding precomputed scene graph, providing a structured representation of its surroundings for the LLM agents. ImaginateAR then displays: "*I'm ready! Let's start decorating!*"

Our system updates the retrieved scene graph to reflect the evolving AR experience. As users request new virtual content, it is added to the local scene graph. Each object has a unique identifier (GUID), a name, and position, rotation, scale, and bounding box dimensions in Unity's world coordinate system. The graph also includes an on-screen visibility flag for handling spatially ambiguous queries (*e.g.,* "*Place the T-Rex here*") and an action tag to track LLM-assigned modifications awaiting execution. Together, this structure provides essential context for iterative, LLM-driven interactions.

**Brainstorming Ideas.** Before editing the AR scene, users can brainstorm using a post-it-style interface triggered by the light bulb button. They can type ideas manually or ask AI for suggestions—either in a single prompt (AI-decided) or through back-and-forth conversation (AI-assisted). When speaking to the Brainstorming agent, ImaginateAR captures audio using Unity's microphone[2], transcribes it with *Whisper* [78], and prompts GPT-4o along with the current scene graph to ground ideas in the user's AR environment. The post-it window is movable to prevent visual obstruction and can be closed by tapping the button again.

**Creating 3D Assets.** Users can add virtual content by selecting from a preset library or asking the AI to generate new assets. For manual selection, tapping the book button in the bottom left opens a scrollable grid of virtual objects. For AI-driven creation, users tap the microphone button and describe what they want. The system returns the top result (AI-decided), with optional left and right arrows to browse alternatives (AI-assisted). To support AI-assisted creation, ImaginateAR runs three asset generators in parallel, each producing a distinct asset aligned with the user's request.

If AI creation is used, ImaginateAR transcribes the user's speech and sends it—along with the current scene graph—to the *Action Plan* agent. This agent assigns each virtual object an action tag: (1) *none* (no change), (2) *remove*, (delete from the scene), (3) *update* (modify properties like position, rotation, or scale), (4) *create_resources* (instantiate a preset model), (5) *create_persistent* (load a previously generated model), or (6) *create_new* (request a new mesh from the remote asset generation server). ImaginateAR then either retrieves an existing model (*create_resources*, *create_persistent*) or generates a new one remotely (*create_new*). The assets are added to the scene to compute spatial properties like bounding box dimensions.

**Arranging Virtual Content.** Users can place, modify, and remove virtual objects either manually or with AI tools. For manual placement, users tap the 'Place Object' button to position a selected model at the blue visual indicator, which marks where a ray from the center of the screen intersects ARDK's live mesh [70] (*i.e.,* the estimated geometry of the real world). Tapping on a placed object opens an editing window for adjusting position, rotation, and scale (manual modification) or deleting the object (manual removal).

In AI mode, users can issue verbal commands such as "*Put a silly hat on the statue.*" The *Assembly* agent interprets action tags assigned by the Action Plan agent and determines how to arrange content. The Assembly agent uses each object's transform, along with its minimum and maximum bounds (computed via a *BoxCollider*), for spatial reasoning—such as aligning the top of a statue with the base of a silly hat or scaling a T-Rex to appear larger than nearby objects. It determines each object's placement, rotation, and scale to make it look realistically situated in the real-world scene. The agent then performs AI-decided placement (for `create_resources`, `create_persistent`, and `create_new` tags), modification (`update`), or removal (`remove`), displaying the top result by default. Users can use the left and right arrows to browse

[2]https://docs.unity3d.com/ScriptReference/Microphone.html

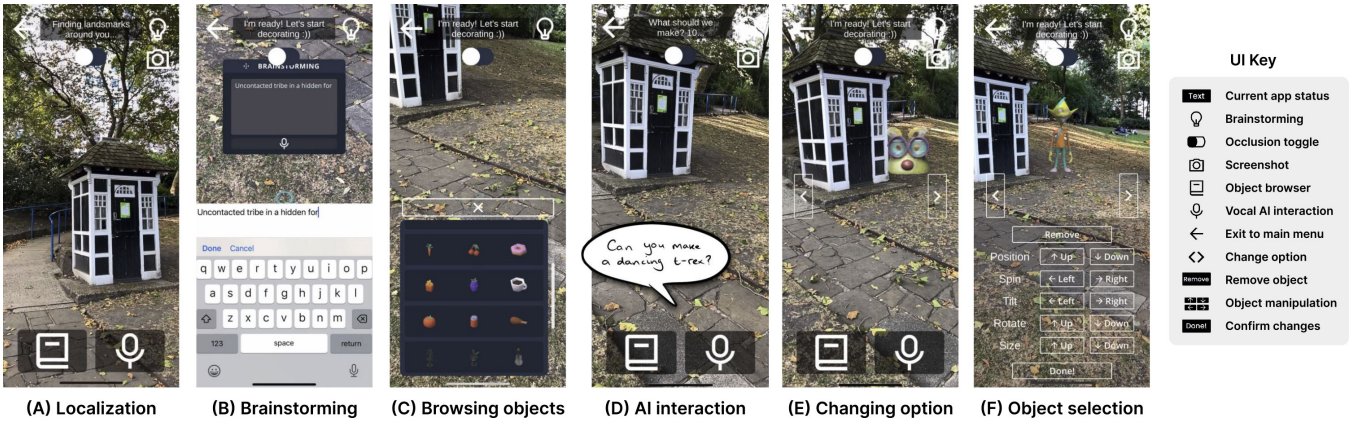



**Figure 5: Different screen captures of the ImaginateAR's mobile interface showing the UI layout and functionalities. Users can access manual, AI-assisted, and AI-decided modes across different features through buttons on the screen.**

alternative placement, modification, or removal options (AI-assisted mode), generated by three parallel Assembly agent (LLM) calls.

**Example AI Creations.** During both the technical evaluation and user study, users had access to the full set of features. Figure 7 showcases AR scenes authored by the research team, while Figure 11 highlights participant-created scenes, often composed using a mix of AI and manual tools. To isolate the performance of ImaginateAR's AI components, we also captured examples generated entirely by AI—without any manual input from participants—in Figure 12.

## 5 Technical Evaluation

We conducted a technical evaluation of ImaginateAR to assess the performance of its core components. First, we measured component-level latency to evaluate its feasibility for in-situ, real-time authoring (Table 1). Across 50 trials, our system averaged 33.92 ± 5.83 seconds—substantially faster than prior systems like LLMR [26], which reports 90.98 ± 24.88 seconds in an empty VR scene and 49.16 ± 7.87 seconds in a virtual bathroom, though with the caveat that its latency primarily stems from iterative refinement, whereas ours is due to asset generation. Next, we compared our two key technical contributions—scene understanding and asset generation—against state-of-the-art baselines. Finally, we conducted a proof-by-demonstration to illustrate that ImaginateAR can scale across diverse outdoor environments.

**Table 1: Latency analysis of key components in ImaginateAR. We report mean ± standard deviation (in seconds) for each pipeline step, averaged over 50 trials.**

| Component | Time |
|---|---|
| Prompt Boosting | 2.53s ± 0.91s |
| Image Generation | 12.53s ± 2.48s |
| Background Removal | 0.04s ± 0.002s |
| Image to Mesh | 9.14s ± 0.08s |
| In-App LLM Agents | 9.68s ± 1.24s |
| Total | 33.92 ± 5.83s |

### 5.1 Scene Understanding Pipeline

We evaluated our scene understanding pipeline on five distinct outdoor scenes. Because existing outdoor benchmarks primarily focus on driving scenarios [17, 30], they are unsuitable for our purposes. We therefore captured our own data and generated ground truth scene graphs by manually labeling each scene. Each node in a graph represents an object as a 3D bounding box and a human-defined semantic label. One member of the research team performed the initial labeling, and two others reviewed it for bias and accuracy.

To create these ground truth graphs, we developed a custom annotation tool that loads point clouds and allows users to brush over points using different colors and brush sizes. This lets users assign a unique color to each object and define its semantic label, producing a structured scene graph. Using this dataset, we evaluated how well different methods detect and describe objects. To compute metrics, we used the Hungarian algorithm to match predicted bounding boxes to ground truth boxes based on Intersection over Union (IoU). A match was counted as a true positive if IoU $\geq 0.25$.

We report the following metrics: mean Recall, computed as the average per-scene Recall (true positives over ground truth instances), and mean Semantic Similarity (mean SS), the average cosine similarity between CLIP [83] embeddings of ground truth and predicted labels for true positives. We also report total predicted masks (N) per method. Across all five scenes, there are 27 ground truth instances. Experiments using GPT-4o were repeated five times with $\text{top}_p = 0.1$ using the latest available model.

Table 2 ablates variants of the scene understanding pipeline. The first row reports OpenMask3D [101] results using a 4,500-class vocabulary from [123] to assign a label to each detected mask. OpenMask3D shows strong recall, but the large number of predicted masks suggests many may be redundant, creating distractors for LLM agents. Ablation A replaces CLIP with GPT-4o and adds a filtering step to reduce the number of masks. While this lowers the total number of masks, it also reduces the number of correctly predicted masks and semantic label quality. Ablation B incorporates dense monocular depth in metric scale, improving both recall and semantic similarity—suggesting that better visibility yields more accurate crops. Ablation C reintroduces CLIP on the same inputs as

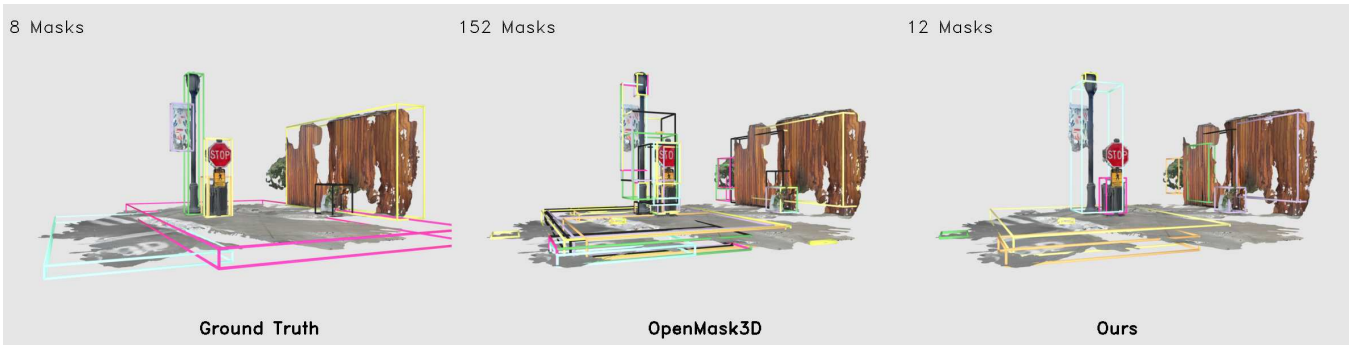



**Figure 6: From left to right: bounding boxes from the ground truth, OpenMask3D [101], and our proposed method. OpenMask3D predicts a large number of masks, resulting in excessive bounding boxes that over-represent the same scene objects. In contrast, our method produces fewer, more accurate boxes. (Box colors are arbitrary and can be ignored.)**

**Table 2: Evaluation of 3D scene understanding pipelines. We build on OpenMask3D [101] to produce more compact scene graphs. The benchmark includes five manually labeled scenes with 27 total ground truth bounding boxes. We report the total number of predicted masks (N), mean Recall, and mean Semantic Similarity (mean SS) using a 0.25 IoU threshold. Rows A–C are ablations: (A) adds GPT-4o labeling and initial mask filtering, (B) incorporates monocular depth, and (C) uses CLIP [83] instead of GPT-4o. GPT-4o results are averaged over five runs and reported as mean ± standard deviation.**

| | Components Used | | | | | Evaluation Metrics | |
|---|---|---|---|---|---|---|---|
| Method | Filtering | Monocular Depth | Labeling | Clustering | N | mean Recall ↑ | mean SS ↑ |
| OpenMask3D [101] | | | CLIP | | 752 | 0.800 | 0.738 |
| Ablation A | ✓ | | GPT-4o | | 59 | 0.508 | 0.659 (± 0.008) |
| Ablation B | ✓ | ✓ | GPT-4o | | 60 | 0.558 | 0.791 (± 0.010) |
| Ablation C | ✓ | ✓ | CLIP | | 60 | 0.558 | 0.730 |
| **Ours** | ✓ | ✓ | GPT-4o | ✓ | 49 (± 1) | 0.622 (± 0.087) | 0.791 (± 0.073) |

B but produces lower semantic scores, indicating that GPT-4o yields more accurate labels. Finally, our full method adds a clustering step to merge nearby masks with the same label, further reducing redundancy and producing compact yet meaningful scene graphs.

## 5.2 Asset Creation with AI

To evaluate the efficiency and quality of our text-to-3D generation pipeline, we leveraged T$^3$Bench [39], a benchmark designed to assess text-to-3D methods across varying scene complexities. T$^3$Bench provides standardized text prompts and computes a quality score based on multi-view 2D renderings generated from 3D input assets. It also includes benchmarking results for state-of-the-art text-to-3D models, including *ProlificDreamer* [110], *MVDream* [93], *DreamFusion* [81], and *DreamGaussian* [102].

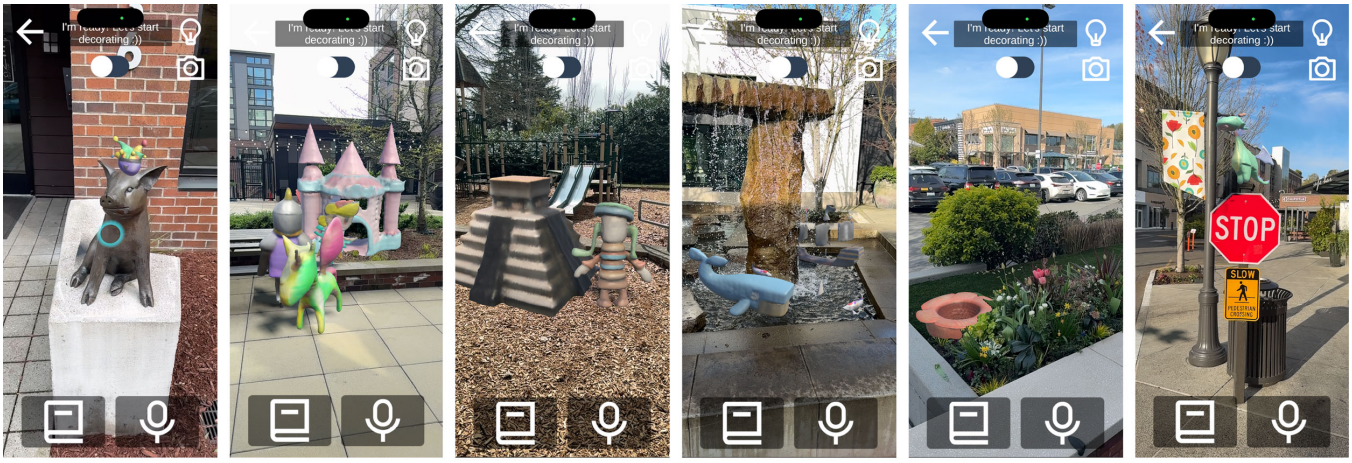
We report official scores and timings for these methods in Table 3 and compare them against our strategy using the single objects generation benchmark. Our method achieves sub-minute generation times—crucial for in-situ AR authoring—while maintaining reasonable visual quality. Although our assets are slightly lower in quality than those from ProlificDreamer and MVDream, they outperform DreamFusion and DreamGaussian. However, higher-quality models come at a significant cost: ProlificDreamer requires 240 minutes and MVDream 30 minutes per asset on a powerful GPU, making them unsuitable for real-time AR. In contrast, our approach balances speed and quality, enabling fast asset generation while preserving usability—making it the most practical solution for in-situ AR authoring. As 3D generative models continue to improve in both speed and fidelity [114–116, 124], future work should explore these evolving alternatives.

**Table 3: Benchmark results comparing state-of-the-art text-to-3D pipelines with our approach, evaluated on the T$^3$Bench dataset [39]. Prior methods are impractical for in-situ AR authoring due to long runtimes. Our approach, combining InstantMesh with Dall-E 2 and prompt boosting, achieves sub-minute generation while maintaining quality.**

| Model Name | Time | Quality ↑ |
|---|---|---|
| DreamFusion [81] | 30 min | 24.9 |
| ProlificDreamer [110] | 240 min | 51.1 |
| MVDream [93] | 30 min | 53.2 |
| DreamGaussian [102] | 7 min | 19.9 |
| InstantMesh [117] + Dall-E 2 [75] | < 1 min | 32.6 |
| InstantMesh + Dall-E 2 + Prompt Boosting (Ours) | < 1 min | 34.8 |

## 5.3 Proof by Demonstration

To evaluate whether ImaginateAR scales across diverse outdoor settings, we conducted a proof-by-demonstration study at 10 Points of Interest (POIs) spanning five distinct sites in two cities. These included statues, flower beds, trees, fountains, play structures, and




(A) A pig sculpture wearing a silly hat.

(B) A backyard fairytale where a knight and their pet unicorn protect a princess.

(C) An interactive Mayan history lesson set in a playground.

(D) An aquarium in a public fountain.

(E) Planting endangered species (Rafflesia) in a flower bed.

(F) A dragon perched on a stop sign.

**Figure 7: Six example creations from our technical evaluation, situated in a park, schoolyard, playground, shopping center, and backyard. Each scene was first generated with AI tools, then refined with light manual adjustments to reflect typical ImaginateAR use. Some are whimsical (A, F), while others are educational (C, E) or playful (B, D).**

more. Figure 7 showcases example AR scenes created by the research team using ImaginateAR. For instance, we authored a fairytale in a backyard, a Mayan history lesson on a playground, and an aquarium inside a public fountain—demonstrating ImaginateAR's adaptability across varied environments.

## 6 User Study

To complement our technical evaluation, we conducted a three-part within-subjects user study with 4 pilot participants and 20 study participants. This in-situ study took place in a public park and aimed to: (1) explore the types of AR experiences users want to author outdoors, (2) observe how and when users engage with manual and AI-driven features, and (3) identify current limitations and future opportunities in AI-infused AR authoring.

### 6.1 Participants

We recruited participants via mailing lists and snowball sampling, screening them through a demographic questionnaire on age, gender, and experience with 2D/3D creativity tools, AR technologies, and AI chat systems. To be eligible, participants had to be at least 18 years old with no visual or auditory impairments. From 147 respondents, we invited 34 to balance demographic diversity and prior experience; 24 participated in the study (4 in pilot sessions).

Participants ranged from 18 to 61 years old ($M$ =35, $SD$ =11.8) and identified as 33.3% female, 58.3% male, and 8.3% non-binary. Half had no prior experience with 3D creativity tools, while 25.0% were slightly familiar, 12.5% very familiar, and the remainder evenly split between moderately familiar and familiar. In AR, 4.2% were unfamiliar, 33.3% moderately familiar, and the rest evenly distributed across slightly familiar, familiar, and very familiar. AI chat systems were more widely used: 12.5% were familiar, 37.5% very familiar, and the remainder evenly divided between slightly and moderately familiar. Participants received a £50 gift card for their time.

### 6.2 Procedure

Our in-person study took place in a busy public park featuring varied terrain, including grass, pavement, stairs, a shed, and trees. This complex setting allowed participants to interact with diverse real-world objects while testing ImaginateAR's adaptability. Study sessions were recorded, capturing participants' phone screens and audio for later analysis. We collected both quantitative and qualitative data through surveys and semi-structured interviews, with full study materials available in the Supplementary Materials. Each 2-hour session included an initial tutorial and three study phases:

**Tutorial.** The session began with participants watching a 5-minute introductory video explaining the study and system features. They then had the opportunity to ask questions before proceeding.

**Part 1: Comparison Task.** As a novel outdoor AR authoring tool, ImaginateAR raises open questions about AI's role in the authoring process. To explore when and how much AI involvement users preferred, we first conducted a structured comparison before allowing free-form creation. Participants began with a 3-minute overview video before interacting with three system modes: (A) *manual*, where users tapped the screen and physically moved to manipulate the AR scene; (B) *AI-assisted*, where the AI suggested options but users made final decisions; and (C) *AI-decided*, where the AI autonomously generated a single output. They performed five core AR authoring tasks—(1) *brainstorming*, (2) *object creation*, (3) *placement*, (4) *modification*, and (5) *removal*—across all three AI modes, completing 15 trials (1A–5C; see Table 4). Mode order was counterbalanced using a Latin Square. After each trial, participants completed a post-task questionnaire with UMUX-LITE [59], a two-item usability measure adapted from SUS [15], and NASA-TLX [38] ratings for mental demand, performance, effort, and frustration. At the end of this phase, we asked which mode participants preferred overall and which they would use for additional features such as music, sound effects, animations, event triggers, and object pinning.

**Table 4: The three AI modes and five features (15 total trials) participants engaged with in Part 1 of the study.**

| Feature | A: Manual | B: AI-Assisted | C: AI-Decided |
|---|---|---|---|
| Brainstorming Ideas | User either thinks aloud or writes down ideas in the app. | User converses with LLM to collaboratively come up with idea(s). User chooses final idea. | Single-turn communication with LLM for ideation. |
| 3D Asset Creation | User searches for and selects 3D assets from pre-existing database. | AI generates three different 3D assets, of which the user selects one. | AI generates and selects a single 3D asset. |
| Object Placement | User moves the cursor by aiming the camera, then taps to place the object. | AI determines three different positions to place the object, of which the user selects one. | AI determines where to position the newly-created object. |
| Object Modification | User taps to select the object, then moves around and taps buttons to edit its pose. | User asks LLM to edit the object's pose. AI determines three edit arrangements, of which the user selects one. | User asks LLM to edit the object's pose. AI chooses the final arrangement. |
| Object Removal | User taps on an object and then taps a button to remove it. | User asks AI to remove object(s). AI shows three possibilities, of which the user selects one. | User asks AI to remove object(s). AI chooses the final removal(s). |

**Part 2: Free-Form Authoring Task.** Beyond structured comparisons, observing how and what users create without researcher intervention is critical—and only possible with a fully functional prototype. In this phase, participants used the full ImaginateAR system to freely author AR scenes of their own imagination for 10–30 minutes. Afterward, they completed the Creativity Support Index (CSI) [23] questionnaire and provided qualitative feedback on ImaginateAR's perceived usability and creativity support.

**Part 3: Brainstorming and Co-Design.** Lastly, we conducted a semi-structured interview to gather insights on participant experiences, preferred features, and ideas for system improvement. We prepared 11 qualitative questions covering what they created, their workflow choices, trade-offs between manual and AI-driven authoring, and desired future enhancements. Follow-up questions were asked based on responses, aiming to identify ImaginateAR's limitations and opportunities for future development.

## 6.3 Analysis

We analyzed data from three sources: questionnaire responses, session observations, and interview transcripts. Quantitative data were examined using a Friedman test, followed by Wilcoxon signed-rank tests with Holm's sequential Bonferroni correction for pairwise comparisons. Qualitative data were analyzed using reflexive thematic analysis [13, 14]. The first author developed an initial codebook, which was refined collaboratively with another researcher. The final codebook comprised 56 codes, applied to 412 participant quotes and reviewed by an additional researcher.

# 7 Results

We first present findings from structured comparison tasks—including perceived usability, task load, and creativity support—to understand how different levels of AI involvement affect AR authoring. Next, we analyze free-form authoring behaviors to offer deeper insight into how users naturally engage with ImaginateAR and the types of AR experiences they create. Finally, we synthesize key themes from qualitative feedback, highlighting user preferences, expectations around AI collaboration, and opportunities for designing future AI-powered AR authoring tools. Participant quotes have been lightly edited for clarity and concision.

## 7.1 Comparing Levels of AI Involvement

In Part 1, we quantitatively compared (A) manual, (B) AI-assisted, and (C) AI-decided modes across five core AR authoring tasks: brainstorming, object creation, placement, modification, and removal. Post-trial questionnaires measured usability (UMUX-LITE) and task load (NASA-TLX), with Table 5 showing overall results and Figure 8 highlighting significant differences. This phase aimed to establish an initial comparison of AI involvement across tasks.

**Usability.** UMUX-LITE scores showed no significant differences in overall usability across AI modes. However, analyzing individual questions revealed task-specific differences in how well each mode met participants' needs. Friedman tests found significant differences for brainstorming ($\chi^2(2, N = 20) = 6.58$, $p < 0.05$) and object modification ($\chi^2(2, N = 20) = 13.07$, $p < 0.01$), but not for other tasks. Post-hoc Wilcoxon signed-rank tests revealed that AI-assisted ($V = 151, p < 0.05$) and AI-decided ($V = 165, p < 0.05$) modes better met user requirements for brainstorming than manual. Conversely, manual outperformed AI-assisted ($V = 11, p < 0.05$) and AI-decided ($V = 23.5$, $p < 0.05$) for object modification. For the ease-of-use question, no significant differences were observed across modes.

**Task Load.** NASA-TLX scores showed a significant difference for brainstorming ($\chi^2(2, N = 20) = 7.21, p < 0.05$), with manual mode inducing significantly higher overall task load than AI-decided ($V = 21, p < 0.05$). We also examined the mental demand, performance, effort, and frustration components separately, as these dimensions were particularly relevant to our study.

**Mental Demand.** No significant differences in mental demand were found across modes, indicating no evidence that any particular mode was more mentally demanding than others.

**Performance.** Object modification performance differed significantly across modes ($\chi^2(2, N = 20) = 11.29, p < 0.01$), with manual outperforming AI-assisted ($V = 82$, $p < 0.01$) and AI-decided ($V = 88, p < 0.01$).

**Effort.** Object creation effort differed significantly across modes ($\chi^2(2, N = 20) = 9.14$, $p < 0.01$), with manual requiring significantly less effort than both AI-assisted ($V = 41.5$, $p < 0.05$) and AI-decided ($V = 36, p < 0.05$). When asked why, participants noted that while AI features demanded less active input and decision-making, they still had to wait for system responses—suggesting they equated effort with overall task duration.

**Table 5: Usability (UMUX-LITE, on the left) and task load (NASA-TLX, on the right) data collected in Part 1. We report average ± standard deviation. For statistically significant data, we also provide a plot.**

| Feature | A | B | C |
|---|---|---|---|
| Brainstorming Ideas | 66.2 ± 16.4 | 76.8 ± 7.4 | 75.4 ± 10.6 |
| Creating Objects | 69.8 ± 11.2 | 70.8 ± 12.2 | 67.9 ± 13.2 |
| Modifying Objects | 73.0 ± 16.0 | 66.2 ± 14.9 | 68.1 ± 13.5 |
| Placing Objects | 72.2 ± 15.3 | 70.6 ± 10.9 | 67.6 ± 15.9 |
| Removing Objects | 79.2 ± 16.7 | 80.9 ± 8.6 | 79.5 ± 12.4 |

**UMUX-LITE**

| Feature | A | B | C |
|---|---|---|---|
| Brainstorming Ideas | 48.6 ± 20.3 | 40.2 ± 20.2 | 37.3 ± 18.0 |
| Creating Objects | 27.5 ± 12.3 | 33.8 ± 12.0 | 35.0 ± 12.2 |
| Modifying Objects | 30.5 ± 15.6 | 32.0 ± 15.0 | 34.1 ± 14.9 |
| Placing Objects | 34.3 ± 18.2 | 28.8 ± 12.7 | 33.2 ± 13.2 |
| Removing Objects | 24.1 ± 14.5 | 24.8 ± 13.5 | 23.2 ± 14.5 |

**NASA-TLX**

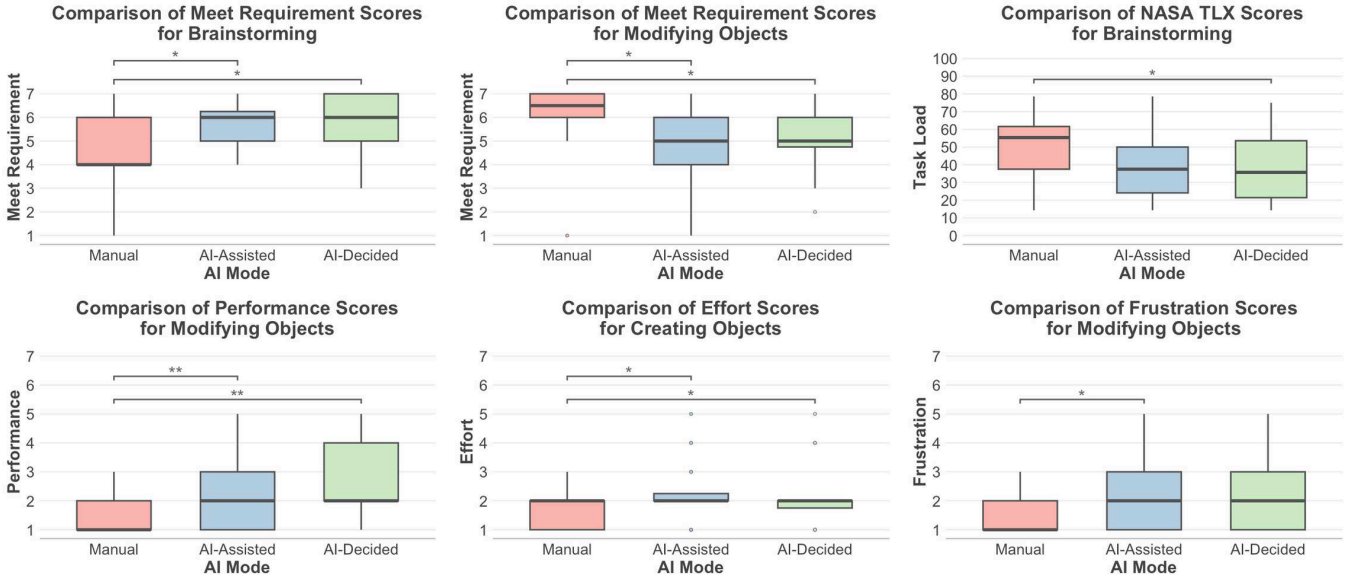



**Figure 8: Boxplots of significant results from Part 1 quantitative data. Higher values indicate better outcomes for Meet Requirements, while lower values are better for NASA-TLX, Performance, Effort, and Frustration scores.**

**Frustration.** Frustration during object modification varied significantly ($\chi^2(2, N = 20) = 8.39$, $p < 0.05$), with manual mode causing less frustration than AI-assisted ($V = 45$, $p < 0.05$).

**Overall Preference.** After completing all trials, 12 participants preferred manual mode, 10 favored AI-assisted, and 2 equally preferred both (P5, P16). Participants appreciated the manual mode for its control (10/20) and precision (9/20), helping them create scenes that more precisely matched their vision. AI-assisted was valued for fostering creativity (6/20) and offering multiple AI-generated options for review (5/20). AI-decided was least favored, as participants found it "*too rigid and deterministic*" (P5), though some acknowledged its ability to quickly generate results (4/20) and reduce the mental effort of decision-making (4/20).

**Authoring Preferences for Additional Features.** Participants proposed future features and indicated their preferred AI mode for each, including background music, sound effects, animations, event triggers, and object pinning. Preferences are summarized in Figure 9. Overall, participants favored manual mode for tasks requiring fine-grained control, such as pinning objects to specific parts of real-world surfaces, and preferred AI-assisted mode for creative, generative tasks like adding sounds and animations.

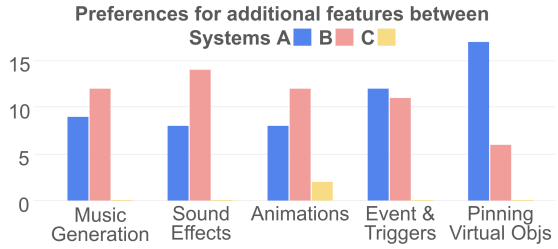



**Figure 9: A bar graph showing participant preferences for level of AI involvement across proposed additional features.**

**Summary.** While AI-assisted mode was expected to be the most preferred, participants' preferences varied across tasks due to trade-offs between speed, creativity, and precision. AI-assisted was appreciated for generating creative options with less decision-making, but manual mode was valued for precise adjustments, such as fine-tuning object placement, despite requiring more active input and

time. AI-decided was helpful for brainstorming but lacked the control needed for tasks driven by specific user intent. These findings suggest that future AI-powered AR authoring tools should support all three modes, enabling users to adjust automation and control based on their needs at different stages of the authoring process.

## 7.2 Free-Form Authoring with ImaginateAR

In Part 2, participants freely authored AR scenes using the full ImaginateAR system for 10–30 minutes before providing quantitative and qualitative feedback. Below, we present findings on creativity support, followed by an analysis of what participants created and how they used the system without researcher intervention.

**Creativity Support.** ImaginateAR received an average Creativity Support Index (CSI) score of 68.8 (SD = 18.0). CSI scores can be mapped to educational grading scales [23], and since our study was conducted in the UK, this corresponds to an '*Upper Second-class Honours*'—the second-highest classification [52]. Participants rated *Results Worth Effort* (M = 2.65, SD = 1.50) and *Exploration* (M = 2.50, SD = 1.24) as the most important factors in AR authoring. On a 1–10 scale, ImaginateAR scored 6.65 (SD = 2.22) for Results Worth Effort and 6.36 (SD = 2.17) for Exploration. The highest-rated aspects of the system were *Enjoyment* (M = 7.71, SD = 1.65) and *Expressiveness* (M = 7.55, SD = 2.24). These results suggest participants valued the ability to explore and achieve meaningful outcomes—well-supported by ImaginateAR—while also finding the experience engaging and expressive. See Figure 10.

**Participant Creations.** All participants successfully authored at least one AR scene. See Figure 11 for all 24 creations. These ranged from "*a sphinx and a pyramid rising from the ground*" (P6) to "*a cat chasing a row of yellow ducks*" (P16) and "*animals drinking coffee while watching a spaceship launch*" (P19). Some built whimsical scenes for general audiences (7/20), while others designed for friends (5/20) or family (3/20). A few explored more story-driven experiences (4/20). Regardless of intent, 14 out of 20 participants explicitly mentioned having fun while using ImaginateAR. The variety of creations suggests that ImaginateAR effectively supported a wide range of authoring goals, demonstrating both flexibility and robustness in real-world use.

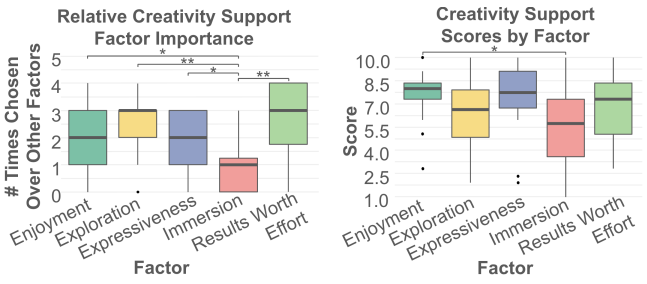



**Figure 10: Left: Average number of times each CSI factor [23] was selected as more important than another. Participants rated Results Worth Effort and Exploration as most important, with Immersion rated significantly lower than all other factors. Right: The scores participants gave ImaginateAR by factor. Participants found ImaginateAR enjoyable and expressive, but not necessarily immersive.**

**Authoring Strategies with ImaginateAR.** Most participants (18/20) preferred a mix of AI and manual tools. Typically, they began with AI-assisted mode to create a "*blueprint layout*" (P5), followed by "*manually tuning the scene as needed*" (P6). AI features were praised for enhancing creativity (20/20), flexibility (16/20), and expressiveness (3/20), though some found them "*too creative*" (P5), leading to unexpected or undesired results (7/20). Others noted subpar asset quality (7/20) and slow generation times (3/20).

Manual tools were valued for their control, precision, and sense of ownership (19/20), as well as ease-of-use (7/20). However, manual editing was also seen as time-consuming and laborious (12/20), requiring "*physically moving and pressing many buttons*" (P15). Some participants found selecting models from a preset list creatively limiting (4/20), while others struggled with tapping accuracy in busy environments due to "*fat finger*" issues (3/20).

Two participants diverged from this hybrid workflow: P2 skipped manual mode entirely, describing AI outputs as "*fun and creative, even when inaccurate*" and arrangements "*correct enough*". P19 avoided AI tools altogether due to slow generation times. Yet when asked how they would ideally use ImaginateAR once AI and manual modes improved, all 20 participants indicated they would prefer a mix of both. As P4 put it: "*AI helped me be more creative and quickly place objects. But even when it was right, I still wanted to tweak things manually. It felt more rewarding when I had the final say.*"

Similar to Part 1, participants preferred the freedom to use AI and manual tools as needed. For brainstorming, however, participants relied solely on the AI agent. Eleven found it helpful, particularly when stuck or unsure what to create next. They especially appreciated how the agent suggested ideas aligned with their environment or theme. P20, for instance, began with a vague Sci-Fi idea and found the AI helpful in "*refining my idea into something more specific and creative*", which led to creating an alien and a robot. Still, several participants (7/20) wished the agent could do more—holding a back-and-forth conversation (5/20), asking clarifying questions (4/20), and eventually generating an entire scene once the idea was fully formed (6/20). As P7 reflected, "*The AI adds flexibility, but also demands that you know exactly what you want and how to describe it*," pointing to the potential for more collaborative, guided brainstorming and authoring workflows.

## 7.3 Brainstorming Future of ImaginateAR

In Part 3, participants shared ideas for improving ImaginateAR and envisioned how they might use it in the future. Below, we synthesize limitations they identified and their proposed enhancements.

**AI Creativity.** While participants agreed the AI was generally more creative than they were, they differed on whether that creativity was actually beneficial. 13 participants appreciated the AI's inventive and surprising results—P2 remarked, "*It gave me a humanoid lion and a two-headed giraffe... I love the randomness of it. I'm just excited to see what it will create next!*" Others found the AI "*too creative*" (P5), generating content that clashed with their intent. For instance, P5 requested a fountain and received a pink one—possibly because previous objects they had generated were pink—when they had envisioned a typical stone fountain: "*Creativity can be a double-edged sword.*" P1 also raised concerns that an

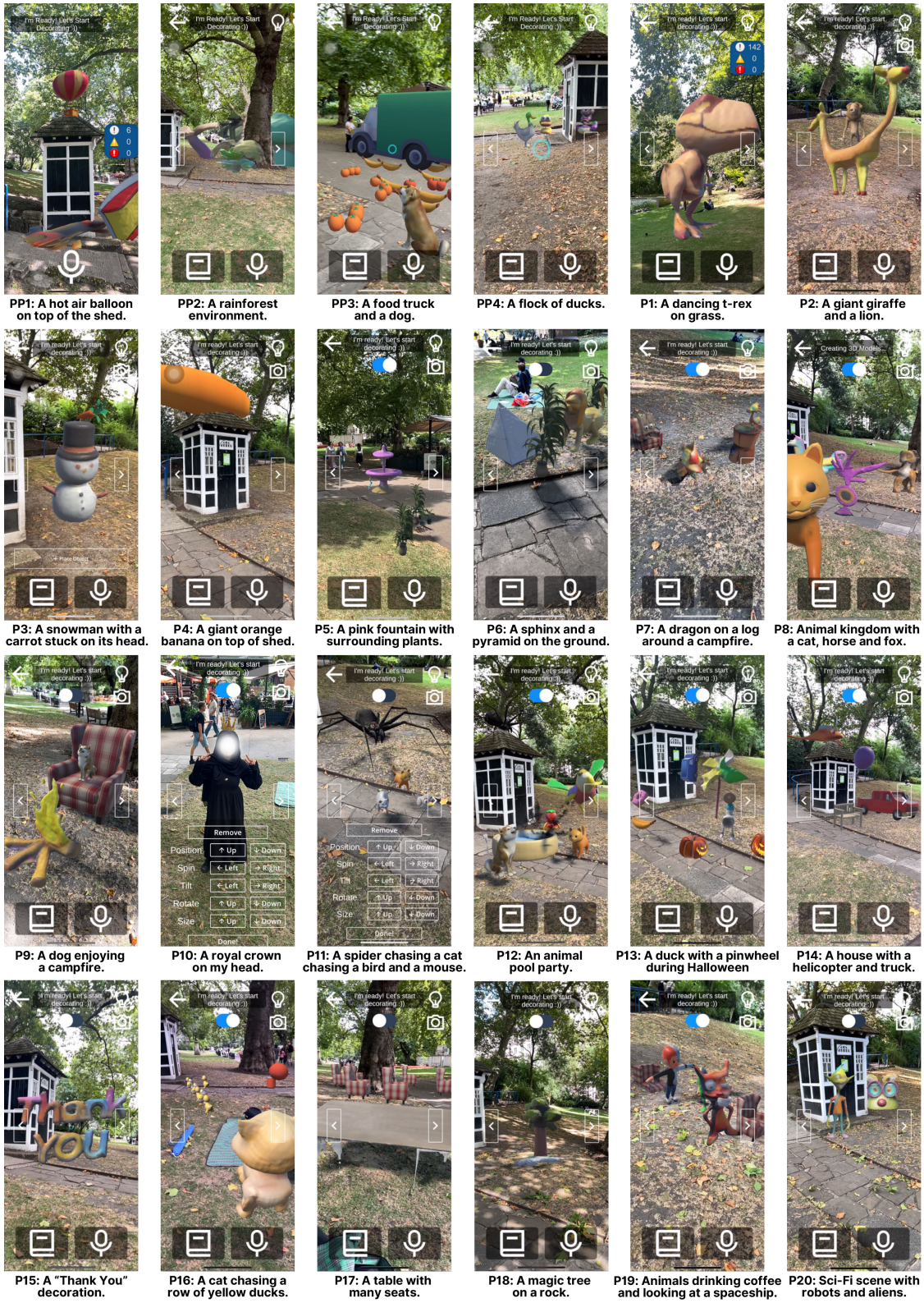



**Figure 11: AR experiences created by participants ($N_{pilot}$=4; $N_{study}$=20) while interacting with the full ImaginateAR prototype in Part 2. Users were encouraged to create freely without limitations. 'PP' denotes pilot participants and 'P' study participants.**

unmoderated AI could produce inaccurate or even inappropriate content, especially for children.

To manage AI creativity, participants proposed several ideas. P14 wanted the AI to clarify ambiguous requests through follow-up questions, rather than making assumptions: "*If I ask for a creature but don't specify the color, the AI should ask, 'do you want it yellow, purple, or something else?' We should talk back and forth until both of us are ready to build something.*" P18 suggested a "*creativity slider*" for more granular control over AI outputs. Participants also appreciated being able to choose from multiple AI-generated options, helping them "*ignore results that don't fit*" (P8).

**Creating Dynamic AR Scenes.** Many participants wanted their scenes to feel alive and reactive, not just static. They suggested adding animations (8/20), music (5/20), and event triggers (3/20). For example, P5 wanted a water fountain with flowing water, P17 imagined dogs running in circles, and P20 envisioned horror scenes with eerie sounds: "*That would make it more realistic, especially if the rendering quality is more like a cartoon.*" Additionally, P8 hoped virtual creatures could respond to touch (*e.g.,* a dog smiling when petted), while P14 suggested NPC-like interactions where virtual humans or animals could talk or bark back in a conversational manner. Still, P18 felt the current features "*cover the basics needed to create a simple AR scene*," but hoped future improvements would focus on AI generation quality and speed.

**Sharing Creations.** 11 participants expressed interest in sharing their AR creations. Some preferred sharing photos or videos (P9, P12, P16), while others (P6, P7, P20) wanted to distribute full AR scenes for others to download and experience. P5, P7, and P19 proposed a searchable catalog of AI-generated models with user ratings: "*If I had a catalog, I could just type in 'pink dolphin' and see what others have used. That would drive inspiration and save me time*" (P5). P7 added that ratings could help users assess model quality before choosing. To further personalize shared assets, P3, P6, and P16 suggested allowing users to customize elements like color. Finally, P1 emphasized that public sharing could help enforce content safety and appropriateness.

**AI Explainability.** Nine participants wanted clearer explanations from the AI about its progress and actions. Currently implemented messages like "*Understanding Your Surroundings*" and "*Creating 3D Models*" were seen as too vague. As P4 explained, "*Instead of just 'thinking' or 'processing,' a more detailed explanation of what's been done would be nice, just so I know the AI heard me right, how much longer I have to wait, and what it will eventually do to my environment.*" Participants also wanted better feedback during AI processing to know whether they could continue interacting, such as looking around or making manual changes. That said, P18 cautioned against overloading users with information, suggesting that even a brief log would help: "*Long messages will go unread. Just tell me what the AI heard and what it's doing.*"

**Access Barriers.** Participants also raised accessibility concerns regarding speech input. P14 and P18 noted misrecognition of non-standard accents (*e.g.,* "*bowl*" interpreted as "*ball*"), while P5 highlighted issues for users with speech impairments or in noisy environments: "*If kids are screaming in the background, it might be easier not to speak out loud.*" While speech input was chosen for its naturalness, participants emphasized the importance of offering alternatives to ensure broader accessibility.

**Envisioning Future Use Cases.** When asked where and how they might use ImaginateAR in the future, participants proposed a wide range of scenarios. Popular ideas included designing mini or board games (P9, P13, P14, P15) and transforming mundane environments—such as turning lecture halls into botanical gardens or adding a beach to an office (P5, P7, P8). Some envisioned practical uses like visualizing furniture layouts (P2, P17) or using AR pets for stress relief (P1, P3). Others imagined playful experiences, such as hiding AR Easter eggs for friends (P4, P11) or creating immersive horror games (P7, P20). P10 even envisioned placing themselves inside the scene: "*I want to wear a crown, sit on a throne in the middle of a desert, and be surrounded by flowers.*" Overall, participants were excited to use ImaginateAR anywhere—from their homes (P2, P5) to parks (P5) and outdoor landmarks (P19).

## 8 Discussion

ImaginateAR combines outdoor scene understanding, fast 3D asset generation, and LLM-driven speech interactions to advance AI-assisted AR authoring. Our study revealed that users often began with AI to generate a creative scene blueprint, then refined it manually for greater control—enabling diverse, accurate, and expressive creations. Here, we provide suggestions for AI-assisted AR authoring tool designs, discuss the broader implications of AI creativity and assistance, and outline limitations and future directions.

### 8.1 Design Implications for AR Authoring Tools

Throughout the study, participants indicated preferences for AI use and proposed a wide range of improvements and future features for ImaginateAR. We summarize and expand on these suggestions.

**What Role Should AI Play in AR Authoring Workflows?** Our key takeaway is that users expect a blend of AI-assisted and manual tools when authoring AR environments—they want to co-create with AI, not just rely on it. While AI offers creativity and expressivity, manual tools provide the control needed to fine-tune scenes and feel ownership over the result. All but two participants combined both during free-form authoring: they reviewed AI-generated blueprints, then refined one to better match their creative intent. AI sped up early prototyping, helping users bring ideas to life with less active input and decision-making, while manual adjustments enabled greater precision and reduced frustration by offering a way to correct AI errors. We recommend that future iterations of ImaginateAR continue supporting hybrid workflows, consistent with human-AI design guidelines [5, 42]. Ultimately, users seek outcomes that justify their effort—AR scenes that best reflect their imagination—which often requires both the creative freedom of generative AI and the precision of manual control.

**How Much AI Creativity is Too Much?** AI's creativity can be a double-edged sword—both engaging and frustrating. Some participants enjoyed the AI's playful interpretations—like P2's whimsical two-headed giraffe—while others felt such outputs strayed too far from their intent. This tension suggests ImaginateAR should avoid extremes: being too rigid, where the AI follows only literal instructions, or too free, where it produces imaginative but irrelevant content. Following the *Human-Centered Artificial Intelligence (HCAI)* framework [94], we recommend giving users ways to adjust

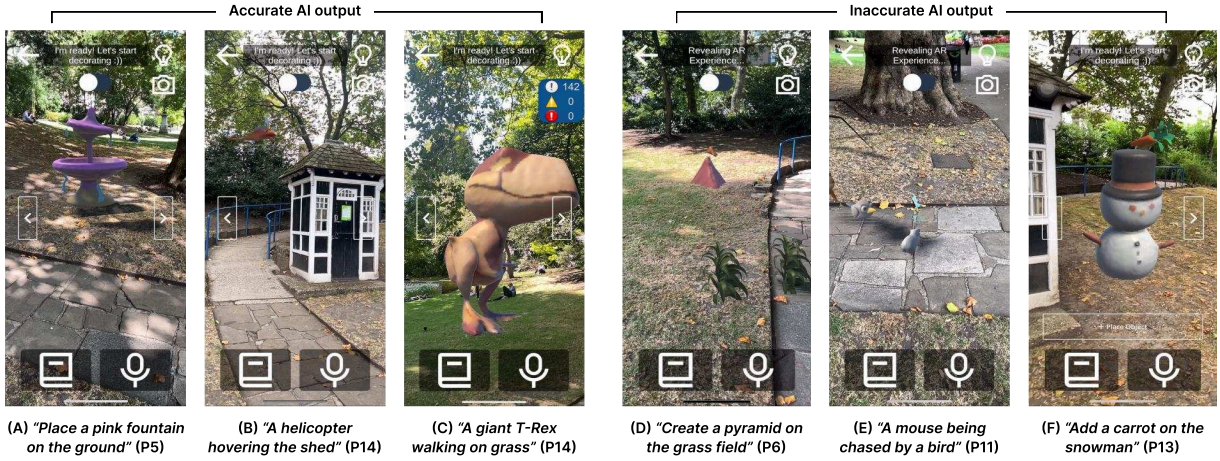



**Figure 12: Examples of accurate and inaccurate AI-generated scene blueprints before any manual input. (A–C) show scenes that align with user intent across object type, placement, and orientation, though users may have made later edits. (D–F) illustrate common issues: clipped geometry (D), incorrect facing direction (E), and imprecise part-level placement (F).**

AI creativity (*e.g.,* a "*creativity slider*" akin to an LLM's temperature setting) while supporting rapid iteration so users stay in control.

**What Might Future AR Authoring Look Like?** Our findings point to a future AR authoring workflow where users and AI co-create through iterative conversation, refining ideas together until both "agree" on what to build. Once aligned, the system could generate a full scene blueprint. For example, a user might say, "*Turn this playground into a coral reef*", imagining an experience where kids can explore and learn about marine life. The AI might suggest creative details like, "*Let's add a surgeonfish and a parrotfish, since they're commonly found in coral reefs*", or ask clarifying questions such as, "*What color corals would you like?*" rather than making assumptions. This kind of dialogue lets users guide the creative direction without needing to specify every object or detail—alleviating the burden of constant input and decision-making. The same workflow can extend beyond 3D assets to include music (*e.g.,* sea breeze), animations (*e.g.,* fish flapping their fins), and event triggers (*e.g.,* picking up corals that break). Once both parties feel ready, AI agents can build the scene. To support this, AR authoring tools need scene understanding to position, rotate, and scale multiple objects appropriately—reducing the user's workload of arranging each asset manually. Users could then make quick manual edits to fine-tune the result, and ideally, share it with others. While ImaginateAR already supports conversational brainstorming, real-time full-scene generation remains limited by current technology: even our fast asset generation pipeline takes 20–30 seconds per model, making scene-level creation too slow for interactive use. This vision also aligns with prior work like LLMR's Planner agent [26], which also supports collaborative scene ideation—but primarily targets VR and still struggles to generate complete scenes efficiently. However, as generative models continue to improve, conversational AR authoring at scale may soon be possible.

## 8.2 Challenges in AI-powered AR Authoring

This work contributes to both HCI and computer vision by integrating outdoor scene understanding and fast 3D asset generation into a simple, speech-driven system for AI-assisted AR authoring. However, our study revealed limitations that impacted user experience. For example, scene understanding sometimes lacked granularity, leading to visual misalignments, while asset generation, though significantly faster than prior work, still required around half a minute—affecting perceived usability. Below, we reflect on key technical challenges. We also dig deeper into CV–specific challenges in Sections 1–3 of the Supplementary Materials, including depth map enhancement, scene understanding, and 3D asset generation.

**Scene Understanding Accuracy and Granularity.** We represent real-world objects as 3D bounding boxes to keep the scene graph compact and make spatial reasoning easier for LLMs. However, this abstraction can limit precision in AI-generated scene blueprints. For example, a sloped ground in our study environment was enclosed in a tall bounding box. When users asked for virtual objects to be placed on this surface, aligning to the bounding box's maximum y-value caused them to float near the bottom of the slope, while using the minimum y-value led to clipping near the top. Placement on irregular, multi-part shapes like the pig statue in Figure 7A was also challenging. A hat worked reasonably well by aligning its base to the statue's bounding box top, but clothing—intended for the body—was harder to position due to the lack of part segmentation. Even with the hat, minor misalignments occurred because the statue's ears extended above its head, meaning the maximum y-value did not match the intended placement point. Figure 12 illustrates scenes created solely by AI, including both successful and unsuccessful examples. While 3D bounding boxes offer an efficient abstraction, future work should explore richer representations to support more precise interactions—such as directly leveraging depth maps [27] or point clouds [111]—though these formats are less readily compatible with LLM-based pipelines compared to textual scene graphs.

**Speed and Quality of Asset Generation.** In-situ AR authoring demands fast, high-quality 3D mesh creation. Although our pipeline generated assets faster than prior work with minimal sacrifice of quality, participants still found the 30 second wait disruptive. Asset quality was also occasionally lacking: some models were flat (princess in Figure 7B), incomplete (knight missing legs in Figure 7B), had holes (castle in Figure 7B), or lacked detail (Mayan person in Figure 7C). Interestingly, some errors were viewed positively—P2 described a two-headed giraffe as "*fun and exciting*"—but overall, more reliable asset quality is needed. At the time of development, we used InstantMesh [117], then state-of-the-art, but more capable models like *LATTE3D* [116], *TRELLIS* [115], and *Hunyuan3D* [124] are emerging. Since ImaginateAR is modular, these can be easily integrated.

**Beyond Static Scenes.** While ImaginateAR supports AR authoring across a wide range of static outdoor scenes, dynamic content remains an open challenge. Our scene understanding pipeline relies on pre-scans to reduce user burden and enable more complete spatial understanding. We use scene graphs for their compact, textual structure that LLMs can reason over—but they do not reflect real-time changes, such as a moved bench or a person walking through the scene. As a result, while we can place a hat on a statue, we cannot place it on a moving person. Authoring truly dynamic scenes would also require richer support for animation, sound, and interactivity. Audio could be integrated using generative models [98] (*e.g., AudioLDM* [61], *MusicLM* [4]), and triggered events could build on prior systems that generate code [20, 26, 31]. Animation, however, is particularly challenging: most prior work uses simple scripted motions [20, 26, 31] or assumes pre-rigged assets [44], which is incompatible with our use of generated 3D models. Auto-rigging remains unreliable, and low-quality animation risks breaking immersion. Therefore, we chose to study animation needs in AR authoring qualitatively (*e.g.,* P5: "*flowing water*"). Future work should explore how to incorporate dynamic changes and behaviors into AR authoring [97] to further enhance creative flexibility.

**System Latency.** Latency remains a core challenge for in-situ AR systems—users expect responsiveness and may find even sub-minute delays disruptive, especially outdoors. While ImaginateAR achieves significantly faster runtimes than prior systems (*i.e.,* 33.92 ± 5.83 seconds), current speeds can still interrupt the flow of in-situ authoring. Because true real-time performance remains difficult to achieve, future tools should offer meaningful feedback (*e.g.,* progress indicators, estimated wait times) and support multitasking—such as manually editing objects while waiting for AI responses. As generative models improve, latency will likely decrease, though offloading to remote servers may remain necessary given the limited computational power of today's AR devices.

## 8.3 Limitations & Future Directions

This work has several limitations. First, we did not support multi-user co-creation. Several participants expressed interest in sharing or building scenes together, suggesting opportunities to study collaborative AR authoring [74]. Second, our user study was limited to a single location. While our technical evaluation shows that ImaginateAR can generalize to diverse outdoor settings, future studies should explore a broader range of environments (and perhaps with other demographics, such as children). Third, while ImaginateAR currently runs on phones, future work could explore deploying it on AR headsets, which may enable new interactions but also raise challenges around social acceptability and physical comfort during extended public use. Fourth, as discussed earlier, improving scene understanding, asset quality, system latency, and support for dynamic scene authoring remains important. Future scene understanding pipelines should also be evaluated on larger outdoor datasets. Finally, although ImaginateAR depends on precomputed scene graphs, participants did not perform scanning themselves. While the system is designed to scale with existing large-scale point cloud datasets, future work could examine how users scan scenes and how systems might better support that process [108].

## 9 Conclusion

We present ImaginateAR, a novel system that advances AI-assisted AR authoring through outdoor scene understanding, fast 3D asset generation, and LLM-driven speech interactions. Our technical evaluation and user study show that users can create diverse AR scenes in different real-world settings. Challenges remain—including improving scene understanding granularity, asset quality, latency, and collaborative AI support—but this work takes a step toward making personalized AR authoring as simple as speaking your imagination.

## References


[1] Aarya and Flip Phillips. 2023. BlenderGPT. https://github.com/gd3kr/BlenderGPT.

[2] Adobe. 2024. Adobe Aero: Augmented reality. https://www.adobe.com/uk/products/aero.html.

[3] Setareh Aghel Manesh, Tianyi Zhang, Yuki Onishi, Kotaro Hara, Scott Bateman, Jiannan Li, and Anthony Tang. 2024. How People Prompt Generative AI to Create Interactive VR Scenes. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 2319–2340. https://doi.org/10.1145/3643834.3661547

[4] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. 2023. MusicLM: Generating Music From Text. arXiv:2301.11325 [cs.SD] https://arxiv.org/abs/2301.11325

[5] Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3290605.3300233

[6] Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 2019. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5664–5673.

[7] Narges Ashtari, Andrea Bunt, Joanna McGrenere, Michael Nebeling, and Parmit K. Chilana. 2020. Creating Augmented and Virtual Reality Applications: Current Practices, Challenges, and Opportunities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376722

[8] Advait Bhat, Saaket Agashe, Parth Oberoi, Niharika Mohile, Ravi Jangir, and Anirudha Joshi. 2023. Interacting with Next-Phrase Suggestions: How Suggestion Systems Aid and Influence the Cognitive Processes of Writing. In *Proceedings of the 28th International Conference on Intelligent User Interfaces* (Sydney, NSW, Australia) *(IUI '23)*. Association for Computing Machinery, New York, NY, USA, 436–452. https://doi.org/10.1145/3581641.3584060

[9] Avinash Bhat, Disha Shrivastava, and Jin L.C. Guo. 2024. Do LLMs Meet the Needs of Software Tutorial Writers? Opportunities and Design Implications. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference* (Copenhagen, Denmark) *(DIS '24)*. Association for Computing Machinery, New York, NY, USA, 1760–1773. https://doi.org/10.1145/3643834.3660692

[10] Mark Billinghurst, Adrian Clark, Gun Lee, et al. 2015. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction* 8, 2-3 (2015),

73–272.
[11] Blender. 2024. Blender 4.3. A stroke of genius. https://www.blender.org.
[12] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. 2024. SF3D: Stable Fast 3D Mesh Reconstruction with UV-unwrapping and Illumination Disentanglement. *arXiv preprint arXiv:2408.00653* (2024).
[13] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa arXiv:https://www.tandfonline.com/doi/pdf/10.1191/1478088706qp063oa
[14] Virginia Braun and Victoria Clarke. 2019. Reflecting on reflexive thematic analysis. *Qualitative Research in Sport, Exercise and Health* 11, 4 (2019), 589–597. https://doi.org/10.1080/2159676X.2019.1628806 arXiv:https://doi.org/10.1080/2159676X.2019.1628806
[15] J Brooke. 1996. SUS: A quick and dirty usability scale. *Usability Evaluation in INdustry/Taylor and Francis* (1996).
[16] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics* 32, 6 (2016), 1309–1332.
[17] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11621–11631.
[18] Eva Cetinic and James She. 2022. Understanding and Creating Art with AI: Review and Outlook. *ACM Trans. Multimedia Comput. Commun. Appl.* 18, 2, Article 66 (feb 2022), 22 pages. https://doi.org/10.1145/3475799
[19] Anpei Chen, Haofei Xu, Stefano Esposito, Siyu Tang, and Andreas Geiger. 2024. LaRa: Efficient Large-Baseline Radiance Fields. In *European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, Heidelberg.
[20] Jiangong Chen, Xiaoyi Wu, Tian Lan, and Bin Li. 2025. LLMER: Crafting Interactive Extended Reality Worlds with JSON Data Generated by Large Language Models. arXiv:2502.02441 [cs.MM] https://arxiv.org/abs/2502.02441
[21] Lianggangxu Chen, Xuejiao Wang, Jiale Lu, Shaohui Lin, Changbo Wang, and Gaoqi He. 2024. CLIP-Driven Open-Vocabulary 3D Scene Graph Generation via Cross-Modality Contrastive Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 27863–27873.
[22] Yifei Cheng, Yukang Yan, Xin Yi, Yuanchun Shi, and David Lindlbauer. 2021. SemanticAdapt: Optimization-based Adaptation of Mixed Reality Layouts Leveraging Virtual-Physical Semantic Connections. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 282–297. https://doi.org/10.1145/3472749.3474750
[23] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (jun 2014), 25 pages. https://doi.org/10.1145/2617588
[24] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 5828–5839.
[25] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. 2017. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)* 36, 4 (2017), 1.
[26] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. LLMR: Real-time Prompting of Interactive Worlds using Large Language Models. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 600, 22 pages. https://doi.org/10.1145/3613904.3642579
[27] Ruofei Du, Eric Turner, Maksym Dzitsiuk, Luca Prasso, Ivo Duarte, Jason Dourgarian, Joao Afonso, Jose Pascoal, Josh Gladstone, Nuno Cruces, Shahram Izadi, Adarsh Kowdle, Konstantine Tsotsos, and David Kim. 2020. DepthLab: Real-time 3D Interaction with Depth Maps for Mobile Augmented Reality. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '20)*. Association for Computing Machinery, New York, NY, USA, 829–843. https://doi.org/10.1145/3379337.3415881
[28] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. 2017. Exploring spatial context for 3D semantic segmentation of point clouds. In *Proceedings of the IEEE international conference on computer vision workshops*. IEEE Computer Society, Los Alamitos, CA, USA, 716–724.
[29] Cathy Mengying Fang, Krzysztof Zieliński, Pattie Maes, Joe Paradiso, Bruce Blumberg, and Mikkel Baun Kjærgaard. 2024. Enabling Waypoint Generation for Collaborative Robots using LLMs and Mixed Reality. *arXiv preprint arXiv:2403.09308* (2024).
[30] Andreas Geiger, Philip Lenz, and Raquel Urtasun. 2012. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.
[31] Daniele Giunchi, Nels Numan, Elia Gatti, and Anthony Steed. 2024. DreamCodeVR: Towards Democratizing Behavior Design in Virtual Reality with Speech-Driven Programming. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*. IEEE Computer Society, Los Alamitos, CA, USA, 579–589. https://doi.org/10.1109/VR58804.2024.00078
[32] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.
[33] Google. 2024. ARCore. https://developers.google.com/ar.
[34] Google. 2024. Build global-scale, immersive, location-based AR experiences with the ARCore Geospatial API. https://developers.google.com/ar/develop/geospatial.
[35] Google Maps. 2024. Create and publish your own Street View imagery. https://www.google.com/streetview/contribute/
[36] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. 2024. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, IEEE, 5021–5028.
[37] Alicia Guo, Shreya Sathyanarayanan, Leijie Wang, Jeffrey Heer, and Amy Zhang. 2024. From pen to prompt: How creative writers integrate AI into their writing practice. *arXiv preprint arXiv:2411.03137* (2024).
[38] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.
[39] Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. 2023. $T^3$Bench: Benchmarking Current Progress in Text-to-3D Generation. arXiv:2310.02977 [cs.CV]
[40] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.
[41] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2023. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400* (2023).
[42] Eric Horvitz. 1999. Principles of mixed-initiative user interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Pittsburgh, Pennsylvania, USA) *(CHI '99)*. Association for Computing Machinery, New York, NY, USA, 159–166. https://doi.org/10.1145/302979.303030
[43] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. 2024. SceneCraft: An LLM Agent for Synthesizing 3D Scenes as Blender Code. In *Forty-first International Conference on Machine Learning*. https://openreview.net/forum?id=gAyzjHw2ml
[44] Han Huang, Fernanda De La Torre, Cathy Mengying Fang, Andrzej Banburski-Fahey, Judith Amores, and Jaron Lanier. 2024. Real-time Animation Generation and Control on Rigged Models via Large Language Models. arXiv:2310.17838 [cs.GR] https://arxiv.org/abs/2310.17838
[45] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2024. An Embodied Generalist Agent in 3D World. In *Proceedings of the International Conference on Machine Learning (ICML)*.
[46] IKEA. 2024. IKEA Place app launched to help people virtually place furniture at home. https://www.ikea.com/global/en/newsroom/innovation/ikea-launches-ikea-place-a-new-app-that-allows-people-to-virtually-place-furniture-in-their-home-170912/.
[47] Niantic Inc. 2024. Lightship ARDK. https://lightship.dev/products/ardk.
[48] Niantic Inc. 2024. Lightship VPS. https://lightship.dev/products/vps.
[49] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. 2022. Zero-shot text-guided object generation with dream fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 867–876.
[50] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. 2023. ConceptFusion: Open-set Multimodal 3D Mapping. *Robotics: Science and Systems (RSS)* (2023).
[51] Cinthya Jauregui, Tiffany T. Nguyen, Sarah Hazel Sallee, Mohan Raj Chandrasekar, Liam A'Hearn, Dominic Jonathan Woetzel, Pinak Paliwal, Shea MacDonald, Madison Nguyen, Lee M. Panich, Danielle M. Heitmuller, Amy Lueck, and Kai Lukoff. 2024. We Are Still Here: The Thámien Ohlone Augmented Reality Tour. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 647, 3 pages. https://doi.org/10.1145/3613905.3649127
[52] Terence Karran. 2005. Pan-European Grading Scales: Lessons from National Systems and the ECTS. *Higher Education in Europe* 30, 1 (2005), 5–22. https://doi.org/10.1080/03797720500087949 arXiv:https://doi.org/10.1080/03797720500087949

[53] Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 4401–4410.

[54] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. 2023. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, Los Alamitos, CA, USA, 4015–4026.

[55] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. 2024. Open3dsg: Open-vocabulary 3d scene graphs from point clouds with queryable objects and open-set relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14183–14193.

[56] LEGO. 2024. New App brings LEGO® bricks to life. https://www.lego.com/en-us/aboutus/news/2019/october/lego-ar-studio.

[57] Germán Leiva, Jens Emil Grønbæk, Clemens Nylandsted Klokmose, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2021. Rapido: Prototyping Interactive AR Experiences through Programming by Demonstration. In *The 34th Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) *(UIST '21)*. Association for Computing Machinery, New York, NY, USA, 626–637. https://doi.org/10.1145/3472749.3474774

[58] Germán Leiva, Cuong Nguyen, Rubaiat Habib Kazi, and Paul Asente. 2020. Pronto: Rapid Augmented Reality Video Prototyping Using Sketches and Enaction. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. https://doi.org/10.1145/3313831.3376160

[59] James R. Lewis, Brian S. Utesch, and Deborah E. Maher. 2013. UMUX-LITE: when there's no time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2099–2102. https://doi.org/10.1145/2470654.2481287

[60] David Lindlbauer, Anna Maria Feit, and Otmar Hilliges. 2019. Context-Aware Online Adaptation of Mixed Reality Interfaces. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology* (New Orleans, LA, USA) *(UIST '19)*. Association for Computing Machinery, New York, NY, USA, 147–160. https://doi.org/10.1145/3332165.3347945

[61] Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D. Plumbley. 2023. AudioLDM: Text-to-Audio Generation with Latent Diffusion Models. arXiv:2301.12503 [cs.SD] https://arxiv.org/abs/2301.12503

[62] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. 2024. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems* 36 (2024).

[63] Xianzheng Ma, Yash Bhalgat, Brandon Smart, Shuai Chen, Xinghui Li, Jian Ding, Jindong Gu, Dave Zhenyu Chen, Songyou Peng, Jia-Wang Bian, et al. 2024. When LLMs step into the 3D World: A Survey and Meta-Analysis of 3D Tasks via Multi-modal Large Language Models. *arXiv preprint arXiv:2405.10255* (2024).

[64] Leland McInnes, John Healy, Steve Astels, et al. 2017. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* 2, 11 (2017), 205.

[65] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. 2024. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 4296–4304.

[66] Michael Nebeling. 2022. XR tools and where they are taking us: characterizing the evolving research on augmented, virtual, and mixed reality prototyping and development tools. *XRDS* 29, 1 (oct 2022), 32–38. https://doi.org/10.1145/3558192

[67] Richard A Newcombe and Andrew J Davison. 2010. Live dense reconstruction with a single moving camera. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 1498–1505.

[68] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. 2011. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*. Ieee, 127–136.

[69] Niantic. 2024. Catch Pokémon. Find your buddy! https://pokemongolive.com.

[70] Niantic. 2024. Meshing. https://lightship.dev/docs/ardk/features/meshing/

[71] Niantic. 2024. Share your world in 3D with Scaniverse 4. https://scaniverse.com

[72] Niantic. 2025. Niantic VPS. https://www.nianticspatial.com/products/visual-positioning-system.

[73] Michael I. Norton, Daniel Mochon, and Dan Ariely. 2012. The IKEA effect: When labor leads to love. *Journal of Consumer Psychology* 22, 3 (2012), 453–460. https://doi.org/10.1016/j.jcps.2011.08.002

[74] Nels Numan, Shwetha Rajaram, Balasaravanan Thoravi Kumaravel, Nicolai Marquardt, and Andrew D Wilson. 2024. SpaceBlender: Creating Context-Rich Collaborative Spaces Through Generative 3D Scene Blending. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 41, 25 pages. https://doi.org/10.1145/3654777.3676361

[75] OpenAI. 2024. Editing your images with DALL·E. https://help.openai.com/en/articles/9055440-editing-your-images-with-dall-e.

[76] OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

[77] OpenAI. 2024. Hello GPT-4o. https://openai.com/index/hello-gpt-4o/.

[78] OpenAI. 2024. Introducing Whisper. https://openai.com/index/whisper/.

[79] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. 2023. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 815–824.

[80] Polycam. 2024. Polycam: 3D scanning platform. https://poly.cam

[81] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[82] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. 2022. Highly Accurate Dichotomous Image Segmentation. In *ECCV*.

[83] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[84] Nikhil Raina, Guruprasad Somasundaram, Kang Zheng, Sagar Miglani, Steve Saarinen, Jeff Meissner, Mark Schwesinger, Luis Pesqueira, Ishita Prasad, Edward Miller, Prince Gupta, Mingfei Yan, Richard Newcombe, Carl Ren, and Omkar M Parkhi. 2023. EgoBlur: Responsible Innovation in Aria. arXiv:2308.13093 [cs.CV]

[85] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[86] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*. Pmlr, JMLR.org, New York, NY, USA, 8821–8831.

[87] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 10684–10695.

[88] Antoni Rosinol, Arjun Gupta, Marcus Abate, Jingnan Shi, and Luca Carlone. 2020. 3D dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289* (2020).

[89] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.

[90] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042* (2023).

[91] Mohamed Sayed, John Gibson, Jamie Watson, Victor Prisacariu, Michael Firman, and Clément Godard. 2022. SimpleRecon: 3D Reconstruction Without 3D Convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, Heidelberg.

[92] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. 2023. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 8216–8223.

[93] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2023. MVDream: Multi-view Diffusion for 3D Generation. *arXiv:2308.16512* (2023).

[94] Ben Shneiderman. 2020. Human-Centered Artificial Intelligence: Reliable, Safe & Trustworthy. *International Journal of Human–Computer Interaction* 36, 6 (2020), 495–504. https://doi.org/10.1080/10447318.2020.1741118 arXiv:https://doi.org/10.1080/10447318.2020.1741118

[95] Rafael M.L. Silva, Ana Maria Cardenas Gasca, Joshua A Fisher, Erica Principe Cruz, Cinthya Jauregui, Amy Lueck, Fannie Liu, Andrés Monroy-Hernández, and Kai Lukoff. 2024. With or Without Permission: Site-Specific Augmented Reality for Social Justice. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 497, 7 pages. https://doi.org/10.1145/3613905.3636283

[96] SnapAR. 2024. Your Creativity Powered by Lens Studio. https://ar.snap.com/lens-studio.

[97] Sruti Srinidhi, Edward Lu, and Anthony Rowe. 2024. Xair: An xr platform that integrates large language models with the physical world. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 759–767.

[98] Xia Su, Jon E. Froehlich, Eunyee Koh, and Chang Xiao. 2024. SonifyAR: Context-Aware Sound Generation in Augmented Reality. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA)

*(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 128, 13 pages. https://doi.org/10.1145/3654777.3676406

[99] Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 2023. 3d-gpt: Procedural 3d modeling with large language models. *arXiv preprint arXiv:2310.12945* (2023).

[100] Lu Sun, Stone Tao, Junjie Hu, and Steven P. Dow. 2024. MetaWriter: Exploring the Potential and Perils of AI Writing Support in Scientific Peer Review. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW1, Article 94 (apr 2024), 32 pages. https://doi.org/10.1145/3637371

[101] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. 2023. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[102] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2023. DreamGaussian: Generative Gaussian Splatting for Efficient 3D Content Creation. *arXiv preprint arXiv:2309.16653* (2023).

[103] Torch. 2024. Torch. Tap the Power of 3D. https://torch-website.webflow.io/tour-the-app.

[104] Unity. 2024. ARFoundation. https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@5.1/manual/index.html.

[105] Unity. 2024. Go Create with Unity. https://unity.com.

[106] Unity. 2024. Unity Mars. https://unity.com/products/unity-mars.

[107] Cyrus Vachha, Yixiao Kang, Zach Dive, Ashwat Chidambaram, Anik Gupta, Eunice Jun, and Björn Hartmann. 2025. Dreamcrafter: Immersive Editing of 3D Radiance Fields Through Flexible, Generative Inputs and Outputs. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 3, 13 pages. https://doi.org/10.1145/3706598.3714312

[108] Jessica Van Brummelen, Liv Piper Urwin, Oliver James Johnston, Mohamed Sayed, and Gabriel Brostow. 2024. Don't Look Now: Audio/Haptic Guidance for 3D Scanning of Landmarks. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 415, 20 pages. https://doi.org/10.1145/3613904.3642271

[109] Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. 2020. Learning 3D Semantic Scene Graphs From 3D Indoor Reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA.

[110] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. 2023. ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[111] Zeyu Wang, Cuong Nguyen, Paul Asente, and Julie Dorsey. 2023. PointShopAR: Supporting Environmental Design Prototyping Using Point Cloud in Augmented Reality. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 34, 15 pages. https://doi.org/10.1145/3544548.3580776

[112] Guande Wu, Jing Qian, Sonia Castelo Quispe, Shaoyu Chen, João Rulff, and Claudio Silva. 2024. ARTiST: Automated Text Simplification for Task Guidance in Augmented Reality. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 939, 24 pages. https://doi.org/10.1145/3613904.3642772

[113] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. 2021. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 7515–7525.

[114] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. arXiv:2412.01506 [cs.CV] https://arxiv.org/abs/2412.01506

[115] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2025. Structured 3D Latents for Scalable and Versatile 3D Generation. arXiv:2412.01506 [cs.CV] https://arxiv.org/abs/2412.01506

[116] Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. 2024. Latte3d: Large-scale amortized text-to-enhanced3d synthesis. In *European Conference on Computer Vision*. Springer, 305–322.

[117] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

[118] Jianing Yang, Xuweiyi Chen, Shengyi Qian, Nikhil Madaan, Madhavan Iyengar, David F Fouhey, and Joyce Chai. 2024. Llm-grounder: Open-vocabulary 3d visual grounding with large language model as an agent. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, IEEE, 7694–7701.

[119] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything V2. *arXiv:2406.09414* (2024).

[120] Hui Ye, Kin Chung Kwan, Wanchao Su, and Hongbo Fu. 2020. ARAnimator: in-situ character animation in mobile AR with user-defined motion gestures. *ACM Trans. Graph.* 39, 4, Article 83 (aug 2020), 12 pages. https://doi.org/10.1145/3386569.3392404

[121] Lei Zhang, Jin Pan, Jacob Gettig, Steve Oney, and Anhong Guo. 2024. VRCopilot: Authoring 3D Layouts with Generative AI Models in VR. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) *(UIST '24)*. Association for Computing Machinery, New York, NY, USA, Article 96, 13 pages. https://doi.org/10.1145/3654777.3676451

[122] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. IEEE Computer Society, Los Alamitos, CA, USA, 3836–3847.

[123] Youcai Zhang, Xinyu Huang, Jinyu Ma, Zhaoyang Li, Zhaochuan Luo, Yanchun Xie, Yuzhuo Qin, Tong Luo, Yaqian Li, Shilong Liu, et al. 2023. Recognize Anything: A Strong Image Tagging Model. *arXiv preprint arXiv:2306.03514* (2023).

[124] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, Huiwen Shi, Sicong Liu, Junta Wu, Yihang Lian, Fan Yang, Ruining Tang, Zebin He, Xinzhou Wang, Jian Liu, Xuhui Zuo, Zhuo Chen, Biwen Lei, Haohan Weng, Jing Xu, Yiling Zhu, Xinhai Liu, Lixin Xu, Changrong Hu, Shaoxiong Yang, Song Zhang, Yang Liu, Tianyu Huang, Lifu Wang, Jihong Zhang, Meng Chen, Liang Dong, Yiwen Jia, Yulin Cai, Jiaao Yu, Yixuan Tang, Hao Zhang, Zheng Ye, Peng He, Runzhou Wu, Chao Zhang, Yonghao Tan, Jie Xiao, Yangyu Tao, Jianchen Zhu, Jinbao Xue, Kai Liu, Chongqing Zhao, Xinming Wu, Zhichao Hu, Lei Qin, Jianbing Peng, Zhan Li, Minghui Chen, Xipeng Zhang, Lin Niu, Paige Wang, Yingkai Wang, Haozhao Kuang, Zhongyi Fan, Xu Zheng, Weihao Zhuang, YingPing He, Tian Liu, Yong Yang, Di Wang, Yuhong Liu, Jie Jiang, Jingwei Huang, and Chunchao Guo. 2025. Hunyuan3D 2.0: Scaling Diffusion Models for High Resolution Textured 3D Assets Generation. arXiv:2501.12202 [cs.CV] https://arxiv.org/abs/2501.12202

# Supplementary Materials for *ImaginateAR: AI-Assisted In-Situ Authoring in Augmented Reality*

Summary of Supplementary Materials



## 1 Depth Map Enhancement Using Monocular Depth Estimator

Figure 1 shows an example. To overcome sensor inaccuracies (especially outdoors for objects farther than 6-10m), we scale a monocular estimator's output [13] using valid sensor depth points, yielding dense metric maps.

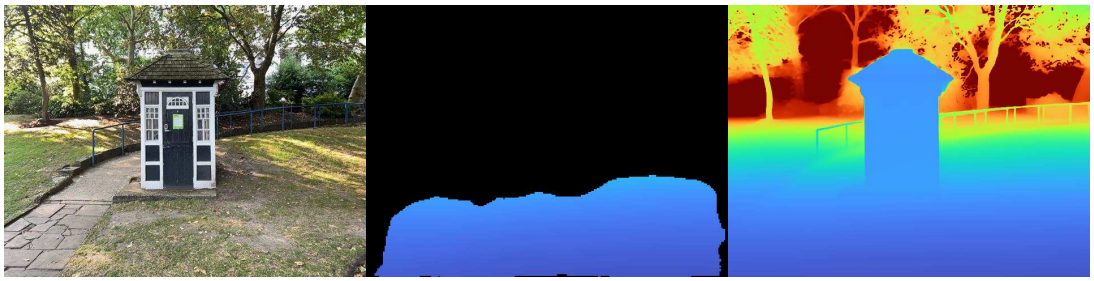


Fig. 1. Comparing inputs and output for monocular depth enhancement: The figure shows the source image (left), the sensor's original depth map (center), and the improved metric monocular depth map after enhancement (right). In the depth maps, red colors signify points farther from the viewer. The depth maps use a colormap where red represents farther points.

## 2 Scene Understanding Pipeline: Limitations and Future Works

The proposed scene understanding pipeline is effective in generating compact scene graph with semantic labels. However, we have identified three main issues that can be addressed in future work. First, results generated by our method are conditioned by the initial set of masks predicted by OpenMask3D [11]. This method, trained indoors [2],

Author's Contact Information:

detects masks in outdoor scenarios, but suffers from a domain gap. Nevertheless, this space is moving rapidly, and more robust methods were released after our user study. For instance, the method in [4], which has an example of scene understanding in an outdoor scenario, may be able to improve our results. A better initial set of masks is crucial to remove the preliminary filtering step: in fact, this filter acts without prior scene information and potentially removes or merges valid masks. Second, our clustering refinement is directly affected by the quality and the consistency of the labels predicted by the vision-language model (VLM). When these labels are incorrect, the results of the clustering step on top of the semantic point cloud can cause errors. For example, this may happen when the VLM ignores the prompt and tries to classify what has been covered by the mask, as shown in Figure 2. In this case, the misclassified bounding box—labeled as drinking fountain instead of bush—increases the chances of it being merged with the bounding box of the actual fountain, thereby generating a noisy bounding box for the fountain. In our experiments, we have noticed that the majority of the predicted labels are stable across different runs; however, a few of them might vary. This effect explains the slightly higher variation observed in the last row of the benchmark (see Table 2 of the main document). Validation checks could be enforced in future work to mitigate the problem. Third, our scene graphs are snapshots of the world at the time of scanning. This is generally not a problem because they mainly represent static objects (dynamic objects—such as parked cars—could potentially be removed using semantic labels), however we do not include real-time scene understanding components during the user experience. Future work can address this limitation by incorporating live components to further enhance the scene graphs.

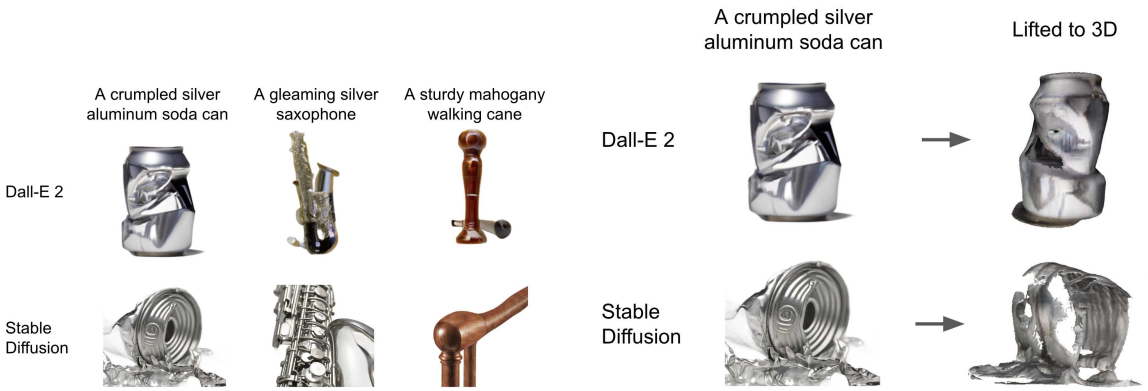
Fig. 2. Object and Context crops for a failure case. In this case, the VLM tried to *look through* the white mask and predicted drinking fountain instead of bush or vegetation.

## 3 AI generated 3D Assets

As reported in the main paper, the *Action Plan* Agent can invoke the generation of a virtual 3D asset on-the-fly.

We chose DALL-E2 [8] for its ability to generate images conditioned on input masks. Although its visual quality is inferior to other generative models, such as Stable Diffusion [9], the inpainting constraint often yields more complete objects on a plain background, which directly facilitates background removal [7] and image-to-3D reconstruction [12]. Figure 3 shows examples of assets generated by StableDiffusion Turbo [10] that are partially visible in the image. These assets can lead to ambiguous 3D models when lifted by InstantMesh [12].

Figure 4 compares the images generated by our pipeline for the same prompts with and without using prompt boosting. Using prompt boosting results in assets that are better suited for 3D reconstruction.




(a) Generated images. (b) Generated 3D assets

Fig. 3. Comparison between DALL-E2 [8] and StableDiffusion Turbo [10]. In these examples the inpainting strategy helps in generating fully visible objects. In contrast, StableDiffusion Turbo generates high-quality images of partially visible objects. When lifted in 3D, partially visible assets can lead to ambiguous or poorly defined 3D models. We used prompt-boosting for both the strategies.

## 4 Prompts

In this section, we report the prompts used by our AI agents.

Figure 5 presents an example of a prompt used by the *Object Classifier* agent. Instructions are given to the agent in GPT's *System mode* [6], while additional inputs—the object and context crops, as well as the list of previously predicted labels—are passed in GPT's *User mode* [6]. Finally, the agent replies with the semantic label of the object. It is worth noting that all inputs are generated by the system itself during execution, with no human intervention required.

Figure 6 shows an example of the prompt used by the *Prompt Boosting* agent. Again, we use *System mode* to instruct the agent about the task, and in *User mode*, we provide two visual examples (these examples are not request-specific; we use the same example in every request) and the initial user prompt to boost. The agent replies with the boosted prompt, which is then used to generate an image suited for image-to-3D methods.

Figure 7 illustrates the prompt used by the *Brainstorming* agent. This agent generates a short story-based description of an AR experience given a list of real and virtual objects in the scene, as well as conversations with the user.

Figure 8 reports the prompt used by the *Action Plan* agent. Based on the current state of the AR experience and a list of already-generated assets, the agent defines the action to apply to each existing and new virtual object to better align it with the user-described experience.

Finally, Figure 9 shows the prompt adopted by the *Assembly* agent. This agent receives the actions planned by the *Action Plan* agent (e.g., modifying an object's position) and applies them to virtual objects.

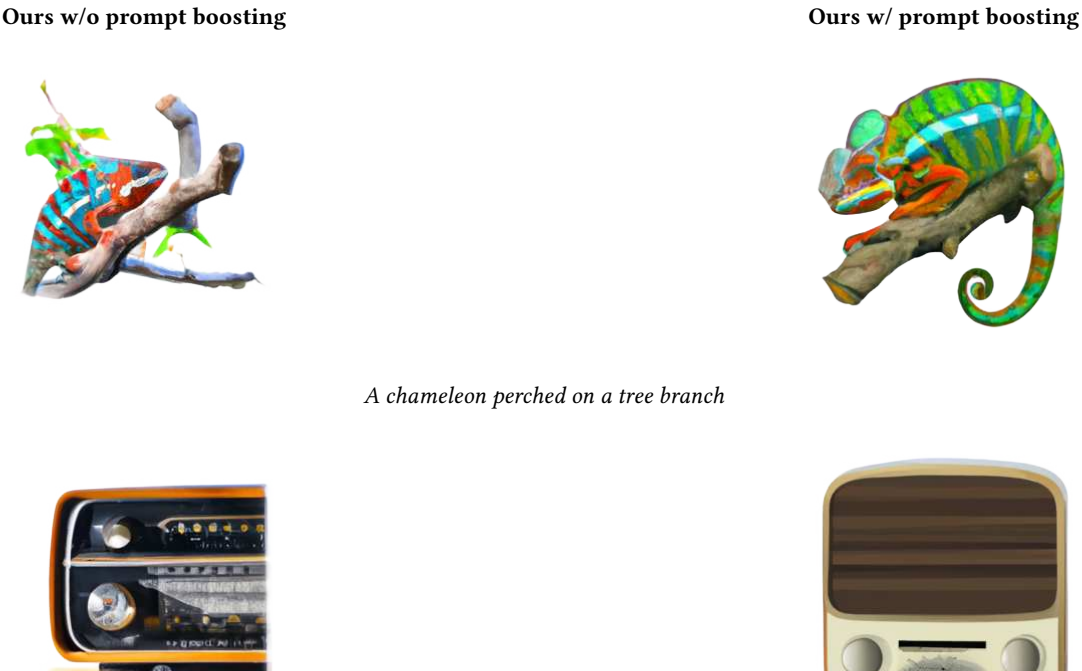



Fig. 4. A qualitative comparison of generated images: without (first column) and with (second column) prompt boosting. Prompt boosting results in objects that are more fully visible.

**Role: System**

We need your help to classify the object depicted in the image. The image contains two parts, divided by a black vertical section:
- on the left you can see the object of our interest. The object could be either things (such as a `vase`) or stuff (such as `vegetation`). You ****MUST IGNORE**** what is masked out with a white mask, because we don't care about it.
- on the right you can see the object plus some context. You can use this image to understand the object better, but the object of interest is in the left image.

We also provide the list of previously detected objects in the scene (comma separated). This list might be empty, meaning that no object has been detected yet. You can use this list to avoid repeating the same label, but you can also provide a new label if you think it's more appropriate to describe the object.

Your task is to provide the semantic label that better describes the visible object in the left image, ignoring the white mask. You have to consider the context of the object to provide a better answer: for example, if the object is not particularly relevant (e.g., the object is a ventilation grille hanged to the wall), you might want to provide a more general label (e.g., `wall`). However, remember to ignore the white mask.

**Role: User**

**Role: User**

Here the list of already detected objects: `path`, `rock`

**Role: User**

****DON'T LABEL**** what has been covered by the white mask in the left image.

**Response**

**class_name:** `grass`

Fig. 5. Example of the prompt used by the Object Classifier agent

**Role: System**

We want to generate a 3D mesh from an initial image. We use DALL-E 2 to generate the initial image starting from a textual prompt given by the user. Then we use a model that generates a 3D mesh starting from the image generated by DALL-E 2. For this reason, it is important that the generated image by DALL-E 2 is easy to segment and that the object of interest is clearly visible, complete and resembles a 3D asset.

We provide you two images, called **BAD** and **GOOD**, both generated by DALL-E 2 using the following prompt: *'a dragon that is fully visible and that looks at the camera.'*

Here why BAD and GOOD are different:
- **BAD** is a bad example because it does not show the object entirely and it looks flat, so it is not good for meshing.
- **GOOD** is a good example because it shows the object in a clear way, it looks a 3D asset and it is good for meshing.

We also provide you the prompt that contains the object to generate. We call this prompt **PROMPT**, and it can contain characteristics of the object. An example of possible **PROMPT** is: *`a cute puppy dragon`*

Your goal is to expand **PROMPT** returning the final prompt that we can use to generate a good image with DALL-E 2. Please only provide the prompt in your response.

Please consider these ****IMPORTANT**** points when you expand **PROMPT**:
- the object must be easy to segment. Difficult backgrounds or complex details might make the object hard to segment. Reflection or difficult lighting conditions are not good for meshing.
- the object must be easy to mesh. Flat, complex and intricate objects are not good for meshing. The object must be clearly visible.
- the object must be fully visible. Partially visible objects are not good for meshing.
- we only care about the object. The background is not important, as well as the context. For instance, if the object is a dragon, we don't care if the dragon is flying or standing on a rock. We only care about the dragon. Don't ask to generate additional objects or elements that are not part of the object.

**Role: User**

Here the **GOOD** example generated by DALL-E 2:

**Role: User**

Here the **BAD** example generated by DALL-E 2:

**Role: User**

Here **PROMPT**: a parrot

**Response**

a vibrant parrot sitting upright, fully visible with its colorful feathers clear against a plain white background, showcasing its distinct beak and eye features, with no distractions in the background.

Fig. 6. Example of the prompt used for *Prompt Boosting* when generating new assets.

**Role: System**

The user wants to create an AR experience. Your job is to help the user with brainstorming this AR experience.
Here is what the user said: **$userinput**
You will be given scene graphs in JSON format representing a 3D AR scene with real and virtual objects. Each object is defined by the 3D bounding box that encloses it (i.e., a Unity BoxCollider). The coordinate system is left-handed, Y-up coordinate system.
Here is scene graph JSON structure in more detail:

```
{“objects” :
  [     {
        "guid": "string. A unique ID of this object. Do not change this.",
        "class_name": "string. Name of the object.",
        "pos": {
           "x": "float. X position.",
           "y": "float. Y position.",
           "z": "float. Z position."
        },
        “rot”: {
           "x": "float. X rotation. Object appears right-side up with this rotation, but rotation still needs to be adjusted.“,
           "y": "float. Y rotation. Object appears right-side up with this rotation, but rotation still needs to be adjusted.”,
           "z": "float. Z rotation. Object appears right-side up with this rotation, but rotation still needs to be adjusted.”
        },
        "scale": {
           "x": "float. X scale.",
           "y": "float. Y scale.”,
           "z": "float. Z scale.”
        },
        "minBound": {
           "x": "float. Minimum x-coordinate when fitting a BoxCollider around the object.",
           "y": "float. Minimum y-coordinate when fitting a BoxCollider around the object.",
           "z": "float. Minimum z-coordinate when fitting a BoxCollider around the object."
        },
        "maxBound": {
           "x": "float. Maximum x-coordinate when fitting a BoxCollider around the object.",
           "y": "float. Maximum y-coordinate when fitting a BoxCollider around the object.",
           "z": "float. Maximum z-coordinate when fitting a BoxCollider around the object."
        },
        "onScreen": "boolean. Represents whether this object is currently visible on the user’s phone screen.“,
        "action": “string. No actions need to be done just yet, so this is likely set to ‘none’.”
     }
  ]
}
```

You will receive two JSONs:
Real World Scene Graph represents real-world objects such as trees in a park. This is to help you understand what is around the user so that you can tailor your brainstorming around the user’s context.
Here is the real world scene graph: **$realobjects**
Virtual Objects Scene Graph represents existing virtual objects. This may be empty, meaning there are currently no virtual objects in the AR scene. If not empty, then you should also consider virtual objects around the user when brainstorming an idea.
And here is the virtual objects scene graph: **$virtualobjects**
Finally, here is the result from a previous discussion between the user and a different AI agent. This may also be empty: **$context**.
Come up with a short, simple, and creative one sentence description of an AR experience based on the user’s request and contextual information. Use simple language. Make sure to not output anything else.

Fig. 7. Prompt used by the *Brainstorming* agent.

**Role: System**

The user wants to create an AR experience using two AI agents. You are the first AI agent. Your job is to create an action plan by assigning actions to virtual objects in the AR environment for the next two agents to execute.
Here is the user requested AR experience: **$userinput**.
You will receive three JSONs:
1. Real World Scene Graph: Represents real-world objects (e.g., trees in a park). Action should always be 'none'. No action should be performed on real-world objects. Use this JSON as a reference to understand the user's surroundings.
Real World Scene Graph Structure:

```
{
  "objects":[
    {
      "guid":"string. A unique ID of this object. Do not change this.",
      "class_name":"string. Name of the object.",
      "onScreen":"boolean. Represents whether this object is currently visible on the user's phone screen."
    }
  ]
}
```

2. Virtual Objects Scene Graph: Represents existing virtual objects. May be empty. Assign actions to each virtual object.
Virtual Objects Scene Graph Structure: Shares the same structure as the Real World Scene Graph.
3. Existing Assets JSON: Represents assets in the user's application database. Based on the user's request, you may pick objects to be created by the system.
Existing Assets JSON Structure:

```
{
  "objects":[
    {
      "class_name":"string. Name of the object.",
      "prefabLocation":"string. Should be one of two values: 'resources' or 'persistent'. 'resources' means the object is in a folder called 'resources' in the database. 'persistent' means the object is in a folder called 'persistent'."
    }
  ]
}
```

Here is your input:
Real World Scene Graph: **$realobjects**
Virtual Objects Scene Graph: **$virtualobjects**
Existing Assets JSON: **$existingassets**
Your output: Generate a modified Virtual Objects Scene Graph by assigning an "action" to each virtual object. You should determine which objects in the user's request are virtual, as we can only perform actions on virtual objects. The possible actions are:
- none: No change needed.
- remove: Remove from the AR scene.
- update: Modify an existing object's properties (e.g., position, rotation, scale).
- create_resources: Create an object. This object must have a "prefabLocation" of "resources" according to the Existing Assets JSON.
- create_persistent: Create an object. This object must have a "prefabLocation" of "persistent" according to the Existing Assets JSON.
- create_new: Generate a new object. This is if an asset does not exist in the user's database according to the Existing Assets JSON. Only at most one object in the output scene graph JSON can be marked as "create_new."
Output Scene Graph Structure:

```
{
  "objects":[
    {
      "guid":"string",
      "class_name":"string",
      "action":"string"
    }
  ]
}
```

For new objects, leave the guid field empty.
Make sure to not output anything else besides this scene graph JSON. Do not include any miscellaneous or trailing characters.

Fig. 8. Prompt used by the *Action Plan* agent.

**Role: System**

The user wants to create an AR experience using two AI agents. You are the second agent. The first AI agent created the necessary virtual objects and assigned action tasks for each virtual object. Your job is to execute these actions, which will involve changing the position, rotation, and/or scale of each virtual object so that it is placed naturally in the real world.
You will be given a scene graph in json format representing a 3D AR scene with real and virtual objects, where each object is defined by the 3D bounding box that encloses it. The coordinate system of the 3D world is a left-handed, Y-up coordinate system. I will help you understand the json format. Each object's size and position is given as an axis-aligned bounding box in xyz-coordinates. So for example 'min_x' represents the minimal value of the bounding box along the x-axis. Your goal is to put arrange the virtual objects in the scene, so that they are placed in sensible locations. Rotation of the object is given as a forward vector, which is the direction the object is facing. In the end, virtual objects should be positioned and rotated in a natural way and be sized appropriately in relation to other objects in the AR scene.
Here is the user requested AR experience: **$userinput**.

You will receive three JSONs:
1. Real World Scene Graph: Represents real-world objects (e.g., trees in a park). Action should always be 'none'. No action should be performed on real-world objects. Use this JSON as a reference to understand the user's surroundings.
Real World Scene Graph Structure:{

```
  "objects":[
    {
      "guid":"string. A unique ID of this object. Do not change this.",
      "class_name":"string. Name of the object.",
      "min_x":"float. minimal value of the bounding box along the x-axis",
      "max_x":"float. maximum value of the bounding box along the x-axis",
      "min_y":"float. minimal value of the bounding box along the y-axis",
      "max_y":"float. maximum value of the bounding box along the y-axis",
      "min_z":"float. minimal value of the bounding box along the z-axis",
      "max_z":"float. maximum value of the bounding box along the z-axis",
      "forward_x":"float. direction the object is facing along the x-axis",
      "forward_y":"float. direction the object is facing along the y-axis",
      "forward_z":"float. direction the object is facing along the z-axis",
      "onScreen":"boolean. represents whether this object is currently visible on the user's phone screen.",
      "action":"string. should be one of five values: 'none', 'remove', 'update', 'create_resources', or 'create_persistent'."
    }
  ]
}
```

2. Virtual Objects Scene Graph: Represents existing virtual objects. May be empty. Assign actions to each virtual object. Adjust bounds as necessary.
Virtual Objects Scene Graph Structure: Shares the same structure as the Real World Scene Graph.
Here is an explanation of each 'action':
- none: No change needed.
- remove: Remove from the AR scene. You should treat this object as not in the AR scene.
- update: Modify an existing object's properties (e.g., position, rotation, scale).
- create_resources: Create an object. This object has already been made by the previous AI agent. Your job is to change its position, rotation, and/or scale so that this object is placed in a sensible location.
- create_persistent: Create an object. This object has already been made by the previous AI agent. Your job is to change its position, rotation, and/or scale so that this object is placed in a sensible location.
- create_new: Generate a new object. This object has already been generated by the previous AI agent. Your job is to change its position, rotation, and/or scale so that this object is placed in a sensible location.
3. Player Scene Graph: Represents the player's current position and rotation. You should use this information to arrange virtual objects in a way that is visible to the player, if possible.
Here are your inputs:
Real World Scene Graph: **$realobjects**
Virtual Objects Scene Graph: **$virtualobjects**
Player Scene Graph: **$player**

Your output: Generate a modified Virtual Objects Scene Graph by rearranging the necessary virtual objects. The final AR scene should be realistic, sensible, and follow natural laws. Objects should never overlap. If virtual objects need to be placed close to other real or virtual objects, ensure that they maintain a reasonable minimum distance to avoid visual clutter or unnatural proximity. Objects should not be inside of other objects or grounds. Grounds, or objects at ground level, can be described in many ways, including dirt and grass. As an example, for a virtual object A to be on top of grass ground B, you should compare the 'min_y' of object A with the 'max_y' of object B and make sure 'min_y' of object A is at least greater than 'max_y' of object B. This way, the virtual object A is on top of grass ground B. Ensure that virtual objects respect real-world physics. For instance, objects should rest on flat surfaces (like the ground) instead of floating in mid-air unless intentional. Objects should also face natural directions. For example, when placing a virtual object next to a real world object, they should face similar directions so that the front of both objects are visible from the same player angle.
Do not change any virtual objects with action of 'none'. Also do not change any virtual objects that are irrelevant to the user's request.
If the user's request is ambiguous, you should not always just place virtual objects near objects with onScreen of true. Try to use the onScreen parameter only if it is necessary to fulfill the user's request.

Objects also should not be too big or too small relative to other objects in the AR scene. Ensure that virtual objects are proportionate to other real and virtual objects in the scene. For example, a virtual chair should be similar in size to any real chairs nearby. If there are no similar real objects, adjust the scale based on common sense.
Lastly, in cases where a virtual object cannot be placed naturally or you are unsure whether to place a virtual object, place it close to the player so that it is at least visible.
Ensure the guid, class_name, onScreen, and action parameters remain unchanged.
Make sure to not output anything else besides this modified Virtual Objects Scene Graph JSON. Do not include any miscellaneous or trailing characters.

Fig. 9. Prompt used by the *Assembly* agent.

## 5 Study Materials

This section contains references to the validated questionnaires we used, as well as custom questions and semi-structured interview questions we developed for the study.

### 5.1 Pre-Study Questionnaire

The pre-study questionnaire asked participants various demographics and experience questions, as shown in Listing 1.

Listing 1. The pre-study questionnaire.

```
1. What is your age?
[Number]

2. What gender do you self-identify with? (E.g. woman, non-binary, man, etc.)
[Short answer]

~ Prior Experience with Technology ~
3. How familiar are you with augmented reality (AR), such as Pokemon GO, Apple Vision Pro, Meta Quest,
      etc.?
[Options: Not at all familiar to Very familiar on a 5-point scale]

4. List any AR technologies and applications you have used before and what you used them for (or write
      "None").
[Long answer]

5. How often do you use augmented reality (AR) technologies or applications?
[Options: Multiple times each day to Never on a 7-point scale]

6. How familiar are you with artificial intelligence (AI) chat systems, such as chatbots, ChatGPT, etc
     .?
[Options: Not at all familiar to Very familiar on a 5-point scale]

7. List any AI chat systems you have used before and what you used them for (or write "None").
[Long answer]

8. How often do you use AI chat system(s)?
[Options: Multiple times each day to Never on a 7-point scale]

9. How familiar are you with 2D creativity tools, such as the Adobe suite (e.g. Photoshop, Premiere
     Pro), Figma, etc.?
[Options: Not at all familiar to Very familiar on a 5-point scale]

10. List any 2D creativity tools you have used before and what you used them for (or write "None").
[Long answer]

11. How often do you use 2D creativity tool(s)?
[Options: Multiple times each day to Never on a 7-point scale]

12. How familiar are you with 3D creativity tools, such as Maya, Blender, Unity, Unreal, etc.?
[Options: Not at all familiar to Very familiar on a 5-point scale]

13. List any 3D creativity tools you have used before and what you used them for (or write "None").
[Long answer]

14. How often do you use 3D creativity tool(s)?
```

```
[Options: Multiple times each day to Never on a 7-point scale]
```

### 5.2 Part 1: Comparison Task Questionnaires

There were three comparison task questionnaires in the first part of the study. We provided the first one to participants after every trial of the five features (e.g., brainstorming, modifying objects, etc.) for one of the three systems (A, B or C). This meant each participant completed this questionnaire 15 times (5 features by 3 systems). It is shown in Listing 2. This survey included custom questions about creativity, the UMUX-LITE [5] questionnaire to assess usability, and questions from the NASA-TLX [3] to assess task load.

The second comparison task questionnaire asked the participants which system they preferred overall and why, as shown in Listing 3. The final comparison task questionnaire provided participants with descriptions of how new features (e.g., adding music, animating objects, etc.) would work if they were added to System A, B or C, and asked which version participants would prefer and why. This is shown in Listing 4.

Listing 2. The first comparison task questionnaire, which participants completed after every trial of a new system feature.

```
1. Which feature did you just try?
[Options:
1 Brainstorming Ideas
2 Choosing / Creating 3D Virtual Object(s)
3 Object Placement
4 Object Selection & Modification
5 Object Selection & Remove Object
]

2. Which version did you use? (See app for the feature version)
[Options: A, B, C]

[If "1 Brainstorming Ideas" was chosen:]
3. In that brainstorming session, how do you feel about the end result, creatively speaking?
[Options: Not at all creative to Very creative on a 5-point scale]

[If "2 Choosing / Creating 3D Virtual Object(s)" was chosen:]
4. When you chose or created an object, how did you feel about the end result, creatively speaking?
[Options: Not at all creative to Very creative on a 5-point scale]

[UMUX-LITE: Both questions]

[NASA-TLX: The Mental Demand, Performance, Effort, and Frustration questions]
```

Listing 3. The second comparison task questionnaire, which participants completed after they finished all of the trials. This questionnaire compared the systems overall (as opposed to each feature).

```
1. Which system did you prefer? (Please choose one, if possible)
[Options:
System A: Manual (e.g. tap to place)
System B: AI system with multiple responses/options (e.g. select one of the AI's object placements)
System C: AI system with single response/option (e.g. AI system places an object)
]

2. Why did you prefer the above system? (Please write at least 1-2 full sentences.)
[Long answer]
```

Listing 4. The third comparison task questionnaire, which asked participants to brainstorm about new features that had not yet been added to the app.

```
1. Imagine you were adding music to your AR experience. How would you prefer to do this? (Please
    choose one, if possible)
[Options:
System A: Manually scroll through a list of music files and choose one.
System B: Ask the AI for music (e.g. "Can you add upbeat music?"), and it gives you 3 options.
System C: Ask the AI for music (e.g. "Can you add upbeat music?"), and it adds that song.
Other - Write in
]

2. Why did you prefer the above? Please write 1-2 full sentences.

3. Imagine you were adding sound effects to your AR experience. How would you prefer to do this? (
    Please choose one, if possible)
[Options:
System A: Choose an object (e.g. a dog), then tap "add sound effects", then manually scroll through a
    list of sound effect files, and choose one.
System B: Choose an object (e.g. a dog), then ask the AI for a sound effect (e.g. "Can you add a
    barking sound effect?"), and it gives you 3 options.
System C: Tell the AI to generate a sound effect with respect to an object (e.g. "Can you add a
    barking sound effect to this dog?"), and it adds that sound effect.
Other - Write in
]

4. Why did you prefer the above? Please write 1-2 full sentences.

5. Imagine you were animating objects in your AR experience. How would you prefer to do this? (Please
    choose one, if possible)
[Options:
System A: Choose an object, then tap "animate". Tap a start location and end location for the object's
     journey. Choose which type of animation (e.g. bounce, walk, run, etc.).
System B: Choose an object, then ask the AI system to generate an animation (e.g. "Make it walk to the
     dog bowl"), and it gives you 3 options proposing a path and walk style.
System C: Tell the AI system to generate an animation (e.g. "Make it walk to the dog bowl"), and it
    generates that animation.
Other - Write in
]

6. Why did you prefer the above? Please write 1-2 full sentences.

7. Imagine you were preparing an object to react to an event in your AR experience.
***An event*** can be triggered by a person pressing a virtual button, standing near a virtual object,
     making a noise or saying a trigger word, etc.
***A reaction*** to an event could be an object animating or changing colour.
How would you prefer to prepare an object to react to an event? (Please choose one, if possible)
[Options:
System A: Choose the reacting object, then tap "events". Choose an event from a dropdown list. Choose
    how the object should react from a list.
System B: Choose the reacting object, then tell the AI system what event should trigger what
    corresponding reaction. The AI gives you 3 options proposing various reactions.
System C: Tell the AI system that an event should trigger a reaction. The AI provides that reaction to
     that event.
Other - Write in
]
```

```
8. Why did you prefer the above? Please write 1-2 full sentences.

9. Imagine you were pinning a virtual object to a real one in your AR experience. E.g. pinning a
     virtual hat to a statue.
How would you prefer to do this? (Please choose one, if possible)
[Option:
System A: Choose the real object, e.g. the head of a statue. Tap "pin" and then choose a virtual
     object from a list, e.g. a hat. Adjust the virtual object's location/scale/orientation to fit the
      real object.
System B: Choose the real object, then tell the AI system what virtual object to pin. The AI gives you
      3 options proposing various objects and locations/scales/orientations.
System C: Tell the AI to pin a virtual object to a real object, and it pins it.
Other - Write in
]

10. Why did you prefer the above? Please write 1-2 full sentences.
[Long answer]

11. Any other comments?
[Long answer]
```

### 5.3 Part 2: Free-Form Authoring Task Questionnaire

After participants completed the free-form authoring task in Part 2 of the study, they completed the Creativity Support Index (CSI) questionnaire [1]. This can be found in Cherry and Latulipe's paper [1]. Note that we did not include the Collaboration factor, as our system is not collaborative.

### 5.4 Part 3: Semi-Structured Interview Questions

Listing 5 contains the main interview questions we asked participants. Since it was a semi-structured interview, we additionally asked follow-up questions about certain responses, e.g., for clarity.

Listing 5. The main interview questions we asked participants in Part 3 of the study.

```
~ Questions about the tool ~
1. What did you build and who did you build it for?

2. What features did you use to build it? Why?

3. Did any of the features you used while building stand out to you? Why?
  a. What was your favourite feature? Why?
  b. What was your least favourite feature? Why?

4. What were some of the pros/cons to using the AI agent vs manual process?

5. Was there anything that you found particularly enjoyable or rewarding?

6. Was there anything that you found particularly stressful?
  a. (E.g. social stressors)

7. Imagine that your job was to create AR experiences (e.g. for the audience you chose in Q1). Would
     you feel confident using this as an initial brainstorming/prototyping tool? Why?
```

```
~ Brainstorming questions ~
8. Imagine you could make this tool infinitely better. How might you improve it? What two features
     might you add?

9. Imagine you had this "better tool" and much more time, and you could build an AR experience in any
     location. What AR experience might you build and where?

~ Final question ~
10. Any other thoughts or comments?
```

## References


[1] Erin Cherry and Celine Latulipe. 2014. Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. *ACM Trans. Comput.-Hum. Interact.* 21, 4, Article 21 (jun 2014), 25 pages. https://doi.org/10.1145/2617588

[2] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. IEEE Computer Society, Los Alamitos, CA, USA, 5828–5839.

[3] Sandra G Hart. 2006. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, Vol. 50. Sage publications Sage CA: Los Angeles, CA, 904–908.

[4] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. 2024. Segment3D: Learning Fine-Grained Class-Agnostic 3D Segmentation without Manual Labels. In *European Conference on Computer Vision (ECCV)*. Springer-Verlag, Berlin, Heidelberg.

[5] James R. Lewis, Brian S. Utesch, and Deborah E. Maher. 2013. UMUX-LITE: when there's no time for the SUS. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 2099–2102. https://doi.org/10.1145/2470654.2481287

[6] Lucas Ostrowski. 2024. What is the difference between system, User, and Assistant roles in ChatGPT? https://lostrowski.pl/tips-news/what-is-the-difference-between-system-user-and-assistant-roles-in-chatgpt.

[7] Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. 2022. Highly Accurate Dichotomous Image Segmentation. In *ECCV*.

[8] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[9] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis With Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 10684–10695.

[10] Axel Sauer, Dominik Lorenz, Andreas Blattmann, and Robin Rombach. 2023. Adversarial diffusion distillation. *arXiv preprint arXiv:2311.17042* (2023).

[11] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. 2023. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[12] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

[13] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. 2024. Depth Anything V2. *arXiv:2406.09414* (2024).