ArK: Augmented Reality with Knowledge Interactive Emergent Ability

Qiuyuan Huang[‡*] Jae Sung Park[§‡*] Abhinav Gupta[†‡*] Paul Bennett[‡]

Ran Gong[♮] Subhojit Som[‡] Owais Khan Mohammed[‡] Baolin Peng[‡]

Chris Pal[†] Yejin Choi[§] Jianfeng Gao[‡]

[‡]Microsoft Research, Redmond [†] MILA [§]University of Washington [♮] UCLA

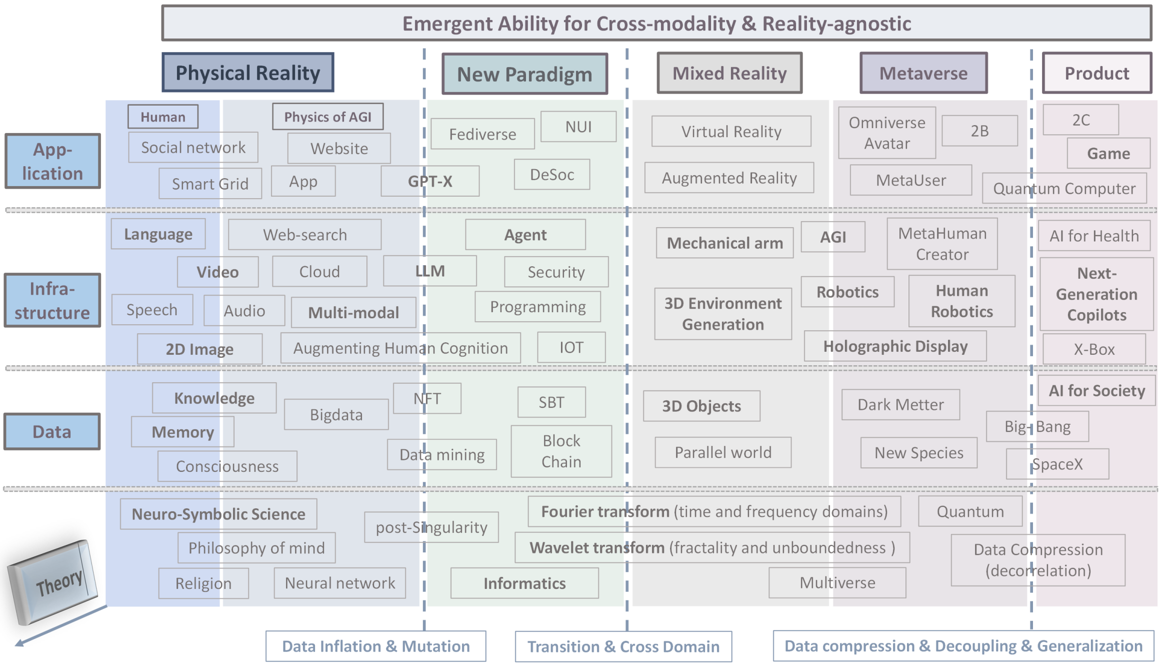


Figure 1: Emergent Ability for Cross-modality & Reality-agnostic observation. The pipeline shows the emerging capabilities of large foundation models for the cross-modality in a requested unseen environment, and loaded the accountability in reality-agnostic scenario automatically with a new paradigm. We present an AI dominating demonstration of a system that enables interactive generation and editing of a gaming/AR environment using a knowledge-enhanced style projection.

Abstract

Despite the growing adoption of mixed reality and interactive AI agents, it remains challenging for these systems to generate high-quality 2D/3D scenes in unseen environments. The common practice requires deploying an AI agent to collect large amounts of data for model training for every new task. This process is costly, or even impossible, for many domains. In this study, we develop an interactive agent that learns to transfer knowledge-memory from general foundation models (e.g., GPT4, DALL-E) to novel domains or scenarios for scene understanding and generation in the physical or virtual world. The heart of our approach is an emerging mechanism, dubbed ***Augmented Reality with Knowledge Inference Interaction (ArK)***, which leverages knowledge-memory to generate scenes in unseen physical world and virtual reality environments. The knowledge interactive emergent ability (Figure 1) is demonstrated as the observation learns *i) micro-action of **cross-modality**:* in multi-modality models to collect a large amount of relevant knowledge-memory data for each interaction task (e.g., unseen scene understanding) from the physical reality; and *ii) macro-behavior of **reality-agnostic**:* in mix-reality environments to improve interactions that tailor to different characterized roles, target variables, collaborative information, and so on. We validate the effectiveness of ArK on the scene generation and editing tasks. We show that our ArK approach, combined with large foundation models, significantly improves the quality of generated 2D/3D scenes, compared to baselines, demonstrating the potential benefit of incorporating ArK in

[*] Equal Contribution. Work done while Jae Sung and Abhinav were interning at Microsoft Research.

generative AI for applications such as metaverse and gaming simulation with emergent ability works invisible.

1 Introduction

There has been a growing amount of work on using large language models (LLMs) and large multimodality models (LMMs) to generate high-quality videos and images based on textual inputs (Saharia et al., 2022; Yu et al., 2022). Despite impressive results reported, it remains challenging for users (creators) to fully control the generation process and interactively edit the generated results if they don't meet users' intent. We envision a future AI system where a creator can interactively create a virtual reality scene that consists of objects that do or don't exist in real world, with the system responding faithfully by leveraging knowledge learned from training data of real-world tasks. For example, an interactive AI agent can incorporate contextual memory and background information, pertaining to a task, into the system by transferring knowledge encoded in pre-trained LLMs/LMMs and multi-sense information collected by sensors during the cause of preforming the task. LMMs and LLMs (foundation models) like DALLE-2 and ChatGPT have a superior capability to solve multimodality and natural language reasoning tasks. But we are not yet able to deploy them in many mission-critical real-world applications (e.g., Bing-search, business analysts, office users). Specifically, existing LLMs cannot always effectively transfer knowledge learned from training data to new mission-critical tasks (Peng et al., 2023). Nor can these models easily solve complex real-world tasks that require reasoning where human and AI agents often need to collaboratively break a complex task into simpler sub-tasks.

This study focuses on developing an interactive AI agent for scene understanding and generation, powered by pre-trained foundation models (e.g., DALLE-2, Chat-GPT). It is crucial for the AI agent to not only generate static scenes, but also predict the behaviors of various objects in the generated scene. To this end, the agent needs to retrieve and transfer the knowledge stored in the foundation model to the setting where the scene is being generated, interactively collect external multi-sense information (provided by human creators), and most importantly perform reasoning to synthesize the above two to generate or understand a scene. The reasoning capability of the agent is learned from relevant examples on-the-fly (in-context learning). Due to the length limit of input, we resort to a retriever to retrieve such examples on the fly via e.g., calling the APIs of the external knowledge bases that store such examples. We also need to access the repository of 2D/3D models which can generate 2D/3D objects in the scene. In addition, other knowledge bases, which store meta-information, descriptions, and use cases of these 2D/3D objects, are also useful.

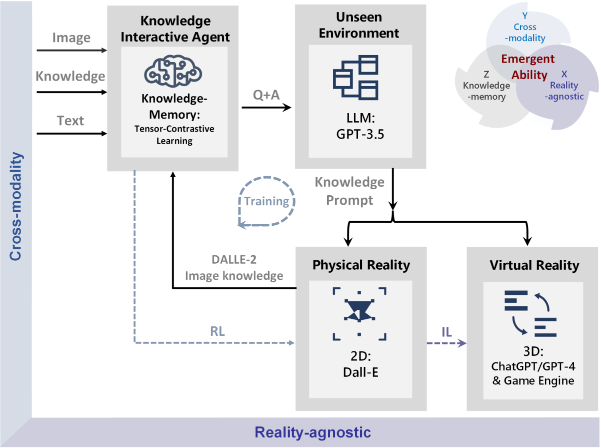


Figure 2: Example of ArK Interactive Emergence Mechanism using an external knowledge agent to identify text relevant to the image from a set of candidates. Our task involves leveraging visual and text knowledge retrieved from web and human-annotated knowledge samples to incorporate external knowledge about the world.

To facilitate human-AI interaction, we have developed a *knowledge-memory* agent, which uses an emerging mechanism, dubbed *Augmented Reality with Knowledge Inference Interaction (ArK)*, for generating and understanding scenes in virtual or real worlds. Specifically, for any particular scene generation or understanding task, the related world *knowledge* is retrieved from a pre-trained foundation model and transferred to the scene, and the *memory* module stores human-AI interactions from which the user intent, or the spec of the scene, can be decoded. Thus, the scene is generated or understood by reasoning over the knowledge and memory.

To demonstrate the effectiveness of ArK, we validate our AI agent on four interactive scene understanding and generation tasks: conversational 2D-image generation in physical world, conversational 3D-scene creating in virtual environment, conversational 3D-scene editing in mixed reality, and interactive gaming simulation scenario. Experiments show that ArK is effective in collecting

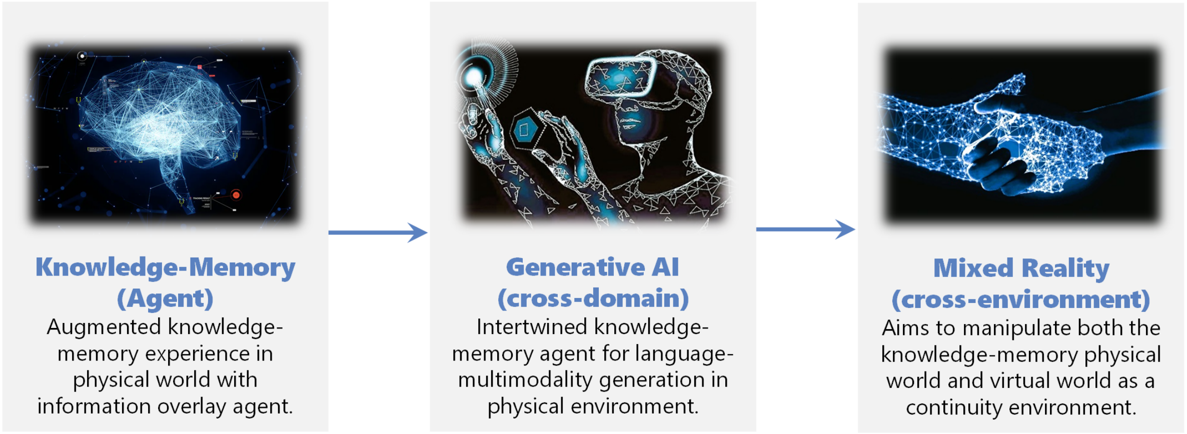

Knowledge-Memory (Agent)
Augmented knowledge-memory experience in physical world with information overlay agent.
Generative AI (cross-domain)
Intertwined knowledge-memory agent for language-multimodality generation in physical environment.
Mixed Reality (cross-environment)
Aims to manipulate both the knowledge-memory physical world and virtual world as a continuity environment.

Figure 3: Overview of the generative AI with the knowledge memory agent in the physical world and virtual environment.

and synthesizing knowledge and memory for scene understanding and generation in different settings, according to human evaluation.

Our contributions can be summarized below. (1) We have developed an interactive AI agent for scene understanding and generation in the physical world and virtual reality environments; (2) We show that the effectiveness of our agent is attributed to the proposed ArK mechanism that can understand and generate scenes in unseen settings by effectively synthesizing world knowledge encoded in foundation models (e.g., ChatGPT, DALLE-2), external knowledge retrieved from knowledge bases (e.g. wiki, Conceptnet), and contextual memory collected via human-AI interactions; and (3) we make the source code and models publicly available [*].

2 Related Work

Recently, text-prompted image generation models such as DALLE-2/GPT4, have been shown to generate images with higher fidelity and relevance to the text query than the image search results and can provide more personalized components (image editing, style change, variations of original image). While these systems can impressively map precise descriptions directly to image, they often fall short of understanding the creator's intention implied in description in an underspecified text query.

Emerging mechanism in LLM. Emergent abilities in large language models is one of its characteristic feature that cannot be predicted simply by extrapolating the performance of smaller models. There are several different types of emergent abilities that have been observed in LLMs. One type of emergent ability is the ability to generate creative text formats. For example, LLMs have been able to generate poems, code, and email (Wei et al., 2022). Another type of emergent ability is the ability to translate languages. LLMs have been able to translate text from one language to another with a high degree of accuracy. Finally, LLMs have also been able to perform different kinds of tasks, such as answering questions in a comprehensive and informative way.

The exact mechanisms by which LLMs develop these emergent abilities are not fully understood (Saharia et al., 2022; Yu et al., 2022). However, it is thought that when LLMs are trained on a large corpus of text, they are able to learn the patterns that exist in language. This allows them to generate text that is similar to human-generated text, and to translate languages with a high degree of accuracy.

Vision-Language transformer. Multi-modal representation learning is essential for joint vision-language tasks, such as image captioning, visual question answering, and visual commonsense reasoning. Large-scale architectures based on Transformers (Vaswani et al., 2017) have achieved impressive performance by pretraining representations for a wide range of natural language processing (NLP) tasks (Peters et al., 2018; Devlin et al., 2018; Yang et al., 2019; Liu et al., 2019; Radford et al., 2019). Recent work on vision-language pretraining (VLP) has shown that these large-scale pretraining methods can also be used for effective cross-modal representations (Lu et al., 2019; Tan and Bansal, 2019; Zhou et al., 2019; Chen et al., 2019; Alberti et al., 2019; Li

[*] https://augmented-reality-knowledge.github.io/

et al., 2020a, 2019, 2020b; Zhang et al., 2021; Kim et al., 2021). Most methods have two stages. First, the architecture is pretrained using a large set of image-text pairs. Then the model is finetuned on task-specific vision-language tasks. For example, Lu et al. (2019); Tan and Bansal (2019) propose multi-stream Transformer-based frameworks with co-attention to fuse these modalities. Zhou et al. (2019); Chen et al. (2019); Alberti et al. (2019); Li et al. (2020a, 2019, 2020b); Zhang et al. (2021) propose unified pretrained architectures to work on both visual-language understanding and visual-language generation tasks. Gardères et al. (2020) uses ConceptNet knowledge graph as is a knowledge base in order to facilitate commonsense vision-language question-answering. Kim et al. (2021) introduces a pretraining approach to learn self-attention representations directly on image patches. Although these models achieve impressive results on standard vision-language tasks, they do not use information from external knowledge graphs. Our proposed ArK architecture shows how the knowledge and reasoning information extracted from text and image facilitates learning more robust and knowledge-aware representations for vision-language tasks.

Language transformer models with knowledge inference. Numerous papers have injected knowledge into language pretraining models (Yu et al., 2020; Xu et al., 2021; Rosset et al., 2021; Zhou et al., 2020; He et al., 2020a; Xiong et al., 2019; He et al., 2020b; Agarwal et al., 2021) with an emphasis on NLP tasks. For example, Yu et al. (2020) extracts knowledge graph information from Wikipedia, and uses it to help the pretraining progress. Xu et al. (2021) injects domain-specific knowledge in pertraining language models for NLP tasks. These methods focus on language tasks, and have not been applied to multi-modal transformers (e.g. for vision and language). More recently, KRISP (Marino et al. (2021)) was proposed to retrieve implicit knowledge stored in pre-trained language models as a supplementary knowledge resource to the structured knowledge base. MAVEx (Wu et al. (2021)) presented an answer validation approach to make better use of the noisy retrieved knowledge. Additionally, some proposed structures and representations are domain-specific and are hard to extend to new tasks. In this paper, we introduce a knowledge-based pretraining model that uses the transformer architecture for multi-modal understanding and reasoning. The knowledge representations in our method can be easily extracted from massive data.

3 Knowledge Source

3.1 Implicit Knowledge Source from OpenAI Models

Text retrieval knowledge: large language models (GPT-3.5). Yang et al. (2022) propose to use GPT-3 for the outside knowledge-based visual question answering task OK-VQA. Instead of using explicit knowledge sources, they use GPT-3 as an implicit source of knowledge. They propose to feed the question and textual descriptions of the image to the GPT-3 model and query it to directly predict the answer. Their model improved the state-of-the-art on OK-VQA by a significant margin of over 9%. Their qualitative analysis shows that the KAT model works quite well on various questions that require external knowledge. Thereby demonstrating the implicit knowledge contained in GPT-3.

Following the KAT (Gui et al., 2022) and PICa model, for each image-question pair, we construct a carefully designed text prompt consisting of a general instruction sentence, the description of the image, the question, and a set of context-question-answer triplets taken from the training dataset that are similar to the current image-question pair. We then input this text prompt to the GPT-3.5 model in its frozen version and obtain the output from GPT-3.5 as the tentative answer candidate to the current image-question pair.

Image retrieval knowledge: image generation module (DALLE-2). DALLE-2 (Ramesh et al., 2022) and Stable Diffusion (Rombach et al., 2021) are text to image generation models that can fill in the scene context via visual inductive bias, which can be considered as source of implicit visual and physical knowledge. For example, one can visualize how the chairs are orientated when there are four chairs around the room, or what objects are typically present in a video conference room. We leverage this model as our knowledge source by running vision language model to extract information from generated image. Specifically, for the task of 3D scene generation, we first generate the 2D image that contains informative scene prior and use the generated 2D content (e.g. orientation, environment objects) to help guide the 3D scene generation.

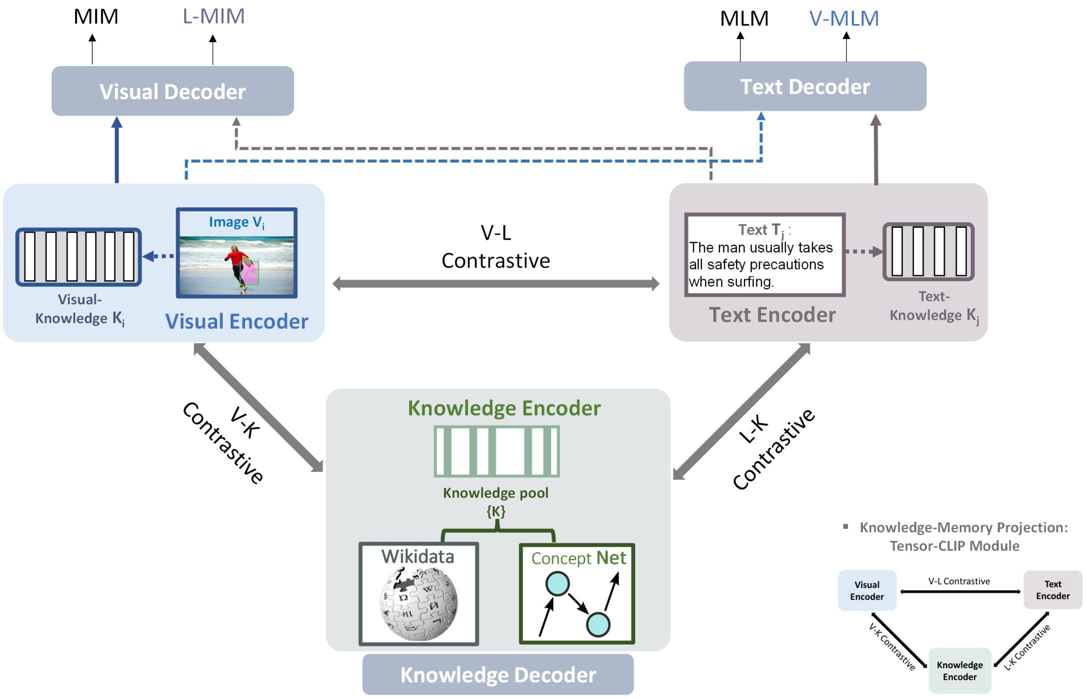


Figure 4: Overview of training *Knowledge-Memory Projection*: Knowledge-Tensor-CLIP module. We use the tensor-CLIP to do the linking from the image patches and text phrases to the wikidata and concept entity. We acquire the positive knowledge for the masked image and text tokens with the nearest neighbor search, and apply weighted contrastive objective. The image and text encoders additionally go through decoder model trained with masked modeling losses respectively. Please find the Section A for knowledge-memory pretraining, and refer the Section 4.1 for more details the knowledge-agent finetune.

3.2 Explicit Knowledge Source from Web Knowledge Bases

We describe the explicit knowledge source to train the Knowledge Tensor-CLIP module. This pool is additionally used for the knowledge-memory agent to retrieve the relevant knowledge for the image and text pairs, and generate new, enhanced prompt for cross-domain space generation.

Factuality knowledge: Wikidata. Wikidata (Vrandecic and Krotzsch, 2014) is an open web-based knowledge base of real-world entities. We use the cleaned version of entity and description text, and format the knowledge text as "`{entity} is a {description}`" following (Wang et al., 2021)[†] and further filter non-English entities, resulting 3,836,524 sentences in total.

Commonsense knowledge: ConceptNet. ConceptNet (Liu and Singh., 2004) is a crowd-sourced project with over 34 million facts organized as knowledge triples collected by translating English language facts into an organized triple structure. It inherently supports common sense knowledge of semantic concepts such as (dog, has property, friendly). We use the dump in Conceptnet 5.5 [‡], and extract 7 types of relation knowledge for English word concepts(IsCapableOf, HasProperty, Causes, AtLocation, PartOf, MadeOf, UsedFor). In total, we obtain 2,697,499 unique triples.

4 Approach

Augmented physical and virtual scenes generation with knowledge-memory agent. The framework of interactive text to 3D scene generation is shown in Fig. 5, which extends the paradigm of calling blackbox models with a trained agent that actively seeks to collect knowledge useful for scene generation. Here, the blackbox models are not trained, and we improve their performance by providing improved text prompts at test time. This involves a knowledge-interactive modeling through a combination of triple systems - one performing knowledge retrieval from image and text query, second performing question and answer generation from the relevant knowledge, and last one writ-

[†] `https://deepgraphlearning.github.io/project/wikidata5m`

[‡] `https://huggingface.co/datasets/conceptnet5`

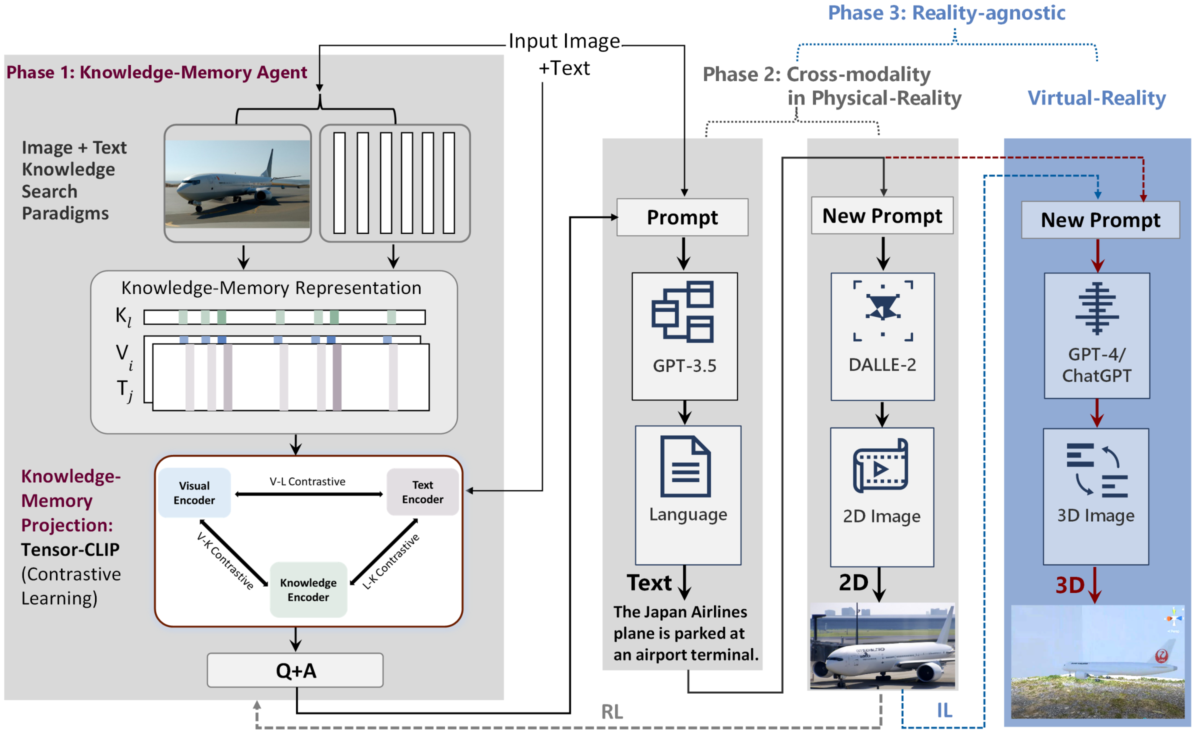

Phase 3: Reality-agnostic
Input Image
+Text
Phase 1: Knowledge-Memory Agent
Phase 2: Cross-modality in Physical-Reality
Virtual-Reality
Image + Text Knowledge Search Paradigms
Knowledge-Memory Representation
K_l
V_i
T_j
Knowledge-Memory Projection: Tensor-CLIP (Contrastive Learning)
Visual Encoder
V-L Contrastive
Text Encoder
V-K Contrastive
L-K Contrastive
Knowledge Encoder
Q+A
Prompt
GPT-3.5
Language
Text
The Japan Airlines plane is parked at an airport terminal.
New Prompt
DALLE-2
2D Image
2D
New Prompt
GPT-4/ ChatGPT
3D Image
3D
RL
IL

Figure 5: The ArK model: At training time, the agent retrieves relevant knowledge for the given image-text pair and asks question and answer for the query. The question and answer are provided to large language models (GPT-3.5, ChatGPT) that generates a new prompt that would be called by the DALL-E model. The similarity between the generated image and original image used as reward to train the agent to learn to select relevant knowledge, while the blackbox models are kept as frozen. At test time, we generate the 2D images with the input text, and model follows the same loop until the new prompt generation step. Instead of feeding back to DALLE-2 for the new prompt, we use ChatGPT to generate a code snippet runnable in a 3D rendering enginne such as Unity. Overall, we use the external knowledge and the visual priors from the generated 2D image to improve the 3D scene generation.

ing a new, informative prompt with reinforcement learning. At test time, we then generate the 2D image output by the three systems and run a final RL process with another knowledge-enhanced DALLE-2/ChatGPT query model to obtain our final image to an image-question pair. Below we describe the details of our proposed triple systems and how we combine the outputs to generate the final 3D image through finetuned knowledge-agent and zero-shot models.

The whole model is trained with three phases. 1) Knowledge-memory agent module for self-supervised Learning; 2) Reinforcement learning module for 2D sense generation in physical world; 3) Imitation Learning module for virtual environment generation.

4.1 Phase 1: Knowledge-memory Agent Training with Self-supervised Learning

Next, we will introduce our trained Knowledge-Memory Agent as the first phase.

Knowledge retrieval system. The knowledge retrieval model takes in the image I and the text caption T to retrieve the useful knowledge statement k^* that aids the understanding of both image and text. This retrieved knowledge statement is used as additional context for powerful instruction fine-tuned language models such as GPT-3.5 to rewrite the text query appropriately.

Training Knowledge-Tensor-CLIP module. For the knowledge retrieval system, as shown in Figure 6, we introduce Knowledge-memory tensor CLIP, a novel image-text-knowledge module that leverages explicit knowledge as bridge to connect the vision and language modalities. The vision encoder is initialized with the CLIP ViT-B/16 (Radford et al., 2021) visual encoder model, and the text and knowledge encoder are initialized with the text encoder model.

To extract knowledge, we follow KAT (Gui et al., 2022), in which the image and text pairs are represented as dense vectors, computed by the image and text encoder of frozen CLIP model. A single

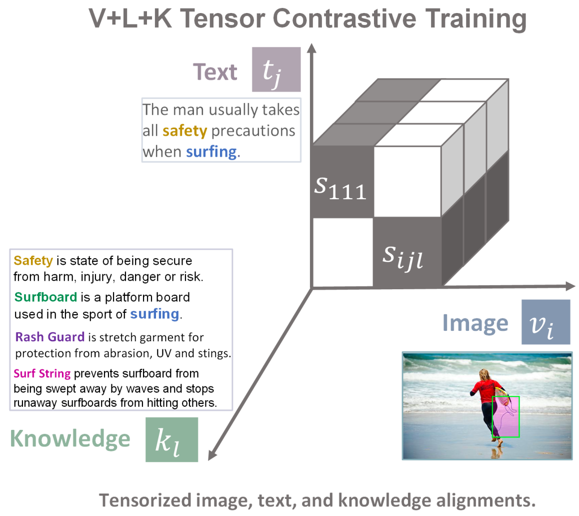

V+L+K Tensor Contrastive Training
Text t_j
The man usually takes all safety precautions when surfing.
s_{111}
s_{ijl}
Safety is state of being secure from harm, injury, danger or risk.
Surfboard is a platform board used in the sport of surfing.
Rash Guard is stretch garment for protection from abrasion, UV and stings.
Surf String prevents surfboard from being swept away by waves and stops runaway surfboards from hitting others.
Image v_i
Knowledge k_l
Tensorized image, text, and knowledge alignments.

Figure 6: Example of ArK task that uses knowledge to identify the relevant knowledge to the image-text candidates. Our task involves leveraging visual and text knowledge retrieved from web-search and implicit knowledge from Open-AI foundation models and the incorporate external web-search knowledge pool about the world.

maximum inner product search (MIPS) index is then built using FAISS (Johnson et al., 2019) to perform nearest-neighbor search. In our setup, we have three dimensions of embeddings (image V, text T, and knowledge K). During training, we keep the knowledge encoder as frozen, while the image and text encoder are updated due to the computational cost to update the knowledge index at every step of training.[§] Thus, we create fixed index embeddings using the frozen knowledge encoder model.

To train the model, we use the contrastive losses used in Radford et al. (2021). As shown in Figure 4, the model is trained with three-way contrastive learning objectives: (Vision-Language, Language-Knowledge, and Image-Knowledge). The vision-language direction loss L_{v2t} and L_{t2v} follows the original objective in CLIP. To acquire the positive knowledge for image-knowledge and language-knowledge direction for the ith batch, we retrieve the top-k knowledge from image K_{vi} and text K_{ti} respectively with nearest neighbor search. The k retrieved knowledge are given the positive label, and the ones from different batch are labeled as negative. If we define u as the knowledge vector, v as the visual vector, and w as the text vector, the model is trained with the loss L with the weighted (a, b, c, d) contrastive losses:

$$L_{v2k} = \sum_{i \in B} \log \frac{\sum_{k \in K_{vi}} u_k^T v_i}{\sum_{k \in \{K_{v1}, \ldots K_{vi}, K_{vB}\}} u_k^T v_i} \quad (1)$$

$$L_{t2k} = \sum_{i \in B} \log \frac{\sum_{k \in K_{ti}} u_k^T w_i}{\sum_{k \in \{K_{t1}, \ldots K_{ti}, K_{tB}\}} u_k^T w_i} \quad (2)$$

$$L_{cont} = aL_{v2t} + bL_{t2v} + cL_{v2k} + dL_{t2k} \quad (3)$$

Following (Wang et al., 2023), we further apply masked image (L_{MIM}), language (L_{MLM}), and vision-language (L_{MVLM}) modeling losses additionally to the image and text encoder based on their effectiveness in the pre-training stages. BEIT-2 is used to get the masked image labels.

$$L_{mask} = L_{MIM} + L_{MLM} + L_{MVLM} \quad (4)$$

The final loss to train the Knowledge-Tensor CLIP module is $L = L_{cont} + L_{mask}$. We refer to Section A in the appendix for more pre-training details. The full training framework is shown in Figure 4.

Inference. At test time, along with the image, we consider extracting knowledge for the individual noun phrases rather than for the entire sentence. This is to ensure that different knowledge for the mentioned objects is extracted that are seldom ignored if only the global sentence context is considered to extract knowledge. To do so, we extract p noun phrases $W_{0,\ldots p}$ with parser tools such as Spacy, and acquire p phrase embeddings $e_{0,\ldots p}$. We then acquire the visual embedding v, and use the average of phrase and visual embeddings from CLIP: $\alpha e_i + (1 - \alpha) v$ as query q_i to perform the nearest neighbor search. We set α as 0.5 and we pick the top-1 best phrase knowledge as our external knowledge based on the cosine similarity score. We evaluate the Knowledage-Tensor-CLIP model on different dataset, and show the result in Section 5.1.

4.2 Phase 2: Knowledge Enhanced Physical Scene Generation with RL

Next, we wish to incorporate the knowledge source to generate a new prompt that contains informative

[§] Because the image and text encoders in CLIP have been pre-trained with contrastive objective, running MIPS at initial training will still retrieve relevant knowledge.

content for physical QA-2D scene generation. After the agent retrieves the relevant knowledge for the given image, text pair, it generates a question and answer tuple using the retrieved knowledge. This model is trained using Reinforcement Learning and is described in the following sections.

Learning knowledge-memory agent. In the first phase of supervised training, we first train the model to ask questions and answer on visual question answering dataset, such as AOKVQA.

Since the parameters of the LLMs such as GPT-3.5 are frozen, during training, the agent receives no information to learn if the retrieved knowledge and generated QAs are indeed useful for the downstream task. Hence we use the feedback from generated images with the knowledge prompt to train the agent using reinforcement learning. We use policy gradient algorithm (Sutton et al., 1999) to train the agent with the reward from similarity between original image and image generated with knowledge enhanced prompt (Red direction in Figure 5. The image is generated with DALL-E and leverages the image-knowledge source to train the agent.

Physical scene generation with knowledge enhanced prompt scheme with RL. After the agent retrieves the relevant knowledge using the knowledge retrieval model $K(V, T)$ for the image V and text T, it generates a question and answer using the retrieved knowledge and image. We use the knowledge-based visual question answer dataset, AOKVQA, as supervision text and image retrieved knowledge to apply reinforcement learning. We further augment the question and answer pairs by prompting GPT-3.5 to generate question and answer using the k retrieved knowledge. The prompt is given as: `Original Sentence: {} Knowledge: {}. Generate question and answer relevant to the sentence and knowledge.` (The details please find in the Figure 12 for prompt in Appendix). This way, the augmented question-answer pairs has the size of k times the size of AOKVQA data, and we use $k = 5$ in our experiments. With this supervision, we then train the agent with seq2seq objective that asks the relevant question and answer in convectional way for both 2D image and knowledge text in knowledge-memory reinforcement learning. Next, we use GPT-3.5 to reformulate the text query using the new knowledge and question-answer with the following prompt: `Original Sentence: {} Question: {} Answer: {} New Sentence:`. (see Figure 13 for prompt template). This Phrase to prepare for the virtual 3D scenes generation, which with the 2D retrieved image from DALL-E, new prompt text and w/knowledge from GPT-3.5, and our convectional question-answer pairs from our trained knowledge memory agent.

Reinforcement learning using feedback. In the first stage of the training, the agent gets no signal from the blackbox model such as ChatGPT and GPT-3.5 to know if the retrieved knowledge and generated QAs are indeed useful for the blackbox models as their weight are frozen. Based on the previous application of reinforcement learning to Natural Language Generation models, we consider the agent to be a policy π_θ with generated question answer sequence qa_k as state. To train the model with reinforcement learning, we use the feedback from generated images with the knowledge prompt k to train the agent. Specifically, we use policy gradient algorithm (Sutton et al., 1999) to train the agent with the reward R from the cosine similarity between original image V and image generated with knowledge enhanced prompt $\tilde{V_k}$ measured by the CLIP ViT-B16 visual encoder model (Red direction in Figure 5. Since we cannot compute the partial reward at each generated token or state, the reward is calculated after the sequence has been fully generated. In the end, the reward R for the image V and text T is computed as follows.

$$qa_k = \pi_\theta(\text{K}(V, T), V) \tag{5}$$

$$\tilde{T_k} = \text{GPT-3.5}(T, qa_k) \tag{6}$$

$$\tilde{V_k} = \text{DALLE-2}(T_k) \tag{7}$$

$$R(V, T) = \cos(CLIP(V), CLIP(\tilde{V_k})) \tag{8}$$

Subsequently, the agent is trained using reinforcement learning to incorporate the feedback using the reward. We use the actor-critic algorithm PPO (Schulman et al., 2017) to update the parameters of the agent using its clipped version:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(R_t(\theta)\hat{A}_t, \text{clip}(R_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right] \tag{9}$$

Here ϵ is a constant which is set to 0.2 and $\hat{A}$ refers to the advantage estimate.

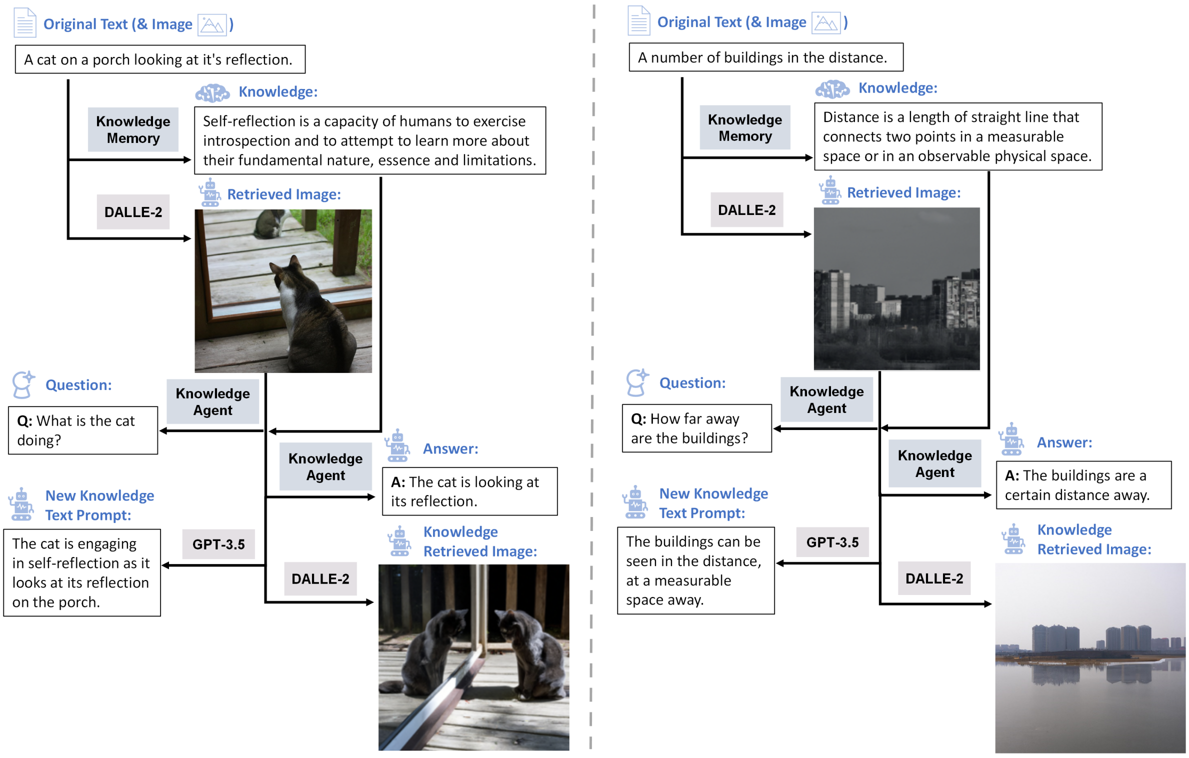

Original Text (& Image)
A cat on a porch looking at it's reflection.
Knowledge Memory
Knowledge:
Self-reflection is a capacity of humans to exercise introspection and to attempt to learn more about their fundamental nature, essence and limitations.
DALLE-2
Retrieved Image:
Question:
Q: What is the cat doing?
Knowledge Agent
Knowledge Agent
Answer:
A: The cat is looking at its reflection.
New Knowledge Text Prompt:
The cat is engaging in self-reflection as it looks at its reflection on the porch.
GPT-3.5
DALLE-2
Knowledge Retrieved Image:
Original Text (& Image)
A number of buildings in the distance.
Knowledge Memory
Knowledge:
Distance is a length of straight line that connects two points in a measurable space or in an observable physical space.
DALLE-2
Retrieved Image:
Question:
Q: How far away are the buildings?
Knowledge Agent
Knowledge Agent
Answer:
A: The buildings are a certain distance away.
New Knowledge Text Prompt:
The buildings can be seen in the distance, at a measurable space away.
GPT-3.5
DALLE-2
Knowledge Retrieved Image:

Figure 7: Qualitative examples of 2D Image with Knowledge Enhanced Prompts.

4.3 Phase 3: 3D Virtual Scenario Diagram with Imitation Learning

In the third phase of the training, the trained knowledge agent is used to perform 3D scene generation. Note the agent requires an image and original text to generate relevant knowledge for the query. Since the image is not provided at test time during scene generation, we use text-to-image generation model DALLE-2 to reconstruct the 2D anchor image which is further used to extract the desired knowledge. Here, DALLE-2 implicitly serves as the image-knowledge source that contains the visual prior knowledge of what we can imagine from the text query. The agent then takes as input the original text and the generated 2D image to retrieve knowledge and outputs a question and answer tuple (ee Figure 9), while GPT-3.5 generates new knowledge-enhanced prompt using the agent output.

To generate the 3D scene from knowledge prompt, we use GPT-4/ ChatGPT to output text code that is then rendered using a 3D rendering engine. We use the prompt and code syntax in GPT-4/ ChatGPT to generate the spatial arrangement in the Unity game engine. We perform experiments with GPT-4/ ChatGPT as the code generation model, and use the Sketchfab API to load the 3D models viewable in the Unity game engine. More information about generating the prompt to run the Unity game engine can be referenced in Roberts et al. (2022).

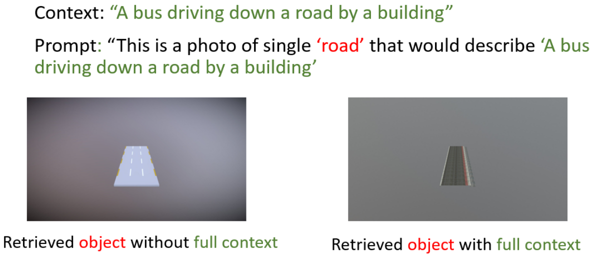

Context: "A bus driving down a road by a building"
Prompt: "This is a photo of single 'road' that would describe 'A bus driving down a road by a building'
Retrieved object without full context
Retrieved object with full context

Figure 8: Comparison of the retrieved object with and without context. We see that prompting CLIP with the full context retrieves a more appropriate object (asphalt road).

$$\theta^* = \arg\max_{\theta} \frac{1}{N} \sum_{i=1}^{N} \log p_\theta(a_i|s_i) \qquad (10)$$

where θ refers to the parameters of the model, N is the number of demonstration trajectories, s_i and a_i are the state and action at time step i. The objective of imitation learning is to find the optimal policy parameters θ^* that maximize the log-likelihood of the expert demonstrations.

4.4 Emergent Behavior for Cross-modality and Reality-agnostic Discussion

We train the knowledge-memory agent and use an RL feedback loop in the real world to randomly initialize a policy, but this strategy does not work well with virtual reality where it is difficult to obtain initial rewards in the 3D environments, especially in the virtual environments with sparse rewards or only terminal rewards. Therefore, a better solution is to use a trained neural network with expert characteristics through imitation learning to help the agent explore better and utilize the unseen environmental space. Imitation Learning (IL), can learn policies directly from expert data.

IL -> Decoupling. In traditional imitation learning, an agent learns a policy by mimicking the behavior of an expert demonstrator. However, directly learning the expert policy may not always be the best approach, as the agent may not generalize well to unseen situations. To address this issue, we propose learning an implicit reward function that captures the essential aspects of the expert's behavior as shown in Phase 2. This results in giving the knowledge-memory agent some information about physical-world behavior to perform tasks by learning from expert demonstrations. It addresses some limitations of traditional imitation learning, which often requires a large amount of expert data and may suffer from compounding errors in complex tasks.

The key idea behind our IL approach involves two components: 1) the agent's physical-world expert demonstrations are collected in the form of state-action pairs; and 2) the virtual environment imitating agent generator. The imitating agent aims to produce actions that mimic the expert's behavior, while the virtual environment imitating agent learns a policy mapping from states to actions by minimizing a loss function that measures the difference between the expert's actions and the actions generated by the learned policy.

Decoupling -> Achieve Generalization. Instead of depending on a task-specific reward function, the agent learns from expert demonstrations. These demonstrations provide a diverse set of state-action pairs that cover various aspects of the task. The agent can then learn a policy that maps states to actions by imitating the expert's behavior. Decoupling in the context of imitation learning refers to separating the learning process from the task-specific reward function. This separation allows the learned policy to generalize across different tasks without explicitly relying on the specific reward function for each task. By decoupling, the agent can learn from expert demonstrations and learn a policy that is adaptable to a variety of situations.

Decoupling enables transfer learning, where a policy learned in one domain can be adapted to other domains with minimal fine-tuning. Since the agent does not rely on a specific reward function, it can adapt to changes in the reward function or environment without the need for significant retraining. This makes the learned policy more robust and generalizable across different environments.

Generalization Element -> Emergent Behavior. Generalization can be used to explain how emergent properties or behaviors can arise from simpler components or rules. The key idea lies in identifying the basic elements or simple rules that govern the behavior of the system. In the context of artificial intelligence, these elements could be individual neurons, simple heuristics, or basic algorithms. Consequently, by observing how these simple components or rules interact with one another. These interactions often lead to the emergence of more complex, higher-level behaviors or properties that cannot be predicted or explained by merely examining the individual components in isolation. Generalization across different levels of complexity: By learning general principles that apply across different levels of complexity, a system can exhibit emergent properties. This generalization enables the system to adapt and respond to new situations, demonstrating the emergence of more complex behaviors from simpler components or rules. Finally, the ability to generalize across different levels of complexity allows for the transfer of knowledge from one domain to another. This transfer contributes to the emergence of complex behaviors or properties in a new context, as the system adapts.

5 Experiments and Results

5.1 Knowledge Agent Training

Knowledge Tensor-CLIP training implementations. We finetune the Knowledge-CLIP model on the WIT dataset (Srinivasan et al., 2021) with the filtered version to only consider English texts, totaling 5M in training data and 30K on test data. We use Wikidata (Vrandecic and Krotzsch, 2014) and ConceptNet (Liu and Singh., 2004) as the ex-

Knowledge Category	*Semantic*		*Encyclopedic*		*Commonsense*			*Open-World*	
Dataset	**Coco**	**Flickr 30K**	**WIT**		**Sherlock**		**VisualCOMET**	**AOKVQA**	
Approach **Metric (%)**	Text -> Image Zero Shot R@1(%)↑	Text -> Image Zero Shot R@1(%)↑	Text -> Image Zero Shot R@1(%)↑	Text -> Image Zero Shot Rank↓	Text -> Image Finetuned Rank↓	Image -> Text Zero Shot Rank↓	Text -> Image Finetuned Acc(%)↑	Text -> Image Finetuned Acc(%)↑	Image -> Text Zero Shot Acc(%)↑
w/o Knowledge									
Contrastive	49.7	51.8	42.0	21.1	28.3	28.7	53.8	60.4	56.3
w/ Knowledge (w/o Mask)									
Knowledge-Tensor-Cont. (ours)	49.8	50.9	**43.4**	-	27.4	-	**54.5**	61.2	-
w/ Knowledge (w/ Mask)									
Knowledge-Tensor-Cont. (ours)	**50.3**	**52.0**	**43.3**	**15.4**	**27.2**	**27.3**	**54.4**	**61.3**	**62.2**

Table 1: Results of text *to* image, and image *to* text retrieval of Knowledge-Tensor-CLIP Memory training. We report the average rank of ground truth image/text, Recall@1 (R@1), and Accuracy (acc) measuring if ground truth image/text is retrieved in top k retrieved knowledge.

Model	Text to Image (R@1)
Srinivasan et al. (2021)	34.4
CLIP (Radford et al., 2021)	42.0
Knowledge-CLIP (Ours)	43.4

Table 2: Text to Image Retrieval on WIT dataset trained with WIT-en. Recall@1 (R@1) is reported for the metric. Ref+Attr is as text input following Srinivasan et al. (2021).

plicit knowledge source as the bridge to connect the image and text modalities. The Knowledge-CLIP model is trained with batch size of 2048, image size of 224, and the Adam optimizer (Kingma and Ba, 2014) with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $= 1e - 6$ for optimization. We use a cosine learning rate decay scheduler with a peak learning rate of 1e-5 and a linear warm-up of 10k steps. The weight decay is 0.05. More pre-training details of the Knowledge-Tensor CLIP model is in Section A of Appendix.

Knowledge-memory retrieval system and evaluation. We first evaluate the performance of knowledge retrieval system on the WIT dataset (Srinivasan et al., 2021), Coco (Lin et al., 2014), Flicker 30K (Plummer et al., 2015), Sherlock (Hessel et al., 2022), and AOKVQA (Schwenk et al., 2022) on the knowledge-based image text retrieval and question answering task. For WIT training and evaluation, we concatenate the reference and attribute to acquire the text representation following Srinivasan et al. (2021).[‖]

In the experiments, we run ablations of the proposed knowledge module and the masking loss in the pre-training stage. We refer to *Contrastive* as the model only on vision-language direction, *i.e.* the same as the CLIP training objective (Radford et al., 2021), *Knowledge-Contrastive* (ours) as Contrastive with knowledge contrastive loss, and *Knowledge-Contrastive-Mask* (ours) as Knowledge-Contrastive trained with the masking loss. We refer to Knowledge-CLIP as this final model.

Table 1 and Table 2 present results on image text retrieval on different knowledge categories dataset: Semantic knowledge, Encyclopedic knowledge, Commonsense knowledge, open-world knowledge, comparing the model trained with (Knowledge-contrastive-training) and without knowledge (CLIP). We see that our knowledge-contrastive-training model provides improvement over the contrastive on the data, for dataset that requires entity-based knowledge.

5.2 Interactive Cross-modality Generation

For the interactive cross-modality generation, we retrieve the top 20 knowledge using the embeddings from the CLIP ViT Base-16 model. We use the text captions for the images in validation set of AOK-VQA (Schwenk et al., 2022) dataset to perform text to 2D scene generation. To train the question and answer model, we initialize the model with BLIP-large (Li et al., 2022) and finetune on the AOK-VQA data in a seq2seq objective with learning rate of $2e^{-6}$, batch size of 128, and for 10 epochs. Policy gradient (Sutton et al., 1999) is used to train the agent after finetuning on AOKVQA data, and the reward is calculated by the CLIP-VIT-base similarity score. Davinci-003 GPT-3.5 model is used to generate the new knowledge prompts for 2D image, and DALLE-2 images are generated with 256x256 resolution, which we used to generate 2D image and the feedback image for the RL algorithm.

QA-2D image generation and evaluation. Figure 7 shows example of DALLE-2 generated im-

[‖] One of reference or attribute is used if both are not available in the data.

Model	Relevance (%)	Naturalness (%)
Conversational-2D Scenes Generation		
DALL-E	78.0	81.0
DALL-E w/ Knowledge (ours)	**87.0**	**84.0**
Conversational-3D Scenes Generation		
GPT4 / ChatGPT - Game Engine	59.9	35.0
GPT4 / ChatGPT - Game Engine w/ Knowledge (ours)	**71.1**	**48.0**

Table 3: Human evaluation of Conversational-2D (from DALL-E 2) and Conversational-3D scenes generation (from GPT4/ChatGPT and the Game Engine (Unity Stage)). We measure the relevance between scene and text and naturalness of generated scene. We asked yes/no questions to 5 human annotators if the and take the majority vote to get the response on M-Turk[¶].

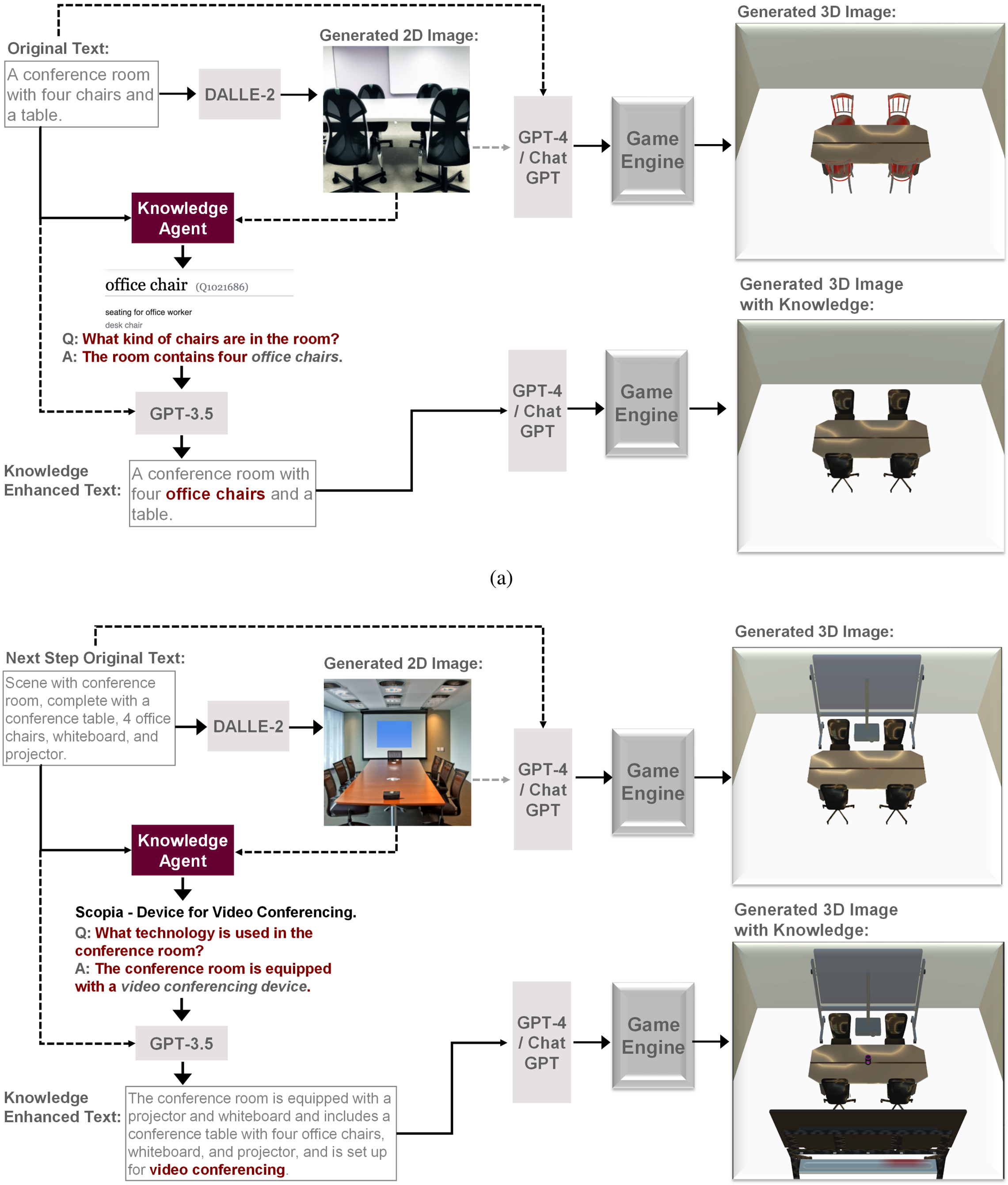
ages with original text and knowledge incorporated text query. We see that by modifying the text query in zero shot setting while keeping the blackbox OpenAI models as frozen, we are able to generate more informative and realistic images. For example, we see more natural portrait of cat drinking water from sink and man looking at himself in the mirror. The last example includes unmentioned entity of GS workstation, but DALLE-2 is able to generate more realistic scene using the entity information. Please find in the table 3 to find our human-evaluation results of the convectional 2D image generation.

5.3 Virtual Environment Scene Generation, Editing, and Evaluation

Loading the relevant 3D objects. Roberts et al. (2022) use the Sketchfab dataset to randomly select 3D models that include the given text input. The caveat with this approach is that the chosen model is not guaranteed to include the correct class of objects due to noisy user labels, as well as their size and orientation. Instead of extracting objects from the Sketchfab dataset, we use the Objaverse (Deitke et al., 2022) that provides access to annotated 3D models from the Sketchfab dataset. To ensure that the loaded models correspond to their true objects, we use CLIP (Radford et al., 2021) to compute a similarity score between the object image and the text and choose the model with the highest score. For each object and text pair, we provide the following prompt: `This is an image of {object} that refers to {text}`. We observe that providing the entire context helps to retrieve the most relevant object as shown in Figure 8.

3D scene generation with knowledge enhanced prompt. Figure 9 shows the pipeline for 3D scene generation using the 2D prior. Without the ground truth image, we only have access to the text query that is used by the agent to retrieve knowledge, we first generate 2D images to determine what the scene would look like in a real world setting. Using the generated image and original text, the agent retrieves explicit knowledge and asks relevant question and answer. Consequently, GPT-3.5 changes the original text to include the knowledge snippet that makes the prompt more informative and realistic in this scene. Using the prompt in (Roberts et al., 2022), we use GPT-4/ ChatGPT for the spatial arrangement and the program synthesis generation for 3D scenes. We next present qualitative results of 3D scene generation with the enhanced knowledge. We finally use GPT4/ChatGPT to generate the spatial arrangement and programming which is then rendered in the Unity game engine. Note that the OpenAI models (colored in gray) are kept frozen while we only train the agent to perform 3D scene editing. As showed the simulation of the figure 9, we observe that with external knowledge the 3D scene a) substitutes wooden chairs to office chairs, and b) includes tools used for video conferencing, making the scene look more realistic.

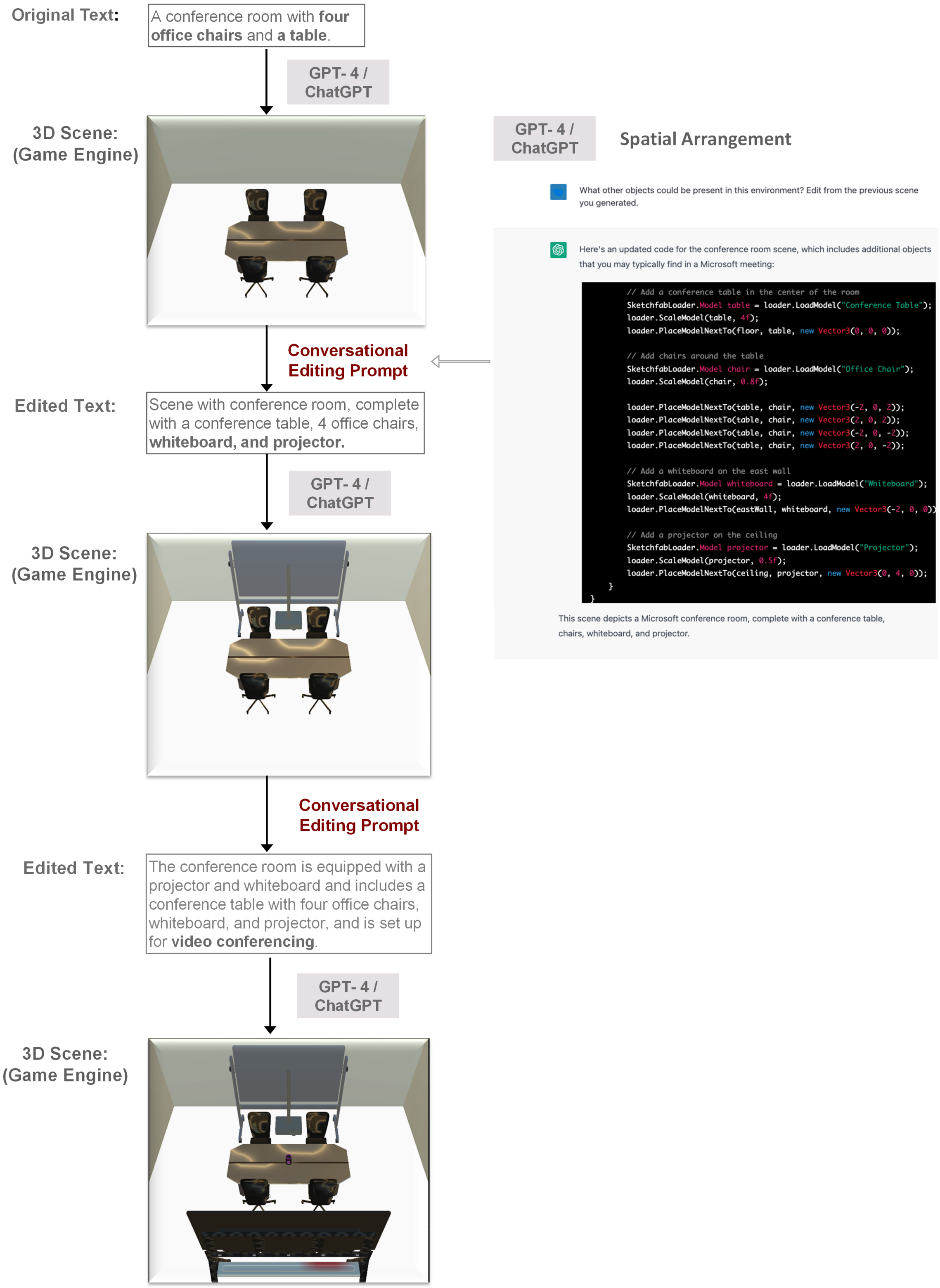
Cross-modality 3D scene editing. Knowledge-enhanced text opens up a novel method of editing 3D scenes with knowledge prior to help improve the naturalness of the scene. We also include the results of editing a 3D scene interactively in a dialogue setting. Figure 10 shows an example in which we provide the previously generated code and ask ChatGPT to fill relevant objects with the new prompt. We observe that ChatGPT is able to understand the previous scene and adds relevant environment objects such as whiteboard and projector in the appropriate orientation and location.


Original Text:
A conference room with four chairs and a table.
DALLE-2
Generated 2D Image:
GPT-4 / Chat GPT
Game Engine
Generated 3D Image:
Knowledge Agent
office chair (Q1021686)
seating for office worker
desk chair
Q: What kind of chairs are in the room?
A: The room contains four office chairs.
GPT-3.5
Knowledge Enhanced Text:
A conference room with four office chairs and a table.
GPT-4 / Chat GPT
Game Engine
Generated 3D Image with Knowledge:
Next Step Original Text:
Scene with conference room, complete with a conference table, 4 office chairs, whiteboard, and projector.
DALLE-2
Generated 2D Image:
GPT-4 / Chat GPT
Game Engine
Generated 3D Image:
Knowledge Agent
Scopia - Device for Video Conferencing.
Q: What technology is used in the conference room?
A: The conference room is equipped with a video conferencing device.
GPT-3.5
Knowledge Enhanced Text:
The conference room is equipped with a projector and whiteboard and includes a conference table with four office chairs, whiteboard, and projector, and is set up for video conferencing.
GPT-4 / Chat GPT
Game Engine
Generated 3D Image with Knowledge:

(a)

(b)

Figure 9: Qualitative examples of 3D scene editing with knowledge enhanced prompts. At inference time, we first generate an image from the input text to learn the prior . The knowledge agent then generates a question and answer tuple which is fed as an input to GPT-3.5. The output of GPT-3.5 is an enhanced version of the input text with added information from external knowledge sources. This text is then given to ChatGPT that outputs the spatial arrangements and low-level program synthesis code. Finally, this code is rendered using Unity engine to output the desired 3D object.



Original Text: A conference room with **four office chairs** and **a table**.

GPT- 4 / ChatGPT

3D Scene: (Game Engine)

Conversational Editing Prompt

Edited Text: Scene with conference room, complete with a conference table, 4 office chairs, **whiteboard, and projector.**

GPT- 4 / ChatGPT

3D Scene: (Game Engine)

Conversational Editing Prompt

Edited Text: The conference room is equipped with a projector and whiteboard and includes a conference table with four office chairs, whiteboard, and projector, and is set up for **video conferencing**.

GPT- 4 / ChatGPT

3D Scene: (Game Engine)

GPT- 4 / ChatGPT

Spatial Arrangement

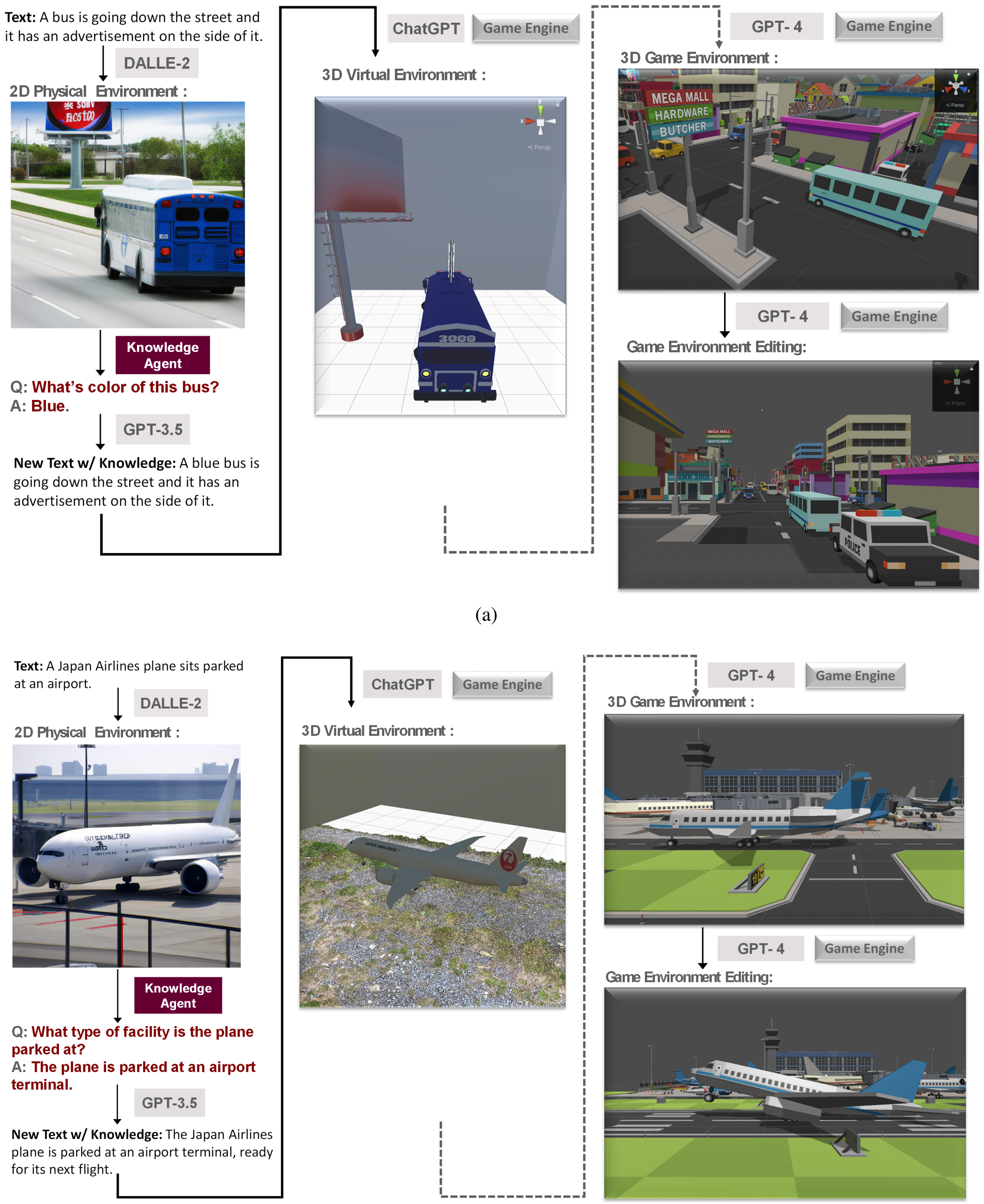
What other objects could be present in this environment? Edit from the previous scene you generated.

Here's an updated code for the conference room scene, which includes additional objects that you may typically find in a Microsoft meeting:

```
        // Add a conference table in the center of the room
        SketchfabLoader.Model table = loader.LoadModel("Conference Table");
        loader.ScaleModel(table, 4f);
        loader.PlaceModelNextTo(floor, table, new Vector3(0, 0, 0));

        // Add chairs around the table
        SketchfabLoader.Model chair = loader.LoadModel("Office Chair");
        loader.ScaleModel(chair, 0.8f);

        loader.PlaceModelNextTo(table, chair, new Vector3(-2, 0, 2));
        loader.PlaceModelNextTo(table, chair, new Vector3(2, 0, 2));
        loader.PlaceModelNextTo(table, chair, new Vector3(-2, 0, -2));
        loader.PlaceModelNextTo(table, chair, new Vector3(2, 0, -2));

        // Add a whiteboard on the east wall
        SketchfabLoader.Model whiteboard = loader.LoadModel("Whiteboard");
        loader.ScaleModel(whiteboard, 4f);
        loader.PlaceModelNextTo(eastWall, whiteboard, new Vector3(-2, 0, 0))

        // Add a projector on the ceiling
        SketchfabLoader.Model projector = loader.LoadModel("Projector");
        loader.ScaleModel(projector, 0.5f);
        loader.PlaceModelNextTo(ceiling, projector, new Vector3(0, 4, 0));
    }
}
```

This scene depicts a Microsoft conference room, complete with a conference table, chairs, whiteboard, and projector.

Figure 10: Editing 3D scene with our trained dialogue interactive scenes projection using GPT4/ChatGPT. We see that GPT4/ChatGPT adds more knowledge context with the spatial arrangement to the original text as shown in the edited text.



Text: A bus is going down the street and it has an advertisement on the side of it.

Q: What's color of this bus?
A: Blue.

New Text w/ Knowledge: A blue bus is going down the street and it has an advertisement on the side of it.

(a)

Text: A Japan Airlines plane sits parked at an airport.

Q: What type of facility is the plane parked at?
A: The plane is parked at an airport terminal.

New Text w/ Knowledge: The Japan Airlines plane is parked at an airport terminal, ready for its next flight.

(b)

Figure 11: Cross-modality and reality-agnostic generation and editing with interactive agent using GPT-4 and ChatGPT.

5.4 Reality-agnostic Generation Observation with Emergent Ability.

We provide DALLE-2 with a text query to generate a real-world 2D image. Then following our pipeline, we give this image as an input to the knowledge agent and consequently the enhanced query is provided to ChatGPT to generate a 3D object. Now this object when further as an input to GPT-4 which adds context to the 3D object by changing the surrounding and appearance of the objects according to the specifications of the scene. Figure 11 showed that examples which we present the VR editing and a novel approach for generating 3D game scene to user shared conversational interactive QA. In contrast to traditional 2D image generation and image-grounded dialogue tasks we focus on synthesizing 3D-gaming editing content that is relevance and naturalness with the Emergent ability for domain-agnostic.

5.5 Human Evaluation.

Since there is no existing metric to auto-evaluate the conversational interactive scene generation, we rely on human evaluation to analyze the results. For each generated scene, we evaluate using scores from 5 humans using crowd-sourcing platform[**]. We ask if the generated interactive scene (knowledge-dialogue 2D and 3D) 1) Relevance: matches the text conversational description, and 2) Naturalness: looks realistic. The results are shown in Table 3. We see that for both 2D and 3D scene types, knowledge-enhanced text results in more realistic scenes. The 2D scenes greatly benefit from the knowledge in terms of relevance, and both necessarily with the 3D scene generation with the imitation and conversational dialogue way.

6 Conclusion

In this work, we explored the usage of OpenAI models to tackle text to augmented reality scene generation using knowledge inference interaction. We found qualitatively that incorporating knowledge in the new prompt itself provides an improvement of generated physical and virtual environment for the foundation models. We leave it as future work to explore real-world different types of human knowledge feedback and thorough investigation of more learning algorithms such as from emergence mechanism to improve our pipeline (e.g., prosody, anaphora, gesture, etc.) for the mix-reality generative AI.

Ethics Statement

LLM has many applications. In addition to 2D and 3D generation, grounded language models could help drive content generation and editing for bots and AI agents, and assist in productivity applications, helping to re-write, paraphrase, translate or synthesize text. Fundamental advances in text derived 2D and 3D generation help contribute towards these goals and many would benefit from a greater understanding of how to model emergent ability and empathetic with language and image in the physical world. Arguably many of these applications could have positive benefits.

However, the emerging ability technology could also be used by bad actors. AI systems that generate content can be used to manipulate or deceive people. Therefore, it is very important that this technology is developed in accordance with responsible AI guidelines. For example, explicitly communicating to users that content is generated by an AI system and providing the user with controls in order to customize such a system. It is possible the emerging ability could be used to develop new methods to detect manipulative content - partly because it is rich with robotic empathy with LLM and virtual environment generation - and thus help address another real world problem.

Acknowledgement

We are especially grateful to Antonio Criminisi, Tom Cashman, Andy Wilson, Bala Kumaravel, Andrzej Banburski-Fahey, Corby Rosset, Michel Galley, Jianwei Yang, Mary Czerwinski, Pan Lu, Kiran Muthabatulla, Silviu Cucerzan, Ahmed Awadallah, Nguyen Bach, Saleema Amershi for their comments, suggestions, painstaking multiple reviews of this paper, and their pointers to the literature.

References

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv:2010.12688*.

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. *Proceedings of EMNLP*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and

[**] https://www.mturk.com/

Jingjing Liu. 2019. Uniter: Universal image-text representation learning. *Proceedings of ECCV*.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2022. Objaverse: A universe of annotated 3d objects. *ArXiv*, abs/2212.08051.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. 2020. Conceptbert: Concept-aware representation for visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. 2022. Kat: A knowledge augmented transformer for vision-and-language. In *NAACL 2022. Long paper, Oral.* arXiv:2112.08614.

Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. 2020a. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv:2012.03551*.

Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. 2020b. Bert-mk: Integrating graph contextualized knowledge into pre-trained language models. *Proceedings of ACL*.

Jack Hessel, Jena D Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. 2022. The Abduction of Sherlock Holmes: A Dataset for Visual Abductive Reasoning. *arXiv preprint arXiv:2202.04800*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv:2102.03334*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2020a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *Proceedings of AAAI*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. 2020b. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv:2004.06165*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2014. Microsoft coco: Common objects in context. *Proceedings of ECCV*.

Hugo Liu and Push Singh. 2004. Conceptnet—a practical commonsense reasoning tool-kit. In *BT technology journal, 22(4):211–226, 2004.*

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Proceedings of NeurIPS*.

Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. 2021. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *The 34th Conference on Computer Vision and Pattern Recognition (CVPR)*.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. 2022. BEiT v2: Masked image modeling with vector-quantized visual tokenizers.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*.

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 *ICML*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents.

Jasmine Roberts, Andrzej Banburski-Fahey, and Jaron Lanier. 2022. Steps towards prompt-based creation of virtual worlds. *ArXiv*, abs/2211.05875.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2021. High-resolution image synthesis with latent diffusion models.

Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. 2021. Knowledge-aware language model pretraining. *arXiv:2007.00655*.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. 2022. Photorealistic text-to-image diffusion models with deep language understanding.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. 2022. A-okvqa: A benchmark for visual question answering using world knowledge. *arXiv*.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. 2021. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*.

Richard S. Sutton, David A. McAllester, Satinder Singh, and Y. Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. *Proceedings of EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Denny Vrandecic and Markus Krotzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 2023. Image as a foreign language: BEiT pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2021. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models.

Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. 2021. Multi-modal answer validation for knowledge-based vqa. In *arXiv preprint, arXiv:2103.12248*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv:1912.09637*.

Song Xu, Haoran Li, Peng Yuan, Yujia Wang, Youzheng Wu, Xiaodong He, Ying Liu, and Bowen Zhou. 2021. K-plug: Knowledge-injected pre-trained language model for natural language understanding and generation in e-commerce. *arXiv:2104.06960*.

Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 2022. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv:2010.00796*.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. 2022. Scaling autoregressive models for content-rich text-to-image generation.

Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. 2021. Vinvl: Revisiting visual representations in vision-language models. *arXiv:2101.00529*.

Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. 2019. Unified vision-language pre-training for image captioning and vqa. *Proceedings of AAAI*.

Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. 2020. Pre-training text-to-text transformers for concept-centric common sense. *arXiv:2011.07956*.

Appendix

A Knowledge-Tensor CLIP Pre-Training Details

The Knowledge-Tensor CLIP model is trained to align the image, text, and knowledge modalities together to optimize the knowledge retrieval mechanism using the both image and text modalities. Inspired by the effectiveness of masked training in BEIT-3 (Wang et al., 2023), we add the decoder-based masked loss in the visual and text encoders, in which the masked image patches and masked text tokens are given as input. The contrastive learning is applied to the same masked inputs during training to allow the model to reason over different image regions and text. The masking loss is not applied to the knowledge encoder the knowledge embeddings are kept as frozen throughout training. We randomly mask 15% tokens of texts and mask 40% of image patches using a block-wise masking strategy as in BEIT2 (Peng et al., 2022). The effect of masked loss is shown in Table 1, which provides a slight boost in the downstream task evaluations.

B Prompt for Knowledge-Memory Agent

B.1 Knowledge-based Question Answer Generation

The prompt for GPT-3.5 to provide additional question-answering supervision data for the knowledge memory agent is shown in Figure 12. We retrieve the top K knowledge using the Tensor-CLIP model with the COCO image and text captions. With knowledge K as context, question and answer are generated to train the agent.

B.2 Prompt for Knowledge-Enhanced Query Generation

Figure 13 shows a prompt to query GPT-3.5 that utilizes the retrieved knowledge, question and answer generated by the agent to reformulate the original text into a ‘knowledge-enhanced" description.

B.3 Prompt for VR and Game Scene Generation for GPT Models

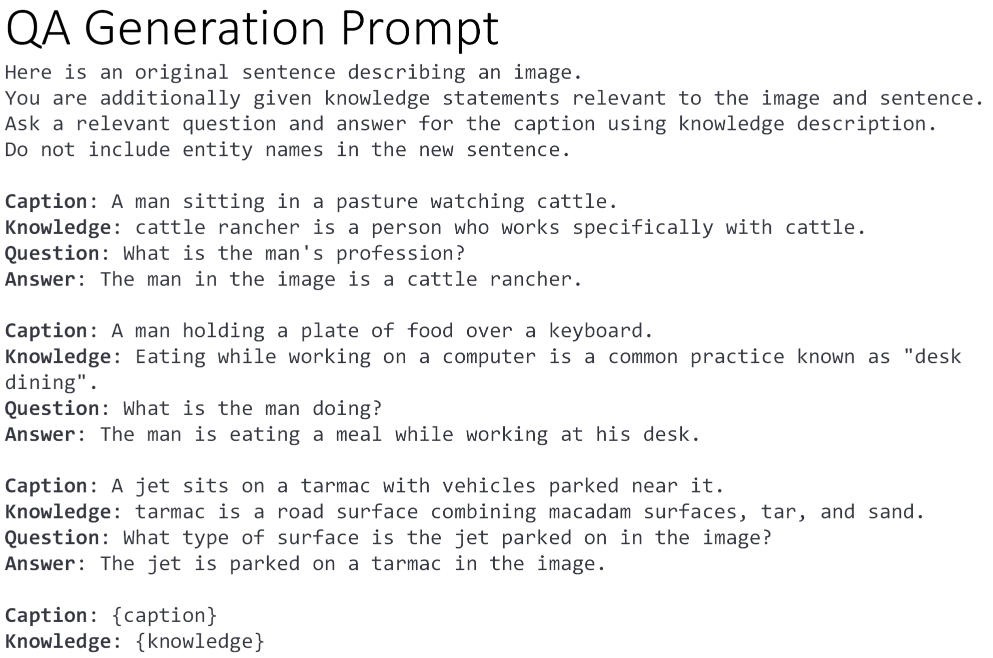
In Figure 14, we provide ChatGPT with a text prompt to generate the program synthesis that is then rendered in the Unity game engine. If the query is about generating a scene in a game, the model is able to find the relevant context information, such as size, physics, relative orientations, and other relative objects in the environment, and output a 3D scene that resembles the scenes from the game. It uses the Sketchfab assets that could be easily placed in the Unity game engine.

C Human Evaluation Details

In Fig. 15, we show screenshots of the instructions that were given to the participants for the human evaluation study using mechanical turk. The results shown in Sec 5.5 are based on two metrics shown in the figure here: Relevance and Naturalness. The users have to select which scene is more relevant and natural separately for both 2D and 3D images. Three human evaluators are asked to choose binary yes/no choice for each image, and we take the average of answers to evaluate the scene generation performances.

D Microsoft Gaming Scenario

We showed the examples of Microsoft 3D Game Scenario. One of the knowledge interactive simulation in Microsoft Fight Simulator; another is knowledge interactive simulation in Minecraft scene generator. The details please refer the Fig. 17.



QA Generation Prompt

Here is an original sentence describing an image.
You are additionally given knowledge statements relevant to the image and sentence.
Ask a relevant question and answer for the caption using knowledge description.
Do not include entity names in the new sentence.

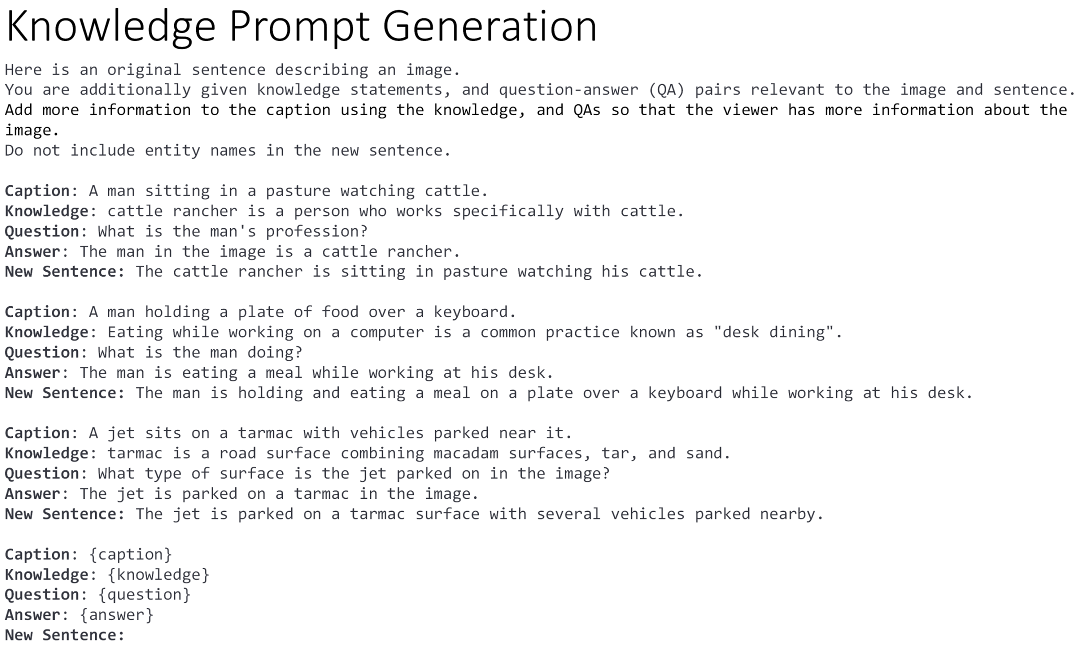
Caption: A man sitting in a pasture watching cattle.
Knowledge: cattle rancher is a person who works specifically with cattle.
Question: What is the man's profession?
Answer: The man in the image is a cattle rancher.

Caption: A man holding a plate of food over a keyboard.
Knowledge: Eating while working on a computer is a common practice known as "desk dining".
Question: What is the man doing?
Answer: The man is eating a meal while working at his desk.

Caption: A jet sits on a tarmac with vehicles parked near it.
Knowledge: tarmac is a road surface combining macadam surfaces, tar, and sand.
Question: What type of surface is the jet parked on in the image?
Answer: The jet is parked on a tarmac in the image.

Caption: {caption}
Knowledge: {knowledge}

Figure 12: An example of the evidence of rationale QA that we obtain from GPT-3.5 by using a combination of image and text candidate to query it.



Knowledge Prompt Generation

Here is an original sentence describing an image.
You are additionally given knowledge statements, and question-answer (QA) pairs relevant to the image and sentence.
Add more information to the caption using the knowledge, and QAs so that the viewer has more information about the image.
Do not include entity names in the new sentence.

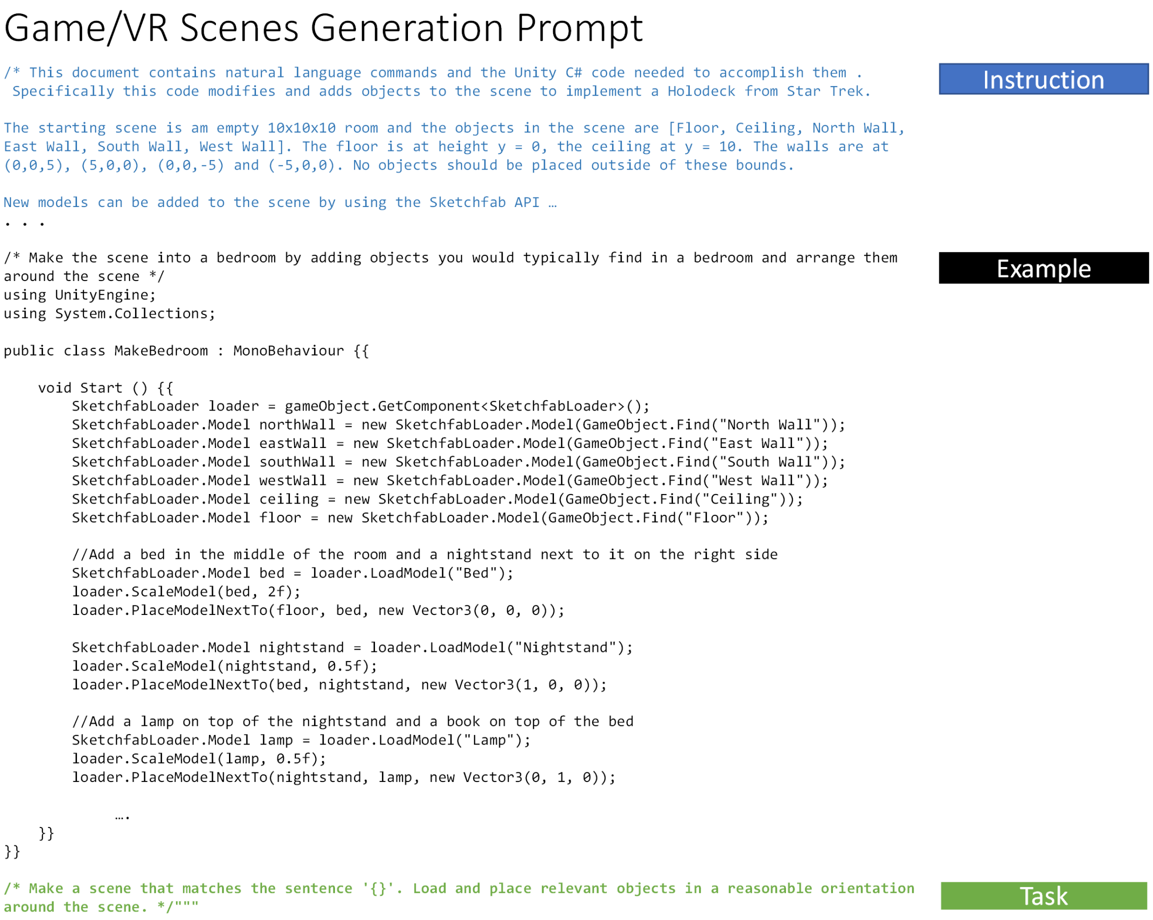
Caption: A man sitting in a pasture watching cattle.
Knowledge: cattle rancher is a person who works specifically with cattle.
Question: What is the man's profession?
Answer: The man in the image is a cattle rancher.
New Sentence: The cattle rancher is sitting in pasture watching his cattle.

Caption: A man holding a plate of food over a keyboard.
Knowledge: Eating while working on a computer is a common practice known as "desk dining".
Question: What is the man doing?
Answer: The man is eating a meal while working at his desk.
New Sentence: The man is holding and eating a meal on a plate over a keyboard while working at his desk.

Caption: A jet sits on a tarmac with vehicles parked near it.
Knowledge: tarmac is a road surface combining macadam surfaces, tar, and sand.
Question: What type of surface is the jet parked on in the image?
Answer: The jet is parked on a tarmac in the image.
New Sentence: The jet is parked on a tarmac surface with several vehicles parked nearby.

Caption: {caption}
Knowledge: {knowledge}
Question: {question}
Answer: {answer}
New Sentence:

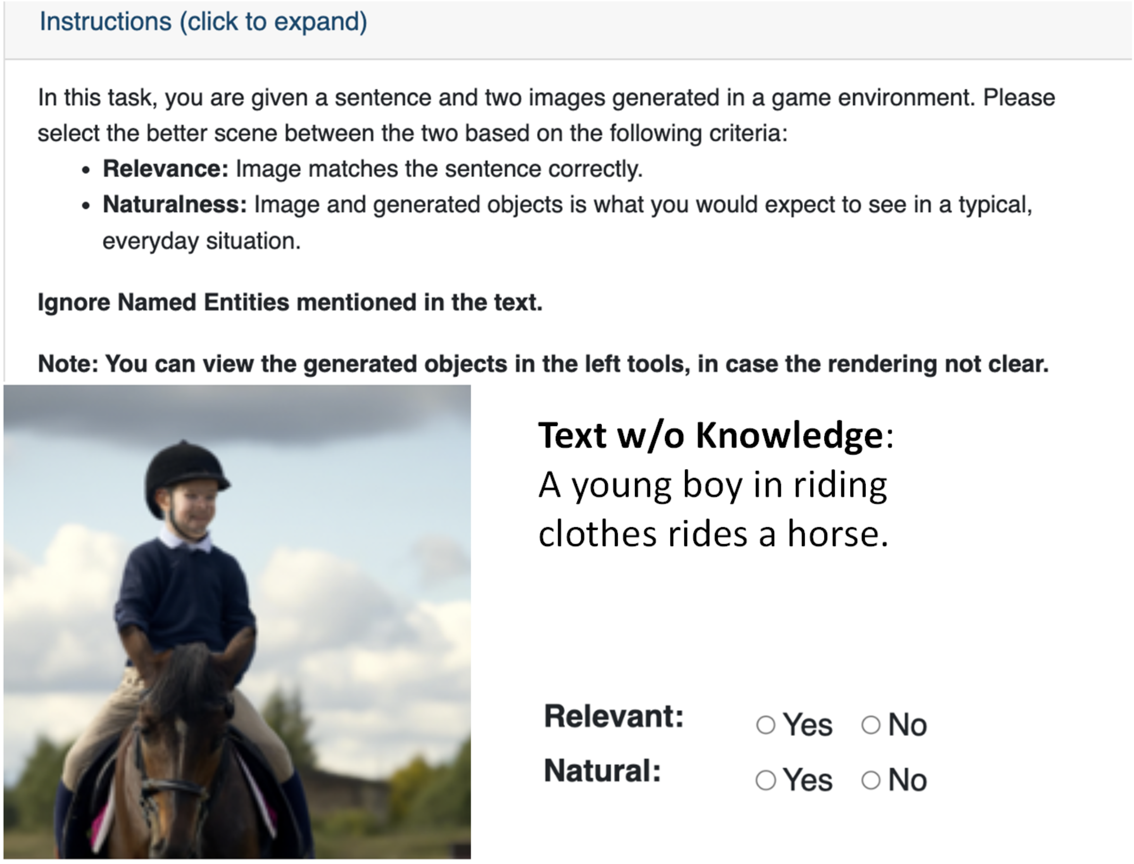
Figure 13: An example of the prompts that we use to query GPT-3.5 in our knowledge-augmented GPT-3.5 query system.



Game/VR Scenes Generation Prompt

```
/* This document contains natural language commands and the Unity C# code needed to accomplish them .
 Specifically this code modifies and adds objects to the scene to implement a Holodeck from Star Trek.

The starting scene is am empty 10x10x10 room and the objects in the scene are [Floor, Ceiling, North Wall,
East Wall, South Wall, West Wall]. The floor is at height y = 0, the ceiling at y = 10. The walls are at
(0,0,5), (5,0,0), (0,0,-5) and (-5,0,0). No objects should be placed outside of these bounds.

New models can be added to the scene by using the Sketchfab API …
. . .

/* Make the scene into a bedroom by adding objects you would typically find in a bedroom and arrange them
around the scene */
using UnityEngine;
using System.Collections;

public class MakeBedroom : MonoBehaviour {{

    void Start () {{
        SketchfabLoader loader = gameObject.GetComponent<SketchfabLoader>();
        SketchfabLoader.Model northWall = new SketchfabLoader.Model(GameObject.Find("North Wall"));
        SketchfabLoader.Model eastWall = new SketchfabLoader.Model(GameObject.Find("East Wall"));
        SketchfabLoader.Model southWall = new SketchfabLoader.Model(GameObject.Find("South Wall"));
        SketchfabLoader.Model westWall = new SketchfabLoader.Model(GameObject.Find("West Wall"));
        SketchfabLoader.Model ceiling = new SketchfabLoader.Model(GameObject.Find("Ceiling"));
        SketchfabLoader.Model floor = new SketchfabLoader.Model(GameObject.Find("Floor"));

        //Add a bed in the middle of the room and a nightstand next to it on the right side
        SketchfabLoader.Model bed = loader.LoadModel("Bed");
        loader.ScaleModel(bed, 2f);
        loader.PlaceModelNextTo(floor, bed, new Vector3(0, 0, 0));

        SketchfabLoader.Model nightstand = loader.LoadModel("Nightstand");
        loader.ScaleModel(nightstand, 0.5f);
        loader.PlaceModelNextTo(bed, nightstand, new Vector3(1, 0, 0));

        //Add a lamp on top of the nightstand and a book on top of the bed
        SketchfabLoader.Model lamp = loader.LoadModel("Lamp");
        loader.ScaleModel(lamp, 0.5f);
        loader.PlaceModelNextTo(nightstand, lamp, new Vector3(0, 1, 0));

             ….
    }}
}}

/* Make a scene that matches the sentence '{}'. Load and place relevant objects in a reasonable orientation
around the scene. */"""
```

Figure 14: An example of the prompts for VR and game scene generation and editing which we use GPT4/ChatGPT in our knowledge-augmented query system.



Instructions (click to expand)

In this task, you are given a sentence and two images generated in a game environment. Please select the better scene between the two based on the following criteria:

- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

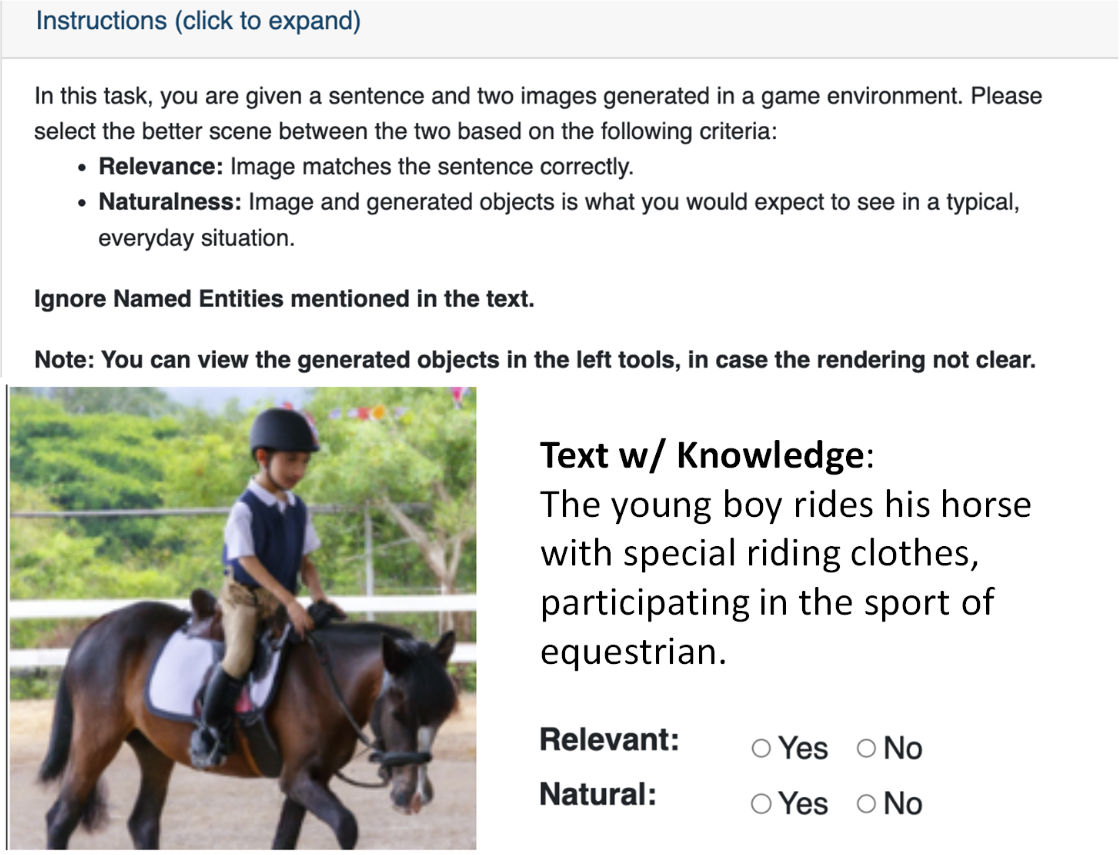
Ignore Named Entities mentioned in the text.

Note: You can view the generated objects in the left tools, in case the rendering not clear.

Text w/o Knowledge:
A young boy in riding clothes rides a horse.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No



Instructions (click to expand)

In this task, you are given a sentence and two images generated in a game environment. Please select the better scene between the two based on the following criteria:

- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

Ignore Named Entities mentioned in the text.

Note: You can view the generated objects in the left tools, in case the rendering not clear.

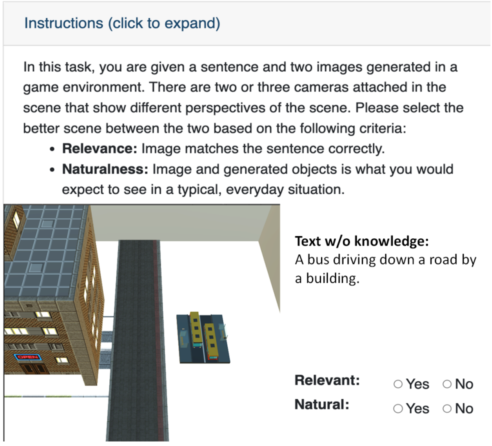
Text w/ Knowledge:
The young boy rides his horse with special riding clothes, participating in the sport of equestrian.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No

Figure 15: An example of the conversational 2D human evaluation.

Human Evaluation Example as Single Camera:



Instructions (click to expand)

In this task, you are given a sentence and two images generated in a game environment. There are two or three cameras attached in the scene that show different perspectives of the scene. Please select the better scene between the two based on the following criteria:

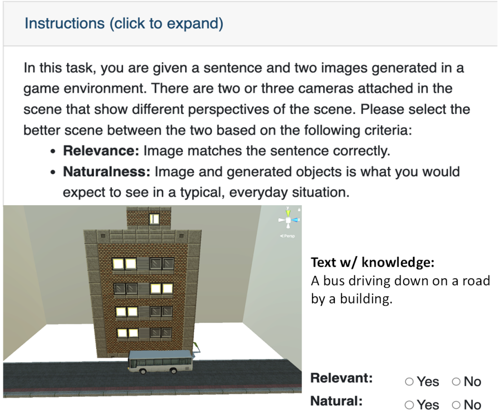
- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

Text w/o knowledge:
A bus driving down a road by a building.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No

Human Evaluation Examples as Three Cameras:



Instructions (click to expand)

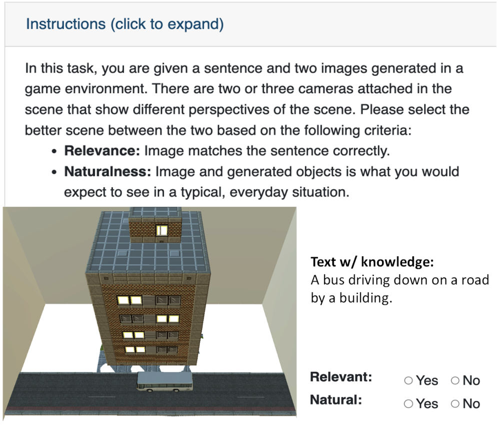
In this task, you are given a sentence and two images generated in a game environment. There are two or three cameras attached in the scene that show different perspectives of the scene. Please select the better scene between the two based on the following criteria:

- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

Text w/ knowledge:
A bus driving down on a road by a building.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No



Instructions (click to expand)

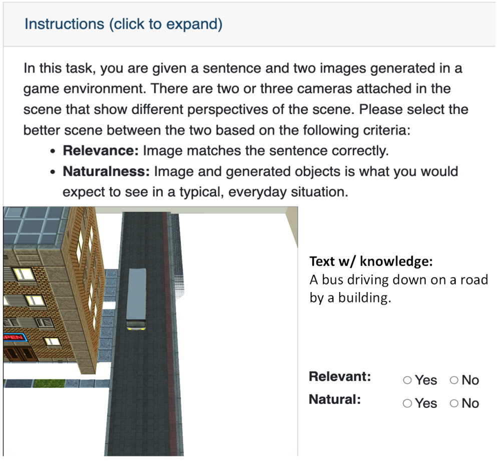
In this task, you are given a sentence and two images generated in a game environment. There are two or three cameras attached in the scene that show different perspectives of the scene. Please select the better scene between the two based on the following criteria:

- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

Text w/ knowledge:
A bus driving down on a road by a building.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No



Instructions (click to expand)

In this task, you are given a sentence and two images generated in a game environment. There are two or three cameras attached in the scene that show different perspectives of the scene. Please select the better scene between the two based on the following criteria:

- **Relevance:** Image matches the sentence correctly.
- **Naturalness:** Image and generated objects is what you would expect to see in a typical, everyday situation.

Text w/ knowledge:
A bus driving down on a road by a building.

Relevant: ○ Yes ○ No

Natural: ○ Yes ○ No

Figure 16: Two examples of human evaluation for the conversational 3D VR Scenario. One is for single camera example; another is for the three cameras examples.

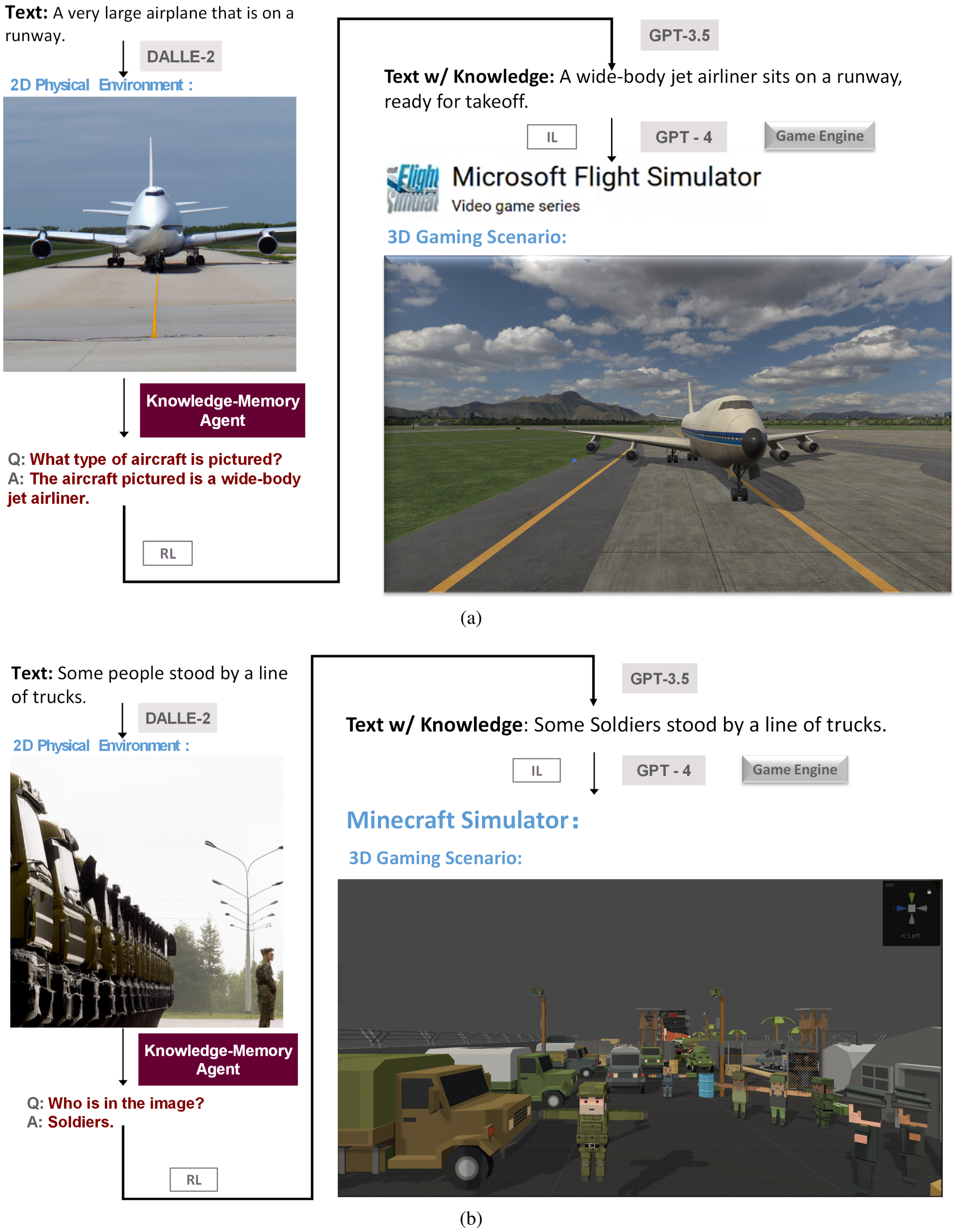

Text: A very large airplane that is on a runway.
DALLE-2
2D Physical Environment :
Knowledge-Memory Agent
Q: What type of aircraft is pictured?
A: The aircraft pictured is a wide-body jet airliner.
RL
GPT-3.5
Text w/ Knowledge: A wide-body jet airliner sits on a runway, ready for takeoff.
IL
GPT - 4
Game Engine
Microsoft Flight Simulator
Video game series
3D Gaming Scenario:
(a)
Text: Some people stood by a line of trucks.
DALLE-2
2D Physical Environment :
Knowledge-Memory Agent
Q: Who is in the image?
A: Soldiers.
RL
GPT-3.5
Text w/ Knowledge: Some Soldiers stood by a line of trucks.
IL
GPT - 4
Game Engine
Minecraft Simulator:
3D Gaming Scenario:
(b)

Figure 17: Two examples of Microsoft 3D Game Scenario. One of the knowledge interactive simulation in Microsoft Fight Simulator; another is knowledge interactive simulation in Minecraft scene generator.