

ArK: Augmented Reality with Knowledge Emergent Infrastructure

Qiuyuan Huang^{†*}

Jae Sung Park^{§‡*}

Abhinav Gupta^{†‡*}

Pan Lu^{‡‡*}

Paul Bennett[‡]

Ran Gong[‡]

Subhojit Som[‡]

Baolin Peng[‡]

Owais Khan Mohammed[‡]

Chris Pal[†]

Yejin Choi[§]

Jianfeng Gao[‡]

[‡]Microsoft Research, Redmond

[†] MILA

[§]University of Washington

[‡] UCLA

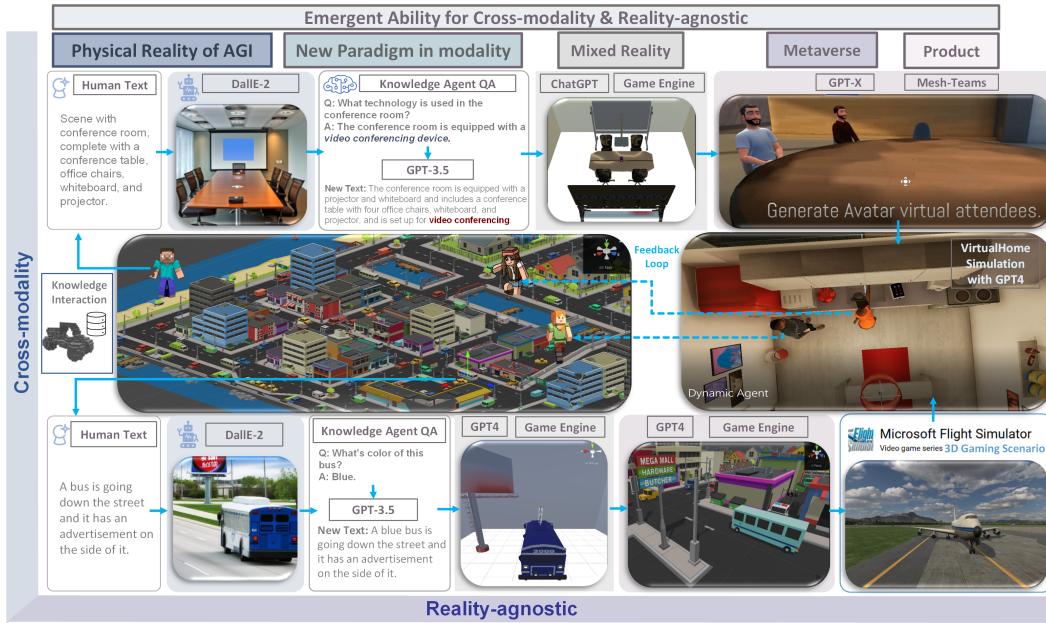


Figure 1: The observations about knowledge emergent infrastructure. The ArK pipeline shows the emerging infrastructure for the cross-modality in a requested unseen environment, and loads the accountability in reality-agnostic scenario automatically with a new paradigm of generative knowledge agent. Large foundation models trained for 3D/2D generation can be applied to speaking virtual worlds into existence (co-creating with the knowledge-memory agent in physical worlds) when integrated with a game engine and 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.

*Equal contribution. Work done when Jae Sung, Abhinav, Pan, and Ran interned at Microsoft Research, Redmond.

Abstract

Despite the growing adoption of mixed reality and interactive AI, it remains challenging to generate high-quality 2D/3D scenes in unseen environments. Typically, an AI agent requires collecting extensive training data for every new task, which can be costly or impossible for many domains. In this study, we develop an infinite agent that learns to transfer knowledge memory from general foundation models (e.g., GPT4, DALLE) to novel domains or scenarios for scene understanding and generation in physical or virtual worlds. Central to our approach is the interactive emerging mechanism, dubbed *Augmented Reality with Knowledge Emergent Infrastructure (ArK)*, which leverages knowledge-memory to generate scenes in unseen physical worlds and virtual reality environments. The knowledge interactive emergent ability (Figure 1) is demonstrated through *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 ArK’s effectiveness in scene generation and editing tasks and show that our ArK approach, combined with large foundation models, significantly improves the quality of generated 2D/3D scenes, highlighting its potential in applications such as metaverse and gaming simulation.

1 Introduction

There has been a growing amount of work on using large language models (LLMs) and large multi-modality models (LMMs) to generate high-quality videos and images from textual inputs [34, 51]. However, it remains challenging for users (creators) to control the generation process and interactively edit generated results if they don’t meet users’ intent. We envision a future AI system where creators can interactively create a virtual reality scene with objects existing or not in the real world, and the system responds faithfully using knowledge from training data of real-world tasks. For example, an interactive AI agent can incorporate contextual memory and background information, pertaining to a task, by transferring knowledge from pre-trained LLMs/LMMs and multi-sense information from sensors during task performance. Foundation models like DALLE-2 [30] and ChatGPT [23] excel in multimodality and natural language reasoning tasks but face limitations in mission-critical real-world applications (e.g., Bing-search, business analysts, office users). Specifically, existing LLMs struggle to effectively transfer knowledge from training data to new mission-critical tasks [24] or solve complex real-world tasks requiring collaborative reasoning between humans and AI agents.

To facilitate human-AI interaction, we develop 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 (Figure 2). 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 knowledge and memory.

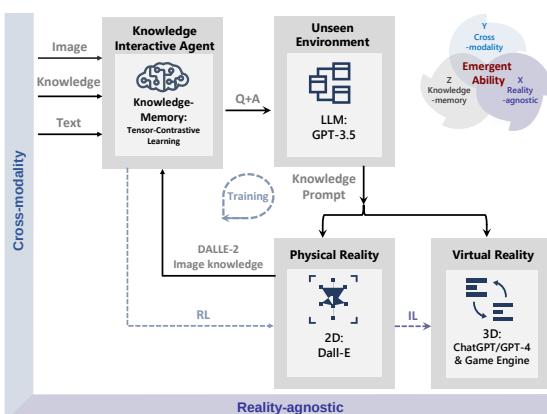


Figure 2: Example of ArK Interactive Emergence Mechanism using an external knowledge agent to identify text relevant to the image from candidates. Our task involves using visual and text knowledge from the web and human-annotated knowledge samples to incorporate external world knowledge.

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 and synthesizing knowledge and memory for scene understanding and generation in different settings.

Our contributions are: i) We develop an infinite knowledge-memory agent for scene understanding and generation in the physical world, with the capabilities of learning knowledge properties and inference relations in virtual reality environments; ii) We show that the effectiveness of our agent is attributed to the proposed ArK mechanism with reinforcement learning (RL), which understands and generates scenes in unseen settings by synthesizing world knowledge from foundation models, external knowledge from knowledge bases (e.g. wiki, Conceptnet), and contextual memory from human-AI interactions; iii) We simulate 3D virtual scenes with imitation learning (IL) in gaming/VR scenarios from the 2D knowledgeable purpose applications (2D->3D) in cross-reality. We present experiments and analysis to demonstrate the effectiveness of our approach; and iv) We observed that the explosion of the overall model works efficiently in the cross-modality and agnostic reality, which depends on the effect of emergent ability in large foundation infrastructure. It enhances the interpretation of the existing deep learning model, optimizes the limitations of the unseen environments, and unifies the abundant knowledge-memory projection in a generative AI system.

2 Related Work

Emerging mechanism in LLMs. Emergent abilities in LLMs 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 [44]. Another type of emergent ability is the ability to translate languages. The exact mechanisms by which LLMs develop these emergent abilities are not fully understood [34, 51]. 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 similar to human-generated text and translate languages accurately.

Language transformer models with knowledge inference. Numerous papers have injected knowledge into language pretraining models [50, 47, 33, 54, 8, 46, 9, 1] with an emphasis on NLP tasks. For example, [50] extracts knowledge graph information from Wikipedia, and uses it to help the pretraining progress. [47] injects domain-specific knowledge in pertraining language models for NLP tasks. These methods focus on language tasks, and have not been extended to multi-modal transformers. More recently, KRISP ([22]) was proposed to retrieve implicit knowledge stored in pre-trained language models as a supplementary knowledge resource to the structured knowledge base. MAVEx ([45]) presented an answer validation approach to make better use of the noisy retrieved knowledge. 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 Approach

The framework of interactive text to 3D scene generation is shown in Figure 3, 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 writing 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.

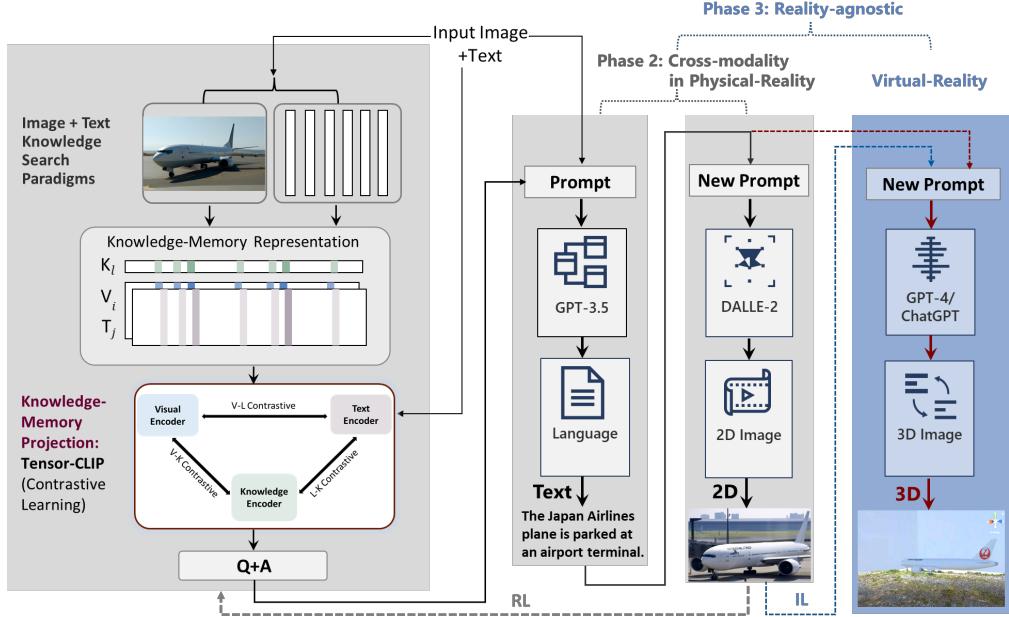


Figure 3: The ArK model: At training time, the agent retrieves relevant knowledge for the given image-text pair and asks a question and answer for the query. The question and answer are provided to large language models (GPT-3.5, ChatGPT) that generate a new prompt that would be called by the DALLE-2 model. The similarity between the generated image and the original image is 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 the 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 engine such as Unity. Overall, we use the external knowledge and the visual priors from the generated 2D image to improve the 3D scene generation.

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.

3.1 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 finetuned 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 4 and Figure 13, 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 [28] visual encoder model, and the text and knowledge encoder are initialized with the text encoder model.

To extract knowledge, we follow KAT [7], 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 maximum inner product search (MIPS) index is then built using FAISS [11] 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 training². Thus, we create fixed index embeddings using the frozen knowledge encoder model.

²Because the image and text encoders in CLIP have been pre-trained with contrastive objective, running MIPS at initial training will still retrieve relevant knowledge.

To train the model, we use the contrastive losses used in [28]. 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 i th batch, we retrieve the top- k knowledge from image K_{v^i} and text K_{t^i} 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:

$$L_{v2k} = \sum_{i \in B} \log \frac{\sum_{k \in K_{v^i}} u_k^T v_i}{\sum_{k \in \{K_{v1}, \dots, K_{vi}, K_{vB}\}} u_k^T v_i}, L_{t2k} = \sum_{i \in B} \log \frac{\sum_{k \in K_{t^i}} u_k^T w_i}{\sum_{k \in \{K_{t1}, \dots, K_{ti}, K_{tB}\}} u_k^T w_i} \quad (1)$$

$$L_{cont} = aL_{v2t} + bL_{t2v} + cL_{v2k} + dL_{t2k} \quad (2)$$

Following [42], 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.

$$L_{mask} = L_{MIM} + L_{MLM} + L_{MVLM} \quad (3)$$

The final loss to train the Knowledge-Tensor CLIP module is $L = L_{cont} + L_{mask}$. We refer to Section C and Section D 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 $W_{0, \dots, p}$ with parser tools such as Spacy, and acquire p phrase embeddings $e_{0, \dots, p}$. 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 Knowledge-Tensor-CLIP model on different dataset, and show the result in Section 4.1.

3.2 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 content for physical QA-2D scene generation. After the agent retrieves the relevant knowledge for the given image, and 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 a 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 [38] to train the agent with the reward from the similarity between the original image and image generated with knowledge enhanced prompt (Red direction in Figure 3. The image is generated with DALLE 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 15 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 a Seq2Seq objective that asks the relevant question and answer in a 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 16 for the prompt template). This Phrase is to prepare for the virtual 3D scene generation, which with the 2D retrieved image from DALLE, 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 [38] to train the agent with the reward R from the cosine similarity between the DALLE retrived image V using the original text and image generated with knowledge enhanced prompt \tilde{V}_k measured by the CLIP ViT-B16 visual encoder model (Red direction in Figure 3. 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:

$$\tilde{T}_k = \text{GPT-3.5}(T, qa_k) \quad \text{where } qa_k = \pi_\theta(K(V, T), V) \quad (4)$$

$$\tilde{V}_k = \text{DALLE-2}(\tilde{T}_k) \quad (5)$$

$$R(V, T) = \cos(CLIP(V), CLIP(\tilde{V}_k)) \quad (6)$$

The agent is then trained via reinforcement learning to incorporate feedback using the reward. We use the actor-critic algorithm PPO [35] to update the parameters of the agent using its clipped version:

$$L_{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(R_t(\theta) \hat{A}_t, \text{clip}(R_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t \right) \right] \quad (7)$$

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

3.3 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 (Figure 8), while GPT-3.5 generates new knowledge-enhanced prompt using the agent output.

Knowledge Category	Semantic				Encyclopedic		Commonsense			Open-World
	Coco	Flickr 30K	WIT		Sherlock		VisualCOMET	AOKVQA		
Approach	Text -> Image	Text -> Image	Text -> Image	Text -> Image	Text -> Image	Finetuned Rank \downarrow	Image -> Text	Text -> Image	Mult. Choice Finetuned Acc(%) \uparrow	
Metric (%)	Zero Shot R@1(%) \uparrow	Zero Shot R@1(%) \uparrow	Zero Shot R@1(%) \uparrow	Zero Shot Rank \downarrow	Finetuned Rank \downarrow		Zero Shot Rank \downarrow	Finetuned Acc(%) \uparrow	Mult. Choice Finetuned Acc(%) \uparrow	
w/o Knowledge										
Contrastive	49.7	51.8	42.0	21.1	28.3	28.7	53.8	60.4		
w/ Knowledge (w/o Mask)										
Knowledge-Tensor-Cont. (ours)	49.8	50.9	43.4	16.2	27.4	27.5	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		

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.

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 Objaverse [4] and 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 [31].

$$\theta^* = \arg \max_{\theta} \frac{1}{N} \sum_{i=1}^N \log p_{\theta}(a_i | s_i) \quad (8)$$

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.

3.4 Emergent Infrastructure for Cross-modality and Reality-agnostic Observation

We train the knowledge-memory agent and use an infinite feedback loop with RL in the real world to randomly initialize a policy. However, it does not work well with virtual reality due to difficulties in obtaining initial rewards in 3D environments or the unseen environment, especially in virtual environments where rewards are sparse or terminal. Thus, a superior solution is to use an infinite-memory agent trained through imitation learning (IL), which can learn policies from expert data, improving exploration and utilization of unseen environmental space with emergent infrastructure as shows in Fig. 5 .

Imitation Learning → Generalization. Traditional IL has an agent mimicking an expert demonstrator’s behavior to learn a policy. However, learning the expert policy directly may not generalize well to unseen situations. To tackle this, we propose learning an implicit reward function that captures key aspects of the expert’s behavior, as shown in Phase 2, 3. This equips the infinite knowledge-memory agent with physical-world behavior data for task execution, learned from expert demonstrations. It helps overcome existing imitation learning drawbacks like the need for extensive expert data and potential errors in complex tasks. Our IL approach has two parts: 1) the infinite agent that collects physical-world expert demonstrations as state-action pairs and 2) the virtual environment that imitates the agent generator. The imitating agent produces actions that mimic the expert’s behavior, while the agent learns a policy mapping from states to actions by reducing a loss function of the disparity between the expert’s actions and the actions generated by the learned policy. Rather than relying on a task-specific reward function, the agent learns from expert demonstrations, which provide a diverse set of state-action pairs covering various task aspects. Decoupling in imitation learning refers to separating the learning process from the task-specific reward function, which enables transfer learning, and allowing the policy to generalize across different tasks without explicit reliance on the task-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 → Emergent Behavior. Generalization explains how emergent properties or behaviors can arise from simpler components or rules. The key idea lies in identifying the basic elements or rules that govern the behavior of the system, such as individual neurons or basic algorithms. These interactions of these components often lead to the emergence of complex behaviors, which are not predictable by examining individual components alone. Generalization across different levels of complexity allows a system to learn general principles applicable across these levels, leading

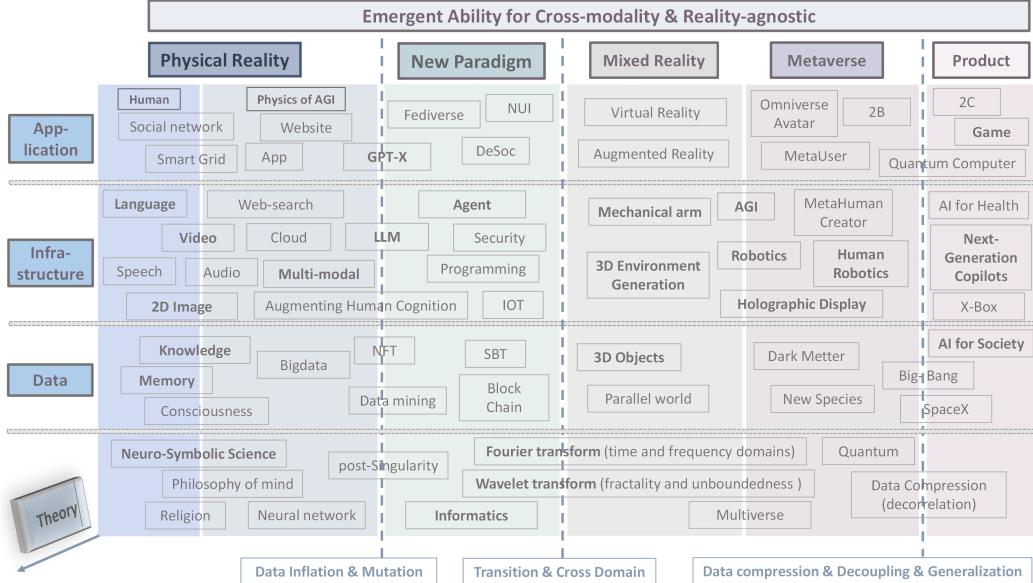


Figure 5: The overview of observations about emergent ability of physical reality, new paradigm, mixed reality, meta-verse, and products for cross-modality and reality-agnostic mechanics from the theory, data, infrastructure, and application of how it blends a plethora of information. It created a new set of methods, visualizations, and controlled experimentation approaches that could become the basis of the new paradigms that will be needed to better understand and improve large-scale model capabilities on emerging tasks.

to emergent properties. This enables the system to adapt to new situations, demonstrating the emergence of more complex behaviors from simpler rules. Furthermore, the ability to generalize across different complexity levels facilitates knowledge transfer from one domain to another, which contributes to the emergence of complex behaviors in new contexts as the system adapts.

Emergent Ability Observation. We provide more information about the emergent ability of our agent in this section. Fig 5 shows the different emerging capabilities of various types of large foundation models. We present a more generic overview of these in a reality-agnostic scenario with an application to Gaming/AR. Emergent abilities have the potential to revolutionize the way that we interact with computers. They could be used to create new types of applications, such as virtual assistants that can understand and respond to our natural language queries. They could also be used to improve the performance of existing applications, such as search engines and machine translation systems. These abilities can be seen in a multimodal scenario, where the LLM is able to process and understand information from multiple modalities, such as text, images, and audio. For example, an LLM can be trained to translate text from one language to another, but it can also be trained to translate images from one language to another. In this case, the LLM is able to use its knowledge of both languages and its understanding of images to generate accurate translations.

4 Experiments and Results

4.1 Knowledge Agent Training

Knowledge Tensor-CLIP training implementations. We finetune the Knowledge-CLIP model on the WIT dataset [37] with the filtered version to only consider English texts, totaling 5M in training data and 30K on test data. We use Wikidata [41] and ConceptNet [19] as the explicit knowledge source as the bridge to connect the image and text modalities. The Knowledge-CLIP model is trained with a batch size of 2048, image size of 224, and the Adam optimizer [13] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 1e - 6$ 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 are in Section D of Appendix.

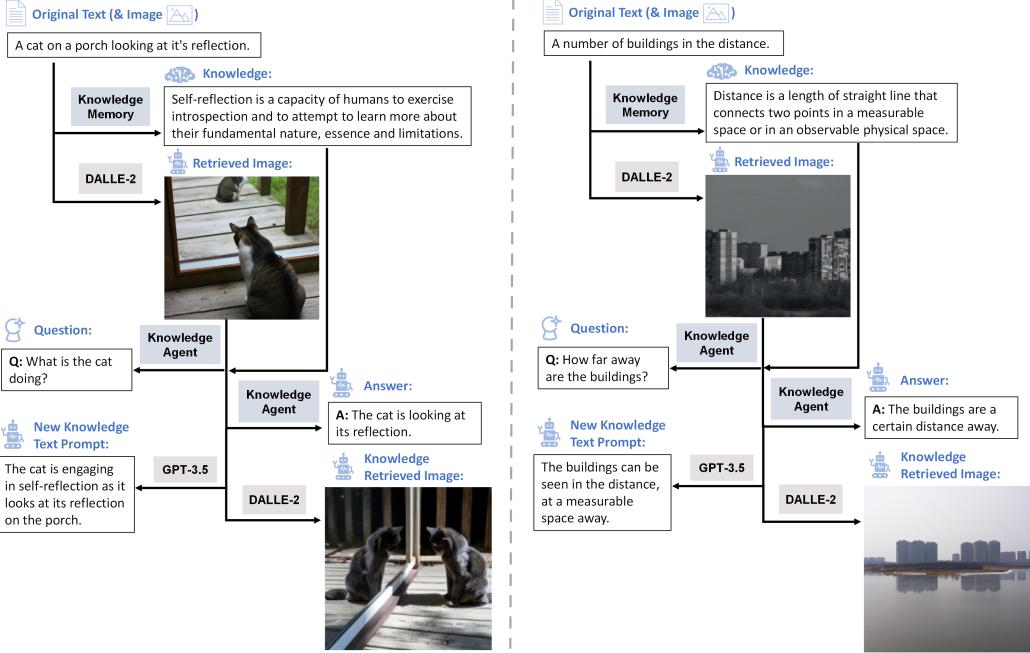


Figure 6: Qualitative examples of conversational-2D Image generation with Knowledge Enhanced Prompts.

Knowledge-memory retrieval system and evaluation. We first evaluate the performance of the knowledge retrieval system on the WIT dataset [37], Coco [18], Flickr 30K [27], Sherlock [10], and AOKVQA [36] 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 [37]. 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 [28], *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 3 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 show analysis plots for the loss tendency of training infinite knowledge-memory agents in Appendix F. We see that our knowledge-contrastive-training model provides improvement over the contrastive data, for the dataset that requires entity-based knowledge.

4.2 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 the validation set of AOK-VQA [36] dataset to perform the text to 2D scene generation. To train the question and answer model, we initialize the model with BLIP-large [15] and finetune on the AOK-VQA data in a SeqSseq objective with a learning rate of $2e^{-6}$, batch size of 128, and for 10 epochs. Policy gradient [38] is used to train the agent after finetuning on AOKVQA data, and the reward is calculated

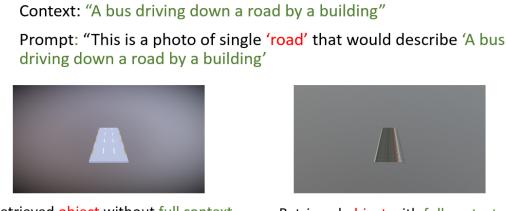


Figure 7: Comparison of the retrieved object with and without context. We see that prompting CLIP with the full context retrieves a more appropriate object (asphalt road).

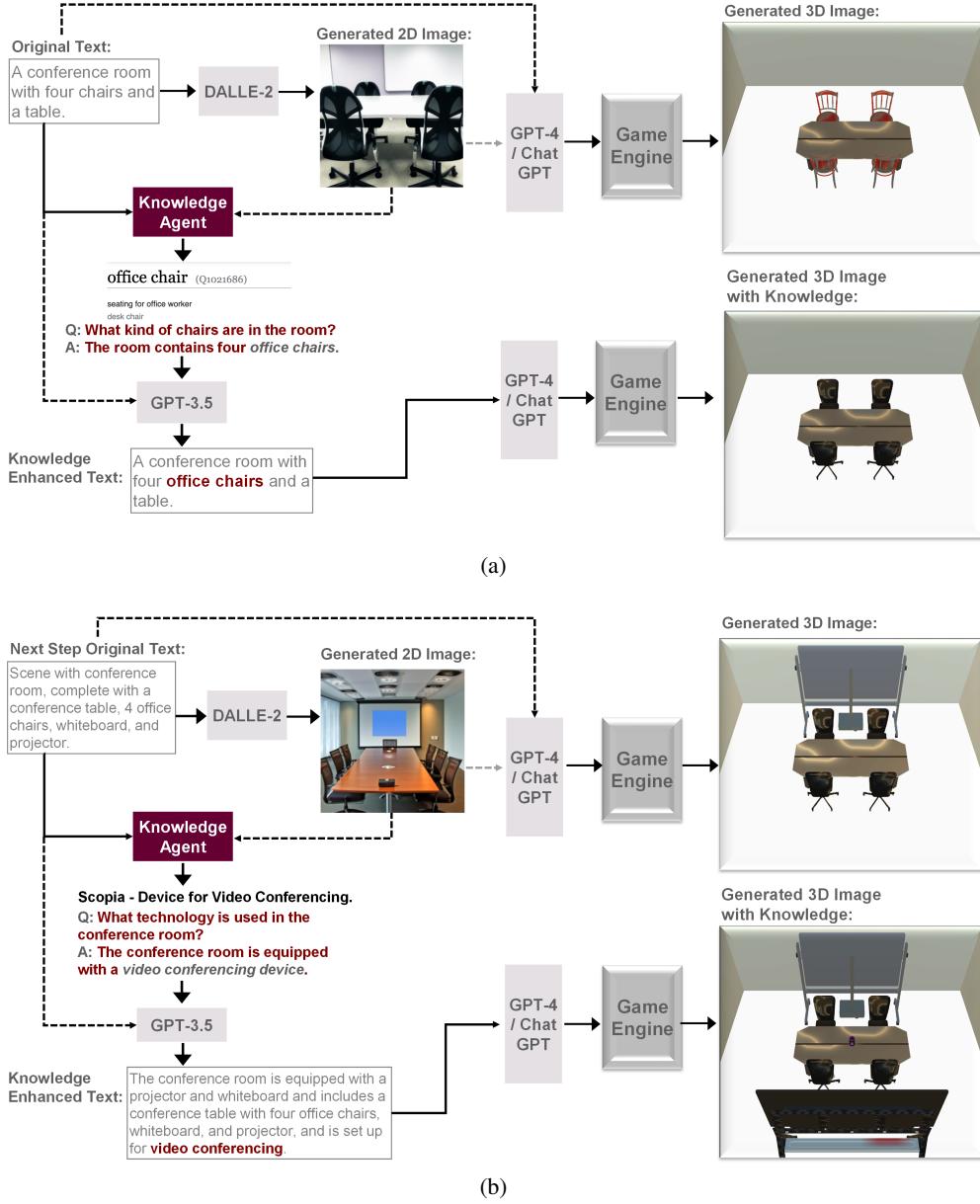


Figure 8: One more example of qualitative examples of 3D scene editing with knowledge enhanced prompts. At inference time, we first generate an image from the input text to learn the prior . The knowledge agent then generates a question and answer tuple which is fed as an input to GPT-3.5. The output 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.

by the CLIP-VIT-base similarity score. Davinci-003 GPT-3.5 model is used to generate the new knowledge prompts for 2D images, and DALLE-2 images are generated with 256×256 resolution, which we used to generate 2D images and the feedback image for the RL algorithm.

QA-2D image generation and evaluation. Figure 6 shows an example of DALLE-2 generated images with original text and knowledge-incorporated text query with an infinite-memory agent. Please refer to more examples of conversational knowledge-2D generation in Appendix ???. By modifying the text query in the zero-shot setting while keeping the OpenAI models frozen, we are

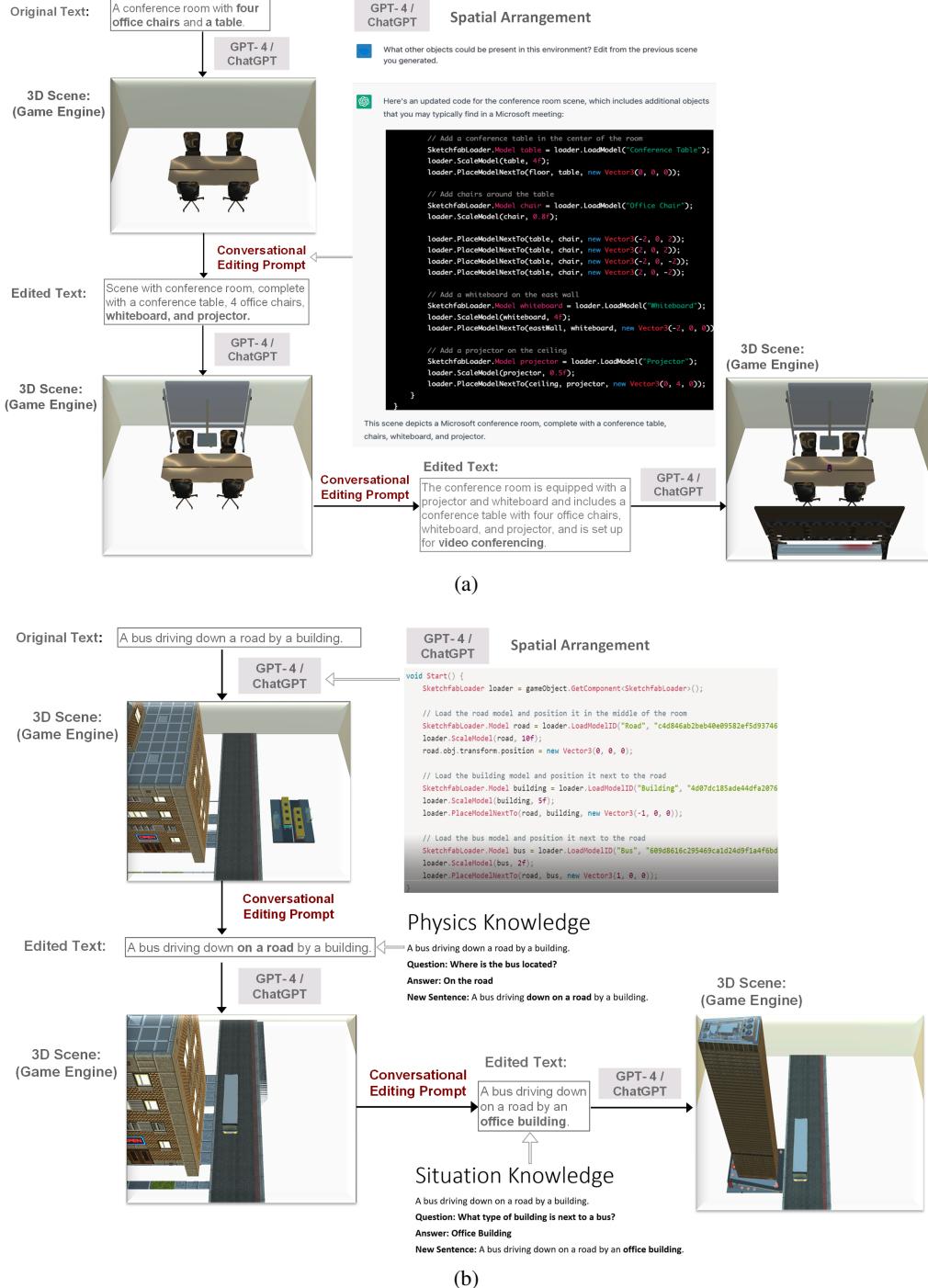
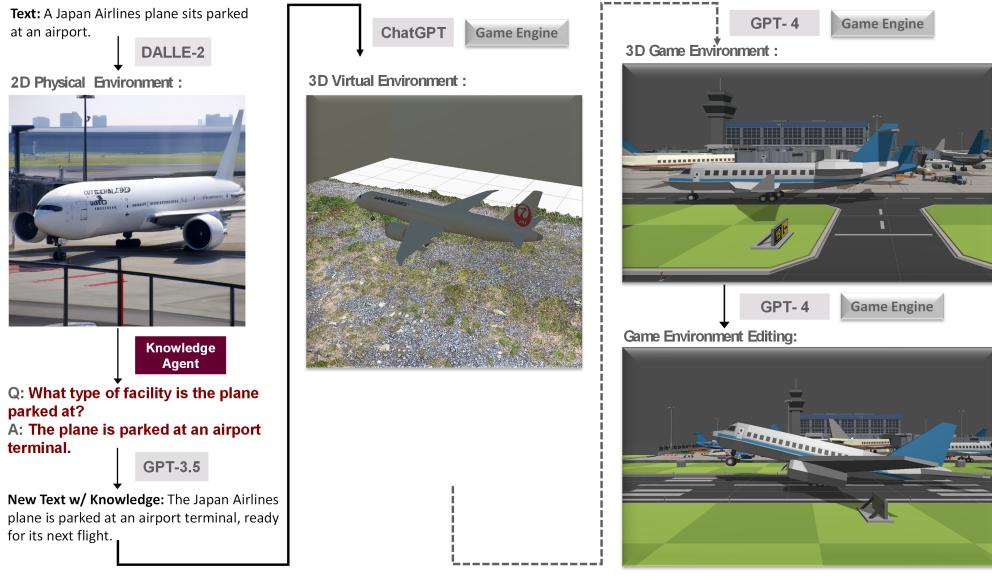


Figure 9: One more example of editing 3D scene with our trained dialogue interactive scenes projection using GPT4/ChatGPT. We see that GPT4/ChatGPT adds more knowledge context with the spatial arrangement to the original text as shown in the edited text.

able to generate more informative and realistic images. For example, we see a more natural portrait of a cat drinking water from the sink and a man looking at himself in the mirror. The last example includes the unmentioned entity of the GS workstation, but DALLE-2 is able to generate a more realistic scene using the entity information. Please find human-evaluation results of the convectional 2D image generation in Table 2.



(a)

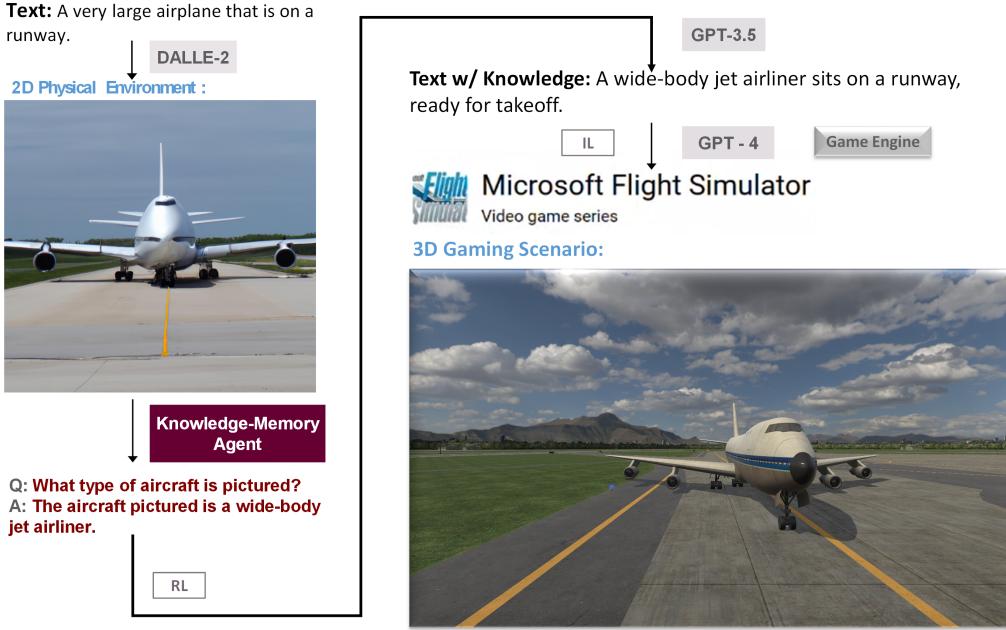


(b)

Figure 10: One more example of Cross-modality and reality-agnostic generation and editing with interactive agent using GPT-4 and ChatGPT. Another examples please find in Appendix.

4.3 Virtual Environment Scene Generation and Editing

Loading the relevant 3D objects. [31] 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 [4] 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 [28] 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 7.



(a)



(b)

Figure 11: 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.

3D scene generation with knowledge. Figure 8 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 how the scene would look like in a real-world setting. Using the generated image and the original text, the agent retrieves explicit knowledge and asks relevant question and answer. Consequently, GPT-3.5 adds relevant knowledge that makes the prompt more informative and realistic in this scene. We use GPT-4/ChatGPT for the spatial arrangement and the program synthesis generation for 3D scenes with the prompt. We next present qualitative results of 3D scene generation with the enhanced knowledge. We finally use GPT4 to generate the knowledge-enhanced prompt of spatial arrangement and programming which is then

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

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

rendered in the Unity game engine. Note that the OpenAI models (colored in pink) are kept frozen while we only train the agent to perform 3D scene editing. As shown the simulation of the Figure 8, 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. 9 shows an example in which we provide the previously generated the spatial arrangements and low-level program synthesis, and ask GPT4 to fill relevant objects with the new prompt. We observe that GPT4 is able to understand the previous scene and adds relevant environment objects such as whiteboard and projector in the appropriate orientation and location.

Observation of emergent infrastructure. 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 loading and the spatial arrangements and the program synthesis for 3D scene generation. Now this object when further as an input to GPT-4 which adds context to the 3D object by changing the surroundings and appearance of the objects according to the specifications of the scene. Figure 10 shows examples of VR editing and a novel approach for generating 3D game scenes to user-shared conversational interactive QA. In contrast to traditional 2D image generation and image-grounded dialogue tasks, we simulate synthesizing 3D-gaming generation and editing content that is relevant and natural with the emergent ability of foundation models for reality-agnostic in generative AI.

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. 11.

4.4 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 AMT, a 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 2. 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 necessary with 3D scene generation with imitation and conversational dialogue way.

5 Conclusion

In this paper, we explore the infinite knowledge-memory agent to enhance large foundation models for physical and virtual reality scene generation, finding that incorporating knowledge into new prompts improves generated environments with RL and IL. We investigate knowledge-guided interactive synergistic effects for collaborative scene generation with large foundation models and show promising promising results in improving 2D and 3D scene generation and editing. We discuss emergent capabilities in cross-modality models and reality-agnostic scenarios, which enhance the depth of generalization and interpretability in complex adaptive AI systems, particularly in generative AI for metaverse and gaming simulation.

References

- [1] Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. *arXiv:2010.12688*, 2021.
- [2] Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. Fusion of detected objects in text for visual question answering. *Proceedings of EMNLP*, 2019.
- [3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. *Proceedings of ECCV*, 2019.
- [4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *ArXiv*, abs/2212.08051, 2022.
- [5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [6] François Gardères, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. Conceptbert: Concept-aware representation for visual question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.
- [7] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. In *NAACL 2022. Long paper, Oral*. *arXiv:2112.08614*, 2022.
- [8] Bin He, Xin Jiang, Jinghui Xiao, and Qun Liu. Kgplm: Knowledge-guided language model pre-training via generative and discriminative learning. *arXiv:2012.03551*, 2020.
- [9] Bin He, Di Zhou, Jinghui Xiao, Xin Jiang, Qun Liu, Nicholas Jing Yuan, and Tong Xu. Bert-mk: Integrating graph contextualized knowledge into pre-trained language models. *Proceedings of ACL*, 2020.
- [10] Jack Hessel, Jena D Hwang, Jae Sung Park, Rowan Zellers, Chandra Bhagavatula, Anna Rohrbach, Kate Saenko, and Yejin Choi. The Abduction of Sherlock Holmes: A Dataset for Visual Abductive Reasoning. *arXiv preprint arXiv:2202.04800*, 2022.
- [11] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.
- [12] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. *arXiv:2102.03334*, 2021.
- [13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [14] Gen Li, Nan Duan, Yuejian Fang, Dixin Jiang, and Ming Zhou. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *Proceedings of AAAI*, 2020.
- [15] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022.
- [16] Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*, 2019.
- [17] 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. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv:2004.06165*, 2020.

- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context. *Proceedings of ECCV*, 2014.
- [19] Hugo Liu and Push Singh. Conceptnet—a practical commonsense reasoning tool-kit. In *BT technology journal*, 22(4):211–226, 2004., 2004.
- [20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [21] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Proceedings of NeurIPS*, 2019.
- [22] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *The 34th Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [23] OpenAI. Chatgpt, 2022.
- [24] Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. Check your facts and try again: Improving large language models with external knowledge and automated feedback. *arXiv preprint arXiv:2302.12813*, 2023.
- [25] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv*, 2022.
- [26] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAAACL*, 2018.
- [27] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *ICCV*, 2015.
- [28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.
- [29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 2019.
- [30] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents, 2022.
- [31] Jasmine Roberts, Andrzej Banburski-Fahey, and Jaron Lanier. Steps towards prompt-based creation of virtual worlds. *ArXiv*, abs/2211.05875, 2022.
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.
- [33] Corby Rosset, Chenyan Xiong, Minh Phan, Xia Song, Paul Bennett, and Saurabh Tiwary. Knowledge-aware language model pretraining. *arXiv:2007.00655*, 2021.
- [34] 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. Photorealistic text-to-image diffusion models with deep language understanding, 2022.
- [35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.
- [36] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. *arXiv*, 2022.

- [37] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*, 2021.
- [38] Richard S. Sutton, David A. McAllester, Satinder Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*, 1999.
- [39] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. *Proceedings of EMNLP*, 2019.
- [40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.
- [41] Denny Vrandecic and Markus Krotzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 2014.
- [42] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. 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*, 2023.
- [43] Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhengyan Zhang, Zhiyuan Liu, Juanzi Li, and Jian Tang. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics*, 9:176–194, 2021.
- [44] 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. Emergent abilities of large language models, 2022.
- [45] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *arXiv preprint, arXiv:2103.12248*, 2021.
- [46] Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. *arXiv:1912.09637*, 2019.
- [47] Song Xu, Haoran Li, Peng Yuan, Yujia Wang, Youzheng Wu, Xiaodong He, Ying Liu, and Bowen Zhou. K-plug: Knowledge-injected pre-trained language model for natural language understanding and generation in e-commerce. *arXiv:2104.06960*, 2021.
- [48] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. An empirical study of gpt-3 for few-shot knowledge-based vqa. In *AAAI*, 2022.
- [49] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *NeurIPS*, 2019.
- [50] Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. Jaket: Joint pre-training of knowledge graph and language understanding. *arXiv:2010.00796*, 2020.
- [51] 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. Scaling autoregressive models for content-rich text-to-image generation, 2022.
- [52] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. *arXiv:2101.00529*, 2021.
- [53] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa. *Proceedings of AAAI*, 2019.
- [54] Wangchunshu Zhou, Dong-Ho Lee, Ravi Kiran Selvam, Seyeon Lee, Bill Yuchen Lin, and Xiang Ren. Pre-training text-to-text transformers for concept-centric common sense. *arXiv:2011.07956*, 2020.

Supplementary Materials for ArK: Augmented Reality with Knowledge Emergent Infrastructure

A Related Work

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 [40] have achieved impressive performance by pretraining representations for a wide range of natural language processing (NLP) tasks [26, 5, 49, 20, 29]. Recent work on vision-language pretraining (VLP) has shown that these large-scale pretraining methods can also be used for effective cross-modal representations [21, 39, 53, 3, 2, 14, 16, 17, 52, 12]. 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, [21, 39] propose multi-stream Transformer-based frameworks with co-attention to fuse these modalities. [53, 3, 2, 14, 16, 17, 52] propose unified pretrained architectures to work on both visual-language understanding and visual-language generation tasks. [6] uses ConceptNet knowledge graph as a knowledge base in order to facilitate commonsense vision-language question-answering. [12] 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.

B Overview of the model.

This study focuses on developing an interactive AI agent for scene understanding and generation, powered by pre-trained foundation models (e.g., DALLE-2, ChatGPT) as shown in Fig 12. 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.

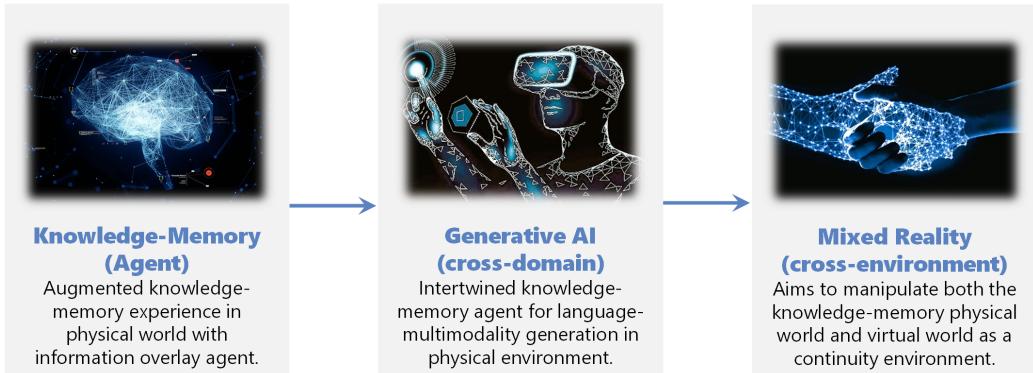


Figure 12: Overview of the generative AI with the knowledge memory agent in the physical world and virtual environment.

C Knowledge Source

C.1 Implicit Knowledge Source from OpenAI Models

Text retrieval knowledge: large language models (GPT-3.5). [48] 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 [7] 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 [30] and Stable Diffusion [32] 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.

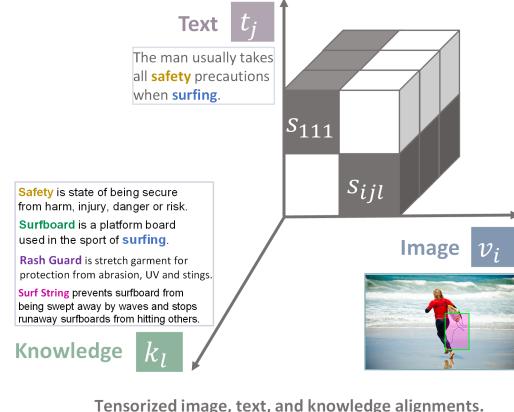
C.2 Explicit Knowledge Source from Web Knowledge Bases

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 [41] 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 [43]³ and further filter non-English entities, resulting 3,836,524 sentences in total.

Commonsense knowledge: ConceptNet. ConceptNet [19] 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 5.5⁴, 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.

V+L+K Tensor Contrastive Training



Tensorized image, text, and knowledge alignments.

Figure 13: 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 OpenAI foundation models, and incorporating external web-search knowledge pool about the world.

³<https://deepgraphlearning.github.io/project/wikidata5m>

⁴<https://huggingface.co/datasets/conceptnet5>

D Knowledge-Tensor CLIP Pre-Training Details

As showed in the figure 13, 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 [42], 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 [25]. The effect of masked loss is shown in Table 1, which provides a slight boost in the downstream task evaluations.

E Evaluation of Infinite Knowledge Agent

In Table 3, we evaluate our knowledge agent on the WiT dataset for the Text to Image Retrieval task and show the Text to Image recall metric.

Model	Text to Image (R@1)
[37]	34.4
CLIP [28]	42.0
Knowledge-CLIP (Ours)	43.4

F Analysis of Knowledge Agent Training on losses

We analyze the effect of knowledge on both types of loss functions i.e. language loss and image loss. The results are shown in Fig 14. We observe that knowledge helps in improving the image loss much better than the language loss. This can be attributed to knowledge providing better learning signal for 2D image generation and hence results in producing more realistic images.

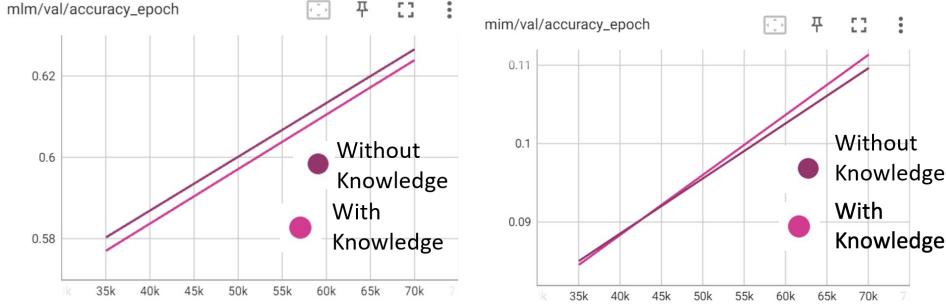


Figure 14: Effect of knowledge on both language (left) and image (right) losses.

G Prompt for Knowledge-Memory Agent

G.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 15. 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.

G.2 Prompt for Knowledge-Enhanced Query Generation

Figure 16 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.

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 15: 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.

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

In Figure 17, 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.

We show more examples for generating the 2D images using the infinite knowledge-memory agent in Table 4.

H Human Evaluation Details

In Fig. 18, 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 4.4 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.

I RL Traning Analysis

In Fig. 20, we show the sum of rewards (divided by 12) obtained when a batch of examples (batch size: 64) was trained using the process described in Sec 3.2. As expected, the expected reward increases as the training progresses showing that the image generated gets more closely grounded with the images present in the dataset.

J Limitations

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

We present cross-model results to generate 3D objects that can be used in a gaming scenario for eg, with a Microsoft Gaming Simulator. We believe one limitation is access to large language models like GPT-4, DALLE-2 which might be expensive when trying to finetune on other datasets. Although in this work, we used AOKVQA dataset for finetuning which has questions that require applying common-sense reasoning and general perception about the real-world.

K Broader Impacts

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 empathy 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.

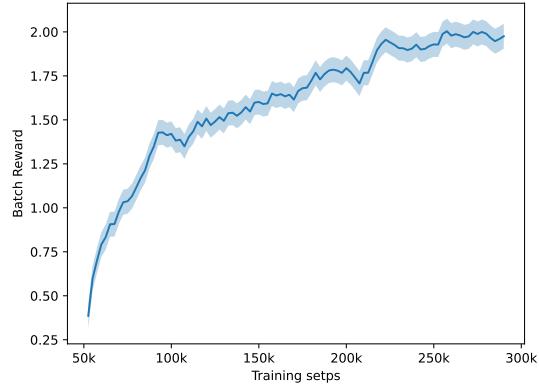


Figure 20: RL Training curve showing the cumulative batch reward calculated according to Eq 6 downscaled by a factor of 12.

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 an 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. */""
```

Instruction

Example

Task

Figure 17: An example of the prompts for VR and game scene generation and editing which we use GPT4/ChatGPT in our knowledge-augmented query system.

Text	Knowledge	Q.A	New Prompt	Image
Cat sleeping in front of a powered on laptop computer.	computer is a general-purpose device for performing arithmetic or logical operations	Q: What is the device in the image? A: The device in the image is a laptop computer.	The cat is peacefully sleeping in front of a powered on laptop computer.	
A group of people riding skis on a snowy surface	skiing as transport or recreation is a the use of (snow)skis for transport purposes or as recreation	Q: What activity is the group participating in? A: The group is skiing as a form of transport or recreation.	The group is skiing on the snow-covered surface as a form of transport or recreation.	
A woman is holding a slice of red and white cake	sweetheart cake is a food	Q: What type of food is the woman holding? A: The woman is holding a slice of sweetheart cake.	The woman holds a slice of sweetheart cake, decorated in red and white.	
A person jumping up in the air on a skateboard.	street skateboarding is a style of skateboarding	Q: What type of skateboarding is the person doing? A: The person is doing street skateboarding.	The person is doing a street skateboarding trick, jumping up in the air on a skateboard.	

Table 4: Knowledge based 2D image generation examples in physical world.

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 18: 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 19: Two examples of human evaluation for the conversational 3D VR Scenario. One is for single camera example; another is for the three cameras examples.