

Crafting Dynamic Virtual Activities with Advanced Multimodal Models

Changyang Li*
Goertek Alpha Labs

Qingan Yan†
Goertek Alpha Labs
Yi Xu‡
Goertek Alpha Labs

Minyoung Kim‡
George Mason University
Lap-Fai Yu||
George Mason University

Zhan Li§
Goertek Alpha Labs



Figure 1: Given a virtual environment as the input, our method leverages multimodal large language models (MLLMs) to automatically generate contextually realistic character interactions. s_t denotes scene snapshots at discrete keyframe t . s_4 here features two characters engaging in a conversation, an elderly female watching TV, and another character participating in a game of darts.

ABSTRACT

In this paper, we investigate the use of multimodal large language models (MLLMs) for generating virtual activities, leveraging the integration of vision-language modalities to enable the interpretation of virtual environments. Our approach recognizes and abstracts key scene elements including scene layouts, semantic contexts, and object identities with MLLMs’ multimodal reasoning capabilities. By correlating these abstractions with massive knowledge about human activities, MLLMs are capable of generating adaptive and contextually relevant virtual activities. We propose a structured framework to articulate abstract activity descriptions, emphasizing detailed multi-character interactions within virtual spaces. Utilizing the derived high-level contexts, our approach accurately positions virtual characters and ensures that their interactions and behaviors are realistically and contextually appropriate through strategic optimization. Experiment results demonstrate the effectiveness of our approach, providing a novel direction for enhancing the realism and context-awareness in simulated virtual environments.

Index Terms: Virtual humans, human-scene interaction, behavior synthesis, multimodal large language models

1 INTRODUCTION

The generation of virtual environments and the simulation of human activities within these spaces are increasingly important in computer graphics, computer vision, and robotics. Recent advances, including the availability of large-scale datasets and sophisticated deep generative models, have significantly improved the capability to produce lifelike human-scene interactions.

However, existing approaches face several challenges. Many current methods primarily address the placement of virtual characters [77, 75, 19, 78] based on low-level geometric and physical constraints, without adequately incorporating high-level contextual information such as the nature of the activities performed. Additionally, although task- or instruction-specific motion synthesis methods [6, 61, 65, 66, 18, 72, 20, 27, 64, 79] have emerged, they often overlook the dynamic interplay among multiple characters and their interactions with surrounding objects, limiting realism and scenario complexity. Addressing these limitations demands methods capable of interpreting and leveraging the semantic context and spatial layout within virtual scenes, and correlating such clues with human activities and behaviors.

Recent advances in language models, particularly those employing deep learning and transformer architectures [63] and supporting multimodality [55, 56, 2, 42], have revolutionized the field of natural language processing and computer vision. These models demonstrate remarkable capabilities in understanding and generating complex language and vision contexts. The integration of linguistic strengths into multimodal large language models (MLLMs) presents new opportunities for enhancing virtual activity generation: Leveraging advanced linguistic reasoning and visual understanding, MLLMs can interpret environments, understand character dynamics, and generate contextually relevant interactions.

Motivated by these developments, our research aims to utilize the potential of MLLMs in generating virtual activities, especially for multi-character human-scene interactions involving multiple characters. By integrating vision-language modalities through a Layout Chain-of-Thought (L-CoT) prompting strategy, we enable MLLMs to interpret virtual environments visually, understand semantic contexts, infer spatial relationships, and perform logical reasoning when generating human activities. We propose a structured framework that explicitly defines virtual activities in terms of character poses, positional references, and detailed interactions on discrete keyframes. Furthermore, our method employs an efficient optimization with a Markov chain Monte Carlo (MCMC) sampling process to precisely position virtual characters, ensuring natural, contextually consistent interactions aligned with their generated high-level activity descriptions.

*e-mail: changyang.li@goertekusa.com

†e-mail: qingan.yan@goertekusa.com

‡e-mail: mkim229@gmu.com

§e-mail: zhan.li@goertekusa.com

¶e-mail: yi.xu@goertekusa.com

||e-mail: craigyu@gmu.com

Generating contextually meaningful virtual activities holds great potential in numerous virtual and augmented reality (VR/AR) application scenarios, including training virtual agents for social scenarios, creating enhanced immersive VR and AR experiences, facilitating advanced human-robot interactions, and supporting virtual storytelling and content creation.

The major contributions of our work include the following:

- Proposing a multimodal reasoning approach using MLLMs for detailed interpretations of virtual environments;
- Formulating a structured representation of virtual activities to effectively prompt MLLMs for context-aware, adaptive, and realistic generation of multi-character interactions;
- Devising an efficient optimization strategy for realistic and scalable character placement in virtual scenes.

2 RELATED WORK

2.1 Scene-Adaptive Activity Generation

The generation of scene-aware virtual human behaviors is pivotal for creating realistic virtual environments. A key challenge in this domain is the adaptive generation of virtual human poses within input scenes. Savva et al. explored the relationship between the geometries and functionalities of 3D environments and the corresponding human poses and object arrangements, leveraging observations from human-scene interactions [59, 60]. Some recent methods utilizing deep generative models have focused on posing virtual humans by considering factors such as geometry, semantics, affordances, and contact points [77, 75, 19, 78].

Beyond static snapshots of human poses, the generation of scene-aware human motion considers temporal dynamics and the interactions between humans and their environments, guided by specific tasks [5, 1, 52, 36], action labels [17, 51] or the surrounding environments [6, 61, 65, 66, 18, 72, 20, 27, 64, 79]. There is also a growing interest in integrating virtual humans into real-world environments for augmented reality applications, highlighting the need for adaptive and interactive behaviors [62, 39, 29].

While those methods mainly concentrate on the motion and behaviors of individual characters, our research shifts focus towards scenarios involving multi-character interactions within scenes. Some related researches that study detailed multi-character interactions [30, 76, 37, 69], but overlook the correlations with the surroundings. Crowd simulations provide strategies for driving large groups of characters in environmental contexts; however, these approaches typically do not account for intricate interactions among characters or the higher-level interpretation of the activities being performed [49, 35, 80, 8]. Li et al. [40] addressed a closely related problem by proposing the synthesis of activity snippets that depict sequenced multi-character and multi-object interaction scenarios. In comparison, our work focuses on adapting virtual activities to given environments, rather than jointly creating virtual environments. Whereas activity snippets [40] may face scalability issues in larger group settings without extensive training data, our approach leverages the knowledge embedded in MLLMs to facilitate large-scale, multi-character interactions within extensive environments, demonstrating a novel adaptation of these models in generating contextually rich and dynamic virtual scenes.

2.2 Language Models for Activity Generation

Advancements in language models have significantly influenced research directions such as activity generation and behavior modeling. For example, LLMs can simulate human behaviors within interactive virtual sandbox environments [50, 43]. More recent research has investigated employing language models to generate detailed human motion directly from textual inputs [14, 4, 81, 38], or

to produce motion sequences adapted to specific scenes and contexts [71, 68, 67, 74, 28]. Existing approaches also use scene graphs that capture rich scene context to guide LLMs through high-level activity or task planning [15, 41].

Recent models with vision-language modalities demonstrate great power in various directions, like 3D scene understanding [24, 26, 23, 10, 11, 16] and VR/AR world enhancement [9, 34]. To realize scene-aware activity generation, adopting vision-language models is critical to understanding scene layouts and semantics. These works provide insights into using MLLMs for an enriched understanding and generation of activities within 3D environments. Our approach similarly employs MLLMs to grasp the environmental cues of the target scenes, facilitating the creation of virtual activities that are both contextually aware and semantically compatible.

3 OVERVIEW

Figure 2 shows an overview of our approach. It begins with enabling multimodal large language models (MLLMs) to interpret virtual 3D environments through two types of scene clues: area descriptions and multi-view image observations (Section 5). To facilitate precise visual grounding, we adopt the Set-of-Mark (SoM) [73], which explicitly labels object identities. We then employ a Layout Chain-of-Thought (L-CoT) prompting strategy, guiding MLLMs through explicit reasoning steps to infer spatial adjacencies between areas and construct area-based scene graphs.

Using the interpreted environmental context, we then prompt MLLMs to generate dynamic, context-aware virtual activities involving multiple characters (Section 5.1). Each virtual activity is structured as a sequence of discrete keyframes, where each keyframe describes character poses, positional references, and detailed interactions with objects or other characters.

Finally, to realize these high-level symbolic descriptions within the 3D virtual environments, we apply an efficient optimization strategy (Section 5.2). This procedure leverages Markov chain Monte Carlo (MCMC) sampling to precisely position characters. Animation clips corresponding to the generated descriptions are applied to produce realistic animations of the virtual activities.

4 UNDERSTANDING 3D ENVIRONMENTS USING VISION-LANGUAGE MODELS

In the initial phase, we focus on enabling vision-language MLLMs to understand virtual 3D scenes using combined visual and linguistic information. We specifically use GPT-4o for all results.

4.1 Scene Clues Preparation

We provide MLLMs two types of scene clues: area descriptions and multi-view image observations. These inputs help MLLMs interpret scene layouts, spatial relationships, object shapes, functionalities, and affordances. Area descriptions are later converted to area-based scene graphs by the MLLMs.

Area-based Scene Graph. Scene graphs provide abstracts of spatial layouts to MLLMs. In this work, we define simplified scene graphs conveying only basic structural information, omitting specific numeric or specialized attributes: Instead of using a highly structured scene graph representation with comprehensive details encoded, which demands careful, object-specific design, we leave such details to be understood using vision-language models without requiring explicit, domain-specific attribute definitions.

Specifically, we first group scene objects into distinct clusters called **scene areas**, considering distances and visibilities. Pairwise visibility is determined if the two objects are visible to each other (e.g., not separated by a wall). An area is modeled as a single node in the scene graph. Scene area descriptions are composed into json format as inputs for MLLMs.

To connect these area nodes to build a scene graph, one could apply standard minimum spanning tree (MST) algorithms such as

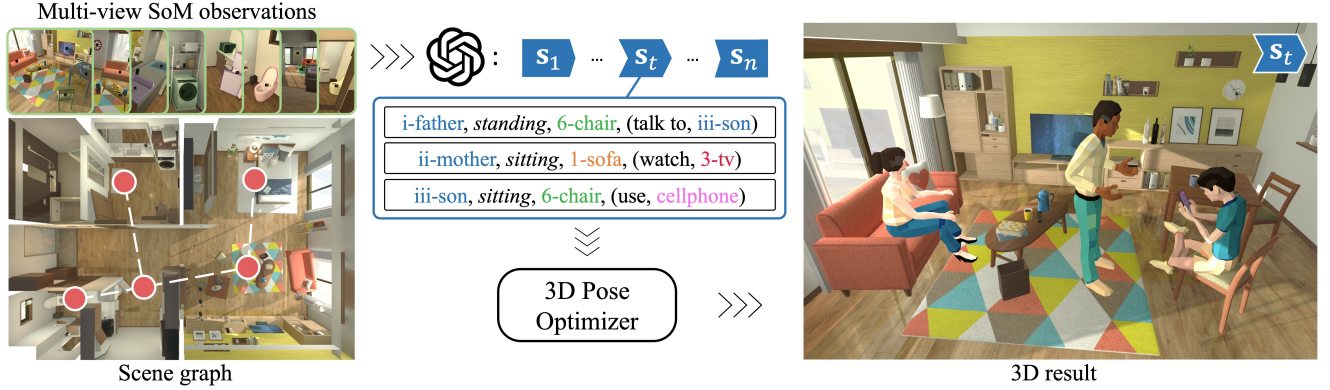


Figure 2: An overview of our approach. We first capture multi-view Set-of-Mark (SoM) [73] observations of the scene, enabling the MLLM to construct an area-based scene graph and interpret spatial layouts. Next, the MLLM generates detailed descriptions of virtual activities at discrete keyframes, specifying characters’ poses, positional references, and interactions. Finally, a 3D pose optimizer positions the virtual characters in the scene. The example shows generated activity descriptions for a selected keyframe (s_t) and their corresponding 3D realization.

Prim’s [53] or Kruskal’s [33]. However, we found that allowing MLLMs to perform this task, guided by visual clues from multi-view images, produces more accurate and spatially coherent results. Given a fixed scene graph with abstract scene information, MLLMs might overlook details that need to be inferred through visual clues, and misinterpret spatial layouts. For example, identical graph distances could correspond to different actual spatial distances. Our ablation tests in Section 6.2 further support this observation. Thus, we defer this step to be finished by MLLMs as an explicit reasoning step after interpreting the visual clues from multi-view images. An example scene graph is visualized in Figure 2.

Multi-view Image Observations. We feed multi-view scene images to MLLMs, which help them recognize scene objects, their functionalities, and affordances. All images are captured by a virtual camera that navigates randomly to ensure comprehensive coverage of the scene. This ensures diverse view angles and improves the MLLMs’ spatial reasoning and contextual understanding.

However, aligning recognized objects precisely with their scene identities remains challenging for current MLLMs. To address this, we employ the Set-of-Mark (SoM) [73] representations, where each object is clearly marked in images with a unique ID label, facilitating consistent identification across different views. The marks are based on instance segmentations sourced from the virtual 3D scenes. Each object within the scene is assigned a unique ID for subsequent reference when the view changes. In practice, we filter out marks based on size, distance, visibility, and overlap criteria, removing those that are too small, overly distant, significantly outside the field of view, or heavily occluded. By default, we remove marks smaller than 4% of the image area, further than 10m from the camera, partially out of view by more than 20%, or substantially overlapped with other marks. In real-world applications, these segmentations for markings could be obtained using state-of-the-art segmentation methods like the Segment Anything (SAM) family [31, 57]. To maintain consistent object identities across different views, 2D segmentation results can be projected onto the 3D digital twin. Alternatively, 3D segmentations can be directly applied, with corresponding 2D segmentation masks rendered for each view.

To reduce information redundancy, we keep a minimum subset of views that still cover all needed scene objects and preserve necessary scene clues. Starting from an empty set, we iteratively select images that cover the largest number of not-yet-covered objects until all objects are covered, and get an initial view coverage set.

Another important consideration is to ensure view traversability, that the views form a connected graph under the rule “two views are adjacent if they share at least one object”. This is an essential consideration for the Layout Chain-of-Thought (L-CoT) we use to

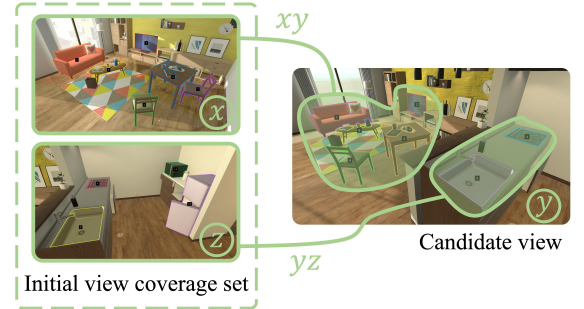


Figure 3: An illustration of maintaining connectivity during the selection of multi-view observations. The initial coverage set includes views x and z covering distinct scene objects. A candidate view y is selected to connect x and z by capturing shared objects.

guide MLLMs in understanding the environments, of which details will be discussed later. Continuing from the initial coverage set, we keep selecting additional views, prioritizing views that connect two or more previously disconnected components, meanwhile maximizing the number of shared objects with the images in those components, as depicted in Figure 3. Note that traversability cannot be guaranteed (e.g. when an area is in a separate room), and in such cases we process the remaining unconnected areas with MST.

4.2 Layout Chain-of-Thought Prompting

Aligning with the principles behind Chain-of-Thought [70], which simplifies complex queries into manageable steps, we propose Layout Chain-of-Thought (L-CoT) for layout understanding. This involves guiding MLLMs through a sequence of reasoning steps: We first require MLLMs to explicitly identify objects associated with their areas according to the area descriptions. Next, MLLMs should list shared objects appearing in different views as an intermediate reasoning step, and further determine the views’ spatial adjacencies. Areas’ adjacencies could be reasoned if one view contains objects from multiple areas, or reasoned from multiple views: For instance, while each view shown in Figure 4 focuses on a distinct small area of the scene, the shared objects (sofa and tea table) suggest the proximity between these areas. Finally, MLLMs should use the reasoned adjacencies as edges to complete the area-based scene graph, given areas defined before as nodes.

L-CoT explicitly guides MLLMs through intermediate reasoning steps to interpret spatial layouts. This reasoning approach improves the model’s ability to accurately infer spatial relationships, thereby enhancing the realism of subsequent virtual activity generation, which also uses L-CoT and is discussed in Section 5.



Figure 4: The proximity between areas captured by two camera views is determined following the Layout Chain-of-Thought (L-CoT) prompting, given their shared common objects.

5 DYNAMIC VIRTUAL ACTIVITIES

In our work, a virtual activity is formally defined as a tuple (C, τ) , summarizing various components essential for activity representation in virtual environments. The set $C = \{c\}$ represents the characters involved in the activity, with each character c is identified by a unique ID and associated with a specific role, which could range from social roles such as a parent to professional roles like a chef. The snapshots $\tau = \{s_t\}_{t=1,2,\dots,T}$ depicts a sequence of scene states across keyframes from 1 to T , thereby discretizing the activity into manageable segments.

5.1 Generating Activity Descriptions with MLLMs

A core aspect of our approach is the generation of snapshots τ of the activity using MLLMs. Each snapshot s_t provides all characters' states at a specific keyframe t , expanded into a series of detailed descriptions such that $s = \{d\}$. A single description d details an individual character's pose and ongoing activity at the given keyframe. We define the following attributes for structuring each description $d = (\alpha, p, r, (v, \beta))$:

1. **Subject α** , which is the focal character of the description.
2. **Pose p** , which distinguishes among three fundamental poses: *standing*, *sitting*, and *lying*. These poses cover a majority of life scenarios as the base, and actual poses can be further detailed with exact body parameters defined in Section 5.2 and specific animation clips (e.g. *sitting + talking*).
3. **Positional Reference r** , which enhances layout comprehension and activity planning through MLLMs by anchoring characters to specific furniture or appliances corresponding to their poses (i.e. *standing near*, *sitting on*, and *lying on*).
4. **Interaction (v, β)** , which enumerates the interactions involving the described character. Each interaction is characterized by a verb v (e.g., "read", "talk to") and a grammatical object β (e.g., object_10-book, character_2) that α interacts with.

This structured format leverages MLLMs' capabilities in interpreting and generating complex scenarios, thereby facilitating detailed and dynamic representations of virtual interactions.

Building on the foundation established in the previous phase, where MLLMs have gained an understanding of the scene layout and objects inside the scene, we next instruct MLLMs to fill out details of the activities occurring within each keyframe. The objective is to enrich the snapshot $s_t = \{d\}$ for every keyframe t by generating the characters' actions and interactions within the scene.

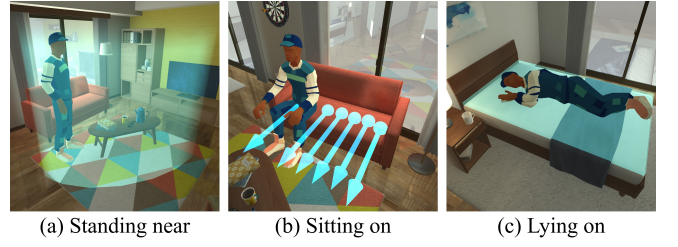


Figure 5: The free spaces considered for different character poses during optimization: (a) *standing near* the furniture, (b) *sitting on* the predicted sitting points, and (c) *lying on* a supporting surface.

For all character in a keyframe, the MLLMs are tasked with detailing their descriptions following the defined format. To ensure accurate spatial reasoning, we prompt MLLMs to explicitly describe intermediate reasoning steps when characters move between different areas, referring to the area-based scene graph and following the Layout Chain-of-Thought (L-CoT) prompting approach. An example prompt illustrating this strategy is shown in Figure 6.

5.2 Accommodating Characters in 3D Scenes

Given the generated activity by MLLMs, we move forward to populate the virtual characters into the virtual environments as specified. This step employs an optimization technique akin to that described in [39, 40], utilizing simulated annealing [32] with a Metropolis-Hastings state-search step [47, 21] to optimize the positioning and poses of characters. The exploration of solution space is conducted via Markov chain Monte Carlo (MCMC) sampling, achieving efficiency and near real-time character population. This efficiency is attributed to the predefined positional references and interaction details, which significantly narrows the potential placement range constrained by the generated high-level descriptions.

A character's pose is parameterized as $(\mathbf{p}, \theta_b, \theta_h)$ conditioned on the fundamental pose, where $\mathbf{p}$ denotes the 3D position, θ_b the body rotation relative to the positional reference r , and θ_h the head rotation relative to the body, with both simplified rotations occurring around an axis perpendicular to the floor. This distinction between body and head rotations mirrors real-life scenarios, such as a person sitting on a chair who may rotate their head to look at someone or something without significantly rotating their body. The spatial allocation for character placement varies with the pose as illustrated in Figure 5: for a *standing* pose, movement is restricted to the proximity of an object; for a *sitting* pose, GoalNet [19] is used to predict multiple sitting points (i.e., positions and forward directions for the hip joint) and orientations; for a *lying* pose, the character is placed on the largest supporting surface available on appropriate furniture.

The placement process involves specific Markov chain dynamics to propose moves during MCMC sampling, detailed as follows:

1. **Translation:** For a *standing* or *lying* character, the position can be adjusted as $\mathbf{p} \rightarrow \mathbf{p} + \delta\mathbf{p}$, following a bivariate normal distribution. Translation occurs on the xy plane, with height fixed during initialization given the positional reference r 's geometry. For the *sitting* pose, a proposed translation moves a character from the occupied sitting point to another nearby.
2. **Body Rotation:** A character's body rotation can be adjusted as $\theta_b \rightarrow \theta_b + \delta\theta_b$, following a normal distribution.
3. **Head Rotation:** Similarly, the head rotation can be adjusted as $\theta_h \rightarrow \theta_h + \delta\theta_h$ following a normal distribution. For the *lying* pose, the head rotation is fixed as 0.

System: You are a visual assistant interpreting 3D scenes containing multiple objects. Inputs provided to you: 1. Scene areas (JSON), each including an “objects” list; 2. Multi-view scene images with unique object IDs labeled at object centers.

You must explicitly perform two tasks below, following the exact steps:

Task I. Build a Scene Graph:

Step 1. List objects grouped by their areas clearly from all provided images.

Step 2. Identify visual adjacencies between scene views based on objects appearing in multiple images from Step 1.

Step 3. Construct a Scene Graph (JSON), using areas as nodes, with an additional “adjacent” attribute, and connect nodes (areas) explicitly if their adjacencies are confirmed by shared objects. Clearly state reasons for each connection using evidence from previous steps.

Task II. Generate a Virtual Activity:

Step 1. Decide the number of characters appropriate for the scene scale, and assign unique IDs and relevant roles (family/social/professional) based on scene context.

Step 2. Activity creation (sequence of keyframes): Each keyframe is a list of all characters’ states at that time. A character’s state is defined as: (ID, pose, reference, interaction), with “ID” being the assigned character ID, “pose” being “standing”, “sitting”, or “lying”, “reference” being the object for “standing nearby” or “sitting on” or “lying on”, “interaction” being a tuple (type, interactee) where “type” is the action performed (e.g., “talk to”, “use”) and “interactee” is the character/object involved in interaction. Ensure continuous transitions between keyframe: Every change in a character’s state triggers a new keyframe. When changing states, please clearly justify your reasoning. For movements to a new “reference” object, explicitly describe intermediate steps based on the scene graph and shared object adjacencies observed in the images.

Provide explicit, structured answers following these guidelines.

for [sample](#) in [fewshot_samples](#):

User: {“area_0” : {“objects”: [“object_0-sink”, “object_1-stove”, “object_2-cooktop”]}, “area_1” : {“objects”: [“object_3-bookshelf”, “object_5-bed”]}, “area_2” : {“objects”: [“object_4-tea_table”, “object_6-chair”, “object_7-chair”, “object_8-table”, “object_9-standing_lamp”, “object_10-tv”, “object_11-sofa”]}}



Assistant: Task I.1: View 0: area.0: [object_0-sink, object_1-stove, object_2-cooktop], area.2: [object_4-tea_table, object_10-tv]; View 1: area.2: [object_4-tea_table, object_6-chair, object_7-chair, object_8-table, object_9-standing_lamp, object_11-sofa]; View 2: area.1: [object_3-bookshelf, object_5-bed]; View 3: area.1: [object_3-bookshelf], area.2: [object_6-chair, object_7-chair, object_8-table].

I.2: View 0 and View 1 are visually adjacent because they both include object_4; View 2 and View 3 are visually adjacent because they both include object_3; View 1 and View 3 are visually adjacent because they both include object_6, object_7 and object_8.

I.3: View 1 contains area.0: [object_0, object_1, object_2], and area.2: [object_4, object_10], thus area_0 and area_2 are adjacent; View 4 contains area.1: [object_3, object_5] and area.2: [object_6, object_7, object_8], thus area_1 and area_2 are adjacent. Overall, the scene graph can be represented as:

{“area_0” : {“objects”: [“object_0-sink”, “object_1-stove”, “object_2-cooktop”], “adjacent”: [“area_2”]}, “area_1” : {“objects”: [“object_3-bookshelf”, “object_5-bed”], “adjacent”: [“area_2”]}, “area_2” : {“objects”: [“object_4-tea_table”, “object_6-chair”, “object_7-chair”, “object_8-table”, “object_9-standing_lamp”, “object_10-tv”, “object_11-sofa”], “adjacent”: [“area_0”, “area_1”]}}

Task II.1: Based on the scale of the scene that looks like a studio apartment for one to two people to live in, I’ll create two virtual characters: character_0: Husband; character_1: Wife.

II.2: keyframe 0: (character_0, sitting, object_11-sofa, (read, newspaper)). Thoughts: character_0 starts with reading newspaper. (character_1, lying, object_5-bed, (sleep, none)). Thoughts: character_1 starts with sleeping on the bed.

keyframe 1: (character_0’s state does not change) (character_1, standing, object_0-sink, (get water, object_0-sink)). Thoughts: character_1 wants to get some water, and the possible destination is object_0-sink. It needs moves area.1:object_5-bed → area.0:object_0-sink, according to the scene graph.

keyframe 2: (character_0, sitting, object_6-chair, (talk to, character_1)). Thoughts: character_0 goes to the dining area to meet character_1. It needs moves area.2:object_11-sofa → area.2:object_6_chair, according to the scene graph. (character_1, sitting, object_7-chair, (talk to, character_0)). Thoughts: character_1 goes to the dining area to meet character_0. It needs moves area.0:object_0-sink → area.2:object_7_chair, according to the scene graph.

Other samples ...

Figure 6: An example of an MLLM prompting format used in our method. The prompt explicitly guides the model to reason through character activities, spatial transitions, and interactions within the environment.



Figure 7: Selected keyframes showcasing an “apartment” activities synthesis, arranged sequentially from left to right.

Interactions generated by MLLMs are converted into constraints, addressing both positional and rotational requirements for realistic interactions, closely mirroring the method described in [40]. Two constraint templates are defined as:

1. **Positional constraints**, which constrain the distances between a character a and another identity b (a character or and objects). The distance between a character and the positional reference can naturally be satisfied given the definition. However, if the character has physical interactions with certain objects, such distances should be further considered. For example, to use a computer, a desired threshold distance D can be necessary to guarantee the computer is accessible. The template is formulated as:

$$C_p(a, b) = \max \left(1 - e^{D - \text{distance}(a, b)}, 0 \right) \quad (1)$$

2. **Rotational constraints**, which constrain the direction of a character by measuring the angle between two directions $\vec{x}$ and $\vec{y}$. For example, if a character is looking at another, it can be desired that the facing direction of the character matches the direction towards the other. The template is defined as:

$$C_o(\vec{x}, \vec{y}) = \frac{1 - \cos(\vec{x}, \vec{y})}{2}, \quad (2)$$

We employ an LLM agent to map textual descriptions to constraints. Specifically, the agent translates textual interactions into positional and rotational constraints using the two templates. It also determines appropriate threshold distances D based on its learned common knowledge (e.g. $0.5m$ for interacting with a computer).

Optimization treats these interactions as independent activity groups $\mathcal{G}$, where a single character, or a few characters interacting with each other, form an independent group, optimizing their poses jointly. In some cases, characters performing their individual activities can also be grouped if they are assigned to the same positional reference. For example, poses of two characters sitting on the same sofa should be considered together to avoid interpenetration. The optimization within $\mathcal{G}$ uses a Boltzmann-like objective:

$$f(\mathcal{G}) = e^{-\frac{1}{t}C(\mathcal{G})}, \quad (3)$$

where $C(\mathcal{G})$ represents the constraints within $\mathcal{G}$ and t is the temperature parameter decreasing over iterations. A proposed move $\mathcal{G}'$ with the Markov chain dynamics is accepted with a probability:

$$\alpha(\mathcal{G}'|\mathcal{G}) = \min \left[\frac{f(\mathcal{G}')}{f(\mathcal{G})}, 1 \right] = \min \left[e^{\frac{1}{t}(C(\mathcal{G}) - C(\mathcal{G}'))}, 1 \right]. \quad (4)$$

This pose optimization ensures a realistic portrayal of characters in the virtual environment, aligning with their textual activities and interactions. Animation clips corresponding to the interaction verbs from the generated activity descriptions are applied to the characters. For activities spanning multiple discrete keyframes, if a character moves between different positions, a walking animation is automatically inserted. These movements follow collision-free trajectories computed using the A* algorithm.

6 RESULTS

6.1 Virtual Activity Generation

We demonstrate the effectiveness of our approach by generating virtual activities across three synthetic scenes of varying scales: a small apartment (4 characters, Figure 7), a medium restaurant (31 characters, Figure 8), and a large office (40 characters, Figure 9).

In the apartment scene in Figure 7, characters engage in various household activities that reflect typical family life dynamics. A highlighted keyframe exemplifies the details: the father cooking, the mother watching TV, the daughter at the dining table, and the son getting snacks from the refrigerator. The generated poses and actions demonstrate that our method effectively produces contextually coherent and realistic household interactions.

The restaurant scenario in Figure 8 showcases our approach’s adaptability by generating a diverse array of character behaviors appropriate for a public dining setting, from dining and interacting with wait staff to engaging in conversations. The accurate depictions of different activities tailored to the restaurant showcase the creativity in handling predefined roles and contextual interactions.

For the large-scale office scenario in Figure 9, our generated results capture a professional environment with characters performing tasks such as working on computers, participating in meetings, or engaging in discussions. The characters are posed in a manner that suggests active engagement with their tasks and surroundings, reflecting the complexity of the generated activities.

Additionally, we validate the applicability of our approach to AR scenarios by testing it on real-world 3D scans from Matterport3D dataset [7]. Figure 10 shows keyframes from two real-world scenes, highlighting lifelike activities and demonstrating our method’s potential for generating realistic human-scene interactions in practical AR applications, utilizing 3D scans to serve as AR digital twins.

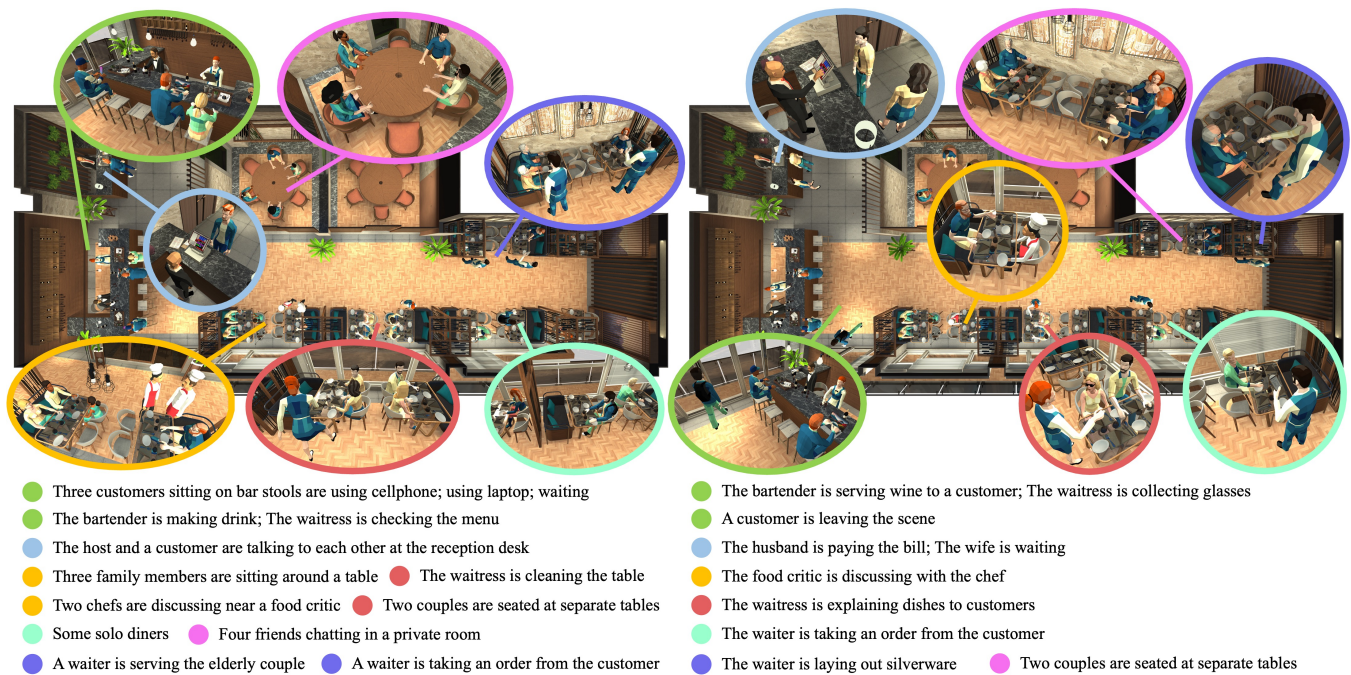


Figure 8: Selected keyframes showcasing an “restaurant” activities synthesis, arranged sequentially from left to right.

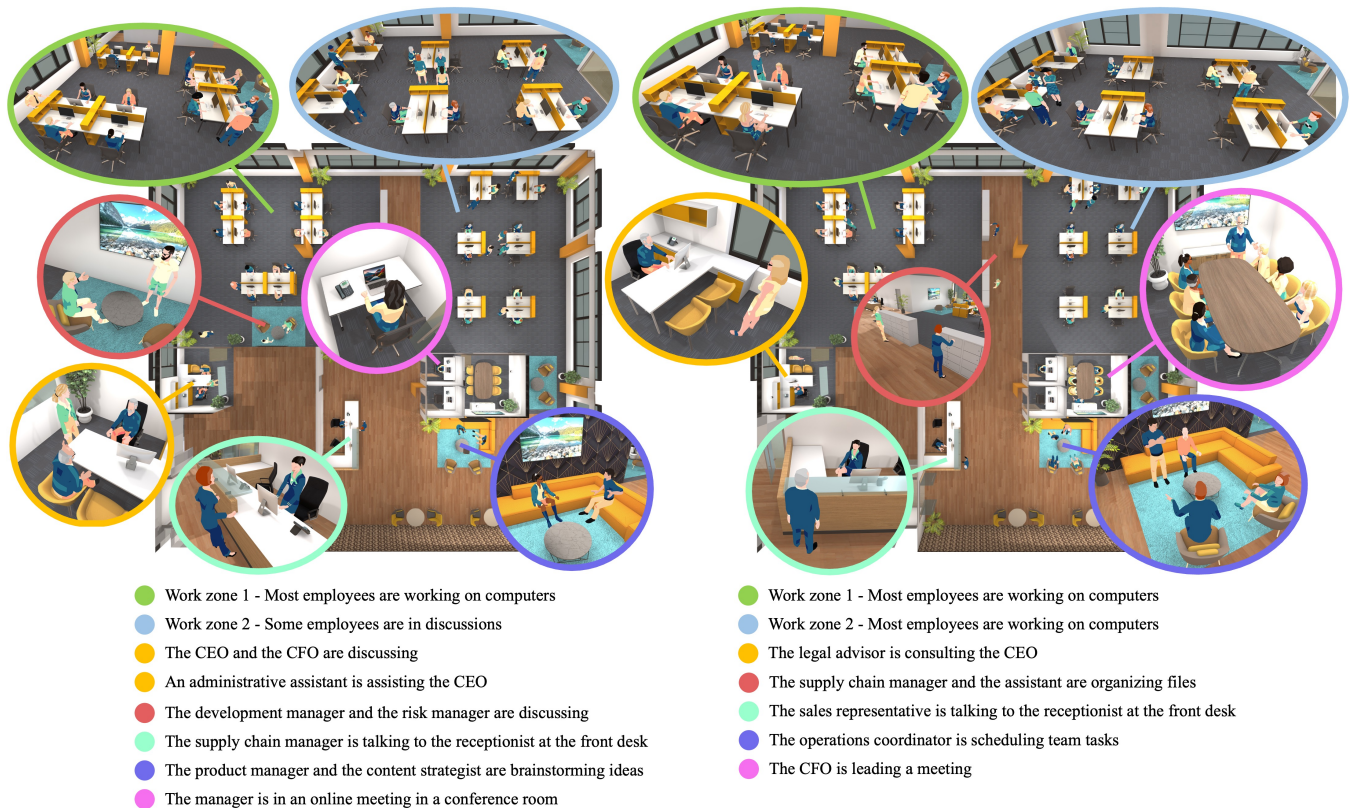


Figure 9: Selected keyframes showcasing an “office” activities synthesis, arranged sequentially from left to right.

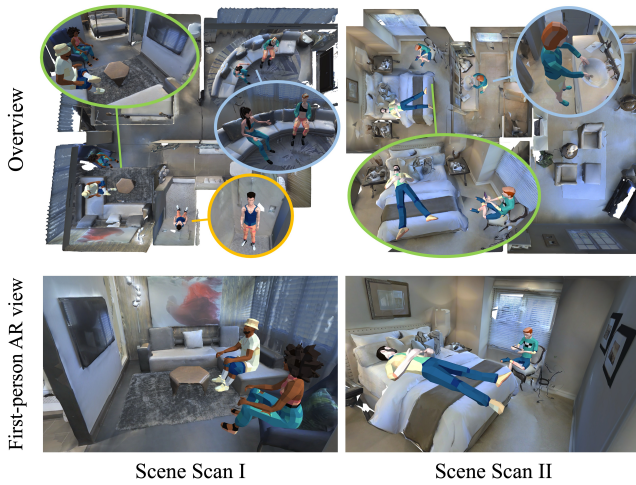


Figure 10: Keyframe snapshots in the 3D scans of two real scenes.

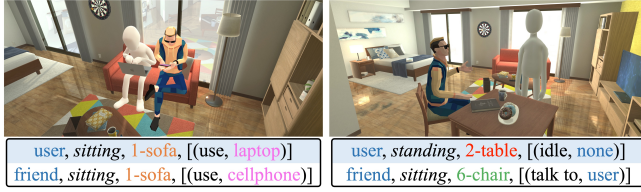


Figure 11: Interactive activity generation in response to user input. When the user avatar (white stickman) takes an action, we fix both the corresponding prompt and 3D pose, then run our method to generate context-aware behaviors for the other characters.

Across all scenes, the attention to spatial dynamics and interaction fidelity is evident. Characters are placed and engaged in a manner that authentically reflects the generated virtual activities, showcasing the adaptability of our method to diverse environments and its effectiveness in creating believable, interactive scenarios. The outcomes validate the generalizability of our approach, demonstrating its scalability and the significant potential of applying MLLMs in enriching virtual experiences through the generation of contextually appropriate and dynamic virtual activities.

Our approach naturally supports dynamic and interactive activity generation. As shown in Figure 11, once a user performs an action, we fix the corresponding prompt to reflect the user’s behavior and run our method to generate responsive behaviors from other characters. The 3D pose optimizer also incorporates the user avatar’s pose as a constraint: For instance, when both the user and a virtual character sit on the same sofa, the character’s placement avoids interpenetration with the user avatar.

The most closely related work to ours is [40], which is, however, not directly comparable. [40] learns human-scene interactions from video data, making it heavily reliant on the quality of the training data, particularly in terms of the sequence length (number of frames) and the scene scale (number of people and objects). We conducted preliminary tests using it in a restaurant setting, one of the dominant scene types in its training set. In controlled cases with fewer than 8 characters, the generated behaviors were generally reasonable. However, as the scene scale increased, the model began to fail. It exhibited unstable behaviors such as repeatedly adding new characters without meaningful interactions and subsequently removing them in the next keyframe. These issues suggest that [40] does not generalize well to larger-scale or more complex scenes without significant additional training. In contrast, our method does not rely on pre-collected interaction sequences or scene-specific training data. Our framework can interpret arbitrary scenes, rea-

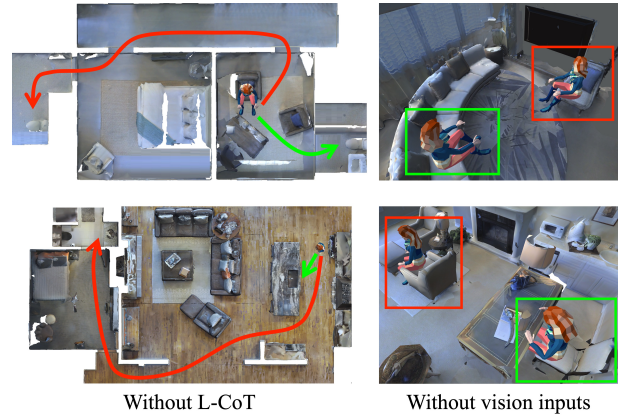


Figure 12: Ablation examples. Left: Without explicit spatial reasoning from L-CoT, the model may select suboptimal, distant destinations (red trajectories) rather than more suitable nearby alternatives (green trajectories). Right: Without vision inputs, the model may choose inappropriate or incorrectly oriented objects for interactions, resulting in unnatural or inaccessible character poses (red boxes). Incorporating visual guidance ensures contextually appropriate object selection and natural interaction poses (green boxes).

son about spatial and social contexts, and generate adaptive multi-character activities at scale. Furthermore, our approach has demonstrated generalizability across diverse scene types without retraining or domain adaptation. We scale up to 40-character scenes with stable behaviors and consistent interaction logic.

6.2 Ablation Tests

To evaluate the effectiveness of L-CoT and the vision inputs used in our work, we accordingly conducted two ablation tests.

We first assessed the activity generation without L-CoT. Given only area descriptions and multi-view scene images, the MLLM can still recognize objects and their visual appearance effectively. However, without guiding it to make explicit spatial reasoning using L-CoT, the MLLM struggled to generate logically consistent activities. As illustrated in Figure 12 (top-left), when prompted to move the character sitting on a chair to “use toilet” next, the model randomly selected one of two bathrooms due to the lack of spatial understanding. Without L-CoT, it occasionally chose a bathroom located inconveniently inside a bedroom (red trajectory). In contrast, the model reliably identified the nearest and most suitable bathroom (green trajectory) with L-CoT. Similarly, in Figure 12 (bottom-left), without L-CoT, the model inefficiently moved a character, who obtained an apple from the kitchen refrigerator, to “use sink” to wash the apple in a distant bathroom (red trajectory), instead of just using the adjacent sink in the kitchen (green trajectory).

We also examined the role of vision inputs in the system. Some recent approaches use scene graph as the only input to LLMs for human activity generation [15, 41], but rely on carefully designed scene graph representations capturing symbolic concepts such as traversability, objects, and people/agents [3, 58]. We used scene graphs generated by a standard MST algorithm in this test, providing MLLMs with general spatial layouts but lacked detailed visual information such as object shapes and poses. For instance, as illustrated in Figure 12 (top-right), when relying solely on the simplified scene graph, the model assigned the character to a chair that resulted in an unnatural and uncomfortable pose (red box) to perform the “watching tv” activity. With visual guidance from images, the MLLM avoided inappropriate chairs, resulting in more natural and realistic character poses (green box). Similarly, in Figure 12 (bottom-right), the character sat on a chair facing away from the desk and positioned farther from it, rendering the desk inaccessible for work (red box). With visual inputs, the model correctly selected

Table 1: Each cell shows the selection rate (SR_{ours}) of our method over manual designs (percentage $\pm$ standard error), along with the p -value from a Chi-square test under the null hypothesis of no preference difference. The four evaluation metrics are: reasonableness (M^r), naturalness (M^n), understandability (M^u), and overall (M^o).

Metrics		Scene 1	Scene 2	Scene 3	Scene 4	Scene 5
M^r	SR_{ours}	51 $\pm$ 5.00	48 $\pm$ 5.00	49 $\pm$ 5.00	43 $\pm$ 4.95	54 $\pm$ 4.98
	p -value	0.84	0.69	0.84	0.16	0.42
M^n	SR_{ours}	53 $\pm$ 4.99	55 $\pm$ 4.97	56 $\pm$ 4.96	48 $\pm$ 5.00	44 $\pm$ 4.96
	p -value	0.55	0.32	0.23	0.69	0.23
M^u	SR_{ours}	47 $\pm$ 4.99	49 $\pm$ 5.00	53 $\pm$ 4.99	45 $\pm$ 4.97	51 $\pm$ 5.00
	p -value	0.55	0.84	0.55	0.32	0.84
M^o	SR_{ours}	45 $\pm$ 4.97	54 $\pm$ 4.98	59 $\pm$ 4.92	52 $\pm$ 5.00	47 $\pm$ 4.99
	p -value	0.32	0.42	0.07	0.69	0.55

a chair closer to the desk and oriented appropriately, enabling natural and functional interactions (green box). While detailed scene graphs designed with comprehensive spatial and object-specific attributes might improve results as well, creating such representations for diverse scene types is challenging. Therefore, directly leveraging visual inputs offers an efficient and effective solution.

6.3 Perceptual Study

We conducted a perceptual study, approved by the Institutional Review Boards, to evaluate the quality of virtual activities generated by our method compared to manual professional designs. We invited 5 technical artists, each with over 5 years of professional Unity3D experience, to create virtual activities in 5 different small virtual scenes (scene 1, 2: apartments; scene 3: barbershop; scene 4: bar; scene 5: office). They were informed to include a desired number of virtual characters, and assign roles to all. They were asked to create activities of at least 8 keyframes: in each keyframe, they described all characters’ behaviors and states, and adjusted their poses and selected proper animation clips to match the context they wanted. The manual creation mainly included clicking and dragging operations using our Unity3D designer tool.

For each manual design, we summarized all characters’ activities using texts, and used these summaries together with characters’ roles and the number of keyframes as additional constraints in the prompt, ensuring contextual similarities for fair comparison.

We recruited 100 participants in Amazon Mechanical Turk, 77 males and 23 females with ages ranging from 25 to 52 years, for this study. We animated both manually created and automatically generated activities, and rendered video recordings for all. For each participant, we showed them all 5 scenes, each consisting of a result under *designer* condition and another under *ours* condition. The study followed a two-alternative forced choice (2AFC) format: for each paired video recordings (randomly shuffled), they watched and chose a better one regarding each of the four evaluation metrics:

- Reasonableness (M^r): How reasonable and contextually appropriate are the characters’ actions and poses, given the scene and the described activity.
- Naturalness (M^n): How natural, and lifelike are the characters’ movements and body poses.
- Understandability (M^u): How easy is it to understand what the characters are doing by watching the video.
- Overall (M^o): General preference considering all aspects.

The results in Table 1 shows the numerical results. The selection rates for *ours* (SR_{ours}) were generally close to 50%, indicating similar preferences to *designer* results. A Chi-square test confirmed that the differences were not statistically significant ($p > 0.05$), supporting the conclusion that our method produces results of comparable perceptual quality to human-designed content.

For efficiency, professional designers spent 13–32 minutes per scene. Our method completed all scenes in 3–5 minutes, including image processing, MLLM generation, and 3D pose optimization.

7 LIMITATIONS AND FUTURE WORK

While our method demonstrates promising results, several limitations remain. MLLMs could still struggle when processing scenes or activities with extended contexts, such as large environments or activities of long keyframe sequences. For instance, in our office scene, although the generated activities mostly matched the environment, certain social conventions, such as employees having assigned desks, were overlooked. We found that generating fewer keyframes per query, rather than requesting multiple keyframes at once, improves context consistency and robustness. It is worth further exploring hierarchical reasoning methods, and incorporating explicit knowledge about social or professional norms, which might allow MLLMs to maintain context coherence over longer activity sequences and across larger spaces.

Additionally, our current method relies on predefined animation clips tied directly to interaction verbs described by the MLLMs, which may lead to repetitive animations across different characters performing similar actions. Although it can be easily extended by adding more animations and interaction verbs to support richer, more diverse behaviors, another promising improvement would be integrating generative motions [48, 22, 54, 25] to allow smoother, more diverse, and adaptive animations that reflect detailed variations between characters and interactions.

Our method separates the process into two stages: activity reasoning with an MLLM and 3D pose optimization. This effective modular design opens up an interesting direction for future research—enabling MLLMs to directly generate human poses. Achieving this goal would require aligning the language modality with parametric human models such as SMPL [46]. Recent work has begun exploring the use of MLLMs to generate SMPL parameters directly [13, 44, 45, 12], reflecting growing interests in this area. However, current approaches remain limited in their ability to model complex multi-character interactions and scene-aware behaviors. Extending MLLMs to reason about and generate physically and socially plausible multi-character poses remains a challenging yet promising direction for future work.

8 CONCLUSION

In this work, we propose a novel framework that enriches virtual environments with realistically virtual activities, leveraging the advanced capabilities of multimodal large language models (MLLMs). By integrating vision-language modalities with Layout Chain-of-Thought (L-CoT) prompting, our approach enables MLLMs to interpret spatial and semantic cues from virtual scenes and generate context-aware, detailed descriptions of adaptive activities. These descriptions, rich in contextual understanding, lay the groundwork for the precise characters’ placements within the virtual space. Our approach ensures that characters are not just accurately placed but are also seemingly engaged in believable interactions, thanks to a robust optimization framework that aligns with the structured format of activity descriptions.

The adaptation of MLLMs for virtual activity generation underscores their potential in crafting complex, dynamic, and immersive scenarios. The implications of our findings suggest promising avenues for enhancing user experiences for applications in VR/AR, gaming and AI-driven storytelling. The adaptability and scalability of our method suggest its applicability across a broad spectrum of virtual environment creations such as future metaverse experiences.

ACKNOWLEDGMENTS

This project is supported by National Science Foundation grants with award numbers 2430673, 2418236, and 1942531.

REFERENCES

- [1] S. Agrawal and M. van de Panne. Task-based locomotion. *ACM Transactions on Graphics*, 35(4):1–11, 2016. 2
- [2] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022. 1
- [3] I. Armeni, Z.-Y. He, J. Gwak, A. R. Zamir, M. Fischer, J. Malik, and S. Savarese. 3d scene graph: A structure for unified semantics, 3d space, and camera. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5664–5673, 2019. 8
- [4] N. Athanasiou, M. Petrovich, M. J. Black, and G. Varol. Teach: Temporal action composition for 3d humans. *arXiv preprint arXiv:2209.04066*, 2022. 2
- [5] Y. Bai, K. Siu, and C. K. Liu. Synthesis of concurrent object manipulation tasks. *ACM Transactions on Graphics*, 31(6):1–9, 2012. 2
- [6] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik. Long-term human motion prediction with scene context. In *European Conference on Computer Vision*, pp. 387–404. Springer, 2020. 1, 2
- [7] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision*, 2017. 6
- [8] P. Charalambous, J. Pettre, V. Vassiliades, Y. Chrysanthou, and N. Pelechano. Greil-crowds: Crowd simulation with deep reinforcement learning and examples. *ACM Transactions on Graphics (TOG)*, 42(4):1–15, 2023. 2
- [9] J. Chen, X. Wu, T. Lan, and B. Li. Llmer: Crafting interactive extended reality worlds with json data generated by large language models. *arXiv preprint arXiv:2502.02441*, 2025. 2
- [10] S. Chen, X. Chen, C. Zhang, M. Li, G. Yu, H. Fei, H. Zhu, J. Fan, and T. Chen. Ll3da: Visual interactive instruction tuning for omni-3d understanding reasoning and planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26428–26438, 2024. 2
- [11] Y. Chen, S. Yang, H. Huang, T. Wang, R. Xu, R. Lyu, D. Lin, and J. Pang. Grounded 3d-llm with referent tokens. *arXiv preprint arXiv:2405.10370*, 2024. 2
- [12] G. Delmas, P. Weinzaepfel, F. Moreno-Noguer, and G. Rogez. Poseembroider: Towards a 3d, visual, semantic-aware human pose representation. In *European Conference on Computer Vision*, pp. 55–73. Springer, 2024. 9
- [13] Y. Feng, J. Lin, S. K. Dwivedi, Y. Sun, P. Patel, and M. J. Black. Chatpose: Chatting about 3d human pose. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2093–2103, 2024. 9
- [14] A. Ghosh, N. Cheema, C. Oguz, C. Theobalt, and P. Slusallek. Synthesis of compositional animations from textual descriptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1396–1406, 2024. 2
- [15] N. Gorlo, L. Schmid, and L. Carlone. Long-term human trajectory prediction using 3d dynamic scene graphs. *IEEE Robotics and Automation Letters*, 2024. 2, 8
- [16] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5021–5028. IEEE, 2024. 2
- [17] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned generation of 3d human motions. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2021–2029, 2020. 2
- [18] M. Hassan, D. Ceylan, R. Villegas, J. Saito, J. Yang, Y. Zhou, and M. J. Black. Stochastic scene-aware motion prediction. In *ICCV*, pp. 11374–11384, 2021. 1, 2
- [19] M. Hassan, P. Ghosh, J. Tesch, D. Tzionas, and M. J. Black. Populating 3d scenes by learning human-scene interaction. In *CVPR*, pp. 14708–14718, 2021. 1, 2, 4
- [20] M. Hassan, Y. Guo, T. Wang, M. Black, S. Fidler, and X. B. Peng. Synthesizing physical character-scene interactions. *arXiv preprint arXiv:2302.00883*, 2023. 1, 2
- [21] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. 1970. 4
- [22] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 9
- [23] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [24] H. Huang, Y. Chen, Z. Wang, R. Huang, R. Xu, T. Wang, L. Liu, X. Cheng, Y. Zhao, J. Pang, et al. Chat-scene: Bridging 3d scene and large language models with object identifiers. *arXiv preprint arXiv:2312.08168*, 2023. 2
- [25] H. Huang, F. De La Torre, C. M. Fang, A. Banburski-Fahey, J. Amores, and J. Lanier. Real-time animation generation and control on rigged models via large language models. *arXiv preprint arXiv:2310.17838*, 2023. 9
- [26] J. Huang, S. Yong, X. Ma, X. Linghu, P. Li, Y. Wang, Q. Li, S.-C. Zhu, B. Jia, and S. Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023. 2
- [27] S. Huang, Z. Wang, P. Li, B. Jia, T. Liu, Y. Zhu, W. Liang, and S.-C. Zhu. Diffusion-based generation, optimization, and planning in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16750–16761, 2023. 1, 2
- [28] N. Jiang, Z. He, Z. Wang, H. Li, Y. Chen, S. Huang, and Y. Zhu. Autonomous character-scene interaction synthesis from text instruction. In *SIGGRAPH Asia 2024 Conference Papers*, pp. 1–11, 2024. 2
- [29] M. Kari, R. Schütte, and R. Sodhi. Scene responsiveness for visuotactile illusions in mixed reality. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–15, 2023. 2
- [30] J. Kim, Y. Seol, and T. Kwon. Interactive multi-character motion re-targeting. *Computer Animation and Virtual Worlds*, 32(3-4):e2015, 2021. 2
- [31] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023. 3
- [32] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. 4
- [33] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. 3
- [34] J. Lee, F. Aleotti, D. Mazala, G. Garcia-Hernando, S. Vicente, O. J. Johnston, I. Kraus-Liang, J. Powierza, D. Shin, J. E. Froehlich, et al. Imaginatear: Ai-assisted in-situ authoring in augmented reality. *arXiv preprint arXiv:2504.21360*, 2025. 2
- [35] J. Lee, J. Won, and J. Lee. Crowd simulation by deep reinforcement learning. In *Proceedings of the 11th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pp. 1–7, 2018. 2
- [36] S. Lee, P. S. Chang, and J. Lee. Deep compliant control. In *ACM SIGGRAPH 2022 Conference Proceedings*, 2022. 2
- [37] B. Li, E. S. Ho, H. P. Shum, and H. Wang. Two-person interaction augmentation with skeleton priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1900–1910, 2024. 2
- [38] C. Li, J. Chibane, Y. He, N. Pearl, A. Geiger, and G. Pons-Moll. Unimotion: Unifying 3d human motion synthesis and understanding. *arXiv preprint arXiv:2409.15904*, 2024. 2
- [39] C. Li, W. Li, H. Huang, and L.-F. Yu. Interactive augmented reality storytelling guided by scene semantics. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 2, 4
- [40] C. Li and L.-F. Yu. Generating activity snippets by learning human-scene interactions. *ACM Transactions on Graphics (TOG)*, 42(4):1–15, 2023. 2, 4, 6, 8
- [41] H. Li, Z. Wang, W. Liang, and Y. Wang. X’s day: Personality-driven virtual human behavior generation. *IEEE transactions on visualization and computer graphics*, 2025. 2, 8
- [42] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and gen-

- eration. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022. 1
- [43] J. Li, J. Li, J. Chen, Y. Li, S. Wang, H. Zhou, M. Ye, and Y. Su. Evolving agents: Interactive simulation of dynamic and diverse human personalities. *arXiv preprint arXiv:2404.02718*, 2024. 2
- [44] Y. Li, R. Hou, H. Chang, S. Shan, and X. Chen. Unipose: A unified multimodal framework for human pose comprehension, generation and editing. *arXiv preprint arXiv:2411.16781*, 2024. 9
- [45] J. Lin, Y. Feng, W. Liu, and M. J. Black. Chathuman: Language-driven 3d human understanding with retrieval-augmented tool reasoning. *arXiv preprint arXiv:2405.04533*, 2024. 9
- [46] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black. Smpl: A skinned multi-person linear model. In *ACM Transactions on Graphics (TOG), Volume 34, Issue 6*, pp. 1–16. 2015. 9
- [47] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953. 4
- [48] J. Min, Y.-L. Chen, and J. Chai. Interactive generation of human animation with deformable motion models. *ACM Transactions on Graphics (TOG)*, 29(1):1–12, 2009. 9
- [49] S. Paris, J. Pettré, and S. Donikian. Pedestrian reactive navigation for crowd simulation: a predictive approach. In *Computer Graphics Forum*, vol. 26, pp. 665–674. Wiley Online Library, 2007. 2
- [50] J. S. Park, J. O’Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–22, 2023. 2
- [51] M. Petrovich, M. J. Black, and G. Varol. Action-conditioned 3d human motion synthesis with transformer vae. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10985–10995, 2021. 2
- [52] S. Pirk, V. Krs, K. Hu, S. D. Rajasekaran, H. Kang, Y. Yoshiyasu, B. Benes, and L. J. Guibas. Understanding and exploiting object interaction landscapes. *ACM Transactions on Graphics (TOG)*, 36(3):1–14, 2017. 2
- [53] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, 1957. 3
- [54] J. Qin, Y. Zheng, and K. Zhou. Motion in-betweening via two-stage transformers. *ACM Trans. Graph.*, 41(6):184–1, 2022. 9
- [55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021. 1
- [56] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. 1
- [57] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 3
- [58] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone. Kimera: From slam to spatial perception with 3d dynamic scene graphs. *The International Journal of Robotics Research*, 40(12-14):1510–1546, 2021. 8
- [59] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner. Scenegrok: Inferring action maps in 3d environments. *ACM Transactions on Graphics*, 33(6):1–10, 2014. 2
- [60] M. Savva, A. X. Chang, P. Hanrahan, M. Fisher, and M. Nießner. Pigraphs: learning interaction snapshots from observations. *ACM Transactions on Graphics*, 35(4):1–12, 2016. 2
- [61] S. Starke, H. Zhang, T. Komura, and J. Saito. Neural state machine for character-scene interactions. *ACM Transactions on Graphics*, 38(6):209–1, 2019. 1, 2
- [62] T. Tahara, T. Seno, G. Narita, and T. Ishikawa. Retargetable ar: Context-aware augmented reality in indoor scenes based on 3d scene graph. In *2020 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pp. 249–255. IEEE, 2020. 2
- [63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 1
- [64] J. Wang, Y. Rong, J. Liu, S. Yan, D. Lin, and B. Dai. Towards diverse and natural scene-aware 3d human motion synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20460–20469, 2022. 1, 2
- [65] J. Wang, H. Xu, J. Xu, S. Liu, and X. Wang. Synthesizing long-term 3d human motion and interaction in 3d scenes. In *CVPR*, pp. 9401–9411, 2021. 1, 2
- [66] J. Wang, S. Yan, B. Dai, and D. Lin. Scene-aware generative network for human motion synthesis. In *CVPR*, pp. 12206–12215, 2021. 1, 2
- [67] Z. Wang, Y. Chen, B. Jia, P. Li, J. Zhang, J. Zhang, T. Liu, Y. Zhu, W. Liang, and S. Huang. Move as you say interact as you can: Language-guided human motion generation with scene affordance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 433–444, 2024. 2
- [68] Z. Wang, Y. Chen, T. Liu, Y. Zhu, W. Liang, and S. Huang. Humanise: Language-conditioned human motion generation in 3d scenes. *arXiv preprint arXiv:2210.09729*, 2022. 2
- [69] Z. Wang, J. Wang, Y. Li, D. Lin, and B. Dai. Intercontrol: Zero-shot human interaction generation by controlling every joint. *Advances in Neural Information Processing Systems*, 37:105397–105424, 2024. 2
- [70] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022. 3
- [71] Z. Xiao, T. Wang, J. Wang, J. Cao, W. Zhang, B. Dai, D. Lin, and J. Pang. Unified human-scene interaction via prompted chain-of-contacts. *arXiv preprint arXiv:2309.07918*, 2023. 2
- [72] S. Xu, Z. Li, Y.-X. Wang, and L.-Y. Gui. Interdiff: Generating 3d human-object interactions with physics-informed diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14928–14940, 2023. 1, 2
- [73] J. Yang, H. Zhang, F. Li, X. Zou, C. Li, and J. Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023. 2, 3
- [74] H. Yi, J. Thies, M. J. Black, X. B. Peng, and D. Rempe. Generating human interaction motions in scenes with text control. In *European Conference on Computer Vision*, pp. 246–263. Springer, 2024. 2
- [75] S. Zhang, Y. Zhang, Q. Ma, M. J. Black, and S. Tang. Place: Proximity learning of articulation and contact in 3d environments. In *2020 International Conference on 3D Vision (3DV)*, pp. 642–651. IEEE, 2020. 1, 2
- [76] Y. Zhang, D. Gopinath, Y. Ye, J. Hodgins, G. Turk, and J. Won. Simulation and retargeting of complex multi-character interactions. *arXiv preprint arXiv:2305.20041*, 2023. 2
- [77] Y. Zhang, M. Hassan, H. Neumann, M. J. Black, and S. Tang. Generating 3d people in scenes without people. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6194–6204, 2020. 1, 2
- [78] K. Zhao, S. Wang, Y. Zhang, T. Beeler, and S. Tang. Compositional human-scene interaction synthesis with semantic control. In *European Conference on Computer Vision*, pp. 311–327. Springer, 2022. 1, 2
- [79] K. Zhao, Y. Zhang, S. Wang, T. Beeler, and S. Tang. Synthesizing diverse human motions in 3d indoor scenes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 14738–14749, 2023. 1, 2
- [80] M. Zhao, W. Cai, and S. J. Turner. Clust: simulating realistic crowd behaviour by mining pattern from crowd videos. In *Computer graphics forum*, vol. 37, pp. 184–201. Wiley Online Library, 2018. 2
- [81] L. Zhong, Y. Xie, V. Jampani, D. Sun, and H. Jiang. Smoodi: Stylized motion diffusion model. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2024. 2