Interaction-aware Shared Scene Synthesis for VR Telepresence

Zhangyao Tan, Qixiang Ma, Runze Fan, Sio Kei Im, Lili Wang

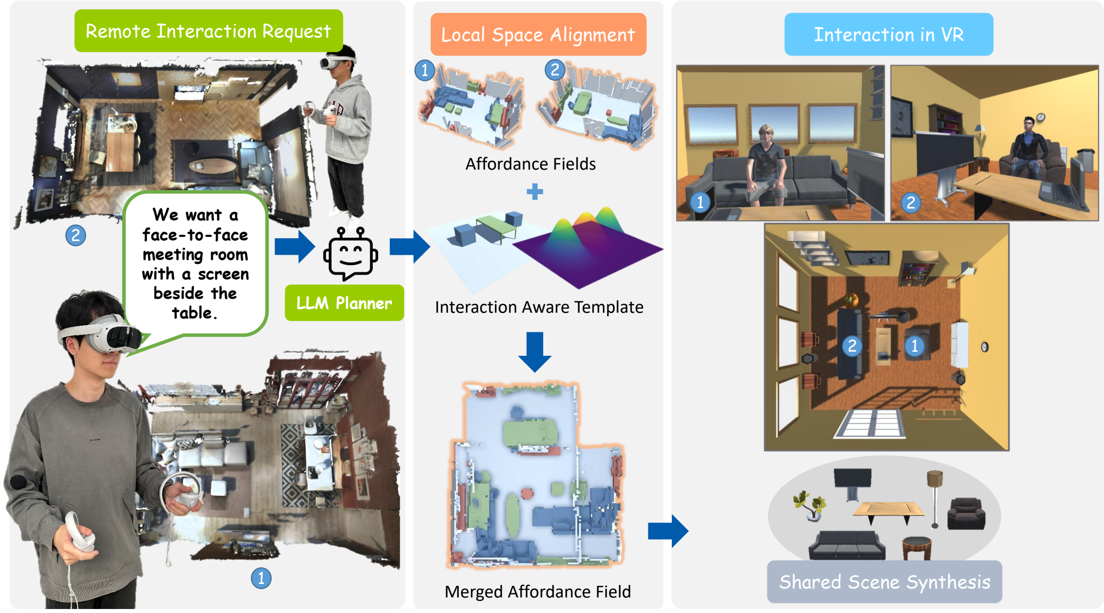


Fig. 1: Taking the scanned and semantically segmented physical space of two remote users and the text of their interaction requirements as input (column 1), our method searches for the local aligned area suitable for interaction, thus obtaining a merged shared space (column 2). Then we synthesize the virtual scene meeting the requirements based on the shared space (column 3).

Abstract— Virtual reality telepresence requires immersive shared virtual environments for real-time remote collaboration across different physical scenes. It supports a wide range of applications in teleconferencing, education, and interactive simulations. However, challenges persist in identifying optimal shared virtual spaces that accommodate diverse user interaction requirements while adhering to local physical constraints during scene synthesis. In this paper, we propose an interaction-aware shared virtual scene synthesis method, which uses the large language model (LLM) to produce collaborative virtual scenes based on interaction demands from remote users in different local spaces. First, we introduce the concept of Interaction Aware Template (IAT) and its construction method using an LLM planner. Then, we propose an IAT-based affordance field alignment method for merging the local spaces of the remote users, maximally ensuring that the aligned space could support the user's desired interaction. Finally, we propose an LLM-based shared scene synthesis method according to the merged affordance field. Experiment results show that, compared to existing text-based scene synthesis and mutual space matching methods, our method achieves better Affordance Consistency, 3D Intersection over Union, and Layout Suitability on both scanned and synthesized datasets. The results of the user study demonstrate that the user's subjective perception of interaction fitness and sense of safety were significantly improved.

Index Terms—Virtual reality, Shared scene, Scene synthesis, Collaborative interaction.

1 Introduction

Virtual reality telepresence aims to create a diverse interactive environment for remote users in different physical spaces, allowing users to immerse themselves in remote collaborative interactions. It has numerous applications in remote meetings, remote collaborations, and remote education. Previously, virtual scenes for remote interaction were manually constructed, requiring manpower and expense. Recently, generative neural network models have significantly improved the efficiency and quality of Virtual Reality (VR) content creation, such as 3D object generation [1], scene synthesis [2–4], and virtual avatar [5, 6]. Since remote user interactions are constrained by their physical environment, the generation of shared virtual scenes must take into account the physical constraints of different remote users.

The construction of shared virtual scene requires optimization in multiple aspects such as the user's environment [7] and behavior [8]. Early research used procedural methods to generate virtual scenes that align with physical environments in the real world [9, 10], but they are limited in terms of scene diversity and interactivity. With the advancement of generative models, studies have integrated Transformer [2], Diffusion [3], and LLM-based [4, 11] automated pipelines guided by text or images to synthesize scenes, and Jiang et al. [7] also consider the physical environments of users. However, these methods

- *Lili Wang is the corresponding author.*
- *Lili Wang, Zhangyao Tan, Qixiang Ma, and Runze Fan are with the State Key Laboratory of Virtual Reality Technology and Systems, School of Computer Science and Engineering, Beihang University, Beijing, China. E-mail: wanglily@buaa.edu.cn, tanzyao@buaa.edu.cn, sycamore_ma@buaa.edu.cn, by2106131@buaa.edu.cn.*
- *Sio Kei Im is with Macao Polytechnic University. E-mail: marcusim@mpu.edu.mo.*

Manuscript received xx xxx. 201x; accepted xx xxx. 201x. Date of Publication xx xxx. 201x; date of current version xx xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: xx.xxxx/TVCG.201x.xxxxxxx

are mainly designed for generalized scene generation or single-user scenarios. Some works optimize shared areas between users through space matching [12, 13], spatial allocation [14], and object clustering [15], enabling natural movement and interaction while maintaining a strong sense of co-presence. Furthermore, other works mitigate the impact of spatial differences between remote environments through avatar positioning [16] and motion control [17, 18]. These methods are primarily applicable to interaction in real spaces and do not take fully advantage of the diverse experiences that virtual reality can provide. Therefore, further exploration is needed to integrate generative techniques for constructing remote shared spaces that maintain consistency between virtual and physical environments while being interaction-aware.

For a well-designed shared virtual scene, we believe that it needs to meet four conditions: (1) the category, style and layout of shared scene should meet the user's collaborative interaction needs; (2) the shared space is supposed to be aligned with real objects that support the interaction, otherwise to place suitable virtual objects to meet the diverse needs; (3) in the local aligned area, real physical obstacles need to be covered by virtual objects to avoid collision while maximizing the movable area of each user; (4) the virtual objects need to avoid providing the wrong interaction intent to the user to ensure security, especially for sittable objects. This brings challenges on how to adaptively find the most suitable local alignment of shared area in the physical scene for telepresence, and how to strike a good balance between the layout that meets users' interaction requirements and adherence to physical constraints for the generated virtual scene.

To address these challenges, we propose a novel interaction-aware shared virtual scene synthesis method. Based on interaction needs described in natural language and physical environments from users, our method dynamically aligns an interaction-aware shared space and generates a shared scene that meets the interaction needs while maintaining physical consistency. Our method consists of three key components. First, we introduce the definition of the Interaction Aware Template (IAT) and its construction method using an LLM planner to understand users' physical space layouts and their interaction needs from textual descriptions. Second, we propose an IAT-based affordance field alignment method for merging the local spaces of remote users. We convert the template elements of IAT into Gaussian probability distributions, serving as a bridge to find suitable local shared areas within each user's space and to compute the alignment between the user scenes accordingly. Third, we propose an LLM-based shared scene synthesis method according to the merged affordance field. Unlike existing methods [2, 4, 7] that incrementally generate scenes, we jointly optimize the layout constraints of the virtual scene and the physical constraints of the local space. We retrieve objects from the asset database to generate an immersive virtual scene. Fig. 1 illustrates a scene synthesis example of our method. We validated our method on both scanned and synthesized datasets [19, 20] and compared it with the state-of-the-art methods, including text-based scene synthesis [4] and mutual space matching methods [13, 14]. Objective quantitative results show that for sitting interactions, our method outperforms the SOTA methods in terms of consistency, including Free Space Ratio(FSR), Affordance Consistency, 3D IoU, and Layout Suitability. For walking interactions, our method also outperforms the SOTA method, except for FSR, where it is slightly inferior to the area-centric method. The results of user study shows that the perceived quality by users has significantly improved. While effective, our method is constrained by users' physical environments and predefined object assets, which may limit scalability and visual fidelity. It also generates static shared scenes without real-time adaptation. Moreover, the user study does not yet assess task-level performance such as completion time or success rate.

In summary, the main contributions of this paper are as follows: (1) We propose an interaction-aware shared virtual scene synthesis pipeline, which uses large language models to generate collaborative virtual scenes that conform to collaborative interaction demands from remote users in different local physical spaces. (2) We introduce the definition of the Interaction Aware Template and its construction methodology using an LLM planner to understand users' physical space layouts and interaction demands. (3) We propose an IAT-based affordance field alignment method for generating the merged affordance fields of the remote users, maximally ensuring that the virtual scene meets users' interaction demands. We also propose an LLM-based shared scene synthesis method according to the merged affordance field.

2 Related Work

There is a lot of existing work dedicated to scene synthesis. In this section, we review three types of prior works: scene synthesis for VR, LLM for VR scenes, and shared space for VR telepresence.

2.1 Scene Synthesis for VR

In recent years, some approaches leverage pretrained 2D image models combined with various scene representations, such as mesh [21], NeRF [22], and 3D Gaussians [23]. While these methods excel at generating or reconstructing realistic views, they lack interactivity in VR applications. Other approaches train neural networks to learn the latent distribution of scenes and generate 3D layouts such as autoregressive transformers [2, 24] and diffusion models [3, 25]. While these methods primarily focus on filling an empty room, they overlook the constraints from users' real-world environments, and are therefore difficult to be used directly for VR interaction. Regarding scenarios for VR, some studies focus on detecting walkable areas and obstacles in real-time based on users' physical environments, ensuring consistency between physical and virtual spaces. For example, Oasis [26] enables real-walking in the generated virtual environment by capturing indoor scenes in 3D and mapping walkable areas. DreamWalker [27] and VRoamer [9] extract walkable areas and detect physical obstacles in real-time using scanning devices. Other approaches generate stylized virtual environments based on reconstructed real-world spaces. RoomDreamer [28] and DreamSpace [29] use diffusion models to generate stylized textures for scanned indoor environments. RealitySkins [10] extracts constraints from local geometric data and employs optimization algorithms to arrange and place geometrically and semantically consistent virtual objects. These methods are used to provide individual users with virtual experiences that conform to the real environment and cannot solve the problem of scene generation under the demand of multi-user remote collaboration. Different from the existing methods, our method find the local alignment suitable for both remote users and synthesize diverse scenes that meet the user's interaction needs.

2.2 LLM for VR Scenes

Recent studies [30, 31] have demonstrated that by leveraging deep understanding of user intent and advanced reasoning abilities, LLMs can efficiently handle tasks such as scene synthesis [4, 32], scoring [33], code generation [34–36], and immersive storytelling [37], introducing new paradigms for VR applications. For instance, [33] employs vision-language models (VLMs) to perform real-time saliency scoring in AR environments, optimizing UI layout rearrangement via constraint-based algorithms, achieving superior performance compared to traditional adaptive layout methods. Additionally, methods such as DreamCodeVR [34], LLMR [35], and VRCopilot [38] integrate LLMs into real-time systems, leveraging their multimodal understanding and code generation capabilities to effectively fulfill user requests and automate complex tasks. For scene synthesis, LayoutGPT [11] guides LLM to generate structured layouts, but directly generating numerical coordinates can lead to object overlap. Holodeck [4] employs constraint solving to optimize object layout, facilitating diverse embodied AI environments. Text2VRScene [39] proposes an LLM-based framework for VR scene generation. Jiang et al. use a scene agent to replace real-world objects with affordance-consistent virtual objects, ensuring scene interactivity. In these frameworks, LLMs play an assisting role and require highly customized designs for different tasks, thus they cannot be directly reused for VR telepresence. Our method also designs an LLM planner to understand user interaction requirements and their local spatial layouts, and combine optimization algorithms to generate reasonable shared virtual scenes.

2.3 Shared Space for VR Telepresence

One research direction of VR telepresence focuses on reducing spatial inconsistencies between users, enhancing mutual awareness in remote interactions, and ensuring user safety within their physical environments. Research in this area can be broadly classified into avatar control and spatial mapping approaches. In terms of avatar control for telepresence, early studies [40, 41] used capture devices to transmit user's image or 3D model to remote spaces. However, these approaches struggled to support complex movement and interactions. Leonard et al. [16] conducted user studies to determine optimal avatar placements across different spaces and trained a neural network incorporating interpersonal, visual, and spatial factors to predict real-time avatar positions. Wang et al. [17] investigated appropriate motion patterns for avatars and introduced a control module to eliminate latency, ensuring smooth avatar movement. Furthermore, Li et al. [18] proposed a reinforcement learning framework to guide avatars toward suitable positions. Wong et al. [42] developed a spatial heterogeneity framework for distributed mixed reality collaboration. Additionally, methods such as visual guidance [43] and motion retargeting [44, 45] have been explored to further optimize user interaction in telepresence environments. These methods explore co-occurrence awareness in the process of remote interaction, but do not include discussions on shared scenarios.

On the other hand, constructing shared spaces requires identifying regions in different physical spaces that are suitable for interaction. Nicolas et al. [12] developed an automated spatial alignment method for remote video-conferencing, while Mohammad et al. [13] extended spatial matching to multi-user scenarios. They defined user spaces in terms of standable and sittable areas and optimized shared spaces by dynamically rearranging furniture. Kim et al. [14] incorporated affordance-based subspace allocation to optimize spatial allocation in host-client collaborative environments, increasing both shared area and user instantiation success rates. To manage object relationships within shared spaces, they proposed Object Cluster Registration [15], leveraging geometric spatial affordance graph to enhance object co-occurrence. SpaceBlender [46] uses generative models to merge users' local spaces into a unified virtual environment. Other related studies [47, 48] explored techniques such as whiteboard and table-based partial alignment [49], position mapping [50–52], and redirected walking [53–55] to enhance remote user interactions. However, the results obtained by these methods are not the optimal solutions for users under different interaction requirements. Therefore, a more usable system needs to introduce user interaction information to guide the construction of the shared space. Our method focuses on adaptive spatial alignment based on user interaction needs, integrating LLM-based reasoning to dynamically align users' local spaces.

3 Method

We aim to generate a virtual scene for users in the remote space that satisfies their collaborative interaction needs. Fig. 3 shows the pipeline of our method, which contains three steps. First, the input of our method consists of the layout information of the user's physical space and the text describing the interaction between the user, then we construct Interaction Aware Template (IAT) through the input (Section 3.1). Secondly, we transform the user's physical space into affordable fields, then we, then we propose an IAT-based affordance field alignment algorithm to calculate the local alignment between the physical scenes of remote users and generate a merged affordance field that meets the interaction requirements (Section 3.2). Finally, we use an LLM-based planner to place related objects based on the merged affordance field, thus generating a shared virtual space for remote users (Section 3.3).

3.1 Interaction Aware Template

We introduce the Interaction Aware Template, a novel virtual shared scene layout representation, which incorporates LLM reasoning for a joint analysis of the physical scene of the remote users and interaction requirements. Later, it is used to guide the alignment of remote scenes and generate the merged affordance field to synthesize the shared virtual scene.

3.1.1 Definition

We define Interaction Aware Template as a set of template elements δ_i.

$$\Delta = \{\delta_1, \delta_2, \delta_3, ...\delta_n\} \tag{1}$$

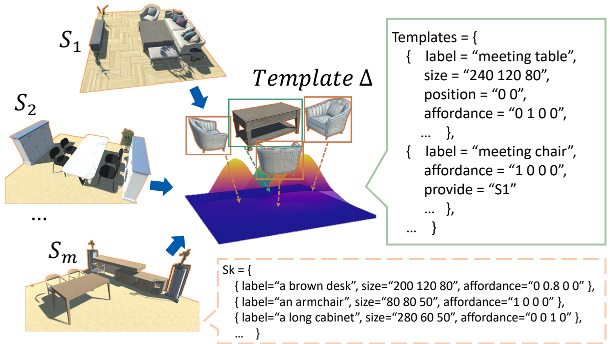
A template element is a region that must provide a specific interaction affordance [56] in the shared virtual scene to meet users' interaction requirements. The distribution of template elements can represent the layout of the user's primary interaction area in the virtual space. For example, when users want to have a face-to-face meeting, three template elements on the floor plane in a "sittable-supportable-sittable" arrangement represent a potential "chair-table-chair" layout for a meeting room. Because the specific size and position of an object directly generated by LLM may cause overlap [11], we use a centroid with dimensions to represent the possible region of a template. Therefore, a template element is defined as $\delta_i = (p_i, l_i, w_i, \vec{v}_i, \varepsilon_i)$, where p_i represents the center of δ_i relative to the global center of Δ, l_i and w_i represent the length and width of the region, $\vec{v}_i$ represents the interaction affordance. In this paper, we use four types of affordances: sittable, supportable, obstructive, and walkable. For ε_i, it is an enumeration predicted by the LLM planner, representing that the affordance of δ_i is expected to be provided by the physical scenes input of the remote users.

3.1.2 LLM-Based Construction

First, we make a detailed definition of the input of the pipeline.

Users' Interaction Text. The user can describe the expected interaction between them in a text, including the type of scene, and some expected details, such as special objects, positional relationships. It is represented as **C**.

Physical Scene Representation. We assume that the information of the users' physical scenes is given by the grid model of RGBD scans and semantic annotation methods [57–59]. We denote the physical scene of remote user i as $S_i = \{O_i, \Omega_i\}, i \in \{1, ...m\}$, where O_i represents the set of objects $\{o_1, o_2, ..., o_k\}$ in the scene. Each object o_j contains the semantic label l_j, the size s_j, the position p_j, and the rotation r_j of the oriented bounding box. Ω_i indicates the region of the floor in the top view. Similarly, the synthesized virtual scene is represented by V.



```
Templates = {
  {   label = "meeting table",
      size = "240 120 80",
      position = "0 0",
      affordance = "0 1 0 0",
     ...   },
  {   label = "meeting chair",
      affordance = "1 0 0 0",
      provide = "S1"
      ...  },
 ...   }
```

```
Sk = {
   { label="a brown desk", size="200 120 80", affordance="0 0.8 0 0" },
   { label="an armchair", size="80 80 50", affordance="1 0 0 0" },
   { label="a long cabinet", size="280 60 50", affordance="0 0 1 0" },
   ...   }
```

Fig. 2: With the text input "we want a remote meeting", we generate Template(solid line box) and affordance of the input objects(dashed box) based on LLM planner.

Then, we construct an IAT based on the users' interaction text and remote physical scenes via LLM. We use the LLM planner $\mathbf{L}_\Delta$ to generate a set of templates with affordance to represent the layout of the core interaction areas in the shared space based on **C**. Fig. 2 gives an example of Eq. (2).

$$\mathbf{L}_\Delta(S_1, ...S_m, \mathbf{C}) \longrightarrow \Delta = \{\delta_i\} \tag{2}$$

3.2 IAT based Affordance Field Alignment

IAT represents scene layout information that is necessary to complete interaction tasks based on user interaction requirements. On the other hand, we propose the affordacne field to represent the layout information that the user's physical space can provide.

We define $\mathbf{M}_{aff}$ as the affordance field for the input physical scene. We ask the LLM planner $\mathbf{L}_{aff}$, combined with the user interaction

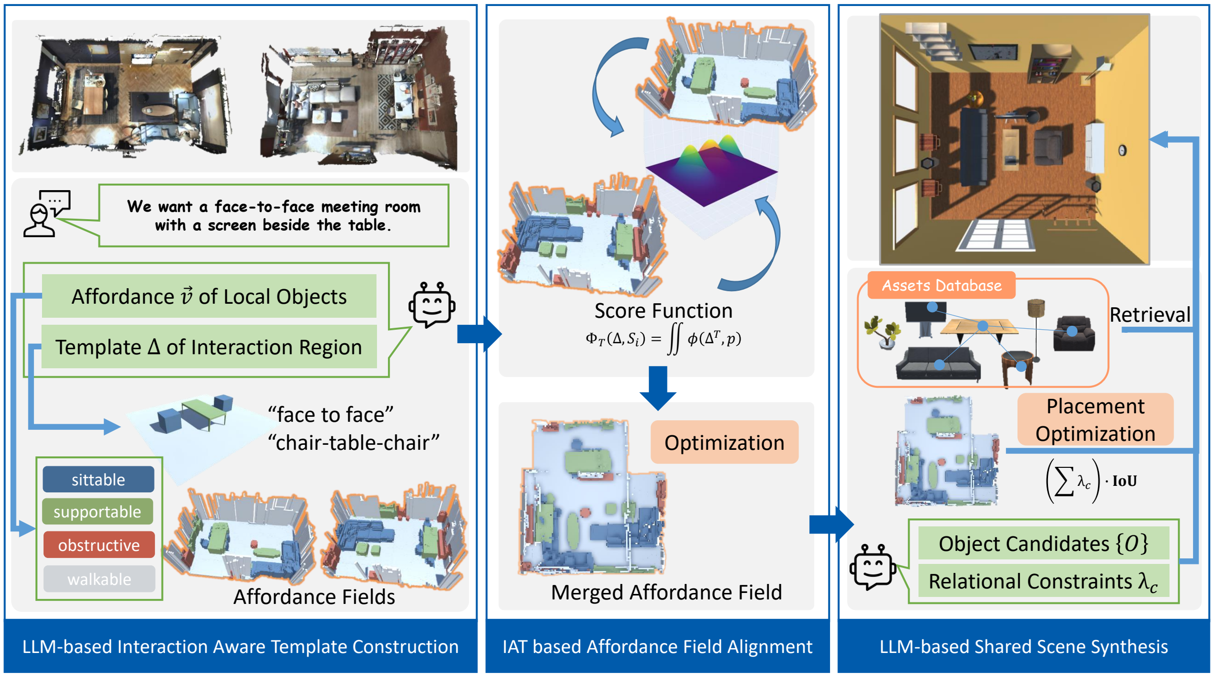

We want a face-to-face meeting room with a screen beside the table.
Affordance $\vec{v}$ of Local Objects
Template Δ of Interaction Region
“face to face”
“chair-table-chair”
sittable
supportable
obstructive
walkable
Affordance Fields
Score Function
$\Phi_T(\Delta, S_i) = \iint \phi(\Delta^T, p)$
Optimization
Merged Affordance Field
Assets Database
Retrieval
Placement Optimization
$\left(\sum \lambda_c\right) \cdot \mathbf{IoU}$
Object Candidates $\{O\}$
Relational Constraints λ_c
LLM-based Interaction Aware Template Construction
IAT based Affordance Field Alignment
LLM-based Shared Scene Synthesis

Fig. 3: The pipeline of our method.

requirement text $\mathbf{C}$, to predict the affordance $\vec{v}_{o_j}$ for each object o_j. Note that $\|\vec{v}_{o_j}\| \in [0,1]$ could measure the importance between different objects with the same type of affordance.

$$\mathbf{L}_{aff}(o_j|S_i,\mathbf{C}) \longrightarrow \vec{v}_{o_j} \tag{3}$$

To calculate the affordance field, we project rays in S_i from top to bottom in Ω_i. The object with which the ray at p first intersects determines the affordance of p, which is denoted as $\vec{v}_p = \vec{v}_{o_j} \in \mathbf{R}^4$. The affordance field of S_i is denoted as below:

$$\forall p \in \Omega_i, \mathbf{L}_{aff}(o_j|S_i,\mathbf{C}) \longrightarrow \vec{v}_{o_j} = \mathbf{M}_i^{aff}(p) \tag{4}$$

The search for shared space among remote users can be transformed into the search for the optimal transformation between affordance fields, thereby obtaining a merged area to support users' remote interaction. To obtain a merged affordance field to synthesize the shared virtual scene, we propose an IAT-based affordance field alignment algorithm to find the optimal rotation and translation $T_{S_1,S_2} = (r,t)$ between the input physical scenes. However, direct traversal of all possible alignments between input physical scenes is complex and time-consuming, and it is hard to evaluate the quality of different alignment results. Instead, we use IAT Δ as a bridge to align physical scenes. The alignment between Δ and S_i can naturally generate a shared area based on the template space. Fig. 2 shows the alignment between $S_{1,\dots,m}$ through a template. We use a matching score based optimization algorithm to search for the appropriate alignment $T_{S_i \to \Delta}$ of the user scenes S_i relative to Δ separately based on a score function, to obtain the alignment T_{S_i,S_j} between the user scenes.

3.2.1 Matching Score Function

The matching score function is introduced to measure the matching degree between S_i and Δ.

To calculate this function, we first establish a coordinate space for the IAT Δ, where the origin is at the center of Δ. For each template element $\delta_i = (p_i, l_i, w_i, \vec{v}_i, \varepsilon_i)$ in Δ, we model its probability distribution in the floor plane through a two-dimensional anisotropic Gaussian distribution function:

$$\mathbf{G}(\delta_i, p) = \exp\left(-\frac{(x - px_{\delta_i})^2}{2 \cdot l_{\delta_i}^2} - \frac{(y - py_{\delta_i})^2}{2 \cdot w_{\delta_i}^2}\right) \tag{5}$$

where $p = (x,y)$ is a point in the coordinate space. This definition means that for a point p in the scene, the probability of providing affordance $\vec{v}_i$ to the user is $\mathbf{G}(\delta_i, p)$. The attributes in Eq. (5) could ensure that the position and orientation conform to the real space constraints and then guide the placement of virtual objects for users to interact with. ε_{δ_i} is used to mark the source user who is expected to provide the affordance, so as to make sure that different user's spaces are aligned to corresponding areas in Δ. We define $\mathbf{G}_\Delta$ as the total distribution of Δ.

After this, we calculate the matching degree between the IAT and the affordance fields of the input scenes. To reduce computational load of $T_{S_i \to \Delta}$, we fix the user physical scene S_i and apply a transformation T to Δ, such that the inverse transformation T^{-1} corresponds to the alignment of S_i relative to Δ. Within the coordinate system of S_i, we compute the matching score between Δ and S_i at position p using a score function $\phi(\cdot)$:

$$\phi(\Delta, p) = \sum_{\delta_i \in \Delta, \varepsilon_{\delta_i} = S_i} \mathbf{G}(\delta_i, p) \cdot \left(\mathbf{M}_i^{aff}(p) \otimes \vec{v}_{\delta_i}\right) \tag{6}$$

where we define the affordance operator $\otimes$ as Eq. (7). When $\vec{u} \cdot \vec{v} \neq 0$, the affordance between $\mathbf{M}_i^{aff}(p)$ and $\vec{v}_{\delta_i}$ is matched, which results in a reward of ω_1. Otherwise, when there is a mismatch at p, a penalty of $-\omega_2$ is obtained.

$$\vec{u} \otimes \vec{v} = \begin{cases} \omega_1, \text{ if } \vec{u} \cdot \vec{v} \neq 0 \\ -\omega_2, \text{ else} \end{cases} \tag{7}$$

Finally, for a given transformation $T : \Delta \to \Delta^T$, the matching score between the user scene S_i and Δ can be expressed as:

$$\Phi_T(\Delta, S_i) = \iint_{p \in \Omega_i} \phi(\Delta^T, p) \tag{8}$$

3.2.2 Optimization

In order to obtain the merged affordance field, we search for the optimal alignment of each affordance field $\mathbf{M}_i^{aff}$ for Δ separately through the score function Φ, merge the affordance fields and fine-tune it to obtain the final result. We use a simulated annealing algorithm to implement this process. The IAT-affordance field alignment optimization algorithm is given by Algorithm 1. To simplify the calculation, we first grid the affordance fields by dividing the floor plane area Ω_i into 5cm× cm × 5cm cells.

For the process, the inputs are affordance fields $\mathbf{M}_i^{aff}$ and the IAT Δ, the outputs are the optimized transformation between $\mathbf{M}_i^{aff}$ and Δ, including rotation and translation. We initialize the beginning transformation T_0 as a zero matrix, the beginning temperature t as 1, the termination temperature t_{end} as 10^{-5}, cooling factor α as 0.95, iterations i as 0, and the matching score E_0 as the score computed with the transformation T_0 (line 1). Then we set T_{best} and E_{best} with T_0 and E_0 (lines 2-3). We optimize T_{best} with the loop until t reaches t_{end} (lines 4-11). In the loop, we perturb T_i with an amplitude of σ as the next transformation T_{i+1} (line 5), and initialize the score E_{i+1} with 0 (line 6). For each cell p in the grid of Ω, we integrate the matching score ϕ at p to calculate the total score E_{i+1} of the affordance field (lines 7-8). After this, we use the Metropolis-Hastings algorithm to compute the acceptance probability and determine whether T_{i+1} is the best solution (lines 9-10). Since the only variable affected by the transformation T during optimization is δ_i in IAT, the sampling process can be efficiently parallelized.

Algorithm 1: IAT-Affordance

```
   Input: affordance field $\mathbf{M}^{aff}$; IAT $\Delta$
   Output: transformation $T$ between $\mathbf{M}^{aff}$, $\Delta$
1  Initialization: $T_0, t, t_{end}, \alpha, E, i$
2  $T_{best} \leftarrow T_0$
3  $E_{best} \leftarrow E_0$
4  while $t > t_{end}$ do
5      $T_{i+1} \leftarrow T_i +$ gaussian_distribution $(\sigma T_i)$
6      $E_{i+1} \leftarrow 0$
7      for $p \in \Omega$ do
8          $E_{i+1} \leftarrow E_{i+1} + \phi(p)$
9      if metropolis_hastings $(E_{i+1} - E_i)$ then
10         $T_{best} \leftarrow T_{i+1}, E_{best} \leftarrow E_{i+1}$
11     $t \leftarrow \alpha t$
12 Return $T_{best}$
```

Based on the transformations T_i between each affordance field $\mathbf{M}_i^{aff}$ and IAT, we can apply the inverse transformation T_i^{-1} to $\mathbf{M}_i^{aff}$ and get the merged affordance field $\mathbf{M}_\cap^{aff}$. In $\mathbf{M}_\cap^{aff}$, the affordance value in the overlapped region from different fields is accumulated by each individual component of affordance. We further fine-tune the results using the score function (Eq. (9)). Similar to related works [13, 14], our objective is to maximize the matching degree of affordance between the scenes while also maximizing the user's movable area. To achieve this, we simplify the scoring function as follows:

$$\phi^{'}(p) = \mathbf{M}_1^{aff}(p) \otimes \mathbf{M}_2^{aff}(p) \tag{9}$$

3.3 LLM-Based Shared Scene Synthesis

We take the merged affordance field $\mathbf{M}_\cap^{aff}$ and the user interaction text $\mathbf{C}$ as input, use LLM planner to generate object proxies and optimize their placements in the merged field. Finally, we retrieve 3D models of the object proxies from the database to synthesize the shared virtual scene.

First, we preprocess the merged field $\mathbf{M}_\cap^{aff}$ to calculate the movable area $\Omega_{S_\cap}$ and the list of real objects $O_\cap$. Since the merged field $\mathbf{M}_\cap^{aff}$ is centered on the center of Δ, we can extract polygons to fit the maximum movable area $\Omega_{S_\cap}$ for each user. Within $\Omega_{S_\cap}$, we need to fully ensure physical consistency. When the user's movable area $\Omega_{S_i \cap V}$ is small, we can use a larger bounding rectangle as the visual boundary of the virtual scene, and highlight the boundary of $\Omega_{S_\cap}$, thus enhancing the immersion of the scene while keeping the physical consistency inside the movable area. Then, we extract the list of real objects $O_\cap$ involved in $\Omega_{S_i \cap V}$ according to the following rule: if the object does not overlap with any object from another user space, we add it directly to $O_\cap$, else we mark the semantic labels of the overlapped objects as obstructive before adding to $O_\cap$.

Object Proxy Generation. An object proxy describes a bounding box with semantic descriptions of a virtual object expected to be placed in the shared scene. We categorize object proxies into three types: (1) Template Objects O_T: to be placed at the position of a template element in Δ and directly involved in collaboration. (2) Additional Objects O_A: to satisfy specific user interaction needs, with positions correlated to template objects, and participate in interactions either directly or indirectly. (3) Decorations O_D: not involved in user interactions, used as decorations to fill the scene. Furthermore, we define a set of constraints $\{\lambda_c\}$ for Additional Objects and Decorations, as shown in Tab. 1, to guide subsequent placement strategies.

Table 1: Object Relational Constraint

Constraint	Placement Rule
λ_{beside}	sample a position beside/near a Template Object
λ_{nearby}	sample a position very close to a Template Object
λ_{upon}	sample a position upon a Template Object
λ_{around}	sample a position around a Template Object
λ_{edge}	sample at the edge of the boundary of the scene
λ_{wall}	sample on the wall

We use LLM planner $\mathbf{L}_{obj}$ to generate suitable object proxies $O_V = \{O_T, O_A, O_D\}$ with a set of relational constraints $\{\lambda_c\}$ based on the user interaction requirement $\mathbf{C}$ (Eq. (10)).

$$\mathbf{L}_{obj}(\mathbf{C}) \longrightarrow \{O_T, O_A, O_D, \{\lambda_c\}\} \tag{10}$$

Placement Optimization. To place the proxies into the merged field, we turn it into a joint optimization problem under coverage and relation constraints $\{\lambda_c\}$. We first turn the template elements δ_i in Δ into O_T based on their semantic labels and bounding boxes, which represent the core objects for interaction. Secondly, for each real object o_r at position p in $O_\cap$, we apply a score term $\lambda_c(o_a, p) \cdot \mathrm{IoU}(o_r, o_a)$ to evaluate whether o_r is appropriate to be covered by a proxy o_a in O_A, where $\lambda_c(o_a, p)$ measures whether the proxies conforms to the relational constraint at p and $\mathrm{IoU}(o_r, o_a)$ measures the area that the proxy could cover o_r. We apply the o_a with the highest score at the position of o_r as the initial state, and update the bounding box of o_r to align the uncovered area. Then we iteratively optimize the proxies through the objective function (Eq. (11)):

$$P(O_A) = \left(\sum \lambda_c\right) \cdot \mathrm{IoU}(O_A, O_\cap) \tag{11}$$

where the $\mathrm{IoU}(\cdot)$ module encourages the cover of $O_\cap$ and punish overlap between proxies. Finally, we place the decorations to fill the scene.

Object Retrieval. Similar to [4], we adopt Objaverse [60] as our virtual object database, which is a large-scale annotated 3D object model library. Based on $\{O_T, O_A, O_D\}$, we use text, CLIP, and size similarity to retrieve the appropriate 3D asset, where the text similarity uses sentense transformer to measure the similarity between the textual description of the proxy and the object in the dataset, the CLIP similarity measures between the 2D renderings of the object in the dataset and the textual description of the proxy, and size similarity uses bounding box dimensions.

The scale factor of the 3D asset is between 0.5 and 1.5 to avoid overstretching. We allow virtual objects to completely cover the real objects in the space, but at the same time, we will ensure that the retrieved virtual objects are similar to their corresponding real objects in terms of shape, size and interaction type. For example, we can use a large piano to cover a table for a music room, but we will retrieve similar virtual chairs based on the chairs that the user needs to sit on, so as to ensure security in the virtual space. When there is ambiguity between the virtual object and the real object in the corresponding location, e.g. a meeting needs to sit face to face but there is only one chair in the real space S_1, so we have to place a virtual chair across from the real chair, we can make the virtual chair translucent to alert user to avoid security problems.

4 Evaluation

In this section, we first provide a description of our experimental settings, including the datasets and the implementation details deployed. We compare our method with state-of-the-art methods, conducting qualitative and quantitative analyzes of our experimental results.

4.1 Datasets and Implementation Details

Datasets. We evaluated our method's capability to generate suitable layouts for users while maintaining scene style diversity on both scanned and synthetic datasets. Specifically, we randomly selected 10 scanned scenes from 3RScan [19] and 20 scenes from 3D-FRONT [20], then paired them to construct user input scene pairs. For each pair of scenes, we defined 6 user interaction requirements. Inspired by [13, 14], we designed user requirements that encompass various interaction needs, such as sitting and standing, along with different scene styles, as detailed in Fig. 4.

Implementation Details. For the weight of the score function, we set $\omega_1 = 10, \omega_2 = 1$. We conducted our evaluations on a desktop with an Intel i9-13900F CPU, NVIDIA RTX 4080 GPU. For the software environment, we followed the framework AI2THOR [61] and implemented all the algorithms on Unity 2020.3.25f1. Throughout the pipeline, we use GPT-4o [62] and we have three sets of outputs through the LLM planner $\mathbf{L}_{aff}, \mathbf{L}_{\Delta}, \mathbf{L}_{obj}$, detailed information are shown in Supplemental Material.

4.2 Comparison

Since existing methods either generate a virtual scene based solely on text constraints or only match and align dissimilar scenes for telepresence, we construct the following three methods to accomplish the same task as ours for comparison.

Text-only Method. As the recent LLM-based scene synthesis method [4], it takes a text describing the scene requirements as input, uses LLM to directly generate the layout structure information of the scene and places virtual objects in it, thereby obtaining diverse virtual scenes. During the generation process, it only considers the interaction requirements given by the user and does not take into account the users physical space. Therefore, in the experiment, to ensure the user's safety, we restricted the user's movement and asked the user not to walk around.

Area-centric Method. Based on the text-only method, before the scene is generated, the area optimization method [13] is adopted to introduce the constraints of the user's physical scene. It divides the user's physical scene into sittable and walkable areas and then maximizes the area.

Target-centric Method. Based on the text-only method, before the scene is generated, the target optimization method [14] is adopted to introduce the constraints of the user's physical scene. Different from [13], the target optimization method introduces an interaction term to enhance interaction quality within the local aligned region. It specifies the interaction target and aligns the shared scene centered on the target.

4.2.1 Metrics

We evaluate our method using two categories of objective metrics. The first measures virtual-real scene consistency (Free Space Ratio, Affordance Consistency, and 3D IoU), while the second evaluates interaction satisfaction via Layout Suitability.

Free Space Ratio (FSR). Similar to [14], we report the ratio of the average user's moveable area within the shared space relative to the local real space to measure the utilization of the user's local space by the shared scene.

Affordance Consistency (AC). To measure the affordance consistency between the aligned user spaces, we compute the affordance match ratio at each position p in the shared virtual scene.

$$\mathrm{AC} = \frac{\sum_p \vec{v}_1 \otimes \vec{v}_2 \ldots \otimes \vec{v}_m}{\sum_p \mathbf{1}} \tag{12}$$

where $\sum$ represents the sum at each position in the shared virtual scene, $\vec{v}_i$ represents the affordance value in S_i after alignment. We set $\omega_1 = 1, \omega_2 = 0$ in Eq. (7).

3D IoU. We measure the 3D intersection over union (3D IoU) to evaluate the degree of overlap between virtual objects and real objects in the aligned physical space. The value closer to 1 indicates that the placed virtual objects better cover the real objects while avoiding unnecessary occupation of moveable areas for users.

Layout Suitability (LS). Similar to [32, 63], we render a top view of the shared scene, and then ask GPT-4o to score (from 0 to 100) whether the layout meets the remote interaction requirements described by users.

4.2.2 Results and Discussion

For comparison on the 3D-FRONT [20] dataset, which contains a large number of synthetic scenes of living-rooms, bedrooms, and libraries, we randomly select living-rooms to simulate daily scenarios for remote interaction. Compared to structured synthetic scenes, real-world scanned scenes from 3RScan [19] dataset exhibit greater diversity in scene types and layouts, while containing more occlusions and interference factors. We follow [13, 14] and divide the interaction types described in user interaction text into two main categories: sit and walk. Specifically, for scenarios where sitting is the main interaction state, we design structured descriptions such as "face-to-face", "side-by-side", etc., to test the ability of the method to find an appropriate alignment that meets user needs. For scenarios where walking or standing is the main interaction state, we further test the ability to satisfy the user's needs by placing appropriate virtual objects after finding the maximum movable area.

Fig. 4 shows examples of scenes synthesized by our method. The first column presents three pairs of remote physical spaces as input, including two pairs of scanned scenes and one pair of synthesized scenes. The second and third columns show the shared interaction scenes generated by our method under different types of interaction requirements, along with the corresponding merged fields. Our method aligns user physical spaces according to the implicit spatial relationships and interaction types from different user requirements and synthesizes interaction-aware virtual scenes with diverse styles. Specifically, the generated meeting room, classroom, and library demonstrate that our method effectively identifies physical-consistent local alignments through affordance analysis and template matching, allowing users to sit and interact safely within the movable area. The workshop, music room, and VR game room examples show that our method can reasonably place virtual objects to satisfy user interaction needs while avoiding collisions with the physical environment.

Fig. 5 shows examples of our method with comparison methods. In the collaborative scenario of a workshop (row 1), the Area-centric method searches for the largest mutual walkable area, and the Target-centric method uses the floor as the interaction target. However, they include numerous real-world objects that are unrelated to the interaction and the mismatch between different user spaces. Furthermore, compared to LLM directly predicting virtual objects to cover the real space, the optimization in our scene synthesis method can better reduce overlap and unreasonable placement. Text-only method tends to generate numerous common sense room layouts through LLM-generated constraints and place objects along the walls, but it fails to meet the demands of specific interaction behaviors. In contrast, although our scene synthesis method adopts a hierarchical placement strategy similar to the anchor and relational constraints from Holodeck [4], we align the shared walkable area with a walkable-affordance template and place a workbench and relevant objects at the center, resulting in user-centric layout that better supports user interaction needs. Similarly, in the VR gaming scenario (row 2), our method achieves a good balance between virtual object diversity and physical consistency through layout optimization.

We provide quantitative comparison results in Tab. 2. For sitting scenarios, our method achieved optimal results on all metrics. Text-only method, which restricts users' movement within shared space, obtained the lowest FSR and AC. Benefiting from Holodeck [4]'s powerful scene generation capability, the Text-only method achieved the second-highest LS, but results show that it is not suitable for shared VR scenes. This finding indicates that directly employing existing

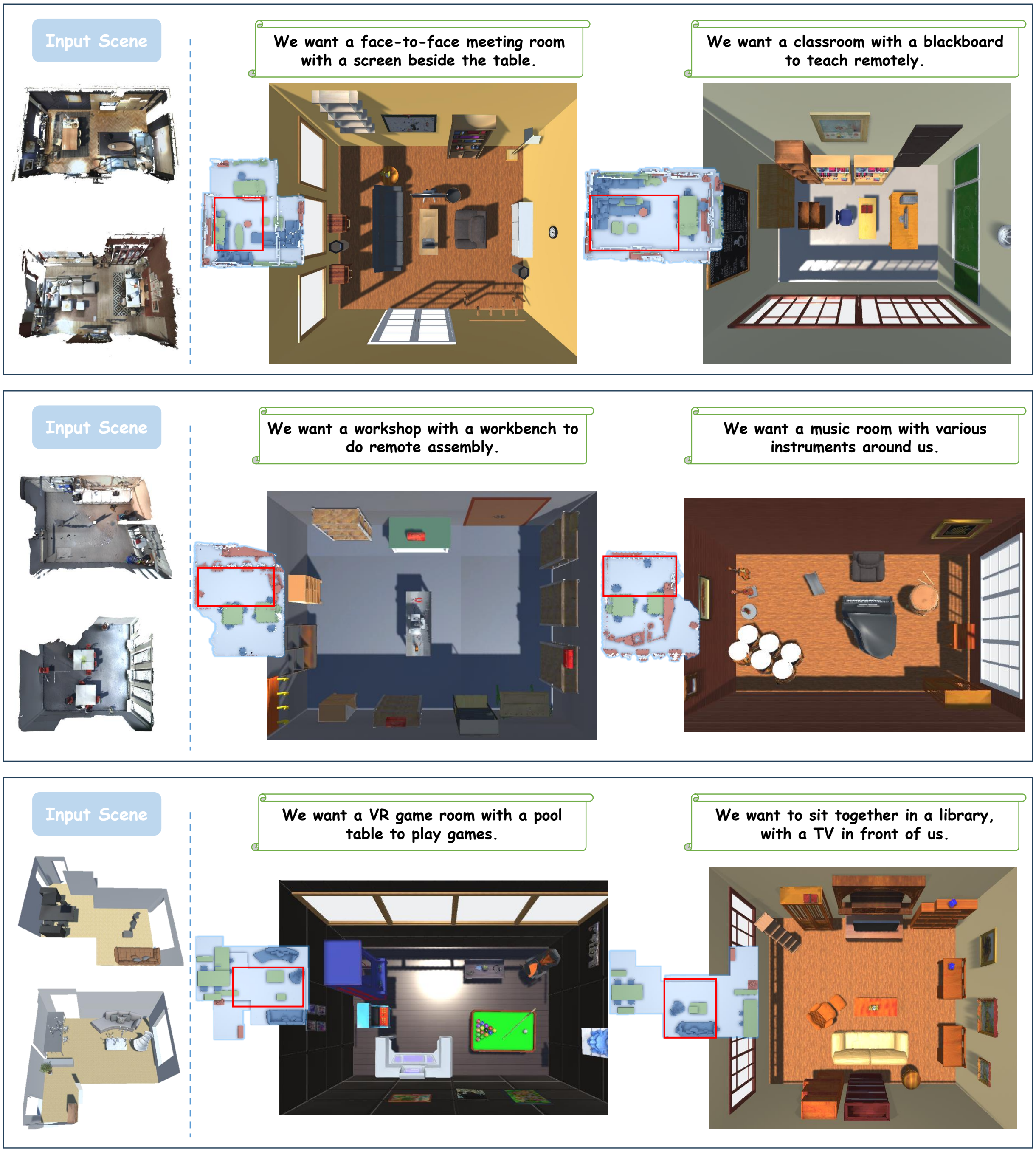

Input Scene
We want a face-to-face meeting room with a screen beside the table.
We want a classroom with a blackboard to teach remotely.
Input Scene
We want a workshop with a workbench to do remote assembly.
We want a music room with various instruments around us.
Input Scene
We want a VR game room with a pool table to play games.
We want to sit together in a library, with a TV in front of us.

Fig. 4: Examples of the synthesized scenes

Table 2: Quantitative comparison results

Dataset	Interaction Type	sit				walk			
	Metric	FSR(↑)	AC(↑)	3D IoU(↑)	LS(↑)	FSR(↑)	AC(↑)	3D IoU(↑)	LS(↑)
3D-FRONT	Text-only	23.06	27.15	0.32	83.17	-	-	0.36	86.32
	Area-centric	49.40	45.87	0.58	65.28	**64.03**	61.32	0.51	68.37
	Target-centric	46.73	74.14	0.65	74.36	52.87	74.05	0.62	78.50
	Ours	**55.85**	**82.32**	**0.76**	**90.25**	60.28	**86.32**	**0.84**	**93.45**
3RScan	Text-only	27.49	38.25	0.21	75.43	-	-	0.28	78.37
	Area-centric	43.74	51.04	0.45	56.82	**70.85**	60.87	0.52	60.25
	Target-centric	32.38	56.15	0.62	70.18	54.29	78.39	0.58	69.06
	Ours	**48.52**	**74.23**	**0.67**	**88.32**	62.16	**83.51**	**0.73**	**91.62**

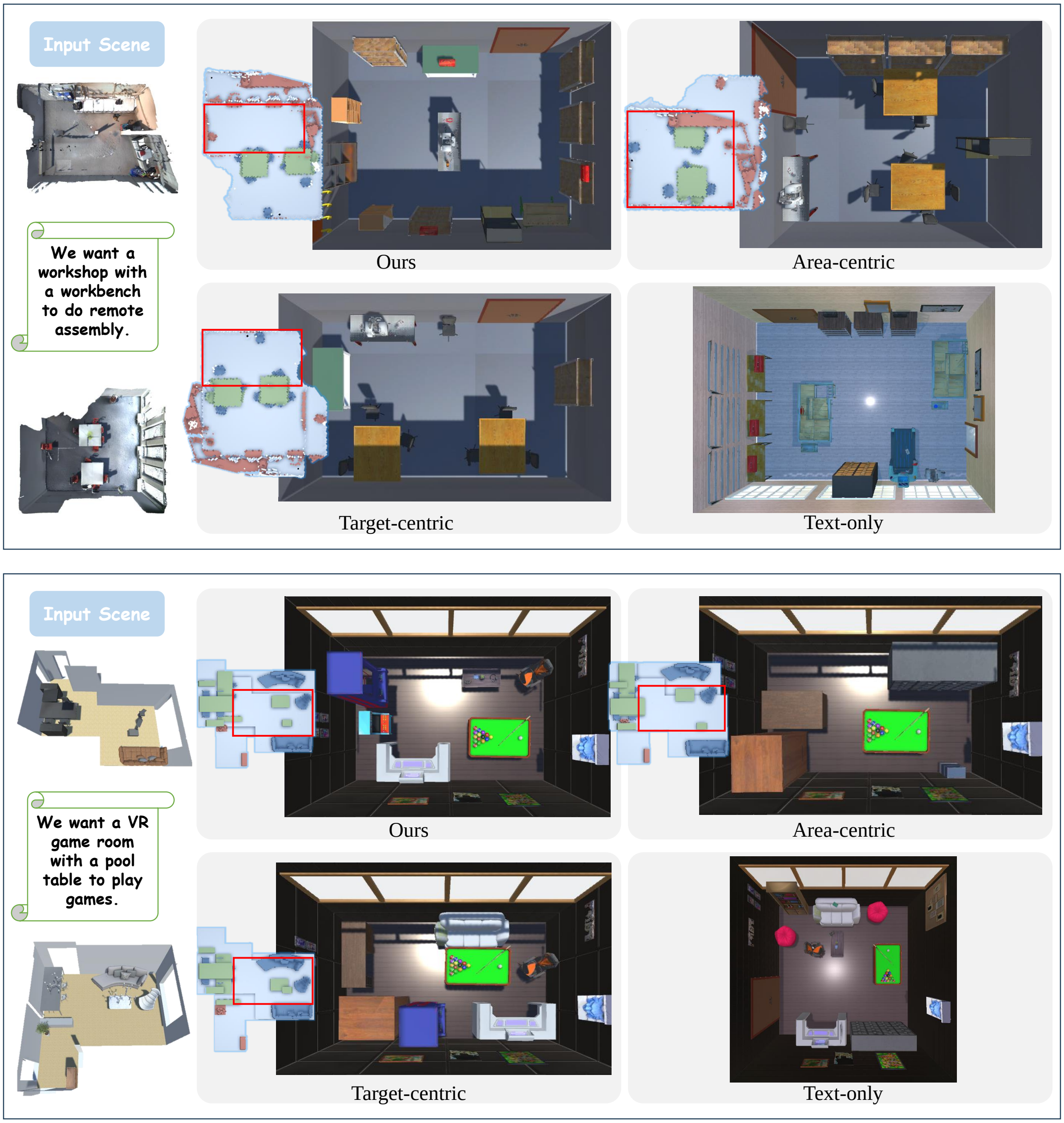

Input Scene
We want a workshop with a workbench to do remote assembly.
Ours
Area-centric
Target-centric
Text-only
Input Scene
We want a VR game room with a pool table to play games.
Ours
Area-centric
Target-centric
Text-only

Fig. 5: Comparison between the four methods

scene generation pipelines or pre-designed virtual scenes for remote user interaction is inappropriate, as it cannot provide interaction-aware spaces based solely on textual descriptions, while also validating the importance of exploring VR remote interaction scene synthesis methods. For walk-dominant scenarios, our method achieved optimal results on three metrics except FSR. Notably, the Text-only method does not calculate alignment between virtual and real scenes due to fixed positions for users, therefore we did not calculate its FSR and AC metrics. The Area-centric method computes the maximized aligned area as the shared space, achieving the highest FSR in walk-dominant scenarios. However, this method overlooks geometric and semantic information within the space, producing numerous affordance mismatch regions that make it difficult to balance between maintaining reasonable layouts and covering real obstacles for the arrangement of virtual objects. Comparatively, the Target-centric method outperforms the Area-centric method in terms of AC and 3D IoU metrics by selecting appropriate target object as local alignment reference. However, results for sit-dominant scenarios on 3RScan [19] dataset indicate limited improvement because it struggles to convert implicit layout relationships from user requirements into appropriate target object.

5 USER STUDY

We designed a within-subject study to evaluate the perceptual synthesis quality of our method compared with the comparison methods.

Participants and Setup. We recruited 14 participants (7 males and 7 females, aged between 21-35 years), all of whom had experience using VR HMDs and had normal vision. Each participant wore an HTC Cosmos headset for the study.

Conditions. The conditions are included: our method (*Ours*), Text-only method, Area-centric method, and Target-centric method.

Task. We randomly selected 6 groups of scenes and user interaction requirements $(S_1, S_2, \mathbf{C}, V)$ from the 3D-FRONT [20] livingroom. For each group, participants take S_1 as their local physical space and browse V in VR as the generated remote shared scene under the interaction requirement $\mathbf{C}$. To be detailed, during the experiment, the room is arranged according to the layout of S_1, participants first moved around the room to get familiar with the physical space, then they wore VR headsets to browse the generated scene V. They were informed of the current interaction requirements $\mathbf{C}$ and could switch the display of the physical space S_1 and the shared scene V with the button from their headset controller. Note that since the Text-only method does not take into account the user's physical space but directly transports them to a fixed location in the virtual scene, we restrict the user's movement under this method. Participants completed experiments according to the order of the method determined by the balanced Latin square. Then they scored the following questions similar to [7]. **Q1**: The shared space is well designed and matches the user interaction requirement. **Q2**: The shared space is well aligned with the physical space and makes user feel safe. Participants filled in the scores for the two questions for each scenario on a 5-point Likert scale (with 1 indicating the least compliant and 5 indicating the most compliant). To mitigate the effects of visual fatigue, after completing the questions, participants are given a 10 second rest before proceeding to the next method.

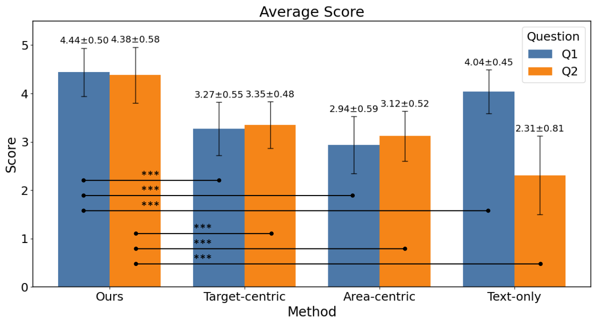


Fig. 6: The average scores and standard deviations for all conditions in the user study.

Results. For each condition, a total of 84 sets of data are collected, and no outliers(±3 standard deviation) were found. Fig. 6 shows the results of two questions. General repeated measures ANOVA tests and paired T-tests with correction are used to analyze the data. There is a significant difference among the four methods (**Q1**: $F_{3,249} = 402.90$, $p < 0.001$; **Q2**: $F_{3,249} = 464.01$, $p < 0.001$). The performance of our method is significantly better than Text-only (**Q1**: $t_{83} = 7.51$, $p < 0.001$; **Q2**: $t_{83} = 37.24$, $p < 0.001$), Area-centric **Q1**: $t_{83} = 27.33$, $p < 0.001$; **Q2**: $t_{83} = 26.14$, $p < 0.001$), and Target-centric (**Q1**: $t_{83} = 28.52$, $p < 0.001$; **Q2**: $t_{83} = 26.20$, $p < 0.001$). The results demonstrate that the user's subjective perception of interaction fitness and sense of safety were significantly improved with our method. After the experiment, we further interviewed the participants about their feelings towards the telepresence system. Regarding the Text-only method, 8 participants indicated that although the virtual scene brought a good experience, they strongly hoped to be able to move freely for interaction (e.g., "If you can't move, then it's no different from sitting in front of a computer."). 5 participants reported that they were worried about touching local obstacles at the beginning, but soon felt safe after getting familiar with them. This might be because users did not have a sufficient understanding of the layout of S_1 at the beginning. 11 participants expressed their expectations for the remote interaction system (e.g., "It is a good application. I can experience different VR environments with remote friends at any time.").

6 Conclusion, Limitations and Future Work

We have proposed a novel interaction-aware shared virtual scene synthesis method that uses LLMs based on users' physical environments and interaction requirements. We introduce Interaction Aware Template as a bridge between users' physical information and shared interaction demands, then we propose an IAT-based affordance field alignment algorithm and a LLM-based scene synthesis methods, enabling remote immersive interactions while ensuring physical consistency. We validate our method on both real and synthetic datasets. Objective quantitative results show that for sitting interactions, our method outperforms the SOTA methods in terms of consistency and suitability. For walking interactions, our method also outperforms the SOTA method, except for FSR, where it is slightly inferior to the area-centric method. The user study shows that perceived quality significantly improved. In summary, our method better integrates the space alignment with the user requirements and introduces the powerful understanding of LLMs to obtain reasonable and diverse results. It is also more suitable for VR collaboration scenarios due to a well-designed IAT representation with affordance, which integrates the physical environment with the generative pipeline and fills the gap in the application of cross-modal generative frameworks in VR telepresence.

Despite introducing a novel approach to shared space generation, our method still has certain limitations. First, the shared space are restricted by the users' local space. When there is a significant difference in size or layout between user spaces, or when it turns to multi-user scenarios, our method may be forced to reduce the user's movable area, thereby affecting the interaction experience. Future work could explore partitioning the Templates for multi-user settings, extending more types of affordances, or integrating scene generation pipelines with redirected walking techniques [64] and avatar control methods [16, 18] to support large-scale, multi-user interaction-aware shared virtual spaces. Second, our method retrieves and places objects from a predefined asset library, but the similarity between virtual and real-world objects remains limited. To address this, future work could integrate 3D generative methods [1], combined with image and depth models to enable better alignment for the local environment, thereby improving visual coherence between virtual and real objects. The randomness of the results generated by LLMs also forces us to explore more reliable mechanisms to ensure user safety. Third, the user study did not further explore the time and success rate of users completing specific remote tasks, which will be our future work. Additionally, the shared scenes generated by our method are static and cannot be updated in real time according to the changes in the user's physical space. To enhance interactivity in virtual spaces, future work could integrate real-time systems, leveraging LLM-generated code [34, 35] to dynamically respond to user actions, further unlocking the potential of VR in remote collaboration, education, and beyond.

Acknowledgments

This work is supported by the National Natural Science Foundation of China through Projects U25A20441, 62507002 and 62372026, the National Key R&D plan 2025YFE0203800, the Fundamental Research Funds for the Central Universities, and the FDCT-MOST (0018/2025/AMJ).

References

[1] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21401–21412, June 2024. 1, 9

[2] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in Neural Information Processing Systems*, 34:12013–12026, 2021. 1, 2

[3] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20507–20518, 2024. 1, 2

[4] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16227–16237, 2024. 1, 2, 5, 6

[5] Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. Gaussianavatar: Towards realistic human avatar modeling from a single video via animatable 3d gaussians. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 634–644, 2024. 1

[6] Zhe Li, Zerong Zheng, Lizhen Wang, and Yebin Liu. Animatable gaussians: Learning pose-dependent gaussian maps for high-fidelity human

avatar modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19711–19722, 2024. 1

[7] Haiyan Jiang, Leiyu Song, Dongdong Weng, Zhe Sun, Huiying Li, Xiaonuo Dongye, and Zhenliang Zhang. In situ 3d scene synthesis for ubiquitous embodied interfaces. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 3666–3675, 2024. 1, 2, 9

[8] Paul Streli, Rayan Armani, Yi Fei Cheng, and Christian Holz. Hoov: Hand out-of-view tracking for proprioceptive interaction using inertial sensing. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2023. 1

[9] Lung-Pan Cheng, Eyal Ofek, Christian Holz, and Andrew D Wilson. Vroamer: generating on-the-fly vr experiences while walking inside large, unknown real-world building environments. In *2019 IEEE conference on virtual reality and 3D user interfaces (VR)*, pages 359–366. IEEE, 2019. 1, 2

[10] Lior Shapira and Daniel Freedman. Reality skins: Creating immersive and tactile virtual environments. In *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 115–124. IEEE, 2016. 1, 2

[11] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36:18225–18250, 2023. 1, 2, 3

[12] Nicolas H Lehment, Daniel Merget, and Gerhard Rigoll. Creating automatically aligned consensus realities for ar videoconferencing. In *2014 IEEE international symposium on mixed and augmented reality (ISMAR)*, pages 201–206. IEEE, 2014. 2, 3

[13] Mohammad Keshavarzi, Allen Y Yang, Woojin Ko, and Luisa Caldas. Optimization and manipulation of contextual mutual spaces for multi-user virtual and augmented reality interaction. In *2020 IEEE conference on virtual reality and 3D user interfaces (VR)*, pages 353–362. IEEE, 2020. 2, 3, 5, 6

[14] Dooyoung Kim, Seonji Kim, Selin Choi, and Woontack Woo. Spatial affordance-aware interactable subspace allocation for mixed reality telepresence. In *2024 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 1256–1265. IEEE, 2024. 2, 3, 5, 6

[15] Seonji Kim, Dooyoung Kim, Jae-Eun Shin, and Woontack Woo. Object cluster registration of dissimilar rooms using geometric spatial affordance graph to generate shared virtual spaces. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 796–805. IEEE, 2024. 2, 3

[16] Leonard Yoon, Dongseok Yang, Jaehyun Kim, ChoongHo Chung, and Sung-Hee Lee. Placement retargeting of virtual avatars to dissimilar indoor environments. *IEEE Transactions on Visualization and Computer Graphics*, 28(3):1619–1633, 2020. 2, 3, 9

[17] Xuanyu Wang, Hui Ye, Christian Sandor, Weizhan Zhang, and Hongbo Fu. Predict-and-drive: Avatar motion adaption in room-scale augmented reality telepresence with heterogeneous spaces. *IEEE Transactions on Visualization and Computer Graphics*, 28(11):3705–3714, 2022. 2, 3

[18] Yi-Jun Li, Hao-Zhong Yang, Wen-Tong Shu, and Miao Wang. Semantics-aware avatar locomotion adaption for indoor cross-scene ar telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 2, 3, 9

[19] Johanna Wald, Armen Avetisyan, Nassir Navab, Federico Tombari, and Matthias Nießner. Rio: 3d object instance re-localization in changing indoor environments. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7658–7667, 2019. 2, 6, 8

[20] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 2, 6, 8

[21] Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023. 2

[22] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7749–7762, 2024. 2

[23] Xiaoyu Zhou, Xingjian Ran, Yajiao Xiong, Jinlin He, Zhiwei Lin, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Gala3d: Towards text-to-3d complex scene generation via layout-guided generative gaussian splatting. In *International Conference on Machine Learning*, pages 62108–62118. PMLR, 2024. 2

[24] Wamiq Reyaz Para, Paul Guerrero, Niloy Mitra, and Peter Wonka. Cofs: Controllable furniture layout synthesis. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023. 2

[25] Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. Physcene: Physically interactable 3d scene synthesis for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16262–16272, 2024. 2

[26] Misha Sra, Sergio Garrido-Jurado, and Pattie Maes. Oasis: Procedurally generated social virtual spaces from 3d scanned real spaces. *IEEE transactions on visualization and computer graphics*, 24(12):3174–3187, 2017. 2

[27] Jackie Yang, Christian Holz, Eyal Ofek, and Andrew D Wilson. Dreamwalker: Substituting real-world walking experiences with a virtual reality. In *Proceedings of the 32nd annual ACM symposium on user interface software and technology*, pages 1093–1107, 2019. 2

[28] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. *arXiv preprint arXiv:2305.11337*, 2023. 2

[29] Bangbang Yang, Wenqi Dong, Lin Ma, Wenbo Hu, Xiao Liu, Zhaopeng Cui, and Yuewen Ma. Dreamspace: Dreaming your room space with text-driven panoramic texture propagation. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 650–660. IEEE, 2024. 2

[30] Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Li Dong, Lei Cui, and Furu Wei. Mind's eye of llms: Visualization-of-thought elicits spatial reasoning in large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 2

[31] Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024. 2

[32] Yian Wang, Xiaowen Qiu, Jiageng Liu, Zhehuan Chen, Jiting Cai, Yufei Wang, Tsun-Hsuan Johnson Wang, Zhou Xian, and Chuang Gan. Architect: Generating vivid and interactive 3d scenes with hierarchical 2d inpainting. *Advances in Neural Information Processing Systems*, 37:67575–67603, 2024. 2, 6

[33] Zhipeng Li, Christoph Gebhardt, Yves Inglin, Nicolas Steck, Paul Streli, and Christian Holz. Situationadapt: Contextual ui optimization in mixed reality with situation awareness via llm reasoning. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2024. 2

[34] Daniele Giunchi, Nels Numan, Elia Gatti, and Anthony Steed. Dreamcodevr: Towards democratizing behavior design in virtual reality with speech-driven programming. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 579–589. IEEE, 2024. 2, 9

[35] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2024. 2, 9

[36] Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An llm agent for synthesizing 3d scenes as blender code. In *Forty-first International Conference on Machine Learning*, 2024. 2

[37] Alan Y Cheng, Meng Guo, Melissa Ran, Arpit Ranasaria, Arjun Sharma, Anthony Xie, Khuyen N Le, Bala Vinaithirthan, Shihe Luan, David Thomas Henry Wright, et al. Scientific and fantastical: Creating immersive, culturally relevant learning experiences with augmented reality and large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–23, 2024. 2

[38] Lei Zhang, Jin Pan, Jacob Gettig, Steve Oney, and Anhong Guo. Vrcopilot: Authoring 3d layouts with generative ai models in vr. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–13, 2024. 2

[39] Zhizhuo Yin, Yuyang Wang, Theodoros Papatheodorou, and Pan Hui. Text2vrscene: Exploring the framework of automated text-driven generation system for vr experience. In *2024 IEEE Conference Virtual Reality and 3D User Interfaces (VR)*, pages 701–711. IEEE, 2024. 2

[40] Sergio Orts-Escolano, Christoph Rhemann, Sean Fanello, Wayne Chang, Adarsh Kowdle, Yury Degtyarev, David Kim, Philip L Davidson, Sameh

Khamis, Mingsong Dou, et al. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th annual symposium on user interface software and technology*, pages 741–754, 2016. 3

[41] Tomislav Pejsa, Julian Kantor, Hrvoje Benko, Eyal Ofek, and Andrew Wilson. Room2room: Enabling life-size telepresence in a projected augmented reality environment. In *Proceedings of the 19th ACM conference on computer-supported cooperative work & social computing*, pages 1716–1725, 2016. 3

[42] Emily Wong, Adélaïde Genay, Jens Emil Sloth Grønbæk, and Eduardo Velloso. Spatial heterogeneity in distributed mixed reality collaboration. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2025. 3

[43] Dongseok Yang, Jiho Kang, Taehei Kim, and Sung-Hee Lee. Visual guidance for user placement in avatar-mediated telepresence between dissimilar spaces. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7558–7570, 2024. 3

[44] Jiho Kang, Dongseok Yang, Taehei Kim, Yewon Lee, and Sung-Hee Lee. Real-time retargeting of deictic motion to virtual avatars for augmented reality telepresence. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 885–893. IEEE, 2023. 3

[45] Jiho Kang, Taehei Kim, Hyeshim Kim, and Sung-Hee Lee. Real-time translation of upper-body gestures to virtual avatars in dissimilar telepresence environments. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 3

[46] Nels Numan, Shwetha Rajaram, Balasaravanan Thoravi Kumaravel, Nicolai Marquardt, and Andrew D Wilson. Spaceblender: Creating context-rich collaborative spaces through generative 3d scene blending. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, pages 1–25, 2024. 3

[47] Mohammad Keshavarzi, Michael Zollhoefer, Allen Y Yang, Patrick Peluse, and Luisa Caldas. Mutual scene synthesis for mixed reality telepresence. *arXiv preprint arXiv:2204.00161*, 2022. 3

[48] Maryam Okhovvat, Elham Andaroodi, and Morteza Okhovvat. Generating common spaces through virtual reality telepresence and shared scene synthesis. *Journal of Building Engineering*, 91:109508, 2024. 3

[49] Jens Emil Sloth Grønbæk, Ken Pfeuffer, Eduardo Velloso, Morten Astrup, Melanie Isabel Sønderkær Pedersen, Martin Kjær, Germán Leiva, and Hans Gellersen. Partially blended realities: Aligning dissimilar spaces for distributed mixed reality meetings. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–16, 2023. 3

[50] Ben J Congdon, Tuanfeng Wang, and Anthony Steed. Merging environments for shared spaces in mixed reality. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, pages 1–8, 2018. 3

[51] Qixiang Ma, Lili Wang, Wei Ke, and Sio-Kei Im. Smigraph: a perceptually retained method for passive haptics-based migration of mr indoor scenes. *The Visual Computer*, 40(11):8023–8043, 2024. 3

[52] Michael Pabst, Linda Rudolph, Nikolas Brasch, Verena Biener, Chloe Eghtebas, Ulrich Eck, Dieter Schmalstieg, and Gudrun Klinker. Mrunion: Asymmetric task-aware 3d mutual scene generation of dissimilar spaces for mixed reality telepresence. *IEEE Transactions on Visualization and Computer Graphics*, 2025. 3

[53] Eric R Bachmann, Eric Hodgson, Cole Hoffbauer, and Justin Messinger. Multi-user redirected walking and resetting using artificial potential fields. *IEEE transactions on visualization and computer graphics*, 25(5):2022–2031, 2019. 3

[54] Tianyang Dong, Yue Shen, Tieqi Gao, and Jing Fan. Dynamic density-based redirected walking towards multi-user virtual environments. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*, pages 626–634. IEEE, 2021. 3

[55] Dooyoung Kim and Woontack Woo. Edge-centric space rescaling with redirected walking for dissimilar physical-virtual space registration. In *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 829–838. IEEE, 2023. 3

[56] James J Gibson. The theory of affordances:(1979). In *The people, place, and space reader*, pages 56–60. Routledge, 2014. 3

[57] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, et al. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile rgb-d data. *arXiv preprint arXiv:2111.08897*, 2021. 3

[58] David Rozenberszki, Or Litany, and Angela Dai. Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19957–19967, 2024. 3

[59] Akshay Gadi Patil, Supriya Gadi Patil, Manyi Li, Matthew Fisher, Manolis Savva, and Hao Zhang. Advances in data-driven analysis and synthesis of 3d indoor scenes. In *Computer Graphics Forum*, volume 43, page e14927. Wiley Online Library, 2024. 3

[60] 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. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023. 5

[61] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, Aniruddha Kembhavi, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai, 2022. 6

[62] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6

[63] Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. Layoutvlm: Differentiable optimization of 3d layout via vision-language models. *arXiv preprint arXiv:2412.02193*, 2024. 6

[64] Marc Aurel Störmer, Thereza Schmelter, Malte Weingart, Levente Hernadi, Johannes Hoster, Eike Langbehn, and Kristian Hildebrand. A study on multi-user interaction-based redirected walking. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, SUI '23, New York, NY, USA, 2023. Association for Computing Machinery. 9