2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Assistant Placement Aria: A Benchmark for Egocentric Placement Assistance

Amir Belder[1], Gonçalo Dias Pais[2], Refael Vivanti[3], Omri Carmi[3], Daniel DeTone[3], Oren Shrout[4], Ido Gattegno[3], and Ayellet Tal[4]

***Abstract*— Human assistance in robotics spans several tasks such as navigation, object manipulation, and placement, where a key challenge is selecting target destinations that align with human intentions or preferences. We focus on this challenge in the context of Virtual Placement (VP), the task of identifying all plausible target locations given scene context and human-centric constraints. This differs from traditional placement tasks that typically focus on a single, predefined target location. The VP problem is complex, as it requires both global and local reasoning about the scene's geometry, semantics, and plausibility. To address this gap, we introduce Assistant Placement Aria, the first benchmark to explore diverse aspects of VP, including global, local, and human-centric constraints. It contains both synthetic and real indoor scenes annotated for three tasks: (i) 2D Panel Placement, (ii) Sitting Suggestion, and (iii) TV Placement. Each scene includes 2D images, a 3D point cloud, and a textual description of the objects within the scene. By contributing this benchmark, we aim to encourage further research in this underexplored and challenging field that is critically dependent on relevant data. We also evaluate several foundation models for object detection and segmentation on our benchmark. The benchmark is available at: https://github.com/amirbelder/-Placement-Aria—Benchmark-for-Egocentric-Placement-Assistance.**

I. Introduction

Human assistance is a central theme in a variety of robotics tasks in both industrial and household settings. Such tasks include object placement and manipulation, as well as navigation to pre-defined target destinations for assistive purposes [1], [2], [3], [4]. A key challenge in these scenarios is determining which target destinations best reflect human intentions or preferences, a problem that is non-trivial and remains relatively underexplored. In this work, we address this challenge in the context of assistive object placement.

In the robotics field, assistive navigation focuses on reaching a specific destination, while object placement involves transferring an object to a target location and ensuring it is positioned with the correct pose. In contrast, object placement in computer vision typically refers to inserting a single foreground object into a background image at a suitable position and scale. This task introduces several challenges that can be grouped into two categories: (1) determining the appropriate size and a *single* placement for the object [5], [6], [7], and (2) rendering the object realistically within the target image [8], [9]. In this work, we adopt the computer vision perspective, but frame it as suggestions or recommendations that can guide either a human or a robot. We refer to this problem as Virtual Placement (VP). Unlike traditional placement tasks, VP focuses on a semantic and human-centric question: *"Given the context of a scene and its environment, where would a person place the object?"* The objective is to identify *all plausible* placement locations that align with human preferences and contextual cues.

The VP problem requires considering human preferences while accounting for global and local physical constraints. For example, when placing a TV screen, it is important to identify a comfortable viewing point (e.g., a sofa or a bed). Global information is necessary to ensure that no objects obstruct the line of sight from the viewing point, while local information is required to identify vacant areas on the wall.

Some question-answering datasets contain a small number of placement-related questions [10], [11], [12], [13], but we believe these are insufficient to capture the complexity of the placement problem. In [14], a semantic (virtual) placement dataset for small man-made objects (e.g., table lamps and books) was introduced. Although this dataset achieves impressive results, its annotations do not explicitly account for human preferences, as they rely solely on the locations where objects were observed. For instance, if a book was not seen on a desk but only on bookshelves, a desk would never be considered a valid placement target. Moreover, the dataset is limited to man-made objects, whereas assistive sitting introduces additional physical and human-centric constraints, such as comfort and accessibility. For example, a person might avoid sitting on a crowded couch, while a book can be placed among other books without disrupting the scene's natural appearance. Due to these challenges, datasets covering diverse VP tasks remain scarce. This highlights the need for a new benchmark that supports broader exploration of VP constraints and enables further progress in the field.

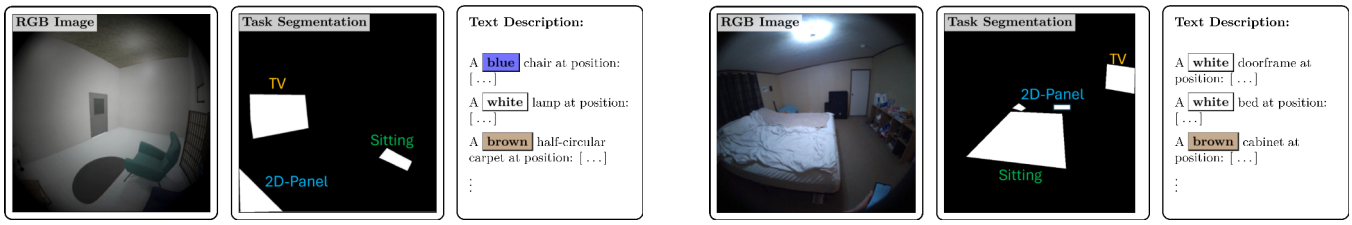
Thus, we introduce **Assistant Placement Aria**, a novel VP benchmark annotated on the Aria Synthetic Environments (ASE) [15] and Aria Everyday Objects (AEO) [16] datasets, covering three semantically relevant tasks: (1) **2D Panel Placement** (e.g., a tablet); (2) **Sitting Suggestion**; and (3) **TV Placement**. These tasks were chosen for their distinct characteristics, which together capture diverse aspects of VP. The first task, 2D Panel Placement, focuses on placing small man-made objects on different pieces of furniture. Since most such objects (e.g., books, table lamps, laptops, cups) typically require identifying vacant surfaces, we abstract them as a generic 2D horizontal panel with varying sizes to represent

[1]Amir Belder was an intern at Meta Reality Labs and is now at Technion, Israel Institute of Technology, Haifa, Israel. amirbelder@campus.technion.ac.il

[2]Gonçalo Dias Pais was an intern at Meta Reality Labs and is now with Sensei, Lisbon, Portugal. gpais@sensei.tech

[3] These authors are with Meta Reality Labs

[4] These authors are with Technion, Israel Institute of Technology, Haifa, Israel



(a) Synthetic (b) Real

Fig. 1: **Virtual placement.** Examples from our benchmark combining all three placement tasks in a single visualization: (a) a synthetic scene and (b) a real one. The tasks include (1) 2D Panel Placement, (2) Sitting Suggestion, and (3) TV Placement. Both are annotated with binary placement maps (white = placeable, black = non-placeable), and a VLM generates per-frame text descriptions including object characteristics and positions.

multiple object types. The second task, Sitting Suggestion, requires considering comfort and accessibility, such as ensuring adequate legroom and avoiding overcrowded areas. The third task, TV Placement, demands both global and local reasoning, as well as human-centric constraints, like selecting a suitable height, locating available wall space, and ensuring that the line of sight from the viewpoint is unobstructed.

In this work, we created ground-truth annotations for both real and synthetic indoor scenes that have previously been used in a variety of applications [15], [16]. We used egocentric datasets to ensure that this robotic task can be trained and evaluated from a first-person perspective that reflects natural human viewpoints. In total, we annotated 250 scenes, resulting in over $500,000$ individual annotations. For the synthetic dataset, we selected ASE [15] due to its high-quality, visually convincing egocentric scenes, and annotated a subset of 225 scenes. For the real dataset, we used AEO [16] for its diverse scenes and annotated all 25 of its scenes. Figure 1 shows two examples where, given an RGB image, we annotate the VP tasks, producing a binary placement map. A person typically sits on a couch or bed (Sitting Suggestion), with the TV placed in front of it at a comfortable height (TV Placement), while the 2D Panel is best placed on a nightstand or desk (2D Panel Placement). We also provide text descriptions of the objects within each image, along with their respective positions, to enable the use of text-based methods for VP.

Our annotations include binary masks and text descriptions in both 2D and 3D for each scene. First, we perform 2D placement annotations separately for each VP task. Since the VP problem is strongly influenced by human experience, the annotations must reflect human preferences. For example, if a robot were to place a laptop on the floor, someone could step on it, whereas a person would naturally place it on a table. To ensure such preferences are captured, we generated the ground truth using a combination of manual tagging and automatic tagging with a VLM (LLaVA [17]), which was prompted to reason about human preferences. Next, we used the available 2D bounding boxes of objects in each frame (provided in all scenes) to generate text descriptions. This was done by cropping each object and passing it individually to LLaVA [17] to obtain its description. Finally, to produce 3D annotations and corresponding 3D text descriptions, we reprojected the 2D points into 3D using the available pose and depth information of each frame.

We found that well-known detection and segmentation models struggle with VP tasks (e.g., IoU ≤ 0.46 on Sitting Suggestion) when trained on our benchmark [18], [19], [20], [21], [14], [22], [23], [24], [25]. These methods are less effective at capturing human preferences, particularly when occlusions occur in a single frame, due to their lack of global scene understanding.

Hence, our work makes the following contributions:

1) We introduce Placement Aria, the first VP benchmark that addresses the placement of both small and large, natural and man-made objects. Our annotations include 2D images and 3D point clouds, with per-object text descriptions and bounding boxes for three distinct VP tasks.
2) We provide manual annotations for all three tasks and propose an automatic tagging method, leveraging a VLM, that captures human preferences and improves the scalability of the dataset.
3) We evaluate several models on our benchmark to establish baselines and highlight the challenges of VP.

II. Related Work

Finding a valid target placement within a scene that aligns with human preferences is non-trivial. It is required for both visual place recognition and object placement tasks. Visual place recognition refers to the ability of systems to recognize previously visited locations based on visual input; see [26] for a comprehensive survey. One of its most common applications is assistive navigation for people with disabilities [1]. Our work extends egocentric data toward human-centric sitting suggestion, as well as assistive placement of common household objects.

Object placement has been extensively studied in both robotics and computer vision. In robotics, it typically involves learning spatial relations by relying on object geometries to enable robots to perform tasks such as object placing and manipulation [27], [28], [29], [30], [31]. For example, [27] places objects into specific targets (e.g., a table), while [28], [31], [29] arrange objects on a surface without interfering with one another. Similarly, [30] explores

placements that do not obstruct people within a room. In computer vision, object placement generally refers to inserting a real object from a source image into a target image [32], [33], [34], [5], [9], [6], [35]. Prior works [5], [6] focus on determining a *single* appropriate location and scale for the source object, while others emphasize rendering it realistically in the target image [8], [34]; see [33] for a comprehensive survey. In contrast, our VP dataset seeks to identify all placement regions within a scene, either for placing objects or for sitting, making it a more semantic and human-centric task.

Some works [36], [37] proposed real-world image datasets for virtual placement. However, their sizes (100 and 308 images, respectively) make them impractical for learning at scale. In contrast, our benchmark provides over 500,000 annotations. In [14], Ramrakhya et al. introduced a virtual placement dataset covering nine small man-made objects. Text descriptions were used to identify images containing these objects, and SAM [19] was applied to localize them within the images. Inpainting was then employed to remove the target objects, creating training images. Overall, their dataset consists of 10 indoor scenes and 1.3M images, which were augmented from a subset of $49,000$ images drawn from the LAION [38] and HSSD [39] datasets. However, the annotations in [14] do not explicitly account for human preferences, and thus may fail to capture all plausible placement locations. For example, if a table lamp only appears on nightstands in the source images, a table would never be considered a valid placement. This limitation highlights the importance of our tagging strategy, which combines human expertise with VLM-guided reasoning about human preferences. **Assistant Placement Aria** provides three semantically distinct VP tasks (2D Panel Placement, Sitting Suggestion, and TV Placement) and encodes human preferences through a combination of manual annotation and VLM-guided tagging. This ensures that plausible placements are captured with human-centric reasoning about comfort, accessibility, and context.

In this work, we focus on egocentric indoor scene datasets, enabling a more complete understanding of human preferences. These datasets can be broadly divided into real and synthetic, and have been used to examine tasks such as segmentation [40], object detection [41], and classification [42] of rooms. Real indoor datasets such as SUN-RGB-D [43], ScanNet [44], and Matterport3D [45] provide reconstructions of large-scale indoor scenes, typically recorded using RGB-D cameras. More recent datasets, including Aria-Twin [46], and AEO [16], offer higher reconstruction quality compared to earlier efforts. In particular, AEO provides 25 inherently different room scenes, making it both large and diverse, which are the reasons that motivated our decision to annotate it. Synthetic scene datasets, on the other hand, enable high-quality reconstructions [46]. Some, such as HyperSim [47] and OpenRooms [48], leverage online 3D models but lack real-world counterpart recordings, leaving a gap between simulated and real data. We chose Aria Synthetic Environments (ASE) [15], which stands out by offering both high-quality and visually convincing egocentric scenes.

III. The Assistant Placement Aria benchmark

VP entails identifying all potential locations for placing an object within a scene. This task requires human-level semantic understanding of the environment and is particularly relevant for object placement and human assistance. In contrast to traditional object placement and assistive navigation, which focus on a single (and usually pre-determined) destination, VP focuses on mapping out all feasible placements within the scene that align with human preferences. This enables the development of systems that can automatically select the most appropriate destination.

VP datasets are scarce, and existing works do not explicitly account for human preferences, which are essential to the task. Current datasets do not address the placement of large man-made objects or human-aiding tasks, both of which require considering global scene constraints, local physical constraints, and human preferences, e.g., placing a TV on a wall in front of a sofa for comfortable viewing.

To address this scarcity, we annotate a novel VP benchmark consisting of both synthetic and real data from egocentric RGB-D cameras to reflect human perception: (1) the Aria Synthetic Environments (ASE) dataset [15], from which we selected a subset of 225 high-quality, realistic synthetic indoor scenes; and (2) the Aria Everyday Objects (AEO) dataset [16], which contains 25 real-world scenes with a high degree of variation. All scenes provide sequences of RGB images and 6DoF camera poses; ASE additionally includes depth images and instance segmentation maps. A 3D point cloud is available for each scene and can be further densified by reprojecting individual frames using their associated depth and pose data. We augment both datasets in two ways: (1) by creating 2D and 3D VP annotations for each task, and (2) by generating textual descriptions of the objects in each frame, along with a global scene-level description summarizing all objects in the 3D scene, to support and enhance the annotation process.

A. Tasks and Labeling

Our annotation includes three distinct placement tasks: **(1) 2D Panel Placement**: Identifying suitable locations for placing a 2D panel representing small man-made objects, which primarily involves detecting vacant surfaces. This task is particularly relevant to object placing and manipulation, as it enables determining plausible target surfaces in a given scene. To generalize across object types, we use the abstract concept of a 2D horizontal panel (perpendicular to the z-axis) to represent objects ranging in size from 8 cm in diameter (e.g., a cup) to 40 cm (e.g., a large laptop). **(2) Sitting Suggestion**: Finding feasible sitting areas for a person, which is important for aiding people with disabilities. This task particularly involves various human-centric constraints such as adequate space for different body parts, along with considerations of comfort and ergonomics. **(3) TV Placement**: Determining appropriate positions for a TV screen, which is relevant to human-assistance tasks. This

(a) RGB input (b) Panel ground truth (c) Sitting ground truth (d) TV ground truth

Fig. 2: **Ground truth example.** Each frame was annotated separately for each task. (a) The original RGB image. (b) The only plausible place to place a 2D-Panel is on the desk at the bottom left corner of the image. (c) A person would sit on the couch. (d) The TV should be placed high enough on the wall and in front of the couch to enable comfortable watching.

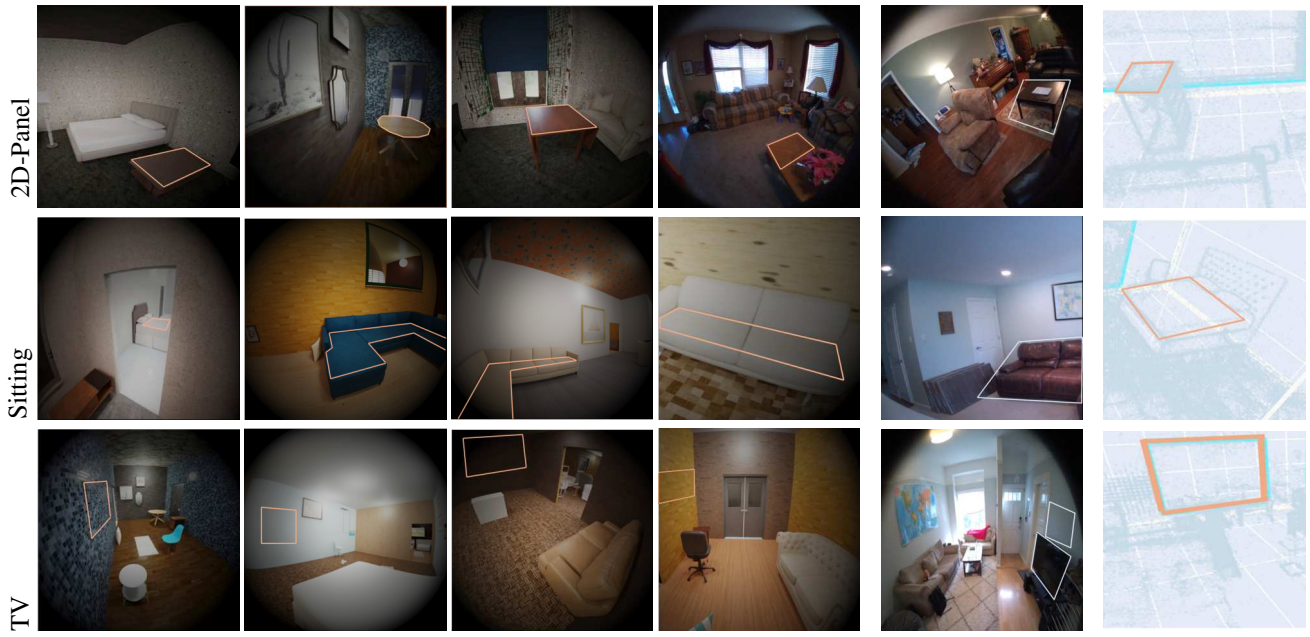
task requires accounting for both global and local factors, including optimal viewing height, unobstructed wall space, and ergonomic viewing angles. For example, the TV should not be mounted in the top or bottom 20% of the wall height to enable comfortable viewing, and should support sizes ranging from 43 to 65 inches (typical TV sizes). These tasks span a wide range of object sizes and spatial reasoning challenges, each reflecting different complexities of VP.

As the object placement task is strongly influenced by human experience, we manually annotated 50 scenes. However, manual labeling limits scalability to new tasks. To address this, we also introduce an automatic labeling approach, which we applied to an additional 200 scenes. All annotations are conducted separately for each task in 2D using a binary classification scheme: placeable regions are labeled as *1*, and non-placeable regions as *0*. Figure 2 shows a typical example of our annotations across the three placement tasks. Both manual and automatic 2D annotations are then reprojected into 3D to generate a placement point cloud for each scene using the available camera poses. The 3D annotation follows the same binary scheme based on their suitability for the given task, i.e., points in placeable regions are labeled as *1*, and non-placeable regions as *0*.

Manual labeling. The annotations were performed by a team of three expert human taggers. Each task was defined as clearly and simply as possible to minimize bias during tagging and was subject to the constraints outlined above. For example: *"Where would one place a TV screen whose size is between* 43 *and* 65 *inches?"*. Figure 3(a) shows several 2D examples from each task, while Figure 3(c) illustrates several 3D reprojections. In cases of disagreement over a specific area or pixel, the majority vote was taken. For instance, one such disagreement was whether a single chair should be considered a comfortable place to view a TV from (the majority voted *yes*). It is worth noting that disagreements were rare, occurring in fewer than 2% of the annotations.

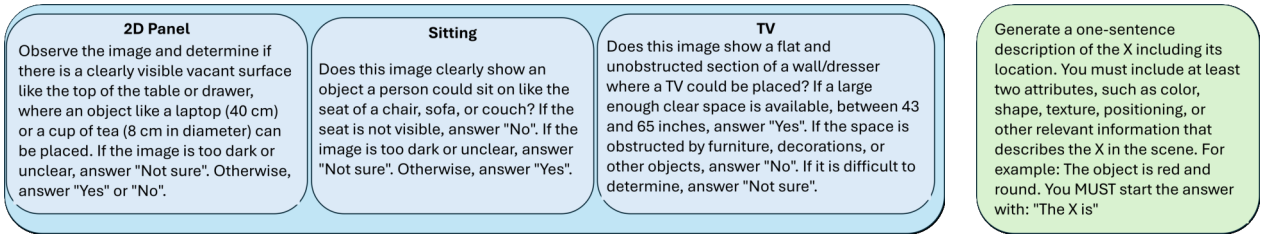
Automatic labeling As explained above, VP requires reasoning about human preferences. To support this, we use LLaVA [17] for automatic annotation, leveraging its reasoning capabilities, which are known to align well with human judgment. To guide LLaVA, it was prompted with 10 detailed examples for each VP task, helping it to better capture human constraints. We annotated 200 scenes from ASE [15], where each scene consists of multiple frames, and each frame was annotated independently. ASE provides a segmentation map for every frame, with each object represented by a unique mask. For each object, we crop the corresponding region from the frame and input it into LLaVA. Then, LLaVA's labeling process (of all three tasks) begins with the following instruction: *"Please answer the following questions while considering what a person would answer"*, to account for human preferences. This instruction is followed by a task-specific question about the object (e.g., *"Could a TV screen between* 43 *and* 65 *inches be placed here?"*). If LLaVA answers *"yes"*, we label the corresponding object mask as placeable; otherwise, it is labeled as non-placeable. The full formulations of all three VP tasks are shown in Figure 4(a). Furthermore, we utilized the 6DoF information of each scene to verify that the TV screens were not placed in the top or bottom 20% of each room.

Although useful for scaling the labeling process, the automatic procedure is less accurate than manual annotation. Automatic labeling tends to be more general in its selection of objects, rather than focusing solely on valid placement regions as a human annotator would. For example, when placing a 2D panel (e.g., a laptop), a human annotator would typically label only the tabletop, whereas the automatic annotation might also include the table's legs, since they fall within the object mask (see Figure 3(b)). To assess the accuracy of the automatic annotations, we compared them with the manual annotations on ASE's 25 manually tagged scenes using the Intersection-over-Union (IoU) between corresponding masks. TV Placement showed the highest agreement, with a mean IoU of 0.70. The 2D Panel task followed with a mean IoU of 0.64, while Sitting Suggestion was the least accurate, with a mean IoU of 0.60, as entire furniture items were often labeled instead of just their seats. Moreover, our human taggers manually verified that LLaVA tagged only reasonable objects for each task (e.g., chairs and couches for Sitting Suggestion), which were removed only in very rare cases (fewer than 3%) across all scenes. These failure cases typically involved partially visible objects that were not clearly captured within the frame, such as the desk in Figure 2(a). While the automatic tagging is



(a) Manual (b) Auto. (c) PC

Fig. 3: **Annotation examples.** Each task is shown with 4 manually annotated frames (a), followed by one automatically generated annotation (b), and one 3D point cloud annotation (c) reprojected from the annotated frames. The manual annotations highlight the high quality and precision of the human tagging. The automatic tagging is a bit less precise, as it covers entire objects.



(a) Assistance placement questions (b) Text description

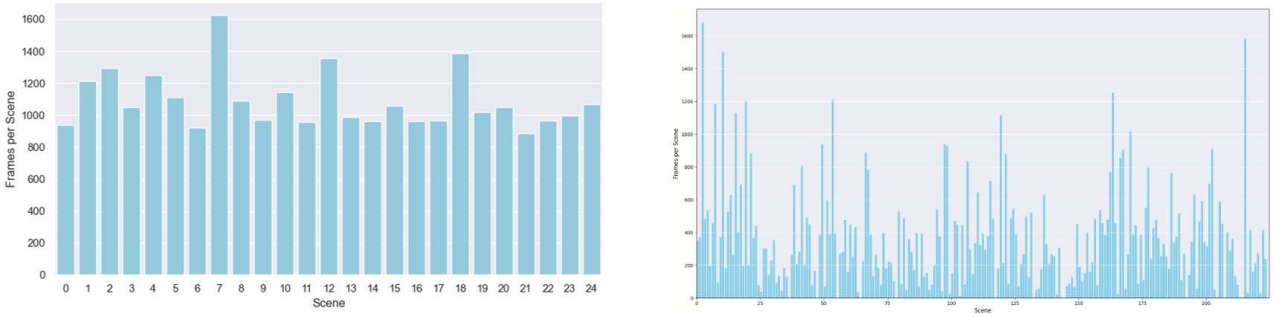
Fig. 4: **Example VP task prompts and scene description.** The questions posed to LLaVA for the VP annotations and the text description of each frame.

less accurate than manual annotation, it is worth noting that many object-placement systems only need to be directed to the target destination (e.g., Sitting Suggestion, which showed lower IoU) and can refine the available spots on a surface as part of their task definition. Thus, coarse annotations may still provide sufficient guidance for practical human-assistance tasks.

Text descriptions. Since textual descriptions are valuable for a range of scene understanding tasks, we also employ LLaVA to generate descriptions for each object in every frame. Leveraging these descriptions may improve VP performance in text-aware systems. For each object in a frame, we crop the image around the object as described above. The cropped region is then passed to LLaVA, which is prompted to generate a description containing at least two attributes, such as color, shape, texture, or other relevant characteristics (see Figure 1). The full prompt used to create these descriptions is shown in Figure 4(b), where `X` denotes the object in question. This process produces a caption for each frame, consisting of the 2D bounding boxes (bboxes) and the descriptions of all objects within it. In addition, we generate a global caption for the entire scene, which includes descriptions of all objects together with their corresponding 3D bboxes. Further details can be found in the appendix (Section VII).

B. Dataset Statistics and Benchmark

As each of our scenes is unique, they vary in terms of eligible placement area, number of frames, and number of points. Figure 5 shows the distribution of frames per scene, divided into real (a) and synthetic (b). For the synthetic data [15], the number of frames ranges between 93 and 1766 with 670 frames per scene on average. The number of points ranges between 85 and 2637 (thousand) with an average of 677 thousand points per scene. For the real data [16], the



(a) Real (b) Synthetic

Fig. 5: **Number of frames per scene.** The number of frames within each scene.

number of frames ranges between 920 and 1625 with 1100 frames per scene on average. The number of points ranges between 289 and 1657 thousand with an average of 647 thousand points per scene.

The percentage of eligible placement areas varies across tasks and data types. For 2D Panel Placement, the average percentage of pixels eligible for placement was 1.8% in the synthetic data and 2% in the real data. For Sitting Suggestion, 2.7% of the synthetic data and 3% of the real data were eligible. For TV Placement, 4.1% of the synthetic data and only 1% of the real data were eligible. This difference is not surprising, as the real data contains more images of walls, leaving less vacant space for televisions. Overall, the average number of annotations per scene is 1905, with each frame containing an average of 2.8 annotations.

Our benchmark is divided into training, validation, and test sets: **Training set**: 200 scenes, consisting of 15 real (scenes 10-24), 15 manually labeled synthetic (scenes 5-9), and 170 automatically labeled synthetic (scenes 40-209); **Validation set**: 25 scenes, consisting of 5 real (scenes 5-9), 5 manually labeled synthetic (scenes 10-24), and 15 automatically labeled synthetic (scenes 25-39); and **Test set**: 25 scenes, consisting of 5 real (scenes 0-4), 5 manually labeled synthetic (scenes 0-4), and 15 automatically labeled synthetic (scenes 210-224).

IV. Experiments

We evaluate our dataset using IoU to compare the predicted placements with the ground-truth annotations. As baselines, we assess a range of 2D and 3D methods, which are used as frozen backbones and trained for the VP task. For 2D, we evaluate the robot base placement method AIS [18] and, following the NYC-Indoor-VPR dataset [1], the detection method AnyLoc-VLAD DINOv2 [20]. We further benchmark two leading segmentation methods: SAM [19] and SigLIP-2 [21]. In addition, we report results for CLIP-UNet [14], which, to the best of our knowledge, is the only existing VP method that can be directly applied to our benchmark. For fairness, CLIP-UNet was trained and evaluated on our dataset. For 3D, we follow the detection methods used for evaluation in LiDAR-CS [49], including PointPillars [23], which transforms points into vertical pillars to form a 2D feature map; SECOND [22], which introduces sparse 3D convolutions to replace traditional dense 3D convolutions; and PointRCNN [25], a point-based framework that uses PointNet++ [50] as its backbone to extract features for segmentation. We also include LPS-Net [24], a cloud-based place recognition system. On top of each backbone (for both 2D and 3D), we train a placement head composed of a fully-connected layer to predict the binary VP maps. All reported results are based on the full benchmark (real + synthetic). Each result represents the average of three independent runs, with a standard deviation of less than 1%.

Table I shows the VP results of the different methods on the Assistant Placement Aria benchmark. In 2D, CLIP-UNet [14] achieves the best results. This is not surprising, as virtual placement requires both segmentation and localization (detection) capabilities, which CLIP-UNet combines, while other methods such as SAM [19] and AnyLoc-VLAD DINOv2 [20] have but one. In 3D, LPS-Net [24] attains the highest IoU, probably because it is a segmentation system, while the other methods are detection systems.

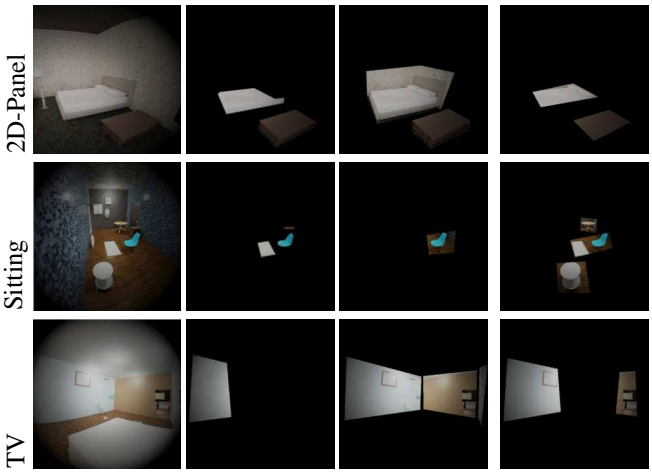
The performance on the full dataset is significantly higher than on the subset of manually tagged scenes across all methods. This is because the manually tagged scenes focus on the specific parts of each object that are placeable (e.g., the seat of a couch), rather than the entire object. As a result, detection and segmentation methods, which treat objects as wholes, perform significantly better on the automatically tagged scenes. Figure 6 illustrates this effect by presenting qualitative VP results overlaid on the input images, showcasing predictions of both segmentation and detection methods. While AnyLoc-VLAD DINOv2 and SAM treat each object as a single entity, as evident in the 2D Panel task, where an entire table is predicted as a valid region rather than just the tabletop, CLIP-UNet identifies more specific placement areas. This highlights the importance of fine-grained, preference-aware annotations for accurately evaluating VP methods.

V. Conclusions

In this paper, we addressed the challenge of virtual placement, a task that has remained underexplored due to the lack of suitable training data, and which has potential applications across various robotic fields, including assistive navigation and object placement. We introduced annotations for the

TABLE I: **Virtual Placement results.** VP results evaluated using IoU on the Assistant Placement Aria benchmark.

Method	Input	Manually tagged scenes			All scenes		
		2D-Panel	Sitting	TV	2D-Panel	Sitting	TV
AIS [18]	Images	25.7%	29.5%	24.1%	65.7%	59.5%	54.1%
SAM [19]		35.1%	40.2%	20.7%	70.7%	59.5%	50.1%
AnyLoc-VLAD DINOv2 [20]		22.4%	24.9%	21.4%	59.4%	57.1%	54.0%
SigLIP-2 [21]		38.2%	39.1%	25.6%	68.8%	60.1%	55.9%
CLIP-UNet [14]		**42.7%**	**45.7%**	**30.7%**	**75.7%**	**62.5%**	**60.3%**
SECOND [22]	Point clouds	15.7%	23.4%	25.7%	35.7%	30.4%	30.2%
PointPillars [23]		14.8%	23.1%	27.2%	24.3%	29.2%	32.1%
LPS-Net [24]		**17.9%**	**27.7%**	**29.8%**	**39.7%**	**34.6%**	**39.2%**
PointRCNN [25]		17.2%	25.7%	28.7%	38.6%	32.4%	32.8%



(a) RGB (d) SAM (f) AnyLoc (g) CLIP-UNet

Fig. 6: **Qualitative results.** Three examples of the placement performed by different 2D models.

ASE [15] and AEO [16] datasets, covering three distinct VP tasks: 2D Panel Placement, Sitting Suggestion, and TV Placement. Our annotation process includes both 2D and 3D labeling, along with per-object text descriptions. To the best of our knowledge, this is the first benchmark to comprehensively address the diverse challenges and constraints inherent to virtual placement, including global, local, and human-centric considerations. We also evaluated several detection and segmentation models on our benchmark, establishing baselines for future research. Future work may explore the development of new methods for both 2D and 3D virtual placement, leveraging the benchmark to advance human-centric scene understanding.

VI. Acknowledgments

This work was supported by the Israel Science Foundation (ISF) under grant 2329/22. We thank Shahar Vilc for her help with the manual tagging.

VII. Appendix

On the prompting process for the scene's textual description. To generate the description of each object, we first identify a representative image patch that can serve as input for text generation. During preprocessing, we traverse the entire scene and record the frame in which each object instance occupies the largest image area. For that frame, we then extract a patch by cropping the instance segmentation with an additional 10-px padding. We found that this approach obtains acceptable object descriptions in the scene for our placement tasks. This description is then used for all other frames in which the object appears.

During preprocessing of ASE [15], we also generated 3D bounding boxes for each object. For each frame, we reprojected the ground-truth depth, pose, and camera intrinsics into a point cloud, preserving instance segmentation labels. Each frame's point cloud was downsampled to 500,000 points using QuickFPS [51] for efficiency. The global point cloud was then aligned with the z-axis via floor segmentation. For each object, we computed the x and y rotation and scale from the convex hull of its projected points, while the bounding box height was obtained directly from the z-extent. We stored the 3D bounding box corresponding to the highest-area frame used for text description generation. These bounding boxes can support applications beyond object placement.

References

[1] D. Sheng, A. Yang, J.-R. Rizzo, and C. Feng, "Nyc-indoor-vpr: A long-term indoor visual place recognition dataset with semi-automatic annotation," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2024, pp. 14 853–14 859.

[2] T. Birr, C. Pohl, and T. Asfour, "Oriented surface reachability maps for robot placement," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2022, pp. 3357–3363.

[3] A. Wachter, A. Kugi, and C. Hartl-Nesic, "Time-optimal tcp and robot base placement for pick-and-place tasks in highly constrained environments," in *IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, 2024, pp. 2251–2257.

[4] T. Yang, J. V. Miro, Y. Wang, and R. Xiong, "Optimal object placement for minimum discontinuity non-revisiting coverage task," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2021, pp. 8422–8428.

[5] S. Zhou, L. Liu, L. Niu, and L. Zhang, "Learning object placement via dual-path graph completion," in *European Conf. Computer Vision (ECCV)*. Springer, 2022, pp. 373–389.

[6] Q. Meng and Q. Liu, "Interactive object placement with reinforcement learning," 2023.

[7] R. Parihar, H. Gupta, S. VS, and R. V. Babu, "Text2place: Affordance-aware text guided human placement," in *European Conf. Computer Vision (ECCV)*. Springer, 2024, pp. 57–77.

[8] S. Zhu, Z. Lin, S. Cohen, J. Kuen, Z. Zhang, and C. Chen, "Topnet: Transformer-based object placement network for image compositing," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 1838–1847.

[9] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Context-aware synthesis and placement of object instances," 2018.

[10] D. Z. Chen, A. X. Chang, and M. Nießner, "Scanrefer: 3d object localization in rgb-d scans using natural language," *16th European Conference on Computer Vision (ECCV)*, 2020.

[11] S. Peng, K. Genova, C. ". Jiang, A. Tagliasacchi, M. Pollefeys, and T. Funkhouser, "Openscene: 3d scene understanding with open vocabularies," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 815–824.

[12] R. Ding, J. Yang, C. Xue, W. Zhang, S. Bai, and X. Qi, "Pla: Language-driven open-vocabulary 3d scene understanding," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2023, pp. 7010–7019.

[13] D. Azuma, T. Miyanishi, S. Kurita, and M. Kawanabe, "Scanqa: 3d question answering for spatial scene understanding," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 19 129–19 139.

[14] R. Ramrakhya, A. Kembhavi, D. Batra, Z. Kira, K.-H. Zeng, and L. Weihs, "Seeing the unseen: Visual common sense for semantic placement," 2024.

[15] J. Engel, K. Somasundaram, M. Goesele, A. Sun, A. Gamino, A. Turner, A. Talattof, A. Yuan, B. Souti, B. Meredith, *et al.*, "Project aria: A new tool for egocentric multi-modal ai research," *arXiv preprint arXiv:2308.13561*, 2023.

[16] J. Straub, D. DeTone, T. Shen, N. Yang, C. Sweeney, and R. Newcombe, "Efm3d: A benchmark for measuring progress towards 3d egocentric foundation models," 2024.

[17] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[18] O. Mees, A. Emek, J. Vertens, and W. Burgard, "Learning object placements for relational instructions by hallucinating scene representations," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, Paris, France, 2020.

[19] 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," in *IEEE/CVF Int'l Conf. Computer Vision (ICCV)*, 2023, pp. 4015–4026.

[20] N. Keetha, A. Mishra, J. Karhade, K. M. Jatavallabhula, S. Scherer, M. Krishna, and S. Garg, "Anyloc: Towards universal visual place recognition," *IEEE Trans. Robot. Automat.*, vol. 9, no. 2, pp. 1286–1293, 2023.

[21] M. Tschannen, A. Gritsenko, X. Wang, M. F. Naeem, I. Alabdulmohsin, N. Parthasarathy, T. Evans, L. Beyer, Y. Xia, B. Mustafa, O. Hénaff, J. Harmsen, A. Steiner, and X. Zhai, "Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features," 2025.

[22] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[23] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*. Computer Vision Foundation / IEEE, June 2019, pp. 12 697–12 705.

[24] C. Liu, G. Chen, and R. Song, "Lps-net: Lightweight parameter-shared network for point cloud-based place recognition," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2024, pp. 448–454.

[25] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779.

[26] S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Trans. Robot. Automat.*, vol. 32, no. 1, pp. 1–19, 2016.

[27] Y. Jiang, M. Lim, C. Zheng, and A. Saxena, "Learning to place new objects in a scene," *The International Journal of Robotics Research (IJRR)*, vol. 31, no. 9, pp. 1021–1043, 2012.

[28] P. Jund, A. Eitel, N. Abdo, and W. Burgard, "Optimization beyond the convolution: Generalizing spatial relations with end-to-end metric learning," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[29] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *IEEE Int'l Conf. Computer Vision (ICCV)*, 2017, pp. 1301–1310.

[30] Y. Jiang, M. Lim, and A. Saxena, "Learning object arrangements in 3d scenes using human context," *arXiv preprint arXiv:1206.6462*, 2012.

[31] K. Zampogiannis, Y. Yang, C. Fermuller, and Y. Aloimonos, "Learning the spatial semantics of manipulation actions through preposition grounding," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1389–1396.

[32] L. Liu, B. Zhang, J. Li, L. Niu, Q. Liu, and L. Zhang, "OPA: object placement assessment dataset," *CoRR*, vol. abs/2107.01889, 2021.

[33] L. Niu, W. Cong, L. Liu, Y. Hong, B. Zhang, J. Liang, and L. Zhang, "Making images real again: A comprehensive survey on deep image composition," *arXiv preprint arXiv:2106.14490*, 2021.

[34] S. Lu, Y. Liu, and A. W.-K. Kong, "Tf-icon: Diffusion-based training-free cross-domain image composition," in *IEEE/CVF Int'l Conf. Computer Vision (ICCV)*, 2023, pp. 2294–2305.

[35] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Context-aware synthesis and placement of object instances," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018, pp. 10 393–10 403.

[36] L. Yoffe, A. Sharma, and T. Höllerer, "Octopus: Open-vocabulary content tracking and object placement using semantic understanding in mixed reality," 2023.

[37] T. Rafi, X. Zhang, and X. Wang, "Predart: Towards automatic oracle prediction of object placements in augmented reality testing," in *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, ser. ASE '22. New York, NY, USA: Association for Computing Machinery, 2023.

[38] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, *et al.*, "Laion-5b: An open large-scale dataset for training next generation image-text models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 25 278–25 294, 2022.

[39] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva, "Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 16 384–16 393.

[40] A. Belder and A. Tal, "Mme: Mixture of mesh experts with random walk transformer gating," 2026.

[41] A. Belder, R. Vivanti, and A. Tal, "A game of bundle adjustment-learning efficient convergence," in *IEEE/CVF Int'l Conf. Computer Vision (ICCV)*, 2023, pp. 8428–8437.

[42] A. Belder, G. Yefet, R. Ben-Itzhak, and A. Tal, "Random walks for adversarial meshes," in *ACM SIGGRAPH*. ACM, aug 2022.

[43] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[44] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2017.

[45] 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," *Int'l Conf. 3D Vision (3DV)*, 2017.

[46] X. Pan, N. Charron, Y. Yang, S. Peters, T. Whelan, C. Kong, O. Parkhi, R. Newcombe, and Y. C. Ren, "Aria digital twin: A new benchmark dataset for egocentric 3d machine perception," in *IEEE/CVF Int'l Conf. Computer Vision (ICCV)*, 2023, pp. 20 133–20 143.

[47] M. Roberts, J. Ramapuram, A. Ranjan, A. Kumar, M. A. Bautista, N. Paczan, R. Webb, and J. M. Susskind, "Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding," in *IEEE/CVF Int'l Conf. Computer Vision (ICCV)*, 2021, pp. 10 912–10 922.

[48] Z. Li, T.-W. Yu, S. Sang, S. Wang, M. Song, Y. Liu, Y.-Y. Yeh, R. Zhu, N. Gundavarapu, J. Shi, *et al.*, "Openrooms: An open framework for photorealistic indoor scene datasets," in *IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 7190–7199.

[49] J. Fang, D. Zhou, J. Zhao, C. Wu, C. Tang, C.-Z. Xu, and L. Zhang, "Lidar-cs dataset: Lidar point cloud dataset with cross-sensors for 3d object detection," in *IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2024.

[50] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 2017.

[51] M. Han, L. Wang, L. Xiao, H. Zhang, C. Zhang, X. Xu, and J. Zhu, "Quickfps: Architecture and algorithm co-design for farthest point sampling in large-scale point clouds," *IEEE Trans. Computer-Aided Design*, 2023.