

# MoMa-Kitchen: A 100K+ Benchmark for Affordance-Grounded Last-Mile Navigation in Mobile Manipulation

## Supplementary Material

This supplementary material extends our main study by providing additional details and data to improve the reproducibility of our MoMa-Kitchen method. It includes further evaluations and a range of qualitative results for NavAff, which reinforce the conclusions drawn in the primary paper. Additionally, we offer some affordance collection videos in the accompanying zip file.

▷ **Sec. 1:** Describes the hierarchical structure of the dataset, including scenes, configurations, and episodes, with detailed information on the generation process, target objects, and simulation settings.

▷ **Sec. 2:** Provides an in-depth explanation of the evaluation metrics used, training configurations, and the baseline models compared in our study.

▷ **Sec. 3:** Presents additional visualizations of predictions, further performance evaluations, ablation results regarding the weight of the MSE loss, some data collection videos, and real-world demo video.

▷ **Sec. 4:** Discusses the limitations of our work and explores prospects for future research.

## 1. Dataset

### 1.1. Dataset Composition and Splits

Our MoMa-Kitchen is hierarchically organized into three levels: scenes, configurations, and episodes. Here, we provide a detailed description of each level.

A scene consists of randomly generated base furniture and layout, where certain articulated objects in the base furniture (*e.g.*, microwaves, oven counters) serve as potential target objects for robotic arm manipulation. To ensure scene diversity, we randomly sample furniture categories, arrangement sequences, and specific instances within categories during scene generation. The statistics of target object assets employed in MoMa-Kitchen are summarized in Tab. 1.

Within each scene, we randomly place a varying number (1-3) of rigid objects, which, together with the articulated objects, constitute the set of target objects. To increase scene complexity, we position obstacles around these target objects. Each unique combination of target objects and obstacles forms a configuration of the scene.

To facilitate first-person view data collection, we sample 10 views for each configuration using a camera mounted on the robotic arm. Each view generates one episode, and the view selection follows two principles: (1) views are ran-
















Rigid-Cats	All	Bottle 	Pot 	Fruit 	Medicine Bottle 	Vegetable 
Rigid-Num	65	6	7	11	8	33
Articulated-Cats	All	Faucet 	Microwave 	Cabinet 	Dishwasher 	Oven Counter 
Articulated-Num	48	11	7	20	1	9
Obstacle-Cats	All	Chair 	Trolley 	Bin 	Table 	Cart 
Obstacle-Num	24	1	8	1	10	4

Table 1. **Statistics of target object assets employed in MoMa-Kitchen.** Distribution of rigid, articulated, and obstacle objects across different categories, showing the number of instances per category used in our dataset configurations.

domly initialized around the target, and (2) views must encompass both the target object and the surrounding floor area.

In total, our MoMa-Kitchen comprises 569 scenes, 14, 155 configurations, and 127, 343 episodes, representing a comprehensive collection of mobile manipulation scenarios.

### 1.2. Details on Simulation

We build our MoMa-Kitchen based on the BestMan [5] simulation environment, maintaining consistent simulation parameters across all scenes and interaction trials. The detailed configuration of our simulation setup is specified below:

- **RGBD rendering.** We render RGB images and depth maps using the BestMan interface. For comprehensive first-person view sampling, we position the camera at varying locations relative to the target object. Specifically, the camera is placed either to the left or right with a lateral offset ranging from 0.0 to 1.5 meters, while the forward distance is sampled between 1.5 and 3.8 meters. The camera orientation is consistently directed toward the target object to ensure optimal coverage of both the target and the surrounding floor area. These sampling ranges were empirically determined to maximize viewpoint diversity while maintaining scene relevance.
- **3D point cloud.** We back-project the depth image into a 3D point cloud using the camera’s intrinsic parameters. Subsequently, we filter out points with z-values below 0.02 meters to obtain the floor point cloud.
- **Target objects sampling.** In each scene, we randomly position 1-3 rigid objects in addition to the pre-existing articulated objects from scene generation. These objects

collectively form our set of target objects, all of which are placed on kitchen countertops. To increase scene complexity, we randomly place 1-3 obstacles within the semicircular region in front of each target object.

- **Interaction Trail.** To collect discrete navigation affordance values, we systematically sample robot positions within a semicircular region around the target object, with the radius set to the maximum reach of the robot arm. At each position, spaced at 10 cm intervals along both x and y axes, the robot attempts to either grasp (for parallel grippers) or suction (for vacuum grippers) the target object.

### 1.3. Additional Visualization Results

We present additional visualization examples from MoMa-Kitchen, including object assets, scene configurations, and affordance maps.

#### 1.3.1. Object Assets

We showcase a diverse set of object assets in Fig. 2. These assets serve as manipulation targets, environmental elements, or obstacles in our generated scenes.

#### 1.3.2. Scene Configurations

We illustrate a comprehensive set of scene configurations in Fig. 3. These examples highlight the complexity and diversity of our generated environments, reflecting real-world manipulation scenarios.

#### 1.3.3. Affordance Map Examples

We obtain sparse discrete navigation affordance values by systematically sampling robot positions and evaluating object interactions in the simulator. These sparse affordance values are then interpolated using a Gaussian-weighted k-nearest neighbor algorithm to generate continuous and dense navigation affordance maps. The comparison between sparse samples and interpolated dense maps is illustrated in Fig. 1.

## 2. Additional Implementation Details

### 2.1. Evaluation Metrics

In this section, we provide a detailed explanation of the evaluation metrics used in our study. To comprehensively evaluate our method, we adopt five metrics: Root Mean Squared Error (**RMSE**), Logarithmic Mean Squared Error (**logMSE**), Pearson Correlation Coefficient (**PCC**), Cosine Similarity (**SIM**), and Continuous Intersection over Union (**cIoU**). Below, we describe each metric, its calculation formula, and its relevance to our task.

- **RMSE:** RMSE measures the numerical alignment between predicted and ground truth values. It penalizes larger errors through squared differences, as defined be-

low:

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2}, \quad (1)$$

where  $y_i$  and  $\hat{y}_i$  are ground truth and predicted values, respectively, and  $N$  is the total number of elements. RMSE reintroduces the original scale of predictions, offering interpretability while highlighting large errors. In our study, RMSE assesses navigation affordance predictions by ensuring precise positioning and minimizing major errors in complex environments.

- **logMSE:** LogMSE evaluates the relative differences between predictions and ground truth, reducing the impact of large outliers. It is calculated as:

$$\text{logMSE} = \frac{1}{N} \sum_{i=1}^N (\log(1 + y_i) - \log(1 + \hat{y}_i))^2, \quad (2)$$

By focusing on proportional consistency, logMSE smooths out outliers and highlights relative accuracy. In this study, it evaluates the model's ability to capture balanced affordance patterns across both low and high value regions.

- **PCC:** PCC quantifies the linear relationship between predicted and ground truth patterns, independent of their magnitudes:

$$\text{PCC} = \frac{\sum_{i=1}^N (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^N (y_i - \bar{y})^2 \sum_{i=1}^N (\hat{y}_i - \bar{\hat{y}})^2}}, \quad (3)$$

where  $\bar{y}$  and  $\bar{\hat{y}}$  are the means of  $y$  and  $\hat{y}$ . PCC highlights pattern consistency, making it ideal for evaluating spatial distributions in affordance maps. A high PCC reflects accurate predictions of affordance trends, which is essential for precise navigation.

- **SIM:** SIM evaluates the alignment of relative spatial patterns between predictions and ground truth. It is calculated as:

$$\text{SIM} = \frac{1}{N} \sum_{i=1}^N \frac{y_i \cdot \hat{y}_i}{\|y_i\| \|\hat{y}_i\|} \quad (4)$$

SIM with higher values indicating better structural alignment. Unlike PCC, which evaluates overall pattern trends, SIM focuses on the overlap of affordance regions, making it particularly effective for assessing spatial alignment. In our study, SIM ensures that predicted affordance maps accurately capture the structural properties of ground truth.

### 2.2. Training Details

To ensure a fair comparison, we train our model and all baseline methods under consistent settings. The implementations for both our methods and the baselines are developed

using PyTorch. All models are trained on a single NVIDIA A100 GPU with a batch size of 64 for 6 epochs, completing the entire process in approximately 8 hours. We utilize the Adam optimizer [1] with betas configured as 0.9 and 0.999. The learning rate is initialized at  $8e-4$  and follows a cosine decay schedule.

### 2.3. Compared Baselines

Since our work is the first to propose a benchmark for navigation affordance grounding, there are no existing methods that directly address this task. Therefore, we adapt several classical methods commonly used for feature extraction and 3D object detection on point clouds for comparison. Specifically, we include PointNet++[3], a foundational model for point cloud feature extraction, VoteNet[4], a pioneering method for 3D object proposal generation, and H3DNet [6], which enhances object detection with hierarchical features. To ensure a fair and comprehensive comparison, we adapt the official implementations of these methods to our MoMa-Kitchen dataset. Specifically, we reimplement their architectures and fine-tune them for the navigation affordance grounding task, conducting training and evaluation under identical experimental settings. This allows us to systematically assess their performance against our proposed NavAff.

**PointNet++.** PointNet++ extends the original PointNet [2] framework by introducing hierarchical feature learning for point cloud processing. This method divides the input point cloud into overlapping regions using a sampling and grouping strategy, applying PointNet locally to extract features, and aggregating them hierarchically. PointNet++ is widely used for tasks such as segmentation and classification in 3D point clouds. In our adaptation, the point cloud data is passed through an encoder to extract features, which are then decoded to predict navigation affordance.

**VoteNet.** VoteNet introduces a deep Hough voting framework for 3D object detection in point clouds. The method employs a point-based network to generate votes for object centers, followed by an aggregation module that clusters votes to produce 3D object proposals. To adapt VoteNet for our benchmark, we removed the Vote Aggregation and Detection components originally used for bounding box regression, retaining the remaining modules to perform navigation affordance grounding. This adaptation ensures the network focuses on predicting affordance maps instead of object detection.

**H3DNet.** H3DNet proposes a Hierarchical 3D Detection Network that improves 3D object detection by leveraging multi-level geometric features. The network integrates instance-level and part-level features using a coarse-to-fine detection pipeline and introduces novel feature aggregation modules to enhance geometric reasoning. Similar to VoteNet, in our adaptation, we removed the bounding box

regression components while retaining the remaining modules to focus on navigation affordance grounding, enabling the network to predict affordance maps instead of object detection outcomes.

## 3. Additional Experimental Results

**Visualization of Predictions.** As shown in Fig. 4, we present the predicted navigation affordance results visualized within the dense global point cloud. Additionally, we provide a comparison with the ground truth affordance to highlight the model’s performance and alignment with the reference data.

**Detailed Evaluation.** Tab. 2 presents a comprehensive evaluation of various metrics within a single scene, offering an in-depth analysis of the model’s behavior and performance. Each scene comprises multiple episodes, each representing distinct configurations and challenges. By evaluating metrics across these episodes, we gain a finer understanding of the model’s ability to generalize under varying conditions.

**Impact of Weight Choices on Weight MSE Loss.** Fig. 5 illustrates that when the weight value is too small, the model experiences underfitting because the influence of the loss function weight is insufficient, preventing the model from effectively learning high-quality navigation affordance grounding. Conversely, when the weight value is too large, the class imbalance issue described earlier persists, which also limits the model’s performance. From the figure, it can be observed that when the weight value is set to 0.7, the Pearson Correlation Coefficient (PCC) reaches its peak. PCC measures the linear correlation between the predicted and ground truth values, effectively reflecting the model’s ability to capture the distribution patterns of navigation affordance. A high PCC value indicates a stronger correlation between the predicted trends and the ground truth distribution, which is particularly critical for navigation tasks in complex environments.

**Real world Experiment.** Please see the real-world captured video demo in the zip file.

**Affordance labeling visualization.** Please see the navigation affordance collection video in the zip file.

## 4. Discussion on Limitations and Future Work

While our work makes significant progress in addressing the “last mile” navigation challenge, we acknowledge several limitations and identify promising directions for future research:

- **Scene Diversity:** Although MoMa-Kitchen contains a large number of episodes, they are currently limited to kitchen environments. Future work should expand to other household scenarios such as living rooms, bedrooms, and bathrooms, which present different challenges

and spatial configurations.

- **Single-Task Focus:** The current approach focuses solely on reaching and grasping tasks. Future work should consider more complex manipulation sequences that require multiple positioning adjustments or different types of interactions (*e.g.*, pushing, pulling, or sliding objects).

#### Future Directions

- **Multi-Task Learning:** Future research could explore how navigation affordances vary across different manipulation tasks and develop models that can adapt their positioning strategies based on the intended manipulation action.
- **Online Adaptation:** Developing methods for online adjustment of affordance predictions based on real-time feedback during task execution could enhance robustness in dynamic environments.
- **Integration with LLMs:** While current LLM-based approaches have limitations, future work could explore hybrid approaches that combine our learned affordance models with LLM reasoning for more sophisticated task planning and execution.
- **Uncertainty Estimation:** Incorporating uncertainty estimation in affordance predictions could help robots make more informed decisions about positioning and potentially trigger replanning when necessary.

These limitations and future directions present exciting opportunities for extending our work and further advancing the field of mobile manipulation.

Table 2. **Evaluation results across individual scenes on MoMa-Kitchen.** Performance metrics evaluated separately for each scene in the dataset.

Scene ID	RMSE ↓	logMSE ↓	PCC ↑	SIM ↑
989172	0.283	0.0439	0.685	0.702
807952	0.269	0.0402	0.652	0.667
502334	0.284	0.0443	0.716	0.732
306938	0.282	0.0435	0.737	0.752
443958	0.232	0.0303	0.615	0.622
66171	0.297	0.0493	0.595	0.619
152285	0.242	0.0332	0.552	0.578
306168	0.223	0.0282	0.595	0.609
636942	0.283	0.0445	0.694	0.713
739657	0.264	0.0392	0.652	0.672
143853	0.291	0.0463	0.742	0.756
739202	0.236	0.0312	0.561	0.584
583009	0.302	0.0487	0.702	0.719
451797	0.301	0.0491	0.728	0.745
116280	0.166	0.0164	0.305	0.364
274269	0.274	0.0405	0.829	0.827
485779	0.298	0.0480	0.730	0.749
772552	0.299	0.0494	0.683	0.706
359363	0.213	0.0255	0.566	0.576
264325	0.292	0.0454	0.593	0.622
194561	0.276	0.0417	0.678	0.695
792629	0.292	0.0462	0.701	0.715
69567	0.273	0.0416	0.722	0.735
783647	0.288	0.0457	0.644	0.666
721930	0.219	0.0268	0.500	0.528
668061	0.266	0.0385	0.587	0.616
615543	0.275	0.0408	0.634	0.648
994972	0.285	0.0436	0.759	0.754
996121	0.266	0.0389	0.587	0.604
67534	0.278	0.0421	0.633	0.654
142672	0.270	0.0402	0.700	0.714
501160	0.276	0.0426	0.692	0.714
487375	0.227	0.0286	0.519	0.528
437964	0.294	0.0465	0.712	0.729
355986	0.299	0.0485	0.744	0.764
453020	0.285	0.0449	0.663	0.686
309033	0.296	0.0465	0.758	0.762
567413	0.225	0.0284	0.608	0.619
23304	0.248	0.0340	0.560	0.586
960190	0.247	0.0336	0.683	0.699
243997	0.288	0.0454	0.750	0.762
466622	0.265	0.0389	0.571	0.592
569661	0.278	0.0424	0.685	0.705
403556	0.195	0.0219	0.481	0.504
297024	0.289	0.0454	0.724	0.742
419493	0.219	0.0278	0.300	0.369
179882	0.213	0.0255	0.566	0.625
328786	0.256	0.0370	0.629	0.643
475545	0.186	0.0202	0.573	0.568
773991	0.278	0.0425	0.665	0.684

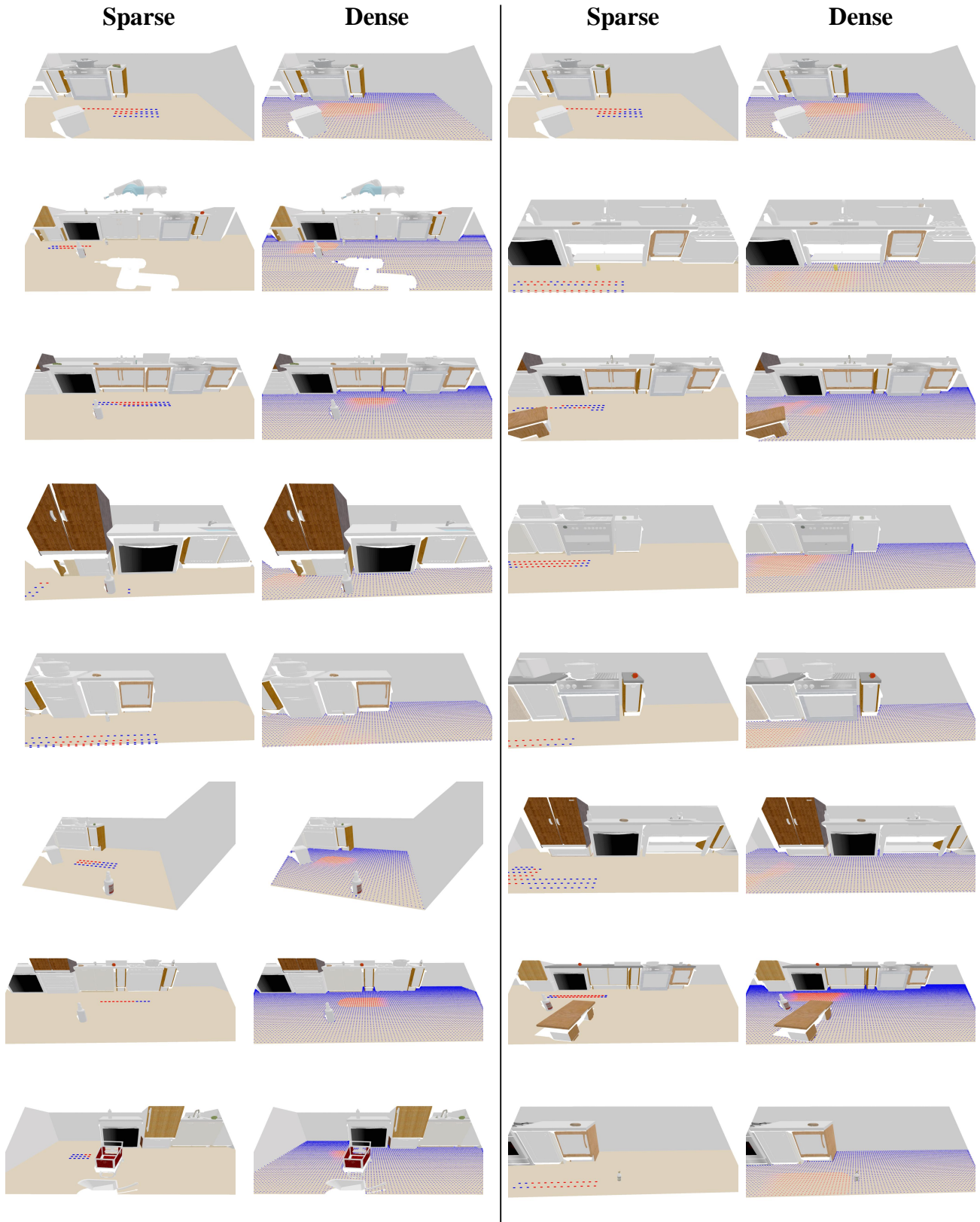


Figure 1. **Visualization of navigation affordance maps in MoMa-Kitchen.** Comparison between sparse affordance values collected through discrete robot-object interactions (left) and their corresponding dense maps generated via Gaussian-weighted k-nearest interpolation (right).





Figure 2. **Visualization of object assets in MoMa-Kitchen.** The collection includes diverse categories of objects commonly found in household environments, ranging from kitchenware and appliances to furniture and daily necessities.

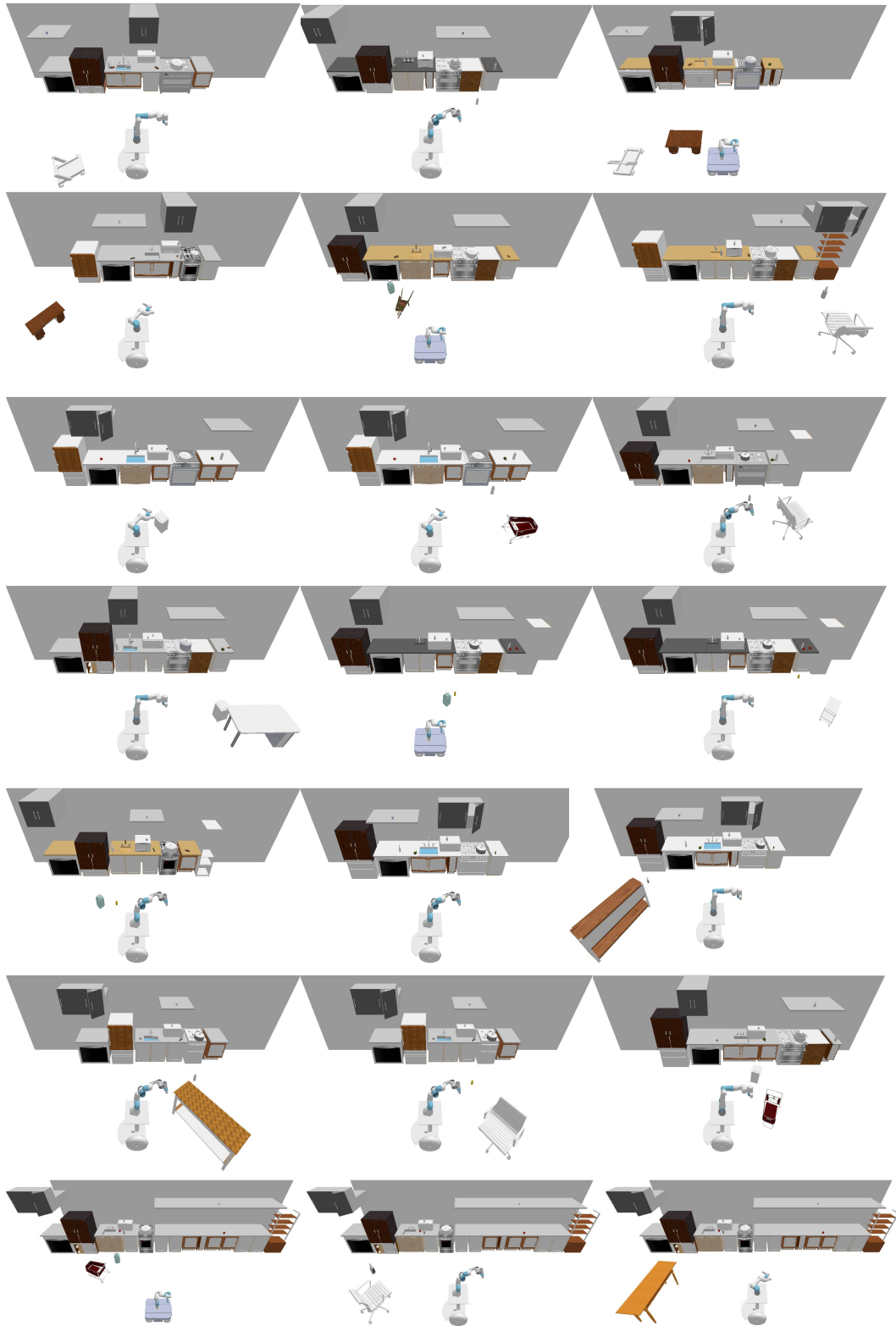


Figure 3. **Visualization of diverse scene configurations in MoMa-Kitchen.** Each example showcases different arrangements of base furniture, layouts, target objects, and obstacles, demonstrating the variety of manipulation scenarios.

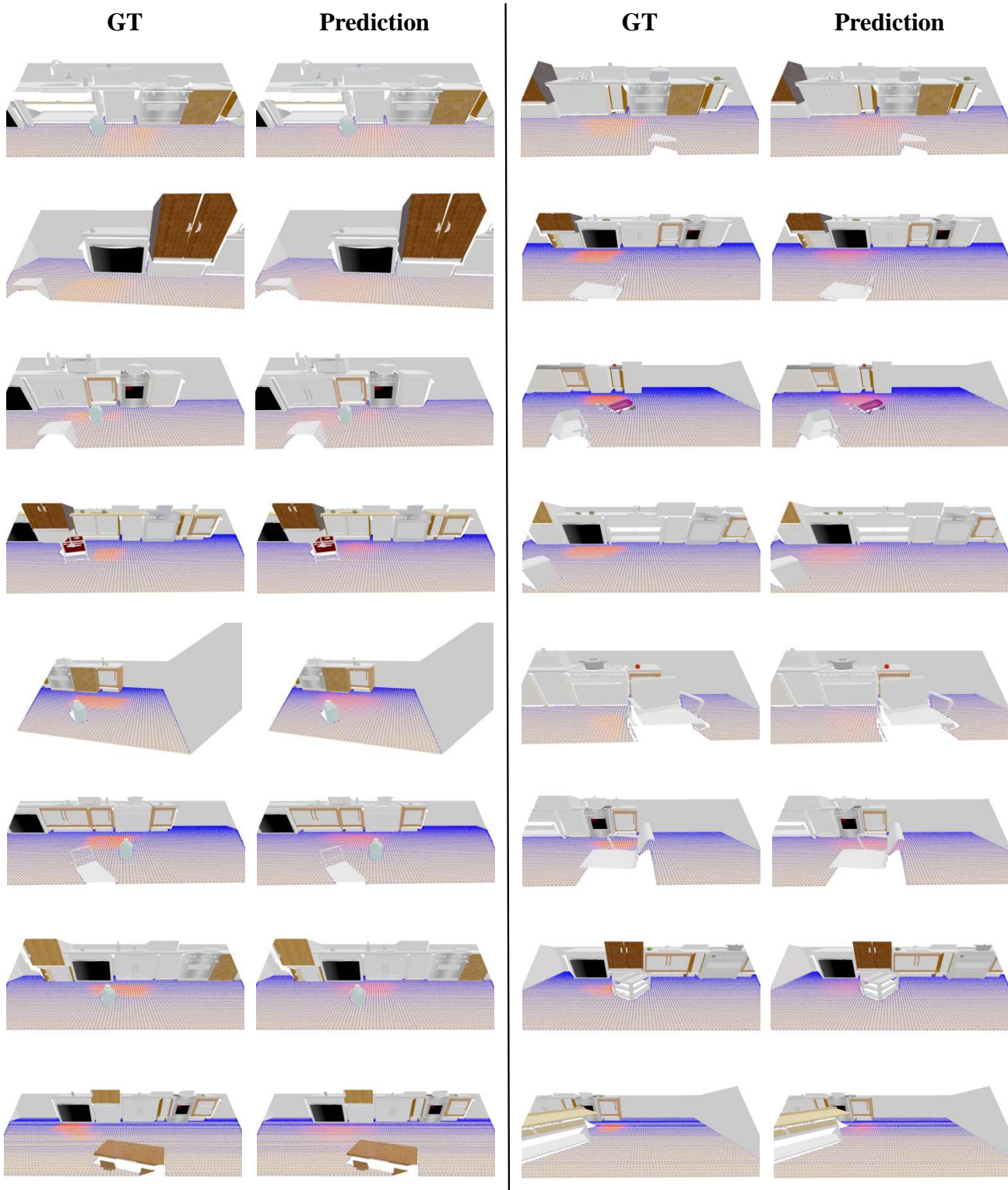


Figure 4. **Predicted vs. Ground Truth Navigation Affordance.** Comparison of the model’s predicted navigation affordance (right columns) and the ground truth affordance (left columns) visualized within dense global point clouds. The visualizations illustrate the spatial alignment and consistency of the predictions with the reference data across different scenes.





Figure 5. **Effect of Weight on MSE Loss and Evaluation Metrics.** Evaluation of the impact of different weight values in the Weighted MSE loss function on various metrics, including RMSE, LogMSE, PCC, SIM, and cIoU.

## References

- [1] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. [3](#)
- [2] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. [3](#)
- [3] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. [3](#)
- [4] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. [3](#)
- [5] Kui Yang, Nieqing Cao, Yan Ding, and Chao Chen. Bestman: A modular mobile manipulator platform for embodied ai with unified simulation-hardware apis. 2024. [1](#)
- [6] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qi-Xing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, 2020. [3](#)