📌 Personalized Instance-based Navigation Toward User-Specific Objects in Realistic Environments

Luca Barsellotti[*] **Roberto Bigazzi**[*]
Marcella Cornia **Lorenzo Baraldi** **Rita Cucchiara**
University of Modena and Reggio Emilia, Italy
`{firstname.lastname}@unimore.it`

Project page: aimagelab.github.io/pin

Abstract

In the last years, the research interest in visual navigation towards objects in indoor environments has grown significantly. This growth can be attributed to the recent availability of large navigation datasets in photo-realistic simulated environments, like Gibson and Matterport3D. However, the navigation tasks supported by these datasets are often restricted to the objects present in the environment at acquisition time. Also, they fail to account for the realistic scenario in which the target object is a user-specific instance that can be easily confused with similar objects and may be found in multiple locations within the environment. To address these limitations, we propose a new task denominated *Personalized Instance-based Navigation* (PIN), in which an embodied agent is tasked with locating and reaching a specific personal object by distinguishing it among multiple instances of the same category. The task is accompanied by 📌 *PInNED*, a dedicated new dataset composed of photo-realistic scenes augmented with additional 3D objects. In each episode, the target object is presented to the agent using two modalities: a set of visual reference images on a neutral background and manually annotated textual descriptions. Through comprehensive evaluations and analyses, we showcase the challenges of the PIN task as well as the performance and shortcomings of currently available methods designed for object-driven navigation, considering modular and end-to-end agents.

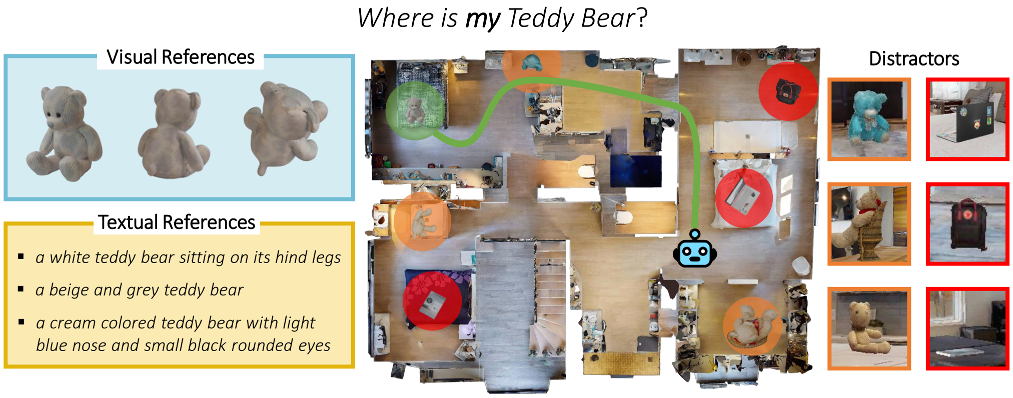

Where is **my** Teddy Bear?
Visual References
Textual References
a white teddy bear sitting on its hind legs
a beige and grey teddy bear
a cream colored teddy bear with light blue nose and small black rounded eyes
Distractors

Figure 1: We introduce the PIN task, where the agent is asked to navigate toward a personalized object instance using multimodal references and distinguish it from distractors (*i.e.*, other objects of the same category as the target or of other categories). The target object, same category distractors, and other distractors are circled, respectively, in green, orange, and red. The total number of available objects in the dataset is 338, corresponding to different instances of 18 object categories.

[*] Equal contribution.

38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

1 Introduction

Imagine a scenario where your child wants his favorite teddy bear, and he lost it somewhere in your house. In the foreseeable future, a "smart" domestic robot could be asked to find it. In that case, the robot will start roaming through the environment searching for the teddy bear. However, a-priori knowledge of the object category and visual cues related to the surroundings are not enough to solve the task, as the teddy bear has no predetermined location in the scene, could be potentially situated in several different places, and can be confused with other stuffed toys. While the recent advances in Embodied AI have significantly fostered the development of autonomous agents that can locate predefined target object categories, a benchmark that evaluates how agents tackle the challenges of reaching personal object instances in a photo-realistic environment is absent.

Motivation. The majority of current object-driven navigation tasks in Embodied AI define their goals as a general semantic category represented through text [2, 6, 70] (*e.g.*, "chair", "sofa") or as a specific target instance defined by an image or description including the surrounding context in which the object can be found [9, 19, 32, 36, 82]. Moreover, these datasets rely on objects which were present at the time of acquisition of the environment [8, 13, 20, 32, 36, 45, 61, 71, 74]. On the contrary, procedurally generated environments can freely contain additional objects and annotations [21, 23, 34, 41]. However, the appearance discrepancy between these environments and the real world or photo-realistic environments could affect the performance of the agents when deployed on robotic platforms [31]. Previous work has proposed loading additional 3D objects inside photo-realistic environments [46] to improve agent navigation performance, to allow object interaction in static environments [64], or to enable navigation towards multiple goals [70]. However, no previous work has targeted loading objects that can be moved frequently and can appear in multiple contexts since loaded 3D models are kept in their initial spawn position.

Overview of the dataset. To overcome these issues, we propose the novel task of *Personalized Instance-based Navigation* (PIN), where the agent needs to locate and reach a specific personalized target instance in the environment provided as reference images and textual descriptions, without information about the surrounding context. An overview of PIN is shown in Fig. 1. In parallel with the definition of the task, we release 📌 PInNED (Personalized Instance-based Navigation Embodied Dataset), a dedicated dataset of episodes for this setting that leverages the main advantages of both photo-realistic and procedurally generated embodied environments. In each episode, along with a unique target instance, distractors objects are placed in the scene to confound the navigation of the agent. Specifically, we built the dataset on top of the semantic annotations [74] and scenes of Habitat-Matterport3D Dataset (HM3D) [56] with the injection of additional photo-realistic 3D objects accurately selected from Objaverse-XL [22]. The objects are positioned in each environment through a procedural spawning method on predefined suitable surfaces. PInNED comprises 865.5k training episodes and 1.2k validation episodes built on top of 338 additional objects.

Finally, we adapt and test currently available navigation agents on the proposed dataset, showcasing the shortcomings of relevant approaches. In particular, we compare the performance of the two main categories of navigation agents for object-driven navigation, modular and end-to-end approaches, where we demonstrate that the versatility of modular methods leads to superior performance compared to the end-to-end counterparts; still, the task is far from being resolved. These experiments assess the difficulties posed by PIN task, highlighting the need for further research on the topic. More details and release information on the codebase for the task, accompanying dataset, and evaluation benchmark are included in the Appendix.

Contributions. To sum up, our key contributions are threefold:

- 📌 We introduce the task of Personalized Instance-based Navigation (PIN). In this task, an agent must find and navigate towards a specific object instance without using the surrounding context. To increase the difficulty and compel the agent to learn to identify the correct instance, object distractors belonging to the same or different categories of the target are also added.
- 📌 We build and release Personalized Instance-based Navigation Embodied Dataset (PInNED), a task-specific dataset for embodied navigation based on photo-realistic personalized objects from Objaverse-XL dataset injected in the environments of HM3D dataset. Overall, it comprises 338 object instances belonging to 18 different categories positioned within 145 training and 35 validation environments, for a total of approximately 866.7k navigation episodes.
- 📌 We evaluate currently available object-driven methods on the newly proposed dataset demonstrating their limitations in tackling the proposed PIN task.

2 Related Work

Object-based Embodied Datasets. In recent years, research aimed at the development of intelligent autonomous agents has acquired increasing interest with the release of simulation platforms like Habitat [53, 61, 66], AI2-THOR [34], RoboTHOR [21], and ProcTHOR [23], as well as datasets of scenes for robotic navigation like Gibson [64, 71], Matterport3D [13], and Habitat-Matterport3D (HM3D) [56]. The evaluation of the capabilities of such agents can be performed on multiple embodied tasks [3, 58, 65] mimicking different real-world requirements. PointGoal Navigation (PointNav) [2] requires the agent to reach specific relative coordinates to its starting position. In object-oriented navigation, the agent is tasked to find any instance of an object category (ObjectNav) [2, 6], multiple objects in sequence (MultiON) [70], or a specific instance of a category (ION) [41]. Other embodied navigation tasks are ImageGoal navigation (ImageNav) [19, 82] that requires the agent to reach the position where the goal image has been taken, and a more object-oriented formulation of ImageNav called Instance-Specific Image Goal Navigation (InstanceImageNav) [36] that requires to reach a precise object instance given a photo of it. Recently, the GOAT-Bench benchmark has been introduced, which requires finding sequences of target objects using multimodal references [32]. However, GOAT-Bench targets are constrained to the objects captured in the environment at acquisition time. To the best of our knowledge, PInNED is the only dataset focused on navigation toward personalized targets that uses multimodal references, injects additional objects into photorealistic environments, and requires the agent to distinguish the correct instance from distractors without relying on context.

Object-based Navigation Agents. Object-based methods for navigation agents can be divided into two categories depending on their design: modular approaches and end-to-end approaches. Modular approaches are composed of multiple components, usually a mapping module, an exploration procedure, and an object detection method. Some approaches adapted the architecture proposed by ANS [16] for object goal navigation by building semantic maps to locate the target [15, 39, 55, 81]. Following, Stubborn [44] proposed a strong baseline using a heuristic exploration method. Among end-to-end methods, Mousavian *et al.* [50] and Yang *et al.* [76] worked on improving visual representations, Mayo *et al.* [47] used spatial attention maps, and Ye *et al.* [77] used auxiliary tasks. Other related work leveraged object relation graphs [27, 28, 52]. THDA [46], instead, used 3D scans of objects from YCB dataset [11] to augment the training dataset. Recently, PIRLNav [57] used a two-stage learning strategy, Chen *et al.* [18] used a method based on recursive implicit maps, and OVRL [72, 73] exploited self-supervised visual pretraining to boost agent capabilities. Additionally, zero-shot object goal navigation has been recently explored by ZER [1], ZSON [45], and ORION [20].

Personalized Instance Recognition. In recent years, foundation models have revolutionized the Computer Vision field. CLIP [54] learned a multimodal embedding space by performing large-scale contrastive training, demonstrating impressive capabilities in zero-shot classification. DINO [12, 51] is trained with a self-supervised paradigm achieving strong semantic correspondence properties among features [4, 5, 79]. Segment Anything (SAM) [33] has been trained to predict precise class-agnostic masks given a prompt. The feature spaces learned by these models are semantically rich and can be exploited in tasks that involve the recognition of general object categories. However, adapting a model for recognizing personalized objects in images remains an open challenge. For example, SuperGlue [60] leveraged an attention-based graph neural network on the local descriptors extracted with the SuperPoint model [25] to perform image matching and has been used in Mod-IIN [35] and GOAT [14] to tackle the InstanceImageNav task. IEVE [40], instead, proposes an Exploration-Verification-Exploitation framework that combines a segmentation model and a keypoint matcher to recognize distant objects and confirm them when the agent is closer; while PerSAM [80], performed personalized segmentation allowing SAM to localize a user-provided target. In the same setting, SegIc [48] introduced a mask decoder with in-context instructions on top of the dense correspondences from DINOv2 [51], while Matcher [43] leveraged DINOv2 to extract prompts for SAM in a training-free paradigm.

3 Personalized Instance-based Navigation

In this section, we outline the Personalized Instance-based Navigation task, highlighting its key characteristics and comparing it to existing embodied tasks. Following, we detail the composition and generation process of the PInNED dataset.

Table 1: Comparison of the different object-driven datasets for embodied navigation, considering the photo-realism of scenes and targets, the availability of additional objects with variable spawn locations, the modalities of the provided references, and whether the dataset is instance-oriented.

Dataset	Photo-Realistic Scenes	Photo-Realistic Targets	Additional Objects	Visual Reference	Descriptive Reference	Variable Placement	Instance Goal
MP3D [13]	✓	✓	✗	✗	✗	✗	✗
AI2-THOR [34]	✗	✗	✓	✗	✗	✓	✗
Gibson [71]	✓	✓	✗	✗	✗	✗	✗
Robo-THOR [21]	✗	✗	✓	✗	✗	✓	✗
MultiON* [70]	✓	✗	✓	✗	✓	✓	✗
HM3D [56]	✓	✓	✗	✗	✗	✗	✗
ProcTHOR	✗	✗	✓	✗	✗	✓	✗
ION [41]	✗	✗	✓	✗	✓	✓	✓
THDA [46]	✓	✓	✓	✗	✗	✓	✗
ZSON [45]	✓	✓	✗	✗	✓	✗	✗
InstanceImageNav [35]	✓	✓	✗	✓	✗	✗	✓
ZIPON [20]	✓	✓	✗	✗	✓	✗	✓
GOAT-Bench [32]	✓	✓	✗	✓	✓	✗	✓
PInNED (Ours)	✓	✓	✓	✓	✓	✓	✓

3.1 Task Definition

The PIN task requires the agent to navigate toward a predetermined specific object instance (*e.g.*, "*a yellow backpack with red straps*") in an unexplored environment. Each target object needs to be found in the environment, distinguishing it from multiple distractors of the same category and other objects of different categories. In this setting, the target object can be provided to the agent in two different modalities: (i) as a set of RGB images depicting the target object rendered in an isolated context on a neutral background, and (ii) as a set of textual descriptions of the object instance appearance.

At the beginning of each episode of PIN, the agent is initialized at a random pose $\boldsymbol{x}_0$ in an unseen environment. A single target instance o^i is selected as the goal g, such that $g \in C^a \subset O$, where C^a is a set of instances belonging to the same object category and O is the set of all available objects. The goal g is placed in the environment at a position $\boldsymbol{z}$. Additionally, n distinct instances o^j ($o^j \in C^a \wedge i \neq j$) are positioned in the environment, along with m distinct instances o^k ($o^k \in (O \setminus C^a)$). At the end of the episode, the navigation is considered successful if the agent selects the '*stop*' action before the maximum allowed number of timesteps T, with an Euclidean distance between the position of the agent at the current timestep $\boldsymbol{x}_t$ and the target position $\boldsymbol{z}$ lower than 1 meter. The action space of the agent for the task is defined by six possible actions, where at each timestep t, the action $a_t \in \{$'*stop*', '*move ahead*', '*turn left*', '*turn right*', '*tilt up*', '*tilt down*'$\}$.

3.2 Comparison with Other Tasks

The proposed task locates itself among PointNav [2], ObjectNav [2, 6], ImageNav [19], and the recently defined task of InstanceImageNav [36]. PIN exhibits similarities to ObjectNav, InstanceImageNav, and the recently introduced GOAT-Bench [32] (see Sec. 2).

However, it diverges from the traditional ObjectNav task because, differently from the standard objective of finding any instance of a general object category, PIN requires locating a specific instance, such as "*black and white striped trekking backpack*" instead of any "*backpack*". PIN leverages zero-shot properties at the instance level, as the object instances used for the training split differ from those included in the validation episodes. This requires agents to focus on the specific characteristics of the target object defined by the input references and avoid being misled by instances of the same category that are not the actual target.

Furthermore, PIN differs from InstanceImageNav and GOAT-Bench in various aspects. First, the target object is represented by a collection of images with neutral backgrounds, rather than being shown in its current spatial context. InstanceImageNav and GOAT-Bench are based on a set of general object categories that are included in the dataset of scenes and, therefore, these objects are static and frozen in the 3D model of the environment. Instead, the peculiarity of PIN is that it is created using a set of additional photo-realistic personal objects from a collection of 3D objects that can be placed and moved in different locations of the environment between different episodes. Using additional objects allows to avoid reconstruction errors and artifacts that can distort the appearance of the target.

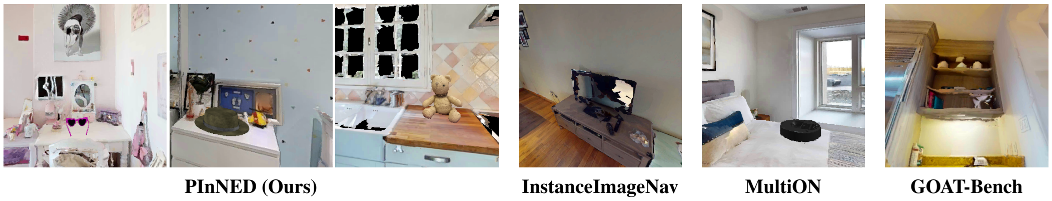


Figure 2: Comparison of observations depicting different targets in the embodied setting of our PInNED dataset with the target objects of MultiON, InstanceImageNav, and GOAT-Bench datasets.

This unique characteristic compels the agent to discern and extract the defining features of the target object while maintaining invariance to the surrounding context in which it is situated since personal objects can be moved frequently and could be placed in multiple suitable locations.

Similarly to GOAT-Bench, PIN provides a multimodal input to the agent, including textual descriptions of the target instances alongside the images. However, GOAT-Bench ignores the presence of instances of the same category of the target in the scene, whereas this is the core challenge of PIN. Additionally, it is worth noting that while text alone can sometimes provide precise identification of the specific instance, it can also be ambiguous. Visual references, although generally clearer, are not always available in real-world scenarios. Therefore, the two modalities cover different real-world requirements and both deserve to be studied. An extensive comparison of current object-driven dataset properties is reported in Table 1, which presents the following columns:

- *Photo-Realistic Scenes*: the presence of photo-realistic scans taken from real-world environments (e.g. the scenes of HM3D are built from scans of real environments, while scenes in AI2-THOR are hand-built by professional 3D artists);
- *Photo-Realistic Targets*: the availability of photo-realistic objects that can be used as navigation targets. In PInNED we carefully selected objects with realistic appearances. Procedurally-generated datasets, instead, tend to favor customizability over realism;
- *Additional Objects*: the inclusion of target objects that were not present at the time of capture. Datasets like GOAT-Bench target objects which were already present in the acquired scene, while PInNED targets objects injected in the scene afterward;
- *Visual Reference*: providing visual target references for each navigation episode;
- *Descriptive Reference*: providing natural language descriptions as targets for each episode;
- *Variable Placement*: the possibility of having variable spawning positions for the targets within the dataset;
- *Instance Goal*: the inclusion of navigation episodes in which the goal is to reach the exact instance indicated to the agent.

Moreover, a qualitative comparison of goal objects observed in their position in the environment from different datasets is depicted in Fig. 2.

3.3 Dataset

Categories and Instances. We selected a pool of 18 object categories from the assets contained in Objaverse-XL dataset [22]: '*backpack*', '*bag*', '*ball*', '*book*', '*camera*', '*cellphone*', '*eyeglasses*', '*hat*', '*headphones*', '*keys*', '*laptop*', '*mug*', '*shoes*', '*teddy bear*', '*toy*', '*visor*', '*wallet*', '*watch*', for a total of 338 additional objects. Each category contains an average of 18.8 objects, with a standard deviation of 5.5. The 3D objects are selected with human supervision to ensure photo-realism and uniqueness, which are critical requirements for tackling the PIN task. Finally, the 3D models of the objects are manually rescaled to have comparable dimensions to their real-world counterparts. In this procedure, we rendered each given object in a scene from HM3D and varied the scale of the object until the result was realistic according to our judgment. Hence, each of the 338 additional objects has a manually fixed scale that is adopted when the object is injected into the navigation episodes.

Input References. The input images for each target personalized object are generated by rendering the 3D mesh of the object in an isolated setting. Specifically, the input images are not expected to match the camera specification of the navigating agent [36]. The digital camera undergoes a

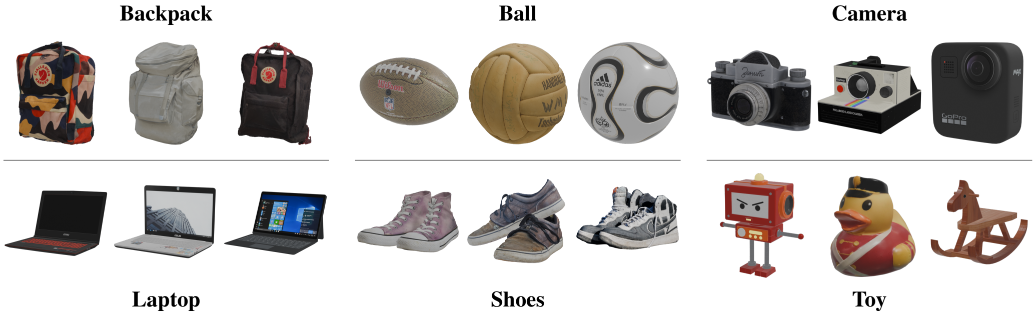


Figure 3: Sample input images of personalized targets from PInNED dataset. We include three instances from various object categories within the dataset.

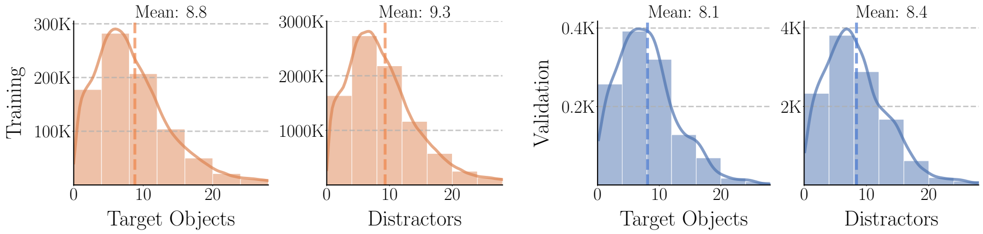


Figure 4: Plots of the distance statistics for the splits of PInNED dataset. The episodes of the training (orange) and validation splits (blue) are presented in terms of geodesic distance from the start position to the target object (left) and to all the distractors (right). All the distances are plotted in meters, and the mean value of each plot is shown on top.

30-degree yaw rotation to capture a favorable perspective of the objects. Each instance is then rotated 180 degrees in yaw to view its reverse side, followed by a 90-degree pitch rotation to observe the object from above. This procedure produces a set of three input images for each target object. An illustration of the acquired reference images is displayed in Fig. 3. Moving on to the textual references, manually annotated descriptions are produced for each target personalized object with the scope of highlighting the details that allow the agent to distinguish it from other instances of the same category. Specifically, we provide three descriptions for each personalized object in the PInNED dataset. To annotate the descriptions, we provided two object instances at a time to the annotators, asking them to describe one of the two objects in such a way that it is distinguishable from the other. This procedure results in a total of 960 unique words and an average of 10.7 words per description. Additional samples of input references are included in the Appendix.

Scenes. The benchmark defined by the PIN dataset is situated in the indoor photo-realistic scenes (*e.g.*, apartments, offices, houses) within the semantically-annotated subset [74] of Habitat-Matterport3D (HM3D) [56] which consists of 145 environments for the training split and 36 for validation set. However, one validation scene is ignored as it represents an art gallery and has no suitable spawnable surfaces. HM3D was selected due to its status as the largest publicly available dataset of semantically annotated indoor spaces with photo-realistic quality for embodied navigation.

Episode Generation. During the generation of the dataset, the bounding boxes of the surfaces in the environment are extracted using the semantic annotations of the scene. To obtain the bounding box from the texture, we extracted the point cloud 3D model of each scene and ensured that each point retained its associated annotation color. Subsequently, points were clustered by annotation color to create the bounding box associated with each piece of furniture. The spawning position of each object is selected by sampling from the positions of a curated set of suitable surface macro-categories included in the semantic annotations of HM3D. The surface categories selected for the creation of the dataset are: *armchair, bed, bench, cabinet, piano, rug, sofa, table*. These specific surfaces are chosen because of the high probability of personalized objects being positioned on top.

In each episode of the PInNED dataset, a single instance of a specific category is chosen as the target object. Consequently, up to 6 instances belonging to the same category, and up to 13 objects from other categories, are added to the environment as distractors. All additional objects placed in

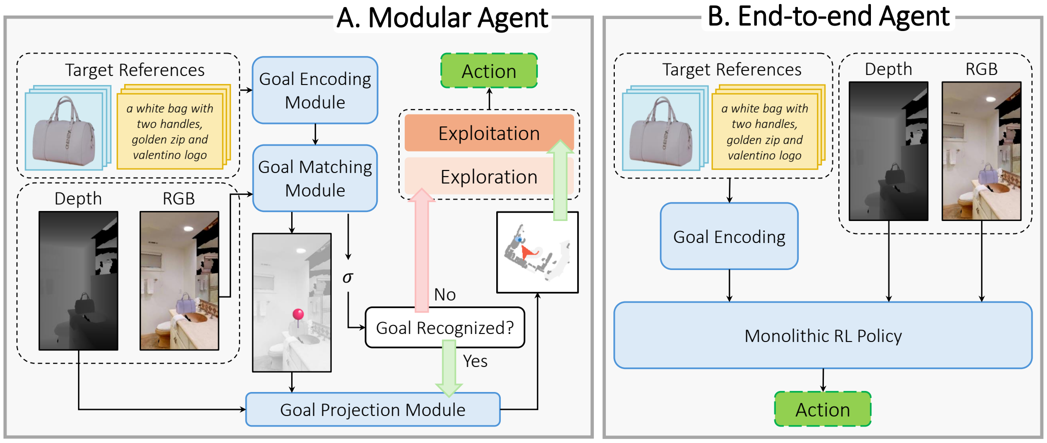


Figure 5: Overview of the baselines designed for the PIN task: modular agent (on the left) and end-to-end agent based on a monolithic reinforcement learning-based policy (on the right).

the environment are constrained to be on the same level/floor as the agent by selecting spawnable surfaces with a bounding box position within 0.5 meters from the starting position of the agent along the vertical axis. For each environment in the training split a set of 400 episodes is sampled for each one of the possible categories. For the generation of the validation split each target category is used twice. Finally, episodes where the target object is not reachable by an agent following the shortest path are removed from the dataset. Refer to the Appendix for more details on dataset generation.

The resulting dataset for PIN is defined by a total of $865,519$ generated episodes for the training split, while the validation split contains $1,193$ episodes. The geodesic distances of the target and distractors from the starting position of the agent in the episodes of PInNED are shown in Fig. 4. In the figure, the distribution of the distances of targets and distractors significantly overlap, hence prior information on the target object distance is hardly exploitable.

4 Baselines

In this section, we present the set of approaches that are revisited and tested on our introduced PInNED dataset. These methods are recent object-driven methods and can be grouped into two categories: (i) **modular agents** that decouple the navigation task into specialized sub-modules and (ii) **end-to-end agents** based on a monolithic policy trained using reinforcement learning. Fig. 5 shows an overview of these two approaches. We refer to the Appendix for more details on the implementation of the baselines.

4.1 Modular Agents

In recent years, modular agents gathered an increasing interest in various embodied settings. These agents tackle the high-level navigation tasks by decoupling them into a chain of specialized sub-modules, each of which handles a smaller task. Specifically, Chaplot *et al.* [15] proposed SemExp, a modular agent designed for the ObjectNav task composed of three main modules: exploration, object detection, and exploitation. The core idea is that the agent explores as much as possible the unseen environment while the detection module localizes the semantic objects in the acquired observations. Inspired by this approach, Mod-IIN [35] and CLIP on Wheels (CoW) [29] adapt the detection module to handle specific instances and open-vocabulary targets, respectively. For our modular agent baselines, we consider the same exploration and exploitation modules used in these previous works, while changing and adapting the object detection module for the PIN task.

Exploration Module. The exploration module is entitled to explore the unseen areas of the environment with the scope of encountering the target object. As in Mod-IIN and CoW, we adopt a frontier-based exploration (FBE [75]) approach. The agent builds an occupancy map of the environment during navigation, and at every time step, if the goal is not detected, the unexplored frontier on the map which is closest to the agent is selected as the current goal.

Object Detection Module. The object detection module receives the visual or textual references and the current RGB observation of the agent. Then, it is tasked with providing (i) a **matching score** that, whether it exceeds a certain matching threshold σ, determines that the goal has been recognized; and (ii) a **series of coordinates** on the observation which correspond to where the goal is located, that are used by the exploitation module to project the goal on a 2D map. We select three categories of approaches to implement this module:

- **Keypoint Matching**: In this category, the visual target references and the current RGB observation are provided to a keypoint matching method. We tested SuperGlue [60], following the approach proposed by Mod-IIN [35], and the framework introduced in IEVE [40]. In particular, SuperGlue outputs a confidence score for each matched keypoint pair. We use the sum of these confidences as the matching score and the keypoints that exceed a given confidence threshold τ as the localization coordinates. Regarding the Exploration-Verification-Exploitation framework proposed in IEVE, we adapted some components to match the different requirements of our task. Specifically, we first collected an auxiliary dataset, which includes, for each goal in the training set, 10 positive samples and one negative sample containing a distractor from the same category as the goal. We trained InternImage [69] to classify the 18 categories of our dataset using the goal images of the training set. Instead of the InternImage segmentation model, since, to the best of our knowledge, no segmentation dataset contains all our categories, we adopted the open-vocabulary segmenter GroundedSAM [59]. For the image-matching step, we exploited LightGlue [42] on the keypoints extracted with DISK [67] as in the original IEVE paper.
- **Patch-level Matching**: A Vision Transformer (ViT [26]) encoder divides an image into patches and extracts patch-level embeddings. Hence, we extract a goal embedding from each reference and compute the cosine similarity with the patch-level feature vectors of the RGB observation. If at least a patch has a similarity that exceeds the matching threshold σ, the goal is considered detected. The center coordinates of these patches are used as the goal localization result. For the visual references, we employ DINO [12], DINOv2 [51], and CLIP [54] performing a region pooling over the reference objects to obtain goal feature vectors. For the textual references, a text-aligned multimodal encoder is required. Hence, we employ CLIP and, inspired by [29], CLIP with gradient relevance [17] (CLIP-Grad). We assume the mean embedding of the set of prompt templates used in CoW applied to the target descriptions as the target feature vector.
- **Detection Model**: We consider detection models that produce output regions according to a given reference. Specifically, we consider PerSAM [80] (both in the standard and one-shot finetuned versions) and OWL [49], which localize regions according to, respectively, visual and textual references. As in CoW, we exploit the output confidence to determine whether the goal has been detected and return the central coordinates of the region as the goal localization result.

Exploitation Module. The exploitation module takes control of the navigation when the goal is recognized in the current observation. After detecting the target object at a given location, the exploitation module is triggered and computes the route to reach the target object. The goal position provided by the object detection module is projected into an occupancy map, and the Fast Marching Method [16, 62] is used to plan the path from the current position of the agent to the detected goal position. When the agent reaches the goal position, the '*stop*' action is called to conclude the episode.

4.2 End-to-End Agents

In contrast to modular agents, end-to-end approaches train a neural network policy to process sensor input and predict the atomic actions needed to complete the required task. We consider two recent approaches for embodied navigation and adapt them for the Personalized Instance-based Navigation task: (i) ZSON [45], which pre-trains an ImageNav agent and evaluates downstream on ObjectNav leveraging the capabilities of CLIP multimodal embeddings; and (ii) RIM [18], which employs a Transformer-based architecture [68] that is trained using auxiliary tasks and uses a recursive implicit map that is updated during the navigation for the ObjectNav task. We finetune both approaches on PInNED dataset. Specifically, ZSON is adapted to use image references as input during its ImageNav pretraining phase. While, for RIM, we employ two finetuning strategies: conditioning the navigation on textual features extracted from the reference descriptions and conditioning on visual features extracted from the image references. The features produced using both modalities of PInNED references are extracted using CLIP.

Table 2: Navigation results on PInNED on the environments of HM3D dataset, considering the presence of distractors from the same category. **Bold** text denotes the best performance among each category of approaches.

			Navigation Metrics					**Detection Metrics**			
	Backbone	Modality	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
Modular Agents											
CLIP [54]	ViT-B/16	Textual	3.10	1.82	9.31	7.94	503.1	62.95	20.07	22.07	57.86
CLIP-Grad [29]	ViT-B/32	Textual	4.53	2.42	6.95	7.94	465.8	77.95	4.65	7.21	84.14
OWL [29, 49]	ViT-B/32	Textual	7.29	3.36	12.66	7.90	871.7	22.97	**62.60**	32.88	4.52
SuperGlue [35, 60]	-	Visual	3.27	1.28	7.38	8.36	804.0	29.42	16.96	3.44	79.60
IEVE [40]	-	Visual	3.52	3.07	12.25	7.73	712.1	30.03	32.39	16.01	51.60
PerSAM [80]	ViT-B/16	Visual	2.77	1.81	6.53	8.20	362.5	**81.98**	1.15	10.43	88.42
PerSAM-F [80]	ViT-B/16	Visual	1.93	1.28	**5.63**	8.12	321.3	36.13	0.60	13.48	85.92
DINO [12]	ViT-B/16	Visual	4.02	1.71	6.88	8.28	826.0	23.89	62.73	**1.36**	35.91
CLIP [54]	ViT-B/16	Visual	9.64	5.39	13.33	7.79	623.5	58.51	32.53	16.35	51.12
End-to-end Agents											
RIM [18]	ResNet-50	Textual	7.12	6.67	**10.44**	8.43	409.3	-	-	-	-
RIM [18]	ResNet-50	Visual	8.80	6.80	13.41	8.48	402.1	-	-	-	-
ZSON [45]	ResNet-50	Visual	**9.14**	**7.18**	21.12	**7.00**	389.9	-	-	-	-

5 Experimental Evaluation

In this section, we present an experimental analysis of the selected baselines on the PIN task, discussing the set of metrics used to effectively evaluate the performances and the obtained results.

5.1 Evaluation Metrics

Traditional metrics for object-driven embodied navigation are **success rate** (SR) and **success rate weighted by path length** (SPL). SR is the ratio between the number of episodes where the agent successfully reaches the target object within a maximum distance of 1 meter and the total number of episodes, while SPL weighs the success rate with the length of the path taken by the agent. Moreover, we report the **average number of steps** taken by the agent and the **average distance from the goal** (D2G) at the end of each episode. The agent designed for tackling the PIN task should be able to distinguish whether the target object is present in the current observation while exploring the unseen environment and correctly localize it, within the timesteps budget T (set to 1,000). The main challenge is represented by distractor instances belonging to the same category as the target object. Hence, we introduce the **category error** (CE) metric, which measures the percentage of episodes in which the agent stopped within one meter from instances belonging to the same category of the goal.

In modular agents, the ability to detect the correct instance resides in having large matching scores when the target is present in the observation and small scores when the target is absent. Since in these agents it is possible to determine whether a given observation matches, we compute four additional metrics: the **percentage of episodes with at least a detected match** (%Match), the **percentage of matched observations** that contain the **target object** (TM), an **instance of the same category of the target** (CM), or **no relevant objects** (NM).

5.2 Experimental Results

Personalized Instance-based Navigation Experiments. In Table 2, we present the results on the PIN task. Among modular agents, DINOv2 performs best according to SR and SPL. The high values of TM, CM, and CE show that the obtained matches usually refer to the same category of the target instance. The same reasoning can be applied to OWL for the modular agents using textual references. However, OWL produces fewer matches as can be noted from the %Match metric. Models such as SuperGlue, PerSAM, and PerSAM-F, which exhibit low SR and TM, have also a corresponding high NM, demonstrating that they are not able to provide significant matching scores for distinguishing the correct instances or even the correct categories. It is noteworthy that SuperGlue struggles to match the instances of PInNED, which are represented on a neutral background, contrary to InstanceImageNav [35], where the reference image is a photo of the object

Table 3: Navigation results on PInNED on the environments of HM3D dataset, without considering the presence of distractors from the same category of the target. **Bold** text denotes the best performance among each category of approaches.

			Navigation Metrics				**Detection Metrics**		
	Backbone	Modality	SR↑	SPL↑	D2G↓	Steps	%Match↑	TM↑	NM↓
Modular Agents									
CLIP [54]	ViT-B/16	Textual	3.35	1.86	8.01	516.5	**61.86**	22.83	77.17
OWL [29, 49]	ViT-B/32	Textual	8.22	3.18	7.88	929.9	13.83	93.91	6.09
CLIP [54]	ViT-B/16	Visual	11.15	5.92	7.65	666.2	52.56	35.57	64.43
DINOv2 [51]	ViT-B/14	Visual	**23.13**	**11.61**	**6.62**	784.5	38.64	**96.09**	**3.91**
End-to-end Agents									
RIM [18]	ResNet-50	Textual	7.46	6.87	7.94	487.1	-	-	-
RIM [18]	ResNet-50	Visual	10.35	7.53	7.75	475.9	-	-	-
ZSON [45]	ResNet-50	Visual	**10.39**	**8.00**	**6.91**	460.1	-	-	-

in the same context in which it is located. Regarding PerSAM and PerSAM-F, the results show that the feature space of SAM [33] is not informative enough to understand whether an instance is present in an observation. IEVE shows an improvement with respect to the other image-matching modular agent, based on SuperGlue. This is motivated by the fact that IEVE, differently from other image-matching approaches, combines LightGlue with a semantic detector, allowing the agent to focus only on observations that contain objects of the target category. This behavior is confirmed by the increased numbers of target matches, category matches, and category errors.

Moreover, end-to-end agents tend to perform worse than modular agents. This can be attributed to the imitation training performed using the ground-truth trajectory to the goal. Since in the PIN task the target instances can be placed in multiple locations, it is not possible to exploit prior semantic knowledge about the estimated location of the target instance. Moreover, end-to-end agents tend to struggle in backtracking and in recovering the navigation when moving in the wrong direction. This behavior can also be noted from the path length, which for end-to-end agents is shorter than modular agents, that continue the exploration until the whole environment is observed.

Ablation on Category Distractors. In Table 3, we introduce an ablation study in which we remove the distractors belonging to the same category of the target instance. Overall, metrics for all the agents improve because the presence of these distractors represents the core challenge of the PIN task. In particular, DINOv2 improves by 8.29 with respect to the main experiments, demonstrating that it embeds strong semantic correspondence properties among the same category, but that it is not trivial to identify a threshold that clearly distinguishes specific instances. The impact of same-category distractors on end-to-end agents is minor since they are finetuned to identify the correct instance.

6 Conclusion

In this work, we presented the task of Personalized Instance-based Navigation (PIN) in which the agent is required to locate and navigate toward a specific target instance. Additionally, we release PInNED, a task-specific dataset built by injecting a set of additional photo-realistic objects in the scenes of HM3D. Finally, we perform an extensive analysis of recent navigation methods adapted for the proposed task. Experimental results demonstrate that the new challenges in the recognition of specific instances introduced in our proposed task are still far from being addressed. This benchmark sets a novel testbed for future work on embodied navigation toward personalized instances.

Acknowledgments and Disclosure of Funding

This work has been conducted under a research grant co-funded by Leonardo S.p.A. and supported by the EU Horizon project “ELIAS - European Lighthouse of AI for Sustainability” (No. 101120237), the project “Personalized Robotics as Service Oriented Applications (PERSEO)” funded under the Marie Sklodowska-Curie Action Horizon 2020 (No. 955778), and the PNRR project “Fit for Medical Robotics (Fit4MedRob)” funded by the Italian Ministry of University and Research.

References

[1] Ziad Al-Halah, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Zero Experience Required: Plug & Play Modular Transfer Learning for Semantic Visual Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[2] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On Evaluation of Embodied Navigation Agents. *arXiv:1807.06757*, 2018.

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[4] Luca Barsellotti, Roberto Amoroso, Lorenzo Baraldi, and Rita Cucchiara. FOSSIL: Free Open-Vocabulary Semantic Segmentation Through Synthetic References Retrieval. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2024.

[5] Luca Barsellotti, Roberto Amoroso, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Training-Free Open-Vocabulary Segmentation with Offline Diffusion-Augmented Prototype Generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[6] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. ObjectNav Revisited: On Evaluation of Embodied Agents Navigating to Objects. *arXiv:2006.13171*, 2020.

[7] Lorenzo Bianchi, Fabio Carrara, Nicola Messina, Claudio Gennaro, and Fabrizio Falchi. The devil is in the fine-grained details: Evaluating open-vocabulary object detectors for fine-grained understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[8] Roberto Bigazzi, Lorenzo Baraldi, Shreyas Kousik, Rita Cucchiara, and Marco Pavone. Mapping High-level Semantic Regions in Indoor Environments without Object Recognition. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024.

[9] Roberto Bigazzi, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Embodied Agents for Efficient Exploration and Smart Scene Description. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023.

[10] Maria A Bravo, Sudhanshu Mittal, Simon Ging, and Thomas Brox. Open-Vocabulary Attribute Detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[11] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The YCB Object and Model Set: Towards Common Benchmarks for Manipulation Research. In *Proceedings of the IEEE International Conference on Advanced Robotics*, 2015.

[12] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[13] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *Proceedings of the International Conference on 3D Vision*, 2017.

[14] Matthew Chang, Theophile Gervet, Mukul Khanna, Sriram Yenamandra, Dhruv Shah, So Yeon Min, Kavit Shah, Chris Paxton, Saurabh Gupta, Dhruv Batra, et al. GOAT: GO to Any Thing. In *Proceedings of Robotics: Science and Systems*, 2024.

[15] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object Goal Navigation using Goal-Oriented Semantic Exploration. In *Advances in Neural Information Processing Systems*, 2020.

[16] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural Topological SLAM for Visual Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[17] Hila Chefer, Shir Gur, and Lior Wolf. Transformer Interpretability Beyond Attention Visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[18] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object Goal Navigation with Recursive Implicit Maps. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2023.

[19] Yunho Choi and Songhwai Oh. Image-Goal Navigation via Keypoint-Based Reinforcement Learning. In *Proceedings of the IEEE International Conference on Ubiquitous Robots*, 2021.

[20] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, Act, and Ask: Open-World Interactive Personalized Robot Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024.

[21] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, et al. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[22] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-XL: A Universe of 10M+ 3D Objects. In *Advances in Neural Information Processing Systems*, 2023.

[23] Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-Scale Embodied AI Using Procedural Generation. In *Advances in Neural Information Processing Systems*, 2022.

[24] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

[25] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperPoint: Self-Supervised Interest Point Detection and Description. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2018.

[26] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*, 2021.

[27] Raphael Druon, Yusuke Yoshiyasu, Asako Kanezaki, and Alassane Watt. Visual Object Search by Learning Spatial Context. *IEEE Robotics and Automation Letters*, 2020.

[28] Heming Du, Xin Yu, and Liang Zheng. Learning Object Relation Graph and Tentative Policy for Visual Navigation. In *Proceedings of the European Conference on Computer Vision*, 2020.

[29] Samir Yitzhak Gadre, Mitchell Wortsman, Gabriel Ilharco, Ludwig Schmidt, and Shuran Song. CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[30] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. Datasheets for Datasets. *Communications of the ACM*, 2021.

[31] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2Real Predictivity: Does Evaluation in Simulation Predict Real-World Performance? *IEEE Robotics and Automation Letters*, 2020.

[32] Mukul Khanna, Ram Ramrakhya, Gunjan Chhablani, Sriram Yenamandra, Theophile Gervet, Matthew Chang, Zsolt Kira, Devendra Singh Chaplot, Dhruv Batra, and Roozbeh Mottaghi. GOAT-Bench: A Benchmark for Multi-Modal Lifelong Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[33] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[34] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv:1712.05474*, 2017.

[35] Jacob Krantz, Theophile Gervet, Karmesh Yadav, Austin Wang, Chris Paxton, Roozbeh Mottaghi, Dhruv Batra, Jitendra Malik, Stefan Lee, and Devendra Singh Chaplot. Navigating to Objects Specified by Images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[36] Jacob Krantz, Stefan Lee, Jitendra Malik, Dhruv Batra, and Devendra Singh Chaplot. Instance-Specific Image Goal Navigation: Training Embodied Agents to Find Object Instances. *arXiv:2211.15876*, 2022.

[37] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision*, 2017.

[38] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari. The Open Images Dataset V4. *International Journal of Computer Vision*, 2020.

[39] Federico Landi, Roberto Bigazzi, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. Spot the Difference: A Novel Task for Embodied Agents in Changing Environments. In *Proceedings of the International Conference on Pattern Recognition*, 2022.

[40] Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware Exploration-Verification-Exploitation for Instance ImageGoal Navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[41] Weijie Li, Xinhang Song, Yubing Bai, Sixian Zhang, and Shuqiang Jiang. ION: Instance-level Object Navigation. In *Proceedings of the ACM International Conference on Multimedia*, 2021.

[42] Philipp Lindenberger, Paul-Edouard Sarlin, and Marc Pollefeys. LightGlue: Local Feature Matching at Light Speed. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[43] Yang Liu, Muzhi Zhu, Hengtao Li, Hao Chen, Xinlong Wang, and Chunhua Shen. Matcher: Segment Anything with One Shot Using All-Purpose Feature Matching. In *Proceedings of the International Conference on Learning Representations*, 2024.

[44] Haokuan Luo, Albert Yue, Zhang-Wei Hong, and Pulkit Agrawal. Stubborn: A Strong Baseline for Indoor Object Navigation. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2022.

[45] Arjun Majumdar, Gunjan Aggarwal, Bhavika Devnani, Judy Hoffman, and Dhruv Batra. ZSON: Zero-Shot Object-Goal Navigation using Multimodal Goal Embeddings. In *Advances in Neural Information Processing Systems*, 2022.

[46] Oleksandr Maksymets, Vincent Cartillier, Aaron Gokaslan, Erik Wijmans, Wojciech Galuba, Stefan Lee, and Dhruv Batra. THDA: Treasure Hunt Data Augmentation for Semantic Navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[47] Bar Mayo, Tamir Hazan, and Ayellet Tal. Visual Navigation with Spatial Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[48] Lingchen Meng, Shiyi Lan, Hengduo Li, Jose M Alvarez, Zuxuan Wu, and Yu-Gang Jiang. SEGIC: Unleashing the Emergent Correspondence for In-Context Segmentation. In *Proceedings of the European Conference on Computer Vision*, 2024.

[49] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple Open-Vocabulary Object Detection. In *Proceedings of the European Conference on Computer Vision*, 2022.

[50] Arsalan Mousavian, Alexander Toshev, Marek Fišer, Jana Košecká, Ayzaan Wahid, and James Davidson. Visual Representations for Semantic Target Driven Navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019.

[51] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning Robust Visual Features without Supervision. *arXiv:2304.07193*, 2023.

[52] Anwesan Pal, Yiding Qiu, and Henrik Christensen. Learning hierarchical relationships for object-goal navigation. In *Proceedings of the Conference on Robot Learning*, 2021.

[53] Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A Co-Habitat for Humans, Avatars and Robots. In *Proceedings of the International Conference on Learning Representations*, 2024.

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the International Conference on Machine Learning*, 2021.

[55] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. PONI: Potential Functions for ObjectGoal Navigation with Interaction-free Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[56] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. In *Advances in Neural Information Processing Systems*, 2021.

[57] Ram Ramrakhya, Dhruv Batra, Erik Wijmans, and Abhishek Das. PIRLNav: Pretraining With Imitation and RL Finetuning for ObjectNav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[58] Niyati Rawal, Roberto Bigazzi, Lorenzo Baraldi, and Rita Cucchiara. AIGeN: An Adversarial Approach for Instruction Generation in VLN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024.

[59] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks. *arXiv:2401.14159*, 2024.

[60] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. SuperGlue: Learning Feature Matching With Graph Neural Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[61] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[62] James A Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. In *Proceedings of the National Academy of Sciences*, 1996.

[63] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A Large-Scale, High-Quality Dataset for Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

[64] Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, et al. iGibson 1.0: a Simulation Environment for Interactive Tasks in Large Realistic Scenes. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2021.

[65] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[66] Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training Home Assistants to Rearrange their Habitat. In *Advances in Neural Information Processing Systems*, 2021.

[67] Michał Tyszkiewicz, Pascal Fua, and Eduard Trulls. DISK: Learning local features with policy gradient. In *Advances in Neural Information Processing Systems*, 2020.

[68] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All You Need. In *Advances in Neural Information Processing Systems*, 2017.

[69] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. InternImage: Exploring Large-Scale Vision Foundation Models With Deformable Convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[70] Saim Wani, Shivansh Patel, Unnat Jain, Angel X. Chang, and Manolis Savva. Multi-ON: Benchmarking Semantic Map Memory using Multi-Object Navigation. In *Advances in Neural Information Processing Systems*, 2020.

[71] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-World Perception for Embodied Agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[72] Karmesh Yadav, Arjun Majumdar, Ram Ramrakhya, Naoki Yokoyama, Alexei Baevski, Zsolt Kira, Oleksandr Maksymets, and Dhruv Batra. OVRL-V2: A simple state-of-art baseline for ImageNav and ObjectNav. *arXiv:2303.07798*, 2023.

[73] Karmesh Yadav, Ram Ramrakhya, Arjun Majumdar, Vincent-Pierre Berges, Sachit Kuhar, Dhruv Batra, Alexei Baevski, and Oleksandr Maksymets. Offline Visual Representation Learning for Embodied Navigation. In *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[74] Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-Matterport 3D Semantics Dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[75] Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*, 1997.

[76] Wei Yang, Xiaolong Wang, Ali Farhadi, Abhinav Gupta, and Roozbeh Mottaghi. Visual Semantic Navigation using Scene Priors. In *Proceedings of the International Conference on Learning Representations*, 2019.

[77] Joel Ye, Dhruv Batra, Abhishek Das, and Erik Wijmans. Auxiliary Tasks and Exploration Enable ObjectNav. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[78] Xiaohua Zhai, Xiao Wang, Basil Mustafa, Andreas Steiner, Daniel Keysers, Alexander Kolesnikov, and Lucas Beyer. LiT: Zero-Shot Transfer With Locked-Image Text Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[79] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A Tale of Two Features: Stable Diffusion Complements DINO for Zero-Shot Semantic Correspondence. In *Advances in Neural Information Processing Systems*, 2024.

[80] Renrui Zhang, Zhengkai Jiang, Ziyu Guo, Shilin Yan, Junting Pan, Hao Dong, Peng Gao, and Hongsheng Li. Personalize Segment Anything Model with One Shot. In *Proceedings of the International Conference on Learning Representations*, 2024.

[81] Minzhao Zhu, Binglei Zhao, and Tao Kong. Navigating to Objects in Unseen Environments by Distance Prediction. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2022.

[82] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2017.

Table A: Configuration of the main parameters used for executing each episode of the PIN task contained in the PInNED dataset.

Action Space		***Episode Configuration***		***Depth Sensor***	
forward step	0.25	*success distance*	1.0	*width*	360
turn angle	30	*max episode steps*	1000	*height*	640
tilt angle	30	***RGB Sensor***		*hfov*	42
Agent Configuration		*width*	360	*position*	[0, 1.31, 0]
visual sensors	rgb, depth	*height*	640	*min depth*	0.5
height	1.41	*hfov*	42	*max depth*	5.0
radius	0.17				
position	[0, 1.31, 0]				

A Dataset and Codebase Release

The dataset and codebase of our work are released at the following link[1]. We provide the instructions to download the assets contained in the PInNED dataset and the codebase to run the main experiments on the Personalized Instance-based Navigation (PIN) task.

B Limitations

A limitation of this work is related to the visual appearance of some of the object instances in the PInNED dataset. For example, the Habitat simulator's [61] rendering can cause a deterioration in the texture quality of some objects, failing to accurately reproduce them in the environment. Moreover, instances with very small or detailed components can also exhibit a degradation in their visual fidelity when instantiated in the simulator. Consequently, as the agent moves farther from these objects, their details become less discernible. As a direct consequence, detecting small target objects is a critical challenge for navigation agents tackling the PIN task.

This behavior is showcased in Sec. E, where agents tackling the PIN task in the episodes of PInNED dataset face significant challenges in successfully detecting instances of inherently small object categories. In fact, despite agents such as the modular agent with DINOv2 [51] showcase good performance on the overall PIN task, detecting small objects represents one of the main limitations of current object-driven agents, as they can only be recognized when the robot is close to them.

A possible future improvement could involve designing novel exploration policies that aim to bring the robot closer to surfaces where the target might be placed while leveraging different detection criteria that take into consideration the scale of the observed objects.

C Broader Impacts

The introduction of the Personalized Instance-based Navigation (PIN) task and the accompanying PInNED dataset has the potential to advance the field of visual navigation and Embodied AI. The PIN task fills the limitations of the current datasets for embodied navigation by requiring agents to distinguish between multiple instances of objects from the same category, thereby enhancing their precision and robustness in real-world scenarios. This advancement can lead to more capable and reliable robotic assistants and autonomous systems, especially in household settings. Moreover, the PInNED dataset serves as a comprehensive benchmark for the development and evaluation of novel algorithms in object-driven navigation. By providing a challenging and extensive dataset, we encourage the research community to develop innovative approaches and solutions.

D Additional Personalized Instance-based Navigation Details

Configurations. In addition to the task definition details provided in Sec. 3.1 of the main paper, relevant hyperparameters employed for executing each episode of the PInNED dataset are presented in Table A.

[1] https://github.com/aimagelab/pin

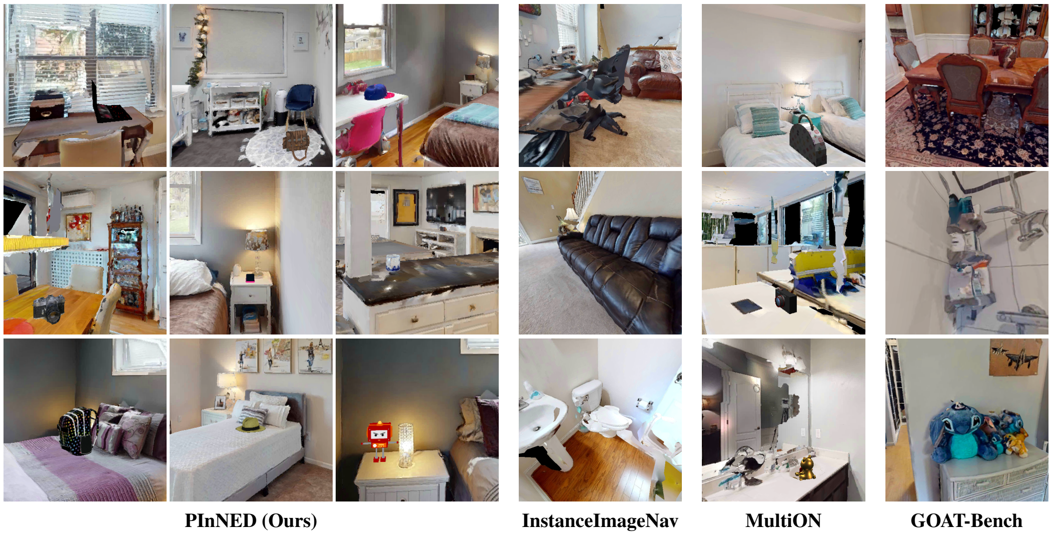


Figure A: Comparison of observations depicting different target objects of PInNED dataset with the target objects of InstanceImageNav, MultiON, and GOAT-Bench datasets.

The configuration used for a PIN episode comprises a maximum duration of $1,000$ time steps, with the agent's action space defined by discrete forward steps of 0.25 m, a turn angle of $30°$, and a head tilt angle of $30°$. Each episode is considered successful if the position of the agent is within 1 meter from the position of the target object, and it predicts the '*stop*' action before the end of the time step budget. The configurations used for the navigation experiments reflect the settings employed to simulate the camera sensors and space occupation of the HelloRobot Stretch[2] platform.

Comparison with Object-oriented Tasks. In addition to Fig. 2 of the main paper, in Fig. A we showcase additional examples of goal objects captured in the embodied setting for different object-driven datasets. The target objects belonging to the PInNED dataset are compared with InstanceImageNav [36], MultiON [70], and GOAT-Bench [32] datasets. It is noticeable that injecting photo-realistic objects allows to have targets that do not present artifacts or reconstruction errors, which is common for InstanceImageNav and GOAT-Bench target objects. Furthermore, when comparing the target objects of PInNED with those in the MultiON dataset, it is noticeable that the PInNED objects exhibit a more photo-realistic visual quality.

Comparison with ProcTHOR. ProcTHOR [23] is a framework built on AI2-THOR [34] to procedurally generate interactive environments, enabling the evaluation of data augmentation and large-scale training in different Embodied AI tasks. PInNED is a dataset designed specifically to study the newly introduced PIN task, in which the agent is tasked with finding a specific instance according to target images or textual descriptions.

ProcTHOR includes 1,633 instances across 108 object categories, with the ability to vary brightness, colors, materials, and object states. These categories include several household objects, covering generic objects, such as '*pen*' or '*apple*', objects that can be personal, such as '*mug*' and '*watch*', and large objects that are unlikely to change their placement in the environment, such as '*fridge*', and '*window*'. PInNED presents 18 object categories that can be personal, with the specific purpose of accompanying the task in which the agent has to distinguish instances belonging to the same category. All the categories represent objects that can be moved frequently in the environment and do not have a predefined location.

As well as most procedural datasets, ProcTHOR sacrifices realism in favor of interactivity, scalability, and customizability. PInNED, as a task-specific dataset, favors photo-realistic environments and objects. Indeed, it is the first instance-based navigation dataset based on both photo-realistic environments and injected objects, that can be moved frequently and with multimodal targets. Interactivity with the objects is out of scope for this work, however, the addition of external objects paves the way for possible future enhancements where interactivity is needed.

[2] https://hello-robot.com/stretch

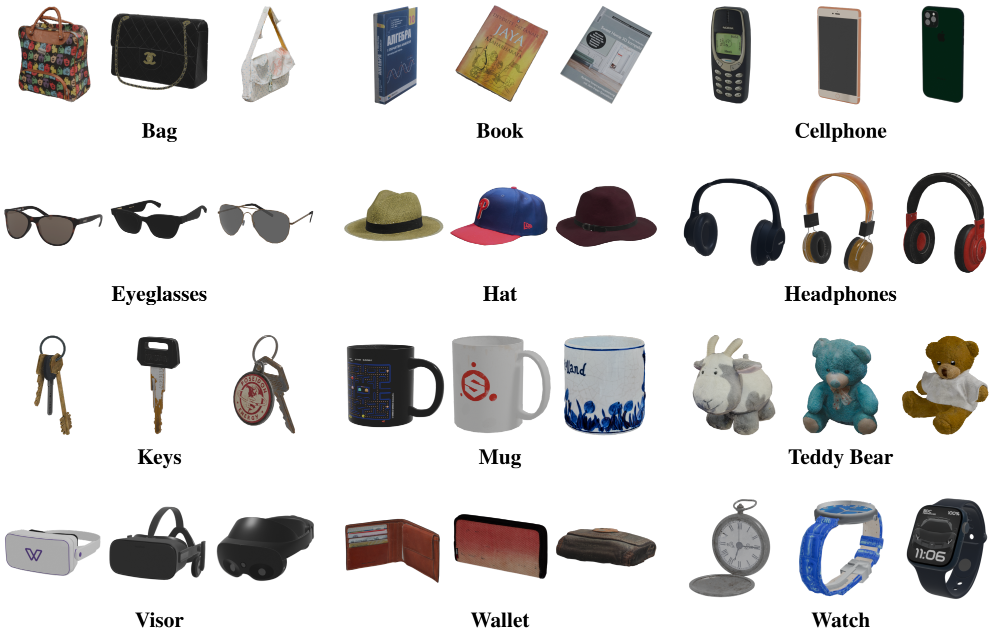


Figure B: Sample frontal visual references of personalized targets from PInNED dataset. We include three instances for each object category, considering the categories not included in Fig. 3 of the main paper.

Table B: Statistics about the number of distractors placed in the episodes of the training and validation sets of PInNED dataset. We consider the distractors belonging both to the same category of the target and to other categories.

# of Distractors	*Same Object Category*		*Other Categories*	
	Train	**Val**	**Train**	**Val**
Max	6	3	13	10
Average	2.93	2.90	7.75	7.19
Standard Deviation	0.33	0.37	2.84	2.82

E Additional PInNED Dataset Details

Additional Reference Samples. To better visualize the content of PInNED dataset, in Fig. B we illustrate additional samples of the acquired visual references for the categories that are not included in Fig. 3 of the main paper.

Additionally, we present samples including both visual and textual modalities for the input references associated with some of the object instances of PInNED dataset in Fig. C and Fig. D. In particular, we show the three views composing the set of visual references and the three manually annotated descriptions for the textual references.

Object Selection and Distribution Criteria. The scope of PIN is to provide a benchmark to evaluate an agent tasked with finding a specific object that can be located anywhere in an unexplored environment, where distractors of the same category are present; hence, the object categories are selected according to the following criteria: (i) objects that are highly customizable in terms of shapes, colors, and other visual aspects, (ii) objects that are frequently moved and can be placed anywhere, and (iii) objects of common use for which is reasonable to ask a robot to find.

Additional Information about Dataset Generation. In Table B, we provide statistics on the number of distractors placed in the training and validation episodes of PInNED dataset. During the generation of PIN episodes, a maximum number of distractors, both from the same category as the target instance and from other categories, is sampled from the set of available objects. The final number of additional objects in each episode is determined by the number of suitable surfaces and the available space on these surfaces. During the dataset generation process, objects are positioned above these surfaces and

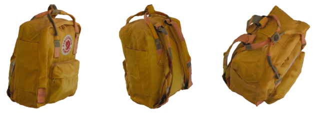
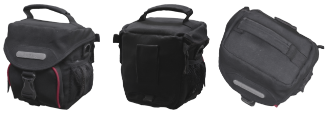
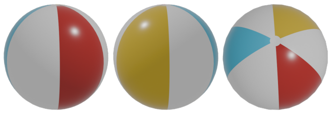
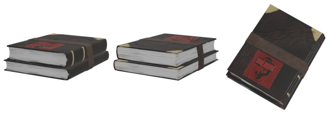
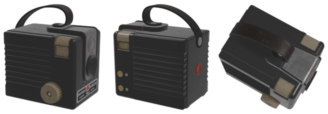
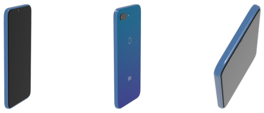
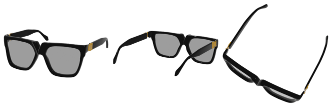
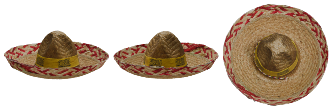
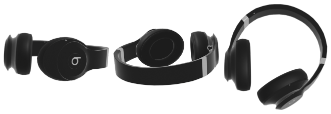
	*a yellow kanken backpack with yellow straps on the top* *a yellow monochrome kanken backpack* *a photo of a yellow backpack with a strap and red circle on the front*
	*a black camera bag with a handle and a mesh pocket* *a black camera bag with a buckle and a small silver plate* *a black camera bag with two red laces, a silver plate and a black buckle in the middle front*
	*a beach ball with alternated red, light blue and white slices* *an inflatable colored beach ball* *a beach ball with a multicolored design*
	*a stack of two books with a leather cover tied using a brown strap* *a brown book with yellowed pages with two straps and golden buckles on top* *two books tied together by a brown lace, with black leather covers and a red jurassic park logo*
	*a big black camera with a black handle and a wheel on the side* *a black cubic camera with a brown knob and a strap* *a kodak brownie hawkeye black flash camera, which is cube-shaped and has a black handle*
	*a blue smartphone with a white text on the back* *a blue phone with a black screen* *a cellphone with a gradient blue to purple color, two lenses, a fingerprint reader and the xiaomi mi logo on the back*
	*a pair of black squared eyeglasses with a golden plate on the arms* *a pair of sunglasses with a black frame and gold detail* *a pair of black thick eyeglasses with squared frame and golden hinges*
	*a sombrero with red details and a yellow stripe* *a straw hat with a yellow ribbon around it* *a sombrero with red elements on the brim and a yellow stripe with chiquito written multiple times on top*
	*a pair of black beats headphones* *a pair of headphones with a black band* *a pair of black headphones with the beats by dre logo on the ear cups and two gray lines on the headband*

Figure C: Visual reference images and textual reference descriptions of personalized targets from PInNED dataset. The samples are taken from '*backpack*', '*bag*', '*ball*', '*book*', '*camera*', '*cellphone*', '*eyeglasses*', '*hat*', and '*headphones*' object categories.

lowered until they contact the surface. If an object cannot be initially placed due to size constraints or collisions with other elements or walls, the placing process for that object is aborted, and another one is sampled from unused object instances. After the generation of the dataset of episodes, an additional assessment is performed through the Habitat simulator to remove the episodes containing objects that are not reachable from the starting position of the agent.

Object Distances. In addition to Fig. 4 of the main paper, Fig. E presents a plot depicting the Euclidean distances of target objects and distractors from the starting position of the agent in the

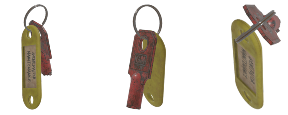
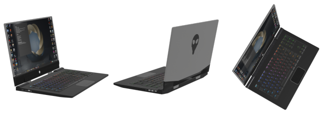
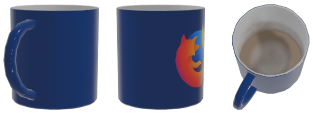
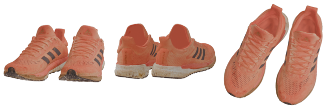
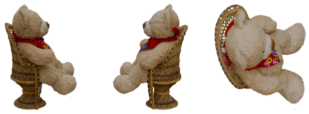
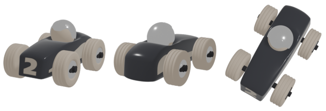
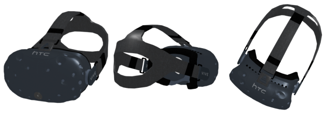
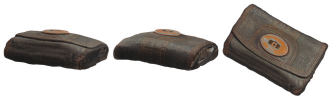
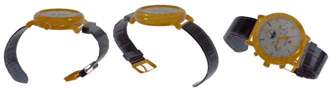
	*a worn red key and a yellow keytag with a black text* *a yellow plastic tag with a red key* *a red rusty key and a yellow keytag with generator maintenance written on it*
	*a black and grey laptop with rgb keyboard* *a black and grey laptop with a alien head on the back* *a laptop having a gray top cover with an alien logo on the back, a black base panel and a rainbow colored keyboard*
	*a blue mug with a red fox logo on it* *a blue mug with a firefox logo on it* *a blue mug with the mozilla firefox logo, composed of a red fox around the globe, printed on it*
	*a pair of orange adidas running shoes* *a pair of orange adidas sneakers with black stripes* *a pair of orange running shoes with orange laces, black adidas stripes, and white outsoles*
	*a beige teddy bear with a red bandana on a wooden chair* *a teddy bear sitting in a wicker chair with a red bandana on its neck* *a cream-colored smiling teddy bear with a red scarf and sitting on a woven chair*
	*a black and white toy car with a number 2 on the front* *a black and white toy car* *a black toy race car, with white wheels, a number 2 painted on the side, and a ball replacing the drive*
	*a black htc visor with blue polka dots* *a blue rounded virtual reality headset with a black strap* *a htc visor having black bands and blue front side with light blue dots*
	*a black leather wallet with an orange plate* *a brown leather wallet with a button on it* *a dark brown leather wallet having an orange patch with fossil written on it*
	*a gold and grey watch with black leather strap* *a brown leather wallet with a button on it* *a rounded watch having a thick golden case, white dial and black leather band with a golden buckle*

Figure D: Visual reference images and textual reference descriptions of personalized targets from PInNED dataset. The samples are taken from '*keys*', '*laptop*', '*mug*', '*shoes*', '*teddy bear*', '*toy*', '*visor*', '*wallet*', and '*watch*' object categories.

episodes of both training and validation splits of PInNED dataset. When considering the Euclidean distance, the distribution of the distances of the additional objects remains consistent with the geodesic distances presented in the main paper. Furthermore, the plots of the distances of all additional objects (target instances and distractors) are presented in Fig. F.

Modular Agent Activations. In Fig. G we present a comparison of the similarities computed between the patch-level features of different backbones on the observations of the agent and the references. In particular, we show these similarities on DINOv2 [51], DINO [12], CLIP with visual references,

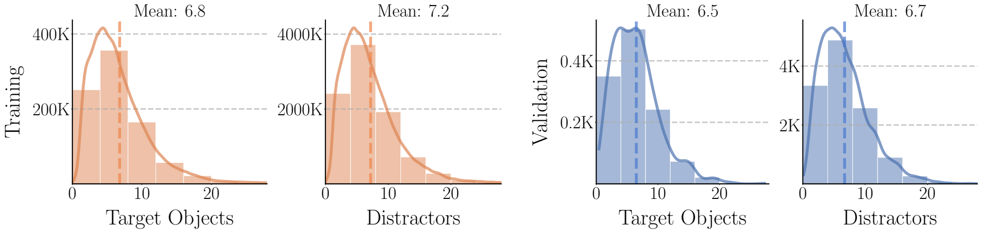


Figure E: Euclidean distances of the objects included in the episodes of training (orange) and validation (blue) splits of PInNED dataset. The plots consider the distances from the start position to the target object (left) and to all distractors (right). Distances are measured in meters, with the mean value for each plot displayed at the top.

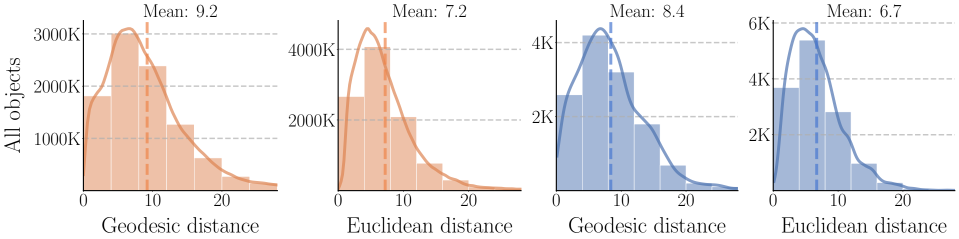


Figure F: Plots of the geodesic and Euclidean distances for all the objects placed in the episodes of PInNED dataset. Training (orange) and validation splits (blue) are presented in terms of distances from the start position to all the spawned additional objects. All the distances are plotted in meters, and the mean value of each plot is shown on top.

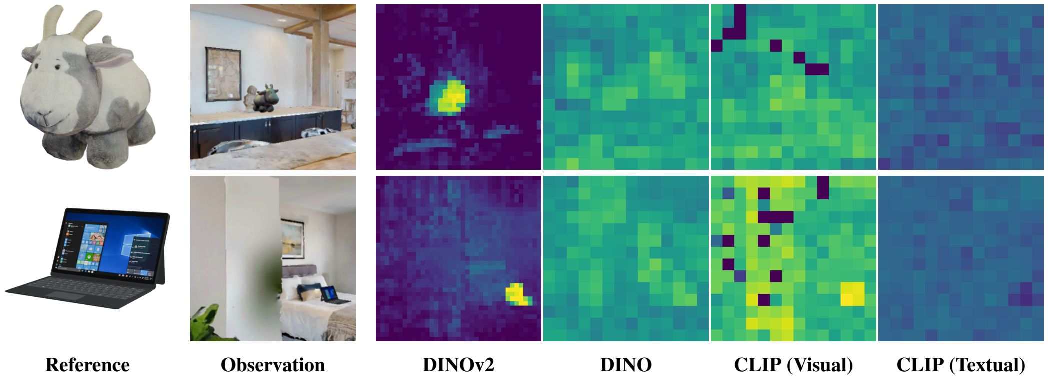


Figure G: Comparison of the similarities between the patch-level features on two observations of an agent extracted with different backbones, DINOv2, DINO, CLIP with visual references, and CLIP with textual references, and the references. Purple values represent low similarity values, while yellow values represent high similarity values.

and CLIP with textual references [54]. The resolution of the similarities extracted from DINOv2 is higher than the others since we employed the input resolution 518 × 518 on which the ViT-B/14 model has been trained, which corresponds to a grid of 37 × 37 patches, whereas DINO and CLIP are based on a ViT-B/16 backbone with 224 × 224 as input resolution. It is noteworthy that DINOv2 exhibits strong semantic localization properties, with high similarity values on the exact location of the image on which the target is observed. On the contrary, DINO and CLIP tend to exhibit less well-localized similarities. Moreover, CLIP with visual references has a high similarity on the patches corresponding to the laptop in the observation, whereas CLIP with textual references has a low similarity on the same patches.

Object Size Analysis. Taking into account that personalized objects are defined as predefined instances with distinct characteristics, the primary challenge in the PIN task lies in effectively recognizing these specific details, especially when dealing with subtle features and limited interaction capabilities within the environment. In this analysis, we present a category-wise size analysis of the

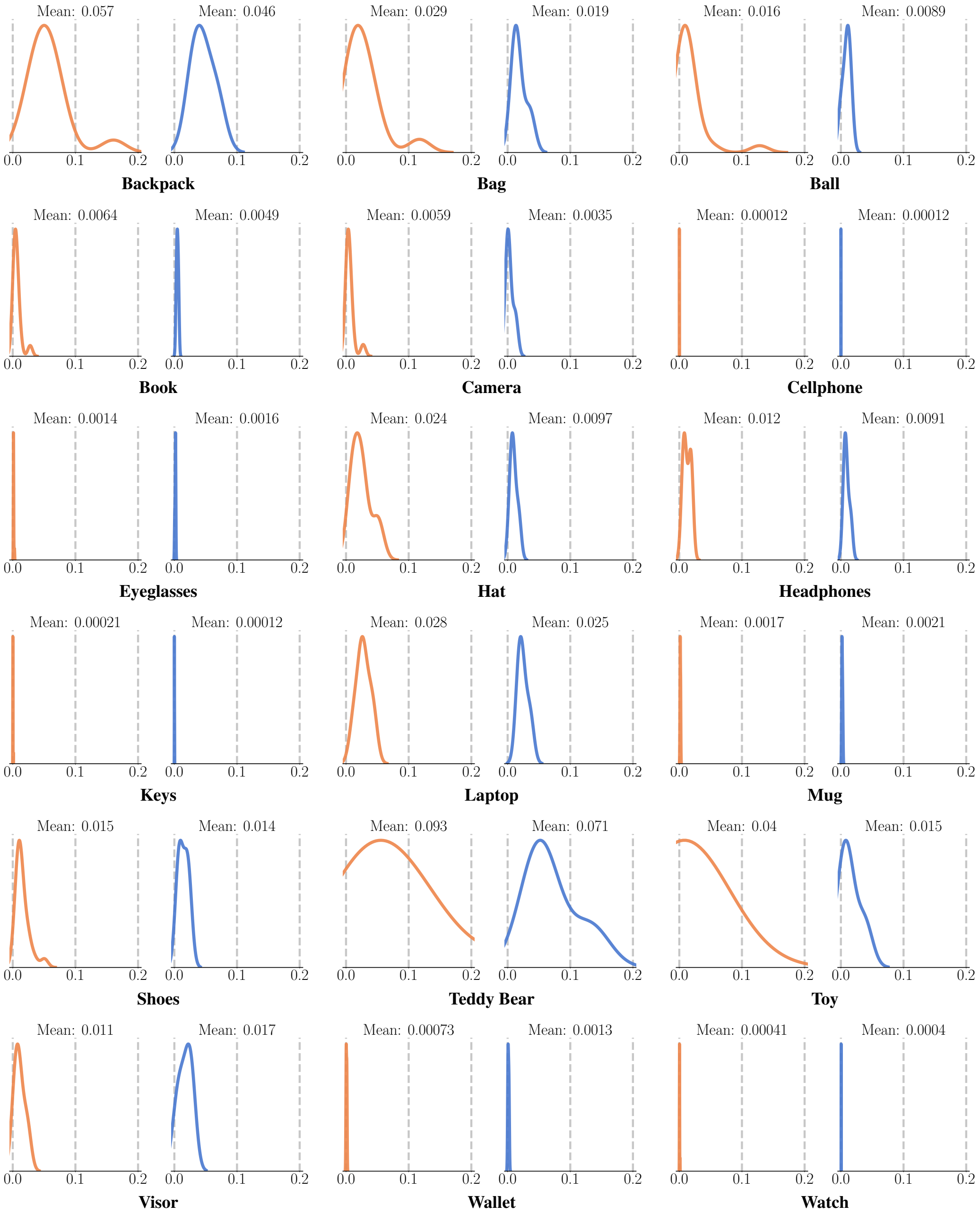


Figure H: Distribution of the volumes in meters of the bounding boxes of the objects in PInNED dataset. Two plots are shown for each semantic category, reflecting respectively the objects of training (orange) and validation (blue) splits. Each plot is accompanied by the corresponding mean of bounding box volumes of the objects in each split.

objects in the dataset by computing and measuring the 3D bounding box of each object. In Fig. H, we plot the distribution of the volumes of the bounding boxes associated with each object category showing that the distributions between training and validation splits remain consistent.

Category-wise Navigation Results. In Table C we present the navigation results of the modular agent based on DINOv2 as the matching backbone in which we compute the metrics for each category. From the results on SR and SPL we can note that there are categories that are easier to locate and reach, such as ‘*backpack*’, ‘*bag*’, ‘*ball*’, ‘*hat*’, ‘*laptop*’, and ‘*toy*’, and there are instances from categories that are never correctly reached, such as ‘*keys*’, ‘*wallet*’, and ‘*watch*’. This result returns the inability of the vanilla matching modules to distinguish these categories in the embodied setting.

Table C: Navigation results of the modular agent that employs DINOv2 as the matching module on the validation episodes of PInNED dataset, considering the performance of the agent for each category. Moreover, we report the average intra-category and inter-category cosine similarities computed on the frontal goal images.

	Navigation Metrics					**Detection Metrics**				**Similarity**	
Category	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓	Intra-Category	Inter-Category
Backpack	26.47	14.04	36.77	5.79	408.7	85.29	53.27	46.57	0.16	0.510	0.110
Bag	23.08	13.65	40.00	6.16	406.5	93.85	44.62	55.04	0.34	0.348	0.121
Ball	20.90	10.29	23.88	6.48	613.1	61.19	36.06	63.87	0.07	0.258	0.068
Book	19.40	10.83	35.82	5.71	484.3	86.57	58.51	40.16	1.33	0.613	0.106
Camera	7.46	3.38	7.50	8.57	883.2	20.90	69.23	23.08	7.69	0.152	0.050
Cellphone	8.96	3.11	14.92	8.63	844.8	32.84	7.81	90.96	1.23	0.506	0.112
Eyeglasses	10.45	5.08	32.83	7.70	682.0	62.69	79.80	19.95	0.25	0.846	0.104
Hat	26.87	11.95	23.88	6.45	652.8	67.16	88.08	11.89	0.03	0.549	0.084
Headphones	16.92	9.71	40.00	7.35	492.8	84.62	14.58	85.29	0.13	0.764	0.098
Keys	0.00	0.00	8.82	8.38	974.2	2.94	0.00	0.00	100.00	0.558	0.102
Laptop	21.54	11.50	49.23	7.01	455.3	93.85	16.86	82.60	0.54	0.348	0.084
Mug	10.61	4.47	10.61	8.10	911.8	22.73	92.00	4.50	3.50	0.298	0.073
Shoes	16.92	12.44	44.62	6.75	318.8	95.38	8.31	91.69	0.00	0.631	0.087
Teddy Bear	19.12	13.48	52.94	7.07	335.5	91.18	68.92	16.62	14.46	0.548	0.066
Toy	26.56	13.18	3.12	6.16	754.6	48.44	99.27	0.00	0.73	0.137	0.087
Visor	11.94	5.99	31.34	7.99	657.0	52.24	52.47	45.33	2.20	0.316	0.148
Wallet	0.00	0.00	6.15	8.39	985.3	1.54	0.00	0.00	100.00	0.282	0.105
Watch	0.00	0.00	7.69	8.39	999.0	0.00	0.00	0.00	100.00	0.566	0.102

Moreover, we can observe that there is an overall positive correlation between SR and average category size, implying that small objects are particularly challenging to detect.

Similarity Analysis. The similarity of objects is a critical factor in the PIN task. The presence of distractors increases the challenge of the proposed task, as the agent must balance between being overly cautious and overly confident when identifying target instances. This trade-off is central to the effectiveness of the navigation approaches. In particular, concerning images as references of the target object, re-identification methods should be a robust solution against distractors due to considering the matching between keypoints instead of the semantic similarity between observation and reference. Indeed, in Table 2 of the main paper, the state-of-the-art re-identification method SuperGlue has a lower category error than DINOv2 and CLIP. However, it presents the worst results according to SR and SPL, showing difficulties in matching keypoints when observation and reference have discrepancies in appearance. For methods based on semantic features, the similarity threshold is the key element in balancing confidence and caution.

In Table C, we report the average cosine similarities in the DINOv2 embedding space per category. In particular, we extracted the CLS token from each frontal goal image of the validation set and computed the cosine similarities against the other goal images from the same category (i.e. intra-category) and against goal images from different categories (i.e. inter-categories). The results show that the intra-category similarity presents a strong relation with the category error (CE) and category matches (CM) metrics. Indeed, the agent tends to mistake instances from categories with large intra-category similarity values, such as '*eyeglasses*', '*headphones*', and '*shoes*', while these mistakes are reduced in categories such as '*camera*' and '*toy*' that are characterized by a larger variability in their instances. When we adopt textual references as targets, the challenges concern how well multimodal spaces embed fine-grained details, and how similarity behaves accordingly. Previous work [7, 10] has shown that this challenge is non-trivial and still open. Our dataset represents a further step in this direction, providing a benchmark to evaluate the capabilities of visual-language models in recognizing fine-grained details. Future works can exploit our training set to instruct the models to distinguish instances of the same category by focusing on adjectives and attributes.

Surfaces Details. As described in Sec. 3.3, the spawning position of each object in the PInNED dataset is selected by sampling from the positions of a curated set of suitable surface macro-categories included in the semantic annotations of HM3D. The surface categories selected for the creation of the dataset are: *armchair, bed, bench, cabinet, piano, rug, sofa, table*. These surfaces are valid for all the object categories and there are no subsets of surfaces dedicated to specific categories. There are categories, especially '*shoes*', that are unlikely to be placed on certain surfaces. However, the scope of the task is to have objects that could be placed everywhere and teach a robotic agent to find them.

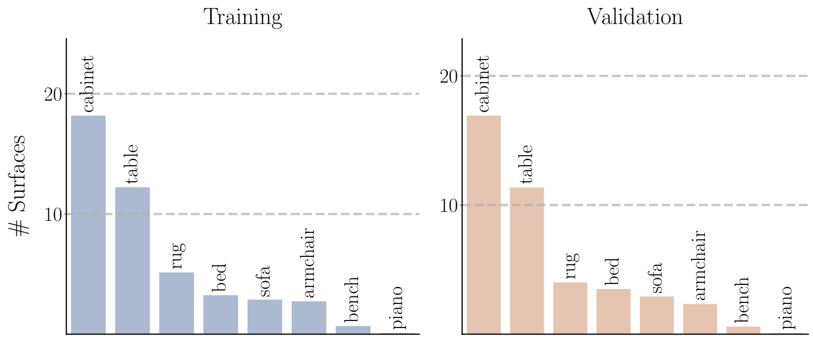


Figure I: Plot of the mean number of surfaces in each environment that are suitable for object placement in the training (left) and validation (right) splits of the PInNED dataset.

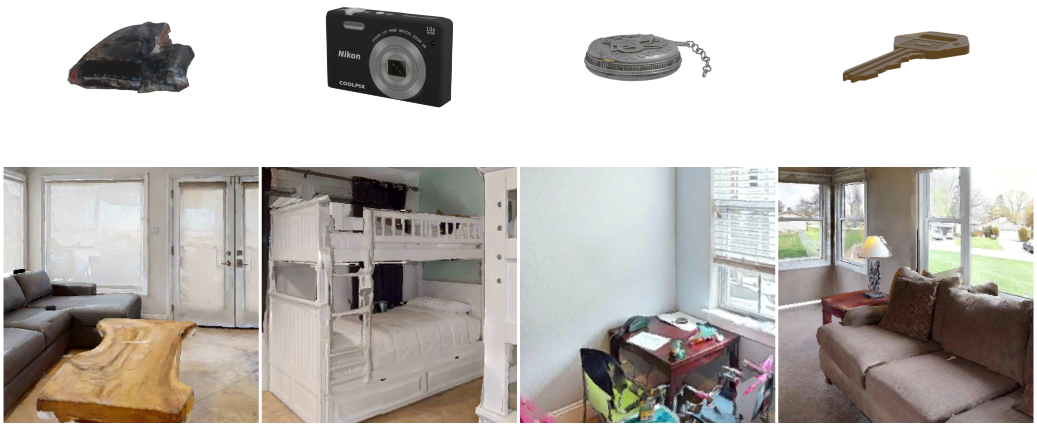


Figure J: Examples of situations in which detecting the target in the embodied environment is particularly challenging. We depict the frontal visual references of the target in the first row and a portion of an agent's observation containing the target in the second row.

A teddy bear is not necessarily located on the bed, but could be located anywhere, even on the kitchen table. If we assume a real-world scenario in which a child forgets the teddy bear on the kitchen table, the agent should not go directly to the bedroom, but look for the object in the whole environment. This is the reason for which we adopted a consistent spawning mechanism across all the categories. We identify this combination of objects that could be placed everywhere and the consistent spawning mechanisms as the correct approach for providing a dataset covering a large set of possible real-world scenarios that avoid the exploitation of prior knowledge on the object placement.

In Fig. I, we showcase the occurrences of the suitable surfaces in the environments of HM3D [56]. Notably, the distribution of spawnable surfaces remains consistent between the training and validation splits. This implies a recurring pattern in the furnishing of indoor spaces contained in the HM3D dataset and used for the PIN task.

Hard Detection Cases. In Fig. J, we show four episodes in which detecting the target is particularly challenging. These targets belong, respectively, to the '*wallet*', '*camera*', '*watch*', and '*keys*' categories. Table C shows that these categories are the most challenging ones for the modular agent with DINOv2, which is the best-performing agent according to Table 2. Indeed, the categories '*keys*', '*wallet*', and '*watch*' all yielded no successful episodes. These objects are hard to detect even for a human, confirming how challenging the PIN task is. Future work should investigate the possibility of moving the agent closer to areas in which there are small objects that cannot be identified as the target from longer distances.

Fine-Grained vs General Descriptions Comparison. In Table E we present an ablation study in which we compare the performance of the baselines with both fine-grained and general object

Table D: Navigation results of the modular agent that employs SuperGlue as the matching module on the validation episodes of PInNED dataset, considering different resize values of the visual references of the target provided to the matching module.

	Navigation Metrics					**Detection Metrics**			
Resize	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
360	2.51	0.82	**7.05**	8.48	881.6	17.77	**43.76**	5.17	**51.07**
180	3.02	1.20	7.21	8.48	864.1	20.70	21.72	3.58	76.35
180, 360	**3.27**	**1.28**	7.58	**8.36**	**804.0**	**29.42**	16.96	**3.44**	79.60

Table E: Navigation results on PInNED on the environments of HM3D dataset, comparing categorical and fine-grained textual modalities.

			Navigation Metrics					**Detection Metrics**			
	Backbone	Modality	SR↑	SPL↑	CE↓	D2G↓	Steps	%Match↑	TM↑	CM↓	NM↓
Modular Agents											
CLIP [54]	ViT-B/16	Categorical	3.52	2.75	10.23	7.98	148.1	95.47	5.12	15.73	79.15
CLIP [54]	ViT-B/16	Fine-Grained	3.10	1.82	9.31	7.94	503.1	62.95	20.07	22.07	57.86
OWL [29, 49]	ViT-B/32	Categorical	7.79	3.73	19.96	7.96	780.6	38.81	26.50	58.49	15.01
OWL [29, 49]	ViT-B/32	Fine-Grained	7.29	3.36	12.66	7.90	871.7	22.97	62.60	32.88	4.52
End-to-end Agents											
RIM [18]	ResNet-50	Categorical	4.61	3.78	14.25	9.23	336.0	-	-	-	-
RIM [18]	ResNet-50	Fine-Grained	7.12	6.67	10.44	8.43	409.3	-	-	-	-

categories. Specifically, we conducted the following experiments. For the modular agents based on CLIP and OWL as the matching module, we leveraged the general object category (e.g. backpack) instead of the fine-grained textual descriptions as navigation targets, while maintaining the same similarity threshold. The results on both CLIP and OWL present similar behaviors: the number of successful episodes is slightly increased, but also the number of episodes in which the agent mistakes reaching distractors of the same category and the number of matches with them increased. Moreover, the reduction in the average number of steps indicates that similarities, on average, are higher. The increase in successful episodes is surprising but in line with the findings of previous works in the literature [7, 10], which demonstrate that current vision-language models struggle with fine-grained details. These results show that our work can help future works in the realization and evaluation of vision-language models with improved understanding capabilities of details. Concerning the end-to-end agent RIM, we trained the model on the CLIP embeddings extracted from the general object categories instead of the fine-grained textual descriptions. The results show a lower number of successful episodes and a higher number of episodes in which the agent reaches a distractor of the same category.

F Additional Implementation Details

Modular Agents. In Sec. 4.1, we introduce the modular agents tested on the PIN task. Their ability to distinguish a specific instance in a given observation depends on the score threshold that maximizes the detection results. We tune this threshold on a subset of the training episodes. For all the backbones except for SuperGlue [60], we extract two squared crops with size 360 × 360 from the 360 × 640 observation and resize them to the image resolutions on which the backbones have been trained. Then, we consider all the matches resulting from the two crops. At least a match over the threshold is required to consider the goal detected in an observation. For the textual modalities, we employ the 80 prompt templates proposed by Radford *et al.* [54] for ImageNet [24]. In this section, we report additional implementation details for each backbone.

- **SuperGlue [60]**: We observe that SuperGlue struggles to match the visual references with the observations of the agent and that the resolution of the references influences the matching capabilities. In particular, we provide the visual references to SuperGlue as squared images 360 × 360, corresponding to the shortest side of the observation of the agent. For each visual reference, namely for each of the three views of the object, we provide two resizes of the object such that the longest side is, respectively, 360 and 180. This procedure results in two reference images for each view of the object, an image entirely occupied by the object and an image where the object occupies a quarter of it. In Table D we show that this approach results in a higher success rate than having a single image per object view. Moreover, we employ the *indoor* weights of SuperGlue with a threshold of 0.2 on the confidence of each matched keypoints pair and a matching threshold σ of 8.0 on the confidence sum of all the matched keypoints pairs.

- **CLIP [54]**: We employ CLIP ViT-B/16 with the pre-trained weights from OpenAI for both the experiments with visual and textual references. We resize the two observation crops to 224×224, resulting in a grid of 14×14 patches. The best matching threshold σ for the visual and textual modalities are, respectively, 0.575 and 0.28.
- **CLIP-Grad**: We follow the implementation of the network interpretability method proposed in CoW [29] on top of CLIP with textual references. We employ CLIP ViT-B/32 with the pre-trained weights from OpenAI and matching threshold 0.85.
- **OWL [49]**: OWL is an open-vocabulary detector that is trained in two steps: (i) a large contrastive image-text pre-training following LiT [78] and (ii) an object-level training on publicly available detection datasets (Open Images V4 [38], Objects 365 [63], and Visual Genome [37]). We employ a matching threshold of 0.25 applied to the predicted bounding box scores.
- **DINO [12]/DINOv2 [51]**: DINO is a self-supervised backbone pre-trained according to a self-distillation training paradigm. DINOv2 is an improved version of DINO with the aim of producing general-purpose visual features. We employ DINO ViT-B/16 and DINOv2 ViT-B/14 trained, respectively, on ImageNet-1k [24] and LVD-142M [51]. We use the same input image resolutions on which they are trained, namely 224×224 and 518×518, producing 14×14 and 37×37 grids of patches. The best matching scores are, respectively, 0.575 and 0.5.
- **PerSAM/PerSAM-F [80]**: We leverage the implementation of PerSAM on SAM ViT-B/16, trained on SA-1B, with input image resolution at $1,024$. PerSAM-F is a variant of PerSAM that fine-tunes the model on the reference image, We follow the training configuration of the original implementation. We consider the maximum patch-level similarity between the reference images and the observation crop as the matching score on which we apply the thresholds 0.925 and 0.61 for, respectively, PerSAM and PerSAM-F.

End-to-End Agents. As mentioned in Sec. 4.2 end-to-end approaches use a neural network policy which is trained end-to-end to directly process sensor observations and predict the atomic actions needed to fulfill the required task. In our case, we adapted two recent end-to-end approaches for ObjectNav finetuning them to perform PIN task: RIM [18] and ZSON [45].

- **RIM [18]**: The model is finetuned using behavior cloning following Chen *et al.* [18] approach and starting from the pre-trained weights for ObjectNav [6]. We evaluate two variants of the fine-tuned model, conditioned on visual features and conditioned on textual features. In RIM approach, besides the episodic implicit map that is updated recursively, the input of the policy at each timestep is composed of the concatenation of the features extracted from RGB and depth observation, the pose of the agent, previous action, and the target object category. To adapt RIM for the PIN task, we modify the features extracted from the object category label. Originally each label is associated with a row in a lookup table containing learnable embeddings of length 32. In our adaptation, we replace such embeddings with CLIP (ViT-B/16) features extracted using the visual or textual references. Since each input reference modality is described by 3 images or descriptions, we compute the mean of the features extracted from each reference. Following, a learnable linear layer is trained to project CLIP features to a vector of length 32. The resulting embedding is used to condition the navigation of the RIM agent. The fine-tuning process is performed on a single GPU for a total of $\approx$ 2M fine-tuning steps over $\approx$ 24 hours.
- **ZSON [45]**: For the adaptation of the ZSON method, we fine-tuned the model pre-trained on the ImageNav task, following the same approach as Majumdar *et al.* [45]. The agent is fine-tuned with reinforcement learning using an adaptation of ZSON reward but ignoring the angle to the goal since it is not a component considered in the PIN task. The resulting reward is $r_t = r_{success} - \Delta_{dtg} + r_{slack}$. We refer to Majumdar *et al.* [45] for a description of the components of the reward. Moreover, while the original approach uses ImageNav goals that are represented as photos captured at the position that the agent is required to reach, we used image references of the target instance to perform the fine-tuning. The model is fine-tuned on a single GPU for $\approx$ 24 hours for a total of $\approx$ 5M fine-tuning steps.

```
{
    "episode_id": "0",
    "scene_id": "hm3d/val/00800-TEEsavR23oF/TEEsavR23oF.basis.glb",
    "start_position": [-0.28, 0.013, -6.54],
    "start_rotation": [0, 0.98, 0, 0.20],
    "info": {"geodesic_distance": 8.24},
    "goals": [
        {
            "object_category": "backpack",
            "object_id": "3f5948f7f47343acb868072a7fe92ada",
            "position": [-5.13, 1.08, -0.81]
        }
    ],
    "distractors": [
        {
            "object_category": "backpack",
            "object_id": "3c47af8b6a3e413f94c74f86d4c396ed",
            "position": [-3.46, 2.20, -4.30]
        },
        {
            "object_category": "backpack",
            "object_id": "0b795895343b44b69191ef9b55b35840",
            "position": [-11.17, 0.88, -0.36]
        },
        {
            "object_category": "backpack",
            "object_id": "d86ee61984544b45a9f11f49e5e02c43",
            "position": [-9.13, 1.22, -3.52]
        },
        {
            "object_category": "mug",
            "object_id": "d26e9bfce2644bb7af6710c6511ea718",
            "position": [-7.84, 0.62, -0.14],
        },
        {
            "object_category": "laptop",
            "object_id": "6495988c6c044c76a2fc9f9278543c16",
            "position": [-1.64, 0.87, -6.15],
        },
        {
            "object_category": "headphones",
            "object_id": "ccf60b0502784fb38e483a6b07cfad53",
            "position": [3.41, 0.84, -8.21],
        },
    ],
    "scene_dataset_config": "data/scene_datasets/hm3d/hm3d_annotated_basis.scene_dataset_config.json",
    "object_category": "backpack",
    "object_id": "3f5948f7f47343acb868072a7fe92ada"
}
```

Listing A: Python dictionary containing a sample of the episodes contained in PInNED dataset. The list of distractors is skimmed for better visualization.

Compute Information. We performed our experiments on a computing platform composed of NVIDIA RTX5000 GPUs and 8 GB of CPU memory for each job. A job can be computed on a single GPU. Each episode step for the modular agents requires an average of $\approx$ 200ms to be executed. Hence, the entire DINOv2 experiment on the $1,193$ validation episodes, with an average number of steps equal to 658.7, requires ≈ 44 computation hours. The entire evaluation on the validation split for the end-to-end agents requires ≈ 5 computation hours.

```
{
    "scale": [0.116, 0.116, 0.116],
    "render_asset": "0a96f1f19afc432bb22c3d74da546338.glb",
    "requires_lighting": true,
    "up": [0.0, 1.0, 0.0],
    "front": [0.0, 1.0, 0.0],
    "COM": [0.0, 0.0, 0.0],
    "gravity": [0, 0, 0],
    "force_flat_shading": true,
    "is_collidable": true,
    "use_mesh_for_collision": true,
    "semantic_id": 2,
    "semantic_category": "ball"
}
```

Listing B: Python dictionary containing the information used by Habitat simulator to instantiate a specific object instance in the environment.

G Licenses and Terms of Use

The episodes of the PInNED dataset are built using the scenes from the HM3D dataset [56]. The scenes of the HM3D dataset are released under the Matterport End User License Agreement, which permits non-commercial academic use.

For the augmentation of HM3D scenes with additional objects, PInNED dataset utilizes 3D object assets from Objaverse-XL dataset [22]. Objaverse-XL is distributed under the ODC-By 1.0 license, with individual objects retrieved from various sources, including GitHub, Thingiverse, Sketchfab, Polycam, and the Smithsonian Institution. Each object is subject to the licensing terms of its respective source, necessitating users to evaluate license compliance based on their specific downstream applications.

Nevertheless, the specific objects included in our dataset are restricted to assets sourced from Sketchfab which are released under various Creative Commons licenses. Specifically, the dataset includes assets under the following licenses: CC BY (311 objects), CC BY-NC (14 objects), CC BY-SA (8 objects), CC BY-NC-SA (3 objects), and CC0 (2 objects).

The episodes of the PInNED dataset, along with the manually annotated object descriptions are released under the CC BY license, while the codebase for the PIN task is released under the MIT license.

The authors accept full responsibility for any rights violations arising from the use or publication of the data and content in this paper. All licenses related to external content included in this paper ensure no infringement on third-party rights.

H Assets

The episodes of PInNED dataset are defined as Python dictionaries containing relevant information for the execution of the PIN task with the Habitat simulator. An example of episode annotation is presented in Listing A. Each episode specifies the environment where it is taking place, the starting position and rotation of the agent, along with the position and object identifier of the target instance and the distractors.

The information used by the Habitat simulator to resize and instantiate each 3D object at the position specified by the episodes of PInNED dataset is also contained in a Python dictionary, where a specific file represents each object. A sample of object annotation is showcased in Listing B.

I Datasheet

In this section, we present a comprehensive datasheet [30] for the proposed dataset, providing a unified reference for relevant information on the PInNED episodes and the objects used to build the dataset.

I.a Motivation

For what purpose was the dataset created? The PInNED dataset has been built with the motivation of fostering future research on smart navigation agents. Such agents need to acquire the capability of distinguishing between different instances of the same object category and leverage different modalities of inputs to reach a specific object asked by the user. The dataset introduces a novel task in Embodied AI research and, in order to run the episode of the PInNED dataset, the Habitat simulator needs to be used. Instructions on how to run and instantiate the episodes of PInNED dataset are included in the public repository described in Sec. A.

Who created the dataset and on behalf of which entity? The dataset was created by researchers at the University of Modena and Reggio Emilia.

Who funded the creation of the dataset? Refer to the Acknowledgments and Disclosure of Funding section in the main paper.

I.b Composition

What do the instances that comprise the dataset represent? The PInNED dataset consists of generated navigation episodes designed to address the PIN task, accompanied by a list of object identifiers used in each episode within the Habitat simulator. As the dataset is composed of navigation episodes, containing all necessary information for the simulator to execute the task, no additional metadata is provided. However, an example of episode annotations is included in Listing A.

How many instances are there in total? The dataset of episodes for the PIN task is composed of a total of $865,519$ training episodes and $1,193$ validation episodes. Moving on to the objects contained in the PInNED dataset, the total number of unique object instances that are injected in the navigation environments is 338.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? While episodes of the PInNED dataset are generated procedurally by the authors of the paper, the objects used as additional objects are part of the objects released from Objaverse-XL dataset [22], which is composed of 3D models from different online sources such as GitHub, Thingiverse, Sketchfab, Polycam, and the Smithsonian Institution. The objects of PInNED are however restricted to 3D models included in Sketchfab.

What data does each instance consist of? The dataset content is defined by the information of the episodes for the PIN task. Each episode is represented as a dictionary containing the information needed by the Habitat simulator [61] to execute the task. A *.json* file including a list of the navigation episodes is produced for each scene included in HM3D dataset. We refer to Listing A for a sample of episode annotation. Each episode in the dataset specifies additional objects that are placed at a specific location loading *.glb* files containing the meshes of the objects. The *.glb* files used to instantiate the episodes of the PInNED dataset are downloadable from Objaverse-XL API using the Python script provided in the codebase. Each 3D object is associated with a *.json* file containing a dictionary with the information needed by the Habitat simulator to correctly instantiate the object in the environment in terms of size and appearance.

Is there a label or target associated with each instance? Each object used for the PInNED dataset is manually associated with an object category label to correctly perform the placement procedure of distractors belonging to the same category of the target instance, as well as computing metrics related to the PIN task. However, the object category label should not be used by the agent to tackle the PIN task. For each episode, only one instance is defined as the correct target to complete the task successfully.

Are there recommended data splits? The episodes of the PInNED dataset are divided into training and validation splits depending on the environment where the episodes are taking place. The environments are divided into training and validation splits following the environmental-level division performed by Ramakrishnan *et al.* [56]. Regarding the additional objects included in PInNED dataset, the object instances are divided into 266 training instances and 72 validation instances. It is worth noting that the sets of instances used for the training and validation splits do not overlap.

Are there any errors, sources of noise, or redundancies in the dataset? The additional objects on the surfaces of HM3D environments could be misplaced due to noise in the original annotations of the scene, or due to the presence of clutter at the acquisition time of the environment. Other sources of noise could be related to possible typos in the process of annotation of the textual descriptions of the additional objects.

Is the dataset self-contained, or does it link to or otherwise rely on external resources? The PInNED dataset relies on the scenes included in the HM3D dataset of 3D spaces and on the 3D object assets included in the Objaverse-XL dataset.

Does the dataset contain data that might be considered confidential? Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? No confidential or disturbing data is contained in the content of PInNED dataset.

I.c Collection Process

How was the data associated with each instance acquired? What mechanisms or procedures were used to collect the data? The additional objects used for the PInNED dataset are manually selected using the Python API from Objaverse-XL dataset.

The generation of the visual references of the target objects has been performed using Blender, where the 3D mesh of the object is rendered and captured in an isolated setting. The camera performs a 30-degree yaw rotation around the object to capture a favorable view of the objects. Then, each instance is rotated by 180 degrees in yaw to view its reverse side, while a 90-degree pitch rotation is used to observe the upper side of the object. This procedure produces three visual references for each target object.

The process of annotation of the textual descriptions of each object is performed by the authors of the paper. Two objects of the same object category are shown to each annotator that is required to describe one of the two instances in a way that is distinguishable from the other. The final procedure used three annotators, for a total of three textual descriptions for each object. Samples of the input references related to the objects of PInNED dataset are shown in Fig. C and Fig. D.

The episodes of PInNED are generated by spawning the selected additional objects in the scene after extracting all suitable surfaces from the semantic annotation of the environment. We refer to Sec. 3.3 for more details on object placement.

If the dataset is a sample from a larger set, what was the sampling strategy? The sampling strategy for selecting the objects from Objaverse-XL is based on a manual assessment of the photo-realistic properties of the selected objects and the corresponding visual appearance of the object when rendered using the Habitat simulator. The sampling strategy of the objects contained in the episodes of PInNED dataset is a random sampling. For each episode, a goal object category is selected, and a specific target instance is sampled from the set of suitable objects. Instances belonging to the same object category of the target object are sampled and positioned in the environment as distractors. If other spawnable surfaces are available, more distractors belonging to other object categories are placed in the environment. Details on the number of additional objects placed in the episodes of PInNED dataset are included in Sec. D. For the final generation of the episodes of PInNED dataset, 400 episodes are generated for each possible object category on the environments of the training split, while 2 episodes for each object category are generated in every environment of the validation split.

Who was involved in the data collection process? The actors performing the data collection and annotation process of the dataset are the authors of the paper.

Over what timeframe was the data collected? The dataset assets were collected and the episodes of the PInNED dataset were generated between November 2023 and May 2024.

Were any ethical review processes conducted? No ethical review process was necessary for the collection of the dataset.

I.d Preprocessing / Cleaning / Labeling

Was any preprocessing/cleaning/labeling of the data done? The objects used in the PInNED dataset are manually resized when rendered with the Habitat simulator adjusting their dimension compared with the surrounding environment to be similar to their real-world counterpart. Each 3D object is associated with a corresponding object category label to allow the usage of different instances of the same object category when tackling the PIN task. The episodes of PInNED dataset are, instead, validated using the Habitat simulator to remove any episode containing objects that are not reachable from the starting position of the agent.

I.e Uses

Has the dataset been used for any tasks already? The PInNED dataset can be used to train and evaluate agents for the Personalized Instance-based Navigation (PIN) task. See Sec. 3 and Sec. 5 for more details on the task definition and the experimental evaluation.

What (other) tasks could the dataset be used for? The dataset could be used for other tasks involving recognition or manipulation on specific instances using visual or textual references as input.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? Users need to follow and respect the licenses associated with the additional 3D objects and the episodes contained in this dataset.

I.f Distribution

How will the dataset will be distributed? The dataset is made public through the release of a public GitHub repository. The repository containing dataset and codebase is released at this url: https://github.com/aimagelab/pin.

When will the dataset be distributed? The dataset has been publicly released on October 2024.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? The dataset and the object annotations are released under the CC BY license. The codebase is released under the MIT license. The additional objects contained in the episodes of PInNED dataset are subject to the licenses that they are released under.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? Any restrictions are related to additional objects and to the licenses which they are released under. Users need to assess license questions based on their use.

I.g Maintenance

Who will be supporting/hosting/maintaining the dataset? The dataset will be maintained by the authors of the paper who commit to maintaining the dataset long-term.

How can the owner/curator/manager of the dataset be contacted? The authors can be contacted at `{firstname.lastname}@unimore.it`.

Will the dataset be updated? A potential future update could involve extending the dataset to include a test split, upon receiving permission from the HM3D dataset owners to access the environments in the test split.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Users are free to extend the dataset at the condition of following and respecting the licenses associated with the dataset and associated additional objects by contacting the authors on the public repository.