# Select2Plan: Training-Free ICL-Based Planning through VQA and Memory Retrieval

Davide Buoso[1], Luke Robinson[2], Giuseppe Averta[1], Philip Torr[2], Tim Franzmeyer[2], Daniele De Martini[2]
[1]Polytechnic University of Turin (Italy) [2]University of Oxford (UK)

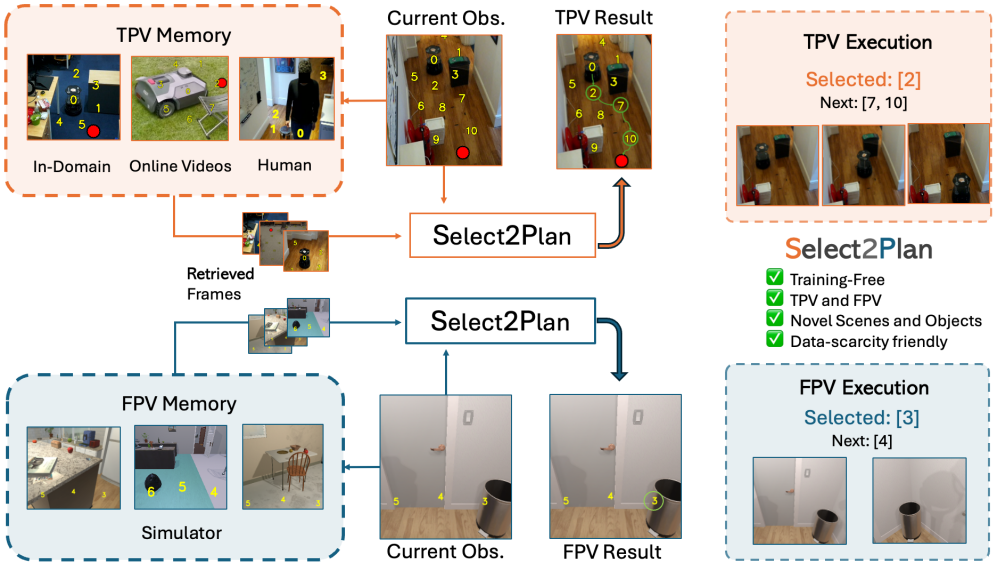


Fig. 1: Select2Plan is a training-free approach for high-level robot planning that can be applied to both Third-Person View (TPV) or First Person View (FPV) scenarios. It queries memory banks built from previous experiences and/or in-the-wild data to retrieve relevant annotated situation-aware samples. Then, given the current observation and the retrieved samples, Select2Plan takes advantage of In-Context Learning to select a sequence of visual candidates in the form of text. These are then mapped to robot actions and directly executed.

***Abstract*— We introduce Select2Plan (S2P), a novel training-free framework for high-level robot planning that leverages off-the-shelf Vision-Language Models (VLMs) for autonomous navigation. Unlike most learning-based approaches that require extensive task-specific training and large-scale data collection, S2P overcomes the need for fine-tuning by adapting inputs to align with the VLM's pretraining data. Our method achieves this through a combination of structured Visual Question Answering (VQA) to ground action selection on the image, and In-Context Learning (ICL) to exploit knowledge drawn from relevant examples from a memory bank of (visually) annotated data, which can include diverse, in-the-wild sources. We demonstrate S2P flexibility by evaluating it in both First-Person View (FPV) and Third-Person View (TPV) navigation. S2P improves the performance of a baseline VLM by 40% in TPV and surpasses end-to-end trained models by approximately 24% in FPV when tasked with navigating towards unseen objects in novel scenes. These results highlight the adaptability, simplicity, and effectiveness of our training-free approach, demonstrating that the use of pre-trained VLMs with structured memory retrieval enables robust high-level robot planning without costly task-specific training. Our experiments also show that retrieving samples from heterogeneous data sources, including online videos of different robots or humans walking, is highly beneficial for navigation. Notably, our method effectively generalizes to novel scenarios, requiring only a handful of demonstrations. Project Page: lambdavi.github.io/select2plan**

## I. Introduction

Vehicle path planning is a long-standing problem in robotics, traditionally addressed using model-based or Reinforcement Learning (RL) approaches [1], [2], [3]. However, methods that directly learn from experience often struggle when faced with unfamiliar or ambiguous scenarios. Interestingly, recent research has shown that Large Language Models (LLMs) and Vision-Language Models (VLMs), demonstrate surprising reasoning capabilities that can be adapted to propose robot paths in arbitrary scenes [4]. Indeed, these models excel at incorporating common-sense reasoning acquired during their long pretraining phase [5]. This ability is crucial in robotics operations, where the deployment scenario rarely aligns perfectly with the training dataset [6], [7]. While methods like LoRA [8] reduce the computational cost of fine-tuning LLMs and VLMs, they still require domain-specific data, which can be costly to obtain. In parallel, In-Context Learning (ICL) and Retrieval-Augmented Generation (RAG) have shown promising results in scoping the ability of LLMs *at deployment time* with no additional fine-tuning, mitigating these costs.

Our framework – Select2Plan (S2P) – combines Visual

Question-Answering (VQA) and ICL with VLMs in a training-free manner, showing remarkable flexibility across various scenes, contexts, and setups.

More specifically, we formulate the planning problem as a VQA task using visual prompting. Inspired by [9], [10], we generate a set of position candidates in the image space and use them as part of a query mechanism to a VLM, to extract the next robot move. We combine this approach with ICL to enhance the model's reliability: we retrieve similar successful samples and use them, along with the current annotated image, as context to support the model's generalization. In this way, we can generate a robust path, which can span multiple planning steps within a single response, in contrast to the iterative approach taken by [9]. Figure 1 presents the overview of our framework. Given that different users could have different setups and needs, we aim to create a framework that can adapt to two different navigation scenarios. The first is a more traditional First-Person View (FPV), where the robot is equipped with a monocular camera and needs to reach specific objects in the scene. The challenge in this case is the sensor's limited view, as the goal object might get out of view while the robot navigates the environment. As a second test-bench, we consider a robot controlled through eye-to-hand visual servoing, as in [11], [12]. Here, the camera is not physically attached to the robot, and the far viewpoint inherently limits the depth [13] and spatial resolution. However, given the widespread use of CCTV cameras, we believe this approach offers new opportunities and, interestingly, also mirrors the training data of VLMs – static RGB images paired with textual descriptions – making these models well-suited for tasks involving external camera navigation. We show how our setup can flexibly adapt to both visual inputs and diverse sources of context, such as videos from the Internet or even human traversal of the scenario.

To summarize, our main contributions are:

1) A framework for planning and navigation using only RGB data, using structured VQA, ICL and retrieval techniques to reduce the task-related data needed to a handful of episodes.
2) S2P can be deployed on multiple navigation paradigms, namely First-Person View (FPV) autonomous navigation and Third-Person View (TPV) infrastructure-driven planning.
3) We extensively test S2P in both scenarios, and present an analysis of the impact of different sources of in-context examples on the system's overall performance.

Our empirical analysis demonstrates that our approach improves the navigational abilities of VLMs without requiring further training, laying the foundation for more sophisticated and flexible planning in autonomous systems. To the best of our knowledge, our approach is the first that can seamlessly adapt to multiple setups *and* utilize multiple sources of in-context samples.

## II. Related Work

Our approach is at the intersection of robot navigation, planning, LLMs, VLMs, and ICL.

Traditional methods for robot navigation tasks, such as Object Navigation and Visual Navigation, have historically relied on RL to train policies for complex tasks. Works like those of [1], [2], [3] employed deep RL models to learn navigation policies based on visual input [14], [15], [16]. [17] embeds semantic relationships into RL frameworks, aiding object search efficiency. More recently, transformer-based models have emerged as a powerful alternative, often yielding better generalization due to their capacity to model long-range dependencies in data. [18] leverages visual transformers to model object and spatial cues for action prediction while [19] further adds dynamic knowledge graphs. Lately VLMs have shown great promise for high-level decision-making in robotics, as they integrate visual perception with language-based reasoning. [20], [4], [21], [22] demonstrated how a pre-trained LLM can be utilized in zero-shot settings to control robots. Recent works, such as [23], have highlighted the effectiveness of ICL integrated with memory-based retrieval for robotics applications. However, these methods typically focus on a specific setup and require several additional trained modules or advanced techniques to extract useful plans.

A more recent line of work introduces Visual Prompting as a paradigm for planning [9], [10]. The first notable difference lies in the core methodology: [9] adapts its VLM by generating a dense set of visual points on the image and iteratively refining the selection. In contrast, our method produces fewer points following navigation-specific patterns and generates a multi-step plan in a single forward pass which supports coherent plan-following. [10], on the other hand, employs a secondary (smaller) VLM to evaluate the behavior of the main model's output. While their method emphasizes navigation aligned with human-acceptable behaviors, our work centers on reasoning-based tasks, such as locating objects through environmental understanding and semantic associations from visual inputs, while simultaneously avoiding obstacles. Unlike these prior methods, our approach also incorporates additional contextual information, which facilitates adaptation in data-scarce scenarios and improves the performance of raw VLMs. This flexibility enables our framework to generalize across both First-Person View (FPV) and Third-Person View (TPV) settings without the need for specialized sensors or extensive fine-tuning. Consequently, our model can efficiently generate navigation plans from image-text pairs, making it more adaptable to diverse scenarios, goals, and setups compared to existing approaches.

## III. Methodology

The framework takes as input an RGB image coming from either the on-board camera or the external camera, depending on the setting chosen. The camera frame is used to retrieve relevant episodes from the Experiential Memory which is passed to the VLM along the current annotated frame and the additional info from the Episodic Memory (optional). Once the VLM produces the plan in form of text, the controller maps it to actual control commands. In this section we present each module of the framework (presented

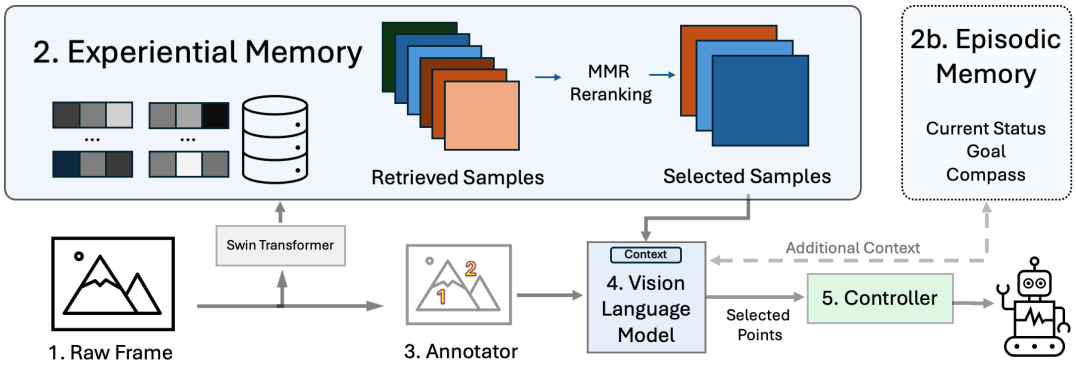


Fig. 2: Overview of the proposed framework. The framework takes as input the live image (1) – from the onboard or CCTV camera – and retrieves relevant experience from an experiential memory in the form of annotated samples (2). The raw frame is then annotated (3) and passed, together with the sampled images and an optional episodic memory (2b), which is composed by current view, last action chosen, navigation plan and compass status, to the VLM to generate the next commands and explanations (4). The current information of scene and the new navigation plan is then added in the episodic memory. The plan in form of text is elaborated by the controller and executed (5).

in Figure 2) individually.

**Memory Retrieval (In-Context Learning)** To provide context, we use an *History-Injection* ICL approach where a fictitious chat conversation is created and in the chat, K episodes are retrieved from a memory database and are split in query (annotated image and prompt) and answer. Here the multi-turn conversation is injected into the model history as turns of the conversation.

Finally, we explicitly ask the model, based on its previous (correct) responses, to provide a plan for the live observation. We incorporate two different types of memory in our framework: Experiential Memory, representing long-term memory and Episodic Memory, useful for short-term navigation.

**Experiential Memory**: The Experiential Memory is a collection of annotated images that is used as context to condition the answer of the VLM via ICL. These experiences can be gathered in different ways, and we discuss different combinations in our result section, where we evaluate the use of data coming from the same or different environments, human demonstrations, or even directly using online footage, processed in the form of annotated frames.

**Episodic Memory**: We define as Episodic Memory, all the information related to the current episode that can be useful for informed navigation. We incorporate a simplified text-based representation of the scene into the context to keep track of current short-term objectives, such as the view angle, the Compass Module (presented in III-A) and future commands to follow robustly the navigation plan over time. For instance, if the VLM suggests to look on the table when seeking for a bowl, the framework should keep track of this initial plan if during the trajectory the table becomes no longer visible (e.g. because the robot is avoiding an obstacle).

**Sampler**: The sampler aims to select the most appropriate samples from the Experiential Memory to be presented to the VLM as context. We feed a Swin Transformer [24] with the live camera image to recover similar situations from the Experiential Memory. We represent the image – and the Experiential Memory samples – as the average output of the last hidden layer and obtain a feature vector which we employ as the query. Empirically, we observed that building a diverse context (both similar and different situations) benefits the model's generalisation to the current situation. To balance out the similarity, we incorporated a re-ranking algorithm adapted to our framework into the retrieving process. Hence, we employ Maximal Marginal Relevance (MMR) [25], which aims to reduce redundancy and increase sample diversity according to a combined criterion of query relevance and novelty of information. MMR is defined as:

$$\texttt{MMR}(Q, M, C) = \arg \max_{s_i \in M \setminus C} \Big[ \lambda < s_i, Q > + \\ - (1 - \lambda) \max_{s_j \in C} < s_i, s_j > \Big] \quad (1)$$

where $Q$ represents the query image embedding and $M$ the experiential memory. This algorithm operates by iteratively selecting images (samples $s_i \in M$), to add to the context $C$, based on a trade-off between two factors: the image's relevance – similarity – to the query and the image's dissimilarity from the samples that have already been chosen. The goal is to ensure that each selected image adds new, informative content rather than repeating information. The scalar $\lambda$ balances this trade-off.

**Annotator**:

Before presenting the live and episodic images to the VLM, we follow the approach of [9] and annotate them with the possible actions the model can choose from. In this previous work, the authors impose on the image numbers embedded in white circles in random positions. These numbers are connected to the current position of the robot through lines. To let the model attend to most of the original picture without obstructions, we opted for a minimalistic annotation which only marginally impacts the original image, using thin bright yellow numbers (which is a rare color to find in common daily living environments), as depicted in Figure 1. These annotations are setup-specific and appear as numerical values superimposed on the image frames. For the TPV setting, annotations are overlaid on a grid, excluding irrelevant parts of the image. In the FPV setting,

they are designed to resemble a video-game interface. Further details on the annotator are provided in the setup-specific subsections of this section. More details are presented in sections III-A and III-B.

**Prompt Templating Engine**: The prompt template engine is the core of our prompt engineering step. It combines information from the Experiential and Episodic Memories with the live, annotated image and the task at hand in a digestible form for the VLM. The prompt contains information about the type of data the VLM is presented – Experiential, Episodic and live – how to decode the annotations into *a sequence* of actions and the request to fulfil the navigation task. We also instruct the VLM to use a JSON format for its output to easily integrate it into any control pipeline already deployed on the robotic platform.

**VLM** The VLM is prompted to select a discrete action from the annotated frame, represented by numerical values, and explain its decision. Depending on the deployment platform, the model may also produce additional outputs, such as identifying actions that could lead to dangerous locations or recognizing objects in the scene.

**Controller** Finally, the low-level, platform-specific controller executes the selected action on the platform, interpreting the VLM's output and transmitting it to the robot. In the case of the TPV scenario, the positions of the annotations chosen by the VLM are used as vertices in a piecewise linear path. The robot is then guided along this path using a PD controller measuring cross-track and heading error as demonstrated in [12]. The commands are then transmitted via Wi-Fi to the rover which has no other sensors on-board. In the FPV scenario, the controller is embedded into the environment. It moves the agent to the target cell in the map grid, which is identified based on the input command, as described in [26].

### A. FPV Setting

We designed our framework considering the robot's and camera's movements as discrete actions. We enable the VLM to interpret visual annotations that resemble a control overlay inspired by video-game interfaces. We display the numbers 1 to 7 on a semicircle at the bottom of the image, providing rotational control, where 4 is the neutral `MOVE_FORWARD` and the others numbers represent various degrees of rotation.

Additionally, the model can select `LOOK_UP` and `LOOK_DOWN` commands, associated with the non-displayed numbers 8 and 9. Lastly, number 0 is associated with `DONE` command to end the episode.

Following a structured procedure, we build the Experiential Memory by manually navigating the robot in the AI2-THOR environment. At each timestamp, the operator selects an appropriate action based on the current visual context – i.e. a command number – and provides a natural language explanation, simulating a "think-aloud" process. For instance, upon observing a microwave on the left, the explanation would state: "A microwave is visible on the left, so the system will steer slightly to the left." The annotated image, the selected command, and the corresponding explanation are inserted in the Experiential Memory. This process aims to imitate human-like physical movements and capture the underlying thought processes that drive these actions.

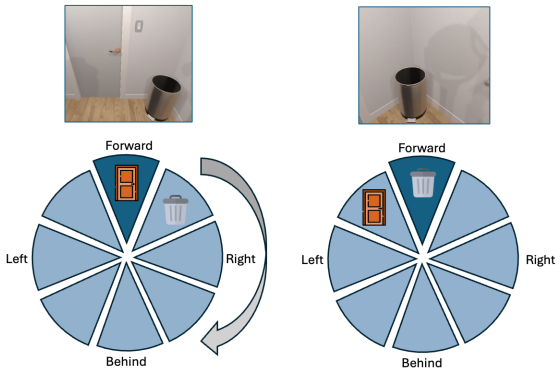


Fig. 3: The figure depicts a scenario where the agent uses the compass. The compass keeps track of the scene content as the robot rotates, recording insightful information about the room's layout. For instance, if the agent is looking for a *bin*, it will likely rotate towards where it last saw a *bin* or to something semantically related to it, although it is now out of sight. In this example it rotates right (grey arrow).

During execution, we request the VLM also a text description of the environment – the list of objects in the frame. Based on this description, we create a "circular compass", as shown in Fig. 3, which rotates along with the agent's rotations and can inform the model's decisions since certain objects can be found near affine items or go out of the field of view due to motion. This is then saved in the Episodic Memory. We also incorporate the last executed action, the current vertical view status, and the previous command list. At each timestamp, the VLM is prompted not only to determine the current action but also to predict the next one, ensuring a more robust adherence to the ongoing navigation strategy (Figure 2).

### B. TPV Setting

We follow the setup of [11], [12] in considering a visual servoing system [27], [28], where a robot is controlled by the cloud via cameras installed in the infrastructure. Here, security cameras capture raw frames of the environment, which are then annotated and passed, together with samples from the Experiential Memory, to the VLM to predict the next few trajectory points for the robot to execute. We start the annotation process of the live frame by identifying the robot's position [11], through a YOLO model and place there the number 0 to make the task embodiment-agnostic. During the system's initial setup, a segmentation mask of the scene's floor is taken through Segment Anything [29] and saved as a binary mask. Importantly, this step is performed *only once* during the framework's first setup, so it does not handle new objects or obstacles appearing in the scene. Starting from the 0 position, we create concentric circles of numbers equally spaced and with increasing radius while using the mask, we filter out locations that are not traversable. Finally, we ask the VLM to choose a sequence of points from the robot to the end goal, which can be given as an object in the scene or a red circle – clear of obstacles.

An optional step is to crop the image to comprehend only the labels and the target to optimise the image size passed to the model, reducing costs and increasing focus on the important part of the picture.

After receiving the sequence of points, we map back the numbers to coordinates in image space and follow the path generated through a PD controller [12]. During inference, the sequence produced is at most 3 or 4 points long but once 1 or 2 points are correctly tracked, the new sequence substitutes the old one.

## IV. Experimental Setup

We preliminarily explored open-source and closed-source VLMs; we empirically chose Gemini 1.5 Pro [30] for its accessibility, performance and large context-window (`gemini-1.5-pro-001`). In fact, its large context-window, VQA capabilities and pre-training on videos make it an optimal choice as the framework's backbone. We hypothesize that models trained on large video corpora could be more suitable to this task, considering the need to reason about dynamics of the robot in the scene. While Gemini 1.5 Pro has been selected as the main VLM for S2P, our framework remains fully modular, leaving room to future works to explore different models without changing the rest of the pipeline. The experimental setup differs slightly for FPV and TPV. In the following, we will discuss both of them and describe the metrics, data collection and baselines we will use to evaluate our system. In both cases, we experimented with different parameters – from the dots color, placement and number, to the memory examples. Here, we report the results of our best combination only.

### A. First Person View

Among the several simulators proposed to facilitate the Embodied Navigation tasks, we selected AI2-THOR's ObjectNav task [26], whose goal is to navigate towards a predefined target object.

We follow the evaluation procedure of [19], using the same metrics and setup, with different object classes for training and testing[1]. We hence compare our approach to [19] [17], [31], [18], importantly we are training-free, so we don't fine-tune on any navigation data, but we rather record and store 23 episodes manually, one per object in the known category, with two challenging classes (*KeyChain* and *WateringCan*) receiving an additional demonstration for their size and sometimes ambiguous location in the scene. These approaches are as different as possible, ranging from models with strong priors about the scene to graph-based transformers trained with Reinforcement Learning. Due to resource constraints, we evaluate our framework on 300 episodes per scene type, and limit the maximum number of steps per episode to 25. This source of data is extremely limited purposefully in order to demonstrate that even few episodes can establish an effective starting point for our framework, thereby challenging the generalization capabilities of trained models. This constitutes a human-like sub-optimal ground truth, which composes the system's Experiential Memory and helps the model generate situation-grounded strategies.

### B. Third Person View

We evaluate our approach on a custom dataset recorded in four indoor environments, where we collected expert trajectories by teleoperating a TurtleBot3 rover. Example tasks can be seen in Figure 5, where the robot has to navigate towards the camera. The trajectories recorded encompass a range of difficulty levels, from simple paths with minimal obstacles to more complex routes, including objects that takes up most of the traversable floor, representing real-world navigation challenges. We annotate – see Section III-B – the images and manually mark the labels overlapping with obstacles or non-traversable locations as *dangerous*. We compare the results of our model against a zero-shot approach on this dataset. We test different Experiential Memories, composed of scenes from the same or different cameras, called scenarios `A` and `D`, respectively – see Fig. 4. In addition, we will show results with trajectories performed by a human in the same scenes – scenario `H` – simulating the images usually captured by security cameras; this would demonstrate how a person move in an office room, avoiding obstacles such as chairs, boxes, etc. Finally, we also use short clips of robots from the web (see [2]) – scenario `O` – to validate how general and different from the target scenario the samples in the context can be, while still allowing the model to understand the task and mimic it successfully. The main metric to evaluate our system, in the TPV scenario, is the Trajectory Score (TS), defined as:

$$\mathrm{TS} = \sum_{i=0}^{N} S_i \frac{Pc_i}{\max(\mathrm{len}(P_i), \mathrm{len}(G_i))} \qquad TS_i \in [0,1] \quad (2)$$

where $i$ refers to a specific episode, $TS_i$ to the score (singular value inside the sum), $Pc_i$ to the number of correctly predicted points, $P_i$ to the predicted sequence and $G_i$ to the ground truth sequence. Finally $S_i$ indicates whether selected point is safe as a binary value. The maximum value of this trajectory score is $N$, in our case 300. The purpose of this metric is to measure to which extent the model is able to reproduce human-like navigation. Moreover, we measure the number of dangerous points selected during the evaluation process.

$$D = \sum_{i=0}^{N} D_i \qquad D_i \in \{0,1\} \quad (3)$$

where $D_i$ indicates if at episode $i$ a dangerous point has been selected or crossed, simulating a collision.

## V. Experimental Results

In the FPV scenario, given the scarcity of training-free methods, we directly compare our model against end-to-end trained approaches, aiming to highlight its performance relative to traditional and well-established methods. We

[1][19] chose as known objects: *HousePlant, Sink, TableTop, Knife, Fridge, Bowl, Cabinet, Cloth, KeyChain, WateringCan, Bed, Lamp, Book, Chair, LightSwitch, Candle, Painting, Watch, Cabinet, Toilet, SprayBottle*. As test objects: *Toaster, Microwave, Television, LapTop, RemoteControl, CellPhone, Mirror, AlarmClock, Toiletpaper, SoapBottle*.

[2]https://www.youtube.com/watch?v=FgftdWrSYzM, https://www.youtube.com/watch?v=hL4MTG0u1K0


Fig. 4: Experiential Memories for Third-Person View (TPV): `D` includes experiences from the same environment excluding the inference room, `O` from online videos and `H` from the same environment but with a human as navigator instead of a robot.

| Eval. set | Method | SR ‖ SPL % | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Kitchen | | Living room | | Bedroom | | Bathroom | | Average | |
| Known scenes & Known objects | Random policy | 10.40 | 4.80 | 11.47 | 3.40 | 13.07 | 8.20 | 21.60 | 11.13 | 14.13 | 6.88 |
| | Scene priors | 50.67 | 34.27 | 67.07 | 29.43 | 69.07 | 22.47 | 71.33 | 28.73 | 64.53 | 28.73 |
| | SAVN | 46.27 | 38.27 | 56.93 | 40.60 | 74.67 | 35.47 | 81.87 | 46.53 | 64.93 | 40.22 |
| | VTNET | 57.60 | 43.20 | 68.53 | 45.03 | 86.53 | 38.53 | 76.13 | 52.30 | 72.20 | 44.77 |
| | GVSN | **63.20** | **50.27** | **88.00** | **52.43** | **94.27** | **59.07** | **89.47** | **66.37** | **83.73** | **57.03** |
| | S2P (Ours) | 58.00 | 31.05 | 44.00 | 19.00 | 58.00 | 25.29 | 64.85 | 36.70 | 56.21 | 28.01 |
| Known scenes & Novel objects | Random policy | 2.93 | 0.66 | 5.60 | 0.93 | 8.80 | 2.53 | 8.13 | 2.03 | 6.37 | 1.54 |
| | Scene priors | 21.07 | 14.55 | 23.20 | 9.40 | 19.47 | 12.17 | 30.53 | 18.47 | 23.57 | 13.65 |
| | SAVN | 17.27 | 8.30 | 27.20 | 7.60 | 37.87 | 20.47 | 32.53 | 16.50 | 28.72 | 13.22 |
| | VTNET | 26.53 | 12.27 | 49.07 | 23.97 | 35.87 | 18.63 | 36.67 | 22.53 | 37.03 | 19.35 |
| | GVSN | 32.67 | **20.13** | **58.53** | **32.50** | **53.07** | 20.97 | 49.07 | 25.60 | 48.33 | **24.80** |
| | S2P (Ours) | **42.33** | 19.01 | 36.00 | 18.70 | 53.00 | **32.44** | **65.00** | **28.51** | **49.08** | 24.66 |
| Novel scenes & Known objects | Random policy | 6.00 | 0.87 | 4.20 | 1.27 | 3.47 | 0.37 | 4.67 | 1.30 | 4.58 | 0.95 |
| | Scene priors | 11.60 | 6.23 | 13.87 | 8.27 | 17.20 | 10.83 | 15.07 | 8.40 | 14.43 | 8.43 |
| | SAVN | 26.80 | 10.70 | 31.33 | 6.00 | 43.87 | 15.63 | 21.73 | 8.33 | 30.93 | 10.17 |
| | VTNET | 35.73 | 12.30 | 40.93 | 13.93 | 57.87 | 17.83 | 47.60 | 10.73 | 45.53 | 13.70 |
| | GVSN | 44.13 | 18.50 | 48.00 | **27.53** | **68.67** | **19.50** | 60.53 | 26.07 | 55.33 | 22.90 |
| | S2P (Ours) | **63.30** | **32.40** | **49.00** | 17.85 | 40.00 | 12.79 | **79.00** | **34.94** | **57.83** | **24.50** |
| Novel scenes & Novel objects | Random policy | 1.60 | 0.43 | 3.87 | 0.80 | 3.60 | 0.33 | 2.27 | 0.93 | 2.83 | 0.63 |
| | Scene priors | 2.93 | 1.13 | 10.80 | 3.13 | 23.07 | 7.60 | 15.87 | 6.37 | 13.17 | 4.56 |
| | SAVN | 17.33 | 5.97 | 13.60 | 4.50 | 25.47 | 5.50 | 12.27 | 4.37 | 17.17 | 5.08 |
| | VTNET | 26.67 | 7.03 | 19.47 | 9.03 | 16.93 | 7.40 | 16.93 | 7.40 | 22.43 | 6.79 |
| | GVSN | 34.40 | 7.30 | 17.87 | 8.10 | 32.40 | 9.60 | 30.00 | 6.57 | 28.67 | 7.89 |
| | S2P (Ours) | **61.83** | **40.75** | **43.00** | **21.61** | **37.00** | **17.48** | **68.00** | **34.65** | **52.46** | **28.62** |

TABLE I: Comparison of Success Rate (SR) and Success weighted by Path Length (SPL) across different environments (kitchen, living room, bedroom, and bathroom). The table presents results for different evaluation settings: (1) Known scenes and Known objects, (2) Known scenes and Novel objects, (3) Novel scenes and Known objects, and (4) Novel scenes and Novel objects. Various training-based methods are compared to the proposed approach, S2P, which is training-free. Please note that for (1) the comparison is biased as the presented baselines are trained on variations of the scenario.

structured the result table in two sections in Table I. The *Known Scenes and Known Objects* setting, while included for completeness, presents an inherently asymmetrical comparison that highlights our framework's design philosophy. The baselines from [19] learn from millions of episodes, while S2P deliberately uses a memory bank composed of as few as 23 demonstrations. This intentional choice prioritizes practicality of operation over extensive data collection. Consequently, in these 'known' scenes, S2P's performance reflects its sensitivity to the specific, and often sub-optimal, trajectories from the few retrieved demonstrations. ICL, by nature, aims to mimic these examples, and when the memory bank is sparse, this can lead to less efficient paths if the retrieved examples aren't perfectly optimal. This explains the "contextual conflict" observed in the *Known scenes & Novel objects* setting, which arises when retrieval correctly identifies visually similar scenes, but the ICL module then processes trajectories for task-irrelevant objects within those scenes, necessitating more robust reasoning beyond direct mimicry. Furthermore, S2P operates under a strict 25-step episode limit, a deliberate choice to manage VLM inference costs and ensure real-time applicability. This contrasts with baselines that benefit from up to 200 steps. This constraint naturally impacts performance in larger, more complex environments, such as the living room, where baselines have ample steps to recover from exploratory moves, whereas S2P must find efficient paths quickly. However, our focus is on demonstrating effectiveness in scenarios requiring generalization. These novel scenarios force the model to move beyond mimicking sparse trajectories and instead leverage the VLM's intrinsic, pre-trained reasoning about object semantics and spatial relationships. This is where our approach excels, achieving remarkable improvements of 23.79% in Success Rate (SR) and approximately 20% in Success weighted by Path Length (SPL) in the most challenging setting, surpassing the best-performing trained

| Components | Success Rate % (Avg) |
|---|---|
| Baseline (CoT [32]) | 29.40 |
| + VQA | 37.30 |
| + VQA + ICL | 42.70 |
| + VQA + ICL + Compass | 52.45 |

TABLE II: Ablation study on framework components in the scenario of Novel Scenes and Novel Objects. The value reported is the Success Rate averaged on 50 runs.

| Mode | CL | Scenario | TS (/300) ↑ | D (/300)↓ |
|---|---|---|---|---|
| Baseline (CoT [32]) | 0 | - | 147.82 | 76 |
| S2P (Ours) | 10 | A | 270.70 | 2 |
| S2P (Ours) | 10 | D | 247.24 | 8 |
| S2P (Ours) | 10 | H | 219.58 | 13 |
| S2P (Ours) | 10 | O | 235.83 | 16 |

TABLE III: TPV results: comparison between the zero-shot and our framework on the same scenario but using different types of context sources (see Figure 4).

model. Additionally, on average our framework outperforms the baselines in all the settings requiring generalization. Our approach reduces the data required for S2P's operation to approximately 0.005% of the data needed by [19] for training. These results validate the effectiveness of integrating ICL and VLMs for navigation tasks, demonstrating that it is possible to construct a competitive framework that surpasses extensively trained models in novel scenarios while requiring only a handful of demonstrations and without additional training needed.

We also present an ablation study on the different components of our framework in Table II, where we exploit CoT with Gemini 1.5 Pro. The goal of the study is twofold: firstly it highlights the significance of each component in the system's overall performance and secondly presents how training-free VLMs performs in the task. Initially, we ground the question in the visual domain, aligning with the OCR and VQA training paradigms of VLMs, while also directly associating spatial commands with the image. Furthermore, we confirm the critical role of ICL, which enhances the model's reasoning capabilities by providing practical examples. Lastly, our Experiential memory plays a crucial role in exploring the environment efficiently.

Our framework in the TPV scenario achieved a maximum

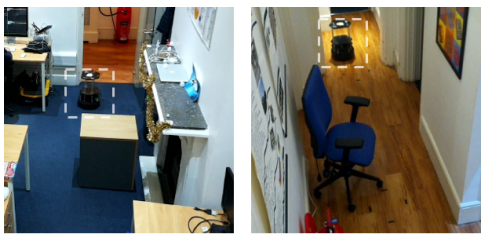

Fig. 5: Examples of TPV tasks. In the TPV scenario, we challenge our planner by introducing obstacles of various types (e.g., a large box or an office chair) along the robot's path. In these examples, the robot, highlighted by the white bounding box, have to navigate towards the camera despite these obstacles.

Trajectory Score of 270.70 out of 300.00 using context scenario A, which allowed unrestricted retrieval from the database. To make the evaluation goal-agnostic, we considered as end-goal a red circle, superimposed on the picture and report results averaging three consecutive runs. The results are presented in III. Our framework significantly outperformed the zero-shot approach – by approximately 40% – demonstrating the effectiveness of our framework when supported by full context. Moreover, the model exhibited a 24% reduction in the selection of dangerous points, which would otherwise lead to potential collisions. This reduction is a critical enhancement, as it directly correlates with safer navigation, an essential factor for real-world autonomous applications. The model's capability to avoid hazardous locations was evident in scenario A but improved also across other contexts, including scenarios from online videos and human-driven trajectories. This versatility underscores the model's adaptability and ability to generalize across varying sources of contextual information, demonstrating its robustness in understanding the task at hand. These findings are significant because they illustrate that effective navigation can be achieved with minimal data collection, or even none, leveraging online data, significantly reducing the costs and time associated with data gathering and model training, making it a quick, yet effective, solution for real-world deployment.

## VI. Limitations & Safety Considerations

Our framework lacks of explicit exploration or active learning mechanisms, which could be beneficial in navigation tasks. Inference latency is another concern, especially in high-stakes applications requiring rapid decision-making. A complete planning step producing the next 2-3 moves, takes indeed from 2 to 2.8 seconds using Gemini APIs, based on client location and network properties. For memory retrieval on Arm-based CPU the model loading and first forward pass take 0.5 seconds and 0.03 seconds from that point on. However, even though inference time is higher due to reliance on large-scale VLMs, this trade-off is justified by the elimination of extensive training requirements and can be further optimized by choosing efficient and lightweight models. From a safety perspective, our approach inherits biases and limitations from pre-trained VLMs, potentially leading to errors in ambiguous environments. Lastly, even if S2P is theoretically subject to hallucinations, due to the auto-regressive decoder, ICL drastically reduces the drifts from the optimal response.

## VII. Conclusions and Future Works

In this work, we introduced a training-free navigation framework that leverages In-Context Learning (ICL) and Vision-Language Models (VLMs) to achieve effective object navigation with minimal demonstrations, using VQA grounding. Unlike traditional approaches that require extensive training on millions of episodes, our method achieves strong generalization across both novel scenes and unseen object categories while drastically reducing data requirements. While training on vast amounts of data in a given

environment can yield strong performance, our framework offers a compelling alternative in cases where such training is infeasible by leveraging the pre-existing world knowledge of VLMs. Through our experiments, we demonstrated that S2P consistently outperforms traditional trained models in generalization tasks. These results validate that pre-trained VLMs can be effectively repurposed for navigation tasks, significantly reducing the cost and effort of large-scale dataset collection. Moreover, we showed that our approach remains effective across different setups, including first-person and third-person perspectives, further emphasizing its adaptability to real-world scenarios and to users' specific needs. While our framework eliminates the need for extensive training, we acknowledge that inference speed and reliance on high-quality visual inputs remain areas for improvement. Future work will focus on optimizing VLM inference for real-time applications and enhancing robustness to visual noise and dynamic environments. Additionally, integrating lightweight strategies directly in the controller could offer a safe and low-cost option to improve the framework.

The results highlight that our approach to training-free navigation is simple yet effective, especially when the key to success is adapting to novel situations in a data-scarcity regime.

## ACKNOWLEDGMENTS

This work was supported by EPSRC Impact Acceleration Account (IAA) "Robotics Inversion".

This study was carried out also within the Future Artificial Intelligence Research (FAIR) and received funding from the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.3 – D.D. 1555 11/10/2022, PE00000013).

Davide Buoso is the corresponding author (davide.buoso@polito.it).

## REFERENCES

[1] A. Staroverov, D. A. Yudin, I. Belkin, V. Adeshkin, Y. K. Solomentsev, and A. I. Panov, "Real-time object navigation with deep neural networks and hierarchical reinforcement learning," IEEE Access, vol. 8, pp. 195 608–195 621, 2020.

[2] K. Zhou, C. Guo, and H. Zhang, "Visual navigation via reinforcement learning and relational reasoning," in 2021 IEEE SmartWorld/SCALCOM/UIC/ATC/IOP/SCI, 2021, pp. 131–138.

[3] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in ICRA. IEEE, 2023, pp. 7226–7233.

[4] Q. Zeng, Q. Yang, S. Dong, H. Du, L. Zheng, F. Xu, and Y. Li, "Perceive, reflect, and plan: Designing llm agent for goal-directed city navigation without instructions," 2024.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever et al., "Language models are unsupervised multitask learners," OpenAI blog, vol. 1, no. 8, p. 9, 2019.

[6] D. S. Williams, M. Gadd, P. Newman, and D. De Martini, "Masked $\gamma$-ssl: learning uncertainty estimation via masked image modeling," in ICRA. IEEE, 2024, pp. 16 192–16 198.

[7] D. S. W. Williams, D. D. Martini, M. Gadd, and P. Newman, "Mitigating distributional shift in semantic segmentation via uncertainty estimation from unlabeled data," T-RO, vol. 40, pp. 3146–3165, 2024.

[8] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen et al., "Lora: Low-rank adaptation of large language models." ICLR, vol. 1, no. 2, p. 3, 2022.

[9] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu et al., "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," in ICML. PMLR, 2024, pp. 37 321–37 341.

[10] A. J. Sathyamoorthy, K. Weerakoon, M. Elnoor, A. Zore, B. Ichter, F. Xia, J. Tan, W. Yu, and D. Manocha, "Convoi: Context-aware navigation using vision language models in outdoor and indoor environments," in IROS. IEEE, 2024, pp. 13 837–13 844.

[11] L. Robinson, M. Gadd, P. Newman, and D. D. Martini, "Robot-relay: Building-wide, calibration-less visual servoing with learned sensor handover networks," in ISER. Springer, 2023, pp. 129–140.

[12] L. Robinson, D. De Martini, M. Gadd, and P. Newman, "Visual servoing on wheels: Robust robot orientation estimation in remote viewpoint control," in IROS. IEEE, 2023, pp. 6364–6370.

[13] D. Zhong, L. Robinson, and D. De Martini, "Nerfoot: Robot-footprint estimation for image-based visual servoing," arXiv preprint arXiv:2408.01251, 2024.

[14] R. Fukushima, K. Ota, A. Kanezaki, Y. Sasaki, and Y. Yoshiyasu, "Object memory transformer for object goal navigation," in ICRA. IEEE, 2022, pp. 11 288–11 294.

[15] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, "Vint: A foundation model for visual navigation," in CoRL. PMLR, 2023, pp. 711–733.

[16] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "Nomad: Goal masked diffusion policies for navigation and exploration," in ICRA. IEEE, 2024, pp. 63–70.

[17] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi, "Visual semantic navigation using scene priors," arXiv preprint arXiv:1810.06543, 2018.

[18] H. Du, X. Yu, and L. Zheng, "Vtnet: Visual transformer network for object goal navigation," arXiv preprint arXiv:2105.09447, 2021.

[19] Z. Wang and G. Tian, "Goal-oriented visual semantic navigation using semantic knowledge graph and transformer," T-ASE, 2024.

[20] D. Shah, B. Osiński, S. Levine et al., "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in Conference on robot learning. PMLR, 2023, pp. 492–504.

[21] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, "Manipulate-anything: Automating real-world robots using vision-language models," in CoRL, 2024.

[22] J. Chen, B. Lin, R. Xu, Z. Chai, X. Liang, and K.-Y. Wong, "MapGPT: Map-guided prompting with adaptive path planning for vision-and-language navigation," in ACL, L.-W. Ku, A. Martins, and V. Srikumar, Eds. ACL, 2024, pp. 9796–9810.

[23] N. D. Palo and E. Johns, "Keypoint action tokens enable in-context imitation learning in robotics," 2024.

[24] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in CVPR, 2021, pp. 10 012–10 022.

[25] J. Carbonell and J. Stewart, "The use of mmr, diversity-based reranking for reordering documents and producing summaries," SIGIR Forum, 06 1999.

[26] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu et al., "Ai2-thor: An interactive 3d environment for visual ai," arXiv preprint arXiv:1712.05474, 2017.

[27] X. Liang, H. Wang, W. Chen, D. Guo, and T. Liu, "Adaptive Image-Based Trajectory Tracking Control of Wheeled Mobile Robots With an Uncalibrated Fixed Camera," IEEE Transactions on Control Systems Technology, vol. 23, no. 6, pp. 2266–2282, Nov. 2015.

[28] X. Liang, H. Wang, Y.-H. Liu, Z. Liu, B. You, Z. Jing, and W. Chen, "Purely Image-Based Pose Stabilization of Nonholonomic Mobile Robots With a Truly Uncalibrated Overhead Camera," T-RO, vol. 36, no. 3, pp. 724–742, Jun. 2020.

[29] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo et al., "Segment anything," in CVPR, 2023, pp. 4015–4026.

[30] M. Reid, N. Savinov, D. Teplyashin, D. Lepikhin, T. Lillicrap, J.-b. Alayrac, R. Soricut, A. Lazaridou, O. Firat, J. Schrittwieser et al., "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," arXiv preprint arXiv:2403.05530, 2024.

[31] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi, and R. Mottaghi, "Learning to learn how to learn: Self-adaptive visual navigation using meta-learning," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 6750–6759.

[32] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou et al., "Chain-of-thought prompting elicits reasoning in large language models," NeurIPS, vol. 35, pp. 24 824–24 837, 2022.