Instance-aware Exploration-Verification-Exploitation for Instance ImageGoal Navigation: Supplementary Materials

Anonymous CVPR submission

Paper ID 4432

1. Switch Policy

1.1. Theoretical Analysis

We consider the following situation: when the agent's current observation contains an object of the same category as in the goal image, a simple Exploration-Exploitation (EE) Policy will directly judge this observation, which can be divided into the following four cases: TP, TN, FP, FN, described in Tab. 1. In this case, the judgment of the current observation can be formulated as:

$$p_{ee}(\text{TP}) = p(\text{TP}|d,t), \tag{1}$$

$$p_{ee}(\text{FP}) = p(\text{FP}|d,t), \tag{2}$$

where $p(*|d,t)$ represents the TP or FP probability at Euclidean distance of d and selected threshold of t. ee refers to the Exploration-Exploitation policy.

On the contrary, our proposed Exploration-Verification-Exploitation (EVE) unfolds this decision-making problem over time, turning it into a sequential decision problem. In Tab. 1, we add a new item in row 5 to illustrate this process. For example, if the Policy cannot determine whether the current observation is the goal or not, it will give the Uncertain prediction. We set the agent forward step size to d_a, then the maximum Verification steps is $l_{max} = \lfloor d/d_a \rfloor$. Hence, the final success rate of the EVE strategy is:

$$\begin{aligned} p_{eve}(\text{TP}) = &\sum_{i=1}^{l_{max}} \prod_{j=1}^{i-1} p(\text{U}|d_c(j), t(d_c(j))) \\ &\times p(\text{TP}|d_c(i), t(d_c(i))), \end{aligned} \tag{3}$$

$$\begin{aligned} p_{eve}(\text{FP}) = &\sum_{i=1}^{l_{max}} \prod_{j=1}^{i-1} p(\text{U}|d_c(j), t(d_c(j))) \\ &\times p(\text{FP}|d_c(i), t(d_c(i))), \end{aligned} \tag{4}$$

where $d_c(i) = d - d_a \times i$ represents the distance at current verification step. In our proposed EVE policy, the threshold t is affected by the Euclidean distance of $d_c(i)$, which will change according to the current verification steps.

We use the formulas in Eq. (3) and Eq. (4) to represent the probabilities of TP and FP obtained by the agent using the EVE Policy after going through a maximum of l_{max} verification steps when the Euclidean distance from the object is d. $\prod_{j=1}^{i-1} p(\text{U}|d_c(j), t(d_c(j)))$ represents the possibility of verifying $i-1$ steps, which can be treated as the weight of $p(*|d_c(i), t(d_c(i)))$, where $*$ denotes TP or FP. To amplify the probability of TP and suppress FP in our EVE policy, the agent should make judgments when the distance is closer, and refrain from making judgments when the distance is farther empirically.

For simplicity, we consider an extreme situation where $p(\text{U}|d_c(i), t(d_c(i)))$ is equal to 1 when the Euclidean distance between the agent and the potential target is not minimal. $p(*|d,t)$ satisfies the following:

$$\sum p(*|d,t) = 1, \quad * = \text{U, TP, FP, TN, FN}. \tag{5}$$

Therefore, $p(\text{TP}|d_c(i), t(d_c(i)))$ or $p(\text{FP}|d_c(i), t(d_c(i)))$ is equal to 0. In other words, the agent only makes judgments when it is closest to the potential target. In such a situation, Eq. (3) and Eq. (4) can be simplified as follows:

$$p^*_{eve}(\text{TP}) = p(\text{TP}|d_c(l_{max}), t(d_c(l_{max}))), \tag{6}$$

$$p^*_{eve}(\text{FP}) = p(\text{FP}|d_c(l_{max}), t(d_c(l_{max}))). \tag{7}$$

Compared with the formulas in Eq. (1) and Eq. (2), we can conclude that in such a extreme situation, EVE Policy changes the decision-making location of the agent from d to d_c, that is, it allows the agent to make judgments from the closest position instead of the original one. Intuitively, we can consider this as being helpful for the agent to make correct decisions. From a quantitative analysis perspective, at the Hard difficulty level with Thresh = 100, which represents the number of matched keypoints, $p_{ee}(\text{TP}) = p(\text{TP}|d = [4, \text{inf}), t = 100) = 0.090$ in Tab. 2. However, for the EVE Policy, it is as if the difficulty level has been switched from Hard to Easy before making a decision, at which point $p^*_{eve}(\text{TP}) = p(\text{TP}|d = [0, 2), t = 100) =$

Event	Description in Context
TP (True Positive)	The policy correctly predicts that the visible object is the goal.
TN (True Negative)	The policy correctly predicts that the visible object is not the goal.
FP (False Positive)	The policy incorrectly predicts that the visible object is the goal.
FN (False Negative)	The policy incorrectly predicts that the visible object is not the goal.
Uncertain (U)	The policy cannot precisely determine whether the visible object is or not the goal.

Table 1. Events and corresponding descriptions.

	Easy				Medium				Hard			
Thresh	TP	TN	FN	FP	TP	TN	FN	FP	TP	TN	FN	FP
40	0.741	0.853	0.259	0.147	0.569	0.848	0.431	0.152	0.526	0.835	0.474	0.165
60	0.651	0.963	0.349	0.037	0.569	0.964	0.431	0.036	0.380	0.961	0.620	0.039
80	0.610	0.977	0.390	0.023	0.500	0.982	0.500	0.018	0.314	0.974	0.686	0.026
100	0.582	0.981	0.418	0.019	0.451	0.986	0.549	0.014	0.090	0.992	0.910	0.008

Table 2. The probabilities of TP, TN, FP, and FN under various difficulty levels (according to the Euclidean distance between the agent and potential target) and selected thresholds (the number of matched keypoints).

0.582. Such a significant improvement will be extremely beneficial for the agent to make accurate and robust decisions.

1.2. Implementation Details

Based on this idea, we design the Goal Map Selection function f_{switch}, depicted in Fig. 1. We analyze the data regarding Instance Re-identification in Tab. 2 and find that when the thresh is set to 100, the TN remains consistently high, ranging from 0.981 to 0.992. This indicates that in the majority of cases, the agent does not mistakenly identify non-existent targets in its field of view. Furthermore, when the threshold is set to 60, regardless of the difficulty level, the TN values still remain relatively high, ranging from 0.961 to 0.964. This indicates that, in cases where the thresh is greater than 60, the probability of the agent making FP errors is extremely low. When thresh is set to 60, the probability of TP increases from 0.380 (hard) to 0.651 (easy). This suggests that there is a significant possibility, under the threshold of 60, for the agent to successfully identify the correct target as it gradually approaches it, without misidentifying the target. Based on the analysis of the data, we design f_{switch} that maps the matched keypoints and the Euclidean distance between the agent and potential targets to the selection signal of the Switch Policy. The function f_{switch} is represented as Fig. 1.

When the distance is large, only the Verification and Exploration strategies exist. This is due to the fact that when the thresh is set to 100, the TP rate is merely 0.090. This suggests that it is challenging for the agent to make accurate judgments about potential targets at a considerable distance. Consequently, we instruct the agent to refrain from making

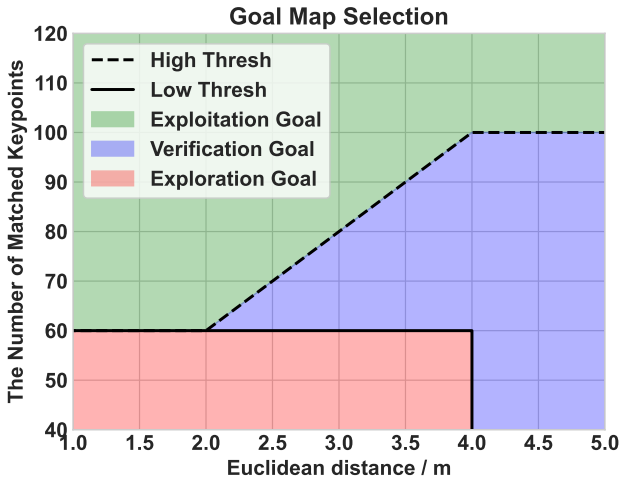

Figure 1. Goal Map Selection f_{switch}.

judgments under such circumstances, and we consistently apply the Verification strategy.

In medium difficulty scenarios, when the thresh is 80, the TN rate remains high at 0.982 while the TP rate reaches 0.500. Therefore, we set the boundary between the Exploitation and Verification strategies from $(2, 60)$ to $(4, 100)$. Given the rapid decline of the TN rate below 60 (dropping from 0.963 to 0.853 in easy condition), we designate values below 60 as the boundary for the Exploration strategy. Since an agent encounters numerous potential targets in a single episode, the occurrence of FP errors is unacceptable. Hence, in our prior strategy determination, we deemed the decline of TN from 0.963 to 0.853 as intolera-

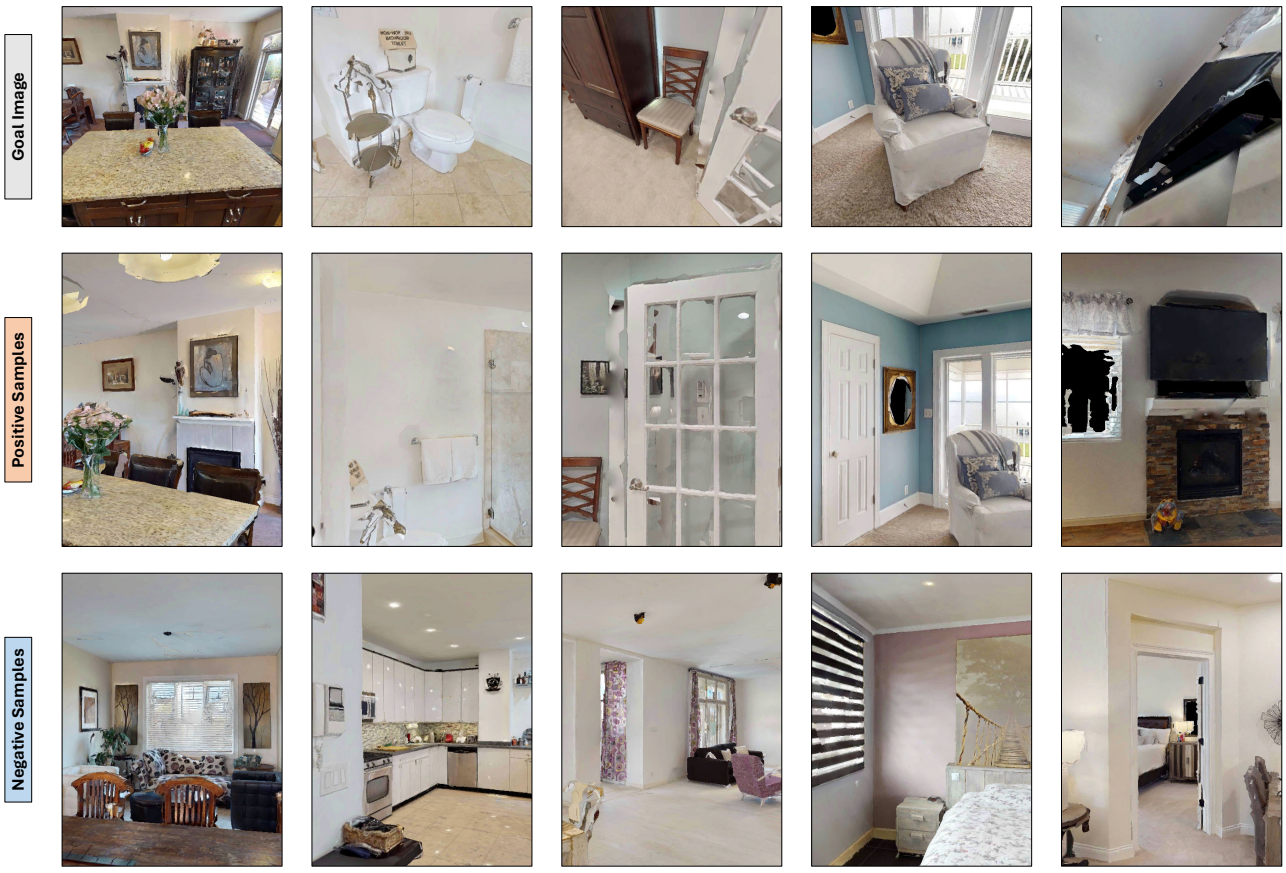

Figure 2. Overview of our Instance Re-identification dataset. The first row indicates the goal image, while the second and third row denotes the positive samples or negative samples respectively.

ble.

2. Exploration-Exploitation Policy Implementation Details

In this section, we introduce the implementation details of our baseline method, *i.e.*, Exploration-Exploitation Policy for Instance Image Goal Navigation. While keeping the Instance Classification and Semantic Mapping modules unchanged, we assess the semantic segmentation of the current observation. If objects of the same category as those in the target image exist, we carry out local feature matching. The quantity of matched key points is then compared with a fixed threshold. If the comparison is successful, the Exploitation policy is executed directly. Otherwise, the Exploration policy continues.

Under the premise of constructing an Instance Re-identification dataset, we set a threshold of 100 through the analysis of the confusion matrix for instance re-identification. We do not choose the threshold based on the highest F1 score because the cost of False Positive (FP) errors is substantial, leading to a steep decline in the success rate. At the same time, successful prediction of positive samples can implicitly depend on the exploration strategy.

3. Local Policy Implementation Details

For a given goal map $\mathcal{M}_g(t)$, the inputs to the Local Policy are its first two channels, representing the distribution of obstacles on the local map $m_o(t)$ and the target distribution on the map $m_g(t)$. Subsequently, the Fast Marching Method (FMM) [1] is utilized to construct the shortest distance field obtained from the flood of the goal map, and the local minimum is selected around the agent. Action is then generated based on the angle between this local minimum and the agent.

In order to enhance the robustness of the agent during path planning, we design the collision map $m_c(t)$ and the visited map $m_v(t)$. Prior to constructing the shortest distance field, we merge the collision map $m_c(t)$ and the visited map $m_v(t)$ with the local obstacle map $m_o(t)$. The construction of the collision map $m_c(t)$ is based on colli-

[1] https://github.com/scikit-fmm/scikit-fmm

sion detection. Collision detection involves inspecting the distance an agent has traveled when a forward action is executed, comparing it with the expected distance. If the actual distance is less than the expected one, a collision is assumed to have occurred. If a collision occurs, the corresponding area in front of the agent on the collision map $m_c(t)$ is set to 1, indicating that the area is untraversable. The construction of the visited map $m_v(t)$ involves setting the path that the agent has traversed prior to timestep t to 1. The collision map sets untraversable area in the front region of the agent when a collision is detected, while the visited map designates the traversed path as traversable area, which are merged to the current obstacle map $m_o(t)$. The processed obstacle map $m_o(t)$ can handle out-of-view collision and memorize previous traversed area. Therefore, under highly complicated indoor environment, it can plan path robustly and efficiently.

4. Instance Re-identification Dataset

We create Instance Re-identification Dataset based on the HM3D-SEM [2] train/val split. Anchor images are defined as goal images that depict specific object instances, which do not share the same parameters as agent's camera. Specifically, anchor images are 512×512 RGB images and are captured from different viewpoints. Viewpoints represent different viewing angles of an object instance at the different height. In contrast, the goal images can be captured from different height, look-at-angle and field-of-view.

For each anchor goal image, we sample 10 positive and negative images using the agent's camera. Positive images are defined as observations where the target instance is visible to the oracle, and the area rate of the visible object instance should not fall below 0.01. Negative images are randomly sampled from the same scene and do not contain the specified goal instance. We have divided the dataset into three distinct difficulty levels: *easy, medium, hard*. *Easy* level is defined as a range of Euclidean distances from the object instance to the current viewpoint of $[0, 2)m$, whereas *medium* level ranges from $[2, 4)m$ and *hard* level encompasses $[4, +\infty)m$. We give a brief overview of our dataset in Fig. 2.

5. Episode Analysis.

We provide three testing episodes of performing Instance ImageGoal Navigation task in the Habitat [1] simulator. Each video frame is divided into three columns. The first column shows the goal image, where the blue text indicates the current policy being executed (Exploration, Verification, Exploitation). The second column represents the current observation, while the third column displays the Goal Map $\mathcal{M}_g(t)$. The three videos demonstrate the process in which our agent explores the environment, locates potential targets, verifies if these potential targets are the objects in the goal image, and makes decisions accordingly. The agent actively switches between Exploration, Verification, and Exploitation policies. After verifying several distracting objects, the agent successfully locates and navigates the target correctly.

References

[1] 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 (CVPR)*, pages 9339–9347, 2019. 4

[2] 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 (CVPR)*, pages 4927–4936, 2023. 4