

A Appendix

A.1 Details of Evaluation Metrics

In this section, we provide detailed explanations the metrics employed for evaluating navigation performance in our experiments:

- **Trajectory Length (TL):** The average distance traveled by the agent during navigation, measured in meters. A shorter TL indicates more efficient navigation independent of success.
- **Navigation Error (NE) (*R2R only*):** The average shortest-path distance from the agent’s final position to the target, measured in meters. A lower NE indicates better localization accuracy.
- **Success Rate (SR):** The percentage of episodes in which the agent stops within a threshold distance (typically 3 meters) from the target location. A higher SR indicates better navigation accuracy.
- **Oracle Success Rate (OSR):** The percentage of episodes in which the agent passes within the success threshold at any point during navigation. A higher OSR indicates better navigation potential assuming optimal stopping.
- **Success weighted by Path Length (SPL):** The average success rate weighted by path efficiency, defined as $SPL = \frac{1}{N} \sum_{i=1}^N S_i \frac{L_i^*}{\max(L_i, L_i^*)}$, where $S_i \in \{0, 1\}$ is the success indicator, L_i^* is the shortest path length, and L_i is the actual path length for episode i . A higher SPL indicates more efficient and accurate navigation.
- **Remote Grounding Success (RGS) (*REVERIE only*):** The percentage of episodes in which the agent successfully identifies the target object upon stopping, determined by a bounding box IoU of at least 50%. A higher RGS indicates better navigation and object grounding accuracy.
- **Remote Grounding SPL (RGSPL) (*REVERIE only*):** A metric integrating RGS with path efficiency, similarly penalizing successful grounding by trajectory length as SPL does. A higher RGSPL indicates efficient navigation combined with accurate object grounding.

A.2 Implementation Details

We search the uncertainty threshold δ within $\{0.1, 0.2, 0.3\}$, and the mixture weight λ from 0 to 1 at 0.1 intervals. Learning rates are chosen from $\{5e-6, 1e-6, 5e-7\}$. The rest of the experimental configurations strictly follow the ones of the pre-trained navigation policies we used for the experiment. To simulate the real-world online test-time adaptation scenarios, we use a batch size of 1. All experiments are conducted on a single NVIDIA RTX 3090 GPU, though the method is lightweight enough to run on lower-powered on-device hardware. The results in the experiment are averaged over 3 different random seeds. To ensure a fair comparison with the episodic update nature of ATENA, we implement TENT [7] in VLN baselines by performing parameter updates at the end of each episode. For FSTTA [37], we set the intervals for slow and fast updates to 4 and 3, respectively, following the original work. However, we modify the learning rates, selecting the fast learning rate from $\{6e-3, 6e-4, 6e-5\}$ and the slow learning rate from $\{5e-3, 1e-3, 3e-4\}$, due to the lack of reproducibility in the original codebase³, aiming to approximate the reported results as closely as possible. For the unseen test split, we adopt GPT-4o as the oracle feedback, following the prompt design introduced in FeedTTA [38].

A.3 Leaderboard Results

Table 6 shows the REVERIE challenge leaderboard ranks⁴ on the test-unseen split, ordered by Success Rate (SR). When integrated with GOAT [60], our proposed method, ATENA, achieves third place on the official leaderboard as of submission date. ATENA offers competitive performance through a lightweight, easily integrable approach that does not necessitate additional pretraining or substantial structural changes, highlighting its practical effectiveness and compatibility with advanced models. In contrast, the top-ranked models, RREx-BoT and RREx-BoT Pre-Explore [61], rely heavily on extensive pretraining with a large-scale vision-language architecture and multiple sophisticated

³<https://github.com/Feliciaxyao/ICML2024-FSTTA/issues/1>

⁴<https://eval.ai/web/challenges/challenge-page/606/leaderboard/1683>

data augmentation techniques. Despite their higher SR and OSR scores, ATENA surpasses RREx-BoT in terms of SPL and RGSPL metrics, further demonstrating its efficiency and practicality in real-world navigation tasks.

Table 6: Leaderboard Performance Comparison

Rank	Model	SR \uparrow	OSR \uparrow	SPL \uparrow	RGSPL \uparrow
1	RREx-BoT Pre-Explore [61]	65.19	73.74	62.04	40.12
2	RREx-BoT [61]	65.18	100.00	42.07	2.78
3	GOAT [60] w/ ATENA (ours)	62.03	64.26	46.82	31.54
4	SRVLN	61.38	66.16	35.62	27.12
5	VinciG	60.46	64.86	41.33	27.84

A.4 Precision of Self-Prediction Head

Since agents rely on Self-Prediction head for relatively certain navigation episodes, its reliability is crucial for stable adaptation. For instance, if Self-Prediction head were unreliable, adaptation could collapse due to false signals. To empirically assess the reliability of our Self-Prediction head, we analyze its predictions of the navigation outcome on the validation unseen split of the REVERIE dataset. The confusion matrix shown in Table 7 demonstrates high reliability with 541 out of 757 negatives (71.46%) and notably 908 out of 1006 positives (90.26%) predicted correctly. Its overall prediction accuracy of 82.19% indicates robust reliability, making it suitable as a pseudo-label source for self-supervised signal. While predictions are not entirely perfect, the proportion of incorrect predictions is significantly lower compared to correct ones, and potential inaccuracies can be mitigated by reducing the weight of self-predicted outputs by γ in Eq. 8. Consequently, as discussed in Table 5 of the main paper, the proposed Self-Prediction head effectively contributes to performance improvements.

Table 7: Confusion matrix of Self-Prediction head

		Prediction	
		Positive	Negative
Actual	Positive	908	98
	Negative	216	541

A.5 Computational Cost

In this section, we analyze the computational cost of ATENA. In Table 8, Nav. time refers to the average duration (ms) taken by the agent to perform a navigation rollout, and Adapt. time denotes the average duration (ms) spent updating the policy in between the episodes. For DUET [4] we only measures the navigation time without performing any adaptation. FSTTA [37] continuously updates its policy during navigation rollout, significantly increasing navigation latency(141.55 ms \rightarrow 1,155.86 ms), whereas TENT [7] and our ATENA introduce minimal additional latency during navigation, which is due to additional collection and calculation of entropy. Although ATENA slightly increases adaptation latency (+0.93%) compared to that of TENT’s, this minor increase is justified by ATENA’s substantial performance improvements over TENT (SR +20.56%, OSR +20.45%). While ATENA’s adaptation time exceeds that of FSTTA, latency during navigation is more detrimental in real-world robotic tasks. Thus, compared to other existing TTA methods in VLN, ATENA stands out as the most efficient and practical approach.

Table 8: Comparison of the average computation time per episode (ms) for TTA methods

Methods	REVERIE - Val Unseen						
	OSR \uparrow	SR \uparrow	SPL \uparrow	RGSPL \uparrow	Nav. Time (ms)	Adapt. Time (ms)	Total (ms)
DUET [4]	51.07	46.98	33.73	23.03	141.55	–	141.55
w/ TENT [7]	51.43	47.55	33.99	23.32	156.23	244.68	400.91
w/ FSTTA [37]	56.26	54.15	36.41	23.56	1,155.86	45.78	1,201.64
w/ ATENA (Ours)	71.88	68.11	45.82	32.26	163.73	246.96	410.69

A.6 Broader Impact

With increasing emphasis on enhancing human-robot interactions, developing effective methods to facilitate these interactions has become crucial. In line with this trend, our proposed method,

ATENA, provides a novel approach enabling robots to adapt effectively to dynamic and complex environments based on individual user feedback. Consequently, ATENA contributes to improved user-centric performance, reliability, and robustness in real-world applications involving active human interaction. However, ambiguity or inconsistency in user feedback might introduce errors in the system’s interpretation, potentially diminishing overall performance and user satisfaction. Therefore, further research into accurately interpreting ambiguous user feedback remains essential.

A.7 Theoretical Analysis of Gradient Update in MEO

We provide a theoretical analysis of the gradient update in Eq. 3 to explain how the proposed pseudo-expert distribution amplifies the directional signal during policy optimization.

Step 1. Standard Entropy Objective.

We define the general entropy objective as:

$$\mathcal{L} = \mathbb{I} \cdot \mathcal{H}(P), \quad (9)$$

where $\mathbb{I} = 1$ for success and $\mathbb{I} = -1$ for failure, and $\mathcal{H}(P)$ denotes the entropy of a probability distribution P . For the policy distribution π_θ , the standard entropy loss at time t is:

$$\mathcal{L}_{\text{standard}} = -\mathbb{I} \sum_{a \in \mathcal{A}_t} \pi_\theta(a|o_t, I) \log \pi_\theta(a|o_t, I). \quad (10)$$

The corresponding gradient is:

$$\nabla_\theta \mathcal{L}_{\text{standard}} = -\mathbb{I} \sum_{a \in \mathcal{A}_t} (1 + \log \pi_\theta(a|o_t, I)) \nabla_\theta \pi_\theta(a|o_t, I). \quad (11)$$

Step 2. MEO Entropy Objective.

In MEO, the policy distribution π_θ is replaced with the pseudo-expert mixture q_{mix} defined in Eq. 1. The loss is formulated as:

$$\mathcal{L}_{\text{MEO}} = -\mathbb{I} \sum_{a \in \mathcal{A}_t} q_{\text{mix}}(a|o_t, I) \log q_{\text{mix}}(a|o_t, I). \quad (12)$$

Since θ only influences q_{mix} through π_θ , the derivative satisfies $\nabla_\theta q_{\text{mix}} = (1 - \lambda) \nabla_\theta \pi_\theta$. Substituting this yields:

$$\nabla_\theta \mathcal{L}_{\text{MEO}} = -\mathbb{I}(1 - \lambda) \sum_{a \in \mathcal{A}_t} (1 + \log q_{\text{mix}}(a|o_t, I)) \nabla_\theta \pi_\theta(a|o_t, I). \quad (13)$$

Step 3. Gradient Discrepancy Comparison.

The per-action weighting terms determine each action’s influence on the update. For standard entropy optimization, the weight gap between a selected action a^{sel} and a non-selected action a' is:

$$\Delta w_{\text{standard}} = \log \frac{\pi_\theta(a^{\text{sel}})}{\pi_\theta(a')}. \quad (14)$$

For MEO, this becomes

$$\Delta w_{\text{MEO}} = \log \frac{q_{\text{mix}}(a^{\text{sel}})}{q_{\text{mix}}(a')} = \log \left(\frac{\lambda + (1 - \lambda)\pi_\theta(a^{\text{sel}})}{(1 - \lambda)\pi_\theta(a')} \right). \quad (15)$$

For any $0 < \lambda \leq 1$, it follows that

$$\Delta w_{\text{MEO}} > \Delta w_{\text{standard}}. \quad (16)$$

This analysis demonstrates that the MEO amplifies the gradient discrepancy between the selected and non-selected actions, thereby concentrating the optimization toward more effectively reinforcing successful and suppressing unsuccessful actions.

A.8 Generalization to Other Navigation Tasks

To further validate the generalizability of ATENA beyond Vision-Language Navigation, we conducted additional experiments on two distinct navigation settings.

A.8.1 Image-Goal Navigation

We apply ATENA to TSGM [62], the baseline of the Image-Goal Navigation task, under the *hard* difficulty setting (see Sec. 4.2 of [62]). ATENA shows notable performance gains even under the more difficult setting, achieving higher success rate (SR) and path efficiency (SPL). This result demonstrates its strong adaptability to purely visual goal-directed navigation.

Table 9: Experimental results on the Image-Goal Navigation.

Method	Hard SR \uparrow	Hard SPL \uparrow
TSGM [62]	70.30	50.00
w/ ATENA	72.49	53.23

A.8.2 IVLN-CE Continuous Navigation

We evaluate ATENA on the more realistic IVLN-CE [63] benchmark, where episodes are sequentially connected without manual resets. Using MAP-CMA [63] as the baseline under the *Inferred via RedNet* map source with the *Iterative* mapping procedure (see Table 4 in [63]), ATENA consistently improves performance across all key metrics on the val unseen split.

Table 10: Experimental results on the IVLN-CE benchmark.

Method	SR \uparrow	SPL \uparrow	nDTW \uparrow
MAP-CMA [63]	35	32	54
w/ ATENA	36	34	56

These results demonstrate the robustness and adaptability of ATENA across both visual-goal and continuous navigation tasks, underscoring its potential as a general test-time adaptation framework for embodied agents.

A.9 Trajectory Visualizations

We illustrate the trajectories of ATENA integrated with DUET in Figure 5, 6, 7, conducted using the REVERIE [17] dataset. In the figure, Trial 1 shows the trajectory before adaptation, where it fails to reach the target destination in the first trial. Trial 2 shows the trajectory after adaptation with our ATENA, which successfully reaches the target destination. The red boxes, defined as Modified Step, highlight navigation points with a high probability of incorrect navigation actions in Trial 1, which shift to a high probability of correct actions in Trial 2 after adaptation, which as a result contributes to success. Green boxes denote that the target object was successfully found, and distances indicate proximity to the target.

A.10 Code base and License

Table 11 provides detailed information about the licenses and official URLs for the datasets and simulators utilized throughout our experiments.

Table 11: License and URLs for Datasets and Simulators

Name	License	URL
R2R [1]	Matterport3D Terms of Use	https://bringmeaspoon.org
REVERIE [17]	Matterport3D Terms of Use	https://github.com/YuankaiQi/REVERIE
R2R-CE [18]	Matterport3D Terms of Use	https://github.com/jacobkrantz/VLN-CE
Matterport3D [64]	Matterport3D Terms of Use	https://niessner.github.io/Matterport/
Habitat Simulator [65]	MIT License	https://github.com/facebookresearch/habitat-sim

Figure 5: In Step 3, Trial 1 selected an incorrect action with 94.5% confidence. After ATENA's adaptation, Trial 2 correctly choose the appropriate action with 99.9% confidence. The intended target—the faucet in the bathroom with a **dark green hand towel**—was missed in Trial 1 (final distance: 10.56), but successfully located in Trial 2.

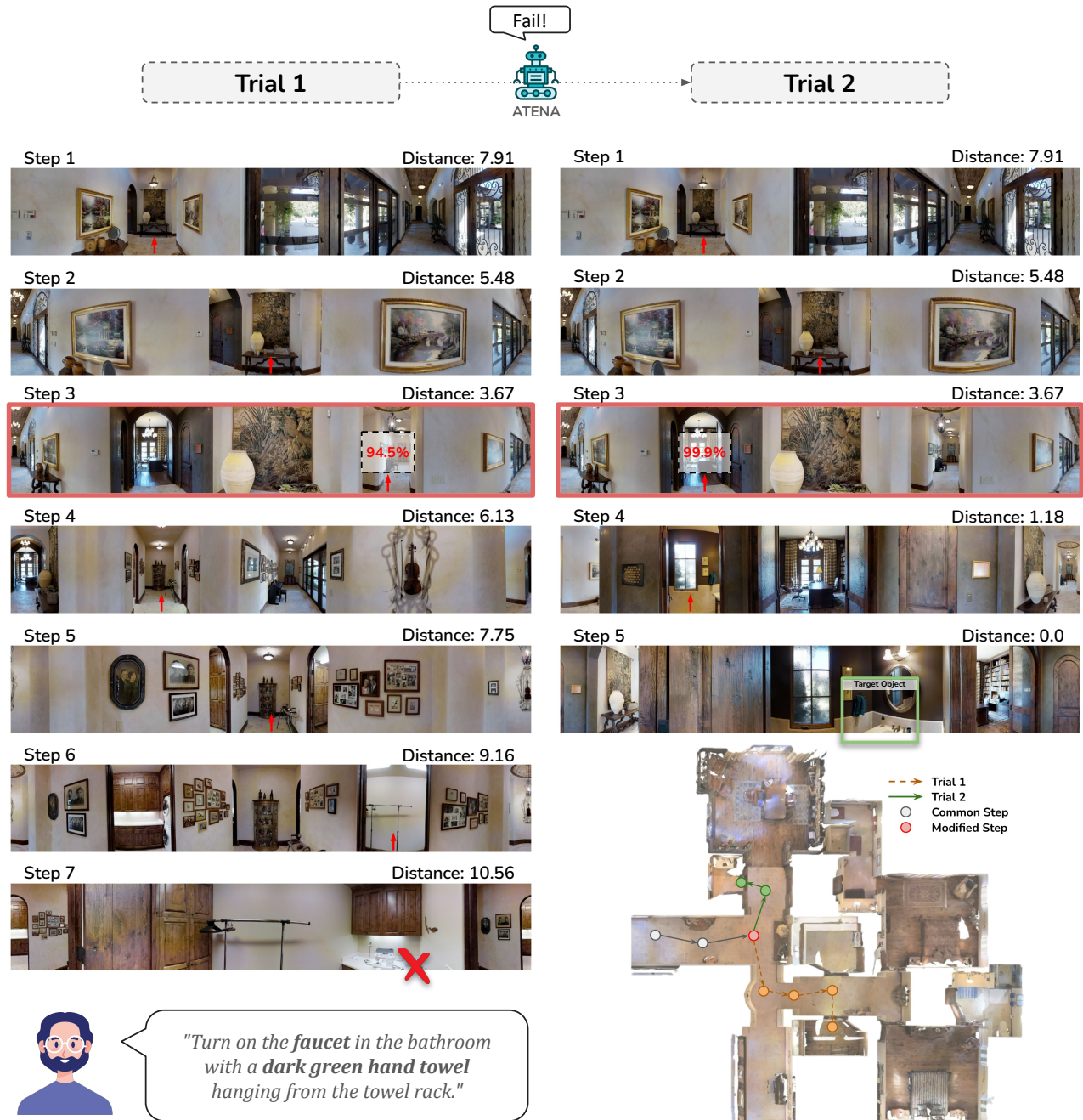


Figure 6: In Step 2, Trial 1 incorrectly selected an action with 85.4% confidence. After ATENA’s adaptation, Trial 2 correctly selects the action with 92.7% confidence. The target object—the **blue pillow** on the beige sofa—was missed in Trial 1 (final distance: 10.69) but was successfully located in Trial 2.

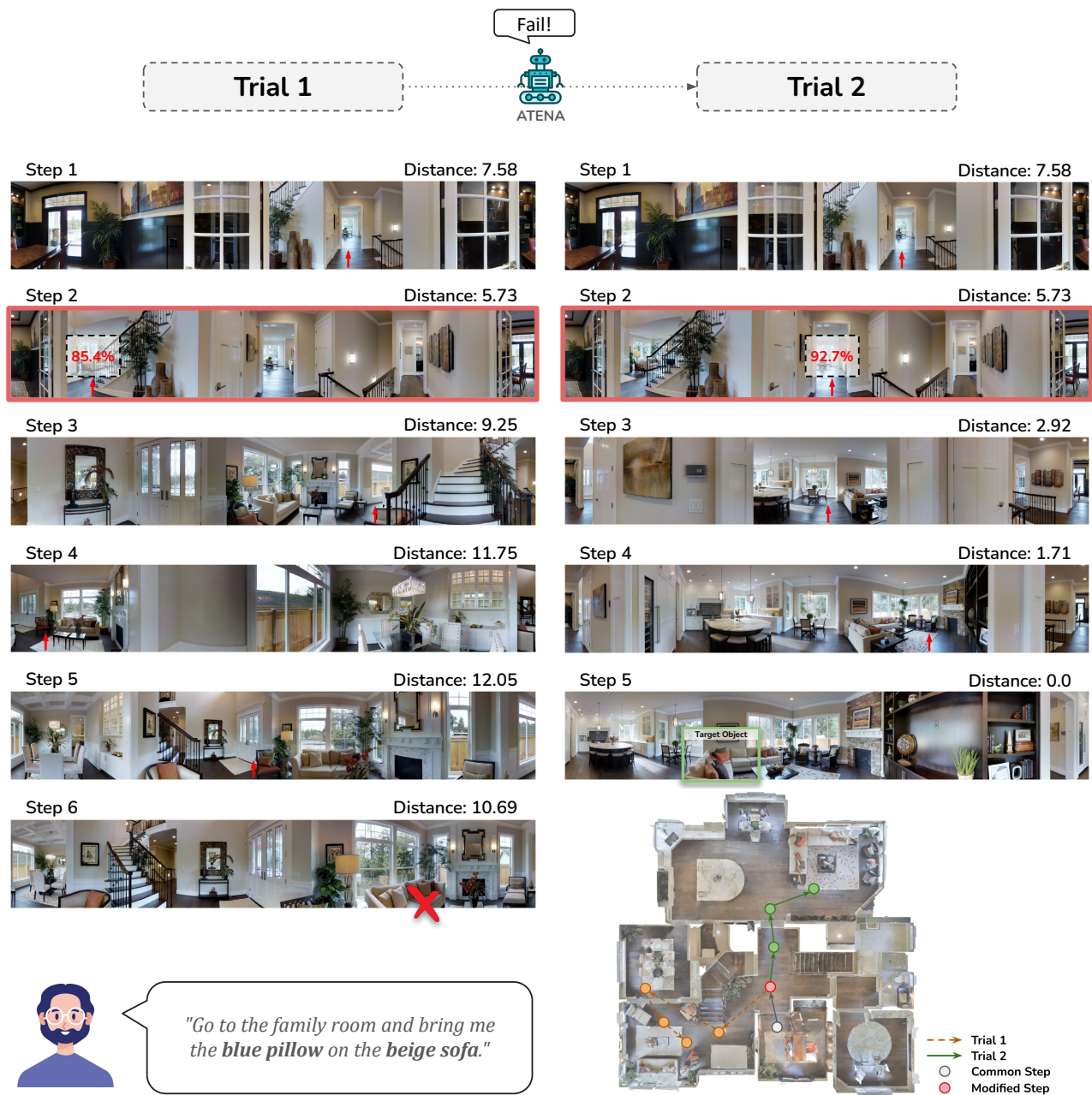


Figure 7: In Step 3, Trial 1 incorrectly selected an action with 98.1% confidence. After adaptation with ATENA, Trial 2 correctly identifies the target action with 100% confidence. The intended target, the **plant on the bathroom counters**, was mistakenly identified in Trial 1 as the plant above the toilet (final distance: 8.3) but successfully located in Trial 2.

