Learning on the Go: A Meta-Learning Object Navigation Model

Xiaorong Qin[1,2], Xinhang Song[1,2], Sixian Zhang[1,2], Xinyao Yu[1,2], Xinmiao Zhang[2], Shuqiang Jiang[1,2,3]
[1]Key Lab of Intelligent Information Processing Laboratory of the Chinese Academy of Sciences (CAS),
Institute of Computing Technology, Beijing, [2]University of Chinese Academy of Sciences, Beijing
[3] Institute of Intelligent Computing Technology, Suzhou, CAS
{xiaorong.qin, xinhang.song, sixian.zhang, xinyao.yu}@vipl.ict.ac.cn, sqjiang@ict.ac.cn

1. Theoretical proofs for our meta-problem

1.1. Transformation of the distribution alignment objective Eq. (1)

It is obvious that $\int q(\tau|o)d\tau = 1$. Then, for the marginal likelihood $p(o)$ of target variable o , we have:

$$\begin{aligned}
\log p(o) &= \int q(\tau|o) \log p(o) d\tau \\
&= \int q(\tau|o) \log \frac{p(o,\tau)}{p(\tau|o)} d\tau \\
&= \int q(\tau|o) \log \frac{p(o,\tau)}{q(\tau|o)} d\tau + \int q(\tau|o) \log \frac{q(\tau|o)}{p(\tau|o)} d\tau \\
&= \int q(\tau|o) \log \frac{p(o,\tau)}{q(\tau|o)} d\tau + D_{KL}[q(\tau|o) \| p(\tau|o)] \\
&= \int q(\tau|o) \log p(o|\tau) d\tau - D_{KL}[q(\tau|o) \| p(\tau)] \\
&\quad + D_{KL}[q(\tau|o) \| p(\tau|o)] \\
&= \left[\mathbb{E}_{q(\tau|o)} \log p(o|\tau) - D_{KL}[q(\tau|o) \| p(\tau)]\right] \\
&\quad + D_{KL}[q(\tau|o) \| p(\tau|o)].
\end{aligned} \tag{13}$$

We know that $\log p(o)$ is deterministic. Next, we can follow the process of variational inference: using the distribution $q(\tau|o)$ to approximate the distribution $p(\tau|o)$, and minimizing the KL divergence between them $D_{KL}[q(\tau|o)\|p(\tau|o)]$. However, since direct minimization is intractable, we transform the problem into maximizing the Evidence Lower Bound (ELBO). Thus, the optimization objective becomes maximizing the ELBO $\mathbb{E}_{q(\tau|o)} \log p(o|\tau) - D_{KL}[q(\tau|o)\|p(\tau)]$.

In our formula Eq. (1), $p_{\theta_i}(\tau|o)$ is $q(\tau|o)$, and $p_\theta(\tau|o))$ is $p(\tau|o))$. Consequently, minmizing $D_{KL}(p_{\theta_i}(\tau|o)\|p_\theta(\tau|o))$ equals to maxmizing $\mathbb{E}_{p_{\theta_i}(\tau|o)} \log p_\theta(o|\tau) - D_{KL}(p_{\theta_i}(\tau|o)\|p_\theta(\tau))$. Our optimization objective Eq. (1) can be transformed as:

$$\max_{p_{\theta_i}(\tau|o)} \mathbb{E}_{p_{\theta_i}(\tau|o)} \left[\log p_{\theta_i}(o|\tau)\right] - D_{KL}\left[p_{\theta_i}(\tau|o) \parallel p_\theta(\tau)\right]. \tag{14}$$

1.2. Simplication of the meta-optimizaiton objective Eq. (3)

We have clarified that sampling from $p_{\theta_i}(\tau|o))$ corresponds to the following process: in the environmental dynamics M, we input the object o into the model f_{θ_i}, which then generates the trajectory τ. Based on this, $p_{\theta_i}(\tau|o)$ can be viewed as a Dirac delta distribution, i.e., $p_{\theta_i}(\tau|o) = \delta(\tau)$, where τ is produced by θ_i given o in the environment M. Consequently, we can derive the first term in Eq. (14) (or Eq. (3) in the main paper) as follows.

$$\begin{aligned}
\mathbb{E}_{p_{\theta_i}(\tau|o)} \left[\log p_\theta(o \mid \tau)\right] &= \int \delta(\tau) \log p_\theta(o \mid \tau) d\tau \\
&= -\log p_\theta(o|\tau)
\end{aligned} \tag{15}$$

1.3. Rewards computing Eq. (6) in the inner-layer optimization of end-to-end methods

Here, we would like to apologize and first address a notation error in the original Equation (6). The correct expression should be:

$$\begin{aligned}
&r_t(a_t, s_t, \tau_{t-1}) \\
&\propto \|o - g_\theta(\tau_{:t})\|_2^2 - \|o - g_\theta(\tau_{:t-1})\|_2^2 - \lambda_1
\end{aligned} \tag{16}$$

We have known that $p_\theta(o|\tau) = \mathcal{N}\left(g_\theta(\tau), \rho^2 I\right)$, so we can obtain that $\log p_\theta(o|\tau)$ is linearly related to $-\|o - g_\theta(\tau)\|_2^2$. Through a series expansion, the proof of the rewards computing in the inner-layer optimization of end-to-end methods can be expressed as:

$$\begin{aligned}
r_t(a_t, s_{t+1}, \tau_{t-1}) &= \log \frac{p_\theta(o \mid \tau_{:t} = [s_{t+1}; \tau_{:t-1}])}{p_\theta(o \mid \tau_{:t-1})} - \lambda \\
&= \log p_\theta(o \mid \tau_{:t}) - \log p_\theta(o \mid \tau_{:t-1}) - \lambda \\
&\propto -[\|o - g_\theta(\tau_{:t})\|_2^2 - \|o - g_\theta(\tau_{:t-1})\|_2^2] - \lambda_1.
\end{aligned} \tag{17}$$

1.4. The outer-layer optimization Eq. (7) in end-to-end methods

We use τ produced by $f_{\theta_i}^*$ updated in the paper to estimate $p_{\theta_i^*}(\tau \mid o)$ and further update meta-parameters about g_θ and f_θ. In the first part of the meta-loss, we utilize the reward signals provided by the simulator and construct the A3C loss to backpropagate higher-order gradients to g_θ and f_θ. And for the second part, trajectory penalty loss, as shown in Eq. (18), we should encourage $p_{\theta_i^*}(\tau)$ by maxmizing $\mathbb{E}_{p_{\theta_i}(\tau|o)}\left[\log p_\theta(o \mid \tau)\right] - D_{KL}\left(p_{\theta_i}(\tau \mid o) \parallel p_\theta(\tau)\right)$ with respect to $p_\theta(o \mid \tau)$ or $p_\theta(\tau)$, through the successful trajectories. In the same way, we should punish negative cases, *i.e.*, the failed trajectories.

$$\min_{f_\theta, g_\theta} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\tau] \cdot \alpha_1 \left\| o - g_\theta(\tau) \right\|_2^2, \text{ where, } \tau \sim f_{\theta_i}^*, \quad (18)$$

where $\mathbb{I}[\tau]$ is a 1/-1 indicator function, which takes a value of 1 if τ is a trajectory where the target is successfully found, and -1 otherwise.

1.5. The outer-layer optimization Eq. (9) in modular methods

Similar to the previous section, we use τ generated by $f_{\theta_i}^*$, as updated in the paper, to estimate $p_{\theta_i^*}(\tau \mid o)$ and further refine the meta-parameters of g_θ and f_θ. In the first part of the meta-loss as Eq. (19), we penalize f_θ based on the correctness of the predicted target point at the end of the episode. The second part applies the same trajectory penalty loss as in the previous section.

$$\max_{f_\theta, g_\theta} \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\tau] \cdot \alpha_2 \left\| z_t - z_T \right\|_2^2, \text{ where, } \tau \sim f_{\theta_i}^*, \quad (19)$$

where z_t is generated by the trajectory after the inner-loop update of the network, and z_T represents the last long-term goal prediction in each episode.The meaning of $\mathbb{I}[\tau]$ is the same as in Equation Eq. (18).

1.6. Benefits of learning the central distribution $p_\theta(\tau \mid o)$ across diverse environments

To prove Proposition 1, we show the following theorem from [2] and [11]. Let $\mathcal{X}$ be a space and $\mathcal{H}$ be a class of hypotheses corresponding to this space. Suppose $\mathbb{S}$ and $\mathbb{Q}$ are distributions over $\mathcal{X}$. Then for any $h \in \mathcal{H}$,

$$\mathcal{E}_{\mathbb{Q}}(h) \leq \lambda + \mathcal{E}_{\mathbb{S}}(h) + \tfrac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}), \quad (20)$$

where λ is the error of an ideal joint hypothesis for $\mathbb{Q}$, $\mathbb{S}$, $\mathcal{E}_{\mathbb{S}}(h)$ can be minimized by Empirical Risk Minimization (ERM), and the divergence can be minimized by learning indiscernible representations of the distributions.

Now we give the proof of Propostion 1.

Given that $h \in \mathcal{H}$, we have the following inequality according to Eq. (20):

$$\alpha_i \mathcal{E}_{\mathbb{Q}}(h) \leq \alpha_i \lambda_i + \alpha_i \mathcal{E}_{\mathbb{S}_i}(h) + \frac{\alpha_i}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}_i) \quad (21)$$

We can define $\lambda_\alpha = \sum_i \alpha_i \lambda_i$, and obtain:

$$\sum_i \alpha_i \mathcal{E}_{\mathbb{Q}}(h) \leq \lambda_\alpha + \sum_i \alpha_i \mathcal{E}_{\mathbb{S}_i}(h) + \frac{\alpha_i}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}_i). \quad (22)$$

We know that $\sum_i \alpha_i = 1$ from Proposition 1, Eq. (22) becomes:

$$\mathcal{E}_{\mathbb{Q}}(h) \leq \lambda_\alpha + \sum_i \alpha_i \mathcal{E}_{\mathbb{S}_i}(h) + \frac{1}{2} \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}_i). \quad (23)$$

The $\mathcal{H}$-divergence follows the triangle inequality for each $\mathbb{S}_i$:

$$d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}_i) \leq d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{S}_i) \quad (24)$$

where

$$\mathbb{S}^* = \arg\min_{\mathbb{S} \in \mathcal{O}} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}). \quad (25)$$

According the inequality Eq. (24), we have:

$$\begin{aligned} &\frac{1}{2} \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}_i) \\ &\leq \frac{1}{2} \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{S}_i) \\ &= \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \sum_i \alpha_i d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}^*, \mathbb{S}_i) \\ &\leq \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{Q}, \mathbb{S}^*) + \frac{1}{2} \max_{\mathbb{S}_i, \mathbb{S}_j} d_{\mathcal{H}\Delta\mathcal{H}}(\mathbb{S}_i, \mathbb{S}_j) \end{aligned} \quad (26)$$

In Proposition 1, it is important to note that under extreme assumptions, the third and the last terms may exhibit a competitive relationship. Specifically, as the last term decreases, the third term may increase. In such cases, it becomes necessary to balance the learning objectives to maintain their relationship.

2. Experimental results for our meta-problem

2.1. Datasets

The evaluation of our meta-learning mechanism integrated with end-to-end RL methods is executed on iTHOR [8] within the AI2-THOR platform. iTHOR provides near-photorealistic observations in 3D environments, encompassing 120 scenes.

For modular methods, we evaluate the meta-learning mechanism on the HM3D [14], MP3D [3] and Gibson [13] datasets. We follow the setup protocol from the previous works [4, 10, 16] For Gibson, we utilize 25 training and 5

validation scenes from the Gibson tiny split and choose 6 goal categories with 1,000 evaluation episodes. For HM3d, we use 80 training, 20 validation and 20 test scenes, offering 2,000 valuation episodes with a total of 21 goal categories. In the case of MP3D, we apply the standard split of 56 training, 11 validation, and 18 test scenes, offering 2,195 episodes with a total of 21 goal categories for evaluation.

2.2. Evaluation metrics

We use the following metrics for evaluating our models in the object navigation experiments. (1) **Succ.**: the success rate. (2) **SPL**: the success weighted by path length [1]. SR can assess the effectiveness of navigation and SPL can assess the efficiency of navigation, which are:

$$\text{Succ.} = \frac{1}{N}\sum_{i=1}^{N} S_i,\ \text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{L_i^*}{\max(L_i^*, L_i)}, \quad (27)$$

where N is the number of episodes in the experiments and S_i is a 0-1 indicator that represents whether the i-th episodes succeeds, L_i and L_i^* are the path length taken by the agent and the theoretical shortest path length in the i-th episode.

2.3. Baselines

DAT [6] integrates two cognitive approaches: "Search Thinking" and "Navigation Thinking." This dual approach adapts to different phases of the navigation task. PEANUT [15] introduces a method for ObjectGoal Navigation that aims to predict the locations of unseen target objects in unfamiliar environments by leveraging spatial and semantic regularities. Meanwhile, we include another method as baseline comparison: SAVN [12], which introduces a self-supervised interaction loss, which the agent uses to adapt to unseen environments during inference. This loss encourages the agent's gradients to align with those obtained from supervised navigation loss during training

2.4. Implementation details

For end-to-end RL methods, we train our model for 3M episodes induced by various environment-specific policies in total with 16 asynchronous agents. We adopt Adam [7] while setting the outer-layer learning rate as $1e-4$ for learning meta-parameters and the inner-layer learning rate $5e-5$ for environment-specific parameters. Meanwhile, we set the size of and the number of updating steps as 2 in the inner layer. For modular methods, we train our model for 3M iterations, and we also use the outer-layer learning rate as $1e-4$ and the inner-layer learning rate $5e-5$. Further details are provided in the appendix.

The implementation of the trajectory decoder involves inputting 512x7x7 features of ResNet-50 from different frames into a Transformer with 6 self-attention layers and 8 heads, followed by a linear layer to output a vector that is ultimately aligned with a one-hot vector.

For the end-to-end methods DAT or HOZ, our implementation of the navigation policies in f_θ is consistent with that in [6] or [17], and the inner-layer update interval K is 6. For the modular method PEANUT, our long-term step interval k is 10, which is consistent with that in [15]. And every $M = 2$ times the network updates the long-term goal outputting probability map, we update the prediction network. Therefore, the step interval we update the prediction network K is 20.

2.5. Evaluation setup

Cross-environment and cross-simulator generalization evaluation. We follow the traditional evaluation setup of previous works [5, 6, 9, 17] for object navigation task. For each episode in the training or test stage, we select a scene and a target object randomly or non-randomly in corresponding sets to perform it. And we keep consistent with previous works on the predefined test episodes in which scenes and targets are picked in advance. To evaluate the performance of the meta-learning mechanism in unknown environments with an expanded generalization gap, we apply the model trained on one simulators to new environments from other simulators to demonstrate the flexibility and adaptability of our approach.

Adaptivity evaluation for one specific environment. The problem is particularly fatal when encountered with the very realistic situation where an agent needs to perform multiple navigation episodes in an identical unfamiliar environment. Therefore, to analyze the average performance of an environment-specific model, we propose a new evaluation setup in which multiple navigation episodes are conducted with randomly selected targets within the same environment after environment-specific model adaptation. We randomly select some environments and then calculate their average results. More details about the dataset setup can be found in the appendix. To be specific, in AI2-Thor, we select 20 scenes from each scene type randomly for meta-training, 5 scenes for meta-validation and 5 scenes for meta-test. As can be seen above, we set the environment-specifc model learning adaptively for each scene. We need to sample K_1 object-navigation episodes $S_m = \{(o^{(1)}, o^{(2)}, ..., , o^{(K_1)}\}$ (where o_i is the target to be located.) in each scene for the learning process 14. After learning the environment-specifc model, we sample $Q_m = \{o^{(1)}, o^{(2)}..., o^{(K_2)}\}$ through it to evaluate it in the current scene.

2.6. Qualitative results

As shown in the Fig. 1, we visualize the results of models PEANUT and PEANUT+LOG and observed that when searching for the bed, proving that our method is more efficient.

References

[1] 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 preprint arXiv:1807.06757*, 2018. 3

[2] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010. 2

[3] Sébastien Bubeck, Varun Chadrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4, 2023. 2

[4] Shizhe Chen, Thomas Chabal, Ivan Laptev, and Cordelia Schmid. Object goal navigation with recursive implicit maps. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7089–7096. IEEE, 2023. 2

[5] Ronghao Dang, Zhuofan Shi, Liuyi Wang, Zongtao He, Chengju Liu, and Qijun Chen. Unbiased directed object attention graph for object navigation. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 3617–3627, 2022. 3

[6] Ronghao Dang, Liuyi Wang, Zongtao He, Shuai Su, Jiagui Tang, Chengju Liu, and Qijun Chen. Search for or navigate to? dual adaptive thinking for object navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8250–8259, 2023. 3

[7] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[8] 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 preprint arXiv:1712.05474*, 2017. 2

[9] Yiqing Liang, Boyuan Chen, and Shuran Song. Sscnav: Confidence-aware semantic scene completion for visual semantic navigation. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 13194–13200. IEEE, 2021. 3

[10] 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*, pages 18890–18900, 2022. 2

[11] Anthony Sicilia, Xingchen Zhao, and Seong Jae Hwang. Domain adversarial neural networks for domain generalization: When it works and how to improve. *Machine Learning*, 112 (7):2685–2721, 2023. 2

[12] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh 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*, pages 6750–6759, 2019. 3

[13] 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 conference on computer vision and pattern recognition*, pages 9068–9079, 2018. 2

[14] 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*, pages 4927–4936, 2023. 2

[15] Albert J Zhai and Shenlong Wang. Peanut: Predicting and navigating to unseen targets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10926–10935, 2023. 3

[16] Jiazhao Zhang, Liu Dai, Fanpeng Meng, Qingnan Fan, Xuelin Chen, Kai Xu, and He Wang. 3d-aware object goal navigation via simultaneous exploration and identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6672–6682, 2023. 2

[17] Sixian Zhang, Xinhang Song, Yubing Bai, Weijie Li, Yakui Chu, and Shuqiang Jiang. Hierarchical object-to-zone graph for object navigation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15130–15140, 2021. 3

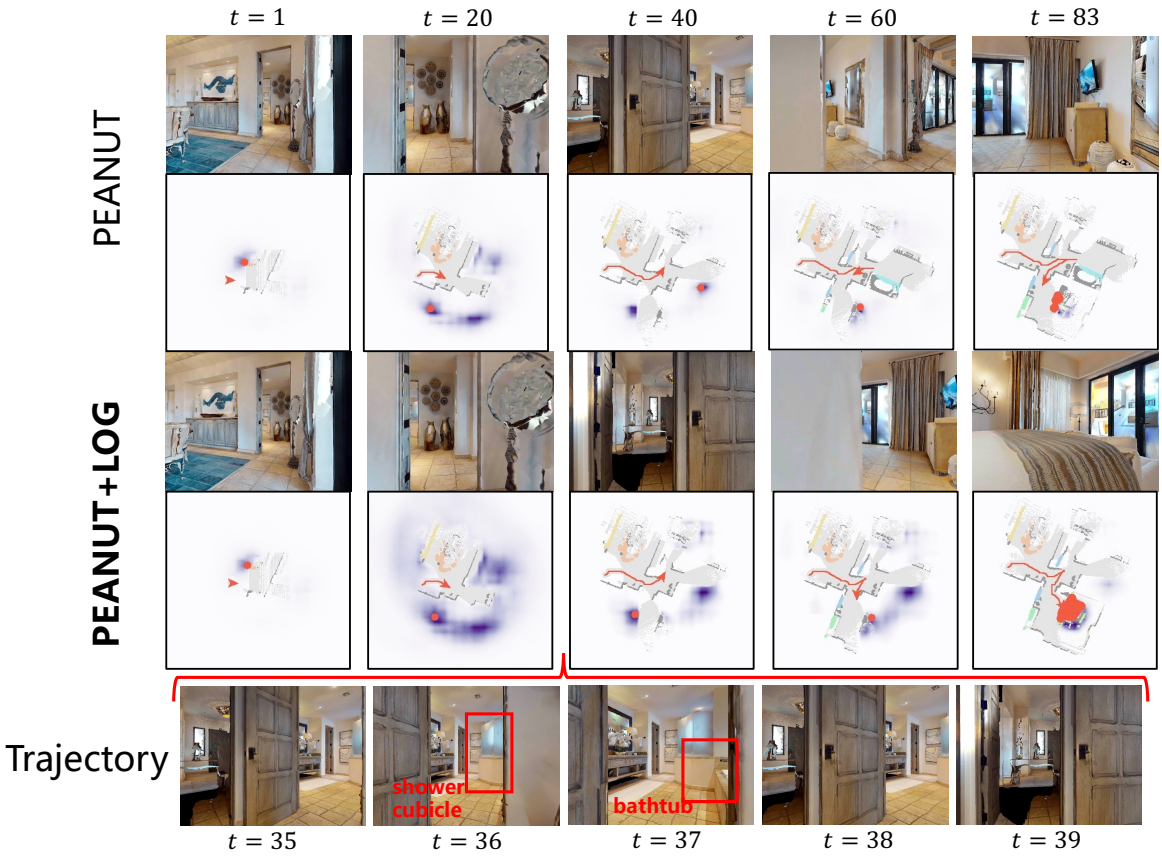

Figure 1. Example navigation episodes from HM3D (val) using PEANUT and PEANUT+LOG.