MEMORY-BASED PARAMETERIZED SKILLS LEARNING FOR MAPLESS VISUAL NAVIGATION

Yuyang Liu, Yang Cong* and Gan Sun

liuyuyang@sia.cn , congyang@sia.cn and sungan@sia.cn

中国科学院沈阳自动化研究所
SHENYANG INSTITUTE OF AUTOMATION,CHINESE ACADEMY OF SCIENCES

MOTIVATION & CONTRIBUTIONS

- **Motivation:**
  Reinforcement learning for mapless visual navigation can generate an optimal policy for searching different targets, but it is still challenging :
  - Ignoring previous knowledge relying on discrete rewards;
  - Pre-trained network cant be quickly generalized into un-trained tasks;
  - Discriminative information among different states is ignored.
- **Contributions:**
  - Parameterizing previous knowledge facilitates generalization to un-trained tasks.
  - Policy parameters are changed with task parameters and entered targets.
  - Feature alignment strategy to learn distinguishable features between different states.

TASKS

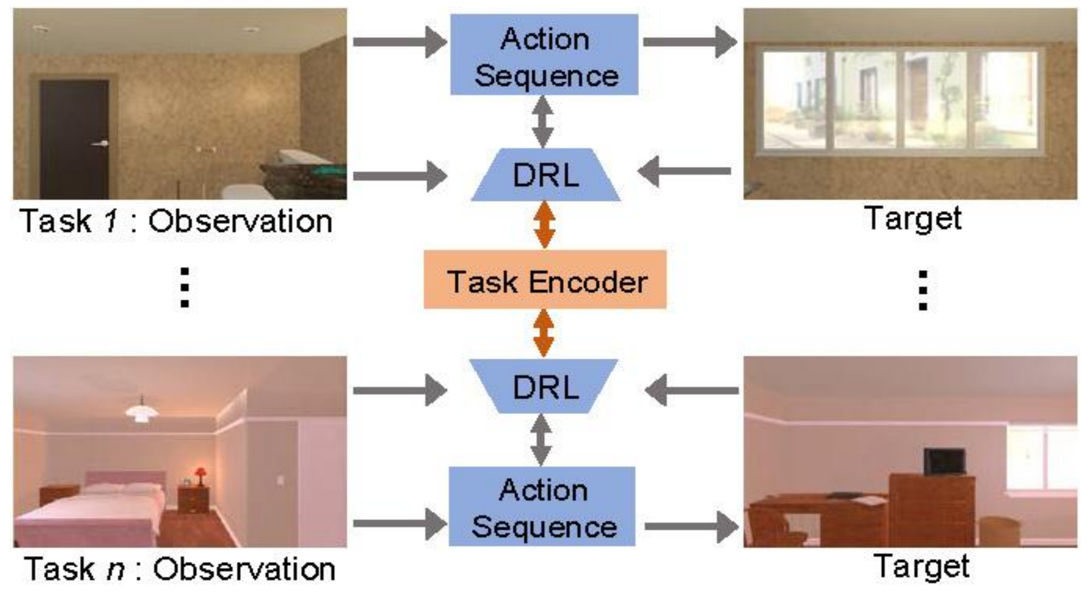


Fig. 1. Illustration of n visual navigation tasks, where a DRL agent should take the shortest sequence of actions to navigate from current state to target state for each task.

- **Scene：** 4 type high quality realistic indoor scenes following AI2THOR[25] : kitchens, living rooms, bedrooms and bathrooms.
- **Task：** Each scene contains 5 targets (100 target tasks in total).
- **Action space:** Move-ahead/back with a 0.5 meters step, Rotate-left/right with a 90 degree rotation.
- **Reward:** 10.00(reaches target state); -0.01(time penalty)

NETWORK ARCHITECTURE

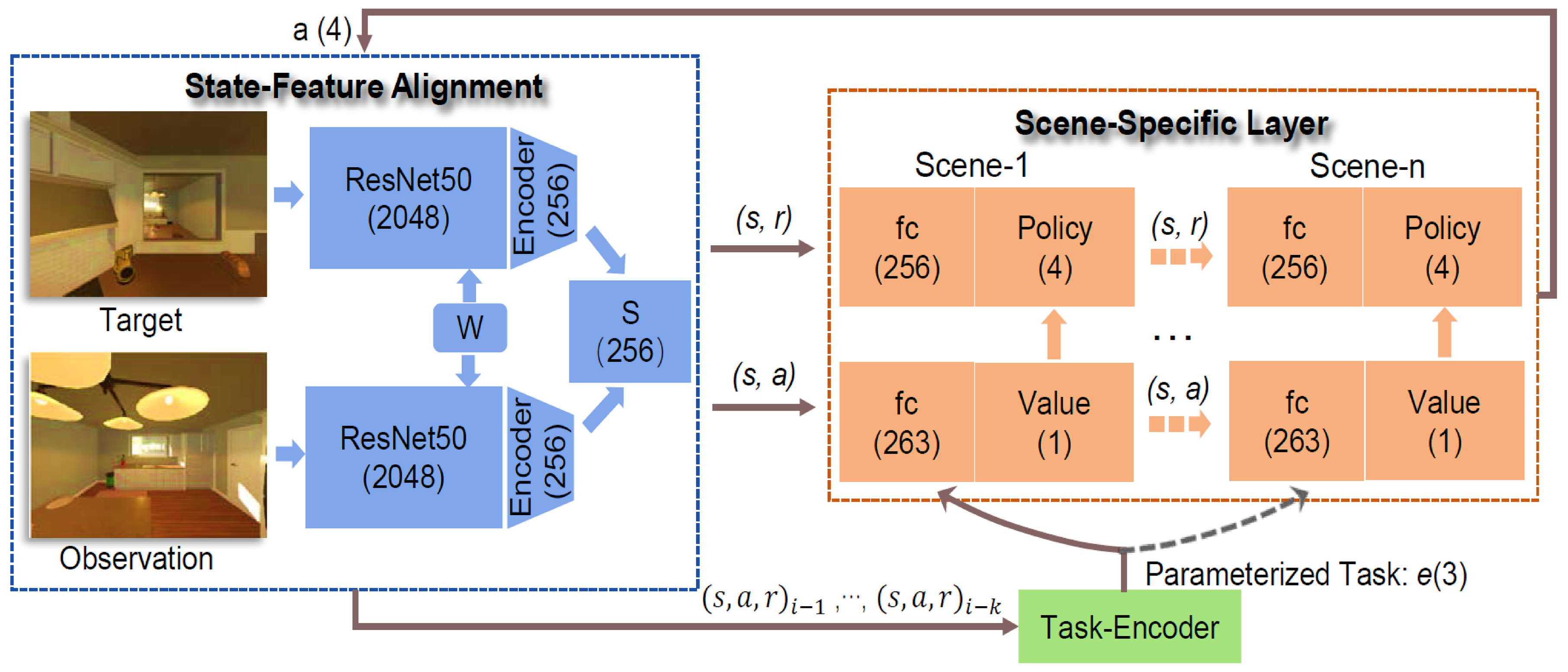


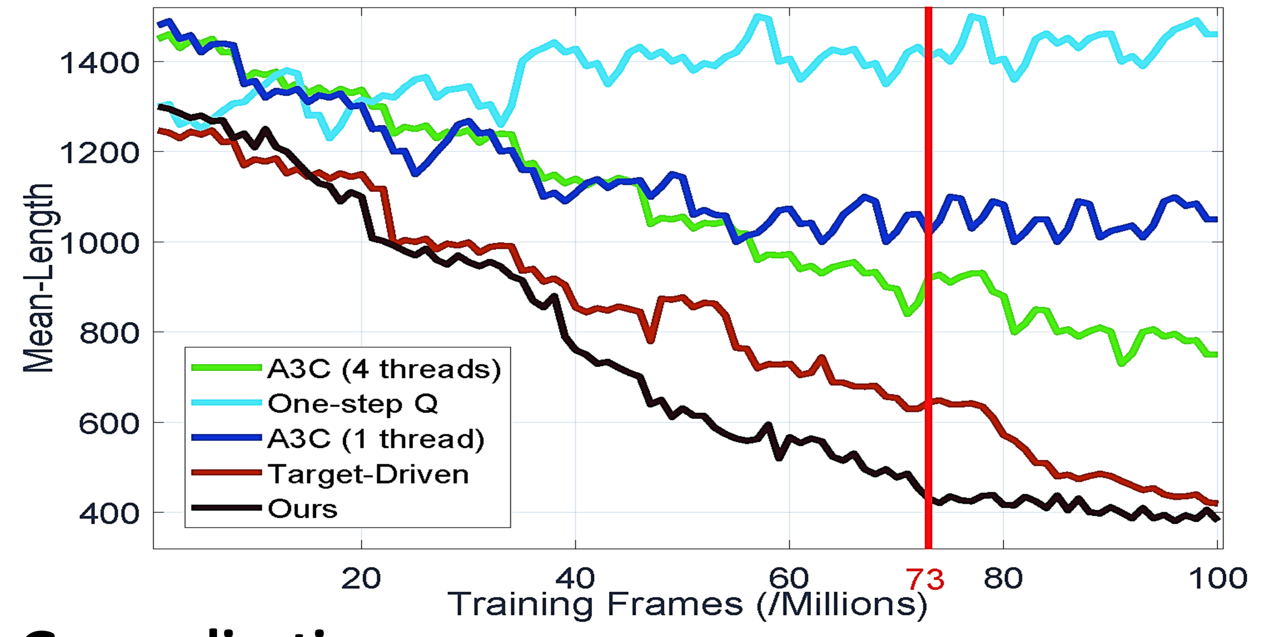
Fig. 2. Network architecture of our Memory-based Parameterized Skills Learning (MPSL), which can learn parameterized task e from the memory sequence of *(s, a, r)*.

PROPOSED METHOD

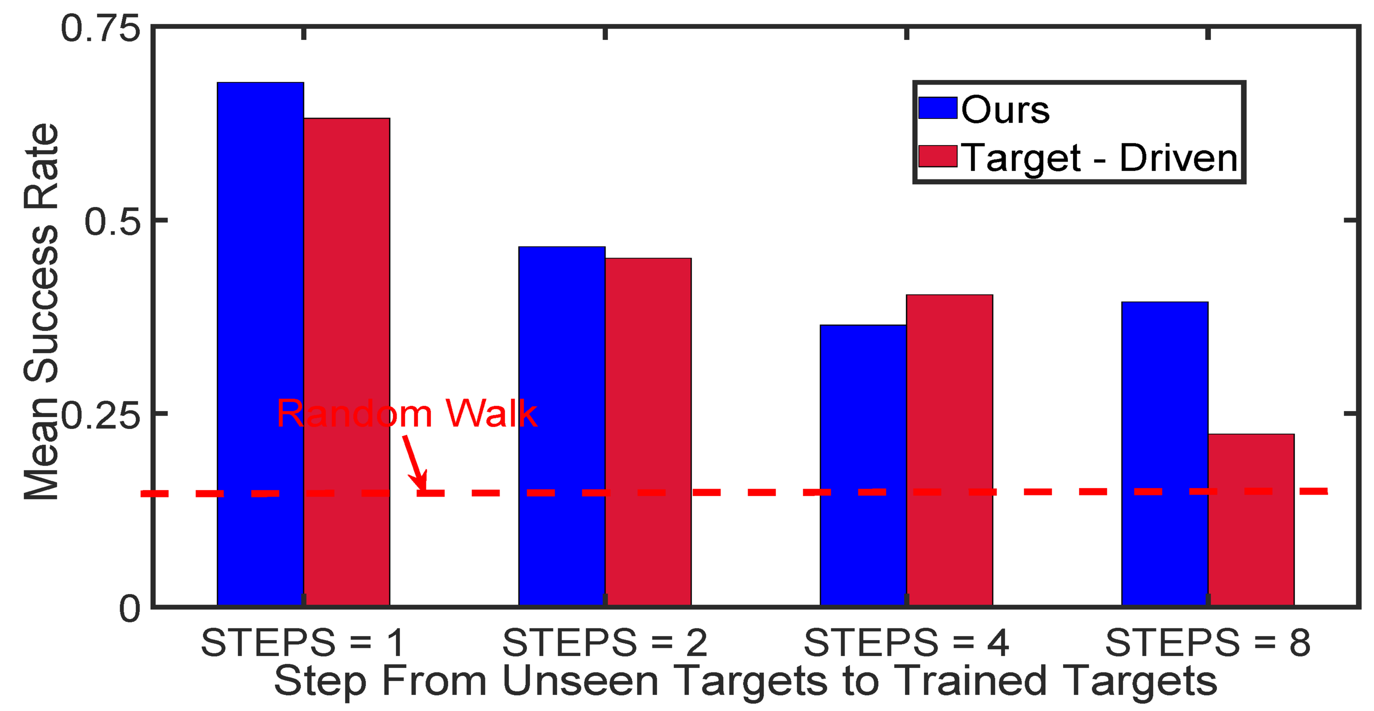
- **Problem Setup：**
  - Task distribution: $t \in [1, m] \sim \mathcal{T}$
  - Embedding state: $s_i^t \subset \mathcal{S} = [s_1^1, ..., s_n^1, ..., s_1^m, ..., s_n^m]$
  - Current observation image: $o_i^t \subset \mathcal{O} = [o_1^1, ..., o_n^1, ..., o_1^m, ..., o_n^m]$
  - Target image: $g_t \in \mathcal{G} = [g_1, ..., g_m]$
  - Expected actions: $\mathcal{A}_t = [a_1^t, a_2^t, ..., a_i{}^t, ..., a_n^t]$
- **State Feature Alignment:** $\mathcal{L}_s = \|\mathcal{F}_o - \mathcal{F}_t\|_2, \quad (1)$
- **Memory-based Parameterized Skills:**
  - Parameterize state-action-reward pairs:
    $$\mathcal{I}_i^t = \sum_{l=0}^{k} \gamma^{k-l}(s_{i-l}^t, a_{i-l}^t, r_{i-l}^t), \quad (2)$$
  - Task-encoder E_ω ：
    $$e_i^t = E_\omega(\mathcal{I}_i^t), \quad (3)$$
- **Scene-specific Layer：**
  - Historical rewards after each k-step action:
    $$\mathrm{Adv}(s_i^t, a_i^t, e_i^t; \theta, \varphi) = Q(s_i^t; \theta, \varphi) - V_\varphi(s_i^t, a_i^t, e_i^t; \varphi), \quad (4)$$
    $$\mathcal{L}_v = \mathbb{E}[\mathrm{Adv}(s_i^t, a_i^t, e_i^t; \theta, \varphi)^2 / \partial\varphi']^2. \quad (5)$$

EXPERIMENTS RESULT

- **Evaluations:** 1) **Mean-Length:** close to the shortest path; 2) **Mean-Reward:** as close as possible to 10.00; 3) **Mean-Collision:** no collision; 4) **Mean-Success-Rate:** the mean navigation length of less than 500 steps per episode is considered as success.
- **Navigation Training:**
  - Training 100 millions frames for 100 tasks;
  - Converge to the minimum mean-length at 73M frames;
  - Feature-aligned loss designation can accelerate learning



- **Targets Generalization**
  - Each target runs 100 episodes, and the maximum steps is 500;
  - Generalization of 10 un-trained targets in each scene, where the red line is the mean-success-rate of Random Walk;
  - Clear understanding in surrounding area of trained targets;



[25] E. Kolve, R. Mottaghi, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi, “AI2-THOR: an interactive 3d environment for visual AI,” arXiv: Computer Vision and Pattern Recognition, vol. 1712.05474, 2017.