SAILOR Learning Poster

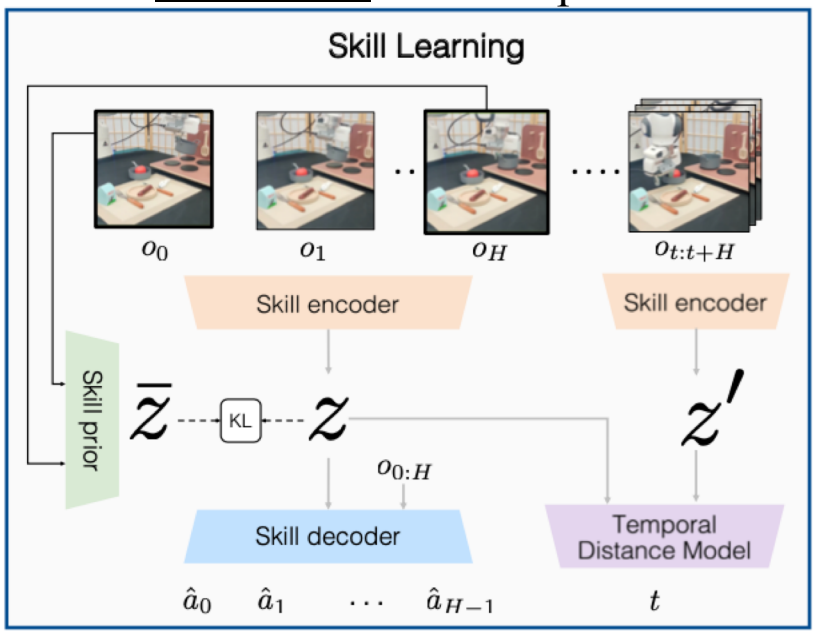
Tags: Robot Learning, Imitation Learning

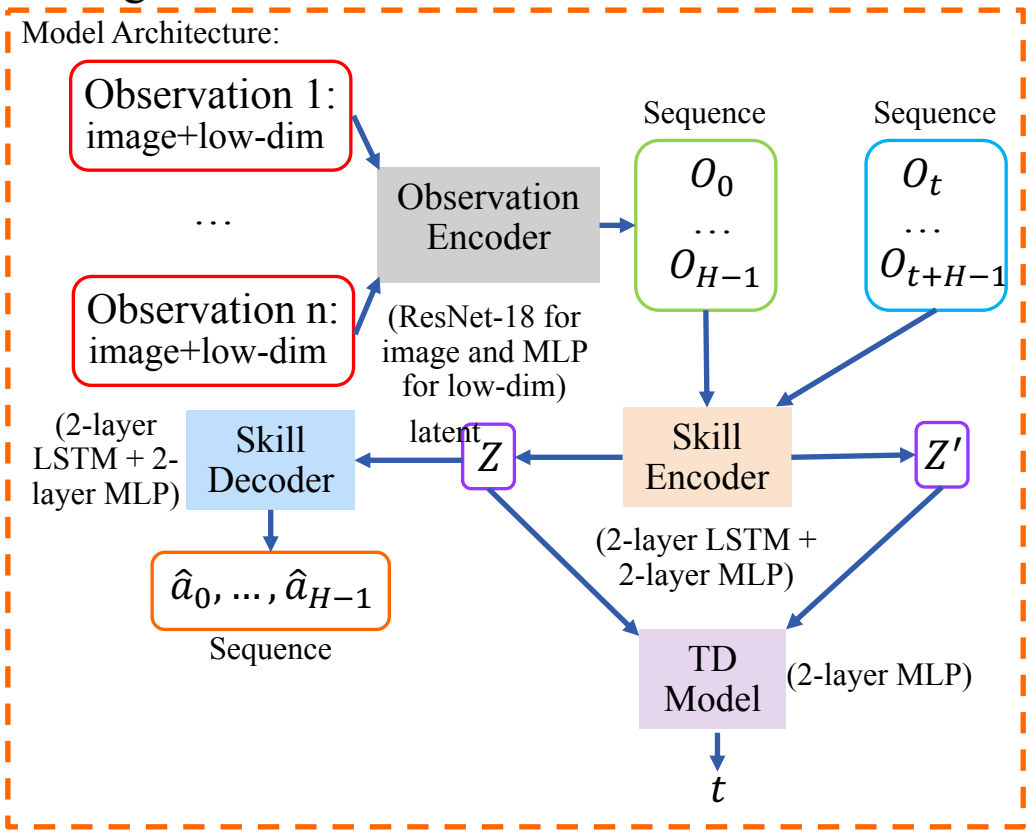
(Reference: Nasiriany, Soroush, et al. "Learning and Retrieval from Prior Data for Skill-based Imitation Learning." arXiv preprint arXiv:2210.11435 (2022).)

Problem: Data-efficient Imitation Learning for Target Task by Utilizing Prior Robotic Dataset

Algorithm:

1. For MDP problem $\mathcal{M}(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho_0)$, learn policy π: $\hat{z} = \pi(o_{fs})$ to maximize the discounted sum of rewards ($\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$) for the task.
2. Pretrain and finetune paradigm:
   a. Predictable Skills Representation Learning from Prior Data





Training Loss: Actually l_2 loss

$$\mathcal{L}_{\text{VAE}}(\phi, \psi, \theta) = -\mathbb{E}_{z \sim q_\phi(z|\tau)} \left[\sum_{t=0}^{H-1} \log p_\psi(a_t | z, o_t) \right] + \beta \cdot D_{KL}(q_\phi(z|\tau) || p_\theta(z|o_0, o_H))$$

$$\mathcal{L}_{\text{TP}}(\omega, \phi) = \left(m_\omega \left(\mu(q_\phi(z|\tau_1)), \mu(q_\phi(z|\tau_2)) \right) - t \right)^2$$

$$\mathcal{L}_{\text{Skill}}(\phi, \psi, \theta, \omega) = \mathcal{L}_{\text{VAE}}(\phi, \psi, \theta) + \alpha \mathcal{L}_{\text{TP}}(\omega, \phi)$$

Encourage the learned latent space to predict the temporal difference between two sub-trajectories

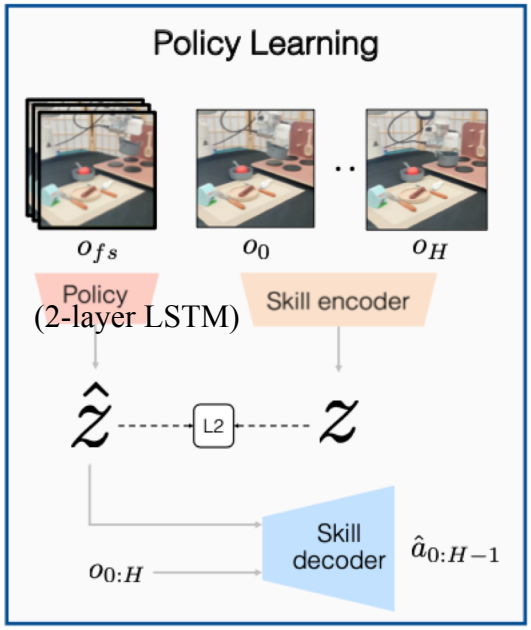
Learned VAE prior (2-layer MLP) to encourage sub-trajectories with similar starting and ending observations to have similar latent representations.

KL divergence: encourage learned skills to be predictable.

Problem: Data-efficient Imitation Learning for Target Task by Utilizing Prior Robotic Dataset

Algorithm:

1. For MDP problem $\mathcal{M}(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho_0)$, learn policy π: $\hat{z} = \pi(o_{fs})$ to maximize the discounted sum of rewards ($\mathbb{E}[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1})]$) for the task.
2. Pretrain and finetune paradigm:
   b. Retrieval-based Policy Learning using Target Task Data



Retrieval-based data augmentation:
Find top K nearest neighbors from $\mathcal{D}_{prior}$ as augmented data for policy learning and skills representation learning model fine-tuning

Increase the scope of supervision for policy training.

// Train policy
while not done **do**
  Sample $(o_{fs}, z) \sim \mathcal{D}_{\text{target}}$ // target dataset sub-traj encoding and frame stack
  Sample $(o'_{fs}, z') \sim \mathcal{D}_{\text{ret}}$ // retrieval dataset sub-traj encoding and frame stack
  $\hat{z} \leftarrow \pi(o_{fs}, \text{id} = 0)$ // predict skill for target dataset sub-traj
  $\hat{z'} \leftarrow \pi(o'_{fs}, \text{id} = 1)$ // predict skill for retrieval dataset sub-traj
  $\mathcal{L}_{\text{Policy}} \leftarrow (\hat{z} - z)^2 + \gamma \cdot (\hat{z'} - z')^2$ // Compute Policy Loss
  update π on $\mathcal{L}_{\text{Policy}}$ via gradient descent
  fine-tune $\mathcal{L}_{\text{Skill}}$ on sub-trajectories sampled from $\mathcal{D}_{\text{prior}}$ and $\mathcal{D}_{\text{target}}$ // see Algorithm 1
end while

Summarization and Personal Thinking

Summarization of innovations:

1. Pretrain and finetune paradigm:
   a. Compared with pretraining policy model on prior task data and then finetuning on target task data, pretraining predictable skills representations as downstream policy model's augmented training data can eliminate harmful effects (harmful effects due to conflicts and divergence between prior and target data when naively training policy on prior and target data).
   b. Nearest pretrained skills representations can increase the scope of supervision for policy learning, thus lead to better generalization.
2. Data-efficient retrieval-based data augmentation:
   a. Target task data's nearest neighbors from learned skills as data augmentation is data-efficient especially when target data set is small.
   b. Retrieval-based data augmentation also considers about relevance property and utilizes this property.

Personal thinking of future work:

1. Multi task imitation learning and merging two phases into one-stage for better computation efficiency.
2. Changing RNN-based encoders and decoders to transformer-based ones for computation efficiency.
3. Investigate the effectiveness of this approach with various forms of prior data at different scales (mentioned in paper).