Supplementary Material for Learning Compound Tasks without Task-specific Knowledge via Imitation and Self-supervised Learning

Sang-Hyun Lee [1 2] **Seung-Woo Seo** [2]

A. Training Details

We propose an imitation learning method that can learn compound tasks without task-specific knowledge. Our method consists of four models: the policy $\pi(a_t|s_t, c_t)$, discriminator $D(s_t, a_t)$, posterior $Q(c_t|s_t, a_t)$, and task transition model $T(c_t|c_{t-1}, s_t)$. These models are represented as neural networks, and the parameters for each network are θ, ω, ψ and υ. The policy and discriminator are trained with adversarial learning, which can be extremely unstable due to the vanishing gradient and mode collapse problem. Several previous works have introduced practical ways of stabilizing the training process (Arjovsky et al., 2017; Gulrajani et al., 2017). We observed that adding instance noise to the discriminator's inputs is sufficient to stabilize our training (Sønderby et al., 2016). Algorithm 1 describes the overall training procedure of our method. The discriminator is updated with RMSprop, and the policy is optimized with PPO (Schulman et al., 2017). The posterior and task transition model are both updated using Adam optimizer (Kingma & Ba, 2014).

B. Derivation Details

B.1. Sub-task Identification

In order to identify sub-tasks, our work maximizes the lower bound of mutual information between sub-tasks and corresponding state–action pairs. The lower bound is instantiated closely following InfoGAN (Chen et al., 2016) except that we use the task transition model as importance sampling distribution instead of the prior over sub-tasks. This allows us to estimate the lower bound $L_I(\pi, T, Q)$ without task-specific knowledge, as discussed in the main paper. The detailed derivation process for the lower bound is as follows.

$$
\begin{aligned}
I(c_t|s_t, a_t) &= \mathbb{E}_{(s_t,a_t)\sim\pi^{c_t}}[\mathbb{E}_{c'_t\sim P(c_t|s_t,a_t)}[\log P(c'_t|s_t, a_t)]] + H(c_t) \\
&= \mathbb{E}_{(s_t,a_t)\sim\pi^{c_t}}[D_{KL}[P(c'_t|s_t, a_t)\|Q(c'_t|s_t, a_t)] + \mathbb{E}_{c'_t\sim P(c_t|s_t,a_t)}[\log Q(c'_t|s_t, a_t)]] + H(c_t) \\
&\geq \mathbb{E}_{(s_t,a_t)\sim\pi^{c_t}}[\mathbb{E}_{c'_t\sim P(c_t|s_t,a_t)}[\log Q(c'_t|s_t, a_t)]] + H(c_t) \\
&= \mathbb{E}_{c_t\sim P(c_t)}\left[\mathbb{E}_{(s_t,a_t)\sim\pi^{c_t}}[\log Q(c_t|s_t, a_t)]\right] + H(c_t) \\
&= \mathbb{E}_{c_t\sim T(c_t|c_{t-1},s_t)}\left[\frac{P(c_t)}{T(c_t|c_{t-1}, s_t)}\,\mathbb{E}_{(s_t,a_t)\sim\pi^{c_t}}[\log Q(c_t|s_t, a_t)]\right] + H(c_t) \\
&= L_I(\pi, T, Q)
\end{aligned}
$$

B.2. Regularization for Preventing Unstable Sub-task Transitions

We enforce a constraint on mutual information between the current sub-task and the current state, conditioned on the previous sub-task. The constraint allows us to ensure that the task transition model is not susceptible to features irrelevant to sub-task transitions. The underlying concept of the regularization technique is information bottleneck, which was first proposed in Tishby et al. (2000) to extract informative representation from an original input. Our work instantiates the constraint by deriving the upper bound of the mutual information, similar to several previous works (Alemi et al., 2016; Achille & Soatto,

[1]ThorDrive, Seoul, South Korea [2]Department of Electrical and Computer Engineering, Seoul National University, Seoul, South Korea. Correspondence to: Sang-Hyun Lee <slee01@snu.ac.kr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

Algorithm 1 Proposed Approach for Learning Compound Tasks

Input: expert demonstrations τ_E, initial parameters of policy, discriminator, posterior, task transition model and dual variable θ^0, ω^0, ψ^0, υ^0 and β^0
for $i = 0, 1, 2, ...$ **do**
Sample initial sub-tasks $c_0 \sim r(c)$
Sample trajectories τ^i using hierarchical policies π_{θ^i} and task transition model T_{υ^i}
Update ω^i to ω^{i+1} with the gradient:

$$\mathbb{E}_{c'_t \sim T_{\upsilon^i}(c_t|c_{t-1},s_t),\, a_t \sim \pi_{\theta^i}(a_t|s_t,c'_t)}[\nabla_{\omega^i} \log D_{\omega^i}(s_t, a_t)] + \mathbb{E}_{\pi^E}[\nabla_{\omega^i} \log(1 - D_{\omega^i}(s_t^E, a_t^E))]$$

Update ψ^i to ψ^{i+1} with the gradient:

$$-\lambda_1 \mathbb{E}_{c'_t \sim T_{\upsilon^i}(c_t|c_{t-1},s_t),\, a_t \sim \pi_{\theta^i}(a_t|s_t,c'_t)}[\nabla_{\psi^i} \log Q_{\psi^i}(c'_t|s_t, a_t)]$$

Update θ^i to θ^{i+1} with the policy gradient method using the following objective:

$$\mathbb{E}_{c'_t \sim T_{\upsilon^i}(c_t|c_{t-1},s_t),\, a_t \sim \pi_{\theta^i}(a_t|s_t,c'_t)}[\log D_{\omega^{i+1}}(s_t, a_t)] - \lambda_1 L_I(\pi_{\theta^i}, T_{\upsilon^i}, Q_{\psi^{i+1}}) - \lambda_2 H(\pi_{\theta^i})$$

Extract sub-task sequences $c_{1:T}$ from the sampled demonstrations $c_t \sim Q_{\psi^{i+1}}(c_t|s_t^E, a_t^E)$
Update υ^i to υ^{i+1} with the gradient:

$$-\mathbb{E}_{c'_t \sim Q_{\psi^{i+1}}(c_t|s_t^E,a_t^E)}[\nabla_{\upsilon^i} \log(T_{\upsilon^i}(c'_t|c_{t-1}, s_t^E))] + \beta^i(\mathbb{E}_{(s_t^E,c_{t-1}) \sim p(s_t^E,c_{t-1})}[\nabla_{\upsilon^i} D_{KL}[T_{\upsilon^i}(c_t|c_{t-1}, s_t^E) \| r(c_t)]])$$

Adjust β^i via dual gradient descent as follows:

$$\beta^{i+1} \leftarrow \max(0,\ \beta^i + \alpha_\beta(\mathbb{E}_{(s_t^E,c_{t-1}) \sim p(s_t^E,c_{t-1})}[D_{KL}[T_{\upsilon^{i+1}}(c_t|c_{t-1}, s_t^E) \| r(c_t)] - I_c))$$

end for

2018). Interestingly, the upper bound can be obtained in a similar manner to a variational autoencoder (VAE). We introduce the variational approximation $r(c_t)$ to the marginal distribution $p(c_t|c_{t-1}) = \int T(c_t|c_{t-1}, s_t)p(s_t)\, ds$, which allows us to estimate the uppper bound without evaluating the intractable marginal $p(c_t|c_{t-1})$ by leveraging the non-negativity of the Kullback-Leiber (KL) divergence. We model $r(c_t)$ as a normal distribution to compute the KL divergence analytically. The overall procedure of the derivation is described below.

$$\begin{aligned}
I(c_t, s_t|c_{t-1}) &= \int_C \Big[\int_C \int_S p(s_t, c_t|c_{t-1}) \log \frac{p(s_t, c_t|c_{t-1})}{p(s_t|c_{t-1})p(c_t|c_{t-1})} dsdc \Big] p(c_{t-1})dc \\
&= \int_C \Big[\int_C \int_S p(s_t|c_{t-1})T(c_t|c_{t-1}, s_t) \log \frac{T(c_t|c_{t-1}, s_t)}{p(c_t|c_{t-1})} dsdc \Big] p(c_{t-1})dc \\
&= \int_C \Big[\int_C \int_S p(s_t|c_{t-1})T(c_t|c_{t-1}, s_t) \log \frac{T(c_t|c_{t-1}, s_t)}{r(c_t)} dsdc \Big] p(c_{t-1})dc \\
&\qquad + \int_C \Big[\int_C \int_S p(s_t|c_{t-1})T(c_t|c_{t-1}, s_t) \log \frac{r(c_t)}{p(c_t|c_{t-1})} dsdc \Big] p(c_{t-1})dc \\
&= \int_C \int_S p(s_t, c_{t-1}) D_{KL}[T(c_t|c_{t-1}, s_t) \| r(c_t)] dsdc \\
&\qquad + \int_C \Big[\int_C p(c_t|c_{t-1}) \log \frac{r(c_t)}{p(c_t|c_{t-1})} dc \Big] p(c_{t-1})dc \\
&= \mathbb{E}_{(s_t,c_{t-1}) \sim p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1}, s_t) \| r(c_t)]] - \underbrace{\mathbb{E}_{c_{t-1} \sim p(c_{t-1})}[D_{KL}[p(c_t|c_{t-1}) \| r(c_t)]]}_{\geq 0} \\
&\leq \mathbb{E}_{(s_t,c_{t-1}) \sim p(s_t,c_{t-1})}[D_{KL}[T(c_t|c_{t-1}, s_t) \| r(c_t)]]
\end{aligned}$$

C. Implementation Details

C.1. Model Architecture

Here, we describe the structure of the models included in our method. The discriminator and policy have the same network structure described in Ho and Ermon (2016). The policy network outputs a categorical distribution over actions for discrete tasks, whereas it outputs the mean and standard deviations of a Gaussian distribution for continuous tasks. The posterior and the task transition model have two layers of 100 units with ReLU activations. Since the task transition model should learn the long-term dependencies between sub-tasks, it is followed by GRU (Cho et al., 2014), which is a recurrent neural network that leverages a gated architecture.

We evaluate our method against several baselines: BC, GAIL, and CVAE. CVAE consists of an encoder and a decoder. The encoder takes a state and an action as inputs and returns a latent variable that encodes sub-tasks. The decoder, which is the policy in our case, takes the latent variable and state as inputs and returns an action. The states are regarded as conditional variables in the training procedure for CVAE. To provide a fair comparison, the policy networks for each baseline and the discriminator network in GAIL have the same architecture as in our method.

C.2. Hyperparameter Setting

Table 1 describes the hyperparameters used for our experiments. These hyperparameters were tuned through a coarse grid search, e.g., policy learning rate over $\{0.00001, 0.00003, 0.0001, 0.0003, 0.001\}$, entropy coefficient over $\{0.0, 0.001, 0.01, 0.1\}$, and mini-batch size over $\{32, 64, 128, 256, 512\}$.

HYPERPARAMETER	VALUE
PPO CLIPPING FACTOR	0.2
DISCOUNT FACTOR	0.99
GAE PARAMETER	0.95
VALUE FUNCTION COEFFICIENT	0.5
ENTROPY COEFFICIENT	0.01
HORIZON	2048
MINI-BATCH SIZE	64
ADAM β_1	0.9
ADAM β_2	0.999
LEARNING RATE(POLICY)	0.0003
LEARNING RATE(OTHERS)	0.0001
I_C	0.2
β_0	0.0

Table 1. Hyperparameters

C.3. Environment Description

Table 2 provides further details about the tasks we used for our experiments. Although we do not evaluate our method on MountainCar and MountainCarContinuous, we add them to the table for comparison with the newly introduced tasks, which we call MountainToyCar and MountainToyCarContinuous.

ENVIRONMENT	STATE	ACTION	TASK TYPE	MAXIMUM STEP
MOUNTAINCAR	2	3(DISCRETE)	PRIMITIVE	200
MOUNTAINTOYCAR	1	3(DISCRETE)	COMPOUND	200
MOUNTAINCARCONTINUOUS	2	1(CONTINUOUS)	PRIMITIVE	999
MOUNTAINTOYCARCONTINUOUS	1	1(CONTINUOUS)	COMPOUND	999
FETCHPICKANDPLACE	28	4(CONTINUOUS)	COMPOUND	50
HOPPER	11	3(CONTINUOUS)	PRIMITIVE	1000
HALFCHEETAH	17	6(CONTINUOUS)	PRIMITIVE	1000
WALKER2D	17	6(CONTINUOUS)	PRIMITIVE	1000

Table 2. Benchmark tasks

References

Achille, A. and Soatto, S. Information dropout: Learning optimal representations through noisy computation. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2897–2905, 2018.

Alemi, A. A., Fischer, I., Dillon, J. V., and Murphy, K. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223, 2017.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sønderby, C. K., Caballero, J., Theis, L., Shi, W., and Huszár, F. Amortised map inference for image super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.