Learning to Compose Skills

Himanshu Sahni, Saurabh Kumar, Farhan Tejani, Charles Isbell

College of Computing, Georgia Institute of Technology

contact: himanshu@gatech.edu

Abstract

We present a differentiable framework capable of learning a wide variety of compositions of simple policies, which we call skills. By recursively composing skills with themselves, hierarchies can be created that display complex behavior. Skill networks are trained to generate skill-state embeddings which are provided as inputs to a trainable composition function, which in turn outputs a policy for the overall task. Our experiments on an environment consisting of multiple collect and evade tasks show that this architecture is able to quickly build complex skills from simpler ones. The learned composition function displays transfer to unseen combinations of skills, allowing for zero-shot generalizations.

Background

Reinforcement Learning

- Maximize a notion of long term reward
- Typically formulated as an MDP <S, A, T, R, γ>

Policy Gradient

$$J(\theta) = \mathbb{E}_{p(S_{1:T};\theta)} \sum_{t=1}^{T} r_t = \mathbb{E}_{p(S_{1:T};\theta)}[R]$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{p(S_{1:T};\theta)} \sum_{t=1}^{T} \nabla_\theta log[\pi(a_t|s_{1:t};\theta)]R_t$$

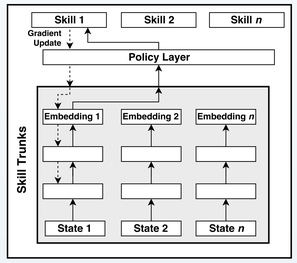
Linear Temporal Logic

- $\mathcal{U}$ - until
- $\Diamond$ - eventually
- $\Box$ - always
- $\mathcal{O}$ - next
- Combined with basic logical connectives

$\vee \quad \wedge \quad \neg$

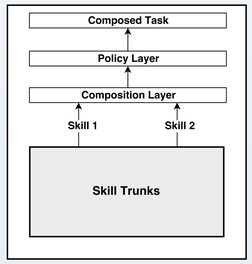
Can specify a wide variety of RL tasks

ComposeNet



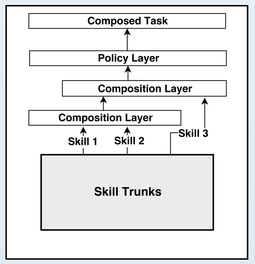
Skill trunks are trained with common policy layer

This forces topmost layer of skill trunks to embed skill-state information.



Trunk and policy layer weights are frozen.

Skill-state embeddings are composed and fed into policy layer.

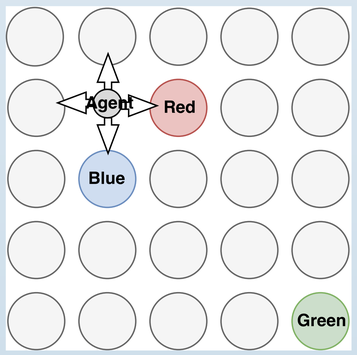


Hierarchies of compositions can be created post-hoc.

$$C: \langle \iota_e^{(1)}(S), \iota_e^{(2)}(S), \ldots \iota_e^{(n)}(S) \rangle \rightarrow \iota_c(S)$$

$$\pi_k: \iota_e^{(k)}(S) \rightarrow p(A)$$

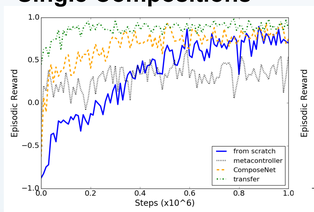
Environment



Agent must collect target objects and evade enemies.

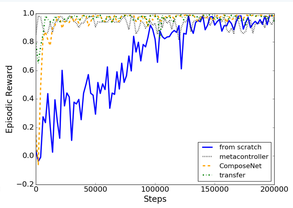
Results

Single Compositions



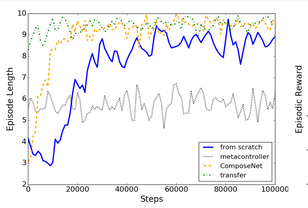
$\neg g \mathcal{U} b$

zero-shot: 0.45



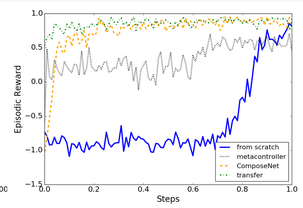
$\Diamond r \vee \Diamond b$

zero-shot: 0.79



$\Box \neg r \wedge \Box \neg g$

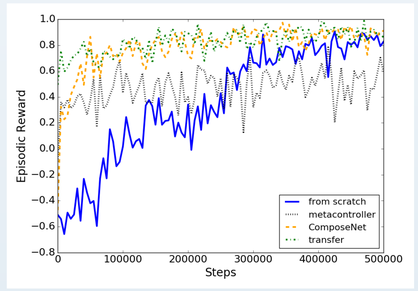
zero-shot: 8.28



$\Diamond(r \wedge \Diamond g)$

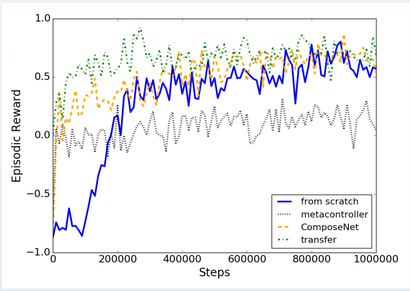
zero-shot: 0.53

Hierarchical Compositions



$\neg b \mathcal{U} (r \vee g)$

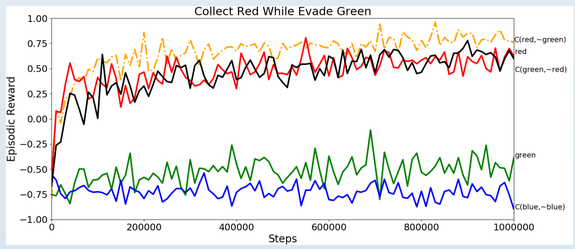
zero-shot: 0.49



$(\neg g \wedge \neg b) \mathcal{U} r$

zero-shot: 0.01

Ablations



Conclusion and Future Work

- **Individual skills can be successfully composed with the ComposeNet architecture.**
- **Learned faster than either of the baselines.**
- **Some zero-shot generalization, can be fine tuned to optimality.**
- **Hierarchies can be created by recursive compositions.**
- **Key to this are skill-state embeddings and a trainable composition function.**
- **Stay tuned for more compositions and environments!**