SHARING SKILLS IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Preserving past knowledge while learning new tasks over non-stationary task distributions is the main challenge in continual learning. We demonstrate that the option-critic architecture can be formulated through a modular composition of options in a flexible manner to address the challenge. Specifically, we leverage the specialization capabilities of Mixtures of Experts and the transfer-interference trade-off to allow learning skills into forward and backward directions in time. We demonstrate the benefits of this marriage of approaches in accelerating learning and preventing catastrophic forgetting while learning new tasks.

1 INTRODUCTION

Continual learning is the ability of a model to learn continuously from streams of data, capable not only remembering previously seen tasks and preventing catastrophic forgetting, but also allowing for forward transfer of knowledge. Continual learning is especially important in solving the broader problem of general intelligence, where adapting quickly to different tasks and reusing experiences is essential. However, the continual learning paradigm does not operate well in delayed and sparse reward domains, where the signal for learning must be frequent and consistent for gradient alignment as shown in standard baselines for continual learning.

On the other hand, the option-critic architecture, one of the major frameworks in hierarchical learning, provides a theoretically justified policy gradient theorem for learning options and their temporal sequencing with modern deep networks. While option-critic has been successfully applied to stationary multi-task settings (Riemer et al., 2018) due to its ability to navigate pareto optimality, it is yet to be applied in continual learning settings. The option-critic is not well suited to such a setting as it does not have a natural prior to prioritize old tasks that do not reoccur. Additionally, option-critic is highly impoverished in its ability to navigate the trade-off between transfer and interference (Riemer et al., 2019) that is critical to achieving strong performance during continual learning. As pointed out by Barreto et al. (2019), standard approaches to option-critic learning do not provide a paradigm for deciding the degree to which and how options are related to each other. Learning to share skills in non-stationary task settings enables adaptation in the current task while preserving past knowledge. Without a sharing mechanism, adapting knowledge with every new task requires many training iterations for fine-tuning and/or learning skills from scratch.

Contribution. In this paper, we propose learning to balance plasticity and catastrophic forgetting in the hierarchical reinforcement learning framework. The approach of our paper is a novel combination of the option-critic architecture with mixtures of experts in continual learning settings. This allows options to self organize in a modular way that allows options to have flexible and compositional relations to each other. Additionally, by providing option-critic with supervision about transfer and interference dynamics across tasks, we provide direct incentive for it to share parameters across skills as much as possible to maximize transfer and orthogonalize parameters as much as possible to minimizing interference from conflicting tasks. Perceived benefits of this approach include enabling sharing between relevant tasks to accelerate learning, preventing catastrophic forgetting while learning in new tasks, and enabling learning in delayed reward settings. This marriage of continual learning and hierarchical reinforcement learning results in strong synergies which overcomes the deficiencies that each approach would have individually. We test our approach

on the Mujoco environment, experimenting with non-stationary in the environment to determine the agent's capabilities of transferring across tasks.

2 BACKGROUND AND NOTATION

Hidden-Mode Markov Decision Process. A Hidden-Mode Markov Decision Process (HM-MDP) (S. Choi & Zhang, 1999) is represented as a 8-tuple $\langle \mathcal{Q}, \mathcal{S}, \mathcal{A}, \mathcal{X}, \mathcal{Y}, \mathcal{R}, \pi, \Psi \rangle$, where $\mathcal{Q}, \mathcal{S}$ and $\mathcal{A}$ represent the sets of modes, states and actions respectively; the mode transition function X maps mode m to n with a fixed probability $x_m n$; the state transition function Y defines transition probability, $y_m(s; a; s_0)$, from state s to s_0 given mode m and action a, the stochastic reward function R returns rewards with the mean value $r_m(s, a)$ and π and Ψ denote the prior probabilities of the modes and the states respectively. We define an HM-MDP as a finite set of MDPs that share the same state space and action space, with possibly different transition functions and reward functions. The MDPs correspond to different modes, and those that are not directly observable with transitions are controlled by a Markov chain. We define the HM-MDP framework when the underlying task distribution is unknown and when the environment is partially observable.

Options framework. R. S. Sutton & Singh (1999) introduces a temporarily coherent framework to learn a series of actions represented by an option. A Markovian option $w \in \omega$ is a triple (I_w, π_w, β_w) where $I_w \subseteq S$ represents an initiation set, π_w represents an intra-option policy, and $\beta_w : S\text{ß}[0, 1]$ represents a termination function. MDP with options become SMDPs with an optimal value function over options $V_\omega(s)$ and option-value function $Q_\omega(s, w)$.

Option critic architecture. P. L. Bacon & Precup (2017) defines a Markovian option $\omega \in \Omega$ as a triple $(I_\omega, \pi_\omega, \beta_\omega)$ in which $I \in S$ is an initiation set, π_ω is an intra-option policy and β_ω is a termination function. An option w is selected according to a policy over options $\pi_\Omega(\omega|s)$ where Ω is the set of possible options. The option-value function can be written as:

$$Q_\Omega(s, w) = \sum_a \pi_{\omega,\theta}(a|s) Q_U(s, \omega, a), \tag{1}$$

where $Q_U(s, \omega, a) = r(s, a) + \gamma \sum_{s'} P(s'|s, a) U(\omega, s') U(\omega, s)$ is called the option-value function upon arrival. The value of executing ω upon entering a state s' is given by:

$$U(\omega, s') = (1 - \beta_{\omega,}(s')) Q_\omega(s', \omega) + \beta_{\omega,}(s') V_\omega(s'). \tag{2}$$

Given a set of Markov options with stochastic intra-option policies differentiable in their parameters θ, the gradient of the expected discounted return with respect to θ and initial condition (s_0, ω_0) is:

$$\sum_{s,\omega} \sum_{t=0}^{\infty} \gamma^t P(s_t = s, w_t = w|s_0, w_0) \sigma_a \frac{\partial \theta_{\omega,\theta}(a|s)}{\partial \theta} Q_U(s, \omega, a) \tag{3}$$

Given a set of set of Markov options with stochastic termination function differentiable in their parameters , the gradient of the expected discounted return objective with respect to ϑ and the initial condition (s_1, ω_0) is

$$\sum_{s',\omega} \sum_{t=0}^{\infty} \gamma^t P(s_{t+1} = s', w_t = w|s_1, w_0) \frac{\partial \beta\omega, \vartheta(s')}{\partial \vartheta} A_\Omega(s', \omega) \tag{4}$$

where $A_\Omega(s', \omega)$ is the advantage function over options.

Mixture of Experts. Jacobs et al. (1991) layer learns individual sets of parameters for a series of K expert networks $e_1, e_2...e_K$ and a gating network g, whose output is an K-dimensional vector. Each expert is a neural network with its own set of parameters, accepting the same sized input and producing the same sized output. We define g(x) to choose a sparse weighted combination of experts, representing a number of functions including a non-sparse gating function, a softmax function, amongst many others.

The final output is given by:

$$y = \sum_{k=1}^{K} g_k(x) e_k(x), \tag{5}$$

where wherever $g_k(x) = 0$, we need not to consider the output of k-th expert $e_k(x)$.

Meta Experience Replay. Riemer et al. (2019) solves the continual learning problem by addressing the temporally symmetric trade-off between transfer and interference that can be optimized by enforcing gradient alignment across examples. This method learns parameter θ that make interference based on future gradients less likely and transfer based on future gradients more likely via meta-learning. Specifically, transfer between two samples (x_i, y_i) and (x_j, y_j) occurs when:

$$\frac{\partial \mathcal{L}(x_i, y_i)}{\partial \theta} \cdot \frac{\partial \mathcal{L}(x_j, y_j)}{\partial \theta} > 0 \tag{6}$$

where $\cdot$ is the dot product operator. Interference occurs when:

$$\frac{\partial \mathcal{L}(x_i, y_i)}{\partial \theta} \cdot \frac{\partial \mathcal{L}(x_j, y_j)}{\partial \theta} < 0 \tag{7}$$

As a result, MER is able to learn parameter θ by encouraging gradient alignment - namely if the two samples are not similar, we penalize through the loss function as shown below.

$$\theta = \arg\min_{\theta} \mathbb{E}_{[(x_i,y_i),(x_j,y_j)]D} \Big[L(x_i, y_i) + L(x_j, y_j) - \alpha \frac{\partial \mathcal{L}(x_i, y_i)}{\partial \theta} \cdot \frac{\partial \mathcal{L}(x_j, y_j)}{\partial \theta} \Big] \tag{8}$$

3 Soft Option Critic

We re-formulate the option critic framework to learn the maximum-entropy intra-option policies and define soft option policy evaluation and soft option improvement. The new objective to maximize the discounted return and entropy over actions for states expected over all trajectories:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{\tau} \Big[r_t + \underbrace{\alpha_\theta \mathcal{H}(\pi_{\omega,\theta}(\cdot|s_t))}_{\text{Intra-Policy Entropy}} \Big] \tag{9}$$

where τ denotes each sampled trajectory, intra-option policy of ω is parametrized by θ, and α_θ represents the entropy constant for the intra-option level.

As done in option-critic, we will employ the the call-and-return option execution model, in which an agent picks option ω according to its policy over options π_Ω, then follows the intra-option policy $\pi_{\omega,\theta}$ until termination dictated by $\beta_{\omega,v}$, a process which is repeated for a number of steps.

We will use function approximators to learn the inter-Q function $Q_\psi(s_t, \omega_t)$, intra-Q function $Q_\phi(s_t, \omega_t, a_t)$, intra-option policies $\pi_{\omega,\theta}(s_t, a_t)$, and termination policies $\beta_{\omega,v}(s_t)$. We do not learn the inter-option policy and derive option ω from the inter-option Q function as in the option-critic paper. The entropy at the inter-option level is also ignored as the the policy over options is deterministic i.e. exploiting Q.

Definitions In order to take the gradients with respect to ψ, θ, ϕ and v for inter-option, intra-option, and termination levels, we must re-define equations similar to those used in option critic by including entropy maximization in their objectives. We define the inter-option value function, given a state s_t and option ω_t:

$$Q_\psi(s_t, \omega_t) = \mathbb{E}_{a_t}\Big[Q_\phi(s_t, \omega_t, a_t) - \alpha_\theta \log \pi_{\omega,\theta}(s_t, a_t)\Big] \tag{10}$$

where $Q_\phi : \mathcal{S} \times \Omega \times \mathcal{A} \to \mathbb{R}$ is the value of executing an action in the context of a state-option pair. We define the intra-option value function, given a state s_t, option ω_t, and action a_t:

$$Q_\phi(\omega_t, s_t, a_t) = r_t + \gamma \sum_{s_{t+1}} P(s_{t+1}|s_t, a_t) U(\omega_t, s_{t+1}), \tag{11}$$

$$U(\omega_t, s_{t+1}) = (1 - \beta_{\omega,v}(s_{t+1}))Q_\psi(s_{t+1}, \omega_t) + \beta_{\omega,v}(s_{t+1})V_\psi(s_{t+1}), \tag{12}$$

$$V_\psi(s_{t+1}) = \max_{\omega_t} Q_\psi(s_{t+1}, \omega_t) \tag{13}$$

We use the max for the value function in the epsilon-greedy case as per the option critic paper.

3.1 Soft Option Learning

We formulate SOC to learn the intra-option policies $\pi_{\omega,\theta}(s_t, a_t)$, the termination policies $\beta_{\omega,v}(s_t)$, intra-Q function $Q_\phi(s_t, \omega_t, a_t)$ and inter-Q function $Q_\psi(s_t, \omega_t)$. We alternate between optimizing the networks with stochastic gradient descent as in SAC.

3.2 Soft Option Evaluation

The inter-option Q value function trained to minimize the squared residual and optimized as follows:

$$J_Q(\psi) = \mathbb{E}_{s_t,\omega_t \sim D}\Big[\frac{1}{2}\Big(Q_\psi(s_t, \omega_t) - \mathbb{E}_{a_t \sim \pi_{\omega,\theta}}\big[\bar{Q}_\phi(s_t, \omega_t, a_t) - \alpha_\theta \log \pi_{\omega,\theta}(s_t, a_t)\big]\Big)^2\Big], \tag{14}$$

where $\bar{Q}_\phi$ is the target intra-Q function.

The intra Q-function is trained to minimize the soft Bellman residual and optimized as follows:

$$\begin{aligned} J_Q(\phi) = \mathbb{E}_{s_t,\omega_t,a_t,r_t,s_{t+1} \sim D}\Big[\frac{1}{2}\Big(Q_\phi(s_t, \omega_t, a_t) - \big(r_t + \gamma \mathbb{E}_{s_{t+1}}\big[(1 - \beta_{\omega,v}(s_{t+1}))\bar{Q}_\psi(s_{t+1}, \omega_t) + \\ \beta_{\omega,v}(s_{t+1})\mathbb{E}_{w_{t+1} \sim \pi_\Omega}\big[\bar{Q_\psi}(s_{t+1}, \omega_{t+1})\big]\big)\Big)^2\Big] \end{aligned} \tag{15}$$

where $\bar{Q}_\psi$ is the target inter-Q function. We sample the beta probability from the current policy. The next-option used in the target inter-Q function come from the current inter-option policy.

3.3 SOFT POLICY IMPROVEMENT

The intra-option policy represents the probability density of option given state for option ω, optimized as follows:

$$\begin{aligned} J_{\pi_{\omega,\theta}}(\theta) &= \mathbb{E}_{(s_t,\omega_t)\sim D, a_t\sim\pi_{\omega,\theta}}\Big[\log_{\pi_{\omega,\theta}}(s_t,a_t) - Q_\phi(s_t,\omega_t,a_t)\Big] \\ \widetilde{a}_t &= \tanh(\mu_{\omega,\theta}(s_t) + \sigma_{\omega,\theta}(s_t)\odot\epsilon), \epsilon\sim\mathcal{N}(0,1) \end{aligned} \tag{16}$$

We apply the reparametrization trick at the intra-option policy level in which a sample from $\pi_{\omega,\theta}$ is drawn by computing a deterministic function of state, policy parameters, and independent noise. Similar to SAC, we use a squashed Gaussian policy to ensure the actions are bounded to a finite range.

The termination policy objective is defined as follows:

$$\begin{aligned} J_\beta(\omega,v) &= \sum_{\omega_t,s_{t+1}\sim D} \beta_{\omega_t,v}(s_{t+1}) A_\psi(s_{t+1},\omega_t) \\ A_\psi(s_{t+1},\omega_t) &= Q_\psi(s_{t+1},\omega_t) - V_\psi(s_{t+1}) \\ &= Q_\psi(s_{t+1},\omega_t) - \max_{\omega_t} Q_\psi(s_{t+1},\omega_t) \end{aligned} \tag{17}$$

4 APPROACH

4.1 KEY RELATED WORK

While mixtures of experts ((Jordan & Jacobs, 1994)) has long remained a popular approach for learning to self organize neural network structure, its application to reinforcement learning has been quite limited in comparison to the success it has had in the supervised learning literature. One reason for this, is that mixtures of experts on their own provide a solution to decomposition in parameter space, but do not all address the issue of decomposition in time. The idea of rapidly switching active modules from time step to time step is very incoherent in the framework of reinforcement learning, particularly when task switches are relatively infrequent. Combining mixtures of experts with option-critic directly addresses the issue of sequencing in time with a theoretically justified policy gradient theorem. Additionally, as pointed out by (Rosenbaum et al., 2018; 2019) mixtures of experts on their own have minimal ability to adjust for transfer and interference dynamics as we backpropogate through the connection to all experts at each step. By combining mixtures of experts with meta-experience replay we provide the network with direct supervision about transfer and interference dynamics. We hypothesize this will make the network better able to self organize in a way that navigates this trade-off efficiently.

While meta-experience replay ((Riemer et al., 2019)) has achieved impressive results for continual learning across tasks that are highly conceptually related by providing supervision to the network about how to navigate transfer and interference dynamics, the approach remains limited in several respects. First, it is limited in its ability to address pareto optimality as it requires a single policy to solve each task. As discussed above, this issue is addressed by integrating with option-critic. Furthermore, generic meta-experience replay has supervision about transfer and interference dynamics while having very little leverage with which it can self organize its structure to address these dynamics Integrating with mixtures of experts addresses this point as well.

4.2 PROBLEM SETTING

In this paper we consider the problem of continual reinforcement learning in which our agent must perform incremental learning over a non-stationary distribution of tasks $T \in \mathbb{T}$. We further assume that the distribution of tasks T evolves and is unaffected by the behaviour of the agent. Each task defines a new environment with a unique transition function $T_T(s'|s,a)$ and reward function $R_T(r|s,a)$ governing MDP dynamics. The problem of learning arbitrarily non-stationary MDPs is intractable, so it is important that we make further

assumptions to develop an approach that will work in general. We assume that while tasks may evolve in such a way that we only take *one pass* through each task and never revisit them again during training, we will be evaluated on our retained performance on the stationary distribution of tasks seen before. This is an important assumption of most work on continual learning because it must be addressed by inserting a prior into the algorithm. Data driven approaches alone do not have the potential to learn that past tasks will reoccur during evaluation as they do not see evidence of this during training. In this paper, we follow (Riemer et al., 2019) and address this issue by leveraging experience replay during continual learning.

It is also important that we make some assumptions about the nature of the relation between different tasks. Because each task constitutes its own environment with specific dynamics, it is possible that tasks may conflict with each other such that we run into issues of *pareto optimality*. This implies that it may be impossible to learn a single policy that simultaneously solves each task. As such, we take the strategy of maintaining a set of *skills*, which we formalize as *options* $O \in \mathbb{O}$. If each option is applied at the correct point in time, it thus may be possible to simultaneously solve tasks with the same agent even if they induce conflicting optimal policies. In this paper, we assume that task relations are not arbitrary and that they are in fact compositionally related to each other. This in turn implies that while the number of tasks our agent needs to solve may be very large, it should be possible to solve these tasks by decomposing them into a much smaller set of skills i.e. $\mathbb{T} >> \mathbb{O}^*$.

4.3 Modular Option Critic

As opposed to standard option-critic that has a fixed user defined architecture, we would like to provide it with the capability for it to self organize the architecture of each component of the agent. This can be addressed by setting at each layer a shared set of modules that are shared among each component. Then for each component π_Ω,π_ω, and β_ω we learn a gating network specific to each component at each layer. This allows us to fully customize the amount of sharing and compositional relation of sharing across each part of our architecture.

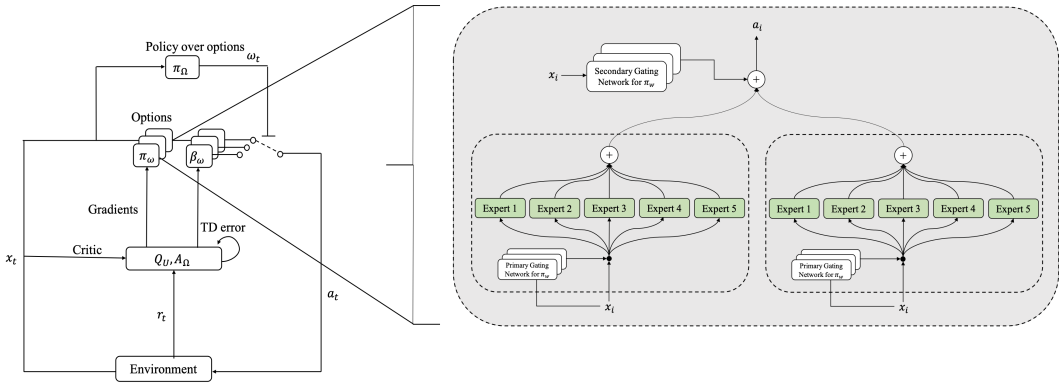

Figure 1: Diagram of the option-critic architecture adapted in our approach by representing each intra-option policy π_ω as a single Mixture of Experts network. Gating function selects expert 1 and 5 for the current task. $K = 5$ is shown for clarity, but we scale to thousands of experts in our experiments.

We incorporate MoE due to it benefits in prioritizing specialization and permitting the network to display a greater variety of behaviors. This is essential in a series of evolving and non-stationary task distributions, where the policy must leverage a different sub-selection of parameters to solve each task. The hierarchical nature of the option-critic architecture is not sufficient to encode this complexity as seen by the relearning of policy of parameters at every new task distribution.

We define a buffer $\mathcal{D}$ containing all previous tasks across t steps, with each task represented as (x, ω, a). Given tasks i and j and $n \in N$ options, we minimize the loss of the current task i while maximizing transfer and minimizing interference across experts of all intra-option policy of a previous task j. Each intra-option policy will have its own gating network, but the hierarchical MoE experts are shared across options. Each expert is represented as two fully connected layers, and input to experts are observation space of x_i and x_j consequently. We optimize the parameter θ^* as follows:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{((x_i,\omega_i,a_i),(x_j,\omega_j,a_j)\sim\mathcal{D})} \Big[\mathcal{L}(x_i) + \mathcal{L}(x_j) - \alpha \frac{\partial \mathcal{L}(x_i)}{\partial \theta} \frac{\partial \mathcal{L}(x_j)}{\partial \theta} \Big] \tag{18}$$

5 Related Works

Hierarchical Reinforcement Learning (HRL) derives inspiration from biological and evolutionary processes to decompose many real world tasks into natural hierarchical structures. There have been two major approaches to HRL; the options framework to learn higher level skills (R. S. Sutton & Singh (1999)) and goal-conditioned hierarchies to learn higher level sub-goals (Kulkarni & Narasimhan (2016)). The former model-based paradigm however requires a predefined representation of the environment and mapping of observation space to goal space, hence failing in the non-stationary case due to complexity. The option-critic formulation on the other hand facilitates the long timescale credit assignment by dividing the problem into pieces, learning higher level skills to solve a certain task and shows promise in being extended to the non-stationary case where these tasks are ever changing. We choose this framework as it is best for our needs of operating well in delayed reward settings.

6 Discussion

The approach of our paper is a novel combination of three popular existing approaches in the literature. We hope to define more complex formulation of non-stationary in future work, however we hypothesize that this could lead to components that learn to all use the exact same parameters or all components to not share parameters at all. Future work would also include experimenting on robotics applications where there is a natural hierarchy present in the problem.

7 APPENDIX

7.1 SOFT OPTION CRITIC ALGORITHM

We learn two Q-functions for the the intra-Q function $Q_{\phi_1}(s_t, \omega_t, a_t)$ & $Q_{\phi_2}(s_t, \omega_t, a_t)$, and two Q functions for the inter-Q function $Q_{\psi_1}(s_t, \omega_t)$ & $Q_{\psi_2}(s_t, \omega_t)$. Similar to SAC, this helps mitigate positive bias in the policy improvement step that is known to degrade performance of the value based method.

The Q-functions for intra-option policy are learned with MSBE minimization, by regressing to a single shared target $\bar{Q}_\phi$. The shared target is computed using two target Q-networks, and the target Q-networks are obtained by polyak averaging the Q-network parameters over the course of training. We use this double-Q trick for both intra-Q functions and inter-Q functions.

REFERENCES

André Barreto, Diana Borsa, Shaobo Hou, Gheorghe Comanici, Eser Aygün, Philippe Hamel, Daniel Toyama, Shibl Mourad, David Silver, Doina Precup, et al. The option keyboard: Combining skills in reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 13031–13041, 2019.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

T. D. Kulkarni and K. R. Narasimhan. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. *Neural Information Processing Systems 2016*, 2016.

J. Harb P. L. Bacon and D. Precup. The option-critic architecture. *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

D. Precup; R. S. Sutton and S. P. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence 112*, 1999.

Matthew Riemer, Miao Liu, and Gerald Tesauro. Learning abstract options. *NIPS*, 2018.

Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *ICLR*, 2019.

Clemens Rosenbaum, Tim Klinger, and Matthew Riemer. Routing networks: Adaptive selection of non-linear functions for multi-task learning. *ICLR*, 2018.

Clemens Rosenbaum, Ignacio Cases, Matthew Riemer, and Tim Klinger. Routing networks and the challenges of modular and compositional computation. *arXiv preprint arXiv:1904.12774*, 2019.

D. Yeung S. Choi and N. Zhang. Hidden-mode markov decision processes. *IJCAI Workshop on Neural, Symbolic, and Reinforcement Methods for Sequence Learning*, 1999.