In this paper, we propose to tackle the problem of multi-task reinforcement learning with a novel modular network model. Instead of using hard selection on modules, we introduce a method called *soft modularization* which softly combines the modules. Our approach enables efficient optimization and sharing across modules. The role of each module can automatically emerge after training without manual specification. We perform extensive **empirical** studies and show significant improvement (**>20%** success rate) over state-of-the-art approaches in the robot manipulation tasks (50 multi-task). We are glad to receive positive feedbacks on our work for both the novelty and the performance: R1:"motivate their work very well, is technical sound," R2:"idea seems to be new," R3:"very important problem, better structure sharing," R4: "the experimental results are very good." We will address reviewers' comments as follows.

—————————————— For Reviewer #2 ——————————————

Theoretical grounding: the paper is not well grounded in neural network theory. While theory is important, our work is focusing on designing novel a modular network for multi-task RL and **empirically** showing its advantage. We are confused about what "neural network theory" R2 is asking for. R2 also asks "Why a dot product for the weighting?" But weighting itself indicates multiplication. R2 has not provided an alternative way for weighting.

Meta-learning is attracting, Comparison to state-of-the-art (e.g. MAML). R2 may have misunderstood that our work with meta-learning. We emphasize here that this paper is tackling **multi-task** learning but not meta-learning. Thus meta-learning approaches like MAML do not apply. We also stress that we have compared to all the baselines we can find codes for and implement including Mixture of Experts [12], Hard Routing [29], and Muti-task Muti-head SAC [43], which is the previous state-of-the-art. R2 also has not provided a reference on multi-task RL for us to compare.

Comparing to hierarchical architectures like capsule networks. Capsule networks have not been applied to multi-task RL. How to adopt it in multi-task RL is an interesting direction to study, but it is out of the scope of our paper.

Writing. While R2 complains about our writing, other reviewers all have positive feedback: "I liked to read the paper, as it is well written"(R1), "The paper is well-written and easy to understand"(R3), "The paper is written well."(R4).

—————————————— For Reviewer #4 ——————————————

Similar idea with Rosenbaum et al. We have extensively addressed the difference between our work and Rosenbaum et al. in both related work (line 83-90) and compare against it in the experiment (denoted as Hard Routing [29]). Our soft modularization approach improves over it in both sample efficiency and performance significantly.

Similar Variance for Hard Routing and Ours. Suffering from higher variance in policy gradient, the success rate (22.9% for MT50) of Hard Routing is significantly lower than our approach (60.0% for MT50). Although the result of Hard Routing has similar variance as our method, this only means all their results are equally bad. In an extreme case, multiple random policies will have 0% accuracy but zero variance. Thus higher variance in gradient does not necessarily convert to higher variance in performance.

Gating/Masking modules instead of routing. We can see gating modules as a special case of routing mechanism, where all the routes connected to the same module will be weighted by the same scale parameter. Our routing network in the paper allows the routes connected to the same module be weighted differently, leading to better flexibility.

Training of routing network. As we mentioned in our paper (line 131-132), the soft combination method we proposed is fully differentiable, so both our base policy network and routing network can be trained together end-to-end.

—————————————— For Reviewer #1 ——————————————

Modular Structure for Q function. For the base network, we concat state and action then feed them as inputs, and output the value. For the routing network, the inputs are the same as the policy including both states and task embedding.

Sharing Data. Sharing data will be an interesting future direction, and it is complementary to our current approach.

—————————————— For Reviewer #3 ——————————————

High dimensional inputs like images. While learning with image inputs is an exciting direction, it is unclear how to learn a good visual representation for RL, which is a common challenge for vision-based RL. Recent solutions involve self-supervised visual representation learning, which introduces extra complexity. We will study this in the future.

How many layers should be routed? We study this problem in two directions: (i) Increasing the routing layers: we have compared our model with 2 routing layers (Ours (Shallow)) and 4 routing layers (Ours (Deep)) in our experiments, and we find improvement by using 4 layers in MT50 tasks. (ii) Increasing the layers before routing starts: We experiment with different number of layers of FC (2,3,4 layers) before routing starts and do not observe obvious performance difference. The reason might be the current input states are in low-dimension. For high dimension visual inputs, we hypothesize that we can first use ConvNets to extract the visual representation in lower-dimension, and then apply our routing modular networks on top of the extracted representation.

Comparing to FiLM (Perez et.al). This works predicts input conditioned feature, and our work predicts task conditioned network routing. While related, they are also tackling very different tasks. We will include and discuss this paper in our related work.