We would like to thank the reviewers for their insight and suggestions. We will aim to incorporate your feedback in future drafts of the paper.

Re: primitive specializations

We would like to clarify that we do not manually assign skills to each primitive. As mentioned at the end of section 3.1 and the start of 3.2, the primitives are trained jointly, and the specializations emerge automatically from the learning process. For example, we do not assign one primitive for walking, and then another for turning. Instead the primitives learn to specialize in different low-level skills that can then be composed to produce different walks, turns, or kicks. We will adjust our writing to improve the clarity on this point.

Re: code release

If accepted, we intend to release the code for MCP, the environments, and the motion dataset used for pre-training.

Individual Responses:

Reviewer 3:

Re: Gaussian primitives

The formulation of MCP in Equation 2 is not restricted to Gaussian primitives. Other distributions can also be used, as long as the product of the primitives produce a tractable distribution. We chose to use Gaussian primitives in this work, because they are widely used for continuous control tasks, and provide a simple analytic form for the composite distribution, since the product of Guassians is another Gaussian. We will include a proof of this property.

Re: activating multiple primitives with the additive model

Thank you for raising this point, we will improve the description of our method to better clarify this. The crucial difference that enables MCP to activate multiple primitives simultaneously is indeed the multiplicative composition scheme. Adding the densities of the primitive distributions produces a mixture policy, where the agent selects a single primitive per time step. Therefore, the action at any given time step would come from only one of the primitives. Multiplying the primitives together instead fuses their distributions, intuitively producing a new distribution that is their “intersection”. If one were to modify the additive model to activate multiple primitives by adding the samples from the individual distributions, then this will no longer result in the additive composite distribution described in Equation 1. In fact, this will produce a composite distribution that is more akin to the MCP model. We will add this discussion to the paper to better clarify the distinction.

Reviewer 1:

Re: option-critic baseline

The options in the option-critic baseline are also pre-trained in the same manner as MCP, using the motion imitation tasks, and the learned options are then transferred to new tasks. We have also experimented with training an option-critic model from scratch on the transfer tasks, but found that pre-training yielded better performance.

Re: additional references and ensemble/mixture policies

Thank you for the pointers, we will include these additional references. In principle, many of the algorithms proposed in prior work for learning options and mixture policies, including DDCO [Krishnan et al., 2017], could also be used to train multiplicative primitives, as in our work – in that sense, our contribution is complementary to these prior methods and largely orthogonal. However, we also observe that the relatively simple training procedure in our work is effective at learning useful multiplicative primitives on a range of difficult tasks.

Re: hyperparameters

Tables with the hyperparameters used for MCP and other baselines are available in the supplementary material. We will also include more detailed explanations for the different parameters in the final draft.

Reviewer 2:

Re: activating multiple actions per timestep

We will improve the writing to better clarify the use of “activation”. The compose policy still proposes a single action per timestep, but the distribution from which the actions are sampled from is the result of composing multiple primitive distributions, which may each specialize in different skills. For example, as shown in the supplementary video and Figure 7, we observe that some primitives specialize in pushing back with a particular leg, while other primitives specialize in lifting the legs. By composing these primitives, MCP can then produce a range of different behaviors.

Re: ablation experiments

Thank you for the suggestions. We will include additional ablation experiments for fixing the primitive weights during transfer, and hiding the goal input from the primitives. One of the primary motivation for not providing the primitives with the goal is to prevent degeneracy. During pre-training, if the primitives also have access to the goal, then it is possible for a single primitive to solve all tasks, while the other primitives become inactive. This degeneracy can lead to poor transfer performance, as the other primitives may not acquire useful skills.