Algorithm 1 Q-switch Mixture of Policies (QMP)

Input: Total number of tasks T, Initialize policies $\{\pi_i\}_{i=1}^T$, and Q-functions $\{Q_i\}_{i=1}^T$, Data buffers $\{\mathcal{D}_i\}_{i=1}^T$
for each epoch **do**
    for $i = 1$ to T **do**
        for each environment step **do**
            Observe current state $\mathbf{s}$
            for $j = 1$ to T **do**
                $\mathbf{a_j} \sim \pi_j(\mathbf{a_j}|\mathbf{s})$
            $\mathbf{a}^* = \arg\max_{\mathbf{a_j}} Q_i(\mathbf{s}, \mathbf{a_j})$
            $\mathcal{D}_i \leftarrow \mathcal{D}_i \cup \{(\mathbf{s}, \mathbf{a}^*, r(\mathbf{s}, \mathbf{a}^*), \mathbf{s}')\}$
    for $i = 1$ to T **do**
        Update π_i, Q_i using $\mathcal{D}_i$ with SAC for k gradient steps
Output: Trained policies $\{\pi_i\}_{i=1}^T$

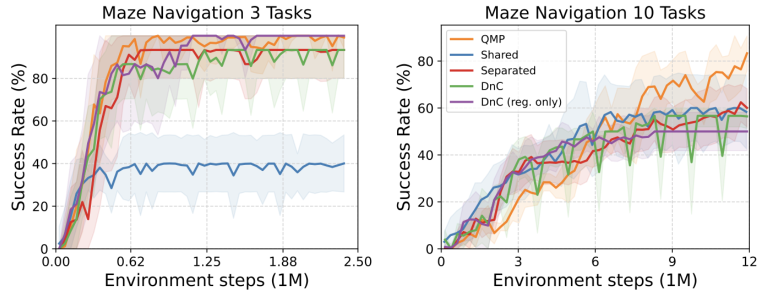

Figure P1: QMP scaling with number of tasks: Maze environments with 3 tasks v/s 10 tasks. With more tasks, the benefit of QMP increases because of the greater proportion of shared behaviors.

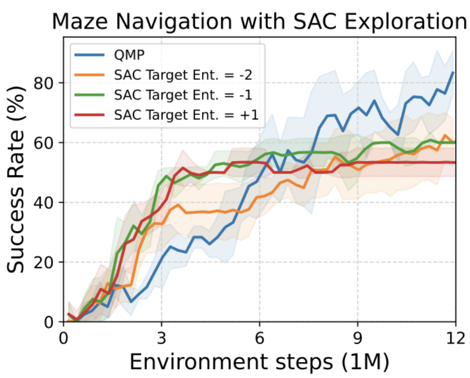

Figure P2: Single-task exploration by varying SAC target entropy. QMP reaches a higher success rate because it shares exploratory behavior **across** tasks.

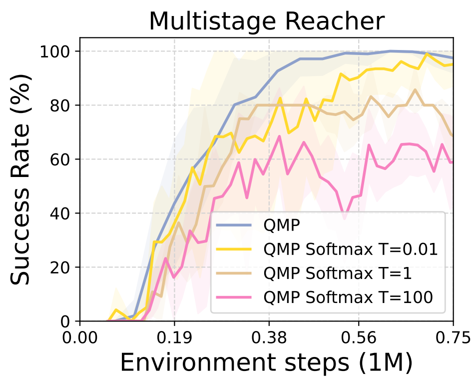

Figure P3: Using probabilistic mixtures with QMP by using a softmax over Q values with temperature T. Softmax results over 3 seeds.

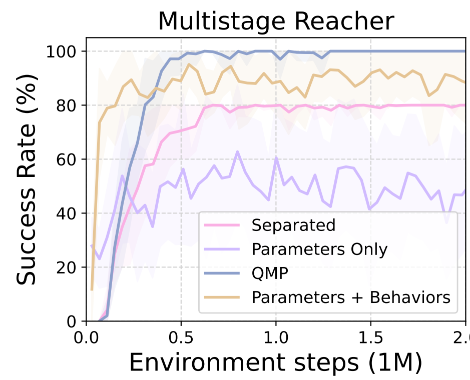

Figure P4: Parameters Only baseline suffers due to negative interference in Task 4. Parameters + Behaviors is still able to reach high success rates.

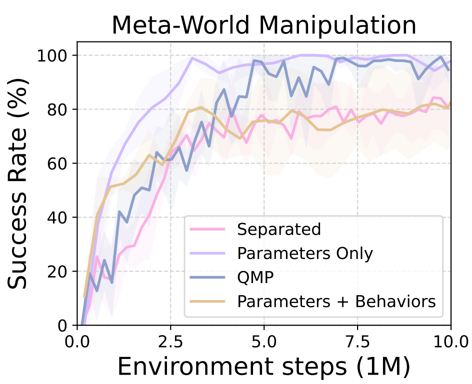

Figure P5: Parameters Only outperforms the other methods. Parameter sharing in this environment seems to de-stabilize QMP training.

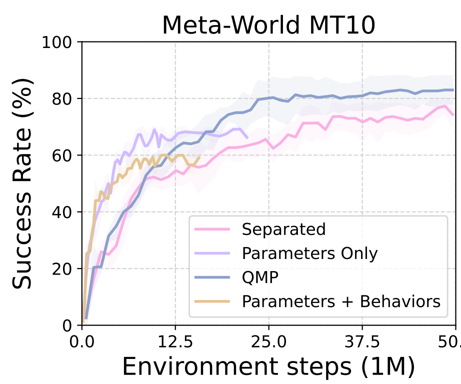

Figure P6: QMP out-performs Separated (no behavior sharing). The Parameter Sharing baselines were difficult to tune and suffer from training instability.

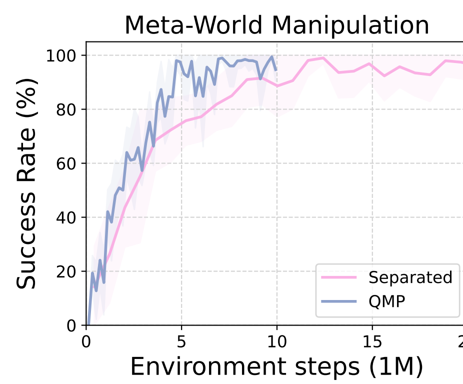

Figure P7: Separated does converge to 100% success rate with a longer run but takes over 2x the number of samples as QMP to converge.