1 ME-TRPO

一个agent，在黑暗中，对周围的state space一无所知，只能通过try-and-error去获取生存策略（policy），了解环境（environment）。它有两种方式，第一种方式不对环境进行研究，只研究policy和state的关系。它通过每次的尝试，来总结在某一个state，如果采取相应的动作（a），将来会得到的总收益($q(s,a)$)当然这个估计一开始是不准确的，因此agent需要每次动态刷新这个收益值。每次刷新之后，agent就会根据当前的值函数去选取最优策略，即选取使值函数最大的action。这是model-free的方法。
还有一种方式需要对环境进行研究，即agent先对发现它在当前的s_0，如果采取某一动作a_1，就会转移到s_1，而若采取动作a_2，就会转移到另一个状态s_2，它发现这其中有个对应关系。于是它转而先研究这个对应关系，然后根据研究的结果来选择策略。如果它的起始状态是s_0，而它想到s_g，则agent会在simulator中构造出一种从s_0到s_g的路径。agent不需要亲自到s_g，而是根据采样的轨迹就可以推知s_t，a_t，s_{t+1}的关系。这就是model-based的方法。这种方法的缺点在于模型和实际世界有很大的偏差。在这篇文章中，作者认为model bias的原因在于agent在policy optimization的过程中exploit了数据不够充足以训练的region。因此，作者提出使用深度网络元组来训练model，以保持model的uncertainty。目前搞不明白的是为什么这样可以让agent避免上述那点原因。

model-based的特性目前还不是很清楚。不过可以思考下面几个问题：
1.为什么vanilla model-based RL会在policy optimization阶段exploit数据并不充足的区域？
2.model learning和policy optimization两者之间的关系是什么？
先来看问题2。agent先使用初始的policy去sample real trajectory，然后使用sample去fit transition function，接着利用fit的model产生synthetic trajectory，然后使用这些模拟的路径去optimize policy。整个过程循环，就是vanilla model-based RL。
分析这个过程，policy optimization和model learning的联系有两个：sample和synthetic trajectory。前者是实际的trajectory，由当前policy执行产生，用来train dynamics model，后者是synthetic trajectory，由dynamics model产生，用来train policy。这是一个包含四个环节的循环。这是Dyna的模型之源头。
分析一下误差在哪里产生。误差主要来源于prediction error。model无论如何精确都不可避免地会和实际世界有误差，而且在产生trajectory的过程中，每一步都会产生误差，并逐渐累积。这部分误差可以通过model-free的方式来减轻。这就是model-free和model-based结合的方法。
policy error:$d(\pi_{real},\pi_{syn})$
Prediction error:$\frac{1}{D}\sum_{(s_t,a_t,s_{t+1}\in D)} \parallel s_{t+1}-f_\phi(s_t,a_t) \parallel^2$
令$\hat{s}_{t+1}=f_\phi(s_t,a_t)$
model learning的缺点在于对$s \in S_{sample}$,prediction 比较准确，而对于$s \notin S_{sample}$,则不一定能保证正确。这也就是所谓的overfitting，或者是在文章里作者所说的在policy optimization阶段agent在这些状态时，会导致误差。