TempLe: Learning Template of Transitions for Sample Efficient Multi-task RL

Yanchao Sun*, Xiangyu Yin† and Furong Huang*

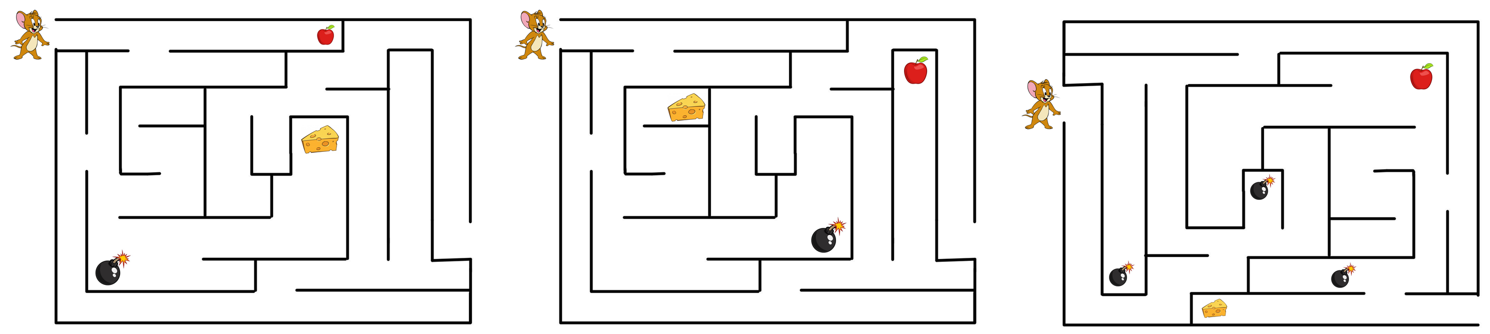
ycs@umd.edu, yinxiangyu@bupt.edu.cn ,furongh@umd.edu

*University of Maryland, College Park, †Beijing University of Posts and Telecommunications, China

Department of Computer Science

Multi-task Reinforcement Learning

Multi-task Reinforcement Learning (MTRL) studies the problem of efficiently learning a series of tasks by **knowledge transfer**.



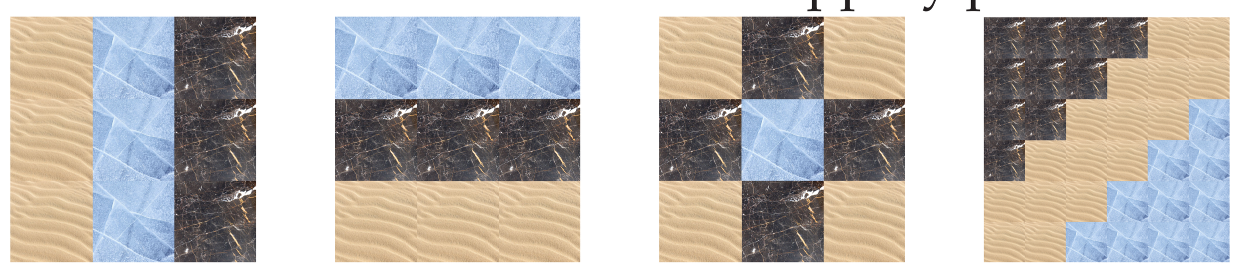
Challenges in MTRL:

- Guarantee the sample efficiency?
- Trade-off between correctness and efficiency?
- Tasks with different state/action spaces?

Our Contributions: our proposed algorithms achieve **SOTA sample complexity** and work for tasks with **varying state/action space**.

A Motivating Example

Various "**landforms**" → various slippery probabilities.



Sand: never slip; *Marble*: slip with prob 0.2; *Ice*: slip with prob 0.4.

G types of landforms → G^N different kinds of mazes (N: # of grids)

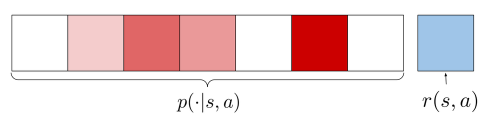
Our Goal: to extract and utilize modular similarities among and within tasks.

Transition Template

The Traditional Representation of Dynamics

The dynamics of a state-action pair can be represented as:

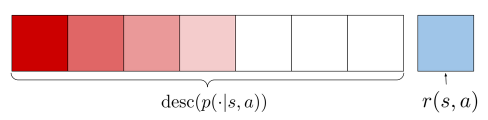
$$\theta(s,a) = [p(s_1|s,a), p(s_2|s,a), \cdots, p(s_S|s,a), r(s,a)]$$

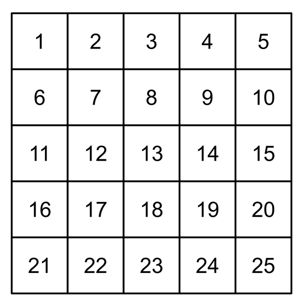


Transition Template (TT): A New Representation of Dynamics

By permuting the transition probability vector, we get:

$$\mathbf{g}_{(s,a)} = [\mathrm{desc}(p(\cdot|s,a)), r(s,a)]$$



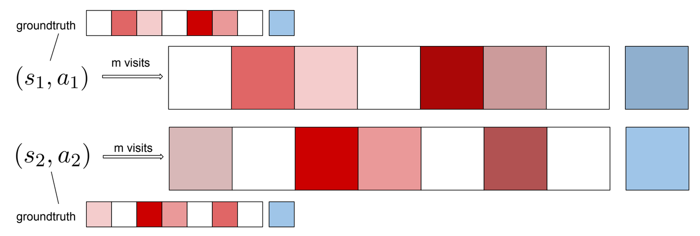


For all 100 s-a pairs in the 5×5 grid world, there are only 2 distinct TTs. Transition Templates can capture modular similarities

Learning with Transition Templates

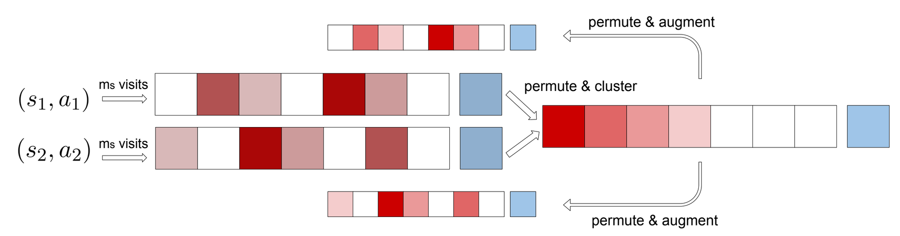
The Old Way: Direct Estimation

- Obtain $\hat{\theta}(s,a) = [\frac{\mathbf{n}(s,a,\cdot); R(s,a)}{n(s,a)}]$ once $n(s,a)$ reaches a known threshold m;



Our Way: Augmented Estimation with Transition Templates

1. **Rough estimation**: obtain $\widehat{\theta}(s,a) = [\frac{\mathbf{n}(s,a,\cdot); R(s,a)}{n(s,a)}]$ once $n(s,a)$ reaches a small known threshold m_s;
2. **Permutation**: permute $\widehat{\theta}(s,a)$ to its corresponding TT $\tilde{\mathbf{g}}_{(s,a)}$;
3. **Template identification**: identify the permuted estimate $\tilde{\mathbf{g}}_{(s,a)}$ as one of the existing groups of TTs $\widehat{\mathbf{g}}$.
4. **Augmentation**: obtain a more confident estimate $\hat{\theta}(s,a)$ by permuting back the accumulated knowledge of its corresponding $\widehat{\mathbf{g}}$.

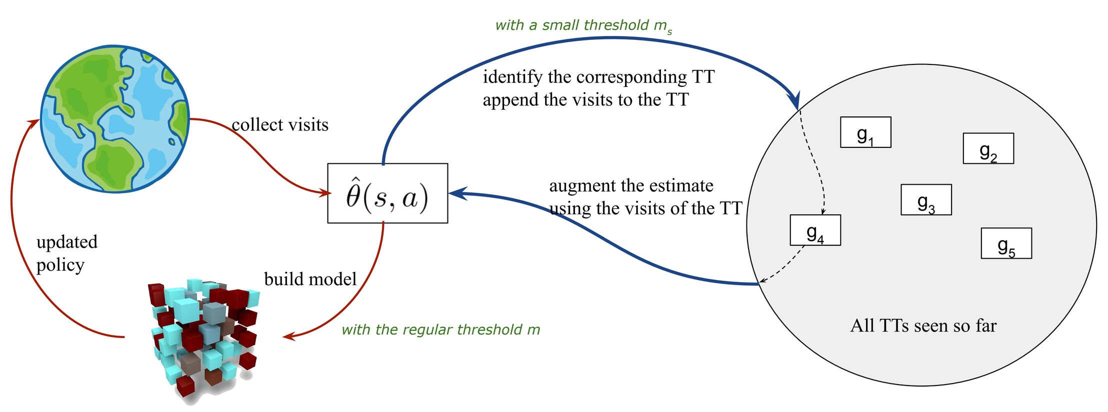


Algorithms for MTRL

O-TempLe: Online Template Learning

• Streaming-in tasks $\mathcal{M}$ may have different state space $\mathcal{S}$, action space $\mathcal{A}$, dynamics $p(\cdot|\cdot,\cdot)$ and rewards $r(\cdot,\cdot)$.

• # of MDPs in $\mathcal{M}$ can be arbitrarily large.



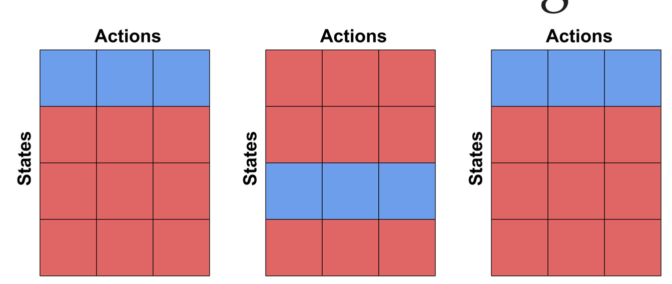
1. **Cluster** estimations to groups of TTs;
2. **Augment** the estimation of dynamics

FM-TempLe: Finite-Model Template Learning

• If # of MDPs is small, we can further accelerate the learning.

Phase 1: collect and cluster MDP models according to their TTs

Phase 2: identify the model of any new task by its TTs.



Theoretical Results

Sample Complexity of O-TempLe

Suppose there are G underlying TTs in total. For any $\epsilon > 0, 1 > \delta > 0$, running O-TempLe on T tasks, each for at least $\mathcal{O}(\frac{DSA}{\omega^2}\ln\frac{1}{\delta})$ steps, generates at most $\tilde{O}\left(\frac{SGV_{\max}^3}{\epsilon^3(1-\gamma)^3} + \frac{TSAV_{\max}}{\omega^2\epsilon(1-\gamma)}\right)$ non-ϵ-optimal steps, with probability at least $1-\delta$.

D: the diameter of the MDP;

ω: the error tolerance for TT identification, see the paper for more details.

Remarks. (1) The sample complexity of RMax (for T tasks) is $\tilde{O}\left(\frac{TS^2AV_{\max}^3}{\epsilon^3(1-\gamma)^3}\right)$. (2) O-TempLe achieves linear dependence on S and A, the cardinality of state space and action space.

Sample Complexity of FM-TempLe

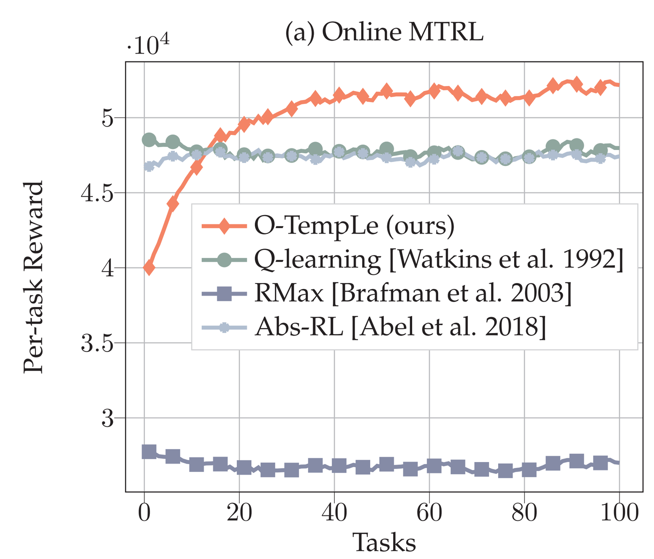
FM-TempLe on T tasks follows ϵ-optimal policies for all but $\tilde{O}\left(\frac{SGV_{\max}^3}{\epsilon^3(1-\gamma)^3} + \frac{T_1SAV_{\max}}{\omega^2\epsilon(1-\gamma)} + \frac{(T-T_1)DC^2V_{\max}}{\omega^2\epsilon(1-\gamma)}\right)$ steps with probability at least $1-\delta$.

$T_1 = \Omega(\frac{1}{p_{\min}}\ln\frac{C}{\delta})$ is the number of tasks in the first phase

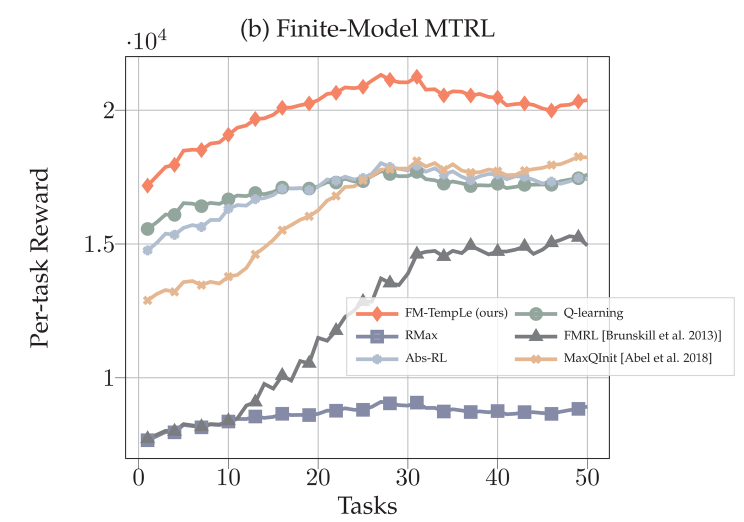
$p_{\min}$ is the minimal probability for a task to be drawn from $\mathcal{M}$.

Remarks. (1) If $DC^2 < SA$ and $T \gg T_1$, FM-TempLe requires less samples than O-TempLe. (2) When T is large and T_1 is small, FM-TempLe can get rid of the dependence on S and A.

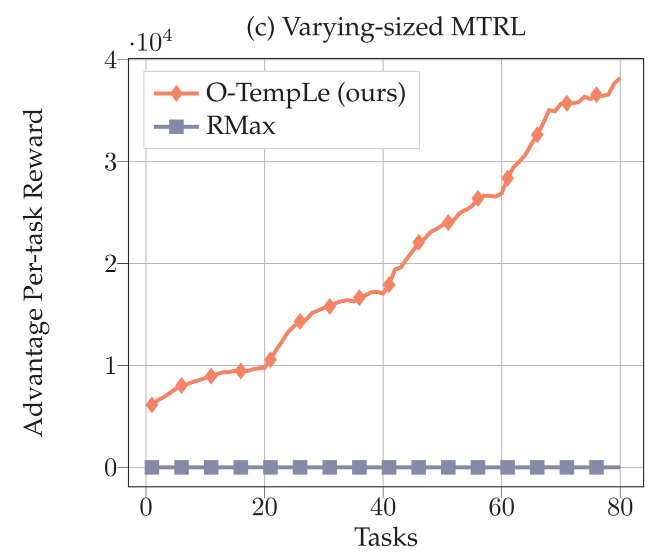
Experiment Results



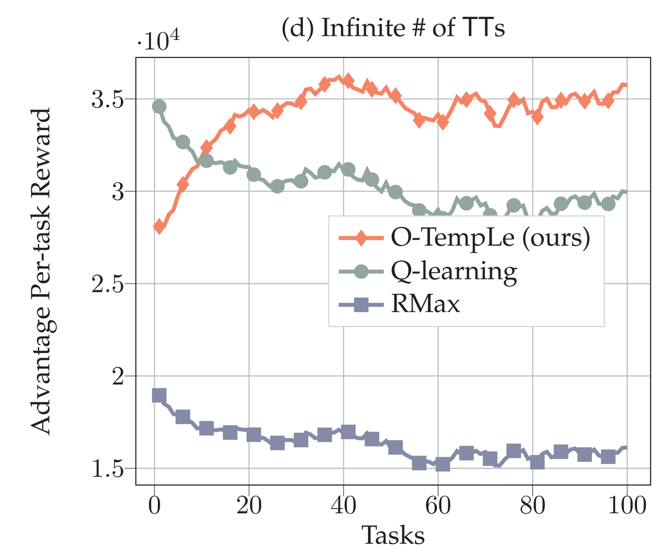
(a) 100 maze tasks with random combinations of 3 "landforms"



(b) 50 tasks sampled from 2 underlying maze models.



(c) mazes with different sizes.



(d) landform ~ mixture of Gaussian