Shota Takeshima: In this document, I write the report of what I have done this week, and mention the goal next week.

Report: week 2 1/22 ~ 1/29

In our storyboad, the current theme is corresponding to **Theme 1.3**.

> We will start implementing algorithms into city-learn or different simulators and evaluate the performance for each algorithms.

During our winter break, we noticed that when we use reinforcement learning(RI) on City Learn, it needs relatively much data (about over 8-year simulation) to get an optimal energy management policy. So, in the actual adoption of this RI, it is not practical to wait several years to get the policy. Hence, now we are trying to take advantage of meta-learning. Using multiple data from other tasks(different regions, buildings, and climates), we can, in advance, averagely optimize RI policy (meta-learning) and, later, adjust the policy for the target environment for a short time.

Hence, the current tasks I have done are:

1. to learn meta-learning itself using UCBerkley's lecture videoes (~1/25)
   - In the lecture, I have learned main three method to implement meta-reinforcement learning.
     - RNN as meta-learning: RI model (agent) produces appropriate action a_t from the state,s_t, of observation,o_t, at current time t. In RNN case, we can input previous steps $(s_{t-k}, a_{t-k}, s_{t-k+1}, r_{t-k+1}), k > 1$ in addition to s_t. Using previous steps as history, RNN agent recalls which task used in training process is similar to the current unkown task. Then it uses its knowledge to calculate an appropriate action.
     - MAML: In the loss function of policy maker, it optimizes average loss for each training tasks (not only specific one task). So, after training, if there is the similarity between unknow new task and training tasks, it takes less time to find optimal parameter for the known task.
     - Infering new task as one of training task: Using the history in additon to current state, this method infer which training task, z, is similar to current situation. Using z and s_t, the agent calculate an action a_t.
   - First, we decided to use MAML because its concept is easy to understand for people who learned ML algorithms once.
2. to get familiar with the meta-learing implementation with garage that includes multiple RI methods. (~1/29)
   - I tried to replicate meta-RL example in garage document understanding waht each line in the code means. Now, I'm trying to use MAML algorithm in the example code.
3. to plan how to implement meta-reinforcement learning using garage and City Learn Simulator. (~1/29)
   - CityLean simulator provides two types of RI agent. One is single central simulator and another is multiple agent for each building. Garage package can handle only single agent, so first we try to use single agent in Garage meta-learning framework.

- As the definition training tasks, we forcus on building types. After training Rl agent with multiple buildings' data, then we'll let the agent fit(test) to new unknown building type.

Goals: week 3 2/1 ~ 2/5

In our storyboad, the current theme is still corresponding to **Theme 1.3**.

> We will start implementing algorithms into city-learn or different simulators and evaluate the performance for each algorithms.

We have learned the concept of meta-learning through the lectures above. Also, we are trying to get experiences of implementing with Garage. This week, I'll implement the simple example using CityLearn and garage.

1. to incorporate CityLean Simulator to Garage meta-learning framework
    - As a prototype, we focus on one single agent that manages only one type of building. Then, we observe the meta-learning works correctly.
2. to read the original paper of MAML
    - This is my individual agenda. To understand MAML algorithm deeper.