Env

type 1
Π policy θ_1^+

possible type:
type 4 type 5 type 6

type 2
Π policy θ_2

type 3
Π policy θ_3

initialize $\theta_1, \theta_2, \theta_3$

done indicate how many round we want to update.

while not done:

for each building i in Env:

for each task:

↳ create new env by switching each building for each loop.

Ex. type 4
Π policy θ_1

type 2
Π policy θ_2

type 3
Π policy θ_3

↳ sample for 1 year and add data into replay buffer.

↳ update $\theta_{i,\text{task}i}$ using below equation

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{T_i}(f_\theta)$$

↳ with updated policy sample for 1 more year and update replay buffer

end loop

MAML

Cityb( )

Update $\theta_i \leftarrow \theta - \beta \nabla_\theta \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$ using each $\mathcal{D}_i'$ and $\mathcal{L}_{\mathcal{T}_i}$ in Equation 4

$\theta_1^+ \leftarrow \theta_1^-$

The updated θ_i will be assign to the building from original environment.

Ex.

If θ_1^* has been updated then it will be assign to building 1.

Also, for updating θ_2 using MAML algorithm, building will use θ_1^* to make action.

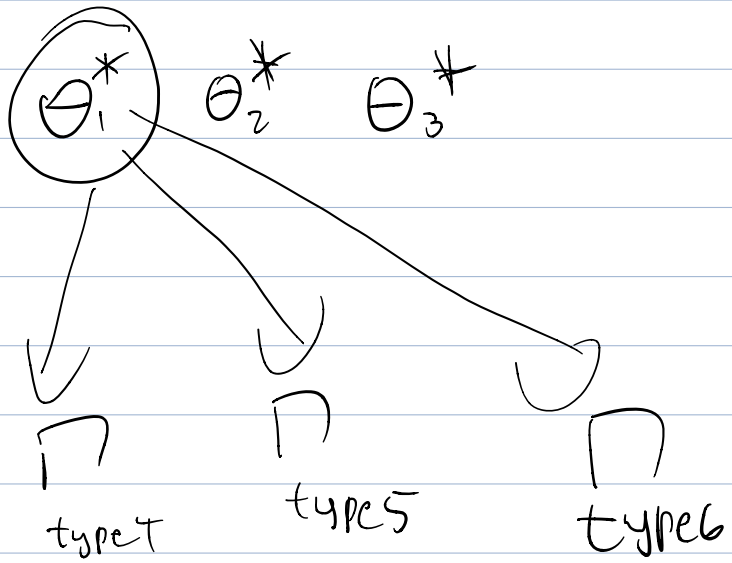
Thus, our update of policy is similar to gibbs sampling.

Question.

- add buffer.
  - do we always start new or keep them.
  - (within MAML and outside of MAML)

add on zeros

add zero to take account



reinforce PG A2C

Valina policy gradient.