

Syllabus: Curriculum Learning Made Easy

Ryan Sullivan
University of Maryland,
College Park
rsulli@umd.edu

github.com/RyanNavillus/Syllabus



Background

Curriculum learning has contributed to most of the big successes in reinforcement learning.

- AlphaGo used self-play, an implicit curriculum of increasingly skilled opponents.
- AlphaStar used league training, a form of population-based self-play.
- GT Sophy used manually designed scenarios and task distributions.

Curriculum learning provides orthogonal improvements to RL algorithms by explicitly modifying the task distribution, which is typically only implicitly controlled via exploration.

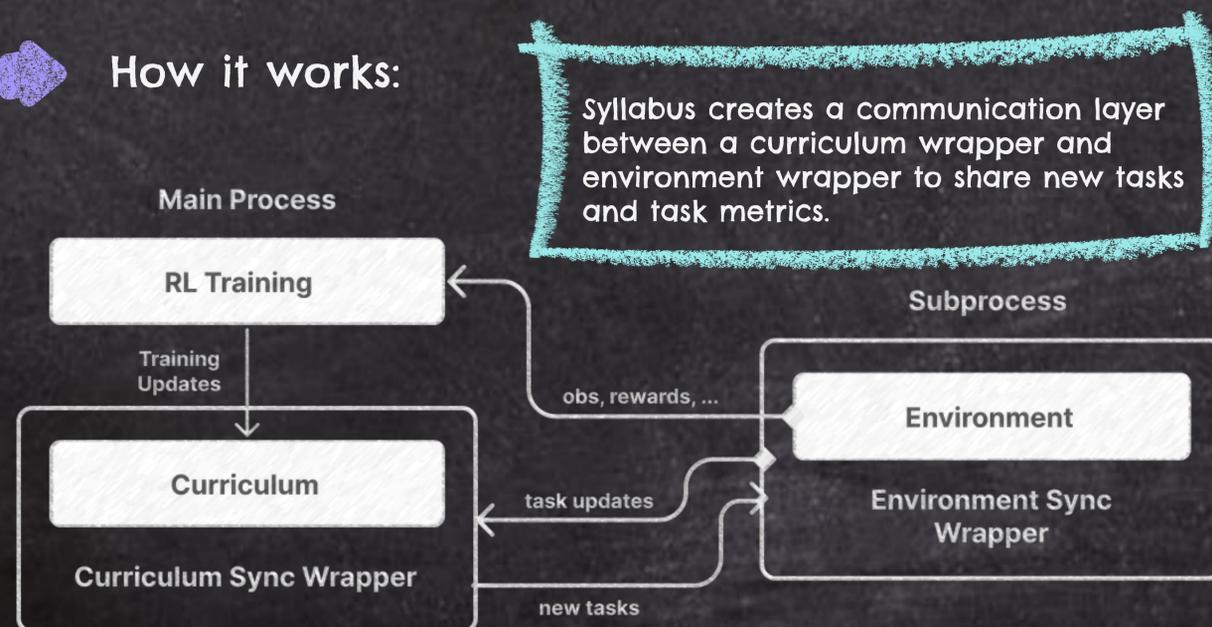
Despite this, curriculum learning is not supported in any major RL libraries.

- Curriculum learning methods require a task generator, environment interface for setting tasks, and systems for updating the task distribution with training data.

What is Syllabus?

- A unified interface for curriculum learning algorithms.
- A library of single-file curriculum learning algorithm implementations.
- A global synchronization system that handles communication between multiprocessed environments and curricula, compatible with most RL libraries.
- Allows you to easily add curriculum learning methods to existing RL code.

How it works:



Curriculum Learning in 4 Lines of Code

All you need to do is create a **curriculum**, define a **task space**, and apply the **synchronization wrappers**.

```
def env_creator(config):
    env = gym.make("CartPole-v1")
    env = CartPoleTaskWrapper(env)
    return env

register_env("task_cartpole", env_creator)

config = {"env": "task_cartpole"}

tuner = tune.Tuner("APEX", param_space=config)
results = tuner.fit()
```



```
task_space = TaskSpace(Box(-0.3, 0.3, shape=(2,)))

def env_creator(config):
    env = gym.make("CartPole-v1")
    env = CartPoleTaskWrapper(env)
    env = RaySyncWrapper(env, task_space=task_space)
    return env

curriculum = SimpleBoxCurriculum(task_space)
curriculum = make_ray_curriculum(curriculum)

register_env("task_cartpole", env_creator)

config = {"env": "task_cartpole"}

tuner = tune.Tuner("APEX", param_space=config)
results = tuner.fit()
```

Implemented Curriculum Methods

Prioritized Level Replay (Jiang et al. 2021) is a popular curriculum learning method.

- It saves a buffer of levels and samples them according to the agent's LI value loss.
- Syllabus provides a benchmarked implementation of PLR and Domain Randomization.

We also have implementations of many other methods which will soon be benchmarked.

- Sequential curricula
- Learning progress (Kanitsheider et al 2021)
- Simulated annealing

Progen Bigfish Test Returns

Syllabus Domain Randomization Syllabus Prioritized Level Replay
Baseline Domain Randomization Baseline Prioritized Level Replay

