```
# 3. Sử dụng policy đã học

def learned_policy(s):

phi = feature_extractor(s)

a_idx = model.predict([phi])[0]

return index_to_action(a_idx)
```

• Đánh giá: tính accuracy trên tập dữ liệu expert; deploy policy:

```
# Môi trường grid world
state = env.reset()
done = False
while not done:
    action = learned_policy(state)
    state, reward, done, info = env.step(action)
    env.render()
```

Nếu môi trường đơn giản và expert data đủ đa dạng, policy này sẽ hoạt động tốt.
 Trong trường hợp phức tạp, có thể cần fine-tune thêm bằng RL để tăng hiệu năng.

4. Sự kết hợp SL và RL

- Ví dụ trên là Imitation Learning (behavior cloning). Khi có expert data, SL giúp khởi tạo policy. Sau đó có thể tiếp tục RL (reinforcement fine-tuning) để cải thiện dựa reward thực.
- Trong trường hợp không có expert data, SL vẫn dùng để học value approximation thông qua regression với target là return/T D-target.

Tổng kết (Summary)



