# Replay Buffer Unpacking Fix

## Problem Found ✅

Your `replay_buffer.sample()` returns a **dictionary**, but some code tries to unpack it as a **tuple of 5 values**:

# ❌ WRONG - causes "too many values to unpack"

```
batch = self.replaybuffer.sample(32)
states, actions, rewards, next_states, dones = batch
```

**Error happens at Episode 14** because:

1. First 5000 samples = no training (warmup)
2. At Ep 14 (~1023 samples), warmup ends
3. First `train_step()` calls `.sample(32)`
4. Tries to unpack dict as tuple → **ValueError**

---

## Good News ✅

Your **trainer.py** already uses correct dictionary access in `train_step()`:

```
batch = self.replaybuffer.sample(self.batch_size)
states = batch["states"].to(self.device)
actions = batch["actions"].to(self.device)
rewards = batch["rewards"].to(self.device)
next_states = batch["next_states"].to(self.device)
dones = batch["dones"].float().to(self.device)
legal_masks_next = batch["legal_masks_next"].to(self.device)
legal_masks_current = batch["legal_masks_current"].to(self.device)
```

✅ **This is correct!**

---

## But Check These Files

There might be other files that call `.sample()` incorrectly:

```
grep -r ".sample(" training/d3qn/*.py | grep -v "trainer.py"
grep -r "replaybuffer.sample" training/ --include="*.py"
```

Common culprits:

- `training/d3qn/selfplay_utils.py`
- `training/d3qn/evaluation.py`

- training/d3qn/train_agent_3_pool_offline.py

---

## Test Command (Updated)

Fix the unpacking issue first:

python -c "
from training.d3qn.train_d3qn import build_d3qn_trainer
from checkers_env.env import CheckersEnv
import torch

env = CheckersEnv()
trainer = build_d3qn_trainer(env, device='cuda', q_clip=100.0)

# Fill to warmup

for i in range(15):
trainer.run_episode(training=True, max_moves=200)
print(f'Ep {i+1}: Buffer size={len(trainer.replaybuffer)}')

```
if len(trainer.replaybuffer) >= 1000 and i == 12:
    # Test batch unpacking
    batch = trainer.replaybuffer.sample(32)
    print(f'  Batch keys: {list(batch.keys())}')
    print(f'  States shape: {batch[\"states\"].shape}')
    print(f'  Rewards: min={batch[\"rewards\"].min():.2f}, max={batch[\"rewards\"
    break
```

print('✅ Warmup complete, no ValueError!')
"

---

## Action Items

1. **Run the test command above** ← Try this first
2. **If it passes:** Training is ready. Move to quality-gate fix (previous document)
3. **If it fails:** Find the file doing wrong unpacking and fix it using the dict access pattern

---

## Key Insight

**The issue is NOT in trainer.py** - it already does dictionary access correctly.

**The issue IS:** Either:

1. Another file imports and calls .sample() wrongly
2. An older version of trainer.py cached somewhere
3. Or the error happens in a different file entirely

**Run the test** to confirm where the real error is. Then we fix THAT file.