**🧠 What I Meant by *"training for too few steps"***

Even with a stable DQN setup (like target\_net syncing), **your agent might not learn anything meaningful** if:

* You're running the training loop for **only a few hundred steps** (like 500 time steps),
* But **exploration is still high** (ε is large for much of that time),
* And the **reward signal is sparse or small** (e.g., only reward when position is closed).

So, even though the model is technically being trained, it doesn’t get **enough experience** to actually differentiate good from bad actions.

**📉 Visual Symptom**

You'll see:

* Flat or random wallet chart,
* Small or erratic reward values,
* Q-values that don’t evolve.

**🔧 What to Do**

| **Action** | **Why** |
| --- | --- |
| Increase training data length (e.g., use 5,000–10,000 time steps) | More opportunities to learn meaningful patterns |
| Reduce ε faster or use linear decay | So it starts exploiting sooner |
| Add **intermediate rewards** (e.g., penalize holding losing trades) | Makes learning less sparse |
| Use a replay buffer size of at least 5,000 and warm it up before training | Helps stabilize early updates |

**✅ Key Point**

DQN needs **many training iterations** to converge — just like a human trader wouldn’t learn with only 30 trades, your model needs **thousands** of interactions to form value-based reasoning.