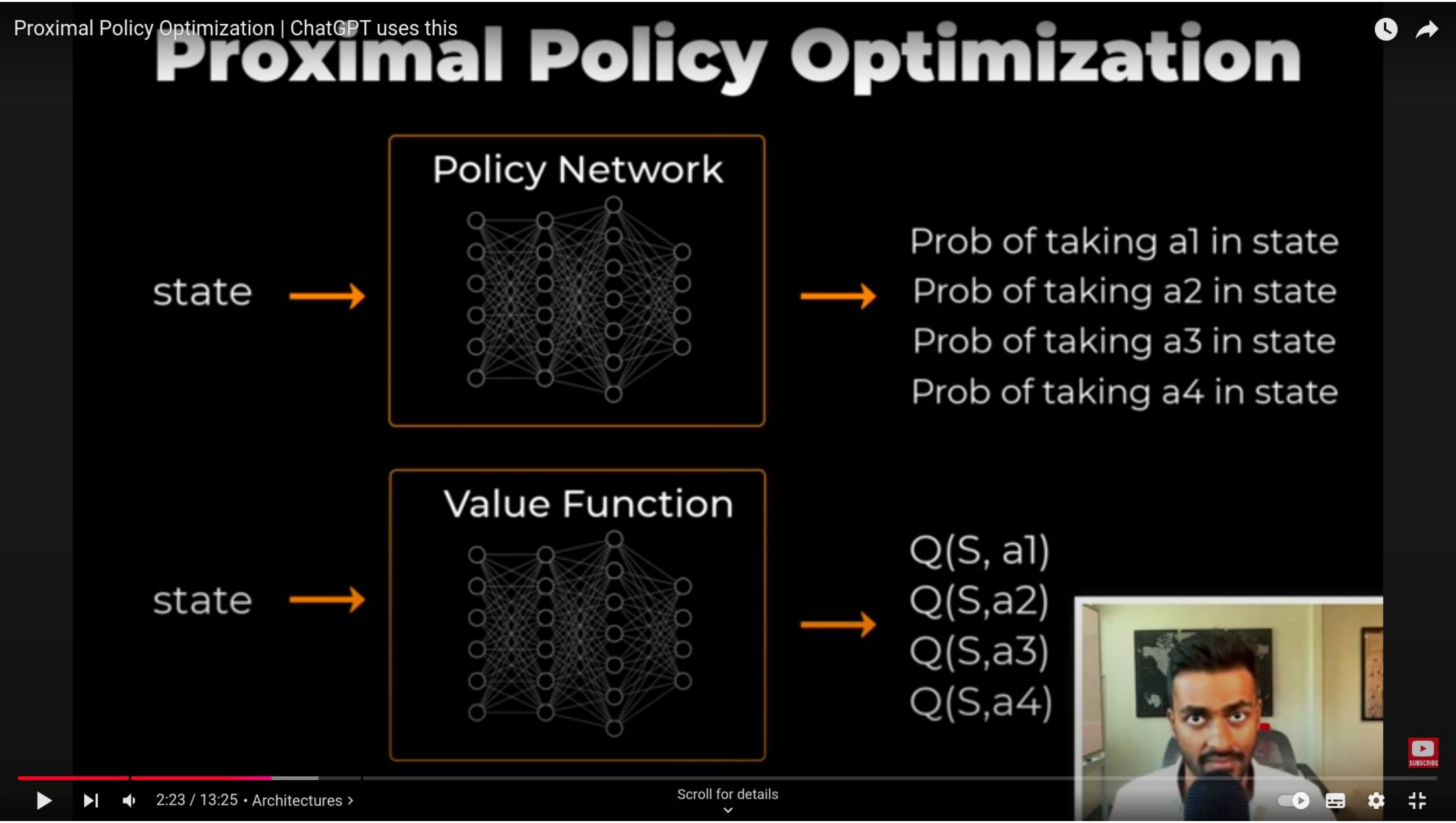
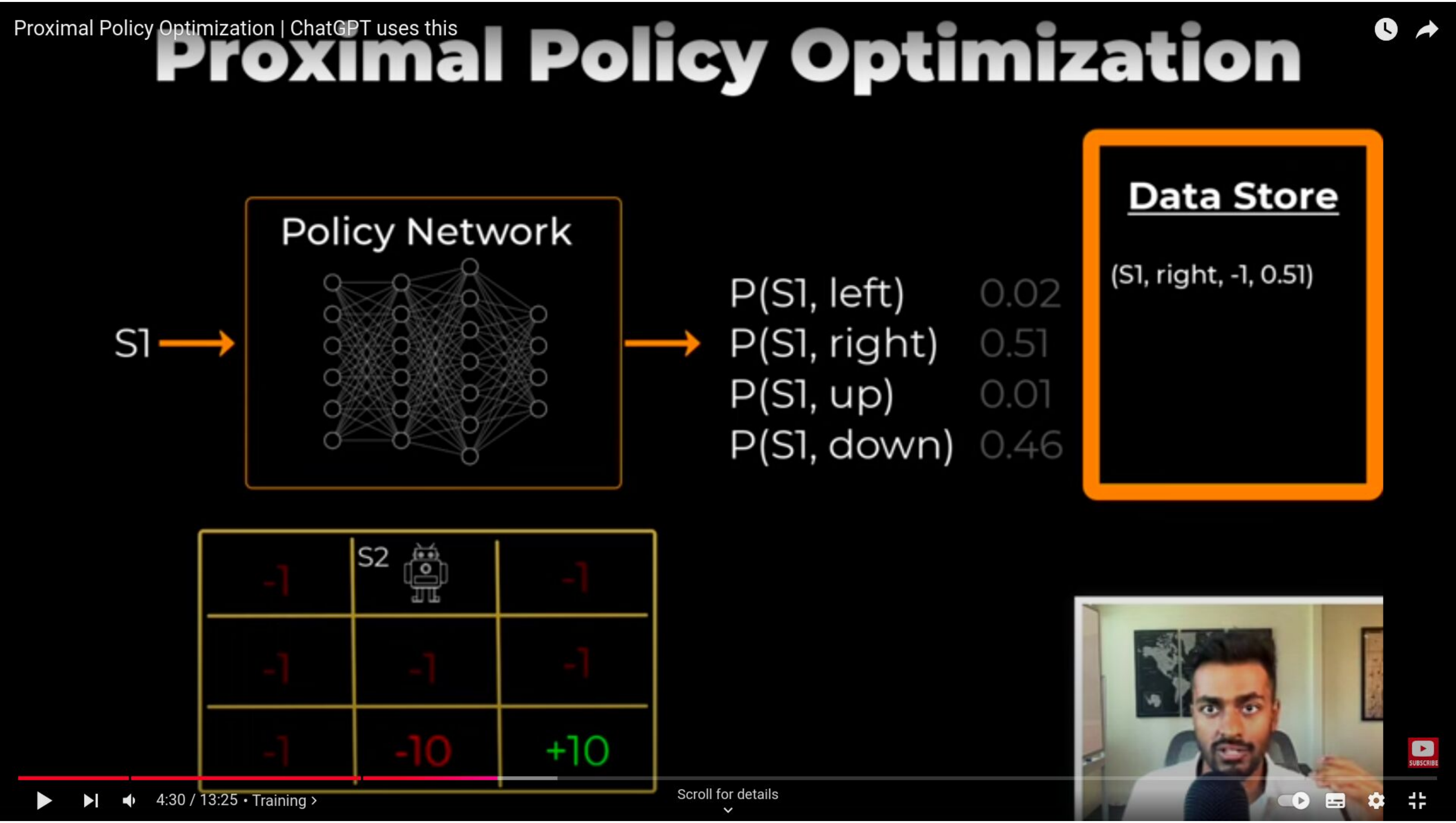
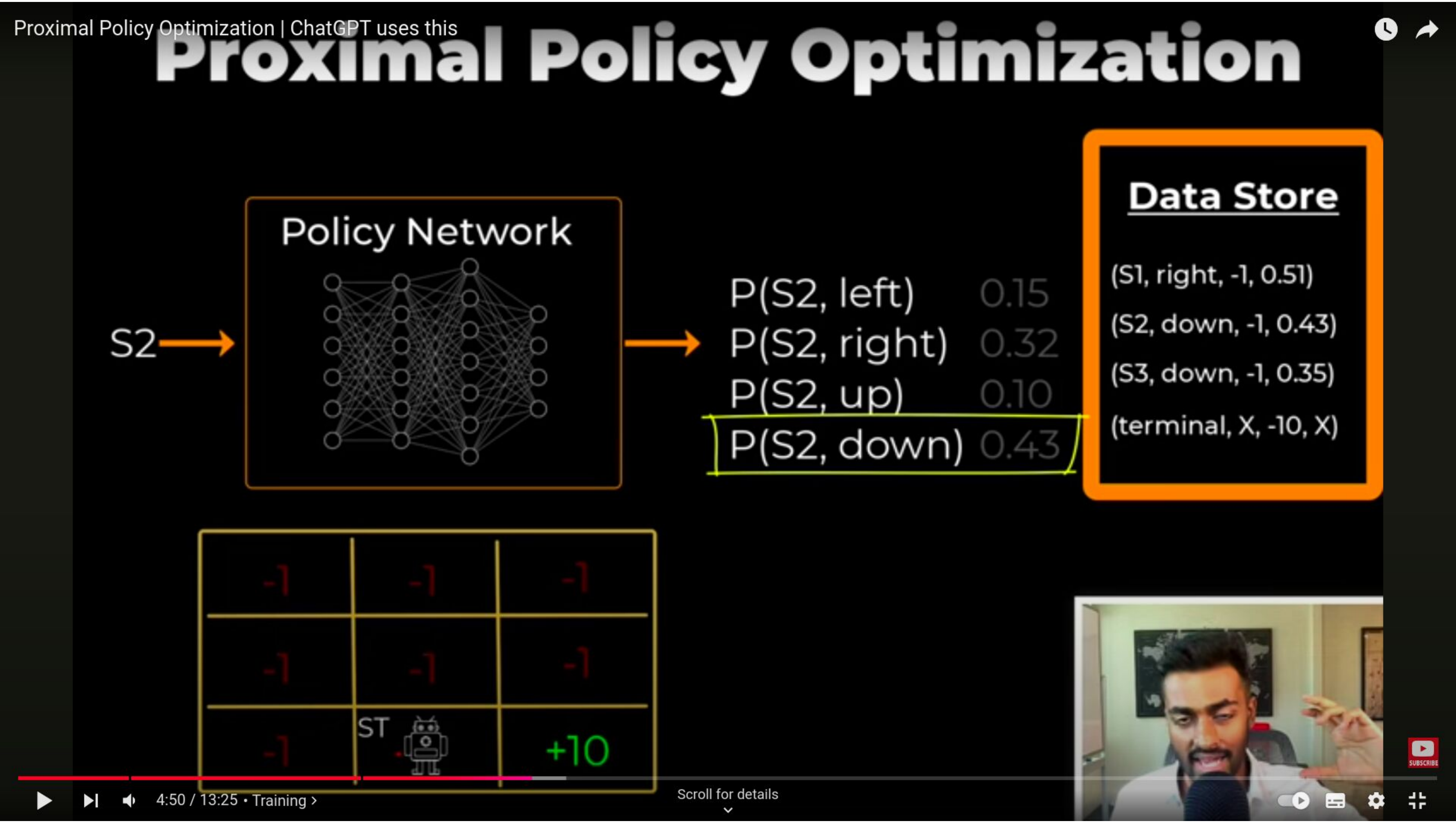
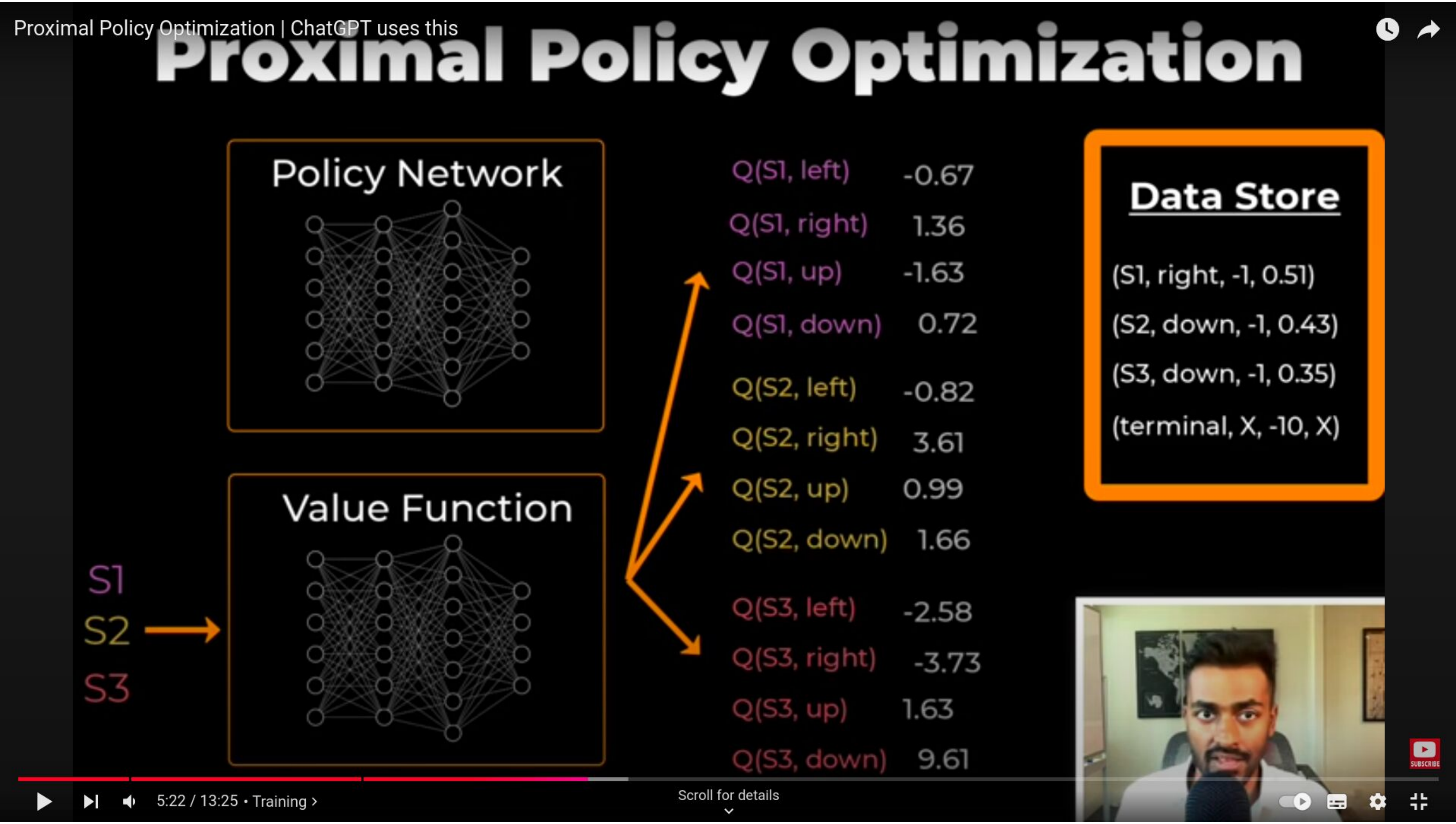
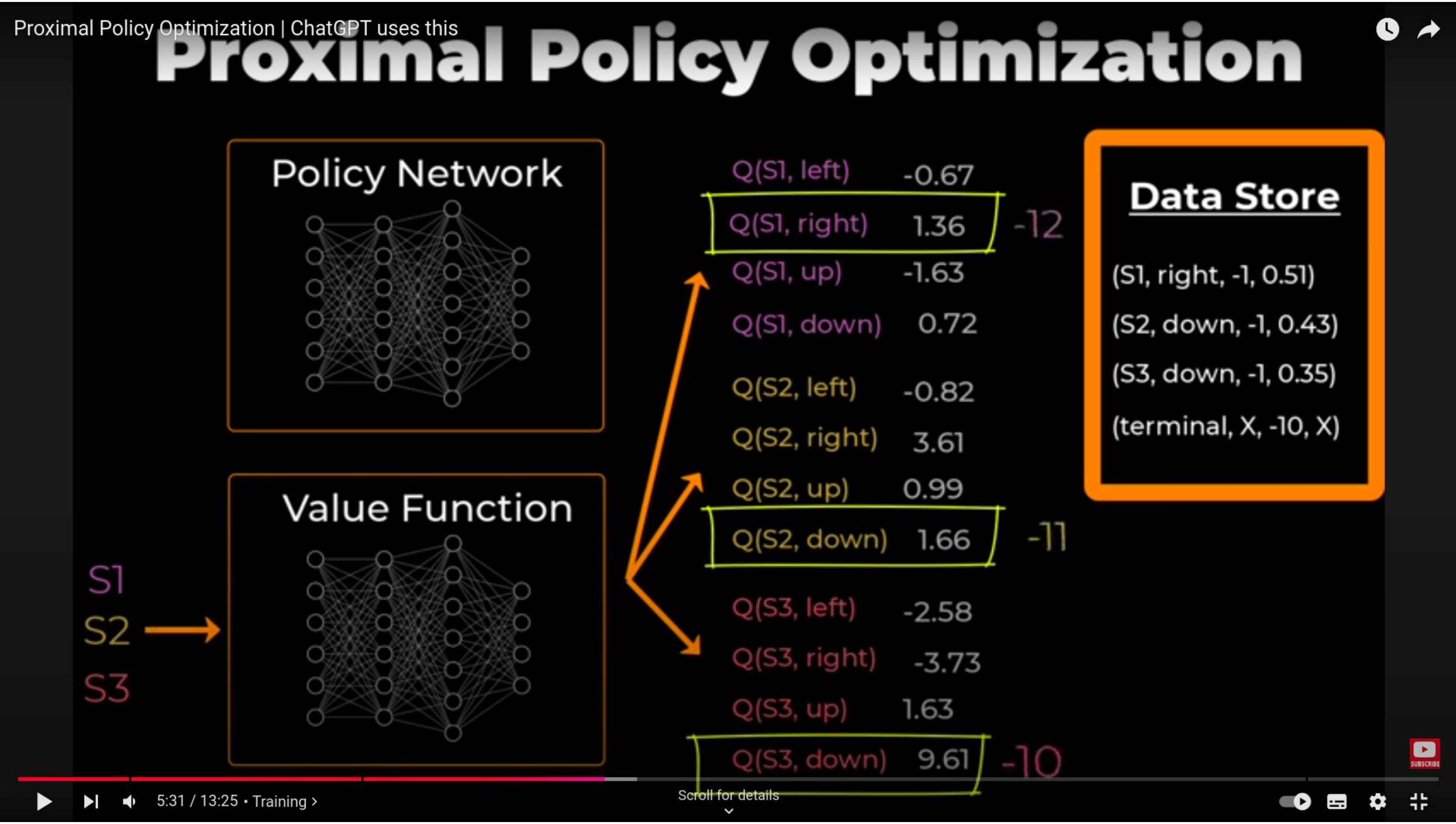

Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
state
Prob of taking a1 in state
Prob of taking a2 in state
Prob of taking a3 in state
Prob of taking a4 in state
Value Function
state
Q(S, a1)
Q(S,a2)
Q(S,a3)
Q(S,a4)
2:23 / 13:25 • Architectures ›
Scroll for details
SUBSCRIBE


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
S1
P(S1, left) 0.02
P(S1, right) 0.51
P(S1, up) 0.01
P(S1, down) 0.46
Data Store
(S1, right, -1, 0.51)
S2
-1
-1
-1
-1
-1
-1
-10
+10
SUBSCRIBE
4:30 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
S2
P(S2, left) 0.15
P(S2, right) 0.32
P(S2, up) 0.10
P(S2, down) 0.43
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
-1
-1
-1
-1
-1
-1
-1
ST
+10
4:50 / 13:25 • Training
Scroll for details
SUBSCRIBE


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
Value Function
S1
S2
S3
Q(S1, left) -0.67
Q(S1, right) 1.36
Q(S1, up) -1.63
Q(S1, down) 0.72
Q(S2, left) -0.82
Q(S2, right) 3.61
Q(S2, up) 0.99
Q(S2, down) 1.66
Q(S3, left) -2.58
Q(S3, right) -3.73
Q(S3, up) 1.63
Q(S3, down) 9.61
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
5:22 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
Value Function
S1
S2
S3
Q(S1, left) -0.67
Q(S1, right) 1.36
-12
Q(S1, up) -1.63
Q(S1, down) 0.72
Q(S2, left) -0.82
Q(S2, right) 3.61
Q(S2, up) 0.99
Q(S2, down) 1.66
-11
Q(S3, left) -2.58
Q(S3, right) -3.73
Q(S3, up) 1.63
Q(S3, down) 9.61
-10
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
5:31 / 13:25 • Training
Scroll for details

Proximal Policy Optimization | ChatGPT uses this

Proximal Policy Optimization

Q(S1, left)	-0.67	
Q(S1, right)	1.36	-12
Q(S1, up)	-1.63	
Q(S1, down)	0.72	
Q(S2, left)	-0.82	
Q(S2, right)	3.61	
Q(S2, up)	0.99	
Q(S2, down)	1.66	-11
Q(S3, left)	-2.58	
Q(S3, right)	-3.73	
Q(S3, up)	1.63	
Q(S3, down)	9.61	-10

Advantage = 1.36 - (-12) = 12.36

Advantage = 1.66 - (-11) = 12.66

Advantage = 9.61 - (-10) = 19.61

↓

Value Function Network Loss function

↓

Value Function Network Loss

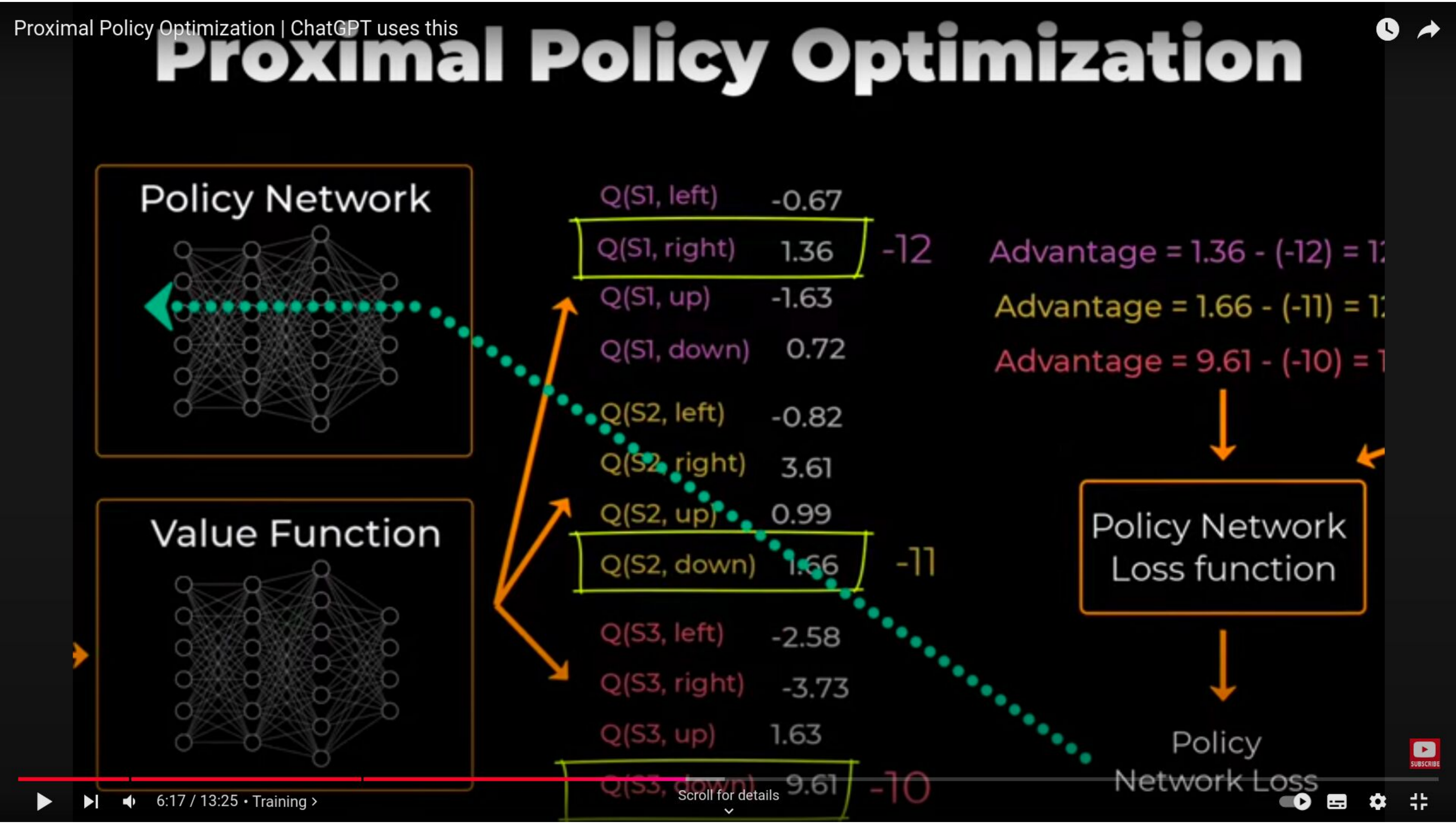
Data Store

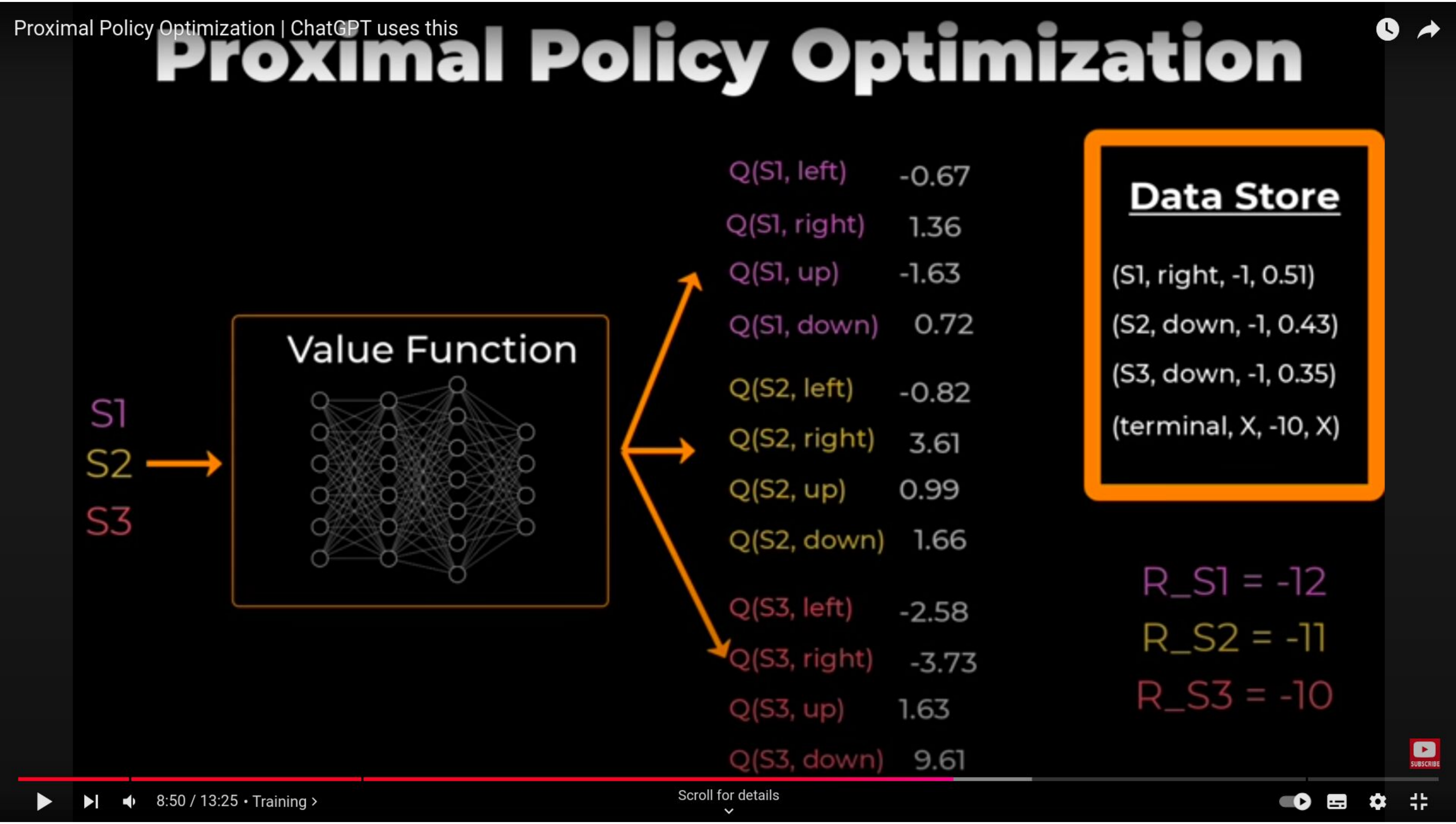
(S1, right, -1, 0.51)

(S2, down, -1, 0.43)

(S3, down, -1, 0.35)

(terminal, X, -10, X)


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Policy Network
Value Function
Q(S1, left) -0.67
Q(S1, right) 1.36
-12
Q(S1, up) -1.63
Q(S1, down) 0.72
Q(S2, left) -0.82
Q(S2, right) 3.61
Q(S2, up) 0.99
Q(S2, down) 1.66
-11
Q(S3, left) -2.58
Q(S3, right) -3.73
Q(S3, up) 1.63
Q(S3, down) 9.61
-10
Advantage = 1.36 - (-12) = 1
Advantage = 1.66 - (-11) = 1
Advantage = 9.61 - (-10) = 1
Policy Network Loss function
Policy Network Loss
6:17 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
S1
S2
S3
Value Function
Q(S1, left) -0.67
Q(S1, right) 1.36
Q(S1, up) -1.63
Q(S1, down) 0.72
Q(S2, left) -0.82
Q(S2, right) 3.61
Q(S2, up) 0.99
Q(S2, down) 1.66
Q(S3, left) -2.58
Q(S3, right) -3.73
Q(S3, up) 1.63
Q(S3, down) 9.61
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
R_S1 = -12
R_S2 = -11
R_S3 = -10
8:50 / 13:25 • Training
Scroll for details

Proximal Policy Optimization | ChatGPT uses this

Proximal Policy Optimization

Q(S1, left)	-0.67
Q(S1, right)	1.36
Q(S1, up)	-1.63
Q(S1, down)	0.72
Q(S2, left)	-0.82
Q(S2, right)	3.61
Q(S2, up)	0.99
Q(S2, down)	1.66
Q(S3, left)	-2.58
Q(S3, right)	-3.73
Q(S3, up)	1.63
Q(S3, down)	9.61

Data Store

(S1, right, -1, 0.51)

(S2, down, -1, 0.43)

(S3, down, -1, 0.35)

(terminal, X, -10, X)

R_S1 = -12

R_S2 = -11

R_S3 = -10

Advantage_S1 = 1.36 - (-12) = 12.36

Advantage_S2 = 1.66 - (-11) = 12.66

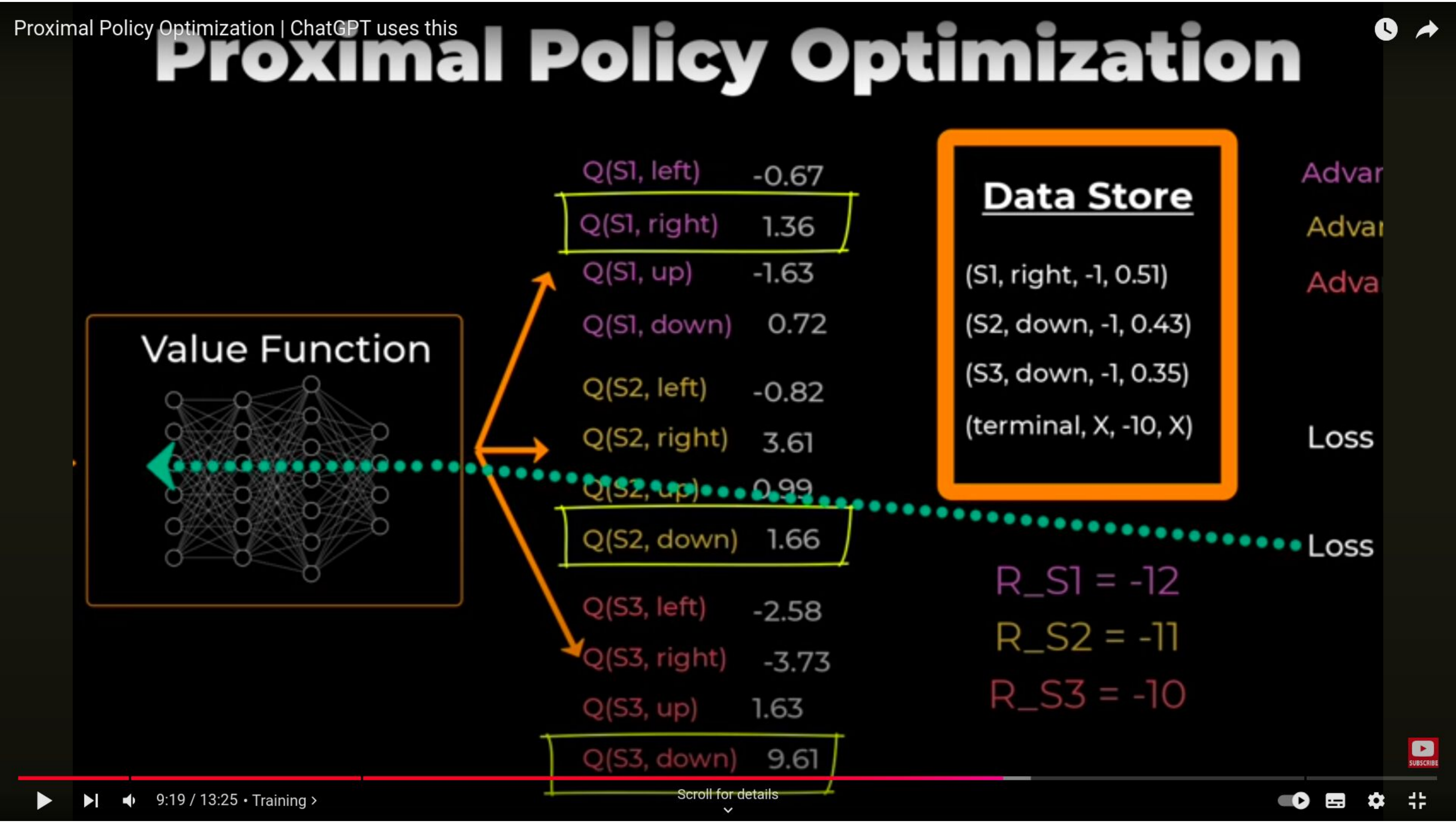
Advantage_S3 = 9.61 - (-10) = 19.61

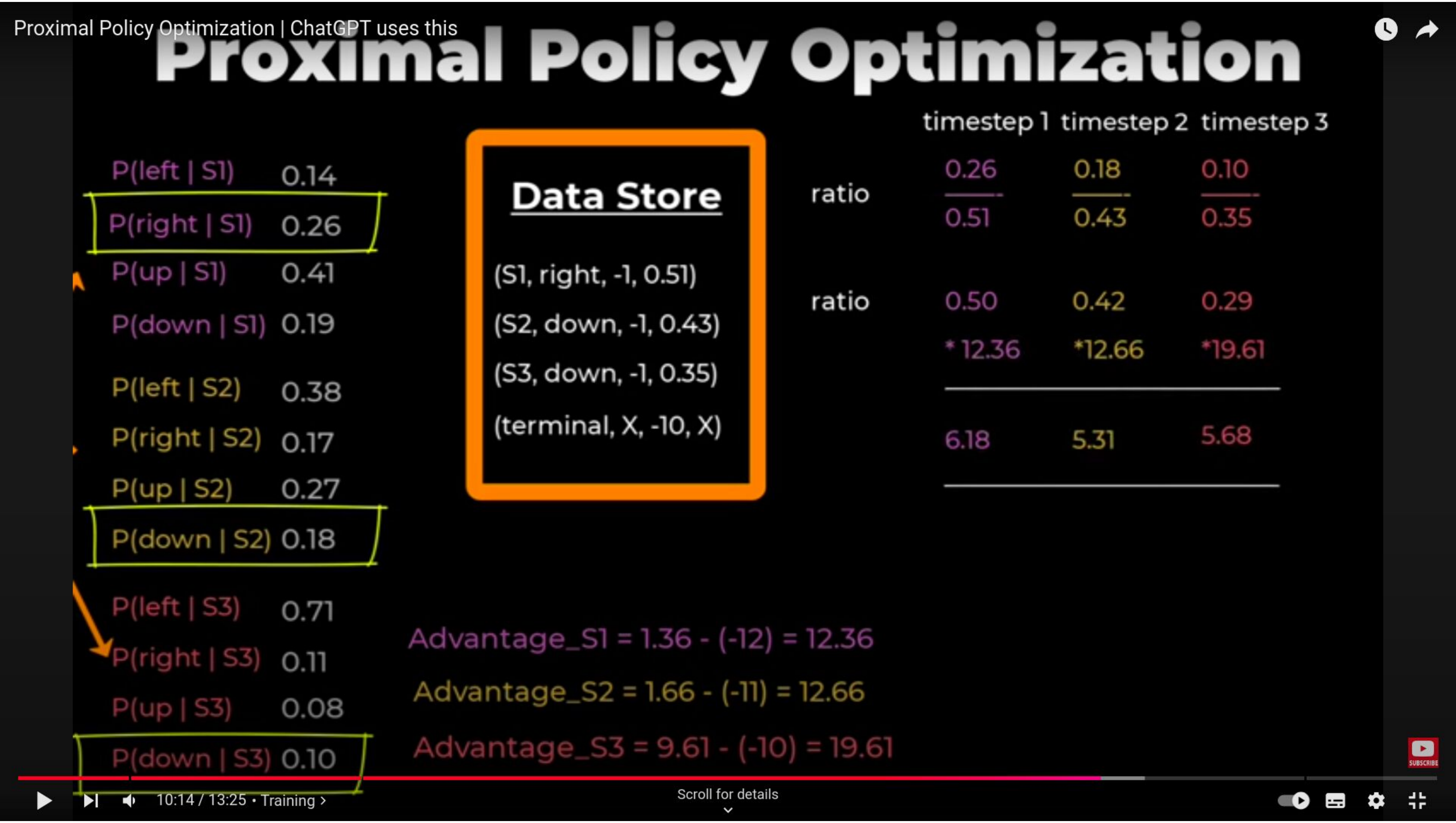
$$\text{Loss} = \frac{12.36^2 + 12.66^2 + 19.61^2}{3}$$

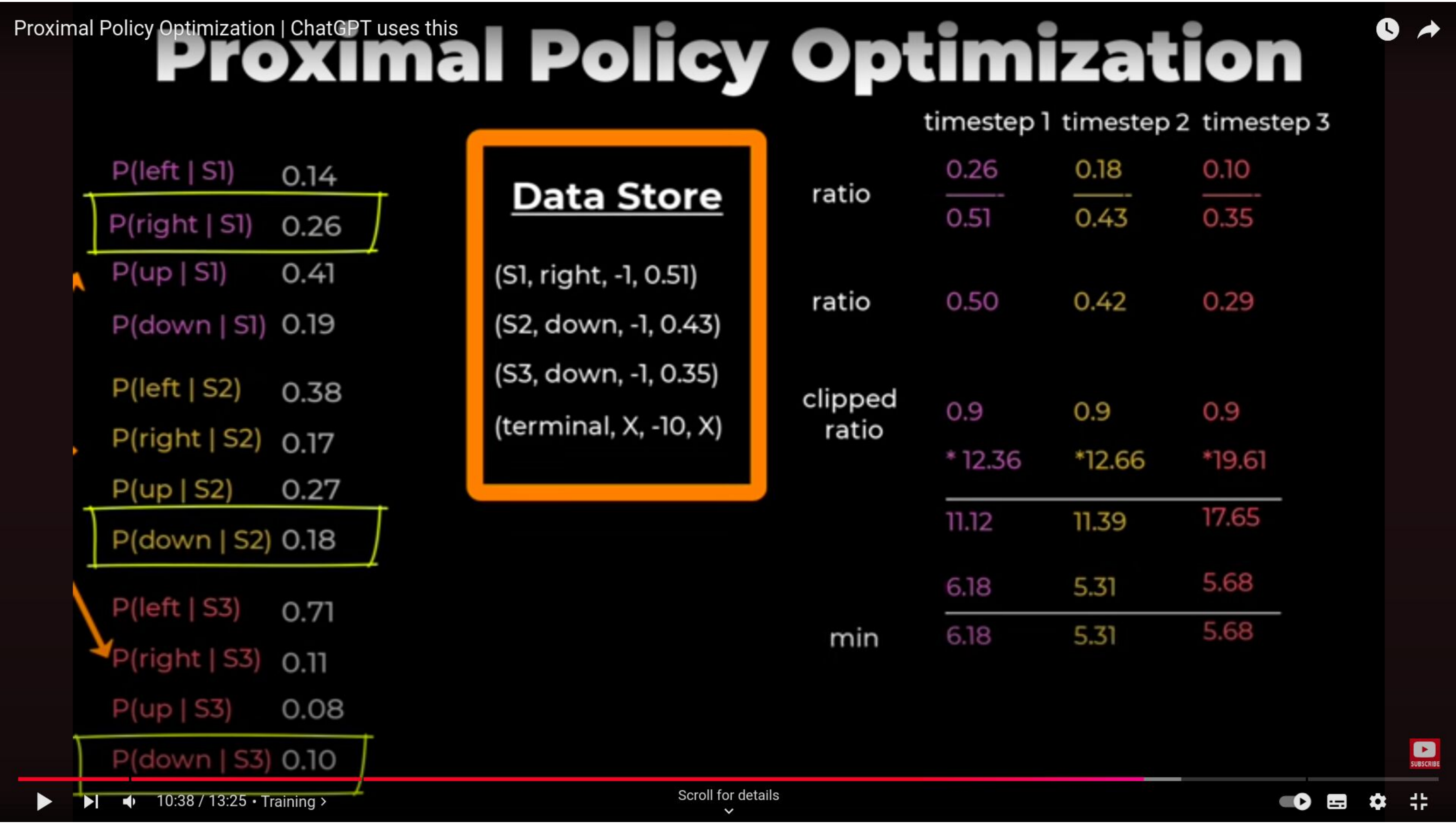
Loss = 252.53

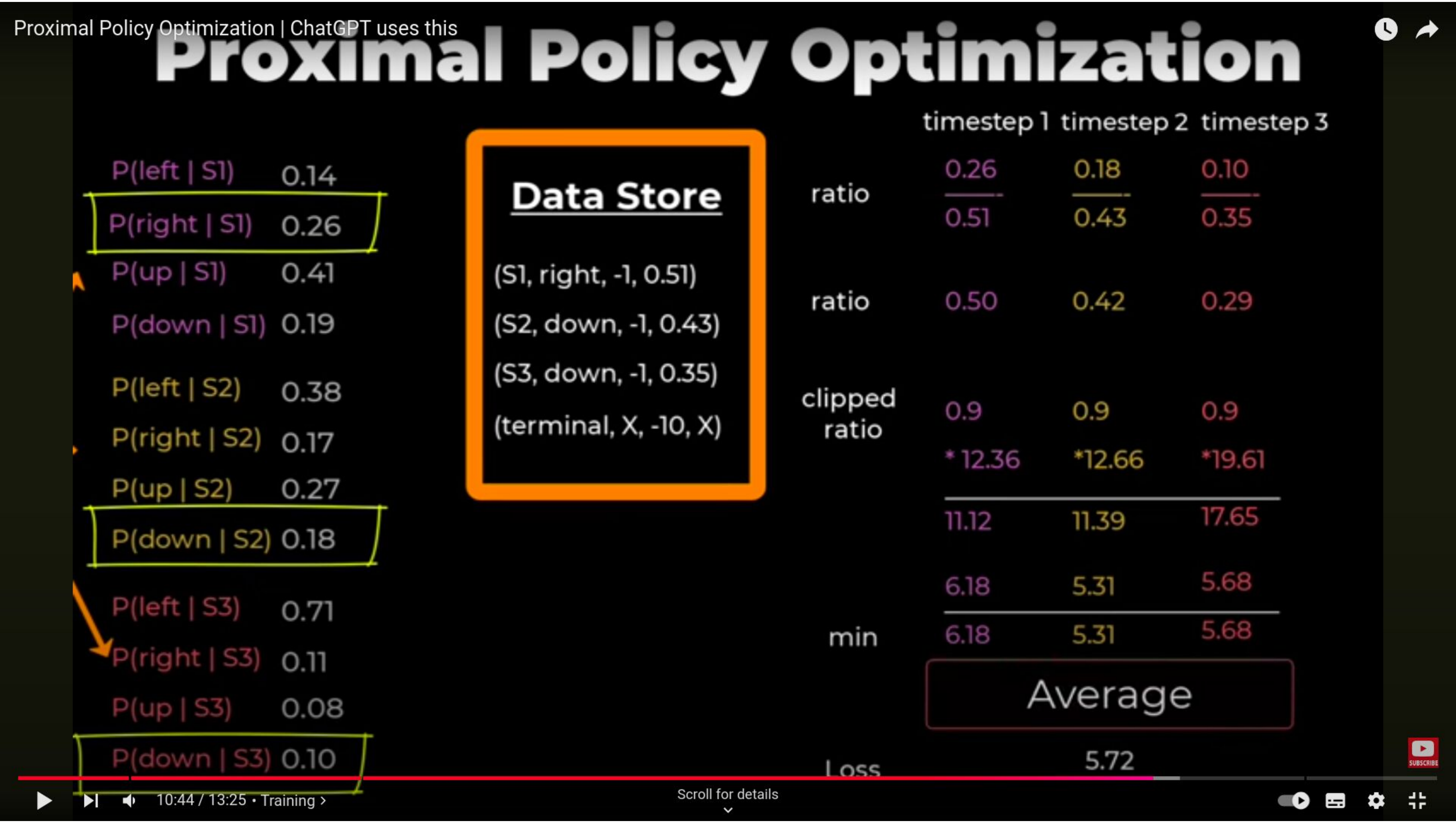
9:07 / 13:25 • Training ›

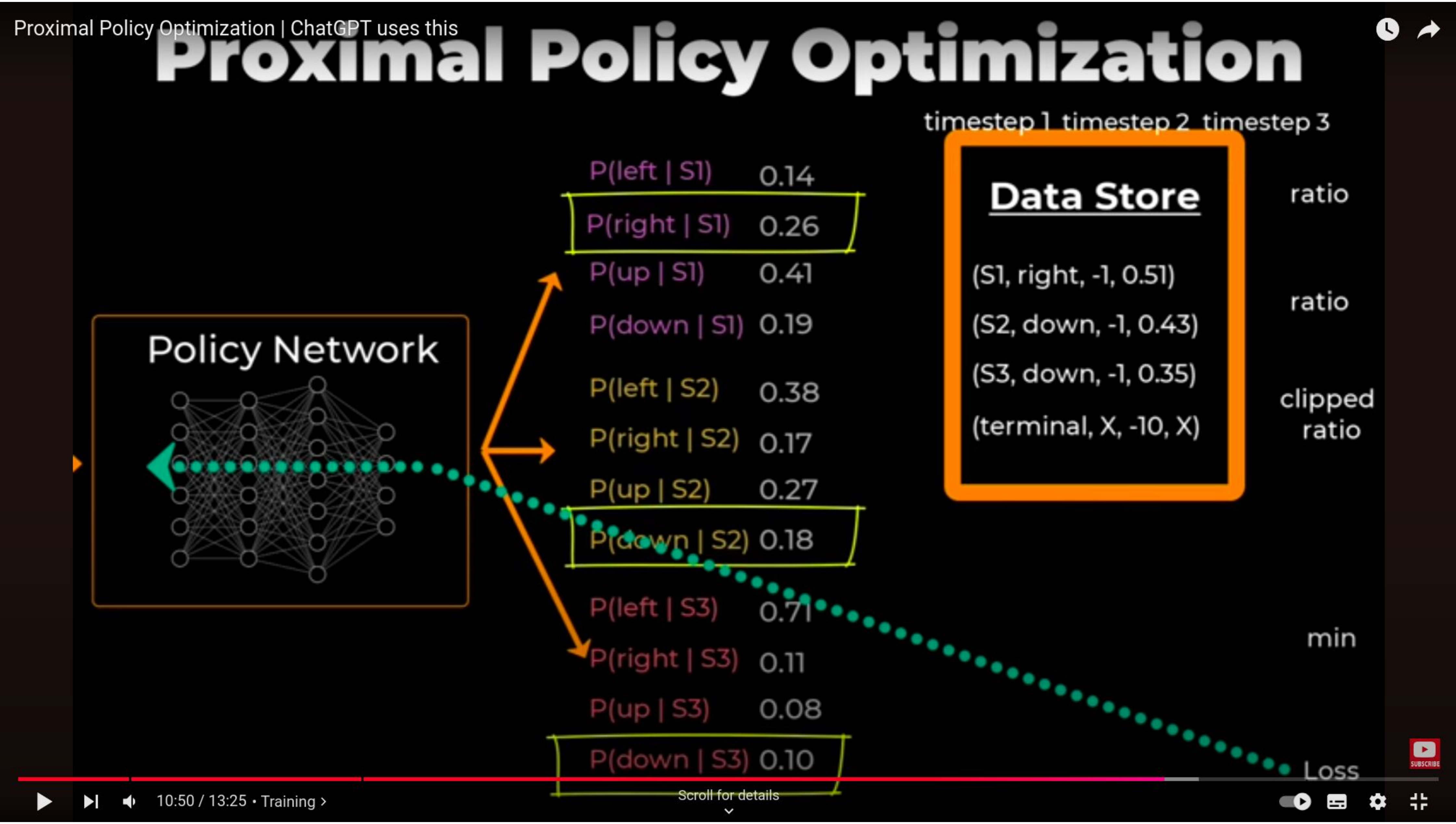
Scroll for details

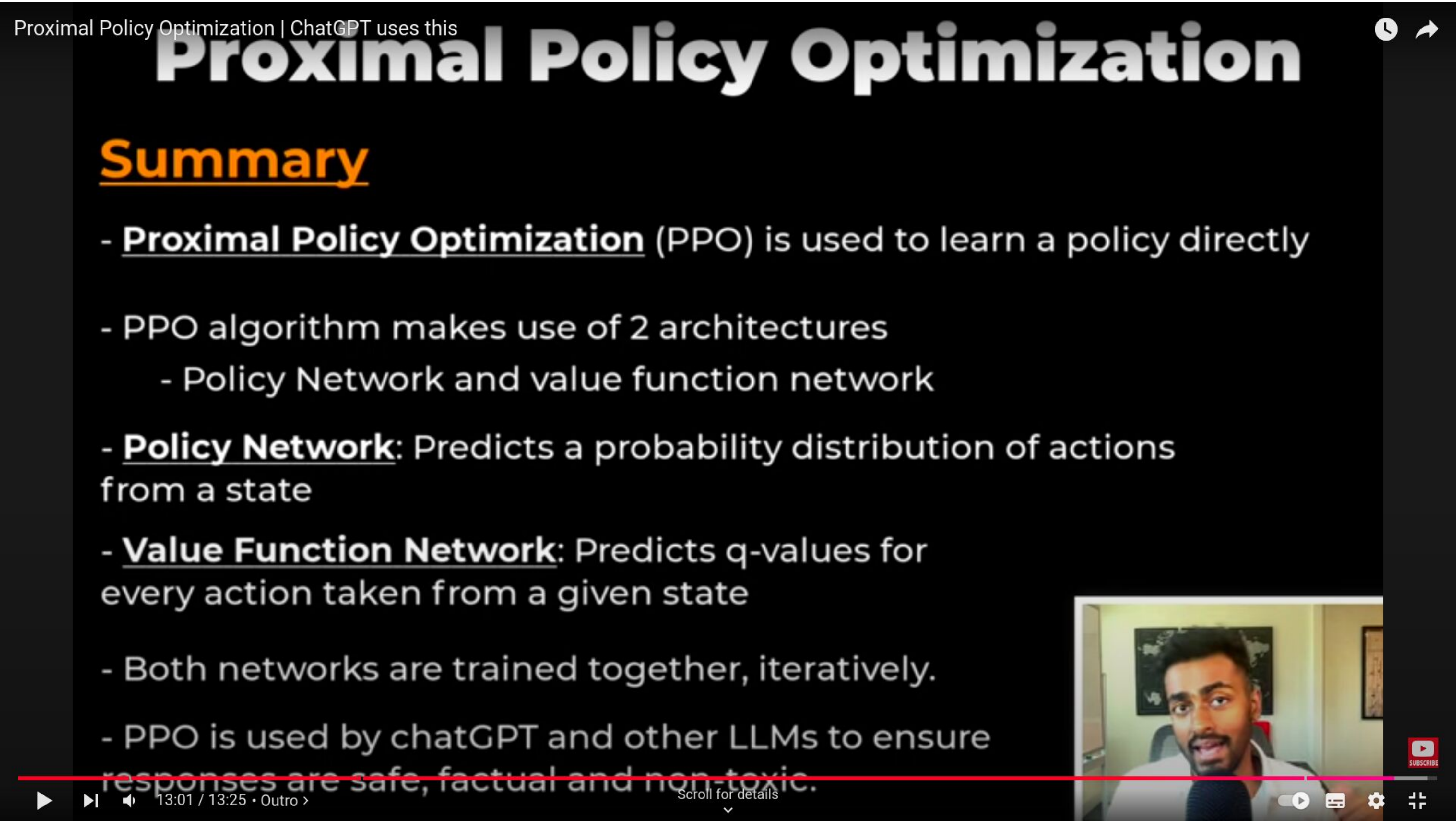

Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Value Function
Q(S1, left) -0.67
Q(S1, right) 1.36
Q(S1, up) -1.63
Q(S1, down) 0.72
Q(S2, left) -0.82
Q(S2, right) 3.61
Q(S2, up) 0.99
Q(S2, down) 1.66
Q(S3, left) -2.58
Q(S3, right) -3.73
Q(S3, up) 1.63
Q(S3, down) 9.61
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
R_S1 = -12
R_S2 = -11
R_S3 = -10
Loss
Loss
9:19 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
timestep 1 timestep 2 timestep 3
P(left | S1) 0.14
P(right | S1) 0.26
P(up | S1) 0.41
P(down | S1) 0.19
P(left | S2) 0.38
P(right | S2) 0.17
P(up | S2) 0.27
P(down | S2) 0.18
P(left | S3) 0.71
P(right | S3) 0.11
P(up | S3) 0.08
P(down | S3) 0.10
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
ratio 0.26/0.51 0.18/0.43 0.10/0.35
ratio 0.50 0.42 0.29
* 12.36 *12.66 *19.61
6.18 5.31 5.68
Advantage_S1 = 1.36 - (-12) = 12.36
Advantage_S2 = 1.66 - (-11) = 12.66
Advantage_S3 = 9.61 - (-10) = 19.61
10:14 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
P(left | S1) 0.14
P(right | S1) 0.26
P(up | S1) 0.41
P(down | S1) 0.19
P(left | S2) 0.38
P(right | S2) 0.17
P(up | S2) 0.27
P(down | S2) 0.18
P(left | S3) 0.71
P(right | S3) 0.11
P(up | S3) 0.08
P(down | S3) 0.10
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
timestep 1 timestep 2 timestep 3
ratio 0.26/0.51 0.18/0.43 0.10/0.35
ratio 0.50 0.42 0.29
clipped ratio 0.9 0.9 0.9
* 12.36 *12.66 *19.61
11.12 11.39 17.65
6.18 5.31 5.68
min 6.18 5.31 5.68
10:38 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
P(left | S1) 0.14
P(right | S1) 0.26
P(up | S1) 0.41
P(down | S1) 0.19
P(left | S2) 0.38
P(right | S2) 0.17
P(up | S2) 0.27
P(down | S2) 0.18
P(left | S3) 0.71
P(right | S3) 0.11
P(up | S3) 0.08
P(down | S3) 0.10
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
timestep 1 timestep 2 timestep 3
ratio
0.26 0.18 0.10
0.51 0.43 0.35
ratio 0.50 0.42 0.29
clipped ratio
0.9 0.9 0.9
* 12.36 *12.66 *19.61
11.12 11.39 17.65
6.18 5.31 5.68
min 6.18 5.31 5.68
Average
Loss 5.72
10:44 / 13:25 • Training
Scroll for details
SUBSCRIBE


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
timestep 1 timestep 2 timestep 3
Policy Network
P(left | S1) 0.14
P(right | S1) 0.26
P(up | S1) 0.41
P(down | S1) 0.19
P(left | S2) 0.38
P(right | S2) 0.17
P(up | S2) 0.27
P(down | S2) 0.18
P(left | S3) 0.71
P(right | S3) 0.11
P(up | S3) 0.08
P(down | S3) 0.10
Data Store
(S1, right, -1, 0.51)
(S2, down, -1, 0.43)
(S3, down, -1, 0.35)
(terminal, X, -10, X)
ratio
ratio
clipped ratio
min
Loss
10:50 / 13:25 • Training
Scroll for details


Proximal Policy Optimization | ChatGPT uses this
Proximal Policy Optimization
Summary
- Proximal Policy Optimization (PPO) is used to learn a policy directly
- PPO algorithm makes use of 2 architectures
- Policy Network and value function network
- Policy Network: Predicts a probability distribution of actions from a state
- Value Function Network: Predicts q-values for every action taken from a given state
- Both networks are trained together, iteratively.
- PPO is used by chatGPT and other LLMs to ensure responses are safe, factual and non-toxic.
13:01 / 13:25 • Outro
Scroll for details
SUBSCRIBE

PPO Algorithm Pseudocode with Key Formulas

1. **Initialize:**
    - Policy network π_θ and value network V_θ with parameters θ.
    - Set hyperparameters:
        - Clipping parameter ϵ.
        - Value loss coefficient c_1.
        - Entropy coefficient c_2.
        - Discount factor γ.
        - GAE parameter λ.
        - Learning rate α.
2. **Repeat for each iteration:**

    a. **Collect Trajectory:**

    - Run the current policy π_θ for T timesteps to collect trajectories (s_t, a_t, r_t, s_{t+1}).

    b. **Calculate Advantages and Returns:**

    For each timestep t in the trajectory:

    1. **Compute discounted return:**

$$\hat{R}_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

2. **Calculate Generalized Advantage Estimate (GAE):**

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}$$

where:

$$\delta_t = r_t + \gamma V_\theta(s_{t+1}) - V_\theta(s_t)$$

c. **Optimize Policy and Value Network:**

For each minibatch of sampled trajectories:

1. **Calculate probability ratio $r_t(\theta)$:**

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

2. **Compute clipped policy loss L^{clip}:**

$$L^{\text{clip}} = \mathbb{E}_t \left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right]$$

3. **Compute value function loss L^{value}:**

$$L^{\text{value}} = \frac{1}{2}\mathbb{E}_t \left[\left(V_\theta(s_t) - \hat{R}_t\right)^2\right]$$

4. **Compute entropy bonus L^{entropy}:**

$$L^{\text{entropy}} = \mathbb{E}_t \left[\mathcal{H}\left[\pi_\theta(\cdot|s_t)\right]\right]$$

5. **Calculate total PPO loss L^{total}:**

$$L^{\text{total}} = L^{\text{clip}} - c_1 L^{\text{value}} + c_2 L^{\text{entropy}}$$

6. **Update policy and value network parameters θ:**

$$\theta \leftarrow \theta - \alpha \nabla_\theta L^{\text{total}}$$

d. **Update old policy parameters:**

$$\theta_{\text{old}} \leftarrow \theta$$

3. **End Repeat.**

Compute Entropy:

- For discrete action space: Sum over all possible actions using the formula:

$$\mathcal{H}[\pi_\theta(\cdot|s_t)] = -\sum_a \pi_\theta(a|s_t) \log \pi_\theta(a|s_t)$$