CHAPTER NINE

Reading the Learned Policy

> *This is the page the whole book was walking toward. 175 numbers become a coloured grid you can read like a map — and, just as valuable, you can read where the agent is only guessing.*

Chapter 8 ended with a promise: a good reward curve earns you the right to look at the policy, but does not excuse you from looking. So we look. The **Learned Policy** panel takes the 175 Q-values and collapses them into 35 decisions — for each state, the single best action — then paints each cell by what that action is: **green** to withdraw, **grey** to hold, **red** to insert. This is the destination sketched in Chapter 1 and left blank in Chapter 4. Now it is full, and real, and — the entire point of the series — readable.

What you already know

You read coloured 2-D maps all the time: a heatmap, a confusion matrix, a spreadsheet under conditional formatting, a decision table shaded by outcome. You already scan such a grid for structure — a band of colour, a diagonal, an outlier cell — and read behaviour straight off it. A learned policy asks for exactly that skill. The agent's whole strategy is a small coloured table, and the patterns in it *are* its behaviour.

How the grid is built

Nothing new is computed here. Each of the 35 cells is one state — row by power deviation, column by trend, from Chapter 4 — and its colour is simply `Greedy(state)` from Chapter 5, the argmax over that state's five Q-values, mapped to a colour. The console paints all 35 in one short loop:

```
for (int dev = 6; dev >= 0; dev--)          // rows: far below → far above
    for (int tr = 0; tr < 5; tr++)            // columns: falling → rising
    {
        int a = Greedy(dev * 5 + tr);          // the state's best action
        Paint(dev, tr, ColourOf(a));            // green / grey / red
    }
```

The grid is therefore not a separate artefact that can drift from the table; it *is* the table, rendered. Change one Q-value enough to flip an argmax and the corresponding cell changes colour. What you see is exactly what the agent will do.

SEE IT IN NEXUS-1 · OPTIMIZATION (RL), LEARNED POLICY (INTERPRETABLE)

*The whole mind of the agent, on one panel. A grey **hold** band runs through the centre; a single green **withdraw +8** sits where power is below target; reds **insert** elsewhere. Read the centre and it is sensible. Read the edges and something stranger is going on — which turns out to be the most useful thing the grid can tell you.*

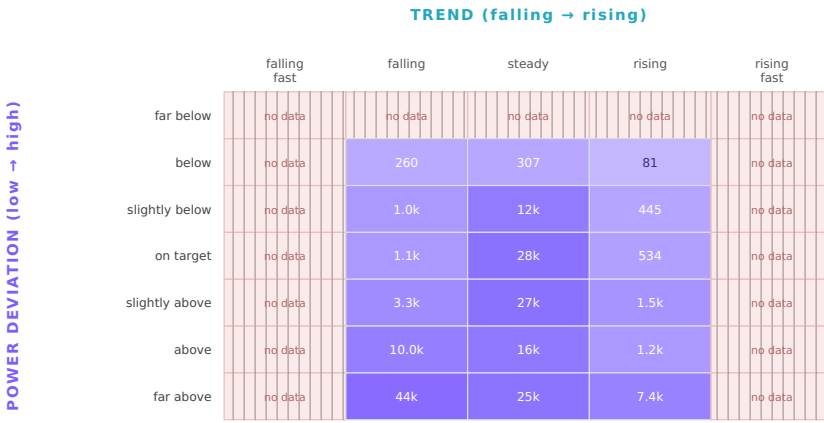
Reading the sensible core

Start in the middle, where the agent has real experience. Through the centre column — power steady, near target — runs a grey band: **hold**. That is the dead-band from Chapter 2 made visible; near target, with the rod movement penalty in force, sitting still beats twitching. Just above it, where power is *below* target and rising, sits the lone green cell: **withdraw +8**, pull rods out to bring power up — the one place lifting power is clearly right. And where power runs *above* target the cells turn red: **insert**, push rods in to bring it down. Hold near target, lift when low, lower when high: the core is exactly the controller you would have hand-written, except no one wrote it.

Reading the edges — where the agent is guessing

Now the strange part. The entire top row, the entire bottom row, and both outer columns are a solid wall of red — **insert −8** — including states where inserting is plainly wrong, like the top row, which is power already far *below* target. An insert there would be a mistake. So has the agent learned something foolish? No. It has learned *nothing* there, and the grid is honest enough to show it. Those red borders are not decisions; they are the absence of one.

FIGURE 9.1 · WHERE THE AGENT ACTUALLY HAS EXPERIENCE (VISITS PER STATE)



The same 35 cells, shaded by how often training visited them. ***17 of the 35 states were never visited at all*** *— both outer columns (power changing faster than 1% per step never occurred in any rollout) and the "far below" row (power starts at 100% and targets never exceed it, so power is never far below target). Those cells kept their initial zeros; with all five actions tied at zero, the argmax breaks the tie by taking the first — insert −8. The red borders of the policy grid are precisely this map of inexperience.*

This is the quiet triumph of a readable table, and it is worth saying plainly. A neural network trained on the same task would also output *something* for those unseen states — an action delivered with the same confident tone as every other, with nothing on its face to say "I have never been here." The grid, by contrast, wears its ignorance openly: the uniform red borders, and the flat all-near-zero rows behind them in the **Why This Action** bars of Chapter 5, are a visible warning that those cells are untrustworthy. You can see what the agent knows *and* what it does not, audit any cell, and decline its advice exactly where it has no business giving any.

Why readable beats a black box

For an advisory system a human must approve, this is not an aesthetic preference; it is the whole design. Interpretability is what lets an operator scan 35 cells in a glance, recognise the sensible core, notice the unlearned borders, and treat the two differently. It is what lets a reviewer find a wrong cell and refuse it. The price was paid back in Chapters 4 and 5 — 35 states and 5 actions is a coarse, limited brain — but the dividend is collected here: every decision, and every *non*-decision, is on the table, in colour, on one screen.

HONEST BOUNDARY

A readable wrong cell is far safer than a hidden one, but it is still wrong, and reading the grid does not repair it — it only reveals it. The borders here are an artefact of the surrogate and the curriculum: power that starts at 100% against targets of 70–100% simply never produces "far below target" or violent trends, so the agent never learns those states. A different training regime would fill some of them in; none would fill all. In deployment two things keep the unlearned cells harmless: the safety clamp of Chapter 10, which bounds whatever the policy suggests to a validated band, and the advisory framing, under which a human reads the grid — borders and all — before anything happens. Interpretability does not make the agent right; it makes the agent's wrongness *visible*, which is the difference between a risk you manage and one you cannot see. Chapter 10 turns this readable policy into clamped, advisory action.