

Lesson 23, Part 2 — Auto-Aim Without RLHF: Two Other Ways to Build the Visor

Draft — not yet published. Part 1 asked whether a reward model could learn Soldier 76's Tactical Visor from play. This part: suppose we skip RLHF entirely.

1. Behavioral cloning — steal the aim, skip the reward

Don't learn what good aiming is. Learn what good aimers do. Record thousands of hours of skilled players — game state in, crosshair movement out — and train a network to predict the next mouse flick. Pure supervised learning. No reward model, no RL loop, no PPO.

The reward problem disappears. Part 1's headache — "hits-on-target is gameable; how the crosshair gets there is not" — never arises, because nothing is being optimized. You're copying trajectories, and human trajectories already contain the "how."

Then compounding errors arrive. The clone drifts off the states it trained on and starts predicting in territory it never saw — each mistake feeds the next. DAgger (Dataset Aggregation) patches it: let the clone play, then ask the human "what would *you* have done here?" and retrain on the corrections. And note the ceiling: it can never exceed its teachers, and it faithfully copies their bad habits too.

Verdict: fastest path to a working visor. Ceiling is "as good as the demos."

2. Classical tracking + PID — no learning at all

This is robotics, not ML. Estimate where the target will be — Kalman filter on position and velocity, or just linear extrapolation — then drive the crosshair onto the predicted point with a PID controller. This is literally how classic aimbots worked before neural nets ate the world.

No data, no training, no reward. It's interpretable — read the gains and you know exactly why it snapped left. And it's *precise*: a well-tuned PID on a clean target estimate out-reacts any human alive.

But it's inhuman in exactly the way Part 1 cared about. Perfect tracking with zero hesitation is trivially detectable (anti-cheat loves this) and tactically dumb — it doesn't prioritize targets, doesn't manage the visor's cooldown economy, doesn't know when *not* to shoot. The "how the crosshair gets there" is mathematically perfect and completely wrong.

Verdict: best raw aim, zero game sense. A visor with no brain.

The comparison

	RLHF (Part 1)	Behavioral cloning	PID tracking
Needs	reward model + RL	human demos	target estimator
Reward engineering	yes — the hard part	none	none
Failure mode	reward hacking	compounding errors	inhuman, detectable
Ceiling	beyond human	as good as demos	perfect aim, zero judgment

The punchline: RLHF was the only approach that could learn *judgment* — when to engage, how to move like a player instead of a machine. Cloning borrows judgment from humans; PID abandons it. If the visor is just "snap to heads," PID wins in an afternoon. If it's supposed to play like a top-500

Soldier, you're back to Part 1's reward problem — which is why the reward signal was the whole problem all along.