July 2026

Reward Engineering

Research Brief · No. 1

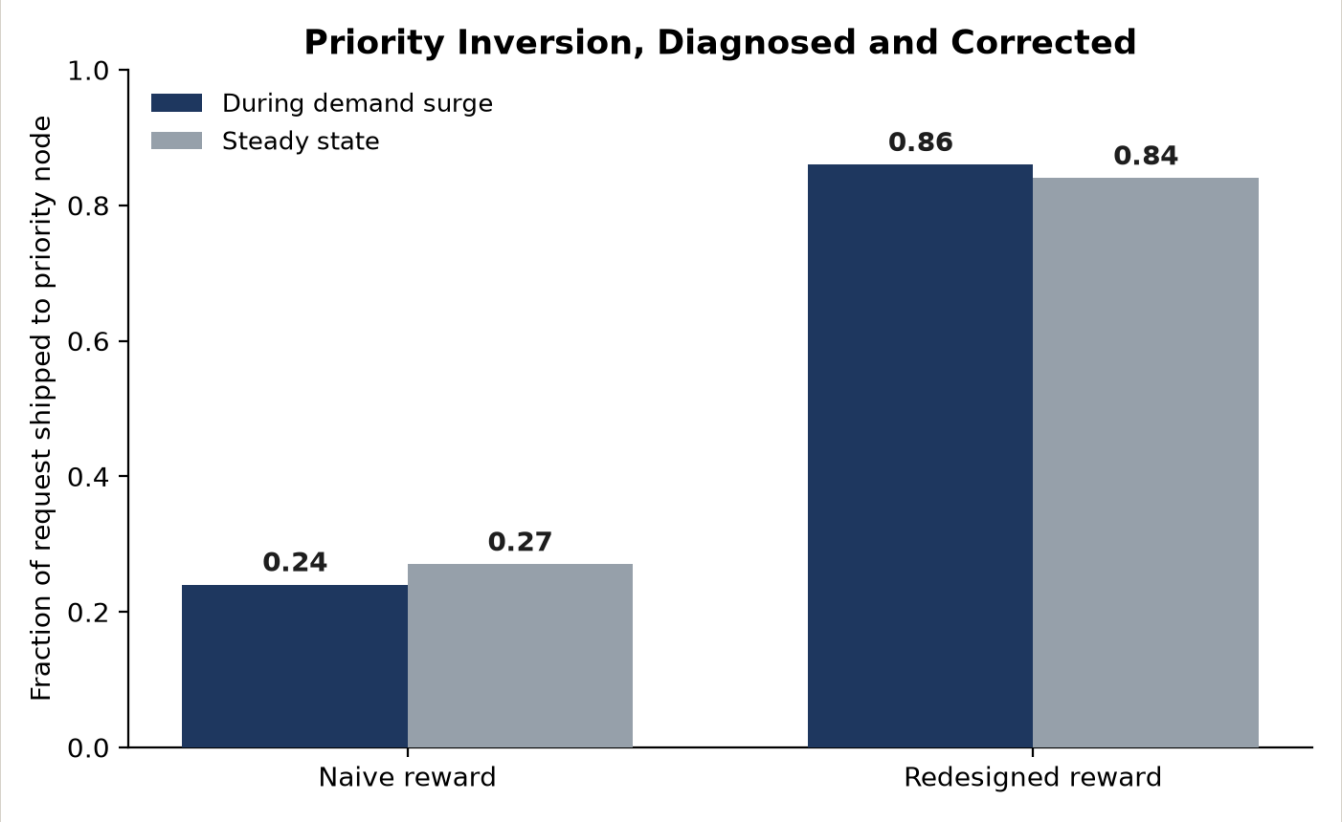
Diagnosing and correcting reward misspecification in a simulated multi-echelon resupply task

Dennis Pezan

Cadet, United States Military Academy at West Point

Diagnosing and Correcting Reward Misspecification in a Simulated Multi-Echelon Resupply Task

Dennis Pezan July 2026



The central finding. Under the original reward the agent shipped a smaller fraction of requested volume to the priority node during demand surges (0.24) than in steady state (0.27), inverting the intended ordering. The redesigned reward corrects it. (Source: author, 101-seed evaluation.)

Executive Summary

- A reinforcement learning agent optimizes the reward it is given, not the behavior its designer intended. Under a naive reward, the agent studied here outscored every hand-coded baseline while behaving worse than all of them on the dimension the task existed to test.
- Four distinct specification failures were isolated, quantified, and independently reproduced on a fresh 900,000-step training run: decision-point inflation, priority inversion, return variance, and costless transport.
- A multi-term redesign, with every term traceable to one diagnosed failure, raised mean request fulfillment from 24.0 to 81.1 percent, corrected the priority inversion, and beat a

hand-coded priority-aware heuristic by 90.93 to 53.29 across 101 held-out scenario seeds.

- The margin over a priority-blind greedy heuristic was +4.27 and did *not* reach statistical significance (paired $t(100) = 1.464$, $p = 0.146$). This is reported rather than omitted, and is attributed to the network's structural scarcity rather than to a deficiency in the reward.
- The transferable result is methodological: reward curves are not evidence of task performance. Reward-independent behavioral auditing is the necessary instrument for detecting specification failure.

Introduction

Reinforcement learning frames sequential decision-making as the maximization of expected cumulative reward. In practice the reward function is not given by the problem; it is authored by the engineer, and it is where most of the design risk lives. An agent trained on a reward that only approximately encodes the designer's intent will optimize the approximation, including whatever gap separates it from the intent. This phenomenon has been documented as specification gaming [1] and as reward hacking [2], and has been shown to intensify rather than diminish as agent capability increases [3].

The practical difficulty is that specification failure is not visible in the signal most practitioners monitor. A reward curve that climbs and plateaus is consistent with an agent that has learned the task and equally consistent with an agent that has learned to exploit the reward. Distinguishing the two requires measurement the reward cannot influence.

This brief reports a case study in which that distinction proved decisive. The environment is a small multi-echelon resupply network: three capacity-constrained warehouses with finite vehicle fleets serve two demand nodes consuming two commodity types, and one node undergoes scheduled surges during which its consumption roughly triples and its assigned priority doubles. The agent's decision, taken each time a node requests replenishment, is how to allocate that request across the three warehouses. The environment is deliberately small; its purpose is not realism but attribution.

The Environment

Three warehouses each hold two commodity types, A and B, with a capacity of 50,000 units per commodity and an initial stock of 30,000. Each operates two vehicles of 5,000-unit capacity drawn from a shared pool. Warehouses are never replenished, so total network supply is fixed at the start of an episode and scarcity is structural rather than incidental.

Two demand nodes each hold both commodities, with capacity and initial stock of 10,000 units per commodity and a reorder threshold of 5,000. Node 0, the priority node, consumes commodity B at a baseline rate outside surge windows and at roughly triple that rate during them. Two surge windows occur per episode, at simulation times 150 to 300 and 450 to 600, within an episode of length 720. Node 0's priority weight rises from 0.5 to 1.0 during a surge; Node 1's remains fixed at 0.5.

At each pause the agent observes a 25-dimensional vector covering node stock and priority, warehouse stock and vehicle availability, elapsed and time-to-surge, a one-hot encoding of the request, and a fulfillability block giving what fraction of the current request each warehouse could satisfy. The action is a vector in $[0,1]^3$ interpreted as an allocation over warehouses rather than as three independent volume controls.

The Baseline Reward

The reward under study at the outset was the mean on-hand stock fraction across both demand nodes and both commodities, paid at every decision:

```
R_naive(t) = (1/4) Σ b(i,k) / cap(i,k)
```

This reads four state variables. It does not read priority, surge state, transport cost, or the number of decisions taken. Trained on it, the agent attained a mean episode return of 63.60 against 27.86 for the strongest hand-coded baseline, an apparent margin of more than two to one.

Diagnostic Method

Two instruments were built. The principle behind both is that a reward function cannot be validated against itself.

Reward-responsiveness battery

The first evaluates the reward directly by running six fixed policies through it, from do-nothing to a hand-coded priority-aware rule, reporting return distributions, per-decision return, and a phase-resolved breakdown separating surge from steady-state decisions. Its purpose is to ask whether the reward discriminates between behaviors a domain expert would rank differently, and whether return is driven by behavior quality or by an incidental variable such as episode length.

Behavioral auditor

The second replays a policy and writes one record per decision, capturing the request, the action, the volume actually shipped from each warehouse, post-decision stock everywhere, and the surge state at the moment of decision. From those records it computes operational metrics the reward does not enter into.

Return is the quantity under suspicion. It cannot also serve as the evidence.

Four Diagnosed Failures

To confirm these were properties of the reward rather than of one training run, the naive reward was retrained from scratch for 900,000 timesteps and re-audited across 101 seeds. All four reproduced.

Decision-point inflation

Because reward accrued at every decision, causing additional decisions increased return independently of decision quality. Across the battery, total return correlated with episode length at r = 0.998. The agent took 184.2 decisions per episode against a maximum of 61.8 across all baselines, achieved by systematically under-shipping so requests recurred rather than resolving.

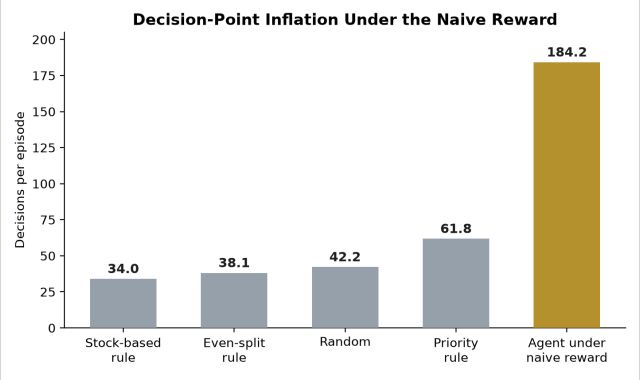


Figure 1. Decisions per episode, 101-seed mean.

Priority inversion

The reward never read the priority weight, and was in fact structurally anti-correlated with it. Because it measured stock as a fraction of capacity, and the priority node's consumption triples during a surge, that node's stock fraction falls during surges regardless of how well it is supplied. Per-decision reward during surges was 0.22 against 0.51 in steady state. The reward paid least at precisely the moments the designer cared about most, and the agent learned the implied lesson.

Return variance

The reward summed an unnormalized per-decision stock fraction across a stochastic simulation with no temporal smoothing, so variation in random draws rather than differences in policy quality dominated the signal, producing unreliable model selection.

Costless transport

Nothing priced logistics activity. Over-provisioning was unpenalized at roughly 84 percent on a random policy, and because no term counted warehouses touched, fragmenting one

shipment across three cost exactly the same as sourcing from one. A trained policy drew from about 2.76 warehouses per decision against the greedy rule's fixed one, more than doubling dispatches for near-identical delivered volume.

Failure	Signature
Decision inflation	r = 0.998; 184.2 vs 61.8 decisions
Priority inversion	0.24 surge vs 0.27 steady
Return variance	Large across-seed spread
Costless transport	~84% over-provision; 2.76 warehouses

Table 1. The four failures and their measured signatures.

The Redesign

The redesigned reward is a weighted sum of one positive term and four penalties, each traceable to a failure above. Here Δt is simulation time since the previous decision, capped so one long idle interval cannot dominate an episode.

```
R(t) =  W_sup   · Δt · Cbar(t)
      − W_under · U(t)
      − ( c_disp·n(t) + c_ship·u(t)
          + W_dwell·Δt·Qbar(t) )
      − W_res   · Δt · Pbar(t)
      − W_stock · Δt · Sbar(t)
```

The barred quantities are two-point trapezoidal time averages. This matters: reading only the end-of-interval value penalizes a policy that ships generously and is consequently not consulted again for a long time, because stock has naturally drawn down by the time that interval closes.

- **Coverage** (1.0) rewards priority- and phase-weighted inventory coverage integrated over elapsed time rather than sampled per decision, so decision count ceases to be a lever.
- **Under-provision** (6.0) applies an immediate squared penalty on the unfilled fraction of the current request, weighted so shortchanging the priority node during a surge costs up to four times more.
- **Transport** charges a flat fee per warehouse drawn from (1.0), a per-unit shipping cost (0.0002), and a dwell charge for volume waiting on a vehicle (0.01).
- **Warehouse reserve** (2.0) penalizes a shared warehouse falling below 30 percent capacity, active only when a surge is imminent or underway.
- **Stockout** (0.5) penalizes any node's coverage falling below a 0.5 safety threshold.

A potential-based shaping term is added, with the coverage potential and discount matched to the trainer. By the result of Ng, Harada, and Russell [4] this cannot alter the optimal policy; it only accelerates credit assignment.

Tested and rejected

Two modifications were rejected on measurement. Raising the warehouse-reserve weight, swept from 2.0 to 24.0, produced a starvation attractor: at every weight, a policy starving the non-priority node outscored any genuine anticipatory strategy. Raising the stockout weight from 0.5 to 2.0 charged policies for structural scarcity rather than poor decisions; the greedy rule's own mean fell from +102 to −37 with no change in its behavior.

Results

The policy is a two-layer 256-unit network trained with PPO [5] for 900,000 timesteps via Stable-Baselines3 [6], evaluated on 101 held-out seeds. Each seed fixes the episode's stochastic draws, so every policy faces an identical scenario, permitting paired comparison.

Policy	Return	Dec.
Uniform random	−80.72	44.4
Even-split rule	−32.61	37.8
Priority-aware rule	53.29	36.4
Greedy stock rule	86.66	34.0
Trained PPO	**90.93**	34.6

Table 2. Mean return across 101 held-out seeds.

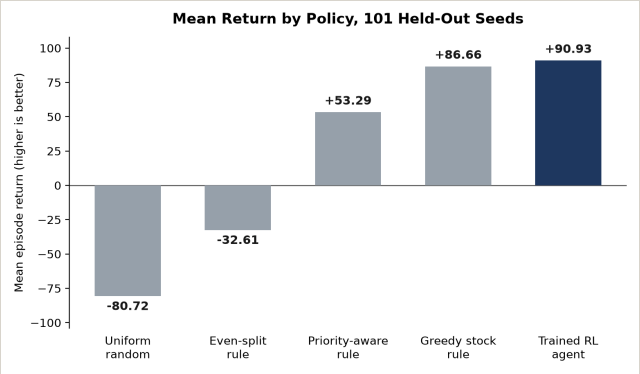


Figure 2. Mean return by policy, 101 seeds.

Return is the quantity under suspicion, so the substantive results are the reward-independent measurements.

Metric	Naive	Redesign
Request fulfillment	24.0%	81.1%
Priority ship, surge	0.24	0.86
Priority ship, steady	0.27	0.84
Decisions / episode	184.2	34.6

Table 3. Operational metrics from per-decision logs.

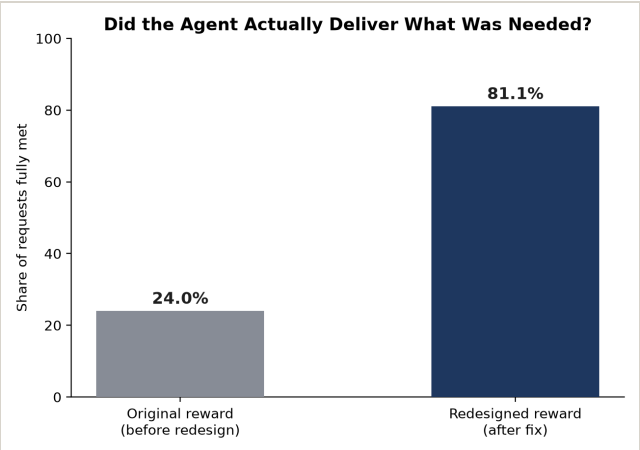


Figure 3. Requests fully met, measured from decision logs.

The greedy comparison

The agent outperforms the priority-aware rule by 37.64 return, a margin not in question. The comparison against greedy is different: the margin is +4.27, and a paired t-test across the 101 seeds gives $t(100) = 1.464$, $p = 0.146$. This does not clear significance at $\alpha = 0.05$, and no claim is made that it does.

This is reported for two reasons. Presenting a +4.27 margin without the test would misrepresent the evidence. And the result is informative: total network supply is fixed and falls short of aggregate demand, so beyond a point, additional prioritization of one node is realizable only by withholding from the other, which the under-provision term correctly penalizes. Under scarcity this severe, sourcing from the fullest warehouse is close to the achievable frontier. The priority-aware rule scores far below greedy precisely because it pursues prioritization past the point the network can support.

Discussion

The central claim is that reward curves are not evidence of task performance. The naive reward produced a return curve that rose and stabilized, and an agent that outscored every hand-coded rule by more than two to one. Every one of those signals was consistent with success. The behavior was inverted on the single dimension the task existed to test. No amount of additional training would have surfaced it, because the agent was not failing to optimize; it was optimizing correctly against an incorrect specification.

A second observation concerns the boundary between reward design and environment design. Two corrections reported here, the action reparameterization and a reorder-lockout defect, are not reward changes at all. Both initially presented as reward problems, and neither would have been fixed by reweighting terms. Distinguishing an incentive defect from a mechanism defect requires the same reward-independent instrumentation used to detect the pathology in the first place.

Finally, the environment exhibits partial observability: the observation does not fully capture in-flight shipments, and the policy is memoryless. On one seed this produced a cascade of 22 consecutive zero-credit decisions, as a

deterministic policy facing near-identical observations repeated an identical failing allocation. This was corrected at the environment level rather than through the reward. A recurrent policy is the natural alternative and remains untested.

Limitations

- Single scenario configuration; generalization to larger or randomized layouts is untested, and weights were tuned against this configuration.
- The 101 seeds vary the evaluation scenario, not the training run. Training-seed reproducibility is not established for the final model.
- PPO was selected on suitability grounds and not benchmarked against alternatives.
- The margin over the greedy baseline is not statistically significant at n = 101.
- Four exploit classes were tested and closed; this establishes that a search was conducted, not that the reward is unexploitable.

Conclusion

An agent trained on a naive inventory reward outscored every hand-coded baseline while inverting the priority ordering the task was built to test, inflating decision volume threefold, and fragmenting shipments at no cost. A multi-term redesign raised request fulfillment from 24.0 to 81.1 percent and corrected the inversion.

The transferable result is not the reward function, which is tuned to one small configuration. It is the procedure: instrument behavior with measurements the reward cannot influence, treat any divergence between return and behavior as a specification defect rather than a training deficiency, trace each defect to a single term, and re-verify against the same independent evidence.

Notes

[1] V. Krakovna et al., "Specification gaming: the flip side of AI ingenuity," DeepMind Blog, April 21, 2020.

[2] J. Skalse, N. H. R. Howe, D. Krasheninnikov, and D. Krueger, "Defining and Characterizing Reward Hacking," NeurIPS 35, 2022, arXiv:2209.13085.

[3] A. Pan, K. Bhatia, and J. Steinhardt, "The Effects of Reward Misspecification: Mapping and Mitigating Misaligned Models," ICLR 2022, arXiv:2201.03544.

[4] A. Y. Ng, D. Harada, and S. J. Russell, "Policy Invariance Under Reward Transformations," ICML 1999, 278–287.

[5] J. Schulman et al., "Proximal Policy Optimization Algorithms," 2017, arXiv:1707.06347.

[6] A. Raffin et al., "Stable-Baselines3: Reliable Reinforcement Learning Implementations," JMLR 22, no. 268 (2021): 1–8.

This brief generalizes the scenario, terminology, and organizational details of the original project for public presentation. Simulation results shown are the author's own. The views expressed are the author's own and do not reflect the official position of the United States Military Academy, Department of the Army, or Department of Defense.