Aegis: Proactive Decision & Counterfactual Coach for Dota 2

Purely-structured, ProAct-style offline planner (Phase 1 results)

Hengxu Li

February 16, 2026

Executive summary

- **Goal:** turn OpenDota match logs into a proactive planner that predicts the near future and recommends actionable decisions (items / wards / fights) with offline evaluation.
- **We built an end-to-end pipeline** (30k parsed matches, patches 7.38–7.40) and a proposal + value decision stack:
  - **Value model** $V_\theta(s_t)$: predicts win prob + 3-min delta advantages.
  - **Policy (proposal)** $\pi_\phi(a \mid s_t)$: next-item distribution (80 classes).
  - **Generate-and-Score**: choose best candidate by value under feasibility filters.
- **Key results (patch 7.40 test, zero-shot):**
  - Value v1 vs v0: AUC **0.7827** (vs 0.7699), $\mathrm{RMSE}_{\Delta gold@3m}$ **2280** (vs 2754), $\mathrm{RMSE}_{\Delta xp@3m}$ **4267** (vs 5011).
  - Next-item policy: Top-1 **0.219**, Top-5 **0.552**, Top-20 **0.889** (80 classes).
  - Generate-and-score (feasible/time-band): avg uplift vs base **+0.0096**; true-feasible hit@1 **0.125** (n=1799).

All results are offline proxies (value-based); we focus on a reproducible benchmark + strong baselines.

Problem & why now

Pain: Draft is solved-ish; in-game decision support is harder and more "agent-like".
Constraint: we start purely structured (no vision), but want a world-model narrative.

What we optimize for:

- Proactive: predict before events happen (lead time).
- Counterfactual: "If we buy X / ward Y now, what changes?" (value uplift).
- Patch drift: evaluate train on old, test on new patch; few-shot curve later.

Actions we can learn from logs (Phase 1):

- BUY(item) from purchase logs
- PLACE_OBS / PLACE_SEN (no location yet in Phase 1)
- Skirmish/fight context from kills + teamfight windows

Next up (Phase 2): ward-*where* policy using (x,y) + left_log survival.

Data & representation

Dataset: 29,957 matches (filtered), patches 57/58/59 (7.38/7.39/7.40).

Train: 7.38 + early 7.39; Val: late 7.39; Test: 7.40 (zero-shot).

ETL outputs (Parquet):

- `matches`: match_id, patch, start_time, duration, radiant_win, region, ...
- `events`: unified timeline (purchase / wards / kills / runes / objectives / teamfight_start/end)
- `state_player_minute`: (match, t, player) gold/xp/lh/dn
- `state_team_minute`: (match, t) radiant_gold_adv, radiant_xp_adv
- Derived: `team_items_minute` (has_item), `team_wards_minute`, `team_skirmish_minute`

State (team-level) s_t:

- Economy/XP aggregates + diffs (radiant vs dire): sum/mean/max, adv curves
- Power-spike indicators: `has_item` for vocab items (80)
- Vision features: active obs/sen + place/left counts (1m, 5m)
- Skirmish features: kills/deaths rolling window; time since last fight

Models: Value + Policy + Generate-and-Score

1) Value model (radiant-centric):

$$V_\theta(s_t) = \big(\ \hat{p}(\text{RadiantWin} \mid s_t),\ \widehat{\Delta G}_{t\to t+H},\ \widehat{\Delta X}_{t\to t+H}\ \big), \quad H = 180s$$

- Implementation: XGBoost (3 heads: classifier + 2 regressors)
- Targets from advantage curves: $\Delta G = G_{t+H} - G_t$; $\Delta X = X_{t+H} - X_t$

2) Next-item policy (proposal model):

$$\pi_\phi(a \mid s_t, \text{team}) \ \text{ over } \ a \in \{1, \ldots, 80\}$$

- Supervised from first purchase time per team/item; decision time = buy time minus lead (bucketed).

3) Planner (Generate-and-Score):

$$a^\star = \arg\max_{a \in \mathcal{C}(s_t)} \ \text{TeamWinProb}\big(\hat{p}(\text{RadWin} \mid \text{Apply}(s_t, a))\big)$$

- $\mathcal{C}(s_t)$ from top-K policy candidates + feasibility filters (cost, time-band).

Evaluation protocol (patch drift + offline metrics)

Splits (deploy-style):

- Train: 7.38 + early 7.39 Val: late 7.39 Test: 7.40 (zero-shot)
- (Optional) few-shot curve: calibrate on first N matches of 7.40, evaluate on the rest

Metrics:

- Value: AUC / LogLoss for win; RMSE/MAE for $\Delta gold$, Δxp (H=3min)
- Policy: Top-1/Top-5/Top-20 accuracy (80-way classification)
- Planner:
  - **policy_hit@K**: is true action in top-K proposal set?
  - **gs_hit@1 (true feasible)**: does value-ranking pick the true action when true is in feasible set?
  - **uplift (proxy)**: $\hat{\rho}_{\text{best}} - \hat{\rho}_{\text{base}}$ (base = HOLD / current state)

Proxy uplift is not causal; it is a reproducible offline improvement signal for iteration.

Phase 1 results (patch 7.40 test)

Value model (v0 vs v1):

Model	AUC	LogLoss	$\text{RMSE}_{\Delta G}$	$\text{MAE}_{\Delta G}$	$\text{RMSE}_{\Delta X}$	$\text{MAE}_{\Delta X}$
Value v0	0.7699	0.5633	2753.9	1881.5	5011.4	3172.6
Value v1	**0.7827**	**0.5508**	**2280.1**	**1573.0**	**4267.0**	**2713.1**

Next-item policy (80 classes): Top1 0.219 Top5 0.552 Top20 0.889

Generate-and-Score (TEST):

- Magic toggle: uplift vs base **+0.0183**; gs_hit@1_true_ok 0.0464 (n=5000)
- Feasible/time-band: uplift vs base **+0.0097**; gs_hit@1_true_ok **0.1226** (n=1998)

Feasible constraints reduce uplift magnitude (expected) but improve agreement with real decisions.

Demo: actionable recommendations (MVP)

Given (match_id, t, team) $\Rightarrow$ (top actions with Δwin, Δadv)

Example (t=900, Radiant): base win 0.554

- BUY(relic) Δwin $+0.048$ BUY(yasha) Δwin $+0.004$ ...
- PLACE_SEN() becomes competitive in mid-game/behind states (vision as a proactive action)

System interface (Phase 1):

- `score_actions_v2`: enumerate feasible actions and score with value
- `recommend_next_item`: policy proposals $\rightarrow$ value ranking (agent-like step)

Phase 2 will add ward-*where* (x,y grid) and short-horizon plan tokens.

Roadmap (what we pitch)

Phase 2 (next): Ward-where + proactive planning

- Learn PLACE_OBS(x,y) / PLACE_SEN(x,y) from (x,y,time) logs
- Add ward survival/value targets using left_log (duration, deward probability)
- Extend planner: multi-step plan tokens and lead-time evaluation (ProAct-style)

Why it is "AI" (not analytics only):

- Offline RL / planning framing with explicit proposal + value scoring
- Patch shift as environment drift; evaluate zero-shot + few-shot adaptation
- Reproducible benchmark from real competitive game logs (structured world)

Ask / next milestone:

- Deliver ward-where policy + generate-and-score uplift on vision decisions
- Package into a lightweight coach UI (timeline + recommendations)