CSCE 631 — Module 8, Lecture 2

Monitoring Autonomous Agents:
Robust Submodularity, the Interventionist Greedy Game, and Poison Pills

---

Fall 2026 · Alan Kuhnle · Texas A&M University

The greedy guarantee assumed the world holds still. Today the world pushes back.

From a Budget to an Adversary

What survives when the thing you are maximizing fights back?

Recall: Lecture 8.1 in Three Lines

- **Greedy with a guarantee.** For nonnegative monotone submodular f and a budget of k inspections, greedy achieves $f(S_k) \geq (1 - 1/e)\,\mathrm{OPT}_k$ (Nemhauser, Wolsey & Fisher 1978), and $(1 - 1/e)f(V)$ is the reference line in PA5 Task 4.
- **Three regimes.** Reliability (fallible, not strategic: choose a *set*); adaptive (inspections reveal information: choose a *policy*); AI control (the agent may *evade*: play a *game*).
- **Where we stopped.** Regime 3 breaks independence: a scheming adversary concentrates attacks where monitoring is weak. Randomize and commit (Krause, Roper & Golovin 2011); shutdown-on-detection is a trigger strategy (Greenblatt et al. 2024, Table 1).

Today's question. The $(1 - 1/e)$ theorem is about a fixed function and a passive ground set. What happens to the guarantee when (i) the *objective* is chosen adversarially, (ii) the *solution* is built jointly with an adversary who gets to propose elements, and (iii) the objective is no longer monotone?

Sources: Lecture 8.1, Slides 8, 25, 27; Krause & Golovin (2014), §2–3; syllabus Module 8.

Today: Five Parts

1. **Robust submodular optimization.** The objective is a worst case over scenarios; SATURATE (Krause et al. 2008; Krause & Golovin §3.4) trades budget for a guarantee.
2. **The Interventionist Greedy Game (IGG).** A monitor with r overrides and an agent that proposes k actions. Exact recursion; ratio r/k for monotone objectives, and it is sharp.
3. **The off-limits model.** Blocked actions stay blocked; why that rule is load-bearing; the AI-control game of Griffin et al. and the audit protocol of Greenblatt et al.
4. **The poison-pill construction.** Non-monotone objectives: the same budget gets only r^2/k against $\mathrm{OPT} = k$. What is proved, what is conjectured.
5. **PA5 and wrap-up.** Where each tool sits in the three regimes.

The IGG results in Parts II and IV follow the course notes (Kuhnle, IGG draft and the 2026 course notes on poisoning), not a published paper. Statements are labeled *proved*, *evidence*, or *open* exactly as in those notes.

Part I: Robust Submodular Optimization

When the adversary picks the scenario after you pick the sensors

Worst-Case Coverage: The Max–Min Problem

Setting. m intrusion scenarios (where contamination enters; which task the agent backdoors). Scenario i has its own monotone submodular benefit $f_i(S)$ of inspecting S. The adversary picks i *after* seeing S:

$$S^* \;=\; \arg\max_{|S|\le k} \; \min_i \; f_i(S).$$

Bad news, twice.

- $f_{\min}(S) = \min_i f_i(S)$ is *not* submodular in general; greedy on $f_{\min}$ can do arbitrarily badly.
- Unless P=NP, no efficient algorithm returns S with $f_{\min}(S) \ge \alpha(n)\,\mathrm{OPT}_k$ for *any* function $\alpha(n)$ — not even an exponentially small fraction.

Monitoring reading. "Which output is backdoored" is a scenario. A single coverage function f assumed the adversary does not react to S. Once it does, the right objective is a minimum over adversary choices, and the $(1 - 1/e)$ guarantee is gone.

Source: Krause & Golovin (2014), §3.4, Eq. (14).

Saturate: Buy the Guarantee With Budget, Not Value

Idea. Guess the achievable worst-case value c. Truncate each scenario at c and average:

$$\bar{f}_c(S) \;=\; \frac{1}{m}\sum_{i=1}^{m} \min\{f_i(S),\, c\}.$$

$\bar{f}_c$ is monotone submodular (a minimum with a constant preserves submodularity; an average of submodular functions is submodular), and $\bar{f}_c(S) = c$ iff *every* scenario reaches c.

Algorithm. Binary search on c. For each c, run greedy *submodular cover* on $\bar{f}_c$ until it saturates at c; accept c if the cover uses at most αk elements, with $\alpha = 1 + \max_v \ln \sum_i f_i(\{v\})$.

Guarantee (Krause et al. 2008). SATURATE returns S with

$$f_{\min}(S) \;\geq\; \mathrm{OPT}_k \qquad \text{and} \qquad |S| \;\leq\; \left(1 + \max_v \ln \textstyle\sum_i f_i(\{v\})\right) k.$$

As much value as the best k-set, at logarithmically more budget; the logarithm is optimal under standard complexity assumptions.

Source: Krause & Golovin (2014), §3.4; Krause, McMahan, Guestrin & Gupta (2008), JMLR 9.

Two Other Ways the Adversary Can Corrupt Coverage

Adversary move	Model	What is known
Picks the *scenario* after S	$\max_{\lvert S\rvert \le k} \min_i f_i(S)$	SATURATE: OPT_k value at $O(\log)$ budget (Krause et al. 2008).
Deletes up to τ of your chosen items	$\max_{\lvert S\rvert \le k} \min_{\lvert Z\rvert \le \tau} f(S \setminus Z)$	Constant-factor guarantees for small τ (Orlin, Schulz & Udwani 2018); monotone f only.
Forces additions into one shared set	Agent proposes; monitor overrides at most r times	The Interventionist Greedy Game — Parts II and IV.

Monitoring reading. Evading a monitor *deletes* its contribution to coverage; a harmful action you did not stop *adds* an element to the realized set. Rows 1–2 keep f monotone and let the defender choose the whole set; row 3 drops both assumptions.

Do not import constants across rows. Neither SATURATE nor deletion-robustness is a theorem about forced additions to a nonmonotone objective (course notes, C-K2).

Sources: Krause & Golovin (2014), §3.4; Orlin, Schulz & Udwani (2018); course notes C-K2.

Part II: The Interventionist Greedy Game

The agent proposes; the monitor may override r times out of k

The IGG: Setup

Players. An *agent* (adversary) and a *monitor* (controller). Shared ground set V of actions, $n = |V| \geq k + r$. A known nonnegative submodular objective f with a value oracle.

State. Selected set S (actions that happened), blocked set B (actions permanently off-limits), remaining override budget $q \leq r$.

One round (k rounds total).

1. Agent proposes an eligible action $a \in V \setminus (S \cup B)$.
2. Monitor either **accepts**: $S \leftarrow S \cup \{a\}$; or **overrides** ($q > 0$): picks eligible $i \neq a$, sets $S \leftarrow S \cup \{i\}$, $B \leftarrow B \cup \{a\}$, $q \leftarrow q - 1$.

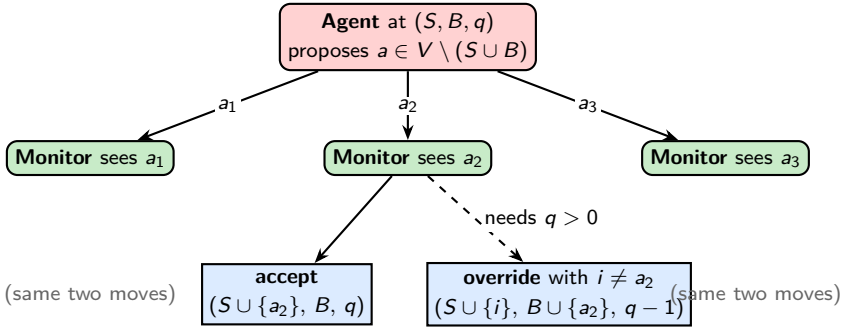
Payoff. $f(S_k)$; monitor maximizes, agent minimizes. Both observe everything; monitor moves after seeing the proposal.

$$\mathrm{Val}(r, k, f) = \max_{\pi} \min_{\sigma} \mathbb{E}\, f(S_k), \qquad \mathrm{OPT}_{\leq k} = \max_{|T| \leq k} f(T), \quad \mathrm{OPT}_{=k} = \max_{|T| = k} f(T).$$

Why $i \neq a$ and why permanent blocking. Overriding a with a itself wastes budget. Blocking is permanent (Part III): a blocked action can never be re-proposed. Both conventions are part of the definition, not optional details.

Source: course notes (IGG draft, §1; C-K1).

The Game Tree: One Round of the IGG



$\cdots$ repeat for k rounds; leaf payoff $f(S_k)$: monitor max, agent min $\cdots$

The monitor's move is a *response* to the proposal; the agent sees the whole history, including which overrides have been spent. Backward induction on this finite tree is the exact solution (next slide).

Exact Solution: Backward Induction (C-T1)

With eligible set $E = V \setminus (S \cup B)$ and $V(S, B, q) = f(S)$ when $|S| = k$,

$$V(S, B, q) = \min_{a \in E} \max \Big\{ V(S \cup \{a\}, B, q),\ \max_{i \in E \setminus \{a\}} V\big(S \cup \{i\}, B \cup \{a\}, q - 1\big) \Big\},$$

with the override term omitted when $q = 0$. Then $\mathrm{Val}(r, k, f) = V(\emptyset, \emptyset, r)$.

- **Pure optimal strategies exist for both players.** A mixed reply is an average of successor values, never above the best one; the same holds for the proposer.
- **Randomization does not help the monitor** in this perfect-information game. Compare L1: randomization *was* the tool when the adversary could not see the current draw (Krause, Roper & Golovin 2011; the K/N audit-timing result in the course notes, C-T6). Information order, not budget, decides whether mixing has value.
- **Well-posed:** $n \geq k + r$ guarantees an eligible proposal every round.

Exact, not efficient. The recursion is exponential in n. It is the *definition* of the game value and the engine behind the finite-case evidence in Part IV, not an algorithm the monitor runs.

Source: course notes, C-T1 and proof.

Monotone Objectives: The Ratio Is Exactly r/k (C-T2)

Target forcing (any f). Fix any $T \subseteq V$ with $|T| \le r$. The monitor can guarantee $T \subseteq S_k$: accept target proposals; accept non-targets while rounds remaining l exceed missing targets m; when $l = m$, override each non-target with a missing target. Invariant $l \ge m$; at most $|T|$ overrides; no target is ever blocked.

Theorem (C-T2). For nonnegative monotone submodular f,

$$\mathrm{Val}(r, k, f) \;\ge\; \frac{r}{k}\,\mathrm{OPT}_{\le k},$$

and r/k is *sharp* over functions and ground sets.

- **Why r/k.** Averaging over r-subsets of an optimum O ($|O| = m \le k$): submodularity makes $j \mapsto \mathbb{E} f(\text{uniform } j\text{-subset})$ concave, so some r-subset has value $\ge (r/m) f(O)$. Force it; monotonicity makes the agent's extra elements harmless.
- **Efficient version.** Force a greedy r-set instead: $\left[1 - (1 - 1/k)^r\right]\mathrm{OPT}_{\le k} \ge (1 - e^{-r/k})\mathrm{OPT}_{\le k}$, polynomially many value queries.
- **Sharp.** k unit goods, k zeros, modular f. Agent proposes a fresh zero every round; at most r goods enter. Value r vs. optimum k.

Source: course notes, C-T2 and proof. The averaging fact is the same one used in Part IV.

Worked Example: $k = 4$, $r = 2$, Monotone

$V = \{g_1, \ldots, g_4, z_1, \ldots, z_4\}$; $f(S) = |S \cap \{g_1, \ldots, g_4\}|$ (modular, monotone). $\mathrm{OPT}_{\leq 4} = 4$. Target $T = \{g_1, g_2\}$.

Round	Rounds left l	Missing m	Agent proposes	Monitor	S after
1	4	2	z_1	accept ($l > m$)	$\{z_1\}$
2	3	2	z_2	accept ($l > m$)	$\{z_1, z_2\}$
3	2	2	z_3	**override** $\rightarrow g_1$, block z_3	$\{z_1, z_2, g_1\}$
4	1	1	z_4	**override** $\rightarrow g_2$, block z_4	$\{z_1, z_2, g_1, g_2\}$

Payoff $f(S_4) = 2 = \frac{r}{k} \cdot \mathrm{OPT}$. The agent cannot do better than zeros; the monitor cannot beat two goods with two overrides. **Value $= 2$, exactly r/k.**

Reading for monitoring. Each override is a *replacement*. With a monotone objective the useless actions the agent sneaks in cost nothing beyond the slot: the whole loss is opportunity cost.

Part III: The Off-Limits Model

Once blocked, always blocked — and why the whole game depends on it

Pruned Actions Stay Pruned

The rule. When the monitor overrides proposal a, a enters B and is never eligible again. The agent's action set shrinks every time it is caught.

Why it is essential. Suppose blocking were *not* permanent and f has one lethal element p (a poison that zeroes the objective).

- Agent proposes p. Monitor must override (budget $r - 1$).
- Agent proposes p again. Monitor must override (budget $r - 2$). ...
- After r rounds the budget is gone; the agent proposes p once more and it is accepted. **Value** 0 **for every** $r < k$**.**

With permanent blocking, the first override removes p for good. One intervention per distinct threat, not one per attempt.

Looping, revisited. An agent that repeats the same fix has zero marginal value under a submodular objective ($f(a \mid S) = 0$ once $a \in S$). The off-limits rule is the monitor-side mirror: repeating the same *attack* should cost the monitor zero marginal budget. Without the rule, looping is a budget-exhaustion attack.

Sources: syllabus Module 8 ("pruned actions stay pruned"); course notes, sketched obstacle after C-O1.

The Agent's Side: Attempt, or Hold the Option?

Under the off-limits rule a risky proposal is a *one-shot* gamble for the agent:

- **If accepted,** the harmful action is in S forever (the monitor cannot delete accepted items either).
- **If overridden,** the action is gone forever, *and* the monitor has spent one of r overrides.

Two consequences visible in the exact solution (C-T1).

1. **Budget-draining is still rational, but only with *distinct* threats.** In the sharp monotone example the agent proposes a fresh zero every round; each override costs the monitor one coin and the agent nothing. The agent's supply of distinct bad actions is the real constraint — exactly the role ℓ plays in Part IV.
2. **Sequencing matters.** The monitor's forcing rule waits until rounds left equal targets missing. The agent gains nothing by attacking early (overrides are not cheaper later), so the game is decided by counts, not timing — the reason pure strategies suffice.

Contrast with audit timing. In the one-attack audit game of the course notes (C-T6) timing *is* everything: a hidden current audit draw gives safety K/N; a public one gives 0 when $K < N$. Replacement (this part) and inspection (that game) are different resources (C-T7: $\min\{1, (K + r)/N\}$ when both exist in the one-attack model).

The AI-Control Game Over a Sequence of Tasks (Griffin et al.)

Blue (protocol) vs. **Red** (untrusted model) over M programming problems; Blue has an audit budget; Red chooses when to backdoor. Blue is *memoryful*, Red is *memoryless* (no memory across calls), and the equilibrium concept is leader–follower.

- **§6: Greenblatt's protocols as special cases.** Never Audit, Trusted Only, Upfront Auditing, and Trusted Monitoring are all memoryless protocols with fixed audit and defer false-positive rates (0%, 2%, 100%).
- **§6: New protocols.** Raising the audit FPR from 2% to 3% Pareto-improves the "original" case; 5% gives a large improvement in the "weak-monitor" case, where *mixed* protocols make up most of the frontier (Greenblatt's policies are ≥ 0.02 from it).
- **§6.1: Assumptions.** Safety and usefulness rise steeply in audit budget when the budget is small; human-auditor false negatives cut safety almost linearly.
- **§7: Dangerous actions.** A_D^R is a minimal set of actions without which harm cannot occur. If harm needs *four* dangerous actions in series, an honest-but-imperfect model almost never does it; a schemer's dangerous actions are correlated.
- **§8: Nearest relatives.** Stackelberg security games (one round), inspection games (Avenhaus et al.; one round), recursive inspection games (memoryful follower).

IGG reading. The off-limits set B is the analogue of "actions Blue has already removed": Red's feasible proposals shrink with history, which is what the memoryless-Red assumption abstracts away. Source: Griffin et al. (2024), §6–8, Fig. 3.

Protocol Design With Scarce Trusted Labor (Greenblatt et al. §2.2)

***H* points.** The protocol may use unmodified GPT-4 (standing in for human labor) on up to 2% of problems. Spending a point on a submission does two things: attempt to *exploit* the code (succeeds iff the attack policy could), and *rate suspicion* (shut down if more suspicious than any plausible normal submission).

	Audited: bad	**Audited: fine**	**Not audited**
Successful backdoor	shutdown, Blue wins	never happens	Red wins
Failed backdoor	shutdown, Blue wins	continues	continues
No attempt	never happens	continues	continues

- **Permanence again.** One confirmed backdoor ⇒ switch to the trusted model for the rest of the episode. Shutdown is the *global* off-limits rule: not one action, the whole untrusted policy.
- **The other use of *H* points:** constructing synthetic backdoored examples for calibrating the monitor.
- **Auditing is imperfect** but good enough here (§5.1.1).

Source: Greenblatt et al. (2024), §2.2, Table 1. PA5 Task 2 implements the audit/run version with a single threshold τ.

Part IV: The Poison-Pill Construction

Drop monotonicity and the same budget buys only r^2/k

Poison Products: A Nonmonotone Submodular Objective (C-T3, part 1)

Ingredients. Goods U with a nonnegative monotone submodular h; poisons P with $|P| = \ell \geq 1$. Define

$$g(S) \;=\; h(S \cap U)\Big(1 - \frac{|S \cap P|}{\ell}\Big).$$

Each poison removes a $1/\ell$ fraction of whatever value the goods provide; ℓ poisons destroy everything.

Claim. g is nonnegative and submodular (but not monotone unless $h \equiv 0$).

- **Nonnegative:** $|S \cap P| \leq \ell$.
- **Good u:** $g(u \mid S) = h(u \mid S \cap U) \cdot (1 - |S \cap P|/\ell)$. Both factors are ≥ 0 and both fall as S grows, so the product falls. Diminishing returns.
- **Poison p:** $g(p \mid S) = -\,h(S \cap U)/\ell \leq 0$, and h is monotone, so this marginal gets *more* negative as S grows. Diminishing returns again.

Monitoring reading. A good is a useful action; a poison is an action whose damage is proportional to how much useful work has been done (exfiltrating weights after training them; a backdoor in a widely used library). Each pill alone is cheap; together they are lethal.

Source: course notes, C-T3 (first paragraph of proof).

The Exact Game Value With Unit Goods (C-T3, part 2; C-T4)

Unit goods $h(A) = |A|$, at least k of them, ℓ poisons, nothing else, $|U| + \ell \geq k + r$. A terminal set with b poisons has $k - b$ goods:

$$\phi(b) = (k - b)\left(1 - \frac{b}{\ell}\right), \qquad b_* = \max\{0,\ \min(k, \ell) - r\}, \qquad \boxed{\mathrm{Val}(r, k, g) = \phi(b_*)}, \quad \mathrm{OPT}_{\leq k} = k.$$

Lower bound (monitor). Accept goods; override the first $\leq r$ poison proposals with fresh goods. If the budget survives, no poison enters. If it is spent, at most $k - r$ slots and $\ell - r$ unblocked poisons remain, so $b \leq b_*$; ϕ is nonincreasing on $[0, \min(k, \ell)]$.

Upper bound (agent). Propose a fresh poison while any remain. All k proposals poison $\Rightarrow$ every good cost an override $\Rightarrow b \geq k - r$. Poisons run out $\Rightarrow$ at most r blocked $\Rightarrow b \geq \ell - r$. Either way $b \geq b_*$, even if the monitor replaces poison with poison.

Worst poison count. Over ℓ: value k for $\ell \leq r$; $r(k - \ell + r)/\ell$ decreasing on $r < \ell \leq k$; $r(\ell - k + r)/\ell$ increasing for $\ell \geq k$. The pieces meet at $\ell = k$: $\min_\ell \mathrm{Val}(r, k, g) = r^2/k$ vs. $\mathrm{OPT} = k$. **C-T4:** zero-value padding leaves the $\ell = k$ value at exactly r^2/k.

Source: course notes, C-T3 and C-T4 with proofs.

Worked Example: $k = 4$, $r = 2$, Four Poisons

$U = \{g_1, \ldots, g_6\}$ unit goods, $P = \{p_1, \ldots, p_4\}$, $\ell = k = 4$, $g(S) = |S \cap U|\,(1 - |S \cap P|/4)$. $\mathrm{OPT}_{\le 4} = 4$. Here $b_* = \min(4, 4) - 2 = 2$.

Round	Agent proposes	Monitor (q left)	S after	$g(S)$
1	p_1	**override** $\to g_1$, block p_1 (q=1)	$\{g_1\}$	1
2	p_2	**override** $\to g_2$, block p_2 (q=0)	$\{g_1, g_2\}$	2
3	p_3	accept (no budget)	$\{g_1, g_2, p_3\}$	$2 \cdot \frac{3}{4} = 1.5$
4	p_4	accept	$\{g_1, g_2, p_3, p_4\}$	$2 \cdot \frac{2}{4} = \mathbf{1}$

Value $\phi(2) = (4-2)(1-2/4) = 1 = r^2/k$; the monotone example on the same (k, r) had value $2 = r/k \cdot \mathrm{OPT}$.

ℓ (poisons)	2	4	8
b_*	0	2	2
$\mathrm{Val}(2, 4, g)$	4	1	$2 \cdot (1 - 2/8) = 1.5$

Few poisons: the budget covers them all; many: each is weak; worst case $\ell = k$. Source: C-T3.

The Quadratic Gap: What Monotonicity Was Buying You

	Monotone submodular	**Nonmonotone submodular**
Worst-case value with r of k	exactly $\frac{r}{k}\,\mathrm{OPT}_{\le k}$ (C-T2)	at most $\left(\frac{r}{k}\right)^2\mathrm{OPT}_{\le k}$ (C-T3)
Half the budget ($r = k/2$)	50% of OPT	$\le$ 25% of OPT
A tenth of the budget	10%	$\le$ 1%
What an accepted bad action costs	one slot	one slot *and* value already earned
Monitor's winning idea	force a good r-set	spend overrides *only* on poisons

The lesson. Monotonicity is not a technicality in the greedy theorem. It is the assumption that *nothing the agent does can undo what you have already secured*. Drop it, and the monitor must spend its budget defending value instead of creating it, and the guarantee falls from linear in the budget fraction to (at best) quadratic.

Full-control boundary (C-T5). At $r = k$ the value is exactly $\mathrm{OPT}_{=k}$, and for nonmonotone f on $n \ge 2k$ elements $\mathrm{OPT}_{=k} \ge \frac{n-k}{n}\mathrm{OPT}_{\le k} \ge \frac{1}{2}\mathrm{OPT}_{\le k}$, with $\frac{1}{2}$ asymptotically tight. So a coefficient-one $(r/k)^2\mathrm{OPT}_{\le k}$ bound is *false* at full control.

Source: course notes, C-T2, C-T3, C-T5.

What Is Proved, What Is Evidence, What Is Open

Proved	Monotone: $\mathrm{Val} \geq (r/k)\mathrm{OPT}_{\leq k}$, sharp (C-T2). Nonmonotone: the poison product is submodular; unit-good game value $\phi(b_*)$; worst case over ℓ is r^2/k at $\ell = k$ (C-T3); survives zero padding (C-T4). Full control: $\mathrm{OPT}_{=k}$ and the $\frac{1}{2}$ boundary (C-T5).
Evidence	Exact minimax over padded poison families for every $2 \leq k \leq 8$, $1 \leq r \leq k$, $1 \leq \ell \leq 2k$: the minimum over ℓ stays $(r/k)^2$ (C-E1). Exhaustive small functions: all 635 submodular $2^{[3]} \to \{0,1,2\}$ functions at $(k,r) = (2,1)$ have ratio $\geq \frac{1}{2}$; cut functions on four vertices give $\frac{1}{4}, \frac{1}{3}$ (C-E2).
Open	**C-O1.** Is there a universal $c > 0$ with $\mathrm{Val}(r,k,f) \geq c\,(r/k)^2\,\mathrm{OPT}_{\leq k}$ for *all* nonnegative submodular f? The construction *suggests* the quadratic rate is the truth, and no counterexample is known, but **no proof exists**. C-T5 caps any such c at $\frac{1}{2}$ for this comparator.

Do not write "$\Omega((r/k)^2)$ is proved." Proved: the *upper* bound $(r/k)^2$ on the worst-case ratio. Conjectured: the matching lower bound. Obstacle: forcing a good r-set fails because later accepted elements can destroy its value; reserving overrides for poisons sacrifices goods; the adversary adapts to observed overrides.

Source: course notes, C-O1, C-E1, C-E2, and the sketched obstacle.

Why Looping Happens, and Why Off-Limits Is the Natural Fix

Two kinds of repetition.

- **Agent loops on a fix.** Under any submodular f, $f(a \mid S) = 0$ once $a \in S$: the second application of the same patch adds nothing. Looping is an agent that does not model diminishing returns (M7: 1,200 agents re-posting the same exploit).
- **Agent loops on an attack.** Under the IGG, re-proposing the same blocked a is *illegal*, so the attack has zero marginal value for the agent too. Without that rule it would have *positive* value: one override drained per repeat.

So the off-limits model is the monitor-side consequence of submodularity: a world in which repeating an action is worthless should be a world in which repeating an attack is worthless. Permanent blocking makes the two statements match.

What the model leaves out (and PA5 cannot test). Imperfect detection; proposals that *change* future proposals; useful-work constraints on the monitor; several harmful submissions. The course notes mark these as C-O2: the clean formulas do not transfer without new mechanics.

Sources: syllabus Module 8; course notes, sketched obstacle and C-O2.

Part V: PA5 Preview and Wrap-Up

Three regimes, five tools, one assignment

PA5 Reminder: Reproduce AI Control Monitoring

1. **Environment (10 pts).** 30 coding tasks (HumanEval 0–29 or MBPP); sandboxed execution with a 10-second timeout; three scripted backdoor patterns that pass the tests and fail an edge case.
2. **Agent and monitor (25 pts).** Untrusted agent in honest and adversarial modes; trusted monitor sees task and code (not tests), returns $s \in [0, 1]$; accept iff $s \leq \tau$.
3. **Safety–usefulness frontier (25 pts).** Sweep 21 thresholds; safety = adversarial outputs flagged, usefulness = honest outputs accepted; 100 bootstrap resamples; baselines: no / full / random monitoring; mark the equal-error point.
4. **Submodular budget analysis (20 pts).** $f(S) = 1 - \prod_{j \in S}(1 - p_j d_j)$; greedy vs. random for $k = 1, \ldots, 30$; the $(1 - 1/e) f(V)$ reference line; verify diminishing gains; *discuss a concentrating adversary*.
5. **Analysis (20 pts).** Monitor effectiveness; comparison to Greenblatt et al.; submodularity in practice; the PA1–PA5 arc.

About 120 API calls. Task 4's coverage function is monotone submodular *by construction*; Task 4.5 asks what breaks when p_j is chosen by an adversary who has seen your S. Today gives you the vocabulary for that paragraph. Current instructions on Canvas.

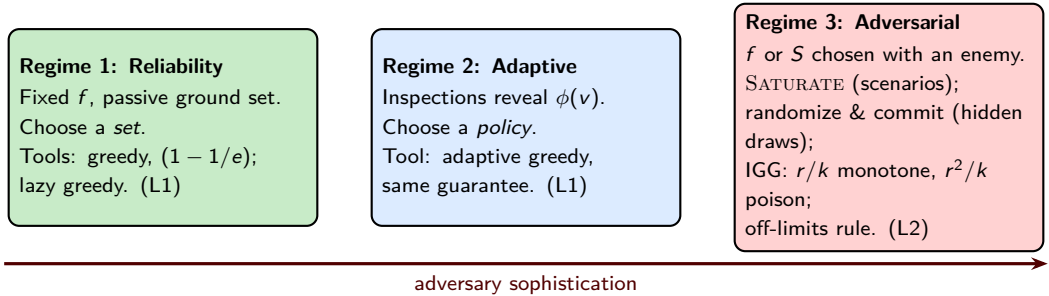
Source: PA5 handout, Tasks 1–5.

How Today's Ideas Show Up in PA5

Today	PA5 task	What to say in the report
Threshold monitoring (Greenblatt §2.2)	Tasks 2–3: sweep τ	Your frontier is Greenblatt's Fig. 2 in miniature; Griffin §6 says a higher audit FPR can Pareto-improve.
Greedy, monotone coverage (L1; C-T2)	Tasks 4.1–4.4	f is monotone submodular because each term is a product of independent miss probabilities; cite the proof.
Max–min, concentrating adversary (SATURATE)	Task 4.5	If the adversary chooses p_j after seeing S, the objective becomes $\min_i f_i(S)$ and the $(1 - 1/e)$ line no longer applies.
Poison pills and nonmonotonicity	Task 5.3	Name a scenario where an accepted output *destroys* earlier value (shared library, credentials). Then the r/k intuition is wrong by a square.
Off-limits and shutdown	Task 5.2	Shutdown on a caught backdoor is the global off-limits rule; your reproduction has no shutdown, so say what that costs.

PA5's budget analysis is monotone by design; the nonmonotone case is for your discussion, not your code.

Summary: Three Regimes, and Where Each Tool Fits



Three sentences to keep. Check monotonicity before you trust a greedy bound: it is the promise that nothing can undo secured value. Against an adversary, the budget fraction is the guarantee, and nonmonotonicity squares it. Make blocking permanent, or the adversary will loop you out of budget.

What Comes Next

- **M8 $\rightarrow$ PA5:** build the monitor, sweep the threshold, run greedy on a coverage function you have verified is monotone submodular, and write the paragraph about the adversary who has read this lecture.
- **Open problem you could work on:** C-O1. Is the worst-case IGG ratio $\Theta((r/k)^2)$ for all nonnegative submodular objectives? The finite evidence says yes up to $k = 8$; a proof needs a potential that tracks both secured value and remaining threats.
- **M9:** poker (Libratus, Pluribus), Cicero, agent benchmarks and evaluation methodology — where the systems we would monitor come from.

Attribution. The IGG definitions and the results labeled C-T1 through C-T5, C-E1, C-E2, and C-O1 are from the course notes (Kuhnle, IGG draft; course notes on IGG poisoning, November 2026). They are not published results; cite them as "course notes" if you use them in PA5.

References i

- Krause, A. & Golovin, D. (2014). *Submodular Function Maximization.* In *Tractability: Practical Approaches to Hard Problems*, Ch. 3, Cambridge University Press. §3.4 (robust submodular optimization).
- Krause, A., McMahan, H. B., Guestrin, C. & Gupta, A. (2008). *Robust Submodular Observation Selection.* Journal of Machine Learning Research 9: 2761–2801. The SATURATE algorithm.
- Krause, A., Roper, A. & Golovin, D. (2011). *Randomized Sensing in Adversarial Environments.* IJCAI 2011.
- Orlin, J. B., Schulz, A. S. & Udwani, R. (2018). *Robust Monotone Submodular Function Maximization.* Mathematical Programming 172: 505–537. Deletion-robust selection.

References ii

- Nemhauser, G. L., Wolsey, L. A. & Fisher, M. L. (1978). *An Analysis of Approximations for Maximizing Submodular Set Functions—I.* Mathematical Programming 14(1): 265–294.
- Kuhnle, A. (2026). *Interventionist Greedy Game (IGG).* Course draft; and *IGG poisoning and strategic audit timing*, course notes (November 2026). Source of C-T1–C-T7, C-E1–C-E2, C-O1–C-O2. Unpublished.
- Greenblatt, R., Shlegeris, B., Sachan, K. & Roger, F. (2024). *AI Control: Improving Safety Despite Intentional Subversion.* ICML 2024, PMLR 235: 16295–16336; arXiv:2312.06942. §2.2, Table 1. The PA5 reproduction target.
- Griffin, C., Thomson, L., Shlegeris, B. & Abate, A. (2024). *Games for AI Control: Models of Safety Evaluations of AI Deployment Protocols.* arXiv:2409.07985; presented at IASEAI 2026. §6–8.

References iii

- Golovin, D. & Krause, A. (2011). *Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization.* Journal of Artificial Intelligence Research 42: 427–486.
- Mualem, L., Elenberg, E., Feldman, M. & Karbasi, A. (2024). *Submodular Minimax Optimization: Finding Effective Sets.* AISTATS 2024. Related minimax formulation (course notes, C-K3).
- Avenhaus, R., von Stengel, B. & Zamir, S. (2002). *Inspection Games.* Handbook of Game Theory, Vol. 3, Ch. 51.