SAFETY AFTER DECENTRALIZATION: A THEORY OF POLICY EXTRACTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline multi-agent reinforcement learning often learns a joint distribution of states and actions, then fits a separate policy for each agent. Acting independently changes which actions occur together and which states the team visits. As a result, a stronger safety penalty can lower the training target's cost while raising the deployed policy's cost, even when optimization and policy fitting are exact. In the one-state setting with Kullback–Leibler (KL) regularization, binary actions, and binary costs, every type of cost that allows such a reversal has an example using at most four joint actions; four are sometimes necessary. We classify the costs for which a stronger penalty never raises deployed cost. We also identify when no penalty can reach a budget that an independent policy can satisfy. These findings motivate Deployed-Cost Projection (DCP), which adjusts the joint target using the gradient of deployed cost, preserves the state-flow constraints, and refits the local policies. We derive this gradient and give conditions for cost reduction when the gradient, projection, and policy fitting are approximate. In a structured safety task with up to 64 agents, DCP reaches budgets that penalty increases cannot reach. Controlled comparisons show how cross-agent action covariance helps DCP retain more reward at the same cost.

1 INTRODUCTION

Offline cooperative control must learn useful team behavior from recorded joint decisions while keeping the deployed team's cost within a prescribed budget. Centralized training can use the full state and joint action, but each deployed agent acts using its own observation. A common approach first learns a joint state–action distribution, called an *occupancy*, subject to cost constraints, then fits a separate policy for each agent to this target (Lee et al., 2021; 2022; Bui et al., 2026). The connection between these two steps is essential: a budget satisfied by the training target must ultimately be satisfied by the agents that actually act.

Local policy fitting can break this connection even when it is exact. A joint target can coordinate agents by assigning probability to different safe action combinations. For example, it can alternate which of two agents enters a shared corridor. Each fitted agent can reproduce its own action frequencies perfectly, yet independent sampling can send both agents into the corridor together. With state transitions, these new action combinations also change the states the team visits. Thus the safety gap can arise from the structure of policy extraction itself, without optimization error or imperfect fitting.

This gap raises a practical question about the usual response to a violated budget: increasing the safety penalty. A larger multiplier makes costly joint decisions less attractive during training, but its effect on the extracted product policy is less clear. The penalty changes both the agents' action frequencies and the correlations that local extraction discards. A decrease in target cost therefore does not determine the direction of the deployed-cost change. We ask: *when does a stronger safety penalty reduce deployed cost, and when can it reach a budget that an independent policy can satisfy?*

We identify two distinct failures. First, increasing the penalty can raise deployed cost while lowering target cost. Second, even when deployed cost never increases, the entire penalty path can remain above an achievable budget. The second failure matters because the budget may be feasible for a simple deterministic product policy: increasing the multiplier alone still cannot reach it. This distinguishes a failure of the penalty path from the absence of any feasible decentralized policy.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

These failures call for different responses. A reversal means that a multiplier step can worsen the constraint it is intended to enforce. A strict floor means that further tuning along the same path cannot recover feasibility, even when the cost curve is nonincreasing. The practical objective is therefore to identify a direction that lowers the extracted team's cost while retaining a valid training target.

Our correction, Deployed-Cost Projection (DCP), uses the cost after policy extraction to guide the target update. It computes how changing the target affects each local policy and the resulting team cost, then preserves the state-flow constraints and refits the agents. The gradient accounts for each agent's effect on team cost and for how often the deployed team encounters each observation. This makes the extraction step part of the optimization objective. Our contributions are:

- **Conditions for penalty failure.** We derive the local target-cost and deployed-cost responses in finite MDPs. For one-state KL regularization with binary actions and Boolean (zero-or-one) costs, we prove that every cost admitting a reversal has a witness on at most four joint actions, and that four are sometimes necessary. We classify the costs that never reverse for any team size and characterize when penalties leave a strict cost floor above an attainable budget.
- **A correction based on deployed cost.** We derive the gradient through local extraction and the DCP update that preserves state flow. We give conditions for cost reduction with approximate gradients, projection, and actor fitting. A tight transfer bound and an information decomposition distinguish the effects of action dependence, information missing from local observations, and fitting error.
- **Tests of the failure and correction mechanisms.** Exact examples and random MDPs check the response formulas and bounds. In a structured task with up to 64 agents, DCP reaches budgets below the penalty floor. Comparisons at the same cost isolate how target covariance helps DCP retain more reward than a local actor update. Finite-data MDP experiments test the correction with estimated inputs.

2 Related Work

Occupancy correction and safe learning. OptiDICE optimizes policies through stationary distribution correction (Lee et al., 2021); COptiDICE adds cost constraints (Lee et al., 2022). ComaDICE combines correction with cooperative value decomposition (Bui et al., 2025). SamaDICE adds shared safety multipliers, a monotonic dual mixer and weighted local extraction (Bui et al., 2026). SamaDICE's corrected-cost sensitivity concerns the learned target; we differentiate cost after local extraction. These responses can have opposite signs even with an exact dual solution and exact local fits. DCP uses the deployed derivative to correct the target produced by the existing occupancy learner. Safe cooperative policy updates and sequence models address complementary learning settings (Gu et al., 2023; Xia & Xu, 2025); Section H gives the detailed relation.

Coordination and the policy gradient. AlberDICE shows that independent cloning can create unsupported joint actions and uses alternating optimization (Matsunaga et al., 2023). AgentMixer studies consistency between centralized correlated policies and local policies (Li et al., 2025). We ask whether increasing a safety multiplier lowers or raises deployed cost, and how this depends on the regularizer's curvature. COMA already uses counterfactual credit to measure each agent's effect on team performance (Foerster et al., 2018). Our gradient additionally accounts for how often each observation occurs under deployment compared with the target. To keep the updated target feasible, DCP uses the Hessian projection for equality constraints from Alvarez et al. (2004). These components play distinct roles: counterfactual credit measures an agent's contribution, the observation-frequency ratio differentiates local extraction, and the projection makes the resulting occupancy direction feasible. Their combination connects a change in the joint training target to the cost of the independently acting team.

Structural guarantees and distribution transfer. Polynomial growth transformations analyze multiplicative updates to independent distributions (Baum & Eagon, 1967; Baum & Sell, 1968). Our Boolean classification asks for a cost decrease for every joint target, including correlated ones. Costs based on the number of active agents connect to pivotal probabilities in reliability theory (Boland & Proschan, 1983); we prove the required covariance inequalities in Section A. Our bound on the

target-to-deployment cost gap uses standard KL contraction, also used in fixed-policy occupancy evaluation (van der Laan & Kallus, 2026). We show that an extracted product policy can attain the bound's factor, so that factor cannot be improved in general. The relevant contraction acts on the full state–action transition channel, before fixing a deployed policy. Local extraction can change both joint-action frequencies and future state visits; the mixing rate of the extracted state kernel alone need not control the resulting occupancy gap (Section E).

3 Problem Setup: Occupancies and Local Policies

Cooperative control and the deployed objective. Consider a finite discounted cooperative MDP with states $\mathcal{S}$, joint actions $\mathcal{A} = \prod_{i=1}^{N} \mathcal{A}_i$, transition kernel P, initial distribution μ_0, reward r, and discount $0 < \gamma < 1$. Write $\mathbf{s} \in \mathcal{S}$ for a global state and $\mathbf{a} = (a_1, \dots, a_N) \in \mathcal{A}$ for a joint action; $\mathbf{S}_t, \mathbf{A}_t$ denote their random values at time t. Agent i observes $o_i = O_i(\mathbf{s})$ and samples from $\pi_i(a_i \mid o_i)$; the deployed joint policy is $\pi(\mathbf{a} \mid \mathbf{s}) = \prod_i \pi_i(a_i \mid O_i(\mathbf{s}))$. Centralized training observes the state and joint action.

For K team costs $0 \le c_k \le c_{\max}$, the deployed requirements are $J_{c_k}(\pi) = \mathbb{E}_\pi \sum_{t \ge 0} \gamma^t c_k(\mathbf{S}_t, \mathbf{A}_t) \le \widehat{c}_k$. Write $\bar{c}_k = (1-\gamma)\widehat{c}_k$ for normalized budgets and $d_\pi(\mathbf{s}, \mathbf{a}) = (1-\gamma) \sum_{t \ge 0} \gamma^t \Pr_\pi(\mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a})$ for the normalized occupancy. It records how often each state–action pair occurs, with later visits discounted, and sums to one. Thus $(1-\gamma) J_{c_k}(\pi) = c_k^\top d_\pi$ is the normalized deployed cost that must stay within budget.

The corrected joint occupancy. Let b be the reference occupancy represented by the offline data. The learner chooses a new joint occupancy q. It must satisfy *discounted Bellman flow*: the weight assigned to each state must match the weight entering that state from the initial distribution and from transitions. This gives the constraint

$$Aq = b_0, \qquad A_{\cdot,(\mathbf{s},\mathbf{a})} = e_{\mathbf{s}} - \gamma P(\cdot \mid \mathbf{s}, \mathbf{a}), \qquad b_0 = (1-\gamma)\mu_0. \tag{1}$$

Here $e_{\mathbf{s}}$ is the state-$\mathbf{s}$ basis vector, and Bellman flow implies $\sum_{\mathbf{s},\mathbf{a}} q(\mathbf{s}, \mathbf{a}) = 1$. For a convex divergence function f and strength $\alpha > 0$, the learner balances reward against departure from the offline reference, subject to target-cost budgets. With $x = (\mathbf{s}, \mathbf{a})$, it solves

$$\max_{q \ge 0,\ Aq = b_0} r^\top q - \Phi(q) \quad \text{subject to } c_k^\top q \le \bar{c}_k \quad (k \in [K]), \qquad \Phi(q) = \alpha \sum_x b_x f(q_x / b_x), \tag{2}$$

Each team cost has a shared multiplier $\lambda_k \ge 0$, which sets the strength of its safety penalty. Its update uses the target cost $c_k^\top q$. This can differ from the eventual deployed cost. The DICE dual and SamaDICE's mixer parameterization are given in Section H. We analyze a normalized target $q = bw$, where $w_x = q_x / b_x$ is its density ratio on the reference support.

The extracted local policy. The joint target q may correlate agents' actions; *policy extraction* fits a product policy for deployment. For a positive normalized target, fitting each local policy by weighted maximum likelihood, with no restriction on its policy class and exact expectations, gives

$$\begin{gathered} \pi_i^q = \arg\max_{\pi_i} \mathbb{E}_q \log \pi_i(A_i \mid O_i), \qquad \pi_i^q(a_i \mid o_i) = q(a_i \mid o_i), \\ \pi^q(\mathbf{a} \mid \mathbf{s}) = \prod_i \pi_i^q(a_i \mid O_i(\mathbf{s})). \end{gathered} \tag{3}$$

Here $q(a_i \mid o_i)$ is agent i's action distribution given its observation under the joint target. Each local policy reproduces this distribution, but their product need not reproduce the joint target. We study the cost after this policy extraction, $F_k(q) = (1-\gamma) J_{c_k}(\pi^q)$. The training target is q; the deployed policy has occupancy d_{π^q}, which can differ in both joint actions and state visits. For a selected cost c we write F_c or F. A fitted local actor is denoted by $\widehat{\pi}^q$, with cost $\widehat{F}(q) = (1-\gamma) J_c(\widehat{\pi}^q)$. The finite-state analysis includes finite-memory policies when the memory is part of the Markov state and uses the same occupancy weights and flow.

Assumptions for differentiating the penalty response. To differentiate how the target changes, we keep its support fixed and assume $q, b > 0$ on that support. We also assume strictly positive

regularizer curvature, $f''(q/b) > 0$, and linearly independent rows in the flow constraints. The full finite model has full row rank; when restricting the support, we retain only independent flow rows. Policy evaluation assumes $d_{\pi^q} \ll b$: every state–action pair visited by the deployed policy has positive reference weight. This holds when the reference is positive over the modeled state–action space.

4 When Do Safety Penalties Survive Decentralization?

A corrected target can look safe while its extracted policy is costly. In one state with two binary agents, let q put mass 0.49 on each of $(1,0),(0,1)$ and 0.01 on each of $(0,0),(1,1)$. For the cost $c(\mathbf{a}) = \mathbf{1}\{a_1 = a_2 = 1\}$, the target cost is 0.01, but each cloned agent chooses action one with probability $1/2$, so deployed cost is 0.25. This gap raises two questions: whether a larger penalty lowers deployed cost, and which budgets its path can reach. The results below answer both; proofs are in Sections A and D.

The response after extraction. Vary λ_k while keeping all other multipliers fixed. At each multiplier value, let q_λ solve the penalized problem $\max_{q \geq 0,\ Aq = b_0} \{r^\top q - \Phi(q) - \sum_j \lambda_j c_j^\top q\}$. Assume an interior solution and a twice continuously differentiable f with positive curvature. Define $H = \nabla^2 \Phi(q) = \alpha \operatorname{diag}\{f''(q_x/b_x)/b_x\}$ and $W = H^{-1}$. The response matrix is

$$M = W - WA^\top (AWA^\top)^{-1} AW. \tag{4}$$

The first term scales an update by inverse curvature; the second removes the component that violates Bellman flow (Section D.1). The matrix is positive semidefinite and satisfies $AM = 0$, so updates Mv preserve flow. Write $\langle u, v \rangle_M = u^\top M v$ and $\|u\|_M^2 = \langle u, u \rangle_M$.

Theorem 4.1 (Target and deployed cost responses)**.** *At the stated solution, for each multiplier λ_k and any differentiable extracted-policy cost F_j,*

$$\frac{dq}{d\lambda_k} = -Mc_k, \qquad \frac{d(c_k^\top q)}{d\lambda_k} = -\|c_k\|_M^2, \qquad \frac{dF_j(q)}{d\lambda_k} = -\langle \nabla F_j(q), c_k \rangle_M. \tag{5}$$

Moreover, $AM = 0$. Thus increasing λ_k does not raise the corresponding deployed cost F_k locally if and only if $\langle \nabla F_k, c_k \rangle_M \geq 0$.

Target cost cannot increase, but deployed cost rises when the target-cost vector and deployed-cost gradient have negative inner product. The latter gradient includes policy extraction and changed state visits. The proof differentiates the optimality conditions along directions satisfying $A\,dq = 0$ (Section D).

In one state, there is no change in state visits, so the difference comes entirely from how agents' actions combine. Under KL regularization, the target-cost derivative is minus a cost variance. The deployed-cost derivative instead depends on covariance between target cost and the effect of each agent's action after extraction. This can have the opposite sign. Other regularizers change the weights in this comparison; Sections D and D.3 give the derivation and a Hellinger example where deployed cost rises even from a product target.

Which binary costs never increase? We now ask which Boolean costs ensure that increasing the penalty never raises deployed cost, for *any* target distribution. For any number of binary agents in a one-state KL problem, start from a positive target $q = q_0$ and increase the penalty through $q_\lambda \propto q \exp(-\lambda c)$, absorbing α into λ (the normalized formula is in (11), Appendix A.1). This update reduces the weight of high-cost joint actions and then renormalizes the distribution. Call c universally monotone if $F_c(q_\lambda)$ is nonincreasing for every positive baseline q and every $\lambda \geq 0$.

A *witness* is a target distribution for which a small penalty increase raises deployed cost. The four-action witness theorem (Theorem A.1, Appendix A.1) says that whenever a cost allows this failure, some witness needs only four joint actions. Four are sometimes necessary, even with three agents. Adding a small amount of probability to every other action preserves a strict reversal, giving a positive target. The theorem does not say that every failing target can itself be reduced to four actions. It applies to the KL update in (11) and exact local policy extraction.

The full classification (Theorem A.2, Appendix A.1) identifies all costs that never reverse: constants, thresholds on the number of active agents, a second family that charges a cost if any agent in one group acts or at least two in another group act, and two-agent XOR, up to simple transformations. The allowed transformations mean relabeling agents, swapping an agent's zero and one actions, replacing c by $1 - c$, or ignoring agents that do not affect cost. Two-coordinate XOR is one when the two actions differ. Thus thresholds on the number of active agents never reverse, even in large teams. In contrast, a cost can increase with each agent's action and still rise after a larger penalty and policy extraction. Complete proofs and the sharp lower bound are in Section A; lower-dimensional specializations are in Sections B and C.

Combining two pairwise hazards gives a concrete example of failure: $c(\mathbf{a}) = \mathbf{1}\{a_1 = a_2 = 1 \text{ or } a_3 = a_4 = 1\}$. Changing any agent's action from zero to one never lowers this cost. Nevertheless, a larger KL multiplier can raise deployed cost for a positive target distribution (Proposition A.13 and Figure 4).

Which budgets can the multiplier reach? Even if deployed cost never increases, it can remain above a cost that some product policy can achieve. Let $\mathcal{A}_* = \arg\min_{\mathbf{a}} c(\mathbf{a})$ and $c_{\min} = \min_{\mathbf{a}} c(\mathbf{a})$.

Proposition D.2 in Appendix D.4 shows that the penalty path has a strict cost floor exactly when $\mathcal{A}_*$ is not a product set. A *product set* has the form $\prod_i \mathcal{A}_{*,i}$: any combination of its agents' allowed actions is still in the set. If the minimum-cost actions do not form such a set, combining their marginals independently can create actions with higher cost, even at an infinite penalty. Write $F_\star = \inf_{\lambda \geq 0} F_c(q_\lambda)$ for the lowest deployed cost along the penalty path. Every budget in $[c_{\min}, F_\star)$ can be met by choosing a minimum-cost joint action deterministically, but no penalty on this path can meet it. For example, the two-agent AND cost penalizes both agents entering a corridor. Its zero-cost actions are $00, 01, 10$; their marginals can also generate the costly action 11. This cost never reverses under a larger KL penalty, yet it has a strict floor. Additive costs and costs with a unique minimizer have product minimum sets and no strict floor.

5 DCP: Correcting Deployed Cost

The penalty-response result suggests adjusting the joint target using *deployed* cost directly. DCP takes three steps: compute the deployed-cost gradient through local policy extraction, project it onto directions that preserve state flow, and refit and evaluate the local policies (Figure 1). For a selected cost c, write $\pi = \pi^q$, $F(q) = (1-\gamma)J_c(\pi)$, and let V_c^π and Q_c^π denote its expected discounted future costs starting from a state and a state–action pair, respectively.

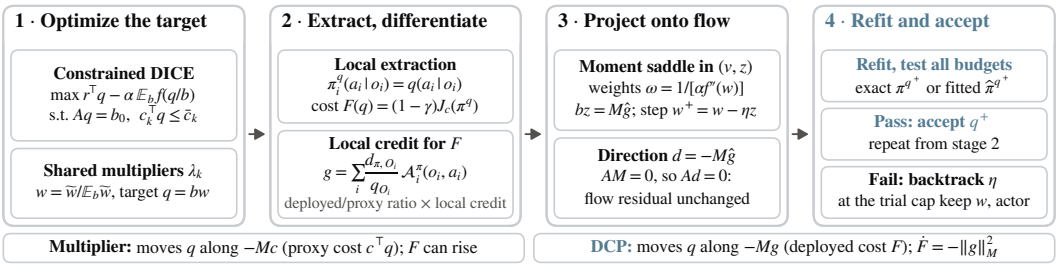


Figure 1: Deployed-Cost Projection (DCP). A constrained occupancy learner produces a joint target; local extraction produces the policy that will act. DCP estimates the deployed-cost gradient, projects it to preserve Bellman flow, refits the local actors, and accepts a candidate only after the cost test (Algorithm 1). The bottom row contrasts the multiplier's joint-cost direction with DCP's deployed-cost direction.

5.1 Differentiate the Extracted Policy

Let $d_{\pi,S}$ and d_{π,O_i} be the state and agent-i observation marginals of the deployed occupancy, and let q_{O_i} be the corresponding target observation marginal. To measure the effect of one agent's action, we average over the states consistent with its observation and over its teammates' actions. Subtracting

the policy's usual cost-to-go gives the local *counterfactual advantage*:

$$\mathcal{A}_i^\pi(o_i, a_i) = \mathbb{E}_{\mathbf{S}\sim d_{\pi,S}(\cdot|o_i),\ \mathbf{A}_{-i}\sim\pi_{-i}(\cdot|\mathbf{S})}[Q_c^\pi(\mathbf{S}, a_i, \mathbf{A}_{-i}) - V_c^\pi(\mathbf{S})]. \tag{6}$$

Theorem 5.1 (Deployed-cost gradient)**.** *For the extractor in* (3)*,*

$$g(\mathbf{s}, \mathbf{a}) := \frac{\partial F}{\partial q(\mathbf{s}, \mathbf{a})} = \sum_i \frac{d_{\pi,O_i}(O_i(\mathbf{s}))}{q_{O_i}(O_i(\mathbf{s}))} \mathcal{A}_i^\pi(O_i(\mathbf{s}), a_i). \tag{7}$$

Terms with zero deployed marginal are zero. The global cost and transition kernel may couple all agents.

The ratio $d_{\pi,O_i}/q_{O_i}$ upweights observations encountered more often under deployment. The advantage measures each action's effect on team cost relative to the agent's usual behavior. Policy evaluation uses centralized information; each gradient term depends only on one agent's observation and action. The proof combines the discounted policy derivative with the derivative of a conditional marginal (Section D.5). Compared with counterfactual policy gradients (Foerster et al., 2018), differentiating the occupancy supplied to the extractor introduces the observation-frequency ratio. The flow projection then uses the DICE regularizer's curvature.

5.2 Project the cost gradient onto Bellman flow

Following the deployed-cost gradient directly could violate the state-flow constraints. DCP uses the regularizer's inverse curvature to scale the gradient, then removes the part that would violate flow. For an estimated gradient $\widehat{g}$ and step size $\eta > 0$, the ideal update is

$$q^+ = q - \eta M \widehat{g}. \tag{8}$$

Because $AM = 0$, the update leaves Aq unchanged. Thus a target that satisfies Bellman flow continues to satisfy it; if the starting target has a flow error, the update preserves that error. A small enough step also keeps all target probabilities positive. With an exact gradient, the continuous update satisfies $\dot{F} = -\|\nabla F(q)\|_M^2 \le 0$, so deployed cost cannot increase. For a finite step with exact projection, set $\rho = \|\widehat{g}\|_M$ and assume $\|\widehat{g} - g\|_M \le \epsilon$. If F is L-smooth along the positive segment from q to q^+, Theorem D.3 gives

$$F(q^+) - F(q) \le -\eta\rho(\rho - \epsilon) + \frac{L\eta^2}{2}\|M\widehat{g}\|_2^2. \tag{9}$$

Thus $\rho > \epsilon$ and $L\eta\|M\widehat{g}\|_2^2 \le \rho(\rho - \epsilon)$ guarantee a decrease of at least $\eta\rho(\rho - \epsilon)/2$ for $\eta > 0$. The estimated descent signal must exceed its error, and the step must control the curvature term. Theorem D.3 also allows projection error. Proposition D.5 and Equation (33) show how to compute the same direction from transition samples without forming M explicitly.

What the correction does to local policies. For one state, binary actions, and KL regularization, write $p_i = \mathbb{E}_q A_i$ and $F(q) = \widetilde{F}(p)$. The exact-gradient DCP direction changes local action probabilities according to Proposition D.6:

$$\frac{dp}{d\eta} = -\frac{1}{\alpha}\operatorname{Cov}_q(\mathbf{A})\,\nabla\widetilde{F}(p). \tag{10}$$

The local actor update scales each agent's gradient by that agent's action variance. DCP also uses cross-agent covariance, which records how agents' actions vary together in the target. For example, if the target mixes two safe patterns in which one agent acts and the other does not, this covariance can steer DCP toward one pattern rather than reduce both agents' action probabilities. The target's joint structure can therefore help preserve reward while reducing cost; Section D.9 derives the natural-gradient interpretation.

5.3 Refit and evaluate the local actors

After each projected update, DCP refits the local actors. The cost decrease transfers from exact extraction to fitted actors when their errors at both the old and new targets are sufficiently small (Corollary D.4).

DCP therefore refits and evaluates each candidate after the projection. For several constraints, it checks every deployed-cost budget before accepting a step (Algorithm 1). Exact finite-state checks use true costs; the offline version uses fitted evaluators. The fitting, normalization, and acceptance rules are detailed in Section G.

5.4 Relate extraction error to deployed cost

A low target cost alone does not guarantee a low deployed cost. We can bound the gap by measuring how much the extracted policy's action distribution differs from the target's at each state. Write $K_q(\pi)$ for this KL divergence averaged under the target. Theorem E.1 bounds the cost gap using $K_q(\pi)$ and a parameter κ that measures how strongly transitions can preserve differences between state–action distributions. The factor $1/(1-\gamma\kappa)$ captures how these differences accumulate over time. The appendix defines κ precisely and gives a model that attains the factor, showing that it cannot be improved in general.

Equation (38) separates this mismatch into three sources: dependence among agents' actions, state information missing from local observations, and error in fitting the local actors. Thus even exact fitting can leave a cost gap when the joint target contains coordination or information the deployed agents cannot reproduce.

6 Experiments

The experiments test three questions. Does deployed cost respond to a penalty as the theory predicts? Can DCP meet budgets that penalty increases cannot, and does cross-agent covariance help it retain reward? Does DCP still reduce cost when its inputs are estimated from offline data? We enumerate finite models or solve them numerically, and verify rational examples with exact arithmetic. Results labeled *estimated* use models fitted to data; results labeled *oracle* use the true data-generating model. Methods in each comparison receive the same information.

6.1 Multiplier response and structural checks

On 16,384 random MDPs with two or three agents, $|\mathcal{S}| \in \{4, 8\}$ and $\lambda \in \{0, 0.5, 1\}$, the analytic derivative agrees with a numerical derivative obtained by re-solving the target at nearby multiplier values. The relative error is at most 2.6×10^{-5}. Increasing the penalty lowers target cost in every KL and Hellinger case. However, it raises deployed cost in 0.10% and 0.19% of fully observed cases, respectively. These rates rise to 7.2% and 15.2% when different states share the same local observation, a setting called *observation aliasing* (Figure 7). Thus deployed cost can move in the opposite direction even when the target optimization is solved accurately.

We also test the structural predictions on 2^{20} one-state problems per experimental condition, with up to six agents. Additive costs have no reversals or multiplier floors. The two-agent Boolean cases and the universally monotone three-agent classes have no KL reversals, matching the exact classification. Indicator costs often leave a gap on the tested multiplier grid. With six agents, the lowest deployed cost found is more than 0.01 above $c_{\min}$ in 95–100% of the pairwise, Boolean, and paired-hazard problems. The median gap is 0.16–0.39 (Table 1 and Figure 6). Separate exact checks cover all four-variable Boolean cost tables. To test the bound with state transitions, a five-state construction attains its factor to 9×10^{-15} relative error, and none of 884,736 random checks violates the bound (Figure 8).

6.2 Budget correction and the role of covariance

We compare methods at the same deployed-cost budget. To test the role of cross-agent covariance, we also run DCP after removing correlations from its target. Our structured safety-switch task allows this comparison with larger teams: agents earn reward for acting, but certain pairs incur a cost if both act.

The family has $N = 4B$ binary agents in B blocks. Each block has cost $C_b = \mathbf{1}\{a_{b,1} = a_{b,2} = 1 \text{ or } a_{b,3} = a_{b,4} = 1\}$ and reward $R_b = \frac{1}{4}\sum_i a_{b,i}$; team cost and reward are block averages. The reference assigns 95% of its weight to the three-action distribution in Section A.11 and 5% to a uniform distribution. For each block, the target is $q_\lambda \propto \widehat{b}\exp((R - \lambda C)/\alpha_0)$ with $\alpha_0 = 0.1$. The estimated comparison uses three datasets of 10^4 joint samples. Every method uses the same smoothed estimates of the block distributions.

Starting from the same target, we compare four methods: increasing the multiplier, DCP, a natural-gradient update of the local actors using the same deployed-cost information, and DCP-prod. DCP-

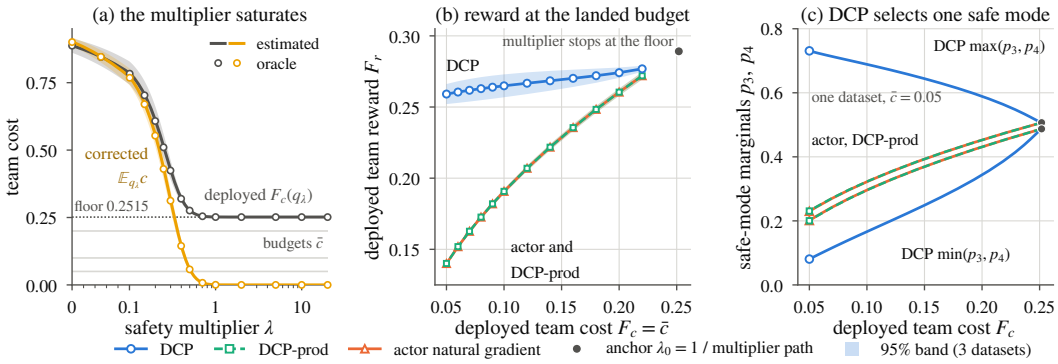


Figure 2: Structured safety switches ($N = 16$, estimated panel; bands are 95% t intervals over three datasets). (a) Along the multiplier path the corrected cost $\mathbb{E}_{q_\lambda} c$ falls to 4×10^{-4} at $\lambda = 1$, while the deployed cost $F_c(q_\lambda)$ approaches 0.252 (oracle floor 0.2515, dotted line), above the displayed budgets; circles show the oracle. (b) Deployed reward when each update method reaches exactly $F_c = \bar{c}$, starting from $\lambda_0 = 1$: DCP keeps more reward than the actor at every plotted budget, by 0.119 at $\bar{c} = 0.05$, and DCP-prod coincides with the actor; the multiplier stays at the floor. (c) Block means of $\max(p_3, p_4)$ and $\min(p_3, p_4)$ along the correction to $\bar{c} = 0.05$ (one dataset): DCP selects one safe mode, whereas the actor and DCP-prod lower both agents.

prod replaces the target by the product of its marginals before each step, removing the cross-agent covariance in Proposition D.6. The three update methods limit each agent's policy change to a KL divergence of 10^{-3} and accept only cost-decreasing steps. Their final step is adjusted to reach the selected budget exactly. As an additional control, we refit the actors without changing the starting target.

In this family, DCP reaches budgets below the multiplier floor. The numerical minimum is 0.251–0.252 across the tested team sizes and datasets, and the argument combining a finite grid with a bound for larger multipliers gives $F_\star > 0.23$ for every $\lambda \geq 0$ (Section I.1). At $\lambda_0 = 1$, corrected cost is about 4×10^{-4} while deployed cost remains above 0.20. DCP and the actor reach all tested budgets. Starting from targets with penalties $\lambda_0 \in \{0.5, 1\}$, DCP retains more reward at $\bar{c} \in \{0.20, 0.10, 0.05\}$ for $N \in \{4, 8, 16, 32, 64\}$ at the first update that meets the budget, and for $N \in \{4, 16, 64\}$ when both methods finish at the same cost (Tables 2, 3 and 5). At $N = 16$, $\lambda_0 = 1$ and $\bar{c} = 0.05$, the matched-cost reward gain is 0.119, with paired 95% interval $[0.111, 0.127]$.

The covariance control identifies how the gain arises. DCP-prod matches the actor within 1.1×10^{-4} in all 54 estimated comparisons at the same cost. DCP instead separates the two safe-mode probabilities: $|p_3 - p_4| = 0.66$, against 0.02 for the actor in the setting above. A controlled 1% reference asymmetry gives an oracle gain of 0.116–0.117 at $\bar{c} = 0.05$ from $\lambda_0 \in \{0.5, 1\}$ (Figure 9). Across the one-state tests, DCP retains more reward than DCP-prod in 79–95% of the problems in each binary-cost condition (Table 6). DCP also retains more reward at $\bar{c} \leq 0.10$ in the reported tests that vary reference weights and step sizes (Table 4).

6.3 Finite-data correction in MDPs

The preceding family isolates the one-state mechanism. We next test whether the correction still lowers deployed cost when dynamics, observation aliasing, and estimation errors are present.

We use eight random MDPs with $|\mathcal{S}| = 8$, two agents, and full or aliased observations. Each setting uses five independent datasets with 100–10,000 discounted trajectories each. To identify the effect of estimation error, we replace the exact gradient, flow projection, and actor fit with estimated versions, first separately and then together. This comparison keeps the same step-size normalization based on true observation frequencies. All 1,200 corresponding DCP steps lower true deployed cost, including 960 steps with fitted components. With 100 trajectories, the mean fully fitted cost change is within 4×10^{-4} of the oracle change. Reversing the direction raises cost in all cases, and the fitted evaluator rejects every such step (Table 7 and Figure 10). We also add controlled errors to test the bounds in

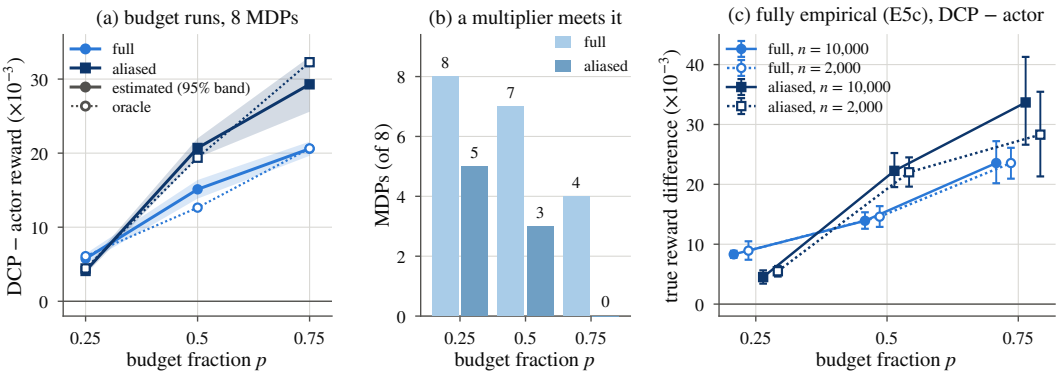


Figure 3: Correction to a budget in the eight random MDPs; reward differences in units of 10^{-3}. (a) Final reward of DCP minus that of the actor at the first update that meets the budget: estimated-panel mean with 95% interval (shaded) and oracle mean (open markers). (b) Number of MDPs in which some multiplier $\lambda \leq 200$ meets the budget (oracle); DCP and the actor meet every budget under the model used in each panel. (c) Fully empirical correction: DCP minus actor true reward over runs in which both meet the budget under the true model, with 95% paired t intervals (Table 9).

Theorem D.3 and Corollary D.4. Neither bound is violated in 0.86 million and 3.4 million bandit checks, respectively, that satisfy the theorem or corollary's assumptions.

Repeated correction reaches budgets that the multiplier path misses. At the tightest budget, the multiplier succeeds in four of eight fully observed MDPs and none of the aliased MDPs. DCP and the actor reach every budget under the model used in each comparison. DCP retains 0.004–0.032 more reward on average, with all six estimated-panel paired intervals above zero (Table 8). A separate fully empirical variant fits all update components and step sizes on 60% of the data. It uses another 20% to decide whether to accept each step and a bootstrap-based allowance in the stopping rule. Under the true model, this variant meets the budget in 95.8% of runs across the reported configurations. Its mean true reward gain over the actor is positive in all twelve settings. Among runs where both methods meet the true budget, the paired intervals are above zero (Table 9). The finite-data results therefore test the correction after fitting both the target and the quantities used to update it.

The safety-switch controls separate budget correction from reward preservation. DCP and the actor receive the same cost information and local KL limit; both cross the multiplier floor. DCP-prod removes cross-agent covariance while preserving action marginals, and its agreement with the actor identifies the source of DCP's reward gain. The MDP experiments test whether estimated gradients, projection, and actor fitting preserve the correction. Evaluating true cost after these updates connects the measured improvements to the error conditions in Theorem D.3 and Corollary D.4.

7 Conclusion

This paper shows that a stronger safety penalty on a joint training target does not necessarily make independently acting agents safer. Deployed cost can increase even when target cost decreases. It can also remain above a feasible budget for every penalty value. In one-state KL problems with binary actions and binary costs, we identify exactly which costs avoid the first failure. Every cost that allows this failure has a counterexample using at most four joint actions. DCP reduces deployed cost by adjusting the joint target based on the policies that agents will actually use. We derive this update and give conditions under which a step lowers cost. In our experiments, DCP reaches budgets that increasing the penalty cannot reach. The safety-switch tests also show that using correlations between agents' actions helps DCP retain more reward than a local actor update at the same cost.

Future work should develop budget guarantees that account for errors in costs estimated from offline data. It should also establish when repeated DCP updates reach a feasible budget and bound the associated reward loss. Further experiments should test DCP with neural policies in larger, less structured multi-agent environments, where policy fitting and cost estimation are more demanding.

Statement of Use of AI

The research questions, the main ideas, the methodology, and the experimental protocols of this paper were chosen by the authors, who divided the work among themselves, supervised every stage of the project, and made all decisions on claims and conclusions. Under the authors' direction, we used generative AI tools as assistants for the following tasks with required disclosure: refining the conceptual framing, discussing candidate conjectures, stress-testing mathematical claims and proof arguments, and implementing and running the experiments. In the theoretical component, generative AI was used in part as a "theory attacker": it assisted in implementing computational searches and large-scale simulations over extensive collections of generated examples in order to identify potential counterexamples, edge cases, or weaknesses in candidate conjectures, theorem statements, and proof arguments. These searches were used to help the authors examine, reject, or refine candidate claims and to increase confidence in statements that survived extensive computational stress testing. The failure of such searches to identify a counterexample does not constitute a proof, and no theorem in this paper is considered established on the basis of AI-assisted simulation, search, or testing.

The authors independently verified all theorem statements and proofs, citations, experimental specifications, and AI-assisted text against their research artifacts. No experimental data or results were produced by generative AI tools themselves: every reported number comes from experiments that were actually executed, and translation and qualitative data analysis are not applicable to this work. Additionally, we used generative AI tools for literature discovery and for drafting and editing the manuscript, including checks of notation and of the consistency of reported numbers. The authors take full responsibility for every claim, proof, citation, and experimental result in this paper; no AI system is an author.

Reproducibility statement

The appendices develop the analytic proofs and exact finite checks, and give the experimental protocols and detailed result tables. The structural checks use exact rational arithmetic; the empirical tests use enumerated or numerically solved finite models. Figures and matched controls connect each reported measurement to the corresponding result. The supplementary material provides the source code for all experiments, figures, and exact checks, together with the recorded outputs behind every reported result.

References

Felipe Alvarez, Jérôme Bolte, and Olivier Brahic. Hessian riemannian gradient flows in convex programming. *SIAM Journal on Control and Optimization*, 43(2):477–501, 2004. doi: 10.1137/S0363012902419977. URL https://www.math.tecnico.ulisboa.pt/~brahic/Legendre10.pdf.

Leonard E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bulletin of the American Mathematical Society*, 73(3):360–363, 1967. doi: 10.1090/S0002-9904-1967-11751-8.

Leonard E. Baum and George Roger Sell. Growth transformations for functions on manifolds. *Pacific Journal of Mathematics*, 27(2):211–227, 1968. URL https://msp.org/pjm/1968/27-2/pjm-v27-n2-p01-p.pdf.

Philip J. Boland and Frank Proschan. The reliability of K out of N systems. *The Annals of Probability*, 11(3):760–764, 1983. doi: 10.1214/aop/1176993520.

The Viet Bui, Thanh Hong Nguyen, and Tien Mai. ComaDICE: Offline cooperative multi-agent reinforcement learning with stationary distribution shift regularization. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2410.01954.

The Viet Bui, Tien Anh Mai, and Thanh Hong Nguyen. SamaDICE: Safe multi-agent reinforcement learning with stationary distribution correction estimation. In *Advances in Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=jLb42822IZ. Accepted for publication.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11794. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, pp. 103905, 2023.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Jongmin Lee, Cosmin Paduraru, Daniel J. Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2204.08957.

Zhiyuan Li, Wenshuai Zhao, Lijun Wu, and Joni Pajarinen. AgentMixer: Multi-agent correlated policy factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025. URL https://ojs.aaai.org/index.php/AAAI/article/view/34048.

Daiki E. Matsunaga, Jongmin Lee, Jaeseok Yoon, Stefanos Leonardos, Pieter Abbeel, and Kee-Eung Kim. AlberDICE: Addressing out-of-distribution joint actions in offline multi-agent RL via alternating stationary distribution correction estimation. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.nips.cc/paper_files/paper/2023/file/e5b6eb1dbabff82838d5e99f62de37c8-Paper-Conference.pdf.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Lars van der Laan and Nathan Kallus. Fitted occupancy-ratio evaluation without Bellman completeness. *arXiv preprint arXiv:2607.05375*, 2026. URL https://arxiv.org/abs/2607.05375.

Yuchen Xia and Yunjian Xu. MOSDT: Self-distillation-based decision transformer for multi-agent offline safe reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

APPENDIX ROADMAP

Section A.1 states the four-action witness theorem and the full Boolean classification summarized in the main text; Section A gives their proofs, including the finite-base argument and sharpness. The two- and three-agent specializations are in Sections B and C. Section D gives the projection matrix and response proofs, the formal multiplier-floor and covariance results, and the correction guarantees. Sections E and F establish the transfer bound and its information decomposition. Implementation and prior-work details follow in Sections G and H; Section I collects the experimental protocols and full results.

A PROOF OF THE FOUR-MODE THEOREM AND UNIVERSAL CLASSIFICATION

This appendix states and proves Theorems A.1 and A.2, including the sharpness of four modes. Here a mode means a joint action with positive target probability. All statements concern the finite one-state KL problem with Boolean costs, binary actions, and exact independent marginal extraction.

A.1 FORMAL STRUCTURAL RESULTS

This subsection states the two structural results summarized in Section 4. The first shows that four joint actions suffice to demonstrate any possible penalty reversal. The second identifies all binary costs for which deployed cost never increases along the KL penalty path. We define that path and the response before stating the results; the following subsections develop their proofs.

Let $q = q_0$ be a positive baseline distribution, and absorb the regularization strength α into the multiplier. The one-state KL penalty path is

$$q_\lambda(\mathbf{a}) = \frac{q(\mathbf{a})e^{-\lambda c(\mathbf{a})}}{\sum_{\mathbf{u}} q(\mathbf{u})e^{-\lambda c(\mathbf{u})}}. \tag{11}$$

For this path, a cost is universally monotone if its deployed cost is nonincreasing for every positive baseline and every $\lambda \geq 0$. The following statements give the exact results discussed in Section 4.

For any distribution q, including one that gives positive probability to only a few actions, let Πq be the product of its action marginals, and define

$$D_c(q) = \frac{d}{d\lambda}\mathbb{E}_{\Pi q_\lambda} c\Big|_{\lambda=0}. \tag{12}$$

Theorem A.1 (Four-action witness)**.** *For every N and every $c : \{0,1\}^N \to \{0,1\}$, universal monotonicity fails if and only if $D_c(\widetilde{q}) > 0$ for some law $\widetilde{q}$ supported on at most four joint actions. The constant four is optimal, already for $N = 3$.*

The costs that never reverse have a simple classification. For sets of agents J, H, I with $H \cap I = \varnothing$, define two families:

$$\begin{aligned} T_{J,k}(\mathbf{a}) &= \mathbf{1}\Big\{\sum_{j\in J} a_j \geq k\Big\}, \\ G_{H,I}(\mathbf{a}) &= \mathbf{1}\Big\{\exists h \in H : a_h = 1 \text{ or } \sum_{i\in I} a_i \geq 2\Big\}. \end{aligned} \tag{13}$$

The threshold $T_{J,k}$ charges a cost when at least k agents in J choose action one. The cost $G_{H,I}$ charges a cost when any agent in H chooses one, or at least two agents in I do so.

Theorem A.2 (Universal Boolean classification)**.** *A Boolean cost is universally monotone if and only if, up to coordinate permutations, input-bit flips, output complementation, and irrelevant coordinates, it is a constant, a cardinality threshold $T_{J,k}$ ($1 \leq k \leq |J|$), a cost $G_{H,I}$, or two-coordinate XOR.*

Why four modes suffice. The proof distinguishes agents whose risky action depends on their teammates from agents with a consistent risky-action label. Small restrictions give sparse reversals in the first case; in the second, four-agent restrictions and a finite base lead to the families in Theorem A.2. The four witness actions can differ across all agents. An exact three-agent construction shows why three actions do not suffice (Section A and Proposition A.5).

Proof dependencies. Lemma A.3 and Theorem A.12 prove the safe families analytically, independently of the finite classification in Theorem A.4. That finite base supplies the sparse witnesses and local exclusions used by Theorems A.9 and A.11 in arbitrary dimension. The main theorems then follow in Section A.9; Proposition A.5 proves optimality of four.

A.2 Response notation and restriction operations

We express the deployed response as a function of action marginals. This form makes it possible to compare a full Boolean cost with its restrictions to fewer coordinates.

Let $c : \{0,1\}^N \to \{0,1\}$, let q be a joint law, and put $p_i = \mathbb{E}_q A_i$. To distinguish the marginal-vector argument from the joint-law argument in the main text, write

$$\mathscr{F}_c(p) = \sum_{\mathbf{a}\in\{0,1\}^N} c(\mathbf{a}) \prod_{i=0}^{N-1} p_i^{a_i}(1-p_i)^{1-a_i}, \qquad F_c(q) = \mathscr{F}_c(\mathbb{E}_q \mathbf{A}).$$

We index coordinates from zero in this appendix, so the integer code of an action $\mathbf{a}$ is $\sum_{i=0}^{N-1} 2^i a_i$. In the finite enumeration below, an unbold integer index denotes this code; unbold s in later Boolean proofs denotes a scalar probability or odds. With the normalized KL tilt in (11), define

$$D_c(q) = \frac{d}{d\lambda} \mathscr{F}_c(\mathbb{E}_{q_\lambda} \mathbf{A})\Big|_{\lambda=0}.$$

The covariance response (28) becomes, for $t = \mathbb{E}_q c$, $k = \mathrm{Cov}_q(c, \mathbf{A})$, and $w = \nabla \mathscr{F}_c(p)$,

$$D_c(q) = -w^\top k = \sum_{\mathbf{a}} q(\mathbf{a}) c(\mathbf{a})\, w^\top (p - \mathbf{a}). \tag{14}$$

Indeed $q_0'(\mathbf{a}) = q(\mathbf{a})(t - c(\mathbf{a}))$, hence $p' = -k$. This formula is a continuous polynomial of degree at most $N+1$ on the entire probability simplex, including laws with zero masses. The multilinear formula for $\mathscr{F}_c$ also defines its derivatives at boundary marginals. A strict sparse reversal therefore survives the addition of sufficiently small positive uniform mass. Conversely, every finite tilt of a positive law is positive, so nonpositivity of D_c for every positive law is equivalent to the universal monotonicity defined in the main text. We call this property *universal safety* below. For a fixed selected cost, $\mathscr{F}_c$ is the marginal-vector function denoted by $\widetilde{F}$ in Proposition D.6.

Input permutations, input-bit flips, and output complementation preserve both the response sign and witness support size. For output complementation, both $\nabla \mathscr{F}_c$ and $\mathrm{Cov}_q(c, \mathbf{A})$ change sign in (14). Restricting a coordinate embeds a law into the original cube by fixing that coordinate; the corresponding component of the covariance is zero. Irrelevant coordinates can be marginalized or fixed. Thus an unsafe restriction provides a witness for the original cost, and every restriction of a universally safe cost is safe. These operations do not require preserving the marginals of a given witness.

A.3 Cardinality thresholds in every dimension

A cardinality threshold charges a cost when enough agents choose action one. We show that increasing the KL penalty never raises deployed cost for this family, regardless of team size or correlations in the target. The proof uses the probability that changing one agent's action changes the cost, called its pivotal probability. This establishes the first family in Theorem A.2.

Lemma A.3 (All-dimensional cardinality safety)**.** *For $c(\mathbf{a}) = \mathbf{1}\{\sum_i a_i \geq k\}$, $1 \leq k \leq N$, every joint law has $D_c(q) \leq 0$.*

Proof. It suffices first to consider positive q. Let $t = \mathbb{E}_q c$, $u_i = \mathbb{E}_q[A_i \mid c = 1]$, $v_i = \mathbb{E}_q[A_i \mid c = 0]$. Then $p = tu + (1-t)v$, $\sum_i u_i \geq k$, and $\sum_i v_i \leq k-1$. For independent Bernoulli variables B_i with means p_i, set $S = \sum_i B_i$, $\beta_j = \Pr(S = j)$, and $w_i = \Pr(S - B_i = k-1)$. These are the classical pivotal probabilities of a k-out-of-N system Boland & Proschan (1983). Direct differentiation gives

$$D_c(q) = -t(1-t) w^\top (u - v), \qquad \sum_i p_i w_i = k\beta_k, \quad \sum_i (1-p_i) w_i = (N-k+1)\beta_{k-1}.$$

Moreover, $w_i \geq \min(\beta_{k-1}, \beta_k)$. To see this, let $a = \Pr(S - B_i = k-2)$, $b = w_i$, $d = \Pr(S - B_i = k)$. Log-concavity gives $ad \leq b^2$. The two quantities $(1-p_i)b + p_i a$ and $(1-p_i)d + p_i b$ cannot both strictly exceed b, since that would force $a, d > b$.

If $\beta_k \leq \beta_{k-1}$, then $w^\top u \geq \beta_k \sum_i u_i \geq k\beta_k = w^\top p$. If $\beta_{k-1} \leq \beta_k$, then $w^\top(\mathbf{1} - v) \geq \beta_{k-1} \sum_i (1 - v_i) \geq (N - k + 1)\beta_{k-1} = w^\top(\mathbf{1} - p)$. In each case $w^\top(u - v) \geq 0$. Continuity covers boundary laws.

For completeness, the log-concavity input follows by Bernoulli convolution. If (a_j) is log-concave with interval support and $b_j = Aa_j + Ba_{j-1}$, $A, B \geq 0$, then

$$b_j^2 - b_{j-1}b_{j+1} = A^2(a_j^2 - a_{j-1}a_{j+1}) + B^2(a_{j-1}^2 - a_{j-2}a_j) + AB(a_j a_{j-1} - a_{j-2}a_{j+1}) \geq 0.$$

The final term is nonnegative by decreasing adjacent ratios; zero entries are handled by interval support. Starting with a point mass and iterating the Bernoulli convolutions proves the required Poisson-binomial property. □

A.4 The exact classification through four agents

We first classify every binary cost for four agents. Costs that differ only by transformations preserving the response sign can be checked together, which reduces the number of cases. The resulting classification supplies both safe families and explicit counterexamples used later to prove the results for any number of agents.

Encode a four-variable table by $h = \sum_{a=0}^{15} c(a)2^a$. Input permutations, input-bit flips, and output complementation define the equivalence classes used below. They preserve the response sign and witness support size. For complementation, both $\nabla\mathscr{F}_c$ and $\mathrm{Cov}_q(c, \mathbf{A})$ change sign, so (14) is unchanged.

Theorem A.4 (Complete four-agent classification)**.** *Exactly* 742 *of the* 65 536 *four-variable Boolean tables satisfy* $D_c(q) \leq 0$ *for every joint law. Their eleven equivalence-class representatives are*

$$0, 1, 3, 7, 15, 23, 63, 255, 279, 831, 4080.$$

Each of the remaining 211 *equivalence classes, representing* 64 794 *tables, has an exact rational witness on at most four joint actions. Consequently the four-mode conclusion holds for every* $N \leq 4$*, by adjoining irrelevant coordinates when necessary.*

A.4.1 An explicit four-mode reversal

We give one concrete member of the four-agent classification whose deployed cost rises with the penalty. Its target uses only four joint actions, so the reversal can be verified directly. The same example will rule out a candidate cost structure in the later classification proof.

Representative 127 is

$$c(\mathbf{a}) = \mathbf{1}\{a_3 = 0,\ (a_0, a_1, a_2) \neq (1, 1, 1)\}.$$

Place mass $t/3$ on each of $3, 5, 6$, and mass $1 - t$ on 8. Then

$$\mathscr{F}_c(p(t)) = t - \frac{8t^4}{27}, \qquad D_c = t(1-t)\left(\frac{32t^3}{27} - 1\right). \tag{15}$$

It is positive for $(27/32)^{1/3} < t < 1$. In particular, $t = 39/40$ gives weights $(13, 13, 13, 1)/40$ and response $7683/3200000$. Here $p(t) = (2t/3, 2t/3, 2t/3, 1 - t)$. The explicit witness is also used in the monotone structural argument below.

A.4.2 Safety certificates and complete coverage

To complete the four-agent classification, we need to verify the safe classes and account for every possible cost table. We express the response as a polynomial whose sign can be checked exactly, then list the safe classes and the number of tables they represent. This connects the individual certificates to the complete enumeration.

For nonnegative masses $x_0, \dots, x_{15}$ let

$$T = \sum_a x_a, \qquad L_{i,b} = \sum_{a:a_i=b} x_a, \qquad N_h(x) = \sum_a c(a) \prod_{i=0}^{3} L_{i,a_i}.$$

Direct differentiation yields

$$-D_c(x/T) = \frac{\mathcal{D}_h(x)}{T^5}, \qquad \mathcal{D}_h = T \sum_a x_a c(a) \partial_a N_h - 4N_h \sum_a x_a c(a). \tag{16}$$

Set $y_i = 1 - a_i$. The safe classes and their orbit sizes are:

Code	Representative	Proof	Size
0	0	constant	2
1	$y_0 y_1 y_2 y_3$	conjunction	32
3	$y_1 y_2 y_3$	conjunction	64
7	$y_2 y_3 (y_0 \vee y_1)$	Theorem A.12	192
15	$y_2 y_3$	conjunction	48
23	$y_3 \mathbf{1}\{y_0 + y_1 + y_2 \geq 2\}$	Theorem A.12	128
63	$y_3 (y_1 \vee y_2)$	restriction of code 7	192
255	y_3	conjunction	8
279	$\mathbf{1}\{\sum_i y_i \geq 3\}$	Lemma A.3	32
831	$\mathbf{1}\{y_1 + y_2 + y_3 \geq 2\}$	Lemma A.3	32
4080	a_2 XOR a_3	binary identity	12

Proof of Theorem A.4. For a conjunction of literals, write t for its joint probability and p_i for the required literal marginals. Each derivative is $p_i' = -t(1 - p_i) \leq 0$, hence the product cost is nonincreasing. Lemma A.3 proves the cardinality entries. The two-variable XOR response has the exact numerator

$$(x_1 + x_2)(x_0 - x_3)^2 + (x_0 + x_3)(x_1 - x_2)^2 \geq 0,$$

with denominator T^3 and a minus sign as in (16). Irrelevant coordinates can be marginalized. Code 63 is a restriction of the code-7 function, followed by a permutation and a dummy coordinate; restriction preserves safety by the embedding argument in Section A.2.

Codes 7 and 23 are monotone duals of $G_{H,I}$ with $(|H|, |I|) = (2, 2)$ and $(1, 3)$, respectively. Their safety therefore also follows from Theorem A.12, whose covariance proof is independent of this finite classification.

For each of the 211 unsafe representatives, an exact finite check verifies a rational law supported on at most four actions, with integer masses having a common denominator at most 21. Direct evaluation of (14) gives a strictly positive response in every case. The code-127 construction above is one explicit example. Exact enumeration of the relabeling group gives 222 disjoint orbits covering all 65 536 tables. The eleven safe orbit sizes sum to 742. Consequently no table is omitted and no numerical non-detection is used to classify a table. Details of the finite verification contract are in Section A.10. □

A.5 Sharpness: three modes do not suffice

The four-action witness bound cannot be reduced to three. We prove this with a three-agent cost that reverses for a four-point law but has no reversal on any three-point support.

Number a three-bit action by $a = a_0 + 2a_1 + 4a_2$. Consider

$$c = (0, 1, 1, 0, 0, 0, 0, 0), \qquad \mathscr{F}_c(p) = (1 - p_2)(p_0 + p_1 - 2p_0 p_1).$$

Proposition A.5 (Three modes do not suffice)**.** *Every law on at most three joint actions has $D_c \leq 0$ for this cost. However,*

$$q = \frac{1}{20}(4, 1, 0, 9, 0, 0, 6, 0) \quad \textit{satisfies} \quad D_c(q) = \frac{1}{800} > 0.$$

Thus any universal support constant is at least four.

Proof. For the displayed law,

$$p = (1/2, 3/4, 3/10), \quad t = 1/20, \quad \nabla \mathscr{F}_c = (-7/20, 0, -1/2), \quad k = (1/40, -3/80, -3/200).$$

Equation (14) gives $1/800$. To exclude every three-mode law, assign unnormalized nonnegative masses x, y, z to a support triple, with $T = x + y + z > 0$. If $\mathcal{N}$ is the degree-three product-cost numerator and C the total costly mass, then

$$D_c(q) = -\frac{\mathcal{D}(x, y, z)}{T^4}, \qquad \mathcal{D} = T \sum_{a:c(a)=1} x_a \partial_a \mathcal{N} - 3\mathcal{N}C.$$

The twelve nonzero support-orbit identities listed below express $\mathcal{D}$ as sums of nonnegative terms. The other twenty triples contain no costly action and give zero. These identities cover all $\binom{8}{3} = 56$ triples; zero masses cover smaller supports. □

A.5.1 THE COMPLETE THREE-SUPPORT EXCLUSION

The sharpness result requires ruling out every target supported on three joint actions for the chosen cost. The table below gives the polynomial certificates for this check, grouping equivalent triples by symmetry. Together with the zero-response cases, these certificates cover all triples and complete the verification of Proposition A.5.

For Proposition A.5, swap coordinates zero and one and/or flip both. These four symmetries preserve the cost and give the following twelve nonzero triple orbits. Each polynomial uses masses x, y, z in the listed action order and $T = x + y + z$.

Triple	Nonnegative numerator $\mathcal{D}$
$(0, 1, 2)$	$xT[x(y+z) + (y-z)^2]$
$(0, 1, 3)$	$yT[(x-z)^2 + y(x+z)]$
$(0, 1, 4)$	$yT[x^2 + xy + xz + 2yz]$
$(0, 1, 5)$	$yT[x^2 + xy + yz + z^2]$
$(0, 1, 6)$	$y[x(x-z)^2 + z(x+y-z)^2 + 2x^2y + xy^2 + 2y^2z]$
$(0, 1, 7)$	$y[x(x-z)^2 + 2xz^2 + 2x^2y + xy^2 + 2y^2z + 2yz^2]$
$(1, 2, 4)$	$2z(x+y)[(x-y)^2 + xy + z(x+y)]$
$(1, 2, 5)$	$z(x+y)[(x-y)^2 + (z-y)^2 + y^2 + 2xz]$
$(1, 4, 5)$	$xT[2xy + xz + yz + z^2]$
$(1, 4, 6)$	$x[z(z-x)^2 + 2x^2z + 2x^2y + 2xy^2 + y^2z + 2yz^2]$
$(1, 4, 7)$	$2x[x^2(y+z) + x(y^2+z^2) + yz(y+z)]$
$(1, 5, 6)$	$x[z(z-x)^2 + 2x^2z + x^2y + 2xy^2 + 4xyz + y^3 + y^2z + yz^2]$

Their orbit sizes are $2, 2, 4, 4, 4, 4, 2, 2, 4, 4, 2, 2$, totaling 36. The other 20 triples contain neither costly action and have zero numerator. Exact expansion checks every identity and the disjoint coverage of all 56 triples.

A.6 THE ALL-DIMENSIONAL NONUNATE ARGUMENT

We now consider costs for which an agent's riskier action can change depending on what its teammates do. These costs are called nonunate. By comparing two contexts in which the same action change has opposite effects, we identify small counterexamples to a nonincreasing penalty response. This extends the finite classification to the nonunate case for any team size.

A cost is *unate* if fixed input-bit flips make it nondecreasing in every coordinate. A nonunate coordinate i has two contexts u, v with opposite edge differences:

$$c(1, u) - c(0, u) = 1, \qquad c(1, v) - c(0, v) = -1.$$

For this coordinate choose an opposing pair at minimum Hamming distance d. Fix the coordinates where the contexts agree and flip the others so they become 0^d and 1^d. Every intermediate context has zero i-edge difference: otherwise there would be a closer opposing pair.

Theorem A.6 (Separated opposing edges admit three modes)**.** *If some nonunate coordinate has minimum opposing-context distance $d \in \{3, 5, 7\} \cup \{8, 9, \ldots\}$, then the cost has a strictly positive response under a law on at most three joint actions.*

Proof. The four block corners $\mathbf{a}_{00} = (0, 0^d)$, $\mathbf{a}_{10} = (1, 0^d)$, $\mathbf{a}_{01} = (0, 1^d)$, $\mathbf{a}_{11} = (1, 1^d)$ have costs $0, 1, 1, 0$. At marginal vector $(r, s, \ldots, s)$ the actual product extractor gives

$$f(r, s) = J(s) + (r - 1/2)H(s), \qquad H(s) = (1 - s)^d - s^d, \tag{17}$$

where

$$J(s) = \tfrac{1}{2}[(1 - s)^d + s^d] + \sum_{y \neq 0^d, 1^d} c(0, y) s^{|y|} (1 - s)^{d - |y|}.$$

The d context coordinates still act independently; they are not treated as a single agent in this formula.

For $0 < s < 1/2$, start with the constant-cost laws

$$q^{(0)} = (1 - s)\delta_{\mathbf{a}_{00}} + s\delta_{\mathbf{a}_{11}}, \qquad q^{(1)} = (1 - s)\delta_{\mathbf{a}_{10}} + s\delta_{\mathbf{a}_{01}}.$$

Contaminate the first by $\varepsilon\delta_{\mathbf{a}_{10}}$ and the second by $\varepsilon\delta_{\mathbf{a}_{00}}$, multiplying the original mass by $1 - \varepsilon$. Their responses are $-\varepsilon B_0 + O(\varepsilon^2)$ and $-\varepsilon B_1 + O(\varepsilon^2)$, respectively, with

$$B_0 = (1 - s)H - sJ' + s(1/2 - s)H', \qquad B_1 = (1 - s)H + sJ' + s(1/2 - s)H'.$$

These expressions follow by inserting the conditional means in (14). At $s = 1/3$,

$$B_0 + B_1 = \frac{4(2^d - 1) - d(2^{d-1} + 1)}{3^{d+1}} = -\frac{(d - 8)2^{d-1} + d + 4}{3^{d+1}} < 0 \quad (d \geq 8).$$

One contamination consequently has a positive response for sufficiently small positive ε. It uses at most three atoms.

For $d = 3, 5, 7$, suppose all three-atom laws were safe. Contaminate $q^{(0)}$ separately by $\mathbf{a}_{10}$ and $\mathbf{a}_{01}$, now for all $0 < s < 1$. Safety requires

$$C_{10} = (1 - s)H - s[J' + (s - 1/2)H'] \geq 0, \qquad C_{01} = -sH + (1 - s)[J' + (s - 1/2)H'] \geq 0.$$

At $s = 1/2$, $H = 0$, so $J'(1/2) = 0$. Then C_{10} has an interior zero; its derivative must vanish, giving $J''(1/2) = 0$. Let k_ℓ count the intermediate contexts of weight ℓ with cost one. The two identities imply

$$\sum_{\ell=1}^{d-1} (2\ell - d)k_\ell = 0,$$
$$\sum_{\ell=1}^{d-1} [(2\ell - d)^2 - d]k_\ell = -d(d - 1).$$

For odd d, the first equation forces $\sum_\ell k_\ell$ even. For $d = 3, 7$, every coefficient in the second sum is 2 modulo 4; its left side is therefore 0 modulo 4, whereas the right side is 2. For $d = 5$, those coefficients are 4 modulo 8, giving 0 on the left and 4 on the right. Each case is impossible. Thus some three-atom contamination has a strict reversal. □

Proposition A.7 (Propagation of an XOR face)**.** *If every three-coordinate restriction of a Boolean cost is universally safe and one two-coordinate face is XOR or XNOR, the entire cost is that same two-coordinate XOR or XNOR, independent of every other coordinate.*

Proof. Theorem C.1 states that a three-variable Boolean cost is universally safe exactly when it is unate or depends on at most two coordinates. Any three-cube adjoining the XOR face is nonunate and already depends on the two face coordinates. Safety therefore forces the third coordinate to be irrelevant, so the parallel face is identical. Induction along the connected cube of outside contexts propagates the same two-coordinate table to every context. Its universal safety follows from the binary identity in Section A.4. □

Lemma A.8 (The one-edge profiles in the finite base)**.** *Let a safe four-variable cost have distinguished edge difference $+1$ only at context 000, and zero at the other seven contexts. On those seven contexts its common endpoint label is a constant $b \in \{0, 1\}$, except possibly at one singleton context.*

Proof. This follows by inspecting the eleven safe classes in Theorem A.4. Conjunctions and their duals give constant profiles. For code 7, the distinguished coordinate can be one of the two optional literals, yielding one exceptional singleton or its complement. An essential coordinate in the other classes has at least two nonzero edges. Equivalently, exact enumeration of the 742 safe tables gives precisely these eight normalized profiles: constant zero or one, and either constant with one of the three singleton labels toggled. The accompanying integer checker verifies this local consequence independently. □

Theorem A.9 (All nonunate costs)**.** *A nonunate Boolean cost has a reversal on at most four joint actions, unless it is XOR or XNOR on two coordinates and ignores all other coordinates.*

Proof. The result for $N \le 4$ follows from the finite classification. For larger N, an unsafe restriction to at most four coordinates already supplies four modes by Theorem A.4, embedded by fixing the other coordinates. Assume all such restrictions are safe. Select a nonunate coordinate and a closest opposing pair at distance d, as above.

If $d = 1$, an XOR/XNOR face exists. Proposition A.7 makes the whole function that safe two-variable cost. If $d = 2$, the restricted three-cube is nonunate and depends on all three coordinates: otherwise a context coordinate could be removed from the opposing pair, contradicting minimality. Theorem C.1 then supplies a reversal, and Theorem A.4, with an irrelevant fourth coordinate, supplies four atoms. Theorem A.6 handles $d = 3, 5, 7$ and every $d \ge 8$. It remains to handle $d = 4, 6$.

Use the frame and intermediate labels $h(y) = c(0, y) = c(1, y)$ of (17). For every three-element set of context coordinates, fix all remaining context coordinates to zero. Its four-variable face has a single distinguished nonzero edge. Lemma A.8 gives one baseline label on every pair and triple in that face. Two such sets sharing a pair must have the same baseline. The graph of three-element subsets connected by one-element replacements is connected, so there is a common baseline b on all contexts of weight two or three, with at most one exception among the singletons.

Apply the same argument at the opposite corner, complementing the distinguished coordinate and all context coordinates. A baseline b' then holds on weights $d-2, d-3$, with at most one exception at weight $d-1$. At $d = 4$, the weight-two layer forces $b = b'$; the opposite weight-one and weight-three conditions eliminate both possible exceptions. Thus every intermediate label is b. At $d = 6$, the common weight-three layer gives $b = b'$, so all intermediate labels equal b except possibly one singleton and one cosingleton.

If all three-atom laws were safe, the midpoint argument in the proof of Theorem A.6 would require

$$\sum_{\ell=1}^{d-1} [(2\ell - d)^2 - d] k_\ell = -d(d-1). \tag{18}$$

For $d = 4$, the two constant profiles give left sides 0 and -24, neither the required -12. For $d = 6$, a singleton or cosingleton has coefficient 10. The possible left sides are $10e$ or $-60-10e$, $e \in \{0, 1, 2\}$, never the required -30. The constant-one values use

$$\sum_{\ell=1}^{d-1} [(2\ell - d)^2 - d] \binom{d}{\ell} = -2d(d-1),$$

which follows by including both endpoints and taking the variance of a $\mathrm{Bin}(d, 1/2)$ variable. Thus a three-atom reversal exists in both remaining cases. This proves the theorem in every dimension. □

A.7 The all-dimensional monotone normal form

After the nonunate case, it remains to consider costs that become nondecreasing in every agent's action after relabeling zero and one. We show that requiring every small restriction to avoid a penalty reversal leaves only the candidate families below. The argument explains why adding more agents does not create new universally monotone families.

For disjoint variable sets H, I, define

$$G_{H,I}(x) = \mathbf{1}\{\exists h \in H : x_h = 1 \text{ or } \sum_{i \in I} x_i \ge 2\}. \tag{19}$$

Its monotone dual is

$$G^*_{H,I}(x) = 1 - G_{H,I}(1-x) = \left(\prod_{h\in H} x_h\right) \mathbf{1}\{\sum_{i\in I} x_i \geq |I| - 1\}.$$

Allow empty sets, constants, and irrelevant variables. Let $\mathcal{C}$ consist of cardinality thresholds and these two families. Write $T_{A,k}$ for the function $x \mapsto \mathbf{1}\{\sum_{i\in A} x_i \geq k\}$. Put $P(a,b,c) = a \vee bc$ and $Q(a,b,c) = a(b \vee c)$. A P or Q restriction here permits variable permutations, not input flips.

Lemma A.10 (Avoiding the two asymmetric three-variable forms)**.** *A monotone Boolean function with no P or Q restriction is a cardinality threshold on its relevant coordinates, or a constant.*

Proof. We first prove a cofactor fact. Write $f_0 = f|_{z=0} \leq f_1 = f|_{z=1}$. If f_0 depends on j but f_1 ignores j, a pivotal j-edge of f_0 gives a (z, j) face equal to $z \vee j$. Decrease an outside context bit i from one to zero. The new face is pointwise below OR, so it is $0, z, j, zj$, or $z \vee j$. The first three possibilities give, respectively, $i(z \vee j)$, $z \vee ij$, and $j \vee iz$, forbidden Q, P, P restrictions. The fourth contradicts the global independence of f_1 from j. Thus the face remains OR. Descending all outside context bits to zero yields $f_1(0) = 1$, so $f_1 \equiv 1$ by monotonicity. Applying this to the monotone dual gives the converse statement: if f_1 depends on j but f_0 ignores j, then $f_0 \equiv 0$.

Induct on the number of variables. Both cofactors satisfy the same forbidden-restriction condition. If both are nonconstant, the preceding fact makes their relevant sets equal, say A. By induction write $f_0 = T_{A,k}$ and $f_1 = T_{A,l}$ with $l \leq k$. If $k \geq l + 2$, leave two coordinates free, fix $l - 1$ others to one and the rest to zero. The lower cofactor becomes zero and the upper becomes OR, giving Q after adjoining z. Hence $k = l$ or $k = l + 1$, yielding an irrelevant selector or the threshold $T_{A\cup\{z\},k}$.

If $f_0 = 0$ and $f_1 = T_{A,l}$, then $l < |A|$ similarly gives Q; otherwise f is a conjunction. If $f_1 = 1$ and $f_0 = T_{A,k}$, then $k \geq 2$ permits an AND-two restriction and gives P; otherwise f is a disjunction. Constant/constant cofactors yield a constant or the selector. The base cases of at most two variables are immediate, completing the induction. □

Theorem A.11 (Safe four-faces force a global normal form)**.** *If every restriction of a monotone Boolean function to at most four coordinates is universally safe, the function belongs to $\mathcal{C}$.*

Proof. A finite consequence of Theorem A.4 is the following: a safe monotone four-variable function with an upper P face has a lower P face. The exact local checker tests all 78 monotone tables among the 742 safe tables, including all permutations of P. It verifies the stronger fact that the same-coordinate lower face equals the upper P. This is also obtained by direct inspection of the safe-class list.

Suppose f has a P restriction at an outside context containing a one. Free that outside coordinate; in the resulting safe four-face replace the upper P face by a lower P face. The number of outside ones decreases by one. Repetition gives a bottom P face. Its singleton true input e_a is true in the full function, so a is a *global passer*: $x_a = 1$ forces $f(x) = 1$.

If a minimal true set B had size at least three, it could not contain a. Retain a and three members b, c, d of B, fix the other members of B to one, and all remaining variables to zero. Minimality makes the $a = 0$ cofactor exactly bcd, while the passer makes the $a = 1$ cofactor one. The resulting function $a \vee bcd$ is unsafe (class 127), a contradiction. Thus every minimal true set has size one or two.

Let H be the nonempty set of singleton minimal true sets. The remaining quadratic minimal true sets are the edges of a graph on the other variables. An induced three-vertex path, together with a passer, gives $a \vee b(c \vee d)$, unsafe class 31. Therefore each graph component is a clique. Two edge-containing components would give $ab \vee cd$, unsafe class 855, after fixing the passers to zero. Hence there is at most one nontrivial clique, and the other nonpasser variables are irrelevant. This is exactly $G_{H,I}$.

If f contains Q, its monotone dual contains P and satisfies the same hypothesis, so $f = G^*_{H,I}$. If neither form occurs, Lemma A.10 gives a cardinality threshold. These alternatives prove the theorem. □

A.8 UNIVERSAL SAFETY OF THE HARD/SOFT FAMILY

The preceding argument identifies a candidate family that charges a cost if any agent in one group acts or at least two agents in another group act. We now prove that increasing the KL penalty never raises its deployed cost, for any target distribution. This establishes the sufficiency part of the classification for this family and its dual.

Theorem A.12 (Safety of hard disjunctions with a soft pair threshold)**.** *For every finite disjoint H, I and every joint law q, $D_{G_{H,I}}(q) \le 0$. The same holds for $G^*_{H,I}$.*

Proof. By continuity consider interior marginals and $0 < t = \mathbb{E}_q c < 1$. Put $\ell = 1 - t$. If fewer than two soft variables occur, the result is the ordinary disjunction case. Otherwise define soft odds and scores

$$r_i = \frac{p_i}{1-p_i}, \quad R = \sum_{i\in I} r_i, \quad W = 2\sum_{i<j} r_i r_j, \quad B = 1 + R, \quad \alpha_i = (1+r_i)(R - r_i),$$

and let $Z = \prod_{h\in H}(1-p_h)\prod_{i\in I}(1-p_i) > 0$. Since $1 - \mathscr{F}_c = Z(1+R)$,

$$\partial_i \mathscr{F}_c = Z\alpha_i \quad (i \in I), \qquad \partial_h \mathscr{F}_c = ZB/(1-p_h) \quad (h \in H).$$

Let $z_i = \Pr_q(c = 0, A_i = 1)$ and $v_i = z_i/\ell$. Safe outcomes have every hard bit zero and at most one soft bit one, so $v_i \ge 0$ and $\sum_i v_i \le 1$. Let $H_0(\mathbf{A})$ indicate that some hard bit is one, and put $h = \mathbb{E}_q H_0$. Then

$$\mathrm{Cov}_q(c, A_h) = \ell p_h, \qquad \mathrm{Cov}_q(c, A_i) = \ell p_i - z_i, \qquad \sum_i p_i\alpha_i = W.$$

Using $\sum_h p_h/(1-p_h) \ge \sum_h p_h \ge h$, we obtain

$$\frac{-D_c(q)}{Z} = \ell\left[B\sum_h \frac{p_h}{1-p_h} + W\right] - \sum_i z_i\alpha_i \ge \ell\left[Bh + W - \sum_i v_i\alpha_i\right]. \tag{20}$$

We prove this last bracket nonnegative. Equivalently its product with ℓ is $\mathrm{Cov}_q(c, Y)$ for $Y = BH_0(\mathbf{A}) + \sum_i \alpha_i A_i$. The case split below compares this weighted action score between safe and costly outcomes through their conditional means.

Write $\alpha_{\max} = \max_i \alpha_i$ and choose a maximizer j. If $\alpha_{\max} \le W$, the conclusion follows immediately from $\sum_i v_i \le 1$. For the remaining cases write $s = r_j$ and $T = \sum_{i\ne j} r_i$. The inequality $\alpha_{\max} > W$ gives

$$(1-s)T > 2\sum_{i<k,\ i,k\ne j} r_i r_k, \tag{21}$$

so $s < 1$. Every pair of soft scores has sum at least $\alpha_{\max}$. This is immediate for a pair containing j; otherwise, if its odds are x, y and U_0 is the sum of all remaining odds except s, x, y, expansion gives

$$\alpha_x + \alpha_y - \alpha_j = 2s + 2xy + U_0(1 + x + y - s) \ge 0.$$

Here the subscripts x, y name the variables with those odds. If $W < \alpha_{\max} \le B$, every safe outcome has $Y \le \alpha_{\max}$, and every costly outcome has $Y \ge \alpha_{\max}$, by either its hard bit or two soft bits. Thus $\mathrm{Cov}_q(c, Y) \ge 0$.

It remains that $\alpha_{\max} > \max(W, B)$. Choose $k \ne j$ with largest odds $r = r_k$ among indices other than j, and put

$$S = R - r, \qquad U = S - s, \qquad \rho = r - 1.$$

The pair sum in (21) is at least $T(T-r)/2$, so $S < 1$. Moreover $\alpha_{\max} > B$ says $s(\rho + U) > 1$. Since $S = s + U < 1$, this implies

$$\rho > 1, \qquad 0 < S < 1, \qquad \rho S > 1. \tag{22}$$

Indeed $\rho > 1/s - U > 1/s - (1-s) \ge 1$, and $\rho S > 1 + U(\rho - s) \ge 1$.

Define

$$M = \sum_{i\ne k} p_i \le S, \quad \delta = 1 - p_k = \frac{1}{1+r}, \quad \beta = \sum_{i\ne k} v_i, \quad z = \ell\beta, \quad L = \ell - z \ge 0.$$

Safe outcomes counted by z have a soft bit other than k equal to one, hence have $A_k = 0$; therefore $z \le \delta$. A costly outcome without a hard bit has at least one such other soft bit, so $t \le h + M - z$, and consequently

$$h \ge \max(0, 1 - M - L), \qquad \beta = \frac{z}{L+z} \le \frac{\delta}{L+\delta}.$$

Since $\sum_i v_i \alpha_i \le \beta \alpha_{\max} + (1-\beta)\alpha_k$, write

$$C = W - \alpha_k = \rho S + 2 \sum_{i<j,\ i,j\neq k} r_i r_j \ge \rho S, \qquad \Delta = \alpha_{\max} - \alpha_k = (\rho + 1 - s)(1 - U) > 0.$$

The bracket in (20) is therefore bounded below by

$$g(L) = B \max(0, 1 - M - L) + C - \frac{\delta \Delta}{L + \delta}. \tag{23}$$

This function is concave on $[0, 1 - M]$, with $g'' = -2\delta\Delta/(L+\delta)^3 < 0$, and increasing thereafter. Its minimum on $L \ge 0$ occurs at 0 or $1 - M$.

At the first endpoint, $M \le S$ and the pair sum excluding k is at least sU, so direct expansion gives

$$g(0) = B(1-M) + W - \alpha_{\max} \ge B(1-S) + 2rS + 2sU - (1+s)(r+U) = 1 - s^2 + U(\rho - S) > 0.$$

At the second endpoint, it suffices to prove

$$f(\rho) := \rho S[(\rho + 2)(1 - S) + 1] - (\rho + 1 - s)(1 - U) \ge 0,$$

because $C \ge \rho S$ and $1 - M \ge 1 - S$. Using $S = s + U$, exact simplification gives

$$\begin{aligned} f(1/S) &= 1 - S + U(1/S - s) > 0, \\ f'(1/S) &= (1 - S)(1 + 2S) + U > 0, \\ f''(\rho) &= 2S(1 - S) > 0. \end{aligned}$$

Equation (22) has $\rho > 1/S$, proving this endpoint positive as well. Thus (23), then (20), are nonnegative in all cases. This proves safety for $G_{H,I}$. Input complementation and output complementation preserve the response sign, giving $G^*_{H,I}$. Continuity covers boundary laws. □

A.9 Completion of the main theorems

We now combine the preceding results to prove the two structural theorems. The safe-family lemmas and the classification arguments establish exactly which costs never reverse. The finite witnesses and their extensions to larger teams then give the four-action bound, while the earlier sharpness example shows why three actions are insufficient.

Proof of Theorem A.2. All listed functions are safe by Lemma A.3, Theorem A.12, the binary XOR identity, and sign invariance. Conversely a safe nonunate function is XOR/XNOR by Theorem A.9. For a safe unate function, flip input bits to make it monotone. All its restrictions are safe by continuity and witness embedding. Theorem A.11 therefore places it in $\mathcal{C}$, which is precisely the threshold family and the family (19) together with their monotone duals. □

Proof of Theorem A.1. A sparse law with $D_c > 0$ gives a positive full-support law with the same strict sign by continuity, so it violates universal monotonicity. Conversely, suppose universal monotonicity fails. If the cost is nonunate, Theorem A.9 supplies a four-mode reversal unless it is universally safe; the latter contradicts the premise. For a unate cost, flip its inputs to make it monotone. If every restriction to at most four variables were safe, Theorem A.11 would place the cost in $\mathcal{C}$, and Lemma A.3 and Theorem A.12 would make it universally safe, again a contradiction. Hence some restriction on at most four variables is unsafe. Theorem A.4 gives an exact four-mode witness there; fixing the other coordinates embeds it into the original cube. Undo the input flips. Proposition A.5 proves that the bound four is optimal. □

A.10 EXACT VERIFICATION OF THE FINITE WITNESSES

This subsection explains how the finite counterexamples are verified with exact arithmetic. We enumerate all four-variable cost tables, group equivalent tables, and check a rational target with positive deployed-cost response for every class that admits a reversal. The integer formulas below make the sign check independent of floating-point rounding.

For a candidate law with support u_j, integer masses m_j and common denominator M, put $\ell_i = \sum_j m_j(u_j)_i$, $r = \sum_j m_j c(u_j)$ and $k_i = M\sum_j m_j c(u_j)(u_j)_i - r\ell_i$. The integer pivotal numerator is

$$w_i = \sum_{\mathbf{a}_{-i}\in\{0,1\}^3} [c(1,\mathbf{a}_{-i}) - c(0,\mathbf{a}_{-i})] \prod_{j\neq i} \ell_j^{a_j}(M-\ell_j)^{1-a_j}, \qquad D_c(q) = -\frac{\sum_i w_i k_i}{M^5}.$$

The sum inserts the displayed bit in coordinate i. Integer evaluation checks the strictly positive response, distinct support indices, at most four atoms, and masses summing to $M \le 21$. A bounded search over these supports and positive integer masses includes a witness for every unsafe representative. The canonical orbits are disjoint and cover all 65,536 tables; the safe orbit sizes sum to 742. The eight one-edge profiles and downward-P property are finite consequences of that classification. The structural and covariance arguments above establish the conclusion for arbitrary N.

A.11 THE FOUR-AGENT HAZARD EXAMPLE

This explicit cost shows why monotonicity in each agent's action does not suffice for a safe multiplier response. The construction also supplies the safety-switch family used in the experiments.

Proposition A.13 (A four-agent boundary)**.** *The monotone cost* $c(\mathbf{a}) = \mathbf{1}\{a_1 = a_2 = 1 \text{ or } a_3 = a_4 = 1\}$ *admits a positive target law for which the KL multiplier increases deployed cost.*

Proof of Proposition A.13. Put mass t on 1100 and $(1-t)/2$ on each of $0010, 0001$. The corrected cost is t, so $t' = -t(1-t)$ under (11). Independent extraction gives

$$F_c(t) = t^2 + \frac{(1-t)^2}{4} - \frac{t^2(1-t)^2}{4}. \tag{24}$$

At $t = 1/8$, the multiplier derivative is $819/32768 > 0$. A sufficiently small uniform perturbation gives a positive target law. □

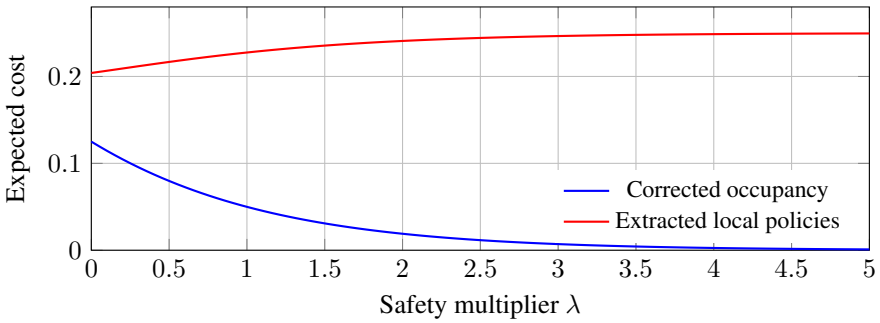


Figure 4: Exact cost curves for Proposition A.13, starting at $t = 1/8$. The multiplier lowers corrected cost and raises deployed cost. The curves evaluate (24) analytically.

B THE TWO-AGENT REAL-VALUED CRITERION

For two agents, a direct criterion is available without restricting costs to Boolean values. It identifies which interaction terms can reverse the deployed response for some target law.

Theorem B.1 (Two binary agents)**.** *Put $d = c_{11} - c_{00}$, $e = c_{10} - c_{01}$, and $z = c_{00} + c_{11} - c_{01} - c_{10}$. Universal monotonicity holds if and only if*

$$|dz| \le d^2, \qquad |ez| \le e^2. \tag{25}$$

Equivalently, whenever opposite entries differ, the sum of the other two entries lies between twice the smaller and twice the larger entry. Every binary-valued two-agent cost passes. General real-valued costs can fail: $c = (0, 1, 1, 0.1)$ and $q = (0.100, 0.001, 0.001, 0.898)$ have a positive deployed-cost derivative.

Let $x = (x_0, x_1, x_2, x_3)$ be unnormalized masses, $T = \sum_i x_i$, and write $q = x/T$. If $N(x)$ is the quadratic numerator of $F_c(q)$, then

$$-\left.\frac{dF_c(q_\lambda)}{d\lambda}\right|_0 = \frac{D(x)}{T^3}, \quad D = T\sum_i x_i c_i \partial_i N - 2N\sum_i x_i c_i.$$

An exact expansion is

$$\begin{aligned} D = {} & (c_0 - c_1)^2 x_0 x_1 (x_0 + x_1) + (c_0 - c_2)^2 x_0 x_2 (x_0 + x_2) \\ & + (c_1 - c_3)^2 x_1 x_3 (x_1 + x_3) + (c_2 - c_3)^2 x_2 x_3 (x_2 + x_3) \\ & + d(d - z) x_0^2 x_3 + d(d + z) x_0 x_3^2 + e(e + z) x_1^2 x_2 + e(e - z) x_1 x_2^2 \\ & + \tfrac{1}{2}(d^2 + 3e^2 - z^2) x_1 x_2 (x_0 + x_3) + \tfrac{1}{2}(3d^2 + e^2 - z^2) x_0 x_3 (x_1 + x_2). \end{aligned}$$

If (25) holds and $(d, e) \neq (0, 0)$, all coefficients are nonnegative. Indeed $|z|$ is at most the absolute value of each nonzero member of d, e; this also controls the last two coefficients. If $d = e = 0$, the same polynomial has the alternative form

$$D = \frac{z^2}{4}\{(x_1 + x_2)(x_0 - x_3)^2 + (x_0 + x_3)(x_1 - x_2)^2\} \ge 0.$$

This proves sufficiency, at any multiplier since q_λ is another positive law. For necessity, restrict first to a mixture of 00 and 11. The endpoint derivatives of its product-projected cost give $d(d - z) \ge 0$ and $d(d + z) \ge 0$. The mixture of 01 and 10 similarly gives $e(e + z) \ge 0$ and $e(e - z) \ge 0$. A strict violation at a boundary law remains a violation after a sufficiently small positive perturbation. These four inequalities are exactly (25).

C THE THREE-AGENT BOOLEAN SPECIALIZATION

For three agents, the general classification has a particularly simple form: a binary cost never reverses exactly when it is unate or depends on at most two agents. Here unate means that fixed swaps of zero and one make the cost nondecreasing in every coordinate. We derive this specialization from the four-agent classification and give a concrete example of a cost that fails it.

Theorem C.1 (Three-agent Boolean costs)**.** *For three binary agents and KL correction, a Boolean cost is universally monotone if and only if it is unate or depends on at most two agents. Exactly 110 of the 256 cost tables satisfy this condition; the other 146 admit full-support rational counterexamples.*

Proof. Add an irrelevant fourth coordinate. Marginalization commutes with the KL tilt, so the extended cost has the same response sign as the three-agent cost. By Theorem A.4, every safe extended table is unate or is two-coordinate XOR up to relabeling. Restricting the dummy coordinate therefore gives the necessary condition.

Conversely, every monotone three-variable cost is, up to input permutations and monotone duality, a constant, a literal, a conjunction, majority, or $a_1 \vee (a_2 a_3)$. These are threshold or hard/soft forms, whose safety is established independently in Lemma A.3 and Theorem A.12. Input flips preserve the sign, and costs depending on at most two variables are safe by Theorem B.1. Counting relabelings gives 104 unate tables. The six XOR/XNOR tables on a pair of coordinates give 110 safe tables in total. Each remaining table inherits a rational witness from the four-variable base; projection onto the relevant coordinates and a small rational uniform perturbation give a full-support three-agent counterexample. □

D PROOFS FOR THE RESPONSE AND CORRECTION

This section supplies the derivatives and descent arguments used by the main method. We first derive the response, then examine a protected product-target case and prove the local gradient and correction bounds.

D.1 SMOOTH MULTIPLIER RESPONSE

We differentiate the constrained optimum on a fixed positive support. The calculation yields the flow-preserving response matrix and the separate derivatives of corrected and deployed cost. This establishes the response formula used to compare the training target with the extracted policy.

The regularizer's Hessian H describes its curvature, W is its inverse, and M removes changes that violate Bellman flow:

$$H = \nabla^2\Phi(q) = \alpha \operatorname{diag}\left(\frac{f''(q_x/b_x)}{b_x}\right), \quad W = H^{-1}, \quad M = W - WA^\top(AWA^\top)^{-1}AW. \tag{26}$$

The matrix M is positive semidefinite and $AM = 0$. The quantity $\|u\|_M^2 = u^\top Mu$ defines a seminorm because M has a nullspace.

Proof of Theorem 4.1. On the full finite model the rows of A are independent. Indeed, $A^\top v = 0$ implies $v(\mathbf{s}) = \gamma\mathbb{E}[v(\mathbf{S}') \mid \mathbf{s}, \mathbf{a}]$ for every available state/action pair. Taking the largest absolute value gives $\|v\|_\infty \le \gamma\|v\|_\infty$, hence $v = 0$. On a restricted support, use the assumed independent flow rows. The positive diagonal matrix H therefore makes $AWA^\top$ invertible. At an interior optimum the KKT equations are

$$\nabla\Phi(q) + \sum_k \lambda_k c_k - r + A^\top\nu = 0, \qquad Aq = b_0.$$

Differentiating with respect to λ_k gives $Hq' + c_k + A^\top\nu' = 0$ and $Aq' = 0$. Eliminating ν' gives $q' = -Mc_k$. The implicit function theorem applies because the KKT matrix is nonsingular: its null equations $Hu + A^\top v = 0$, $Au = 0$ give $u^\top Hu = 0$, hence $u = 0$ and then $v = 0$.

Write

$$M = W^{1/2}(I - \mathsf{P})W^{1/2}, \qquad \mathsf{P} = W^{1/2}A^\top(AWA^\top)^{-1}AW^{1/2}.$$

Here P is an orthogonal projection. Thus $M \succeq 0$ and $AM = 0$. The cost derivatives follow by the chain rule. The statement is local to the positive solution, so it uses the curvature at the actual ratios rather than a global bound on $f^{*\prime\prime}$. □

For one state, let $\mathbf{A} = (A_i)_i \sim q$, write q_i for the action marginal of agent i, and $\Pi q = \prod_i q_i$ for the extracted product law. Define $h_{\mathbf{a}} = b_{\mathbf{a}}/[\alpha f''(q_{\mathbf{a}}/b_{\mathbf{a}})]$, $Z_h = \sum_{\mathbf{a}} h_{\mathbf{a}}$, and $\bar{h} = h/Z_h$. Let

$$U_c^q(\mathbf{a}) = \sum_i \mathbb{E}_{\mathbf{A}_{-i}\sim\prod_{j\neq i} q_j} c(a_i, \mathbf{A}_{-i}). \tag{27}$$

The quantity $U_c^q(\mathbf{a})$ adds the counterfactual cost of each agent's action while the other agents follow their extracted marginals. It is the derivative of product-policy cost with respect to joint mass, up to a constant. Flow reduces to normalization and $M = \operatorname{diag}(h) - hh^\top/Z_h$, which annihilates constants. Equation (5) becomes

$$\frac{dF_c}{d\lambda} = -Z_h \operatorname{Cov}_{\bar{h}}(c, U_c^q), \qquad \frac{d\mathbb{E}_q c}{d\lambda} = -Z_h \operatorname{Var}_{\bar{h}}(c). \tag{28}$$

KL gives $h = q/\alpha$, Pearson gives $h = b/\alpha$, and squared Hellinger gives $h = 2q^{3/2}/(\alpha\sqrt{b})$. Thus a reversal occurs when the cost and local counterfactual signal have negative covariance under $\bar{h}$. The divergence changes that weighting: squared Hellinger even admits a two-agent reversal with factorized behavior and target laws. Section D.3 and Figure 5 give the exact construction and comparison with KL.

D.2 Product-target protection under KL

When the current target already has independent actions, a small KL penalty increase cannot raise deployed cost. We prove this by writing the derivative as a sum of negative variances. The result is local to the current target: the penalty update can introduce correlations, so the calculation does not establish monotonicity along the entire path.

At a one-state product law $q = \bigotimes_i q_i$, let $m_i(A_i) = \mathbb{E}_q[c(\mathbf{A}) \mid A_i]$. Then $U_c^q(\mathbf{A}) = \sum_i m_i(A_i)$, so (28) for KL gives

$$\frac{dF_c}{d\lambda} = -\alpha^{-1} \sum_i \mathrm{Cov}_q(c, m_i(A_i)) = -\alpha^{-1} \sum_i \mathrm{Var}_q(m_i(A_i)) \le 0.$$

The last equality is conditional expectation. It holds for every real cost at this product target, but the tilted law need not remain a product. Thus this instantaneous identity does not establish monotonicity for arbitrary correlated targets. Classical polynomial growth transformations (Baum & Eagon, 1967; Baum & Sell, 1968) concern improvement on products of simplices; their product-state guarantee does not imply the universal sign classification in Theorem A.2.

D.3 Exact Hellinger construction

The preceding protection for product targets depends on the choice of regularizer. Here we construct a two-agent example with squared Hellinger regularization in which a larger penalty lowers target cost but raises deployed cost, even though the starting target is a product distribution. Exact rational calculations verify both derivatives.

Proposition D.1 (An interior Hellinger reversal)**.** *For two binary agents, in order* $00, 01, 10, 11$*, take*

$$q = \frac{(4,6,6,9)}{25}, \qquad b = \frac{(1024,7776,7776,59049)}{75625}, \qquad c = (0,1,1,0). \tag{29}$$

Both q and b are product laws. With squared Hellinger, $\alpha = 1$, $r = (39,19,19,-26)/55$, and $\lambda = 0$, q is the unique interior optimum and

$$\frac{d\mathbb{E}_q c}{d\lambda} = -\frac{286}{375}, \qquad \frac{dF_c}{d\lambda} = \frac{22}{375} > 0. \tag{30}$$

The largest density ratio is $3025/256 < 12$*.*

Proof of Proposition D.1. The two local action-one probabilities under q are $3/5$, and those under b are $243/275$. This proves that both distributions factorize. For $f(w) = (\sqrt{w}-1)^2$, $f'(w) = 1 - w^{-1/2}$ and $f''(w) = 1/(2w^{3/2})$. The ratios in (29) have square roots $(55/16, 55/36, 55/36, 55/81)$. Consequently $f'(q/b) = (39,19,19,-26)/55 = r$. The strictly convex regularizer then makes q the unique normalized interior optimum at $\lambda = 0$.

The action derivatives are $U_c^q = (6/5, 1, 1, 4/5)$ and the curvature distribution is $\bar{h} = (9,6,6,4)/25$, with $Z_h = 55/18$. Direct rational calculation gives

$$\mathrm{Var}_{\bar{h}}(c) = \frac{156}{625}, \qquad \mathrm{Cov}_{\bar{h}}(c, U_c^q) = -\frac{12}{625}.$$

Multiplication by $-Z_h$ proves (30). The maximum ratio is $3025/256$. □

D.4 The multiplier floor

The limiting KL target concentrates on minimum-cost joint actions. Whether independent extraction can attain their minimum cost depends on the product structure of that minimum set.

For a cost c, write $\mathcal{A}_* = \arg\min_{\mathbf{a}} c(\mathbf{a})$ and $c_{\min} = \min_{\mathbf{a}} c(\mathbf{a})$.

Proposition D.2 (The multiplier floor)**.** *Let $q > 0$, let c be nonconstant, and let q_λ be as in* (11)*. Then $F_c(q_\lambda) > c_{\min}$ for every finite λ, $dF_c(q_\lambda)/d\lambda \to 0$ as $\lambda \to \infty$, and $\inf_{\lambda \ge 0} F_c(q_\lambda) > c_{\min}$ if and only if $\mathcal{A}_*$ is not a product set.*

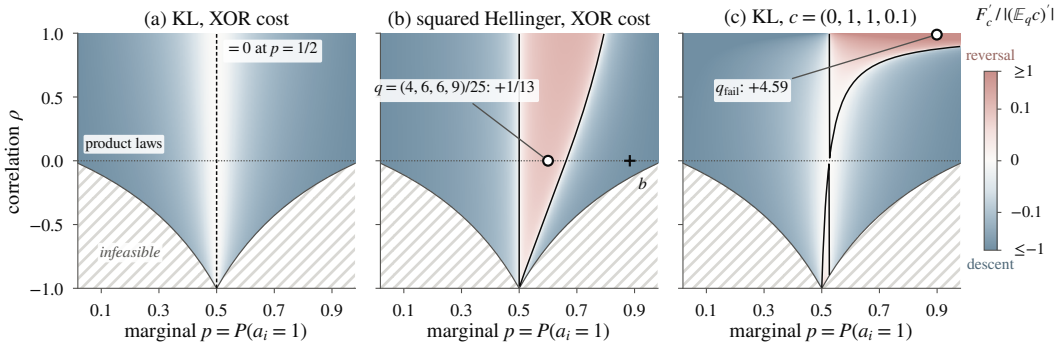


Figure 5: Normalized deployed response $(dF_c/d\lambda)/|d\mathbb{E}_q c/d\lambda|$ from (28) for two binary agents, over targets with equal marginals $p = P(a_i{=}1)$ and correlation ρ (hatched: no full-support law; dotted: product laws). Red: a larger multiplier raises the deployed cost; blue: it lowers it; black: zero level (symmetric-log scale, clipped at ± 1). (a) KL, XOR cost $c = (0, 1, 1, 0)$: nonpositive everywhere, as Theorem B.1 requires. (b) Squared Hellinger, same cost, reference b of Proposition D.1 (+): reversals cover 23% of the slice, including product targets with $\frac{1}{2} < p < 0.663$; the circle is the product-target example in Section D.3. (c) KL, $c = (0, 1, 1, 0.1)$: the circle marks the real-valued example of Section B, with response ratio 4.59.

Proof of Proposition D.2. Because $q > 0$, every q_λ and its product extraction have full support; since c is nonconstant, $F_c(q_\lambda) > c_{\min}$. As $\lambda \to \infty$, $q_\lambda \to q_\infty = q(\cdot \mid \mathcal{A}_*)$, which is positive exactly on $\mathcal{A}_*$. Marginals are continuous, so $F_c(q_\lambda) \to F_c(q_\infty)$. The local support of agent i under q_∞ is the coordinate projection $\mathrm{pr}_i \mathcal{A}_*$, and the extracted law is positive exactly on $\prod_i \mathrm{pr}_i \mathcal{A}_* \supseteq \mathcal{A}_*$. Hence $F_c(q_\infty) = c_{\min}$ if and only if $\mathcal{A}_* = \prod_i \mathrm{pr}_i \mathcal{A}_*$, that is, $\mathcal{A}_*$ is a product set. If $F_c(q_\infty) > c_{\min}$, the continuous positive function $\lambda \mapsto F_c(q_\lambda) - c_{\min}$ has a positive limit and is therefore bounded below by a positive constant on $[0, \infty)$. If $F_c(q_\infty) = c_{\min}$, the infimum is $c_{\min}$. Finally, (28) with $h = q_\lambda$ gives $|dF_c/d\lambda| \le \mathrm{Var}_{q_\lambda}(c)^{1/2} \mathrm{Var}_{q_\lambda}(U_c^{q_\lambda})^{1/2}$, where $|U_c^q| \le N \max_{\mathbf{a}} |c(\mathbf{a})|$ and $\mathrm{Var}_{q_\lambda}(c) \to 0$. □

D.5 Local gradient of deployed cost

The local-gradient formula differentiates through policy extraction and subsequent policy evaluation. The proof separates these two effects before combining them by the chain rule.

Proof of Theorem 5.1. Let $\pi = \pi^q$. Differentiating its Bellman equation gives the normalized discounted policy derivative

$$dF = \sum_{\mathbf{s}} d_{\pi,S}(\mathbf{s}) \sum_{\mathbf{a}} Q_c^\pi(\mathbf{s}, \mathbf{a})\, d\pi(\mathbf{a} \mid \mathbf{s}).$$

Differentiate the product policy one factor at a time. The conditional marginal in (3) satisfies

$$d\pi_i(a_i \mid o_i) = \frac{dq_{O_i,A_i}(o_i, a_i) - \pi_i(a_i \mid o_i) dq_{O_i}(o_i)}{q_{O_i}(o_i)}.$$

For fixed (o_i, a_i), summing over global states with observation o_i and over the other agents' actions produces

$$H_i(o_i, a_i) = \sum_{\mathbf{s}: O_i(\mathbf{s}) = o_i} d_{\pi,S}(\mathbf{s}) \mathbb{E}_{\pi_{-i}(\cdot \mid \mathbf{s})} Q_c^\pi(\mathbf{s}, a_i, \mathbf{A}_{-i}).$$

The coefficient of $dq(\mathbf{s}, \mathbf{a})$ is

$$\sum_i \frac{H_i(O_i(\mathbf{s}), a_i) - \sum_{u_i} \pi_i(u_i \mid O_i(\mathbf{s})) H_i(O_i(\mathbf{s}), u_i)}{q_{O_i}(O_i(\mathbf{s}))}.$$

The subtracted term equals $\sum_{\mathbf{s}': O_i(\mathbf{s}') = O_i(\mathbf{s})} d_{\pi,S}(\mathbf{s}') V_c^\pi(\mathbf{s}')$. Factoring out d_{π,O_i} gives (7). If this deployed marginal is zero, its entire contribution is zero. □

D.6 Gradient and projection errors

An implemented DCP step may use both an estimated gradient and an approximate flow projection. We bound how these errors affect the change in deployed cost and give a step-size condition that still ensures a decrease. The result also tracks changes in flow feasibility and states the condition needed to preserve total probability.

Theorem D.3 (Finite-step correction)**.** *Assume $\eta \geq 0$, $\|\widehat{g} - g\|_M \leq \epsilon$, $\|g\|_2 \leq G$, and $d = -M\widehat{g} + e$ with $\|e\|_2 \leq \delta$ and $\mathbf{1}^\top e = 0$. Suppose F is L-smooth on the segment from q to $q + \eta d$, which lies in the positive support. Write $\rho = \|\widehat{g}\|_M$. Then*

$$F(q + \eta d) - F(q) \leq -\eta\{\rho(\rho - \epsilon) - G\delta\} + \frac{L\eta^2}{2}\|d\|_2^2. \tag{31}$$

If $m = \rho(\rho - \epsilon) - G\delta > 0$, any positive step satisfying $L\eta\|d\|_2^2 \leq m$ decreases F by at least $\eta m/2$. Its flow residual changes by ηAe; exact projection has $e = 0$. If $d = 0$, the correction leaves q unchanged.

Proof of Theorem D.3. Set $g = \nabla F(q)$ and $\rho = \|\widehat{g}\|_M$. Cauchy–Schwarz for the semidefinite form M gives

$$g^\top(-M\widehat{g}) = -\rho^2 + (\widehat{g} - g)^\top M\widehat{g} \leq -\rho^2 + \epsilon\rho.$$

The projection error satisfies $g^\top e \leq G\delta$. The full flow operator has $\mathbf{1}$ in its row space because $\mathbf{1}^\top A_{\text{full}} = (1-\gamma)\mathbf{1}^\top$. Retaining independent flow rows preserves this row space. Since $AM = 0$, the ideal direction therefore has zero mass. The condition $\mathbf{1}^\top e = 0$ therefore preserves normalization. The smoothness inequality on the stated positive segment gives

$$F(q + \eta d) \leq F(q) + \eta g^\top d + \frac{L\eta^2}{2}\|d\|_2^2,$$

which proves the bound and its stepsize consequence. Finally $Ad = Ae$ because $AM = 0$, so the residual changes by ηAe. □

For exact projection and exact gradients, the continuous update is $\dot{q} = -M\nabla F$ and $\dot{F} = -\|\nabla F\|_M^2$. For a finite step, positivity can be enforced by bounding $\eta < \min_{x:d_x<0} q_x/(-d_x)$. A finite ratio cap adds the upper-face condition $q_x + \eta d_x \leq b_x W_{\text{corr}}$. A zero allowed step returns the current target.

D.7 Actor error at the endpoints

The target-level descent result assumes exact extraction. We bound the additional deployed-cost error from fitting local actors at both the anchor and candidate targets.

Let $q^+ = q + \eta d$, and for $v \in \{q, q^+\}$ define $\tau_v = \sup_{\mathbf{s}} \mathrm{TV}(\widehat{\pi}^v(\cdot \mid \mathbf{s}), \pi^v(\cdot \mid \mathbf{s}))$. The policies are Markov on the same finite model.

Corollary D.4 (Actor error at both endpoints)**.** *Under Theorem D.3 and $0 \leq c \leq c_{\max}$,*

$$\widehat{F}(q^+) - \widehat{F}(q) \leq -\eta m + \frac{L\eta^2}{2}\|d\|_2^2 + \frac{c_{\max}(\tau_q + \tau_{q^+})}{1-\gamma}, \qquad m = \rho(\rho - \epsilon) - G\delta. \tag{32}$$

Proof of Corollary D.4. For two Markov policies $\pi, \widetilde{\pi}$, let $\tau = \sup_{\mathbf{s}} \mathrm{TV}(\pi(\cdot \mid \mathbf{s}), \widetilde{\pi}(\cdot \mid \mathbf{s}))$. Their discounted state occupancies obey

$$\mathrm{TV}(d_{\pi,S}, d_{\widetilde{\pi},S}) \leq \gamma\,\mathrm{TV}(d_\pi, d_{\widetilde{\pi}}), \qquad \mathrm{TV}(d_\pi, d_{\widetilde{\pi}}) \leq \mathrm{TV}(d_{\pi,S}, d_{\widetilde{\pi},S}) + \tau.$$

The first inequality follows from the common initial law and contraction of total variation through P; the second inserts the measure $d_{\pi,S}\widetilde{\pi}$. Thus the full occupancy distance is at most $\tau/(1-\gamma)$. Since $c \in [0, c_{\max}]$, $|(1-\gamma)J_c(\pi) - (1-\gamma)J_c(\widetilde{\pi})| \leq c_{\max}\tau/(1-\gamma)$. Apply this at q and q^+ and insert (31) between the two ideal costs. The product-policy bound follows by telescoping over agents (or independent couplings), and total variation is at most one. □

D.8 MOMENT SADDLE FOR THE FLOW PROJECTION

The projected DCP update can be computed without constructing the matrix M. We express it as a saddle problem involving a state potential and a signed state–action function, with expectations that can be estimated from transitions. The proposition below shows that solving this problem recovers the same direction as the matrix formula.

The projection in (26) does not require forming M. Put $\omega(\mathbf{s}, \mathbf{a}) = 1/[\alpha f''(w(\mathbf{s}, \mathbf{a}))]$, so that $W = \operatorname{diag}(b\omega)$. The following moment saddle uses a state potential v to enforce flow and a signed state–action function z to represent the projected direction. The expectation $\mathbb{E}_{bP}$ draws $(\mathbf{s}, \mathbf{a})$ from b and $\mathbf{s}'$ from $P(\cdot \mid \mathbf{s}, \mathbf{a})$:

$$\min_{v} \max_{z} \; \mathbb{E}_{(\mathbf{s},\mathbf{a},\mathbf{s}')\sim bP} \left[z(\mathbf{s}, \mathbf{a}) \{ \widehat{g}(\mathbf{s}, \mathbf{a}) - v(\mathbf{s}) + \gamma v(\mathbf{s}') \} - \frac{z(\mathbf{s}, \mathbf{a})^2}{2\omega(\mathbf{s}, \mathbf{a})} \right]. \tag{33}$$

Proposition D.5 (Matrix-free correction)**.** *At the unrestricted saddle of* (33)*, $bz = M\widehat{g}$. Therefore the density-ratio step $w^+ = w - \eta z$ equals $q^+ = q - \eta M\widehat{g}$.*

Because the next-state value is linear inside the sampled objective, an observed transition supplies an unbiased sample. The first-order condition in v is $A(bz) = 0$, so the update preserves the anchor's flow residual (Section D.8).

Proof of Proposition D.5. Put $Bv(\mathbf{s}, \mathbf{a}) = v(\mathbf{s}) - \gamma \mathbb{E}_P v(\mathbf{S}') = A^\top v(\mathbf{s}, \mathbf{a})$. Conditioning on $(\mathbf{s}, \mathbf{a})$, the maximizer in z is $z = \omega(\widehat{g} - Bv)$. The maximized objective is

$$\frac{1}{2}\mathbb{E}_b \omega(\widehat{g} - A^\top v)^2 = \frac{1}{2}(\widehat{g} - A^\top v)^\top W(\widehat{g} - A^\top v).$$

Its normal equation is $AWA^\top v = AW\widehat{g}$. Hence $bz = W(\widehat{g} - A^\top v) = M\widehat{g}$. The term $z(\mathbf{s}, \mathbf{a})\gamma v(\mathbf{s}')$ in the saddle has conditional mean $z(\mathbf{s}, \mathbf{a})\gamma \mathbb{E}_P v(\mathbf{S}')$, which proves the sample correspondence. □

D.9 COVARIANCE FORM OF THE ONE-STATE CORRECTION

This subsection explains how changing the joint target changes each agent's action probability in the one-state setting. Under KL regularization, DCP uses the full action covariance matrix, whereas a local natural-gradient update uses only each agent's variance. The derivation provides the theoretical basis for the covariance comparison in the safety-switch experiments.

For one state and binary actions, let $p = (p_i)$ be the extracted probabilities of action one, write the product-policy cost as $F_c = \widetilde{F}(p)$, and let $\mathbf{A} \in \{0, 1\}^N$ be a joint action drawn from q.

Proposition D.6 (How DCP changes local action probabilities)**.** *Under KL correction, the exact-gradient* DCP *direction $-Mg$ and an increase in the multiplier change local action probabilities as follows:*

$$\frac{dp}{d\eta} = -\frac{1}{\alpha} \operatorname{Cov}_q(\mathbf{A}) \nabla \widetilde{F}(p), \qquad \frac{dp}{d\lambda} = -\frac{1}{\alpha} \operatorname{Cov}_q\big(\mathbf{A}, c(\mathbf{A})\big) \tag{34}$$

The first derivative describes DCP; the second describes the multiplier. A natural-gradient update of the local policy logits gives $dp/d\eta = -\operatorname{diag}\{\operatorname{Var}_q(A_i)\}\nabla \widetilde{F}(p)$.

Proof of Proposition D.6. Let J be the $N \times |\mathcal{A}|$ matrix with entries $J_{i,\mathbf{a}} = a_i$, so that $p = Jq$ on the simplex and, along any zero-mass direction, $dp = J\,dq$. The chain rule gives $g = J^\top \nabla \widetilde{F}(p)$ up to a constant vector, which M annihilates. For one state and KL, $M = \alpha^{-1}\{\operatorname{diag}(q) - qq^\top\}$, hence

$$J(-Mg) = -\alpha^{-1}\{J \operatorname{diag}(q) J^\top - Jqq^\top J^\top\}\nabla \widetilde{F}(p) = -\alpha^{-1} \operatorname{Cov}_q(\mathbf{A}) \nabla \widetilde{F}(p).$$

The multiplier direction $-Mc$ gives $-\alpha^{-1} \operatorname{Cov}_q(\mathbf{A}, c(\mathbf{A}))$ in the same way. For the logits θ_i with $p_i = \sigma(\theta_i)$, the Fisher information is $p_i(1 - p_i) = \operatorname{Var}_q(A_i)$ and the natural gradient of $\widetilde{F}$ is $\partial_{p_i} \widetilde{F}$, so $dp_i = -\operatorname{Var}_q(A_i)\partial_{p_i} \widetilde{F}$. For the tilt $q_\theta \propto q\, e^{\theta^\top \mathbf{A}}$, both $\partial p/\partial \theta$ and the Fisher information at $\theta = 0$ equal $\operatorname{Cov}_q(\mathbf{A})$, which is positive definite because $q > 0$; the natural gradient in θ is therefore $\nabla \widetilde{F}(p)$, and it moves p by $-\operatorname{Cov}_q(\mathbf{A}) \nabla \widetilde{F}(p)$, which is (34) up to the factor $1/\alpha$. For a general divergence, $M = \operatorname{diag}(h) - hh^\top / Z_h$ and the same algebra gives $-Z_h \operatorname{Cov}_{\bar{h}}(\mathbf{A}) \nabla \widetilde{F}(p)$. Categorical actions use one-hot encodings in place of $\mathbf{A}$. □

E DISCOUNTED OCCUPANCY TRANSFER

A difference between the target's and the deployed policy's action distributions can also change future state visits. We bound the resulting occupancy and cost differences, accounting for this accumulation over time. The transition assumption below includes a shared probability of moving to the same state distribution, which limits how long the effects of earlier actions persist.

Let q satisfy Bellman flow, let q_S be its state marginal, let π be an extracted Markov policy, and define $K_q(\pi) = \mathbb{E}_{q_S} D_{\mathrm{KL}}(q(\cdot \mid \mathbf{S}) \| \pi(\cdot \mid \mathbf{S}))$. Suppose the full transition channel has a common refresh component, with a fixed state distribution ζ, a residual transition kernel R, and $0 \le \kappa \le 1$:

$$P(\cdot \mid \mathbf{s}, \mathbf{a}) = (1-\kappa)\zeta + \kappa R(\cdot \mid \mathbf{s}, \mathbf{a}). \tag{35}$$

Theorem E.1 (Sharp occupancy transfer)**.** *For finite $K_q(\pi)$,*

$$D_{\mathrm{KL}}(q \| d_\pi) \le \frac{K_q(\pi)}{1-\gamma\kappa}. \tag{36}$$

The constant is attained by a five-state model with two binary agents and exact local extraction. For $0 \le c \le c_{\max}$,

$$(1-\gamma) J_c(\pi) \le \mathbb{E}_q c + c_{\max} \sqrt{\frac{K_q(\pi)}{2(1-\gamma\kappa)}}. \tag{37}$$

The KL chain rule gives $D_{\mathrm{KL}}(q \| d_\pi) = D_{\mathrm{KL}}(q_S \| d_{\pi,S}) + K_q(\pi)$. Bellman flow and the common-refresh decomposition bound the first term by $\gamma\kappa D_{\mathrm{KL}}(q \| d_\pi)$. The coefficient concerns the full state–action transition channel. The proof below gives the equality construction and an example distinguishing it from the mixing rate of the extracted state kernel.

Proof of Theorem E.1. Flow feasibility gives $q_S = (1-\gamma)\mu_0 + \gamma P^\top q$ and the analogous equality for $d_{\pi,S}$. Joint convexity and the common refresh law yield

$$D_{\mathrm{KL}}(q_S \| d_{\pi,S}) \le \gamma D_{\mathrm{KL}}(P^\top q \| P^\top d_\pi) \le \gamma\kappa D_{\mathrm{KL}}(q \| d_\pi).$$

The chain rule then proves (36). Finite conditional KL ensures that actions used by q are supported by π on its reachable states. Since q satisfies discounted flow, these states are also reachable under π, justifying the finite-support chain rule. Boundary cases follow by approximation. The cost inequality uses $|\mathbb{E}_q c - \mathbb{E}_{d_\pi} c| \le c_{\max} \mathrm{TV}(q, d_\pi)$ and Pinsker's inequality.

For equality, take one root and four leaves indexed by the joint binary action. Start at the root. With probability $1-\kappa$ every transition returns to the root; otherwise a root action chooses its leaf and a leaf stays put. At the root let the central law be a nonproduct u and let $v = \prod_i u_i$ be its extraction. At every leaf, both policies take the same fixed action. Root occupancy is $1-\gamma\kappa$ for both, and the leaf occupancies are $\gamma\kappa u$ and $\gamma\kappa v$. Therefore $D_{\mathrm{KL}}(q \| d_\pi) = D_{\mathrm{KL}}(u \| v)$ and $K_q(\pi) = (1-\gamma\kappa) D_{\mathrm{KL}}(u \| v)$. □

The full channel matters. Take a safe state that stays safe under equal binary actions and enters an absorbing costly state under unequal actions. The cost is the indicator of the costly state. A central policy mixing 00 and 11 is safe, while independent extraction is uniform at the safe state. At $\gamma = 0.9$,

$$\mathbb{E}_q c = 0, \quad (1-\gamma) J_c(\pi) = \frac{9}{11}, \quad K_q(\pi) = \log 2, \quad D_{\mathrm{KL}}(q \| d_\pi) = \log 11.$$

The extracted state kernel has Dobrushin coefficient $1/2$, but $\log 11 > \log 2/(1 - 0.9/2)$. This example distinguishes the full state–action channel coefficient in (35) from a coefficient computed after fixing the extracted policy.

F INFORMATION LOST BY LOCAL EXTRACTION

The transfer bound depends on how closely the local policies reproduce the target's joint action distribution. We separate this mismatch into action dependence, state information missing from local

observations, and policy-fitting error. The decomposition explains why exact fitting alone need not remove the gap between target cost and deployed cost.

$$K_q(\pi) = \mathrm{TC}_q(A_1, \dots, A_N \mid \mathbf{S}) + \sum_i I_q(A_i; \mathbf{S} \mid O_i) + \sum_i \mathbb{E}_{q_{O_i}} D_{\mathrm{KL}}(q(A_i \mid O_i) \| \pi_i(A_i \mid O_i)). \tag{38}$$

Here TC is conditional total correlation and I_q is conditional mutual information under q.

For $\pi(\mathbf{a} \mid \mathbf{s}) = \prod_i \pi_i(a_i \mid O_i(\mathbf{s}))$, insert $q(A_i \mid \mathbf{S})$ and $q(A_i \mid O_i)$ in the log density ratio:

$$K_q(\pi) = \mathbb{E}_q \log \frac{q(\mathbf{A} \mid \mathbf{S})}{\prod_i q(A_i \mid \mathbf{S})} + \sum_i \mathbb{E}_q \log \frac{q(A_i \mid \mathbf{S})}{q(A_i \mid O_i)} + \sum_i \mathbb{E}_q \log \frac{q(A_i \mid O_i)}{\pi_i(A_i \mid O_i)}.$$

The terms are respectively conditional total correlation, conditional mutual information, and local fitting KL. Since O_i is a function of $\mathbf{S}$, the second term is $I_q(A_i; \mathbf{S} \mid O_i)$. This proves (38).

G Implementation of the correction

This section turns the projected direction into a candidate target and an evaluated local policy. It records the numerical domains, fitted quantities, and acceptance rules needed to interpret the experiments.

Algorithm 1 Deployed-Cost Projection (DCP)

Input: offline trajectories, normalized budgets, base weights, local actors, weight cap, and a fixed step schedule.

1. Freeze the current actor and normalized target $q = bw$. Fit its cost critic, occupancy ratio, local ratios, and counterfactual advantages; form $\widehat{g}$.
2. Fit the flow-projection saddle (33). Propose $w^+ = w - \eta z$ subject to support, mass, and cap tests.
3. Refit a candidate local actor. Fit/evaluate its own frozen-policy costs on the designated selection data and check all budgets. Use the infeasible-start rule (42); backtrack within the prespecified trial cap and accept when the cost test passes, otherwise return the current weights and actor.
4. Repeat within the correction budget. Audit the selected policy on untouched data and fresh evaluation episodes.

G.1 Baseline and numerical domains

We specify the target and actor supplied to DCP, the allowed range of occupancy ratios, and the conditions checked before applying an update. These choices determine which candidate targets are valid and where the smooth-response analysis applies. We also describe how an existing neural occupancy learner can supply the starting target.

DCP takes an occupancy target and its local extractor as inputs. One optional neural implementation obtains the target from the constrained dual, shared multiplier and monotonic mixer of Bui et al. (2026), with a weighted local MADT actor. DCP adds frozen-policy evaluation and a signed correction head; these centralized components are used in training.

For a finite ratio cap $W_{\max}$, use $f^*_{[0,W_{\max}]}(y) = \sup_{0 \le w \le W_{\max}} \{yw - f(w)\}$. Its derivative is the bounded inner optimizer. This defines the Hellinger computation directly at the cap, including arguments for which the unrestricted conjugate is infinite. The smooth response analysis concerns an interior face; the Hellinger construction in Proposition D.1 lies strictly below the cap. Normalize the fitted target weights by their reference-data mean before extraction: the base dual uses $\widetilde{w}$, while the extractor and correction use $w = \widetilde{w}/\mathbb{E}_b\widetilde{w}$. A zero normalizer keeps the previous extraction target. Choose a correction cap $W_{\mathrm{corr}} \ge \max\{W_{\max}, \|w\|_\infty\}$ on the training support, so the normalized

anchor satisfies its correction bounds. During a correction, freeze these anchor weights, their curvature, and the policy used for evaluation.

The active correction support is $\mathcal{X}_+ = \{x : w(x) > 0\}$. Set $z = 0$ outside this support and evaluate z^2/ω only where $\omega > 0$. The active observation marginals define the local ratio fits. For an approximate signed head, enforce zero mass by

$$z(x) \leftarrow z(x) - \frac{\mathbb{E}_b[\mathbf{1}_{\mathcal{X}_+} z]}{b(\mathcal{X}_+)} \quad (x \in \mathcal{X}_+), \qquad z(x) = 0 \quad (x \notin \mathcal{X}_+).$$

Compute this mean over the correction split, rather than centering each minibatch independently. Backtracking keeps $0 < w^+(x) \le W_{\mathrm{corr}}$ on the active support. If the current bounds admit no positive step, retain the anchor. Projection error in Theorem D.3 includes the effect of the fitted and centered head. In an active-face tabular implementation, retain an independent set of flow rows after restricting to the active coordinates. The moment saddle can instead be fitted directly on that support. For an observation with zero proxy mass, keep a fixed local conditional; ratio fitting and differentiation use the active observation marginals.

G.2 Fixed-policy occupancy evaluation

The correction uses the state visitation of the actor that will be deployed. With that actor frozen, its occupancy ratio can be fitted from flow moments before estimating the local cost gradient.

For a frozen policy π, its ratio $u_\pi = d_\pi/b$ is determined by the state–action flow moments

$$\mathbb{E}_{bP}\big[u_\pi(\mathbf{s},\mathbf{a})\{y(\mathbf{s},\mathbf{a}) - \gamma \mathbb{E}_{\mathbf{a}'\sim\pi(\cdot|\mathbf{s}')} y(\mathbf{s}',\mathbf{a}')\}\big] = (1-\gamma)\mathbb{E}_{\mathbf{s}_0\sim\mu_0,\mathbf{a}_0\sim\pi} y(\mathbf{s}_0,\mathbf{a}_0). \tag{39}$$

One population saddle for fitting the ratio is

$$\begin{aligned}\min_{u\ge 0}\max_{y} \quad & \frac{1}{2}\mathbb{E}_b(u-1)^2 + (1-\gamma)\mathbb{E}_{\mu_0\pi} y \\ & + \mathbb{E}_{bP}[u(\mathbf{s},\mathbf{a})\{\gamma \mathbb{E}_{\mathbf{a}'\sim\pi} y(\mathbf{s}',\mathbf{a}') - y(\mathbf{s},\mathbf{a})\}]\,. \end{aligned} \tag{40}$$

With unrestricted test functions, maximizing in y imposes (39), whose solution is the unique fixed-policy discounted occupancy on the supported model. The empirical saddle uses offline transitions and samples the new actions from the product policy. Initial-state observations come from trajectory starts. In tabular models, exact linear solves provide the reference ratio.

The centralized cost critic uses fitted policy evaluation, with target $c(\mathbf{s},\mathbf{a}) + \gamma \mathbb{E}_{\mathbf{a}'\sim\pi} Q_c(\mathbf{s}',\mathbf{a}')$ and a frozen target critic during each regression phase. The policy and the dataset are fixed during this evaluation block.

G.3 Local ratio and counterfactual fits

The local-gradient formula has an observation-frequency ratio and a counterfactual cost term. Here we describe separate fits for these quantities under a frozen actor and target.

An optional neural implementation accepts weights from an existing learner, such as Bui et al. (2026). Freeze the weights and actor, fit the cost critic and $u_\pi = d_\pi/b$, and estimate the local ratio by

$$\min_{\rho_i} \ \frac{1}{2}\mathbb{E}_b[w\rho_i(O_i)^2] - \mathbb{E}_b[u_\pi \rho_i(O_i)]. \tag{41}$$

The moment saddle gives $w^+ = w - \eta z$. The population curvature factors for KL, Pearson, and squared Hellinger are w/α, $1/\alpha$, and $2w^{3/2}/\alpha$, respectively. Neural ratios and critics fitted to $\widehat{\pi}^q$ define a surrogate direction; their discrepancy from g belongs to ϵ.

At fixed o_i, differentiating (41) gives $q_{O_i}(o_i)\rho_i(o_i) - d_{\pi,O_i}(o_i) = 0$, proving its population solution. Both terms can be estimated with weights on the same reference transitions; no behavioral action-density model is required.

For a state sample, draw the designated local action and the other agents' actions from the frozen policy. Use the target $Q_c(\mathbf{s}, a_i, \mathbf{A}_{-i}) - \widehat{V}_c(\mathbf{s})$, where $\widehat{V}_c$ averages separate policy-action draws. Regress this target onto (o_i, a_i) with state weights u_π. Since the sampled a_i depends on the state

only through o_i, the conditional regression target is exactly (6) at the population fits. Separate baseline-action draws avoid canceling the counterfactual target against itself.

The correction head $z(\mathbf{s}, \mathbf{a})$ and state potential $v(\mathbf{s})$ solve (33), with $\widehat{g}$ and ω frozen. The head is signed; its output is a change of density ratio, not a new nonnegative ratio by itself. The updated ratio is $w - \eta z$. At the population saddle, $\mathbb{E}_b z = 0$ follows by testing the constant potential, and the normalization is preserved.

G.4 Step acceptance and multiple costs

A projected direction is only a candidate update. The final step must respect support and ratio limits, lower the selected deployed cost, and satisfy every active budget after refitting the actor.

The population reference checks positivity and the ratio cap, then backtracks η until the finite-step cost test passes. Neural runs refit a candidate actor for the trial weights and fit a separate fixed-policy evaluator for that frozen candidate: a critic for the anchor policy is not reused as its value. Use trajectory-level training/selection/audit splits; candidate selection can reuse only the selection split, while the audit split and simulator episodes remain unseen until selection is complete. Each candidate is compared with the current policy; failure to accept returns the current policy.

For an offline estimate $\widetilde{F}_k(q)$ of the actual fitted actor cost $\widehat{F}_k(q)$ and an allowance $\delta_k(q)$, write $U_k(q) = \widetilde{F}_k(q) + \delta_k(q)$ and $L_k(q) = \widetilde{F}_k(q) - \delta_k(q)$. A budget-aware test is

$$U_k(q^+) \le \max\{\bar{c}_k, L_k(q)\} \quad \text{for every } k, \qquad U_{k_*}(q^+) \le L_{k_*}(q) \tag{42}$$

for the selected correction cost k_*. If these intervals contain the true costs, previously feasible constraints remain feasible and the selected true cost does not increase. Indeed, for a feasible current cost, $L_k(q) \le \widehat{F}_k(q) \le \bar{c}_k$, while the selected inequality gives $\widehat{F}_{k_*}(q^+) \le U_{k_*}(q^+) \le L_{k_*}(q) \le \widehat{F}_{k_*}(q)$. The exact tabular reference uses zero radii. Any probabilistic allowance used for an adaptive candidate search must hold simultaneously for the tested candidates. Fresh simulator episodes measure outcomes and never select offline candidates.

The default selection rule chooses the largest positive normalized budget excess; ties use a fixed index order. When all fitted costs are below budget, a correction may be applied to the cost closest to its limit. A geometric step schedule uses a prespecified initial step, reduction factor and trial cap. Those settings are frozen before the comparison runs and reported with the results.

H Relationship to prior work

This section explains which learning components and mathematical tools come from prior work and which results address the cost after local policy extraction. We first relate the analysis to the underlying occupancy learner, then discuss the projection, gradient, and transfer arguments. The final subsection states the inherited learning objective explicitly.

Bui et al. (2026) introduce the safe offline MARL backbone: a constrained occupancy program, shared multipliers, a monotonic dual mixer, and weighted local MADT extraction. Its dual-approximation and corrected-cost sensitivity results concern the quantity $c^\top q$. The present question concerns $F_c(q)$ after extraction. In particular, a zero dual approximation error does not force their derivatives to have the same sign. The exact Hellinger construction, sharp four-mode theorem, all-dimensional Boolean classification, extraction gradient, and sharp transfer construction address this distinction. DCP applies to the occupancy-and-extraction setting studied here; its neural implementation may use the existing SamaDICE backbone. That backbone and its earlier benchmark tables are attributed to the predecessor.

The generic inverse-Hessian projection is established in Alvarez et al. (2004), and counterfactual credit in Foerster et al. (2018). Our local derivative includes the $d_{\pi,O_i}/q_{O_i}$ factor because the differentiated variable is the occupancy supplied to conditional extraction. The transition-sampled saddle computes the associated flow projection. The finite-step and actor-error bounds state the accuracy needed to preserve descent. Independent policy factorization can already create unsupported joint actions (Matsunaga et al., 2023); the Hellinger witness isolates a sign reversal even at a product target with full support.

The KL contraction used in Section 5.4 follows joint convexity and data processing. The same generic argument contracts a fixed-policy adjoint Bellman operator in van der Laan & Kallus (2026, Lemma 3.1). Here Bellman flow and extraction error give the factor $1/(1-\gamma\kappa)$ for the full transition channel; the five-state construction attains equality with an extracted product policy.

H.1 The inherited occupancy learner

For clarity, we state the occupancy objective and extraction parameterization inherited from the baseline. DCP acts after these components produce a target and does not change their dual objective.

The value function $\nu(\mathbf{s})$ is the dual variable for Bellman flow, and $\lambda_k \geq 0$ is shared across agents for the kth team cost. With $\Delta_{\nu,\lambda} = r - \sum_k \lambda_k c_k - A^\top \nu$, the reduced dual is

$$\mathcal{L}(\nu, \lambda) = b_0^\top \nu + \lambda^\top \bar{c} + \alpha \mathbb{E}_b f_+^*(\Delta_{\nu,\lambda}/\alpha), \qquad \widetilde{w}_{\nu,\lambda} = (f_+^*)'(\Delta_{\nu,\lambda}/\alpha), \tag{43}$$

where $f_+^*(y) = \sup_{w \geq 0}\{yw - f(w)\}$. The base multiplier update uses $\mathbb{E}_b[\widetilde{w}_{\nu,\lambda} c_k] - \bar{c}_k$. For extraction, use $w = \widetilde{w}/\mathbb{E}_b \widetilde{w}$ and $q = bw$. At an unrestricted population optimum the normalizer equals one. The construction follows constrained DICE (Lee et al., 2022).

SamaDICE (Bui et al., 2026) represents $\nu(\mathbf{s})$ by a monotonic mixing network

$$\nu_\theta(\mathbf{s}) = \mathcal{M}_\theta(\nu_1(o_1), \ldots, \nu_N(o_N); \mathbf{s}), \qquad \frac{\partial \mathcal{M}_\theta}{\partial \nu_i} \geq 0, \tag{44}$$

following cooperative value factorization (Rashid et al., 2020; Bui et al., 2025). Reward and costs remain global inside $\Delta_{\nu,\lambda}$. The mixer parameterizes the centralized dual; the local policy is fitted by (3).

I Experimental details and further results

This section gives the protocols and detailed results for the experiments in Section 6. We specify the information available to each method, how targets and updates are computed, and how costs and rewards are evaluated. The tables and figures then cover the structural checks, safety-switch comparisons, and finite-data MDP tests, followed by a bound covering all multiplier values.

Implementation and information access. All floating-point experiments use float64 on one RTX 4080 SUPER GPU (TF32 disabled); the rational witnesses and the certificate replay use exact rational arithmetic on the CPU. Per-row statistics (marginals, product laws, deployed costs, gradients, covariances, step-size bisection) are explicit Triton kernels, each checked against an eager reference to at most 10^{-12}. Loop bodies are compiled with `torch.compile(fullgraph=True)`, and all configurations advance in lockstep with masks. Occupancies and deployed values use finite enumeration or numerical solves: MDPs are solved by batched linear solves, and DICE targets by a batched damped Newton method on the dual (flow residual at most 10^{-12}). Estimated panels (E1, E3 and the MDP budget runs) see only empirical tables fitted with a pseudocount of 0.5 per cell. There is no rollout noise, so the only randomness is the sampled data; intervals are 95% paired t intervals over the three datasets or five independent datasets, for fixed model instances.

Correction to a budget. From an anchor λ_0 with DICE target q_0, an arm is iterated until the deployed team cost meets $\bar{c}$. Each DCP or actor step has a mean per-agent KL of at most $\kappa = 10^{-3}$ between successive local policies, found by 60 bisection steps; DCP also caps its step at $\eta \leq \eta_{\mathrm{pos}}/2$, where η_{pos} is the largest step that keeps q positive. A step is accepted if the deployed team cost strictly decreases; otherwise the step is halved up to 29 times, and the run stops if none of these 30 trials is accepted. At most 3,000 steps are allowed. The actor arm is the natural gradient of the local logits ($\theta_i \leftarrow \theta_i - \eta\, \partial F_c/\partial p_i$ for one state, the tabular natural policy gradient on local advantages for MDPs). It uses the same deployed-cost gradient as DCP; neither arm uses the reward. The multiplier arm re-solves the target on a 4,001-point grid of $\lambda \in [\lambda_0, 200]$, refined by bisection, and returns the smallest λ that meets the budget.

Response and structure (E2a, E2c, screen). For the random MDP screen, $\gamma = 0.9$, μ_0 is uniform, and each transition row is $0.1\,\mathrm{Unif} + 0.9\,\mathrm{Dirichlet}(0.5)$. Independently in each state, cost

Table 1: One-state screen under KL correction. Each entry gives the percentage of problems whose deployed response is positive at some $\lambda \in \{0, 0.5, 1, 2, 5\}$, the percentage whose finite-search minimum $\min_{\lambda \in \Lambda_{\text{scr}}} F_c(q_\lambda)$ exceeds $c_{\min}$ by more than 0.01, and, among 4,096 of these floor problems per cell corrected from $\lambda = 1$, the percentage in which DCP meets the midpoint budget $\bar{c} = \frac{1}{2}\big(c_{\min} + \min_{\lambda \in \Lambda_{\text{scr}}} F_c(q_\lambda)\big)$ with more reward than the actor at the first iterate within budget. Real-valued costs have unique minimizers, so their floor vanishes as $\lambda \to \infty$; their entries reflect $\lambda \leq 200$.

Cost class	$N = 2$	3	4	5	6
additive	0.0 / 0.0 / –	0.0 / 0.0 / –	0.0 / 0.0 / –	0.0 / 0.0 / –	0.0 / 0.0 / –
pairwise hazard	0.0 / 41.8 / 80	0.0 / 53.6 / 85	0.1 / 71.5 / 87	0.1 / 86.5 / 88	0.0 / 94.9 / 88
paired hazards	0.0 / 41.8 / 81	0.0 / 53.6 / 87	1.0 / 84.1 / 90	0.5 / 94.2 / 90	0.2 / 99.4 / 94
Boolean	0.0 / 22.3 / 80	2.3 / 60.5 / 76	5.8 / 90.0 / 83	10.1 / 99.1 / 90	15.7 / 100.0 / 92
real-valued	15.8 / 0.2 / –	33.4 / 0.7 / –	48.7 / 2.0 / –	58.8 / 4.9 / –	63.7 / 11.1 / –

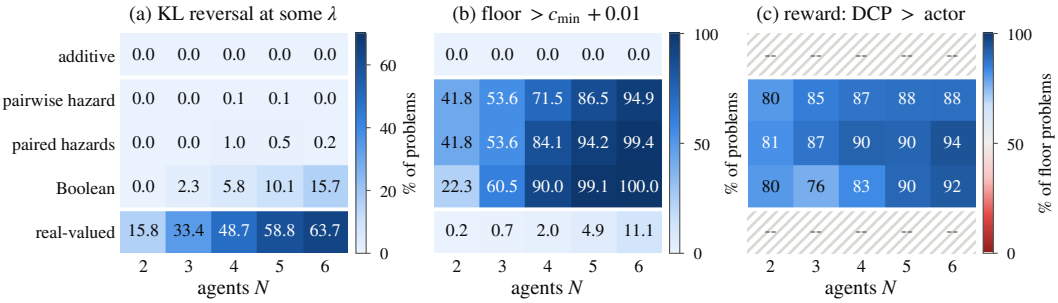


Figure 6: One-state screen under KL correction, 2^{20} problems per cell (values of Table 1). (a) Percentage of problems whose deployed response is positive at some $\lambda \in \{0, 0.5, 1, 2, 5\}$. (b) Percentage whose finite-search minimum $\min_{\lambda \in \Lambda_{\text{scr}}} F_c(q_\lambda)$ exceeds $c_{\min}$ by more than 0.01. (c) Among 4,096 floor problems per binary-class cell, corrected from $\lambda = 1$ to the midpoint budget, the percentage in which DCP meets the budget with more reward than the actor. Hatched cells are not applicable.

is either $\mathbf{1}\{\text{not all } a_i \text{ are equal}\}$ or $\mathbf{1}\{\sum_i a_i \geq 2\}$, with equal probability. Reward is $r(\mathbf{s}, \mathbf{a}) = (N^{-1} \sum_i a_i) U_{\mathbf{s},\mathbf{a}} + 0.1 V_{\mathbf{s},\mathbf{a}}$, with independent $U_{\mathbf{s},\mathbf{a}} \sim \text{Unif}[0.5, 1.5]$ and $V_{\mathbf{s},\mathbf{a}} \sim \text{Unif}[0, 1]$. The behavior policy puts mass 0.45 on a uniformly chosen joint action and 0.45 on its bitwise complement, plus 0.10 uniform mass, separately in each state. Its exact discounted occupancy is b. Full observations reveal the state; aliased observations independently permute the states for each agent and merge consecutive pairs. Regularization uses $\alpha = 0.2$. Central differences with $h = 10^{-3}, 10^{-4}, 10^{-5}$ converge at order h^2; with $h = 10^{-5}$ the largest relative error of the deployed response is 5.1×10^{-6} (KL) and 2.5×10^{-5} (Hellinger), and 3.6×10^{-6} for Pearson targets on a fixed face. Pearson targets with $\alpha = 0.2$ are compared on their active face; 15% of the aliased cases then reverse (Figure 7). Table 1 and Figure 6 summarize the one-state screen, with 2^{20} problems per cell. No reversals occur in two-agent Boolean cases or the universally monotone three-agent subclasses, as predicted by Theorems B.1 and C.1. The 4,096 correction instances per binary-class cell are the first 4,096 floor cases of this screen, in a fixed order. The E2c certificate replay reproduces the split of the 256 three-agent Boolean cost tables into 110 safe and 146 unsafe ones, all positive remainders and all rational witnesses. The separate exact checks in Section A.10 cover the four-variable inputs to the all-dimensional theorem: 742 safe and 64,794 unsafe tables, with witnesses for all 211 unsafe symmetry classes. These finite certificates support the structural proof, not a numerical extrapolation in the number of agents.

One-state sampling laws. For each problem, draw $v \sim \text{Dirichlet}(0.3)$ on the 2^N joint actions and choose ζ uniformly from $\{0.01, 0.05, 0.20\}$. Set $b = (1 - \zeta)v + \zeta\,\text{Unif}$; reward entries are independent $\text{Unif}[0, 1]$ values. Real-valued cost entries have the same uniform law. Boolean entries are independent $\text{Bernoulli}(0.5)$, conditioned on a nonconstant table. Additive costs average independent uniform local-action costs. Pairwise hazards use independent $\text{Bernoulli}(0.5)$ edges, conditioned on a

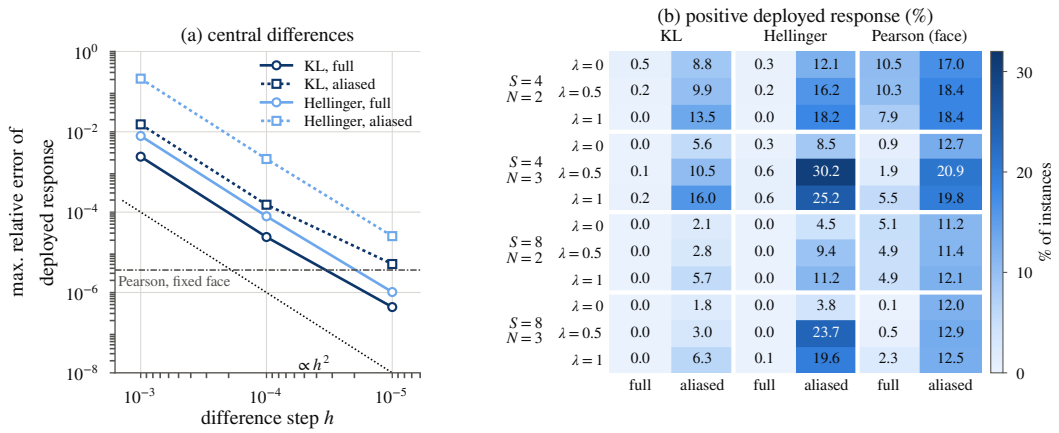


Figure 7: Response identity on 16,384 random finite MDPs ($\lambda \in \{0, 0.5, 1\}$). (a) Largest relative error between the deployed response (5) and central differences of re-solved targets with step h; the errors decrease as h^2 (dotted guide), and the dash-dotted line is the Pearson error on a fixed active face. (b) Percentage of the 4,096 instances in each $(|\mathcal{S}|, N, \lambda)$ cell, per divergence and observation type, with a positive deployed response (Pearson uses the valid fixed-face subset). The proxy response is negative in every KL and Hellinger instance.

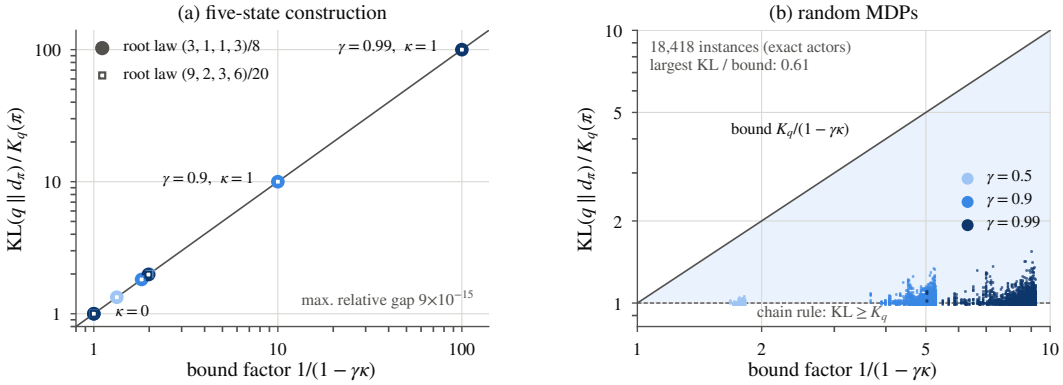


Figure 8: Occupancy transfer. (a) Five-state construction of Section E: $D_{\mathrm{KL}}(q\|d_\pi)/K_q(\pi)$ equals the factor $1/(1-\gamma\kappa)$ for $\gamma \in \{0.5, 0.9, 0.99\}$, $\kappa \in \{0, 0.5, 1\}$ and two root laws (largest relative gap 9×10^{-15}). (b) Random MDPs with exact local extraction: every ratio lies between the chain-rule floor 1 and the bound of Theorem E.1; color gives γ.

nonempty graph, and charge one if any selected pair acts jointly. Paired hazards use the fixed disjoint pairs $(1, 2), (3, 4), \ldots$. The KL target is proportional to $b \exp((r - \lambda c)/0.1)$. The screen grid Λ_{scr} contains zero and 2,000 geometrically spaced values from 0.001 to 200. Its reported minimum is the minimum over this grid.

Transfer and information (E2d). Figure 8 plots both checks. In the five-state construction, $D_{\mathrm{KL}}(q\|d_\pi)/K_q(\pi)$ equals $1/(1-\gamma\kappa)$ to a relative error of 9×10^{-15} for $\gamma \in \{0.5, 0.9, 0.99\}$ and $\kappa \in \{0, 0.5, 1\}$, reaching 100 at $\gamma = 0.99$, $\kappa = 1$. The two-state example reproduces $\mathbb{E}_q c = 0$, $(1-\gamma)J_c(\pi) = 9/11$, $K_q = \log 2$ and $D_{\mathrm{KL}}(q\|d_\pi) = \log 11$. Over 884,736 random checks (three divergences, $\gamma \in \{0.5, 0.9, 0.99\}$, exact and perturbed actors) neither (36) nor (37) is violated. The information decomposition (38) holds to 5×10^{-16}; reducing the behavior coordination lowers the total-correlation term, aliasing adds 0.15–0.27 of mutual information, and mixing the actors toward uniform adds only the fitting term.

Structured safety switches (E1, E3, E4). Data are 10^4 joint samples per dataset drawn blockwise by inverse-CDF sampling. The multiplier floor is 0.2509–0.2523 in the estimated panel and 0.2515 in

Table 2: Correction to a deployed-cost budget $\bar{c}$ in the safety-switch family ($N = 16$, estimated panel, mean over three datasets). Entries are final deployed rewards after each step arm lands exactly on the budget; — means that no multiplier $\lambda \leq 200$ meets the budget. DCP-prod removes the cross-agent covariance of the target. The anchors at $\lambda_0 = 0.5$ and 1 have deployed costs 0.277 and 0.252 and rewards 0.323 and 0.289, respectively.

	$\lambda_0 = 0.5$			$\lambda_0 = 1$		
Update	$\bar{c} = 0.20$	0.10	0.05	0.20	0.10	0.05
Increase multiplier	—	—	—	—	—	—
Actor natural gradient	0.279	0.204	0.149	0.260	0.191	0.140
DCP-prod (no cross-covariance)	0.279	0.204	0.149	0.260	0.191	0.140
DCP	**0.286**	**0.277**	**0.265**	**0.274**	**0.265**	**0.259**

Table 3: Final deployed reward of DCP minus that of the actor at the budgets $\bar{c} \in \{0.20, 0.10, 0.05\}$, all below the multiplier floor, at the first iterate within budget (estimated panel: mean and 95% paired t interval over three datasets; oracle: exact). Tables 2 and 5 instead report rewards after landing exactly on the budget.

λ_0	N	$\bar{c} = 0.20$	$\bar{c} = 0.10$	$\bar{c} = 0.05$
0.5	4	+0.015 [+0.009, +0.021]	+0.059 [+0.022, +0.096]	+0.094 [+0.011, +0.177]
0.5	8	+0.014 [+0.010, +0.017]	+0.083 [+0.064, +0.103]	+0.121 [+0.092, +0.150]
0.5	16	+0.014 [+0.007, +0.021]	+0.082 [+0.067, +0.097]	+0.122 [+0.113, +0.131]
0.5	32	+0.014 [+0.012, +0.017]	+0.087 [+0.082, +0.092]	+0.120 [+0.108, +0.133]
0.5	64	+0.014 [+0.010, +0.018]	+0.087 [+0.083, +0.091]	+0.122 [+0.115, +0.130]
0.5	oracle	+0.015	+0.034	+0.016
1	4	+0.014 [+0.004, +0.024]	+0.069 [+0.021, +0.118]	+0.099 [+0.025, +0.173]
1	8	+0.016 [+0.009, +0.023]	+0.085 [+0.067, +0.103]	+0.124 [+0.101, +0.148]
1	16	+0.017 [+0.011, +0.023]	+0.087 [+0.077, +0.097]	+0.125 [+0.117, +0.134]
1	32	+0.017 [+0.012, +0.022]	+0.087 [+0.079, +0.094]	+0.125 [+0.113, +0.136]
1	64	+0.017 [+0.013, +0.021]	+0.087 [+0.081, +0.093]	+0.125 [+0.118, +0.133]
1	oracle	+0.012	+0.048	+0.019

the oracle, for every N; at $\lambda_0 = 1$ the deployed response lies between -1.8×10^{-3} and -1.5×10^{-3}, compared with -0.72 to -0.61 at $\lambda_0 = 0$. Table 3 gives the paired reward differences for every N and Table 4 the stress settings at $N = 16$. From $\lambda_0 \in \{0.5, 1\}$ in the estimated panels, the difference at $\bar{c} \leq 0.10$ is at least 0.035 in every dataset of every stress setting. A DCP step takes 0.22–0.96 ms and an actor step 0.22–1.20 ms (Figure 10, right) for $N \in \{4, \ldots, 64\}$ (steady state, 48 configurations per step, on a shared GPU).

Covariance ablation and controlled asymmetry (E5a, E5b). DCP-prod is DCP with one change: before every step the target is replaced by the product of its own normalized marginals, so the extracted policy is unchanged and, by Proposition D.6, the induced local step uses $\mathrm{diag}\{\mathrm{Var}_q(A_i)\}$ instead of $\mathrm{Cov}_q(\mathbf{A})$. Step rule, positivity cap, KL matching, acceptance and stopping are those of DCP. In these runs every step arm (DCP, DCP-prod, actor) lands exactly on the budget: after the step that first meets it, the step length is re-chosen by bisection so that $F_c = \bar{c}$ to 10^{-12} (all 414 block runs and all $61{,}440 \times 3$ screen runs landed; the largest residual is below 10^{-15}). The oracle panel perturbs the witness law to masses $1/8$, $\frac{7}{16}(1+\delta)$ and $\frac{7}{16}(1-\delta)$ on 1100, 0010 and 0001, with 5% uniform mass as before (Figure 9). In all 84 oracle cases (Table 5 shows $\lambda_0 = 1$), DCP-prod stays within 0.0023 of the actor, and in all 54 estimated ones within 1.1×10^{-4}, so removing the cross-agent covariance removes at least 97.6% of the DCP gain in every oracle case and at least 99.9% in every estimated one. The oracle gain rises steeply with small asymmetries, from 0.020 at $\delta = 0$ to 0.116 at $\delta = 0.01$ and 0.129 at $\delta = 0.1$ ($\bar{c} = 0.05$, $\lambda_0 = 1$). On the one-state screen (Table 6), DCP keeps more reward than DCP-prod in 79–95% of the floor problems of each cell, while the median ratio of DCP-prod to actor rewards is below 1.0008.

Finite-data correction and bounds (E2b, E2e). The eight MDPs have $|\mathcal{S}| = 8$, two agents and $\gamma = 0.9$; trajectories start from μ_0 and continue with probability γ, and smaller datasets are prefixes

Table 4: Stress settings at $N = 16$ (estimated panel): mean final reward, at the first iterate within budget, of DCP minus that of the actor over three datasets, varying one axis at a time.

Axis	Value	$\lambda_0 = 0.5$		$\lambda_0 = 1$	
		$\bar{c} = 0.10$	0.05	0.10	0.05
uniform mass ζ	0.01	+0.113	+0.156	+0.091	+0.146
uniform mass ζ	0.2	+0.044	+0.096	+0.052	+0.102
step KL κ	10^{-4}	+0.076	+0.119	+0.077	+0.123
step KL κ	10^{-2}	+0.111	+0.147	+0.077	+0.133

Table 5: Covariance ablation in the safety-switch family ($\lambda_0 = 1$; rewards after landing exactly on the budget). Here prod denotes DCP-prod. Oracle rows vary the asymmetry δ of the reference ($N = 4$; $N = 16$ gives the same values); estimated rows use the three E1 datasets.

		$\bar{c} = 0.10$			$\bar{c} = 0.05$		
Panel	δ	DCP−actor	DCP−prod	prod−actor	DCP−actor	DCP−prod	prod−actor
oracle	0	+0.036	+0.036	+0.000	+0.020	+0.020	+0.000
oracle	10^{-4}	+0.036	+0.036	+0.000	+0.024	+0.024	+0.000
oracle	10^{-3}	+0.046	+0.046	+0.000	+0.075	+0.075	+0.000
oracle	0.01	+0.070	+0.070	+0.000	+0.116	+0.116	+0.000
oracle	0.03	+0.079	+0.079	+0.000	+0.127	+0.127	+0.000
oracle	0.1	+0.083	+0.082	+0.000	+0.129	+0.129	+0.001
estimated, $N = 4$	0	+0.057	+0.057	+0.000	+0.093	+0.093	+0.000
estimated, $N = 16$	0	+0.074	+0.074	+0.000	+0.119	+0.119	+0.000
estimated, $N = 64$	0	+0.074	+0.074	+0.000	+0.119	+0.119	+0.000

of larger ones. The ladder replaces one component at a time by its fitted version and keeps the others exact. The anchor is the true DICE target at $\lambda_0 = 0.5$, except in the all-fitted setting, which starts from the empirical target. Step lengths are matched using the anchor's true observation marginal. Table 7 and Figure 10 (left) report the true change of deployed cost for one DCP step as the components are replaced by their fitted versions; the fitted actor is the weighted maximum-likelihood fit on the sampled observation–action pairs. Every one of the 1,200 matched steps lowers the true cost, and a step ten times larger, shortened only to keep q positive, lowers it in all 1,200 cases as well. In E2e, bandit and MDP anchors receive gradient errors of known M-seminorm, zero-mass projection errors of known norm and actor mixtures of known total variation. Bandit bounds use a certified curvature: $t \mapsto F_c(q + td)$ is a polynomial whose exact coefficients bound $|F_c''|$ on the segment. Among the premise-satisfying bandit trials, (31) is never violated in 863,814 distinct checks and (32) never in 3,418,644 checks that add the actor mixtures (largest excess 5.6×10^{-16}); (31) is negative in 99.7% of the exact trials and in 86%, 83% and 71% with gradient, projection, and combined errors. The MDP trials, with a sampled curvature, show the same pattern (100%, 94%, 92% and 85%).

Correction to a budget in MDPs. On the eight E2b MDPs, the budget is $\bar{c} = F_0 - p\,(F_0 - F_{\mathrm{lo}})$, where F_0 is the deployed cost of the anchor and F_{lo} is the lowest cost the actor reaches from the anchor; in 78 of the 96 instance–dataset units the actor reaches a zero-cost product policy, so $F_{\mathrm{lo}} = 0$. The estimated panel uses 10^4 trajectories and five independent datasets; its mean reward differences change by less than 3×10^{-4} when evaluated on the true model. Table 8 and Figure 3 show how often the multiplier meets each budget and the reward difference between DCP and the actor, both of which reach every budget of their panel's model.

Fully empirical correction (E5c). Fresh datasets of 10^4 and 2,000 trajectories (five independent datasets; the smaller is a prefix of the larger) are split by trajectory index into fit (60%), selection (20%) and held-out (20%) parts. The anchor target, gradient, projection, extraction and the KL step normalization use only the fit model, with the current actor's fitted observation marginal. Acceptance (strict decrease of the selection-model cost) and stopping use only the selection model. A run stops when $F_{\mathrm{sel}} + \delta_{\mathrm{sel}} \le \bar{c}$, where δ_{sel} is the 95th percentile of $F_{\mathrm{boot}} - F_{\mathrm{sel}}$ over 200 bootstrap resamples of the selection trajectories for the current candidate, and then lands on $F_{\mathrm{sel}} + \delta_{\mathrm{sel}} = \bar{c}$. Budgets follow

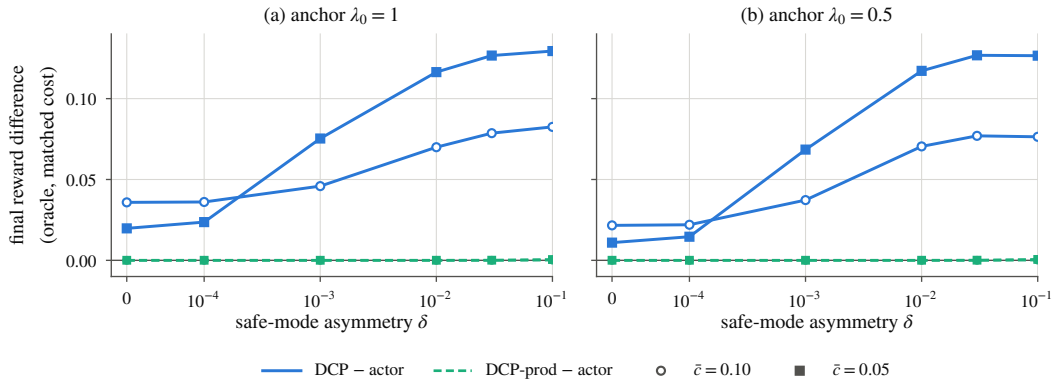


Figure 9: Controlled asymmetry in the safety-switch family (oracle panel). Final reward of DCP and of DCP-prod minus that of the actor after each arm lands exactly on the budget ($F_c = \bar{c}$), as the reference masses on 0010 and 0001 become $\frac{7}{16}(1 \pm \delta)$. (a) $\lambda_0 = 1$; (b) $\lambda_0 = 0.5$. The axis is linear below 10^{-4} and logarithmic above. Removing the cross-agent covariance (DCP-prod) removes at least 97.6% of the gain across the oracle grid.

Table 6: Covariance ablation on the one-state screen: 4,096 floor problems per cell corrected from $\lambda = 1$ to the midpoint budget of Table 1, rewards after landing exactly on the budget; prod denotes DCP-prod.

		% of problems with more reward			median reward ratio		
Cost class	N	DCP>prod	DCP>actor	prod>actor	DCP/prod	DCP/actor	prod/actor
pairwise hazard	2	80.5	80.5	79.8	1.0094	1.0096	1.0002
pairwise hazard	3	85.5	85.5	81.3	1.0156	1.0158	1.0002
pairwise hazard	4	88.6	88.6	81.7	1.0218	1.0221	1.0003
pairwise hazard	5	90.1	89.9	79.4	1.0298	1.0300	1.0004
pairwise hazard	6	91.1	90.8	80.2	1.0439	1.0446	1.0005
Boolean	2	81.3	81.3	51.8	1.0004	1.0004	1.0000
Boolean	3	79.4	79.2	65.7	1.0045	1.0046	1.0000
Boolean	4	84.9	84.9	67.1	1.0157	1.0159	1.0001
Boolean	5	91.5	91.6	65.6	1.0456	1.0461	1.0003
Boolean	6	92.9	92.9	66.1	1.0779	1.0788	1.0005
paired hazards	2	81.4	81.4	80.6	1.0102	1.0104	1.0002
paired hazards	3	87.5	87.2	79.8	1.0167	1.0169	1.0002
paired hazards	4	92.1	91.8	78.4	1.0278	1.0281	1.0004
paired hazards	5	91.6	91.5	75.2	1.0368	1.0372	1.0004
paired hazards	6	94.5	94.4	78.1	1.0641	1.0651	1.0007

the MDP budget runs, computed on the fit model. The multiplier reaches this criterion in 268 of 480 configurations. Table 9 reports true-model budget attainment and rewards. The held-out trajectories fit an independent policy-evaluation model. The displayed rates and rewards use evaluation on the generating MDP after policy selection. Step landing uses $F_{\text{sel}} + \delta_{\text{sel}}$ as its endpoint criterion.

I.1 A BOUND BEYOND THE MULTIPLIER GRID

The reported numerical minima use a finite multiplier grid. We next bound the path between grid points and beyond its endpoint to certify that the tested budgets remain out of reach for every multiplier.

The floors 0.251–0.252 are minima over the E1 grid of λ. The following conservative bound shows that the tested budgets remain out of reach over the whole multiplier path. Write $F(\lambda)$ for the block-average deployed cost and $\bar{t}_1$ for the block-average corrected cost at $\lambda = 1$. Within each block, $|\partial F/\partial p_i| \le 1$ and $|p_i'| = |\operatorname{Cov}(c, A_i)|/\alpha_0 \le 1/(4\alpha_0)$, so $|F'(\lambda)| \le 1/\alpha_0 = 10$. Let m be the

Table 7: True change of deployed cost ($\times 10^{-3}$) for one DCP step with actor KL change at most 10^{-3}, mean over eight MDPs and five datasets; n is the number of behavior trajectories. The oracle column uses the exact gradient g, flow projection M and actor; each other column replaces the named component by its fitted version. Step sizes use the true observation marginal even in the all-fitted column. Every step lowers the true cost; the reversed step raises it in every case and is rejected by the empirical evaluator.

Obs.	n	oracle	fitted g	fitted M	fitted actor	all fitted
full	100	-16.33	-16.33	-16.35	-16.10	-16.67
full	1000	-16.33	-16.27	-16.34	-16.33	-16.42
full	10000	-16.33	-16.32	-16.33	-16.33	-16.32
aliased	100	-15.78	-15.65	-15.72	-15.66	-15.79
aliased	1000	-15.78	-15.73	-15.76	-15.78	-15.84
aliased	10000	-15.78	-15.78	-15.78	-15.78	-15.81

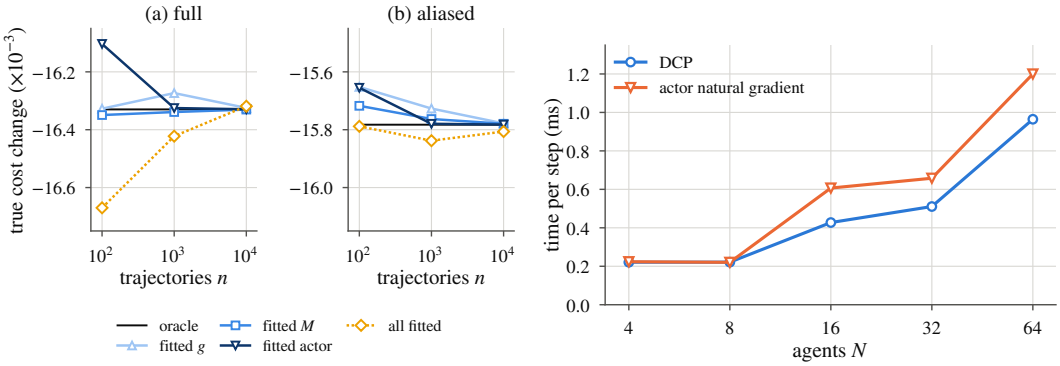


Figure 10: Left: true change of deployed cost (units of 10^{-3}) for one DCP step as the gradient, projection, actor, or all three are replaced by fitted versions, with full and aliased observations (values of Table 7); every step lowers the true cost. Right: steady-state time per step of DCP and of the actor natural gradient in the safety-switch family (48 configurations per step).

minimum over the grid beginning at zero, and put $\rho = (200/10^{-4})^{1/3999}$. The grid contains zero and 4000 geometrically spaced points from 10^{-4} to 200. On $[0, 1]$ every point is within $(\rho - 1)/2$ of a grid point, hence $F(\lambda) \geq m - 5(\rho - 1)$.

For the tail, write each block law as $(1 - t_\lambda)q(\cdot \mid c = 0) + t_\lambda q(\cdot \mid c = 1)$. The conditional laws are fixed and $t_\lambda \leq t_1$ for $\lambda \geq 1$. Each marginal moves by at most t_1, so the block's deployed cost changes by at most $4t_1$. Consequently

$$\inf_{\lambda \geq 0} F(\lambda) \geq \min\{m - 5(\rho - 1),\ F(1) - 4\bar{t}_1\}.$$

Across all E1 panels, the recorded float64 quantities give lower bounds at least 0.2327 on the first term and 0.2493 on the second, rounded down. Thus the conservative bound $F_\star > 0.23$ separates every tested budget at or below 0.20 from the whole path, including beyond 200.

Table 8: Correction to a budget in random MDPs, using the first within-budget iterate for step arms: number of MDPs in which some multiplier $\lambda \leq 200$ meets the budget (oracle), and final reward of DCP minus that of the actor (oracle mean over eight MDPs; estimated mean and 95% intervals across datasets).

Observations	Budget p	Multiplier reaches	DCP − actor (oracle)	DCP − actor (estimated)
full	0.25	8/8	+0.006	+0.006 [+0.005, +0.006]
full	0.5	7/8	+0.013	+0.015 [+0.014, +0.016]
full	0.75	4/8	+0.021	+0.021 [+0.020, +0.022]
aliased	0.25	5/8	+0.004	+0.004 [+0.004, +0.005]
aliased	0.5	3/8	+0.019	+0.021 [+0.019, +0.022]
aliased	0.75	0/8	+0.032	+0.029 [+0.026, +0.033]

Table 9: Fully empirical correction on the eight MDPs (40 instance–dataset units per row). Columns 4–5 count final policies that meet the budget on the true model. The last column is DCP minus actor true reward ($\times 10^3$) over units where both meet the true budget: mean over those units, 95% paired t interval of the per-dataset means, and number of such units.

			True budget met		
Obs.	n	p	Mult.	DCP	DCP − actor ($\times 10^3$)
full	10,000	0.25	38/40	38/40	$+8.3$ [$+7.7$, $+8.9$] (38)
full	10,000	0.5	34/40	38/40	$+13.9$ [$+12.6$, $+15.3$] (38)
full	10,000	0.75	16/40	38/40	$+23.6$ [$+20.2$, $+27.2$] (38)
full	2,000	0.25	38/40	38/40	$+8.9$ [$+7.4$, $+10.5$] (38)
full	2,000	0.5	34/40	38/40	$+14.6$ [$+12.9$, $+16.4$] (38)
full	2,000	0.75	18/40	38/40	$+23.6$ [$+20.9$, $+26.1$] (38)
aliased	10,000	0.25	25/40	38/40	$+4.5$ [$+3.4$, $+5.6$] (38)
aliased	10,000	0.5	14/40	38/40	$+22.3$ [$+19.5$, $+25.2$] (38)
aliased	10,000	0.75	0/40	38/40	$+33.7$ [$+26.6$, $+41.3$] (37)
aliased	2,000	0.25	24/40	40/40	$+5.5$ [$+4.6$, $+6.4$] (40)
aliased	2,000	0.5	15/40	39/40	$+22.0$ [$+19.6$, $+24.5$] (39)
aliased	2,000	0.75	0/40	39/40	$+28.3$ [$+21.3$, $+35.5$] (39)