SOLVING STOCHASTIC MINIMAX BANDIT GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

Minimax regret is the smallest worst-case regret achievable by an online learning algorithm. At a fixed horizon, this objective provides a computational route to algorithm design: optimizing the complete history-dependent policy offline. The resulting optimization has a natural two-player zero-sum structure, with the learner minimizing regret and the environment choosing hidden reward parameters to maximize it. We develop an approximate policy-space solver for these stochastic minimax bandit games. Each iteration plans a learner response to an environment mixture, evaluates the complete policy, and updates both players' mixtures. Grid interpolation and point-based value iteration represent finite-horizon costs in sufficient statistics. Shared two-arm costs avoid a joint state grid, while policy evaluation and mixture selection retain the original multi-arm objective. The resulting policy mixture defines an online learning algorithm that adapts its actions to observed rewards. In four exactly evaluable three-arm games, complete iterations reduce strategy-pair gaps from 0.115–0.349 to 0.001–0.016. Larger-horizon experiments cover 60 settings and track regret against tuned references over training time. Independent evaluation quantifies uncertainty in their regret differences. Response controls distinguish belief resolution from pair-prior capacity; complete runs confirm that larger priors improve regret in one Beta setting. The resulting framework uses offline game solving to design adaptive online exploration policies.

1 INTRODUCTION

Stochastic bandits model the tradeoff between reward and information. Posterior sampling dates to Thompson (1933); the theory of Lai & Robbins (1985) characterizes the information needed to distinguish an optimal arm. Finite-time confidence bounds (Auer et al., 2002), minimax algorithms (Audibert & Bubeck, 2009), and information-theoretic analyses of posterior sampling (Russo & Van Roy, 2016) formalize this tradeoff. Later algorithms combine worst-case and instance-dependent guarantees (Ménard & Garivier, 2017; Jin et al., 2021). These results characterize efficient online algorithms across horizons and problem difficulties.

At a given horizon, one can also optimize a complete exploration policy, which maps the observed history to an action distribution. The practical question is whether offline computation can produce an online learning algorithm with lower regret at an affordable cost. Adusumilli (2025) derives numerical Bayes and minimax policies from a limiting control problem. Planning under model uncertainty (Poupart et al., 2006) and differentiable policy optimization (Boutilier et al., 2020) provide further computational approaches.

For minimax regret, this optimization has a natural two-player zero-sum structure (Buening et al., 2023; Adusumilli, 2025). The environment chooses hidden reward parameters to maximize regret; the learner chooses a history-dependent policy to minimize it. Parameters stay fixed during an episode, with rewards generated by chance. An extensive-form representation makes the learner's information explicit.

Policy-space game solvers optimize mixtures over complete contingent strategies and expand the population with responses to the opponent's mixture (McMahan et al., 2003; Lanctot et al., 2017). For a bandit, the environment mixture supplies a prior for each response calculation. The reward

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.

likelihood then determines how an action changes both immediate reward and future evidence. This makes the learner's policy a product of offline planning.

Response computation determines the solver's computational scale. Sufficient statistics compress the history, but their joint state still grows with the number of arms. A shared finite horizon and a coupled prior also prevent directly using the independent-arm discounted decomposition of Gittins indices (Gittins, 1979). We approximate these responses through numerical costs in sufficient-statistic coordinates, using Grid interpolation or point-based value iteration (PBVI) (Zhu et al., 2023; Pineau et al., 2003). One two-arm planner is reused across pairs; its cost differences combine into a multi-arm policy without a joint state grid. Original-game evaluation supplies its payoff row, updating both players' mixtures.

The policy mixture defines an online learning algorithm: one policy is sampled per episode and selects actions from its observed history. Exact strategy-pair gaps in small games first test the quality of the computed solutions. Larger games measure regret and computation against tuned references, with independent evaluation of fixed policies. Response controls then examine how the numerical representation affects policy-space optimization.

2 Related work

Bandit exploration and numerical policy optimization. Index-based and posterior-sampling algorithms provide strong benchmarks for direct policy computation (Audibert et al., 2009; Ménard & Garivier, 2017; Jin et al., 2021). Adusumilli (2025) computes Bayes and minimax policies from limiting partial differential equations, using a game between decision rules and priors. Our planner retains the finite horizon and generates responses for a policy population. Boutilier et al. (2020) instead optimize differentiable policies on sampled bandit instances; here the environment sampling weights are themselves a strategy in the restricted game.

Imperfect-information game solving. Counterfactual regret minimization optimizes strategies through regret over solver iterations (Zinkevich et al., 2007); bandit regret is the within-episode payoff here. Computational game solving scales to large imperfect-information games (Brown & Sandholm, 2018; Brown et al., 2019; Perolat et al., 2022). Double-oracle methods and policy-space response oracles (PSRO) expand populations through responses to mixtures (McMahan et al., 2003; Lanctot et al., 2017). Our solver expands learner policies on a fixed environment support.

Minimax-regret and Minimax-Bayes reinforcement learning. MIRROR formulates robust reinforcement learning as a minimax-regret agent–nature game and uses a double-oracle procedure with learned responses (Xu et al., 2021). It expands both policy and environment sets. Our finite environment support stays fixed, and reward-model planning generates the learner responses. The closest work is Buening et al. (2023), who explicitly model policy and prior selection as a zero-sum game and develop gradient descent–ascent and a Bayes-oracle cutting-plane method. Their experiments include discounted Bernoulli bandits with Gittins responses and finite-horizon Markov decision processes. We implement bandit responses through finite-horizon sufficient-statistic planning and shared two-arm costs. Our Minimax-Bayes reinforcement-learning (RL) comparator adapts their gradient-based route.

Planning under model uncertainty. Values of finite-state partially observable Markov decision processes admit conditional-vector representations (Smallwood & Sondik, 1973), and PBVI approximates planning using selected reachable beliefs (Pineau et al., 2003). BEETLE exploits an analytic value representation for approximate Bayesian RL planning (Poupart et al., 2006). For bandits, Zhu et al. (2023) numerically solve a continuous-time limiting problem in normalized sufficient statistics. Our Grid retains each finite-horizon layer. Grid and PBVI also apply to Bayesian policy optimization with a prescribed prior.

3 Stochastic minimax bandit games

An environment $\theta \in \Theta$ specifies K reward laws $p_\theta(y \mid a)$ with means $\mu_a(\theta)$ and $\mu_*(\theta) = \max_a \mu_a(\theta)$. Conditional on θ, rewards are independent draws from the selected arms' laws. After

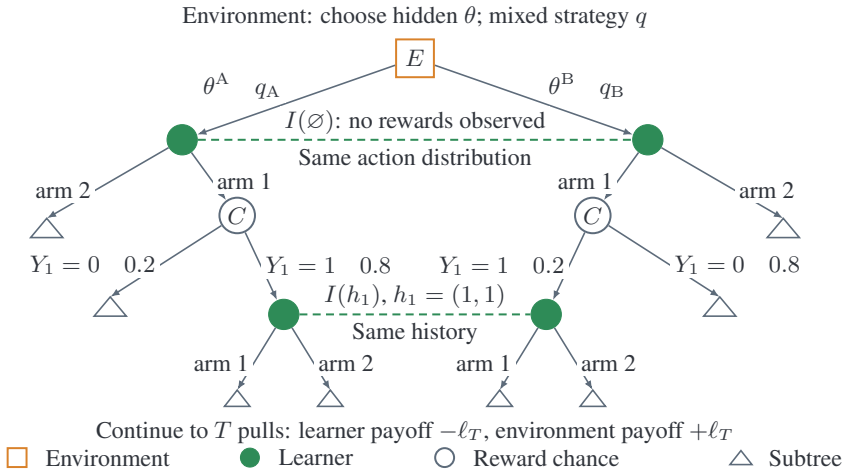

Figure 1: Illustrative stochastic minimax bandit game with Bernoulli means $\theta^{\mathrm{A}} = (0.8, 0.2)$ and $\theta^{\mathrm{B}} = (0.2, 0.8)$. Root weights $q_{\mathrm{A}}, q_{\mathrm{B}}$ mix the hidden models. Dashed links join learner nodes with the same history: $h_1 = (1, 1)$ means arm 1 returned reward 1 under either model. Triangles abbreviate subtrees; colors identify the players.

t pulls, the learner observes $h_t = (A_1, Y_1, \ldots, A_t, Y_t)$; $\mathcal{P}_T$ denotes randomized policies using this history and private randomness. Their expected cumulative pseudo-regret is

$$R_T(\pi, \theta) = \mathbb{E}_{\pi,\theta} \sum_{t=0}^{T-1} [\mu_*(\theta) - \mu_{A_{t+1}}(\theta)]. \tag{1}$$

Definition 1 (Stochastic minimax bandit game)**.** For a reward family indexed by Θ, arm count K, and horizon T, the environment player selects a hidden θ at the root, fixed for the episode. At round $t + 1$, the learner chooses A_{t+1} from h_t, and chance draws $Y_{t+1} \sim p_\theta(\cdot \mid A_{t+1})$, revealing only the selected arm's reward. At termination, the environment receives $\ell_T(\theta, h_T) = \sum_{t=0}^{T-1} [\mu_*(\theta) - \mu_{A_{t+1}}(\theta)]$ and the learner receives $-\ell_T$.

Nodes (θ, h_t) and (θ', h_t) share an information set $I(h_t)$ and must use the same action distribution. Remembering all past actions and observations gives perfect recall. The environment acts strategically only at the root; subsequent reward draws are chance moves. Terminal regret defines the evaluation payoff, while the learner receives only selected-arm rewards as online feedback. Figure 1 illustrates this information structure.

The environment's mixed strategy $q \in \Delta(\Theta)$ is a probability distribution over instances. Its expected payoff is regret averaged under q, $\mathbb{E}_q R_T(\pi, \theta)$. Joint optimization gives the minimax target

$$V_T^*(\Theta) = \inf_{\pi \in \mathcal{P}_T} \sup_{q \in \Delta(\Theta)} \mathbb{E}_q R_T(\pi, \theta) = \inf_{\pi \in \mathcal{P}_T} \sup_{\theta \in \Theta} R_T(\pi, \theta), \tag{2}$$

because averages cannot exceed the supremum and point-mass distributions are available. Thus minimax regret is the learner's minimax objective in this zero-sum game (Lattimore & Szepesvári, 2020; Buening et al., 2023). A fixed q defines the learner's expected-regret response problem. The reward family, K, and T specify which stochastic minimax bandit game is being solved.

4 A SOLVER FOR STOCHASTIC MINIMAX BANDIT GAMES

The solver builds a population of history-dependent policies on a finite environment set. Each iteration plans a response to its environment mixture, evaluates the complete policy in the original game, and solves the expanded payoff matrix (Figure 2).

4.1 RESTRICTED GAME AND RESPONSE GENERATION

Fix M representative environments $\theta^1, \ldots, \theta^M$ and a policy pool $\Pi_r = \{\pi_1, \ldots, \pi_r\}$. For the exact game, each column averages regret over the hidden arm assignments of its representative; write this

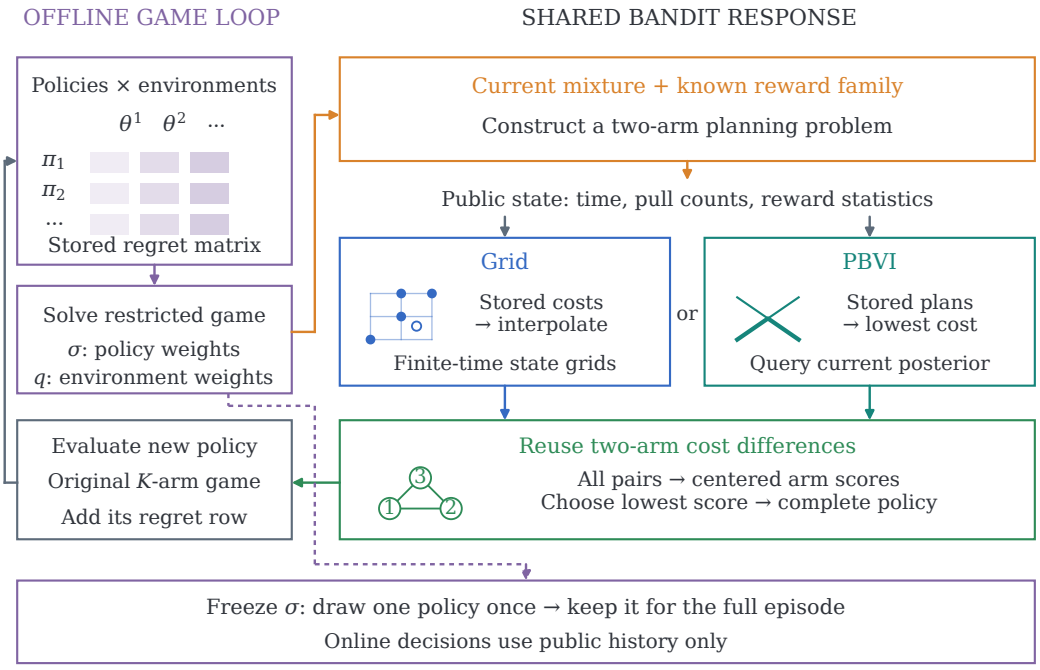

Figure 2: The offline solver alternates response planning, original-game evaluation and mixture optimization. Grid and PBVI share the pair-cost interface. The dashed path deploys the final policy mixture.

payoff as $\overline{R}_T(\pi, \theta^m)$ (Appendix A). A numerical matrix $\widehat{M}^{(r)}$ supports the restricted game

$$\min_{\sigma \in \Delta_r} \max_{q \in \Delta_M} \sigma^\top \widehat{M}^{(r)} q. \tag{3}$$

Here Δ_n is the probability simplex. The learner's mixed strategy σ weights policies, and the environment's mixed strategy q weights instances. Each configuration starts with an empty learned pool and uniform q_0. A response to q_0 is evaluated and added as the first row. Later rounds solve the restricted game, construct a response to its q, evaluate that policy, and add its row. Each pool contains policies from a single configuration. This is PSRO-style one-sided expansion (Lanctot et al., 2017): environment weights adapt on a fixed support. Runs stop at the round budget or when a numerical response repeats.

Solution quality. For the exact finite game, define the policy upper value and the optimal learner-response value against the environment mixture by

$$U_M(\sigma) = \max_m \overline{R}_T(\sigma, \theta^m), \qquad L_M(q) = \inf_{\pi \in \mathcal{P}_T} \sum_m q_m \overline{R}_T(\pi, \theta^m). \tag{4}$$

They bracket the minimax value on the finite set of all represented arm assignments. With $R_T(\sigma, q) = \sum_m q_m \overline{R}_T(\sigma, \theta^m)$, the pair's Nash gap is

$$g_M = U_M - L_M = \underbrace{U_M - R_T(\sigma, q)}_{\text{environment improvement}} + \underbrace{R_T(\sigma, q) - L_M}_{\text{learner improvement}}. \tag{5}$$

The restricted solution gives mutual responses within the matrix; responses from the full history-dependent policy space may improve upon them. The numerical benchmark averages two arm orientations for $K = 2$ and uses ordered representatives for $K > 2$; the latter equals an averaged expectation only for permutation-equivariant policies. The development comparison follows that protocol; separate frozen-policy checks use a common root-relabelling wrapper, and small-game calibration uses exact expectations.

4.2 Response planning from public statistics

Fix the environment mixture and expand it over hidden arm assignments to obtain a labelled prior $\widetilde{q}$. Since the benchmark $T\mathbb{E}_{\widetilde{q}}\mu_*(\theta)$ is independent of the learner, minimizing expected regret is equivalent to maximizing expected cumulative reward under $\widetilde{q}$. Bayesian planning uses the likelihood to predict rewards and update posteriors. Let s collect public sufficient statistics, $b_q(\theta \mid s)$ their posterior, $p_q(\cdot \mid s, a)$ the predictive reward law, and $F(s, a, y)$ the statistic update. The ideal recursion is

$$Q_t(s,a) = \mathbb{E}_{b_q}[\mu_*(\theta) - \mu_a(\theta)] + \mathbb{E}_{Y\sim p_q(\cdot|s,a)} J_{t+1}(F(s,a,Y)), \qquad (6)$$
$$J_t(s) = \min_a Q_t(s,a), \qquad J_T(s) = 0.$$

Gaussian rewards with known variance use counts N_a and reward sums S_a; Bernoulli rewards use N_a and success counts S_a. For fixed-concentration Beta rewards, the evidence is N_a and $L_a = \sum_{u:A_u=a} \log[Y_u/(1-Y_u)]$, not the logit of an empirical mean. Likelihoods are given in Appendix B.

4.3 Numerical cost representations

Grid. At each finite-horizon layer, Grid stores continuation values or action-cost differences in count/evidence coordinates. Multilinear interpolation weights four or eight adjacent nodes at an unstored state. Backups query future values at predicted next evidence; online decisions query action-cost differences. Time layers retain the original horizon independently of grid resolution.

The response models vary with the family. Gaussian Grid uses a positive-gap Gaussian score model. Bernoulli uses exact binary branches for $K = 2$ and a normal-score approximation for $K > 2$; Beta uses its likelihood and numerical reward integration. Table 3 and Appendix B.3 specify the coordinates and operators.

PBVI. Representative reachable posteriors guide conditional-cost backups (Pineau et al., 2003). For m retained prior atoms, each vector $\alpha \in \mathbb{R}^m$ stores a plan's conditional cost at every atom. A backup fixes an action, branches over rewards, and selects next-layer continuations at the updated posteriors. It integrates the resulting costs separately under each atom. Bernoulli branches use the two exact outcomes; continuous families use quadrature under the family likelihood. For remaining horizon h, let $\Gamma_{h,a}$ contain plans starting with action a. The score at the actual posterior b is

$$\widetilde{Q}_h(b,a) = \min_{\alpha\in\Gamma_{h,a}} b^\top \alpha. \qquad (7)$$

Design beliefs choose the saved plans; the online posterior directly weights their costs. Prior compression and vector selection control the representation size (Appendix B.5).

4.4 Constructing multi-arm responses

For $K > 2$, draw an environment from q, choose an unordered arm pair uniformly, and randomize its orientation. This induces a two-arm prior for the shared response. Finite-prior implementations compress this measure with arm-swap symmetry; Gaussian and multi-arm Bernoulli score grids instead use positive-gap reductions (Appendix B.6). For fixed horizon, prior cap and numerical resolution, the shared two-arm value representation needs no additional state coordinates as K grows; each decision uses $O(K^2)$ pair queries.

At a public state, let d_{ij} be the response's cost of selecting arm i minus that of selecting arm j, queried on their ordered statistics. Antisymmetrize and aggregate:

$$D_{ij} = \tfrac{1}{2}(d_{ij} - d_{ji}), \quad D_{ii} = 0, \qquad c_i = \frac{1}{K}\sum_j D_{ij}. \qquad (8)$$

Select an arm with minimum c_i, handling ties symmetrically. These centered scores fit pairwise differences in least squares (Appendix B) and retain the magnitudes of pairwise preferences when selecting an arm. Each pair omits other arms' evidence and future sampling opportunities. Original-game evaluation measures the resulting complete policy's regret and supplies its payoff row. Queries use either global remaining time $T - t$ or pair-local time $T - N_i - N_j$, with selected conventions recorded in the appendix.

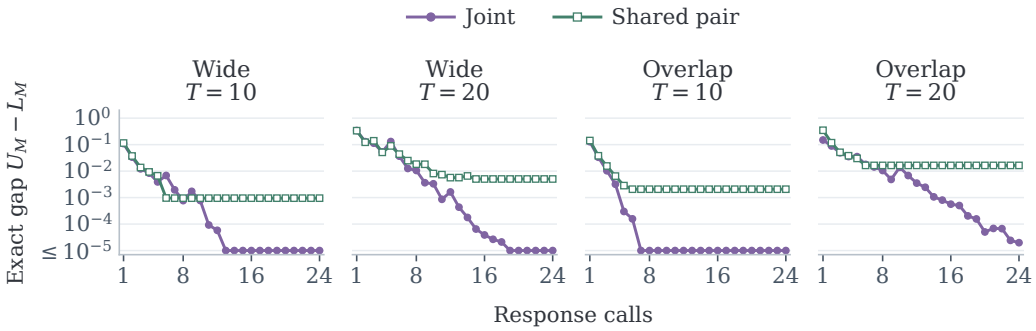

Figure 3: Complete solver iterations on four finite Bernoulli games. Wide and Overlap denote the 35- and 20-column environment supports. Each point evaluates the current mixture and a separate joint response to its environment strategy. All 24 calls are shown; the log-axis floor is 10^{-5}. Final values: Table 7.

Representation cost. PBVI with m retained prior atoms and at most V conditional-cost vectors per layer stores $O(mV)$ coefficients per layer and takes $O(K^2mV)$ arithmetic per multi-arm decision. A larger belief budget adds plans; more prior atoms increase each plan's dimension. Full pair priors can grow with K; fixed-size representations require a prior cap. The experiments vary belief and prior capacities separately. Appendix B.5 gives construction and storage costs.

At deployment, draw one policy component from the final mixture for the whole episode. Its prior, value representation and clock remain fixed; its evidence and actions change with observed rewards. Uniform root relabelling, with a consistent input/output mapping throughout the episode, makes any learner or baseline invariant in expected regret to arm labels (Appendix A).

5 Experiments

Benchmark configurations are selected by setting; controlled studies vary resolution, prior capacity or response targets while fixing other parameters.

Problems and comparators. Gaussian rewards have known standard deviation 1, Bernoulli rewards take values in $\{0, 1\}$, and Beta rewards follow $\mathrm{Beta}(2\mu, 2(1-\mu))$. Each family covers $K \in \{2, 3, 5, 8\}$ and $T \in \{20, 100, 200, 500, 1000\}$. Mean domains are $[-4/\sqrt{T}, 4/\sqrt{T}]$ for Gaussian, $[0, 1]$ for Bernoulli and $[10^{-5}, 1 - 10^{-5}]$ for Beta; the Gaussian sweep fixes a standardized range.

The nine baselines are upper confidence bound (UCB) (Auer et al., 2002), MOSS (Audibert & Bubeck, 2009), UCB-V (Audibert et al., 2009), KL-UCB++ (Ménard & Garivier, 2017), Thompson Sampling (Thompson, 1933), SoftElim (Boutilier et al., 2020), MOTS (Jin et al., 2021), EXP3-IX (Neu, 2015), and a Minimax-Bayes RL adaptation (Buening et al., 2023). Classical parameters are searched and refined per setting. The learned comparator uses 64 updates in the 60-setting sweep. Its optimization-budget study covers 64, 256, and 1,024 updates, three seeds, and two deployment-exploration choices. The primary frozen comparison includes all 1,024-update actors alongside the nine baselines. Appendix C specifies searches, compute budgets and reward-family adaptations.

5.1 Solution quality on exactly evaluable games

Small games allow us to measure how response approximation affects the minimax quality of the final policy mixture. Four Bernoulli games compare joint and uncompressed pairwise responses: $K = 3$, $T = 10, 20$, with 35-column and 20-column supports. Each solver starts with an empty pool and uniform environment weights, then adds 24 responses using exact expected payoff rows. A separate joint response to each saved environment mixture computes L_M.

Pairwise gaps fall from 0.115–0.349 after the first response to 0.00095–0.01642 after 24 calls; joint gaps reach at most 1.99×10^{-5} (Figure 3). Final pairwise worst regret is within 0.35% of joint

Table 1: Frozen worst-regret differences from the strongest reference at $K = 3, T = 100$, with approximate simultaneous 95% intervals. Negative favors the proposed policy. References include nine baselines and all higher-budget RL actors. Evaluation protocol: Appendix C.6.

Family	Best reference (regret)	Grid difference	PBVI difference
Gaussian	RL (11.555)	-0.458 $[-1.470, 0.638]$	-0.898 $[-1.752, 0.077]$
Bernoulli	MOSS (5.433)	0.011 $[-0.475, 0.497]$	-0.157 $[-0.539, 0.317]$
Beta	MOSS (3.252)	0.166 $[-0.170, 0.536]$	0.578 $[0.225, 0.931]$

planning. Because U_M and L_M bracket the finite-game minimax value, the gap also bounds excess worst regret on that game. Pairwise gaps remain above the joint-planning gaps under complete pair-state planning. At $T = 20$, pairwise/joint solver CPU times are 5.08/6.45 seconds on the wide grid and 3.57/5.99 on the overlapping grids; joint diagnostics for all 24 pairs add 2.32–3.32 seconds per run.

For the numerical implementations, complete-state evaluation of Bernoulli $K = 2, T = 20$ gives Grid/PBVI gaps 0.088626/0.093047 on 400 labelled environments. For Grid, the environment mixture is obtained by solving the empirical game at the selected training iteration. Fixed-prior controls further compare magnitude and sign aggregation (Appendices C.3 and C.1).

5.2 Online regret and offline computation

Figure 4 compares selected policies across all 60 settings used for configuration selection (the development benchmark). Grid and PBVI have lower estimated worst regret than the strongest of the nine baselines in 52/60 and 56/60 settings; 28/60 and 47/60 improvements exceed 5%, respectively. Scores use 256 episodes per environment. For $K > 2$, evaluation uses ordered representatives. The frozen comparisons below separately test fixed policies with independent streams and common root relabelling.

Figure 5 follows one fixed middle-resolution configuration per representation and family. All six runs finish below their nine-baseline reference. Final PBVI point regrets are 6–18% lower; complete runs take 5.2–10.0 wall-clock minutes. Grid takes 1.2, 0.8 and 32.8 minutes for Gaussian, Bernoulli and Beta, respectively. Progress slows as the pools expand, with different time–regret profiles across families. Appendix C.12 compares all three resolution levels.

5.3 Frozen-policy comparisons

The primary comparison evaluates Grid/PBVI at $K = 3, T = 100$ against the nine baselines and all 1,024-update RL actors. Configurations and mixture weights are frozen before drawing streams independent of configuration selection. Within each environment, component returns are combined and arm assignments averaged under common episode-root relabelling before taking the environment maximum. The fixed-resolution training curves in Figure 5 form a separate controlled study.

Table 1 summarizes 57 deployments with 6,144 rollout contributions per environment and component. Gaussian PBVI has the largest favorable point difference, with an interval including zero; the approximate Beta PBVI interval favors MOSS. All conservative Hoeffding difference intervals contain zero (Appendix C.6). Frozen tests at $K = 5, T = 100$ and $K = 2, T = 500$ cover six settings and compare with the nine sweep baselines. Grid and PBVI have lower point regret in three and five of these settings, respectively; all approximate difference intervals include zero. Separate environment-transfer tests evaluate policies outside their design sets (Appendix C.8). Appendix D specifies the evaluated policy tables.

Deployment cost. Grid/PBVI take 5.06/8.20, 7.87/11.22 and 45.83/13.20 CPU seconds per million decisions for Gaussian, Bernoulli and Beta, including feedback generation in batched rollouts. MOSS takes 0.14, 0.09 and 3.49 seconds. PBVI has lower measured deployment cost than Grid for Beta; both cost more than MOSS.

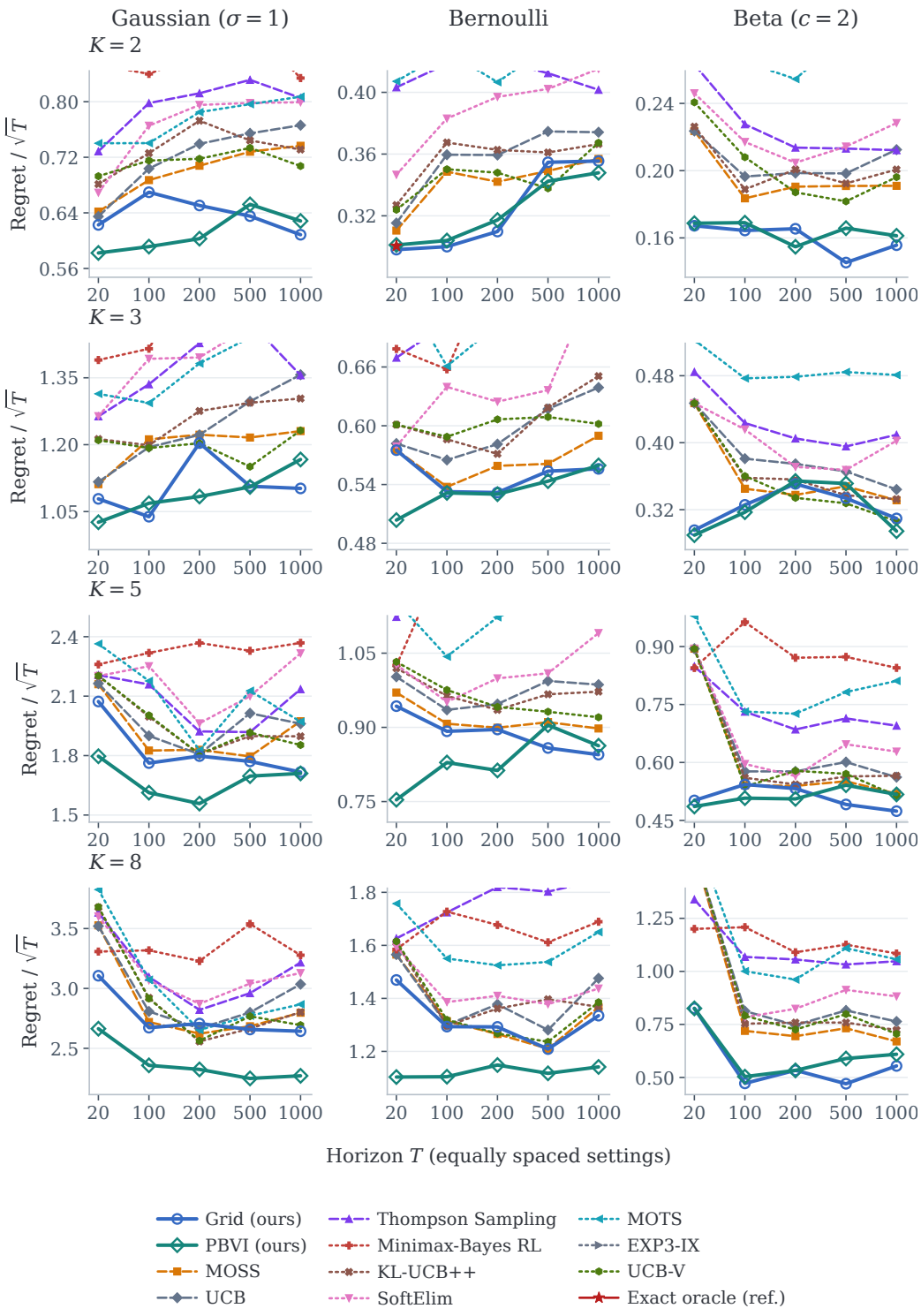

Figure 4: Regret across all 60 development settings: arm counts in rows, reward families in columns. Estimated worst regret is divided by $\sqrt{T}$; lower is better. In the headings, Gaussian σ is reward standard deviation and Beta c is concentration. All nine baselines, Grid and PBVI are plotted; zoomed axes clip higher regrets. Horizons are equally spaced. These selected scores use ordered representatives for $K > 2$, unlike the common root relabelling in Table 1. The exact-planning reference is at Bernoulli $K = 2, T = 20$. Raw values: Table 27.

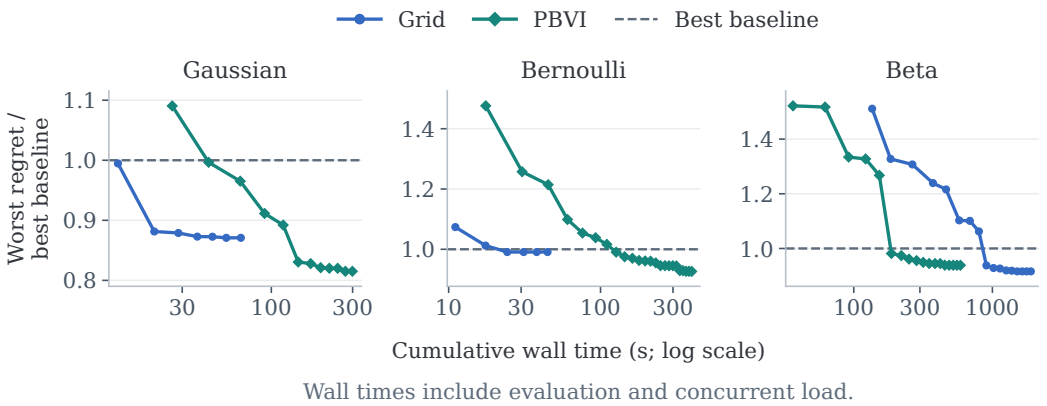

Figure 5: Performance during fixed-configuration training at $K = 3, T = 100$. Grid uses 33×65 nodes for Gaussian/Bernoulli and 33^3 for Beta; PBVI uses 256 design beliefs. Curves include every saved round from an empty pool; total run times also include stopping checks. Regret is divided by the lowest worst-regret estimate among the nine sweep baselines; the grey line marks equality. Cumulative wall time includes payoff evaluation under concurrent load. These training-stream traces summarize fixed-configuration progress; CPU budgets are not matched across runs.

5.4 Allocating response capacity

Response quality depends on both the represented prior and its numerical resolution. At Beta $K = 3, T = 100$ with fixed q, full pair support reduces estimated regret under q from 3.331 to 2.234 at $1.15\times$ construction CPU; quadrupling the belief budget gives 3.260 at $12.46\times$ CPU. Changing support also changes the induced design beliefs (Appendix C.10).

Six-call runs from empty pools test whether this improvement carries into policy-space iteration. Across three paired seeds, full adaptive priors reduce design-set regret from 4.110–4.266 to 2.974–3.063 (best reference: 3.353). All nine full-minus-compressed approximate intervals across three environment sets are negative; design-set comparison intervals with the best reference contain zero. Feasible learner improvements against the final environment mixtures fall from 2.129–2.352 to 0.181–0.248. These feasible deviations do not upper-bound the full-policy Nash gap.

To assess environment adaptation, responses target either the current mixture or a fixed uniform prior; both conditions solve the restricted game for the output policy. At $K = 3, T = 100$ with compressed PBVI priors, adaptive-minus-uniform differences over three paired seeds are -2.767 ± 0.406, -2.684 ± 0.162, and -3.118 ± 0.588 for Gaussian, Bernoulli, and Beta (mean $\pm$ sample standard deviation; six calls per run). With full Beta priors, uniform targets give lower point regret on both the structured stress set and the Sobol evaluation set for all seeds; only the third seed's approximate difference intervals exclude zero. On the Sobol set, one adaptive seed has regret 4.183 versus a reference of 3.183, although all full-prior runs improve over compressed priors.

Adaptive targets cost 378–402 training CPU seconds, versus 2,422–2,594 for uniform targets at the same six-call budget. Full-prior adaptive deployment takes 101–121 CPU seconds per million decisions versus 14–27 for compressed priors and 5.56 for MOSS. Thus the benefit and cost of environment adaptation depend on the response representation (Appendices C.9 and C.11).

6 Discussion and conclusion

Our solver computes policy mixtures for stochastic minimax bandit games. The resulting online learning algorithms improve on tuned references in most development settings. Exact small-game gaps quantify solution quality; independent comparisons quantify regret differences and uncertainty. Training curves and response controls relate policy quality to computation and prior capacity. Extending gap evaluation to larger games is a natural next step.

Reproducibility statement

The supplementary code and data contain the 60 benchmark environment sets, 540 selected baseline configurations, and 109 policy deployments. They support response training, evaluation of fixed policies, exact small-game calculations, and recomputation of the reported statistical comparisons. The appendix specifies numerical representations, parameter searches and evaluation protocols. Appendix D describes the scope of the accompanying materials; execution instructions are provided with the code.

AI use statement

Generative AI tools assisted with method implementation, experiment design and execution support, interpretation of numerical results, mathematical derivations, literature searches, figure creation, and manuscript drafting. We are responsible for the correctness, originality, and attribution of all resulting claims and artifacts.

References

Karun Adusumilli. Risk and optimal policies in bandit experiments. *Econometrica*, 93(3):1003–1029, 2025. doi: 10.3982/ECTA21075. URL https://onlinelibrary.wiley.com/doi/10.3982/ECTA21075.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009. URL https://www.learningtheory.org/colt2009/papers/022.pdf.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19):1876–1902, 2009. doi: 10.1016/j.tcs.2009.01.016. URL https://www.sciencedirect.com/science/article/pii/S030439750900067X.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2–3):235–256, 2002. doi: 10.1023/A:1013689704352. URL https://link.springer.com/article/10.1023/A:1013689704352.

Craig Boutilier, Chih-Wei Hsu, Branislav Kveton, Martin Mladenov, Csaba Szepesvári, and Manzil Zaheer. Differentiable meta-learning of bandit policies. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/hash/171ae1bbb81475eb96287dd78565b38b-Abstract.html.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018. doi: 10.1126/science.aao1733. URL https://www.science.org/doi/10.1126/science.aao1733.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 793–802. PMLR, 2019. URL https://proceedings.mlr.press/v97/brown19b.html.

Thomas Kleine Buening, Christos Dimitrakakis, Hannes Eriksson, Divya Grover, and Emilio Jorge. Minimax-Bayes reinforcement learning. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pp. 7511–7527. PMLR, 2023. URL https://proceedings.mlr.press/v206/buening23a.html.

John C. Gittins. Bandit processes and dynamic allocation indices. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(2):148–164, 1979. doi: 10.1111/j.2517-6161.1979.tb01068.x. URL https://academic.oup.com/jrsssb/article/41/2/148/7027626.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830.

Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1):203–244, 2011. doi: 10.1007/s10107-010-0419-x. URL https://link.springer.com/article/10.1007/s10107-010-0419-x.

Tianyuan Jin, Pan Xu, Jieming Shi, Xiaokui Xiao, and Quanquan Gu. MOTS: Minimax optimal Thompson sampling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5074–5083. PMLR, 2021. URL https://proceedings.mlr.press/v139/jin21d.html.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985. doi: 10.1016/0196-8858(85)90002-8. URL https://www.sciencedirect.com/science/article/pii/0196885885900028.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Perolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/3323fe11e9595c09af38fe67567a9394-Abstract.html.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020. doi: 10.1017/9781108571401. URL https://www.cambridge.org/core/books/bandit-algorithms/8E39FD004E6CE036680F90DD0C6F09FC.

H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *Proceedings of the 20th International Conference on Machine Learning*, 2003. URL https://www.cs.cmu.edu/~ggordon/mcmahan-ggordon-blum.icml2003.pdf.

Pierre Ménard and Aurélien Garivier. A minimax and asymptotically optimal algorithm for stochastic bandits. In *Proceedings of the 28th International Conference on Algorithmic Learning Theory*, volume 76 of *Proceedings of Machine Learning Research*, pp. 223–237. PMLR, 2017. URL https://proceedings.mlr.press/v76/m%C3%A9nard17a.html.

Gergely Neu. Explore no more: Improved high-probability regret bounds for non-stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/hash/e5a4d6bf330f23a8707bb0d6001dfbe8-Abstract.html.

Julien Perolat, Bart de Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T. Connor, Neil Burch, Thomas Anthony, Stephen McAleer, Romuald Elie, Sarah H. Cen, Zhe Wang, Audrunas Gruslys, Aleksandra Malysheva, Mina Khan, Sherjil Ozair, Finbarr Timbers, Toby Pohlen, Tom Eccles, Mark Rowland, Marc Lanctot, Jean-Baptiste Lespiau, Bilal Piot, Shayegan Omidshafiei, Edward Lockhart, Laurent Sifre, Nathalie Beauguerlange, Remi Munos, David Silver, Satinder Singh, Demis Hassabis, and Karl Tuyls. Mastering the game of Stratego with model-free multiagent reinforcement learning. *Science*, 378:990–996, 2022. doi: 10.1126/science.add4679. URL https://arxiv.org/abs/2206.15378.

Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Point-based value iteration: An anytime algorithm for POMDPs. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, pp. 1025–1030, 2003. URL https://www.ijcai.org/Proceedings/03/Papers/147.pdf.

Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete Bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, 2006. URL https://cs.uwaterloo.ca/~ppoupart/publications/bayesian-rl/icml-brl-8pages.pdf.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016. URL https://jmlr.org/papers/v17/14-087.html.

Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973. doi: 10.1287/opre.21.5.1071. URL https://pubsonline.informs.org/doi/10.1287/opre.21.5.1071.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3–4):285–294, 1933. doi: 10.1093/biomet/25.3-4.285. URL https://doi.org/10.1093/biomet/25.3-4.285.

Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 257–267. PMLR, 2021. URL https://proceedings.mlr.press/v161/xu21a.html.

Yuhua Zhu, Zachary Izzo, and Lexing Ying. Continuous-in-time limit for Bayesian bandits. *Journal of Machine Learning Research*, 24(305):1–35, 2023. URL https://jmlr.org/papers/v24/22-1181.html.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, volume 20, 2007. URL https://papers.nips.cc/paper_files/paper/2007/hash/08d98638c6fcd194a4b1e6992063e944-Abstract.html.

A GAME FORMULATION AND ADDITIONAL SOLVER DETAILS

A.1 FINITE ENVIRONMENT COLUMNS

For computation, let $\theta^1, \ldots, \theta^M$ denote representative mean vectors and let Θ_M^{perm} contain all their arm permutations. Arm order is hidden from the learner. A column in the ideal finite game draws a uniform permutation of its representative, with payoff

$$\overline{R}_T(\pi, \theta^m) = \frac{1}{K!} \sum_{\rho \in S_K} R_T(\pi, \rho\theta^m). \tag{9}$$

Here q denotes weights on representatives: the environment assigns mass q_m to representative m, hence mass $q_m/K!$ to each hidden assignment, counting multiplicities. Uniformly relabelling a policy once per episode yields the same expected regret on every permutation of an instance. This shows that the permutation-averaged game has minimax value $V_T^*(\Theta_M^{\mathrm{perm}})$.

Deployment under arbitrary arm labels. This symmetry has a direct implementation for any learner or baseline. At the start of an episode, independently draw a uniform permutation ρ of the K arm indices. Present physical arm $\rho(a)$ to the policy as internal arm a: its observations update that internal arm's history, and an internal action a is executed as physical action $\rho(a)$. Keep this mapping fixed throughout the episode, including initialization and tie breaking. A policy mixture also keeps its sampled component fixed. This wrapper requires neither the arm means nor knowledge of an environment's hidden assignment, and its expected regret is unchanged by any external relabelling of the environment. Thus label dependence can be handled at deployment without changing a policy's learned parameters. Merely renaming an environment and its policy by the same known permutation is instead a change of notation; the independent randomization above handles unknown assignments at deployment.

The main benchmark uses two-orientation averages for two arms and ordered representative scores for more arms. The exact column definition and the numerical evaluation convention are distinct; the latter is stated in Section 5. The equivalence under episode-level symmetrization is proved in Appendix B.

A.2 RESPONSE OBJECTIVE FOR A FIXED ENVIRONMENT MIXTURE

Let $\widetilde{q}$ be the prior on labelled environments obtained by expanding the fixed representative weights q over hidden permutations. Minimizing expected regret is equivalent to maximizing Bayesian expected reward, because

$$\mathbb{E}_{\theta \sim \widetilde{q}} R_T(\pi, \theta) = T\mathbb{E}_{\widetilde{q}}\mu_*(\theta) - \mathbb{E}_{\widetilde{q},\pi} \sum_{t=0}^{T-1} \mu_{A_{t+1}}(\theta), \tag{10}$$

and the first term does not depend on π. This equivalence defines the response objective; numerical and pairwise approximations determine how we solve it.

A.3 GRID COORDINATES AND WORKED RESPONSE QUERIES

Beta Grid uses the fixed-concentration Beta likelihood. After at least one pull of either arm, its three coordinates are

$$x_N = \frac{N_2}{N_1 + N_2}, \qquad x_a = \frac{1}{2} + \frac{1}{\pi} \arctan\left(\frac{L_a/N_a}{\pi/\sqrt{3}}\right), \quad a \in \{1, 2\}, \tag{11}$$

with $x_a = 1/2$ when $N_a = 0$ and a separate root evaluation when both counts are zero. A regular grid is used at each time layer, with interpolation and Gauss–Jacobi observation integration. The likelihood is exact for the specified model, while the grid, endpoint representation, and quadrature remain numerical approximations.

Illustrative Bernoulli backup: two pulls remaining

Models (arm 0, arm 1): A = (0.8, 0.2), B = (0.2, 0.8)

Current belief: P(A) = 0.75, P(B) = 0.25; no pulls yet

Action	Immediate regret		Expected future regret		Total cost
Arm 0	0.15	+	0.65 × 0.0462 (reward 1) + 0.35 × 0.2571 (reward 0)	→	$Q_0 = 0.270$ choose arm 0
Arm 1	0.45	+	0.35 × 0.2571 (reward 1) + 0.65 × 0.0462 (reward 0)	→	$Q_1 = 0.570$

Each term: P(reward) × best one-pull cost after updating belief (rounded).

Figure 6: Illustrative Bernoulli backup with two pulls remaining. Each action combines immediate expected regret with observation-weighted optimal cost for the last pull, giving $Q_0 = 0.270$ and $Q_1 = 0.570$. Displayed continuation costs are rounded. Local labels $0, 1$ correspond to arms $1, 2$.

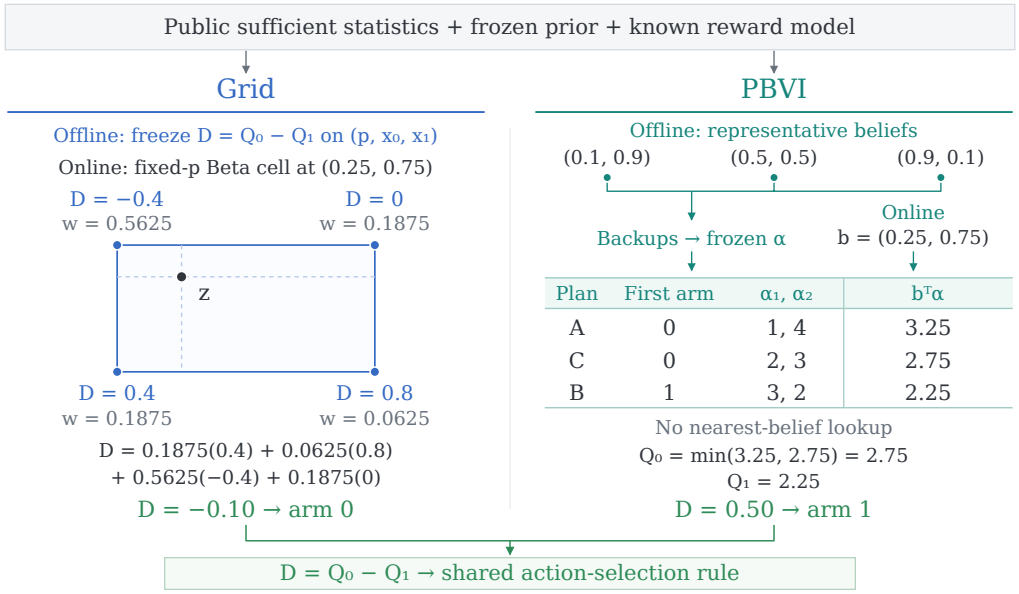

Figure 7: Two numerical representations of action costs, queried from public statistics and a frozen response prior. Left: a fixed-allocation slice of the Beta grid. A query one-quarter across and three-quarters up the cell gives $D = 0.1875(0.4) + 0.0625(0.8) + 0.5625(-0.4) + 0.1875(0) = -0.10$. The full three-dimensional query also interpolates along $p = x_N$; evidence coordinates are defined in Equation (11). Right: representative beliefs generate conditional cost vectors offline. Each row's (α_1, α_2) gives its costs under two environment hypotheses. The online belief $b = (0.25, 0.75)$ scores the saved vectors at $3.25, 2.75, 2.25$. Minimizing within each first action gives $Q_0 = 2.75$, $Q_1 = 2.25$, and $D = 0.50$, without a nearest-belief lookup. Both representations supply a cost difference to the same pairwise action-selection rule. Numbers are illustrative and local arm labels are $0, 1$.

A.4 Two-arm distribution and multi-arm aggregation

For $K > 2$, the shared response is trained using a two-arm distribution induced by q. Draw an environment, choose an unordered physical arm pair uniformly, and randomize its orientation. Before

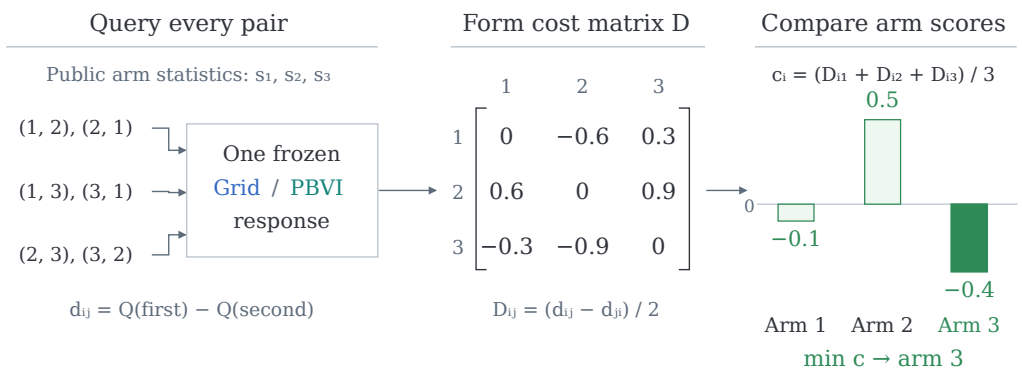

Figure 8: Reusing one frozen two-arm response on three physical arm pairs. Antisymmetrized differences populate D; row averages are $(-0.1, 0.5, -0.4)$. The score plot shares a zero axis, and the smallest score selects arm 3. Every pair is queried in both arm orders. The three-arm simulator evaluates this policy to produce a new row of the regret game. The numerical example illustrates a pairwise approximation to joint planning.

finite compression, this gives

$$q^{(2)} = \sum_{m=1}^{M} \frac{q_m}{2\binom{K}{2}} \sum_{i<j} \left[\delta_{(\mu_i(\theta^m),\mu_j(\theta^m))} + \delta_{(\mu_j(\theta^m),\mu_i(\theta^m))}\right]. \tag{12}$$

The finite-prior responses compress this measure while maintaining arm-swap symmetry; absolute means and zero-gap pairs enter the measure. The Gaussian score response and the Bernoulli response based on local asymptotic normality (LAN) instead use positive-gap reductions, as detailed in Appendix B.6. The resulting prior and numerical response are frozen together.

For the antisymmetrized pair-cost matrix D, the scores in Equation (8) solve

$$c = \operatorname*{arg\,min}_{\sum_i u_i = 0} \sum_{i<j} [(u_i - u_j) - D_{ij}]^2. \tag{13}$$

This is a complete-graph least-squares ranking problem (Jiang et al., 2011). Appendix B gives the derivation and the effect of retaining cost magnitudes.

B Method algebra and implementation distinctions

B.1 Fixed-prior regret and Bayesian reward

Write $(w_\ell, \vartheta^\ell)_\ell$ for the labelled atoms of the expanded prior $\widetilde{q}$. When q indexes representative environments, these atoms include all hidden arm assignments with their weights $q_m/K!$; identical atoms may be combined. For any non-anticipating policy, the environment parameter is constant over the episode and $\mathbb{E}[Y_{t+1} \mid \theta, h_t, A_{t+1}] = \mu_{A_{t+1}}(\theta)$. Consequently,

$$\begin{aligned}\mathbb{E}_{\widetilde{q}} R_T(\pi, \theta) &= \sum_{t=0}^{T-1} \mathbb{E}_{\widetilde{q},\pi}[\mu_*(\theta) - \mu_{A_{t+1}}(\theta)] \\ &= T \sum_\ell w_\ell \mu_*(\vartheta^\ell) - \mathbb{E}_{\widetilde{q},\pi} \sum_{t=0}^{T-1} Y_{t+1}.\end{aligned} \tag{14}$$

The first term is policy independent. The minimizers of expected pseudo-regret and the maximizers of Bayesian expected reward therefore coincide whenever these expectations exist. The identity holds for a fixed environment distribution regardless of the accuracy of the numerical response.

Given sufficient statistics s, the posterior and predictive law are

$$b_q(\vartheta^\ell \mid s) = \frac{w_\ell \mathcal{L}_\ell(s)}{\sum_j w_j \mathcal{L}_j(s)}, \qquad p_q(y \mid s, a) = \sum_\ell b_q(\vartheta^\ell \mid s) p_{\vartheta^\ell}(y \mid a). \tag{15}$$

For positive-probability histories, action-selection factors in the history likelihood do not depend on the unknown environment once the public history is fixed. Thus they cancel in the posterior. Applying the law of total expectation to the next observation yields (6). For a two-arm response the same recursion is defined on a two-arm prior, with the two-arm maximum as its regret reference. This reference is constant across policies under that prior; changing the prior and the available future actions is what separates the response problem from the full-K problem.

B.2 Family-specific likelihoods

The following expressions are likelihoods of the observed history with parameter-independent action-selection factors removed. They remain valid under adaptive allocation. Arms are conditionally independent given θ.

Gaussian. For known common variance τ^2 and $Y \mid a, \theta \sim \mathcal{N}(\mu_a(\theta), \tau^2)$, expansion of the quadratic log density gives

$$\mathcal{L}(\theta; s) \propto \exp\left\{ \sum_a \frac{\mu_a(\theta) S_a - \frac{1}{2} N_a \mu_a(\theta)^2}{\tau^2} \right\}, \qquad S_a = \sum_{u: A_u = a} Y_u. \tag{16}$$

Squared-reward sums contribute only a parameter-independent factor, so $(N_a, S_a)_{a=1}^K$ is sufficient.

Bernoulli. For $Y \mid a, \theta \sim \mathrm{Bernoulli}(\mu_a(\theta))$,

$$\mathcal{L}(\theta; s) = \prod_a \mu_a(\theta)^{S_a} (1 - \mu_a(\theta))^{N_a - S_a}, \qquad S_a = \sum_{u: A_u = a} Y_u. \tag{17}$$

Endpoint parameters use the corresponding limiting probabilities. A Bernoulli Bellman backup has exactly two observation branches, with probabilities $1 - \mu_a(\theta)$ and $\mu_a(\theta)$ conditional on an atom.

Fixed-concentration Beta. Write $\mu = \mu_a(\theta) \in (0, 1)$ and fix $\nu > 0$. The density is

$$f_\mu(y) = \frac{y^{\nu\mu - 1} (1 - y)^{\nu(1-\mu) - 1}}{B(\nu\mu, \nu(1 - \mu))}, \qquad 0 < y < 1. \tag{18}$$

Since

$$(\nu\mu - 1) \log y + (\nu(1 - \mu) - 1) \log(1 - y) = (\nu\mu - 1) \log \frac{y}{1 - y} + (\nu - 2) \log(1 - y), \tag{19}$$

the last term is independent of μ, giving

$$\mathcal{L}(\theta; s) \propto \exp\left\{ \sum_a \left[(\nu\mu_a(\theta) - 1) L_a - N_a \log B(\nu\mu_a(\theta), \nu(1 - \mu_a(\theta))) \right] \right\}. \tag{20}$$

For the implemented $\nu = 2$ model, this is exactly the log-weight formula used by the mean-logit Grid response. An unpulled arm has $N_a = L_a = 0$ and contributes no likelihood evidence. Raw reward sums are not sufficient for this likelihood, although an implementation may retain them as auxiliary public quantities.

B.3 Grid coordinates, stored fields, and query rules

For a query z inside a d-dimensional cell, let $\eta_j \in [0, 1]$ be its relative position between the two bracketing nodes of axis j. If D_ε is the saved cost difference at corner $\varepsilon \in \{0, 1\}^d$, the query is

$$\widetilde{D}(z) = \sum_{\varepsilon \in \{0,1\}^d} D_\varepsilon \prod_{j=1}^{d} \eta_j^{\varepsilon_j} (1 - \eta_j)^{1 - \varepsilon_j}. \tag{21}$$

The implementations below use $d = 2$ or 3 and apply the same multilinear rule to continuation values within a Bellman backup.

Table 2: Response variants. The coordinates below define u, e_a, x_a; p is allocation to the second queried arm. Selected clocks appear in Appendix B.6.

Variant	Response model	Frozen query representation
Gaussian Grid	Symmetric positive-gap Gaussian model	Bilinear $Q_1 - Q_2$ on (p, u)
Bernoulli Grid, $K > 2$	Normal score model at mean $1/2$	Bilinear $Q_1 - Q_2$ on (p, u)
Bernoulli Grid, $K = 2$	Labelled prior, binary observations	Trilinear $Q_1 - Q_2$ on (p, e_1, e_2)
Beta Grid	Labelled prior, Beta likelihood	Trilinear cost differences on (p, x_1, x_2)
PBVI, all families	Labelled prior, family-specific likelihood	Action-indexed cost vectors evaluated at b

Gaussian and multi-arm Bernoulli score grids. For an ordered pair (i, j), let $m = N_i + N_j$, $p = N_j/m$, and

$$x = \frac{N_i N_j}{m\sqrt{T}} \left(\frac{E_j}{N_j} - \frac{E_i}{N_i} \right), \qquad u = \mathrm{sigmoid}(2\delta_{\mathrm{ref}} x). \tag{22}$$

Set $p = 1/2$ if $m = 0$, and $x = 0$ if either arm is unpulled. For Gaussian noise standard deviation τ, $E_a = (S_a - N_a\mu_0)/\tau$; the reference center cancels in x. For Bernoulli, $E_a = 2S_a - N_a$ is the standardized likelihood score at $\mu_0 = 1/2$. The retained signed-gap prior ρ has $\delta = \pm|\mu_j - \mu_i|\sqrt{TI_0}$, with $I_0 = 1/\tau^2$ for Gaussian and $I_0 = 4$ for Bernoulli. δ_{ref} is its smallest positive active magnitude; u is a reference coordinate, not generally the full posterior probability.

At depth n, the response model uses

$$\mathcal{I}(n, p) = \frac{n}{T} p(1-p), \qquad b(\delta \mid n, p, x) \propto \rho(\delta) \exp\{\delta x - \tfrac{1}{2}\delta^2 \mathcal{I}(n, p)\}. \tag{23}$$

For the first or second action, respectively, set $p' = np/(n+1)$ or $p' = (np+1)/(n+1)$ and $v = \mathcal{I}(n+1, p') - \mathcal{I}(n, p)$. The modeled evidence update is $x' = x + \delta v + \sqrt{v}Z$, $Z \sim \mathcal{N}(0, 1)$. A backup averages the interpolated next value over b and a nine-node normal Gauss–Hermite rule, and adds $[\delta]_+/T$ or $[-\delta]_+/T$ for the two actions. Endpoint evidence nodes remain at their respective boundaries. Costs are normalized by $\tau\sqrt{T}$ for Gaussian and by $\sqrt{T}/2$ for the Bernoulli score model.

Allocation nodes are $p_k = [1 - \cos(\pi k/(P-1))]/2$ and evidence nodes are $u_l = l/(U-1)$. Backward computation stores values $\min(Q_1, Q_2)$; deployment stores $\Delta_n = Q_1 - Q_2$ and interpolates it bilinearly. There is no interpolation across integer time layers. Pair-local queries use $n = m$; global queries use $n = t$ with the same physical p, x. Global querying thus changes the encoded information $np(1-p)/T$ as well as the continuation horizon.

Two-arm Bernoulli grid. The labelled variant uses both success coordinates:

$$p = N_2/t, \qquad e_a = \frac{1}{2} + \frac{\mathrm{asinh}((S_a - N_a/2)/\sqrt{N_a})}{2\,\mathrm{asinh}(\sqrt{N_a}/2)}. \tag{24}$$

Unpulled arms use $e_a = 1/2$. A uniform dyadic grid uses $2^\ell + 1$ nodes on each axis. Inverting a node gives $N_1 = t(1-p)$, $N_2 = tp$ and

$$S_a = N_a/2 + \sqrt{N_a}\sinh\left[(2e_a - 1)\,\mathrm{asinh}(\sqrt{N_a}/2)\right]. \tag{25}$$

Backups maintain conditional continuation costs for each environment atom, use exact binary branches, and weight the action costs by the node posterior. Deployment stores their difference at nonroot layers and a separate root action, using the eight trilinear corner weights for lookup. The base policy breaks an exact tie toward its first labelled arm; the random orientation at the start of each episode makes deployment symmetric.

Beta coordinate control. The mean-logit coordinate is (11). The square-root-count logit control uses $x_a^{\mathrm{sqrt}} = 1/2 + \arctan(L_a/\sqrt{N_a})/\pi$, taking $1/2$ at zero count. Thus the comparison changes the evidence normalization, including its scale, while retaining the 17^3 grid and other stated training settings. The same-round comparison removes the difference in final stopping rounds.

B.4 THE MEAN-LOGIT GRID AND PBVI SCORES

The Beta Grid chart in (11) has a fixed scale $\pi/\sqrt{3}$ in all selected settings. Its inverse at a positive count is

$$L_a = N_a \frac{\pi}{\sqrt{3}} \tan\left(\pi\left(x_a - \frac{1}{2}\right)\right). \tag{26}$$

For inverse evaluation, endpoint coordinates are clipped to $[\epsilon, 1-\epsilon]$ with $\epsilon = 10^{-10}$ so that grid nodes have finite evidence representatives. Counts of zero always map to zero evidence. The same chart is used in direct two-arm and selected pairwise Beta Grid responses. A configuration such as $17 \times 17 \times 17$ specifies the node counts along the allocation axis and the two evidence axes at each time layer. Gauss–Jacobi integration approximates the continuous observation expectation under the Beta model; exact sufficient statistics do not remove this approximation or interpolation error.

For PBVI, α_k represents a numerical conditional cost at frozen prior atom k. At a selected design belief, a backup considers a first action, updates the posterior at observation branches, chooses a minimizing retained continuation vector for each branch, and adds the immediate regret costs. Backups and their arm-swapped counterparts form a finite vector set, with duplicate vectors removed. The stored root-action labels permit separate minima for the two first actions in (7). Retained posterior points determine which backups are performed, while a deployment posterior is evaluated directly by dot products. The number of posterior points is therefore not the number of nearest-neighbor action entries. For continuous observations, these vectors incorporate numerical quadrature and are not used here to assert rigorous policy-value bounds.

B.5 PBVI DESIGN POINTS AND RETAINED CONTINUATIONS

Generating reachable posteriors. The shared two-arm PBVI implementation generates design trajectories under the frozen response prior, using independent uniformly random actions on each base path. For a posterior-point budget P, it generates $2P$ paths in three strata: P from the full prior, $P/2$ from its highest-mass unordered pair type, and $P/2$ from its largest-gap pair type. An unordered pair type groups a labelled pair of means with its arm-swapped counterpart. Each path is paired with a path that swaps the hidden means, complements every action, and uses the same reward realization. Exact family likelihoods convert each partial history into a posterior over the frozen prior's atoms.

Selecting points at each depth. At every positive elapsed depth, the three strata contribute respectively $P/2$, $P/4$, and $P/4$ posterior rows, counting both arm orientations. The root has only the prior. Within each stratum, selection starts with the candidate nearest the prior and then greedily selects the candidate farthest from the already selected set. For beliefs b, c, the distance is the smaller of their Fisher–Rao distance and the distance after swapping one belief's arm labels:

$$d(b,c) = \min\left\{2\arccos\sum_j \sqrt{b_j c_j},\ 2\arccos\sum_j \sqrt{b_j c_{\tau(j)}}\right\}, \tag{27}$$

where τ maps each prior atom to its arm-swapped atom. Both orientations of a selected candidate are retained. Identical posterior rows can occupy separate slots; P is a row budget rather than a guarantee of P distinct beliefs. Thus $P = 256$ uses 512 simulated paths and 256 design rows at each positive depth. A backup with h decisions remaining uses the rows at elapsed depth $T - h$.

Backing up and retaining vectors. Starting from the zero continuation vector at horizon zero, the backup at a design belief forms one candidate for each root action. At each observation branch it updates the belief and chooses the retained continuation with the smallest posterior-weighted cost. It keeps the better root-action candidate and its arm-swapped vector, with opposite root-action label. At the symmetric root prior both action labels are retained. Deduplication removes vectors only when they have the same root-action label and agree componentwise to absolute tolerance 2×10^{-12}; no additional linear programs remove dominated vectors. The design beliefs choose which plans to construct, whereas online queries evaluate the action-specific lower envelopes in (7), without projecting to a design belief.

Table 3: Response implementations. Each maps pair statistics and remaining time to a two-arm action-cost difference. They share this interface; their response models and numerical approximations differ.

Implementation	Response model	Stored representation / query
Gaussian Grid	Symmetric positive-gap Gaussian score model; nine-point quadrature	Allocation and score; bilinear interpolation
Bernoulli Grid, $K = 2$	Labelled prior; exact binary observation branches	Allocation and two success-evidence coordinates; trilinear interpolation
Bernoulli Grid, $K > 2$	Normal-score approximation around mean $1/2$; nine-point quadrature	Allocation and score; bilinear interpolation
Beta Grid	Beta likelihood; numerical observation integration	Allocation and two mean-logit coordinates; trilinear interpolation
PBVI, all families	Family likelihood on retained prior atoms; binary branches or five-point quadrature per atom and action	Conditional-cost vectors; lower envelope at the current posterior

Integrating observations. Bernoulli backups enumerate rewards 0 and 1 exactly. In the shared Gaussian and Beta implementation, five-point Gauss–Hermite and Gauss–Jacobi rules, respectively, integrate observations conditional on each prior atom and action. Five points therefore means five per atom and action, not five for the whole predictive mixture. For every such observation, likelihoods under all prior atoms determine the updated belief and the selected continuation; the integral uses that continuation's component for the source atom. Zero-weight branches are skipped. If a positive conditional branch is impossible under a design belief with zero mass on its source atom, its otherwise undefined posterior is replaced by selecting the lowest-cost continuation for that source atom; the choice does not affect the cost at the design belief. This is numerical observation integration, separate from the finite selection of design posteriors.

Computational dimensions. For one two-arm response, let m be the number of labelled prior atoms, P the design-row budget, V_h the retained-vector count at remaining horizon h, $V = \max_h V_h$, and J the observation nodes per atom and action. The saved vectors use $m \sum_{h=0}^{T} V_h$ scalar entries, plus action labels. In the shared constructor, swap-paired design rows retain at most P vectors at each non-root, nonterminal layer; the root and terminal layers have two and one, respectively. The resulting bound is $m[(T-1)P + 3]$ entries. A two-arm query costs $O(mV)$ arithmetic; the multi-arm wrapper makes $K(K-1)$ directed pair queries, sharing one table, for $O(K^2 mV)$ arithmetic per decision. During construction, each candidate scores up to mJ updated posteriors against the preceding vectors. These dense products contribute $O(TPJm^2V)$ arithmetic, in addition to trajectory generation, belief selection and deduplication. Here $J = 2$ for Bernoulli and $J = 5$ in the shared continuous-family implementation.

B.6 Pair-prior compression and response variants

The support-capped PBVI implementation first discards environment masses at most 10^{-12} after normalization, renormalizes, and aggregates unordered pair types from (12). Each type is represented by its center and nonnegative gap. For a labelled support-size cap r (the rank cap), the unordered type budget is $u = \lfloor r/2 \rfloor$. If compression is required, types are sorted by center, then gap, and selected at cumulative-mass quantiles $(j + 1/2)/u$, $j = 0, \ldots, u-1$. Repeated selections are removed; any shortfall is filled by farthest-point selection in Euclidean center–gap coordinates, each scaled by its observed range (floored at 10^{-12}). Each original type assigns its mass to its nearest representative in these coordinates. Representatives are selected from the existing pairs. Finally, each unequal pair receives its swapped copy with half its type mass, while equal pairs need only one atom. The resulting labelled prior has at most r atoms, possibly fewer when equal-mean pairs survive. Zero-gap types enter this procedure but are not guaranteed to remain separate atoms after compression. In particular, the procedure preserves arm-swap symmetry, not every posterior or every Bayes value of the original pair distribution.

The selected two-arm Beta PBVI runs use a different prior-construction route: they directly symmetrize the active environment prior, without center–gap rank compression. They share the design-

posterior construction and backup implementation described above. Their trajectory seed changes with the response index, whereas the shared Gaussian/Bernoulli implementation uses a fixed setting-level trajectory seed across responses. Its prior, and hence its design posterior values, still changes as the outer game updates q. The selected PBVI configurations must therefore be read per setting; they are not all a single rank-16 configuration.

The Gaussian score-table construction instead reduces the pairwise measure to gap magnitudes, conditions on positive gaps, and uses Gaussian location evidence. The Bernoulli $K > 2$ Grid results similarly use a positive-gap score construction with a Fisher-information-based LAN Gaussian response approximation around the reference mean $1/2$. By contrast, Bernoulli PBVI responses use a finite-prior posterior and binary observation branches based on the Bernoulli likelihood. The two-arm Bernoulli Grid response also uses a fully labelled Bernoulli table. The selected Beta Grid responses use the mean-logit coordinates and the fixed-concentration Beta likelihood.

The selected multi-arm clocks are recorded below. "Global" means $h = T - t$ and "pair-local" means $h = T - N_i - N_j$ for a queried pair. Here Gaussian, Bernoulli, and Beta refer to the three reward families, and a setting is specified by its arm count and horizon.

Response implementation	Selected clock convention for $K > 2$
Gaussian score Grid	Global for $K = 3, 5$; pair-local for $K = 8$.
Bernoulli normal-score Grid	Global except $(K, T) = (8, 500)$, which uses pair-local.
Selected Beta Grid	Pair-local in every selected multi-arm setting.
Gaussian/Bernoulli PBVI	Global in every selected multi-arm setting.
Beta PBVI	Global except $(K, T) = (3, 1000), (5, 1000)$, which use pair-local.

For $K = 2$, both clocks coincide because $t = N_1 + N_2$. A multi-arm global-clock query can have fewer pair observations than its elapsed-time layer; PBVI still evaluates the actual pair posterior against that layer's vectors. These are choices of the proposed policy, not identities between two-arm and full-K continuation values. Point counts, rank caps, and round budgets also vary in the selected configurations, particularly for Beta PBVI. The comparison is not a controlled single-factor ablation of Grid versus PBVI.

B.7 Least-squares meaning of the arm scores

Let D be any real antisymmetric $K \times K$ matrix with zero diagonal. Define $F(u) = \sum_{i<j}[(u_i - u_j) - D_{ij}]^2$ under the constraint $\sum_i u_i = 0$. Differentiating and using antisymmetry gives

$$\frac{1}{2}\frac{\partial F}{\partial u_i} = \sum_{j \neq i}(u_i - u_j - D_{ij}) = K u_i - \sum_j u_j - \sum_j D_{ij}. \tag{28}$$

The gradient components sum to zero, so the Lagrange multiplier for the centering constraint vanishes. The stationary solution is therefore $u_i = K^{-1}\sum_j D_{ij}$. It is itself centered because $\sum_{i,j} D_{ij} = 0$. Uniqueness follows from

$$\sum_{i<j}(v_i - v_j)^2 = K\sum_i v_i^2 - \left(\sum_i v_i\right)^2 = K\|v\|_2^2 \quad \text{when } \sum_i v_i = 0. \tag{29}$$

Thus (13) is a direct property of the implemented aggregation. The residual $D_{ij} - (c_i - c_j)$ has zero row sums and does not affect the final costs. Magnitudes still matter: for $K = 3$, the triples $(D_{12}, D_{13}, D_{23}) = (-1, -1, -1)$ and $(-1, -1, -100)$ have identical pairwise signs but select arms 1 and 2, respectively. Replacing cost differences by pairwise winner labels would therefore change the policy.

Under a relabelling of arms, the shared ordered query relabels d, explicit antisymmetrization relabels D, and row averaging relabels c. A symmetric minimum-cost tie rule then relabels the action probabilities. This establishes permutation equivariance of the construction, without an optimality claim for the full-K problem.

B.8 WHY THE MIXTURE IS SAMPLED ONLY ONCE

For a frozen root mixture and fixed environment, the complete trajectory law is

$$\mathbb{P}_\sigma(h_T \mid \theta) = \sum_z \sigma_z \mathbb{P}_{\pi_z}(h_T \mid \theta). \tag{30}$$

An implementation that removes the stored component index would need the history-conditional weights $\mathbb{P}(Z = z \mid h_t)$, rather than the original σ_z, to reproduce the next-action distribution. These weights generally change after the observed actions. Keeping the sampled component index is the direct implementation of the mixture in the restricted game. The two-arm Bernoulli Grid component's arm-swap randomization is likewise fixed for the episode. Component identities, frozen priors, numerical tables or vectors, clock conventions, and the selected mixture weights jointly specify the deployed policy.

B.9 SCOPE OF NUMERICAL GAME VALUES

Equations (4)–(5) define exact game values for permutation-averaged columns. To see why their minimax value equals that on Θ_M^{perm}, let π^{sym} uniformly relabel the input and output arm names of π once per episode. Then $R_T(\pi^{\mathrm{sym}}, \rho\theta^m) = \overline{R}_T(\pi, \theta^m)$ for every ρ. For any π, its maximum permutation-averaged regret is at most its maximum labelled regret; conversely, the symmetrized policy attains that averaged regret on every labelled permutation. Taking infima establishes equality of the two minimax values. It also identifies $U_M - L_M$ with the pair gap for σ^{sym} and $\widetilde{q}$.

The empirical worst-regret estimate is $\widehat{U} = \max_m \sum_\ell \sigma_\ell \widehat{M}_{\ell m}$. For $K > 2$ its columns evaluate only stored ordered representatives. Equality with a permutation-averaged expectation is guaranteed when the policy's entire action law, including initialization and ties, is permutation equivariant. This is not inferred for all baselines from the symmetry of their index formulas alone.

Initialization versus ties. Pulling every arm once in a fixed order is not stepwise equivariant, but its total regret is $K\mu_* - \sum_a \mu_a$, invariant to labels. With independent stationary rewards, the resulting evidence also permutes in distribution. Thus, for $T \geq K$, label-neutral indices and a symmetric tie rule suffice for invariant expected total regret after this initialization. Under ideal continuous observations, the same holds if all subsequent ties have probability zero; finite numerical implementations need not inherit that property.

Discrete rewards can create positive-probability ties. As an illustrative counterexample, consider two Bernoulli arms with means $p > q$, horizon 3, one initial pull per arm, and a final empirical-mean choice breaking ties toward the lower index. The two confidence bonuses are equal at that decision for the MOSS and UCB-V updates described below. If the better arm has the lower index, the final inferior-pull probability is $(1-p)q$; if it has the higher index, it is $1-p(1-q)$. Their expected-regret difference is $(p-q)[pq+(1-p)(1-q)]$, generally positive. This is a counterexample to a universal invariance claim, not a measurement of any reported setting. The common episode-root wrapper removes dependence on external labels without requiring a separate argument for each initialization and tie rule.

Under the theoretical column payoff, a feasible response's expected cost is at least $L_M(q)$, so it cannot establish a lower bound on the game's value. Section 5.1 instead computes the Bayes minimum by complete discrete-state planning on the small Bernoulli problem. For a larger environment domain $\Theta \supseteq \Theta_M^{\mathrm{perm}}$, exact $L_M(q)$ remains a lower bound on $V_T^*(\Theta)$, but $U_M(\sigma)$ is only an upper bound for the finite-set game; a continuous-domain upper bound additionally requires maximizing over the larger domain.

C ADDITIONAL EXPERIMENTAL DETAILS AND NUMERICAL RESULTS

Reading guide. The experiments are organized by scientific question. Complete-state controls (Appendices C.1–C.4) measure small-game gaps, pairwise-reduction loss and its effect on full policy-space iterations. The primary frozen $K = 3, T = 100$ comparison (Appendix C.6) evaluates Grid/PBVI against the nine benchmark references and every 1,024-update RL actor. Cross-setting comparisons and additional-environment tests extend this evaluation (Appendices C.7–C.8). The

Table 4: Complete discrete-state calibration at Bernoulli $K = 2, T = 20$. Frozen-mixture values U, L use Equation (4) for this finite game, separately from the development Monte Carlo estimates.

Selected pair	Policy regret U	Bayes response L	Pair gap $U - L$
Grid	1.326424	1.237798	0.088626
PBVI	1.317024	1.223978	0.093047

capacity and target-update controls (Appendices C.9–C.12) distinguish single responses at fixed q from complete solver runs, including compressed versus full pair priors and adaptive versus uniform targets. The final sections specify environment sets, configuration selection, learned-comparator optimization budgets and the 60-setting development benchmark. Reproducibility coverage and the identity of regenerated policies are specified in Appendix D. In compact setting labels, G, B, and C denote Gaussian, Bernoulli, and Beta rewards, respectively: for example, G3–T100 means Gaussian rewards with $K = 3$ arms and horizon $T = 100$. CPU times measure processor time; wall-clock times measure elapsed time.

C.1 Complete-state calibration and additional response values

For Bernoulli $K = 2, T = 20$, the benchmark has 400 stored labelled environments, including both arm orientations. Backward dynamic programming over all sufficient-statistic states computes a Bayes response for each fixed environment mixture. Posterior likelihoods retain endpoint parameters 0 and 1. The frozen Grid and PBVI policies are evaluated by the corresponding finite-state recursion, component by component, before applying episode-level mixture weights and arm-swap averaging. These are complete discrete-state expectations in floating-point arithmetic, without trajectory sampling or state interpolation in the evaluator.

Grid's environment mixture is obtained by solving the empirical game formed by the policy pool at its selected checkpoint. Since the training-time environment weights were not saved and the linear-program (LP) optimum may be nonunique, the calibrated pair uses the environment weights from this solve. PBVI uses its saved selected-checkpoint mixture.

The selected Grid and PBVI mixtures admit learner-side improvements of 0.036380 and 0.028311 under their respective priors; environment-side deviations contribute the remainder of the gaps in Table 4. The exact-expectation payoff table differs from the sampled table used to choose the mixtures, which are not reoptimized during this calibration. The 256-point PBVI responses are within 0.001118 and 0.002545 of the Bayes optima under the Grid and PBVI priors, respectively. Thus response accuracy and the equilibrium gap of a deployed pair measure different aspects of the solver.

Complete-state evaluation. Action-probability tables cover 8,855 nonterminal sufficient states; adding terminal states gives 10,626 states. Each positive-weight component is evaluated before episode-level mixture weights and the Grid arm-swap lottery are applied to whole-policy values. Backward induction computes each Bayes response using exact Bernoulli endpoint likelihoods and zero prior masses. The same recursion evaluates the 256-point response tables used for the reported response errors. Appendix D specifies which construction and evaluation computations are reproducible.

C.2 Three-arm response accuracy on a short-horizon problem

To compare pairwise planning with an unrestricted joint response, we use Bernoulli $K = 3, T = 10$ and the 50 representative mean vectors from the $K = 3, T = 100$ environment set. Three fixed priors weight those representatives: uniform, the Grid mixture obtained as in Appendix C.1, and the saved PBVI mixture. The latter two priors are transferred without optimization for the shorter horizon. Each representative is expanded over all six arm assignments, with its probability divided equally among them. The full-joint Bayes dynamic program enumerates 5,005 nonterminal sufficient-statistic states (8,008 including terminal states) and integrates every Bernoulli outcome.

For each prior, we build two pairwise PBVI responses with 256 posterior points, a global clock, and pair-support caps 16 and 32. These are new $T = 10$ responses with 11 value layers, not truncated

$T = 100$ policies. Construction of all six responses uses seed 93300010. Evaluation integrates each complete action-probability table exactly over all states and assignments, without Monte Carlo sampling.

Table 5: Short three-arm response check. Excess cost is the fixed-prior expected regret of the approximate response minus the full-joint Bayes minimum L. The two support caps are fixed comparisons, not a rank search. These are response errors for the new $T = 10$ problem, not gaps for the main $T = 100$ strategy pairs.

Prior	Exact Bayes L	Excess, cap 16	Excess, cap 32
Uniform	0.781952	0.017425	0.002045
Transferred Grid	1.080012	0.000000	0.000000
Transferred PBVI	0.999870	0.000356	0.000201

The broader uniform prior has the largest measured loss, which falls when the support cap increases. Both responses attain the reference value for the transferred Grid prior, within floating-point precision. The excess combines belief-point approximation, pair-support compression, and pairwise decomposition; the remaining error cannot be attributed to decomposition alone. The uniform prior is still compressed at cap 32, whereas the transferred PBVI prior retains all ten pair orbits.

C.3 Exact pairwise reduction and aggregation control

To separate pairwise reduction from numerical response approximation, we use the same 50 Bernoulli representatives and three priors as the preceding short-horizon check, with predeclared horizons $T = 10, 15, 20$. The transferred Grid and PBVI priors are not reoptimized for these horizons. Every comparison uses the same full three-arm game, averaging a hidden root assignment over all six permutations. Neither Monte Carlo evaluation nor a policy-pool search is used.

The reference solves the complete joint-state Bellman recursion. The first pair policy uses the uncompressed pair prior in Equation (12), merging only exactly identical mean pairs. Complete-state two-arm planning integrates each Bernoulli observation exactly. For a queried pair, its posterior uses only that pair's counts and successes, while its remaining planning horizon is $T - t$ for global elapsed time t. Thus the response also includes queries with pair elapsed time smaller than t. Antisymmetrized cost differences D_{ij} are aggregated as in Equation (8). The second pair policy uses the identical costs but replaces each D_{ij} by its sign before aggregation. Absolute differences at most 10^{-9} are zero in the sign control; both policies use the same 10^{-9} symmetric action-tie rule.

Table 6: Exact pairwise mechanism results for all nine prior–horizon cases. Prior names identify the weight sources. The last two columns give excess expected regret above L, the full-joint optimum for the indicated fixed prior. Machine-scale values below 10^{-12} display as zero. Complete-state expectations have no Monte Carlo sampling error.

Prior	T	Joint optimum L	Magnitude excess	Sign excess
Uniform	10	0.781951905	0.000096495	0.002551819
	15	1.050831651	0.001275769	0.004191745
	20	1.281726066	0.002523183	0.007042513
Transferred Grid	10	1.080011542	0	0
	15	1.488489060	0	0.000032524
	20	1.843275685	0.000001818	0.000021557
Transferred PBVI	10	0.999870301	0	0.000109473
	15	1.400052464	0.000500603	0.000827975
	20	1.760931967	0.002234512	0.003874215

All policies are evaluated under their own induced histories using the full three-arm posterior recursion. The magnitude policy's excess measures the *combined* pairwise reduction: the marginal pair prior, omission of other-arm evidence and future sampling opportunities, and aggregation. It does not isolate these three sources individually. Comparing magnitude with signs, in contrast, changes only the aggregation input.

The magnitude policy has smaller regret in eight cases and ties in one (Table 6). Under the uniform prior, its excess grows from 0.0000965 to 0.0025232 as T increases from 10 to 20, respectively 0.012% and 0.197% of the joint optima. At $T = 20$, its action-distribution total variation from the joint policy averages 3.88% over time under joint-optimal visitation; magnitude versus signs gives 2.48% under that same visitation.

The complete joint state counts, including terminal states, are 8,008, 54,264, and 230,230. The corresponding two-arm state–remaining-time backups number 2,002, 11,628, and 42,504. At $T = 20$ under the uniform prior, the joint and pair solves took 1.139 and 0.0528 CPU seconds in this run; building both lifted action tables took 0.0175 seconds. These exclude shared state/lookup construction and evaluation, whose times are separately recorded. "Exact" here means complete discrete-state integration in 64-bit floating-point arithmetic (float64) rather than interval-certified arithmetic.

Magnitude versus signs in a frozen main-horizon deployment. A control fixes the five positive-weight PBVI components, priors, clocks, tables and root weights at Bernoulli $K = 3, T = 100$, replacing D_{ij} by literal signs with $\mathrm{sign}(0) = 0$. All variants share new reward and action quantiles on 50 representatives, six root assignments and 64 episodes per assignment and component. Component costs are weighted before assignment averaging and environment maximization. Worst regret is 5.281721 for magnitudes and 5.315701 for signs: the difference 0.033980 has descriptive 95% interval $[-0.201410, 0.296332]$, leaving performance unresolved. A prespecified threshold sensitivity sets $|D_{ij}| \le 10^{-9}$ to zero before taking signs, retains the symmetric action-tie rule, and reuses the same magnitude rollouts. It gives regret 5.314626 and difference 0.032905 with interval $[-0.201296, 0.292796]$. Both intervals use 1,000 paired resamples within assignment, preserve episode pairing across methods, environments and components, and recompute both maxima. They describe conditional Monte Carlo variation without guaranteed coverage.

Additional arm-count and prior controls include a symmetric one-good-arm family and independent and coupled three-point priors at $K = 2, 3, 4$ and $T = 8, 12, 16$: nine symmetric-family comparisons and 18 richer-prior comparisons. Magnitude and sign aggregation agree in the symmetric family; their ordering varies with the prior in the richer families.

C.4 Complete policy-space iterations with joint and pairwise responses

This control connects fixed-prior response accuracy to the complete policy-space optimization loop. Both solvers use Bernoulli rewards with $K = 3$, horizons $T \in \{10, 20\}$, and the same finite environment columns. The *wide mean grid* contains the 35 sorted triples drawn with replacement from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The *two overlapping mean grids* contain the ten sorted triples from $\{0.1, 0.35, 0.6\}$ and the ten from $\{0.4, 0.65, 0.9\}$, giving 20 columns. Triples mixing the two grids are excluded. These specify environment supports; the environment weights are optimized during each run.

Each of the eight runs starts with an empty policy pool and uniform mass on its representative columns. At each of 24 rounds, the current environment strategy is expanded over the six hidden arm assignments. The joint condition solves the complete three-arm Bellman recursion. The pairwise condition induces the full ordered-pair prior, solves the two-arm recursion, and applies magnitude aggregation with global remaining time $T - t$. Both use the same restricted-game linear programs after adding the new policy's regret row. No policy is imported from another configuration, and repeated responses remain in the fixed-budget trajectory.

All payoff rows use complete-state Bernoulli expectations. Exchangeable response priors and symmetric action ties make the constructed policies permutation-equivariant, so their expected regret on an ordered representative equals its arm-assignment average. An implementation check verifies this identity explicitly across all six assignments. The learner mixture weights the regret rows, corresponding to one policy draw per episode. At each saved restricted-game solution, U_M is the maximum mixture regret and a separate full-joint Bellman calculation gives $L_M(q)$. In particular, the final lower value uses the environment mixture after row 24 has been added. Their difference is the finite-support strategy-pair gap.

The implementation uses float64 complete-state recursions and a 10^{-9} action-tie tolerance. The strict Bellman minimum supplies L_M, while payoff evaluation uses the actual tolerance-tied policy.

Timing separates state/lookup preparation, prior construction, response planning, pairwise policy lifting, exact payoff evaluation and linear programming, together with bookkeeping and output. End-to-end solver cost includes a fresh joint evaluation lattice for each run; full-joint gap diagnostics are recorded separately. The pairwise condition is expanded into a joint-state action table for exact evaluation, and this lifting cost is included. The shared lifting routine also computes an unused sign-policy table, whose cost remains included.

Table 7: Final strategy-pair values after all 24 response calls. Wide denotes the 35-column support and Overlap the 20-column support. Pair uses the shared uncompressed magnitude response. All values are complete-state expectations; gaps below 10^{-12} are displayed as zero.

Support	T	Response	U_M	$L_M(q)$	$U_M - L_M$
Wide	10	Joint	1.525333	1.525333	0.000000
Wide	10	Pair	1.525996	1.525044	0.000953
Wide	20	Joint	2.146036	2.146036	0.000000
Wide	20	Pair	2.150402	2.145359	0.005043
Overlap	10	Joint	1.563239	1.563239	0.000000
Overlap	10	Pair	1.564348	1.562272	0.002076
Overlap	20	Joint	2.043026	2.043006	0.000020
Overlap	20	Pair	2.050076	2.033651	0.016424

Table 7 reports the final values for the trajectories in Figure 3. Pairwise U_M is within 0.35% of the joint solver in all four problems, with strategy-pair gaps of 0.000953–0.016424. Joint gaps are at most 1.984×10^{-5}.

Table 8: CPU seconds for the complete 24-call solver and, separately, all 24 full-joint gap diagnostics. Solver time includes setup, response construction, policy lifting, exact payoff evaluation, linear programs and bookkeeping. Every run is sequential, with one thread for numerical linear algebra and one for linear programming.

Support	T	Response	Solver CPU (s)	Gap diagnostics CPU (s)
Wide	10	Joint	0.316	0.095
Wide	10	Pair	0.260	0.084
Wide	20	Joint	6.446	2.334
Wide	20	Pair	5.083	2.408
Overlap	10	Joint	0.240	0.069
Overlap	10	Pair	0.235	0.064
Overlap	20	Joint	5.986	3.316
Overlap	20	Pair	3.568	2.316

Pairwise solver CPU time is 2%–40% lower across the four matched comparisons (Table 8). These measurements use complete-state floating-point expectations on the two finite supports at $K = 3$, $T = 10, 20$.

C.5 Shared frozen-policy evaluation protocol

Frozen-policy studies hold every checkpoint, parameter, component and mixture weight fixed before collecting their evaluation streams. For each environment, they combine complete component returns using the frozen episode-level weights, average over arm assignments, and only then maximize over environments. This integrates the initial component lottery, not an action-by-action mixture, and evaluates the common root-relabelling wrapper in Appendix A. Each study below specifies its finite sets, comparator family, strata and episode budget. Appendix D identifies the four deployments with regenerated tables; scores evaluate the specified frozen policies.

Methods share primitive reward quantiles to couple Monte Carlo estimates. Evaluation retains implementation-specific observation handling: in Beta evaluations, the Grid/PBVI evaluator clips

endpoints produced by the inverse cumulative distribution function into the open unit interval, whereas the unified baseline evaluator retains raw endpoints. Shared quantiles therefore need not give bitwise-identical observations.

Uncertainty after maximizing over environments. For representative j, episode pseudo-regret lies in $[0, W_j]$, where $W_j = T(\max_a \mu_{ja} - \min_a \mu_{ja})$. This remains true for Gaussian observations and convex averages of component episode returns. Let $\widehat{r}_{ij}$ be method i's mean estimate. Hoeffding's inequality (Hoeffding, 1963) and a union bound give simultaneous mean intervals, clipped to $[0, W_j]$, with radius

$$W_j \sqrt{\frac{\log(2C/\alpha)}{2n}}, \qquad \alpha = 0.05, \tag{31}$$

where C is the number of scalar means in the study's coverage family and n its contributions per mean. Contributions are independent across episodes and fixed assignment strata under the usual independent-random-stream assumption; they need not have identical distributions. Dependence across methods and environments does not affect the union bound. Zero-gap rows are exactly zero.

Taking maxima of mean lower endpoints and of mean upper endpoints bounds each policy's worst expected regret. Taking minima across the study's declared reference set bounds its best-reference benchmark. Subtracting these intervals bounds candidate excess over that benchmark, accounting for both extrema. A strict improvement is resolved by this construction only when the difference interval lies below zero.

For approximate precision, 2,000 stratified bootstrap replicates resample episodes within each block and assignment, sharing indices across methods and environments. A global studentized maximum of centered mean deviations yields approximate simultaneous mean bands, propagated through the same extrema. These are not ordinary percentile intervals for a nonsmooth maximum and have no finite-sample coverage guarantee; degenerate empirical rows can understate uncertainty. Both constructions concern frozen policies and finite sets, not training-seed variability or regret outside those sets. Studies retain separate coverage families unless explicitly identified as joint.

C.6 Primary frozen comparison and rollout cost at $K = 3, T = 100$

Table 1 gives the candidate-versus-expanded-reference comparisons and their approximate intervals. Table 9 groups the point regrets and rollout costs by reward family, retaining both frozen mixtures, all nine benchmark references, and the lowest-regret 1,024-update actor in each family.

The frozen $K = 3, T = 100$ deployments use 64 Gaussian, 50 Bernoulli and 58 Beta design environments. Four independent blocks of 256 episodes under each of six arm assignments give 6,144 contributions per environment and component, with the same budget for every deployment. Aggregation and numerical observation conventions follow the shared protocol in Appendix C.5.

The comparison includes the two frozen mixtures and all nine benchmark references in each family. The benchmark RL reference uses a selected checkpoint within a 64-update budget, width 64 and deployment $\epsilon = 0.1$: finite-support environment weights for Gaussian/Beta and product-Beta parameters for Bernoulli. It also includes every 1,024-update actor from Appendix C.16: three training seeds, each with deployment exploration $\epsilon = 0.1$ and $\epsilon = 0$. Bernoulli includes both finite-support and product-Beta training branches. Consequently there are 17 Gaussian, 23 Bernoulli, and 17 Beta deployments. Each actor is evaluated separately; the 57 entries involve no ensemble construction. The primary comparator is the minimum worst regret among the nine benchmark references and every 1,024-update actor in the family, as in Table 1. The nine-reference-only contrast is a diagnostic subset, not a second primary benchmark; all actor means remain in the simultaneous family.

Uncertainty. The simultaneous family contains $17 \cdot 64 + 23 \cdot 50 + 17 \cdot 58 = 3{,}224$ scalar means. We use the within-stratum standard errors and mean-band projection described in Appendix C.5. A stratified bootstrap with 2,000 replicates resamples episode columns within each block and assignment, sharing multiplicities across methods and environment rows within a reward family. Families are resampled independently. The maximum standardized mean deviation supplies the common critical value. These approximate bands and the separate bounded-mean Hoeffding bands are propagated through the environment maximum and baseline minimum. Coverage is defined within this study; its intervals do not supply joint coverage with the low-budget stream in Appendix D or the

	Gaussian		Bernoulli		Beta	
Deployment	Regret	CPU	Regret	CPU	Regret	CPU
Grid†	11.097	5.06	5.444	7.87	3.418	45.83
PBVI	10.657	8.20	5.276	11.22	3.830	13.20
MOSS	11.664	0.14	5.433	0.09	3.252	3.49
KL-UCB++	11.642	0.16	5.728	0.33	3.333	3.63
UCB-V	11.823	0.15	5.733	0.09	3.273	0.63
UCB	12.090	2.97	5.638	0.08	3.479	0.61
Thompson Sampling	13.099	0.24	7.165	2.41	4.055	32.88
SoftElim	12.809	0.19	5.948	0.18	3.628	0.72
MOTS	12.916	0.23	6.623	0.21	4.513	0.75
EXP3-IX	24.674	0.23	15.212	0.21	14.914	0.93
Minimax-Bayes RL	14.699	11.39	7.334	12.55	6.495	8.34
Best 1,024-update RL‡	11.555	11.09	5.654	6.66	5.430	12.80

Table 9: Point regret and rollout CPU seconds per million decisions at $K = 3, T = 100$. Each entry is a distinct deployment, not a method ensemble. CPU measures component-weighted batched rollouts, including feedback generation. †: regenerated Gaussian Grid tables (Appendix D). ‡: minimum point regret among all 1,024-update actors within each reward family; all three selected actors use finite-support training and seed 923303, with $\epsilon = 0$ for Gaussian/Bernoulli and $\epsilon = 0.1$ for Beta. The simultaneous uncertainty calculation retains all 57 deployments (3,224 scalar means).

other frozen evaluations. The Hoeffding bands condition on the frozen policies and finite sets, under the independent-episode assumption stated above.

Measured cost. Each component has a separate warm-up of two episodes on each of two environments. We then record parent-plus-child process CPU during the complete batched rollouts, including reward generation, statistic updates and worker overhead. For component costs C_j and mixture weights σ_j, we report $\sum_j \sigma_j C_j$ per million decisions: the expected execution cost of drawing one complete policy at the root. The integrated evaluator instead incurs $\sum_j C_j$ because it evaluates every component. These weights combine measurements of homogeneous-component batches; different online batching can change the realized cost. Numerical processes use one thread; timed evaluation parallelizes existing environment batches across two workers for Gaussian/Bernoulli and eight for Beta. CPU measurements describe these implementations and batch sizes, rather than isolated action-query latency or training costs.

Results. The primary point reference is the finite-support 1,024-update actor (seed 923303, $\epsilon = 0$) for Gaussian and MOSS for Bernoulli and Beta. All Hoeffding difference intervals include zero. The analysis includes all 57 deployments individually.

C.7 Frozen comparisons across arm counts and horizons

The cross-setting comparison fixes all three reward families at $K = 5, T = 100$ and $K = 2, T = 500$ before collecting outcomes. It uses the selected Grid/PBVI deployments, all nine benchmark reference configurations and the stored design environments. At $K = 5$, two blocks evaluate eight episodes for each of all 120 root permutations, giving 1,920 stratum contributions per environment and component. At $K = 2$, two blocks evaluate 512 episodes for each of the two permutations, giving 2,048 contributions. Both use the shared root-relabelling and aggregation protocol. The uncertainty family contains $C = 11 \sum_c M_c$ scalar means across these six settings, where M_c is the number of environments in setting c. The Hoeffding and 2,000-replicate studentized-bootstrap bands use Appendix C.5; their 95% simultaneous scope is this six-setting study only.

Cross-setting results. Tables 10 and 11 report all 66 method–setting evaluations. Grid and PBVI have lower point regret than the best of nine baselines in three and five of the six settings, respectively. All twelve approximate bootstrap and Hoeffding difference intervals contain zero. This study uses a separate 95% uncertainty family.

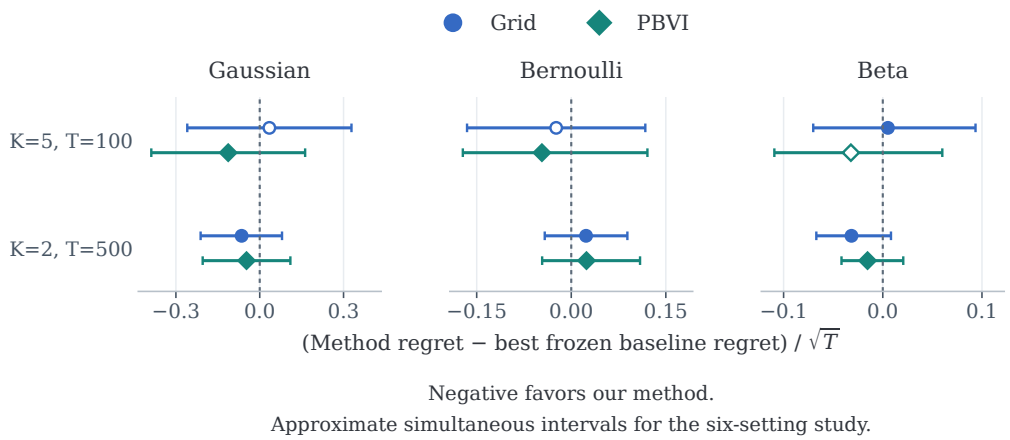

Figure 9: Frozen-policy comparison at six settings: the three reward families at $K = 5, T = 100$ and $K = 2, T = 500$. Negative favors the proposed method over the best of nine benchmark references. Differences and approximate simultaneous intervals are divided by $\sqrt{T}$; all include zero. This six-setting study has its own 95% coverage family. Open markers identify regenerated numerical tables (Appendix D).

Table 10: Frozen-policy evaluation at six additional settings: worst mean cumulative pseudo-regret for all eleven methods. G, B, and C denote Gaussian, Bernoulli, and Beta rewards; the following number is K. † identifies regenerated policy tables documented in Appendix D.

Method	G5-T100	B5-T100	C5-T100	G2-T500	B2-T500	C2-T500
MOSS	18.3728	8.9026	5.6810	15.9075	7.7590	4.4579
UCB	19.1796	9.3310	5.8391	16.8752	8.3019	4.6396
Thompson Sampling	19.9430	11.7336	7.2746	18.5280	9.4790	4.9742
Minimax-Bayes RL	22.3001	11.3454	9.7907	20.1891	17.2955	11.9679
KL-UCB^{++}	18.6133	9.6399	5.6798	16.4044	7.9966	4.4488
SoftElim	19.9962	9.4219	5.9359	18.0520	8.9869	5.0517
MOTS	19.7161	10.4180	7.4171	17.6274	9.6278	6.2108
EXP3-IX	39.4758	28.0747	25.5627	34.9419	25.2081	23.4008
UCB-V	18.1079	9.3876	5.5707	15.5794	7.6808	4.1046
Grid	18.4531†	8.6635†	5.6224	14.1306	8.2059	3.3990
PBVI	16.9800	8.4371	5.2491†	14.5220	8.2204	3.7610

Table 11: Frozen-policy evaluation at six additional settings: candidate minus the minimum worst regret among all nine baselines. Negative is favorable. Approximate bootstrap and finite-sample Hoeffding intervals each apply simultaneously within this study; the primary comparison, this six-setting study, the low-budget stream and the structured-stress test have separate 95% families. †: regenerated policy tables (Appendix D).

Setting	Method	Difference	Approximate 95%	Hoeffding 95%
G5-T100	Grid†	0.3452	[-2.5935, 3.2839]	[-6.8058, 7.1461]
G5-T100	PBVI	-1.1279	[-3.8810, 1.6253]	[-7.6764, 5.3121]
B5-T100	Grid†	-0.2391	[-1.6529, 1.1747]	[-2.8064, 3.1217]
B5-T100	PBVI	-0.4655	[-1.7199, 1.2081]	[-3.3022, 3.6480]
C5-T100	Grid	0.0518	[-0.7013, 0.9350]	[-3.9574, 3.5117]
C5-T100	PBVI†	-0.3216	[-1.0932, 0.6001]	[-4.2842, 3.9037]
G2-T500	Grid	-1.4488	[-4.7300, 1.7845]	[-11.9199, 9.2823]
G2-T500	PBVI	-1.0574	[-4.5655, 2.4533]	[-11.6103, 6.7426]
B2-T500	Grid	0.5251	[-0.9404, 1.9906]	[-13.5011, 15.7975]
B2-T500	PBVI	0.5396	[-1.0323, 2.4399]	[-13.4069, 16.6776]
C2-T500	Grid	-0.7056	[-1.4983, 0.1847]	[-11.0965, 11.4853]
C2-T500	PBVI	-0.3437	[-0.9308, 0.4617]	[-11.2599, 11.7145]

C.8 Evaluation on additional environments

Structured-stress design. A structured-stress test evaluates the $K = 3, T = 100$ Grid/PBVI deployments and all nine benchmark references on additional environment vectors. Each family receives 128 scrambled Sobol triples plus 36 structured vectors: one-best and two-best configurations at three normalized centers $0.2, 0.5, 0.8$ and six normalized gaps $0.01, 0.025, 0.05, 0.1, 0.15, 0.2$. Coordinates are sorted and exact duplicates are merged. The mean domains are $[-0.4, 0.4]$ for Gaussian, $[0, 1]$ for Bernoulli, and $[10^{-5}, 1 - 10^{-5}]$ for Beta. Two blocks, all six root assignments, and 128 episodes per stratum give 1,536 contributions per environment and component. Any exact overlap with the design set is reported; no generated point is removed because of its observed performance. This tests transfer within the specified reward family, not maximization over the continuous domain or reward-model misspecification.

Structured-stress uncertainty. Mean-band projections follow Appendix C.5, with $C = 11 \cdot (164 + 164 + 164)$ scalar means. The 95% simultaneous Hoeffding and 2,000-replicate approximate studentized-bootstrap bands cover this three-family stress study only. Sampling preserves block and assignment strata and shares multiplicities across methods and environments within each setting. Coverage is not joint with the primary comparison, cross-setting study, low-budget stream or Sobol set comparison below.

Structured-stress results. Each structured stress set contains 164 environments, with no exact overlap with its design set. Tables 12 and 13 report all methods and contrasts. Grid/PBVI have lower point regret than the strongest reference for Gaussian and Bernoulli. For Beta, their regrets are 3.6304/6.3480 versus 3.1919 for KL-UCB^{++}. The Beta PBVI difference is 3.1562 $[2.4761, 3.7928]$ under the approximate simultaneous interval; the other five approximate intervals and all six Hoeffding intervals contain zero. Appendix C.11 evaluates the capacity and target-update interventions in complete Beta solver runs.

Gaussian Sobol evaluation with the expanded reference set. A separate evaluation compares the 17 frozen Gaussian deployments on 128 sorted scrambled-Sobol triples in $[-0.4, 0.4]^3$. The comparison includes Grid/PBVI, all nine benchmark references and all six finite-support 1,024-update RL deployments. Four blocks, six assignments and 128 episodes per stratum give 3,072 contributions per environment and component under the shared aggregation protocol.

This evaluation belongs to the same statistical family as the three Beta sets in Appendix C.11: one multiplicity covers all deployment–environment means across those four sets, including the Beta diagnostic responses. Approximate simultaneous bands use 2,000 within-stratum paired bootstrap resamples and projection through environment maxima and reference minima; separate Hoeffding envelopes retain the same multiplicity and bounded pseudo-regret ranges. These conditional intervals are not pooled with the structured-stress study above and do not estimate training-population variation.

Table 14 retains every Gaussian deployment, both interval constructions and rollout cost. Grid and PBVI have point regrets 10.920 and 10.576; the lowest-point reference is the finite-support 1,024-update actor with seed 923303 and $\epsilon = 0$, at 11.084. These point orderings do not establish a resolved improvement.

Table 12: Structured stress set evaluation: worst mean cumulative pseudo-regret for all eleven methods. †: regenerated policy tables (Appendix D).

Method	G3-T100	B3-T100	C3-T100
MOSS	11.3252	5.4161	3.2455
UCB	11.8762	5.8595	3.3128
Thompson Sampling	13.2480	7.0709	4.0411
Minimax-Bayes RL	14.6749	7.7373	8.0672
KL-UCB^{++}	11.5495	5.7330	3.1919
SoftElim	12.6357	6.0378	3.5135
MOTS	12.8943	6.6148	4.6689
EXP3-IX	24.2946	16.1119	16.4369
UCB-V	11.8091	5.8582	3.2092
Grid	11.0606†	5.2320	3.6304
PBVI	10.6757	5.3676	6.3480

Table 13: Structured stress set differences from the best of nine baselines. Negative is favorable. Intervals have simultaneous scope within this study, separate from the primary design-set comparison and the low-budget stream. Bootstrap intervals are approximate; Hoeffding intervals have finite-sample coverage under the stated sampling assumptions. †: regenerated policy tables (Appendix D).

Setting	Method	Difference	Approximate 95%	Hoeffding 95%
G3-T100	Grid†	-0.2646	[-2.6121, 2.0810]	[-6.6467, 6.0792]
G3-T100	PBVI	-0.6495	[-2.5841, 1.3188]	[-6.6799, 5.5923]
B3-T100	Grid	-0.1842	[-1.2437, 0.8754]	[-5.1974, 5.2652]
B3-T100	PBVI	-0.0485	[-0.8031, 0.8677]	[-5.0693, 5.3911]
C3-T100	Grid	0.4386	[-0.2681, 1.1917]	[-4.8059, 6.5156]
C3-T100	PBVI	3.1562	[2.4761, 3.7928]	[-4.9820, 9.4331]

Deployment	Regret	Approx. 95%	Hoeffding 95%	CPU s/10^6
EXP3-IX	24.164	[23.558, 24.771]	[20.973, 27.356]	0.11
KL-UCB++	11.473	[10.609, 12.336]	[9.659, 13.691]	0.10
MOSS	11.505	[10.672, 12.415]	[9.857, 13.697]	0.08
MOTS	12.509	[11.563, 13.467]	[10.576, 14.991]	0.12
Minimax-Bayes RL	14.560	[13.547, 15.574]	[12.062, 17.668]	13.56
SoftElim	12.552	[11.457, 13.667]	[10.535, 15.001]	0.12
Thompson Sampling	12.769	[11.675, 13.863]	[10.721, 15.433]	0.10
UCB	11.982	[11.103, 12.885]	[10.142, 14.457]	3.96
UCB-V	11.896	[11.119, 12.673]	[10.013, 14.448]	0.08
RL, seed 1, $\epsilon = 0$	11.655	[11.004, 12.329]	[10.020, 13.490]	8.74
RL, seed 1, $\epsilon = 0.1$	12.570	[11.945, 13.195]	[10.735, 15.168]	8.33
RL, seed 2, $\epsilon = 0$	11.107	[10.179, 12.035]	[9.635, 13.358]	8.82
RL, seed 2, $\epsilon = 0.1$	12.287	[11.432, 13.174]	[10.233, 15.553]	9.57
RL, seed 3, $\epsilon = 0$	11.084	[10.365, 12.056]	[9.454, 13.187]	8.12
RL, seed 3, $\epsilon = 0.1$	12.354	[11.681, 13.170]	[10.620, 15.377]	8.81
Grid†	10.920	[10.150, 11.695]	[9.199, 13.611]	5.59
PBVI	10.576	[10.053, 11.172]	[8.959, 13.066]	12.47

Table 14: Gaussian evaluation on 128 Sobol environments. All 17 deployments share evaluation streams. Intervals retain the joint study-wide scope across this Gaussian set and the three Beta sets in Appendix C.11, not the structured-stress family above. All RL actors use 1,024 updates and finite-support training; seeds 1/2/3 are 923301/923302/923303, and ϵ is deployment exploration. CPU is component-weighted batched rollout time, including reward generation and evidence updates. †: regenerated tables (Appendix D).

C.9 Capacity and response targets: compressed-prior solver control

The next three controls separate target updates at a fixed compressed capacity, individual-response sensitivity at fixed q, and their interaction in complete full-prior solver runs. For each reward family at $K = 3, T = 100$, three paired training seeds compare adaptive and fixed-uniform response targets. Every run starts with an empty pool and uniform environment weights, then makes exactly six PBVI response calls with 256 design beliefs, pair-support cap 16 and a global clock. Both conditions use the minimax master for their output policy weights. The adaptive condition uses its preceding dual weights for the next response; the control keeps its response target uniform. Design seeds vary across calls but agree within each pair, as do training reward streams. Duplicate responses count toward the six-call budget.

Training evaluates 64 episodes per representative under each of six arm assignments. The sixth-round mixtures are frozen before evaluation on two independent blocks, six assignments and 64 episodes per assignment and environment: 768 contributions per component and environment. Returns are combined with frozen component weights, averaged over assignments, and then maximized over the design environment set. The evaluation streams do not select checkpoints or seeds.

Table 15: Three-seed matched-call comparison. Entries are adaptive-minus-uniform final finite-set regret. SD is the sample standard deviation of three paired differences, combining training variation and finite evaluation noise.

Family	Seed 1	Seed 2	Seed 3	Mean $\pm$ SD
Gaussian	-2.898	-3.092	-2.311	-2.767 ± 0.406
Bernoulli	-2.497	-2.773	-2.782	-2.684 ± 0.162
Beta	-2.459	-3.306	-3.589	-3.118 ± 0.588

All nine paired differences favor adaptive updates (Table 15) under this compressed-prior configuration. Appendix C.11 separately compares response targets with full pair priors. The 108 response constructions and 1,944 evaluation tasks for this three-seed study used 2.307 CPU hours.

C.10 Fixed-prior response sensitivity to support and belief budgets

We construct three individual responses to the fixed final PBVI environment mixture at Beta $K = 3, T = 100$. Eight of its 58 environment rows have positive mass. Sampling an unordered arm pair induces 19 distinct pair types, or 33 labelled prior atoms after accounting for arm swaps and ties. The reference uses support cap 16 and belief capacity 256; it retains eight types and 15 labelled atoms. Increasing the cap to 64 preserves all 19 types and 33 atoms. A third response retains cap 16 and increases belief capacity to 1,024. These are separate response constructions at one fixed q, with no policy-pool expansion or new mixture fitting.

All three use horizon 100, concentration 2, quadrature order five, global remaining time, and the same design seed. At belief capacity P, the prior-weighted, largest-mass-pair and largest-gap-pair strata use $P, P/2, P/2$ candidate histories and retain $P/2, P/4, P/4$ beliefs. Increasing capacity therefore changes the candidate-history and retained point budgets together. Changing the support cap also changes the reachable-belief design induced by that prior, even with a common seed. The controls measure these configuration changes, rather than pure compression error with an identical fixed belief set.

Each frozen response is evaluated on the design set and the structured stress set defined in Appendix C.8. Two blocks, six assignments, and 128 episodes per block and assignment give 1,536 contributions per environment. Shared action and reward quantiles pair the responses. We report both the q-weighted regret on the design set and the maximum regret on each set, averaging assignments before maximization. The expected q-weighted regret of a feasible policy bounds the optimal response cost from above. The reported Monte Carlo point estimate is neither a certified bound nor a lower bound on the game's value.

Retaining the full induced pair prior reduces q-weighted regret by 1.0976, with a descriptive paired 95% bootstrap interval $[-1.1986, -1.0010]$. Its design-set and structured-stress maximum-regret changes are -4.9060 ($[-5.3372, -4.4796]$) and -7.8005 ($[-8.3872, -7.2103]$). Increasing belief

Table 16: Single-response Beta controls at one fixed final PBVI environment mixture. CPU seconds measure a single-threaded response construction, excluding evaluation. Maxima use the 58-environment design set and the 164-environment stress set. Each row is one complete policy, distinct from the benchmark eight-component PBVI deployment.

Cap / beliefs	q-weighted regret	Design max	Stress max	CPU (s)
16 / 256	3.3313	10.0728	13.2935	17.92
64 / 256	2.2336	5.1668	5.4930	20.52
16 / 1,024	3.2596	10.3625	13.5246	223.20

capacity instead changes the three point estimates by -0.0717, $+0.2897$, and $+0.2311$; all three descriptive intervals contain zero. The full-support construction uses 1.15 times the reference CPU, while the larger belief budget uses 12.46 times as much.

Intervals use 1,000 paired percentile bootstrap resamples of episode columns within block and assignment, preserving pairing across policies and environment rows and recomputing each maximum. These descriptive intervals have neither simultaneous coverage across all controls nor a finite-sample guarantee. Regret changes in individual environments have both signs.

The support-cap contrast identifies a substantial sensitivity of this fixed-prior response that increasing the belief budget does not resolve. The selected benchmark mixture contains responses to different priors. This control instead fixes one target prior and compares individual responses. The complete solver study below tests the effect of the support rule on trained mixtures. Benchmark mixture scores use their stated evaluation streams.

C.11 Complete solver control of pair-prior capacity and target updates

At Beta $K = 3, T = 100$, three paired training seeds compare labelled support caps 16 and 348. The latter retains every induced pair type from the 58 training environments, since $2\binom{3}{2} \cdot 58 = 348$. Both use the same prior constructor, retaining environment masses above 10^{-12} and recording discarded mass. The initial uniform prior has 122 unordered pair types and 192 labelled atoms.

Each run starts with an empty pool and uniform environment weights, makes six response calls, and deploys its sixth restricted-game solution. Both conditions use 256 design beliefs, order-five quadrature, a global clock, and paired design-seed schedules. Training averages all six arm assignments with 64 episodes each, on paired streams separate from evaluation. The restricted-game solution supplies each subsequent response target. Changing support also changes the induced design beliefs.

Full-prior control with a fixed response target. Three additional full-prior runs hold the response target uniform, with the same empty-pool initialization, six-call budget, numerical settings, design seeds and training streams. Their own restricted-game solutions still determine the output policy mixtures. This control was specified after observing the six adaptive training trajectories, including the lower first-call full-prior values, but before its own outcomes.

Evaluation sets and comparators. All nine final mixtures, nine benchmark references, six finite-support 1,024-update Beta RL deployments, and benchmark Grid/PBVI mixtures share evaluation streams on the 58 design environments, the 164-environment structured stress set, and 128 sorted scrambled-Sobol triples in $[10^{-5}, 1 - 10^{-5}]^3$. The Sobol seed and points were fixed before training outcomes and have no exact overlap with either other set. These interventions concern Beta only; the Gaussian Sobol set comparison and its shared interval scope are reported in Appendix C.8.

Four blocks, six arm assignments and 128 episodes per stratum give 3,072 contributions per environment and component. Frozen component weights are applied before assignment averaging and environment maximization. These finite-set comparisons report every training seed individually; their summaries are descriptive.

A response check at the final strategy pair. For each of the six frozen adaptive-target pairs, an additional full-prior response ρ uses 512 design beliefs and a separate seed. It is evaluated on

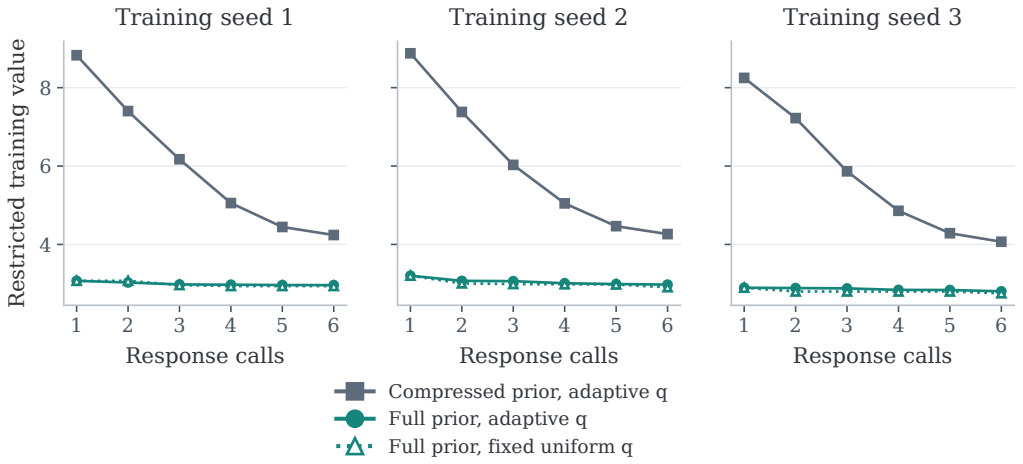

Figure 10: Complete Beta solver trajectories for three paired training seeds. Each panel compares compressed and full pair priors with an adaptive target, and a full prior with a fixed uniform target. The vertical axis is the restricted LP value on the training streams; final policy weights come from call six in every run.

the design set, without entering the pool. Its signed improvement is $R_T(\sigma, q) - R_T(\rho, q)$. Since $L_M(q) \le \min\{R_T(\sigma, q), R_T(\rho, q)\}$,

$$0 \le U_M(\sigma) - \min\{R_T(\sigma, q), R_T(\rho, q)\} \le g_M(\sigma, q). \tag{32}$$

This feasible-deviation quantity lower-bounds the Nash gap; a small value does not certify a small gap.

Uncertainty and cost. One simultaneous family covers all deployment–environment means across the three Beta sets and Gaussian Sobol set (Appendix C.8), including diagnostic responses. We project approximate bands from 2,000 within-stratum paired bootstrap resamples through environment maxima, reference minima and fixed-q averages. Separate Hoeffding envelopes use the same multiplicity and range $[0, T(\max_a \mu_a - \min_a \mu_a)]$. These intervals condition on the frozen policies and sets, rather than measuring population-level training-seed variation. Construction, training rollouts and diagnostics have separate CPU records. Deployment timings cover batched rollouts, reward generation and evidence updates, under shared load on the same server.

Training trajectories and cost. Full priors lower training values throughout all three adaptive pairs (Figure 10), at 184–200 construction CPU seconds versus 58–59 for compressed priors. Including rollouts gives 378–402 versus 156–157 seconds (Table 17). No mass is discarded in these 36 calls; each final policy mixture has four positive-weight components. Uniform-target training takes 2,422–2,594 seconds, retaining 192 labelled atoms at every call, while final adaptive environment mixtures concentrate on four instances. Budgets match response calls and numerical settings, not CPU time.

Effect of the pair-prior representation. Full priors reduce worst regret in all nine seed–set comparisons (Table 18). Reductions are 1.048–1.275 on design set, 1.839–3.093 on structured stress set and 1.818–3.186 on Sobol set; all approximate simultaneous intervals exclude zero. On design set, full-prior regrets are 2.974, 3.033 and 3.063, versus compressed-prior values 4.249, 4.266 and 4.110. The full-prior values are below MOSS's point regret of 3.353, with intervals $[-0.818, 0.060]$, $[-0.769, 0.134]$ and $[-0.715, 0.134]$ for candidate minus reference. On structured stress set and Sobol set, seed three has regrets 4.278 and 4.183 versus references 3.213 and 3.183, with difference intervals $[0.478, 1.652]$ and $[0.429, 1.577]$; the other two seeds' reference comparisons include zero. Resolved contrasts here use approximate bootstrap bands; the Hoeffding envelopes are wider.

Effect of adapting the response target. Updating the full-prior target gives no resolved design-set regret advantage (Figure 11). Uniform targets give lower point regret for every seed on both

Table 17: Complete Beta training cost. Construction and training rollouts are separate; the last column is the additional post-training response diagnostic, excluded from the policy mixture. The last three rows keep the response target uniform; all others update it. Timings are CPU seconds.

Seed	Pair prior	Training value	Construction	Rollouts	Diagnostic
1	Compressed	4.240	58.62	98.10	42.91
1	Full	2.962	199.55	202.05	44.47
2	Compressed	4.264	59.37	97.29	43.42
2	Full	2.975	198.00	203.86	39.64
3	Compressed	4.068	57.61	98.17	42.16
3	Full	2.804	183.58	194.72	44.91
1	Full, fixed q	2.938	1319.15	1274.51	—
2	Full, fixed q	2.908	1333.07	1229.35	—
3	Full, fixed q	2.753	1286.84	1135.13	—

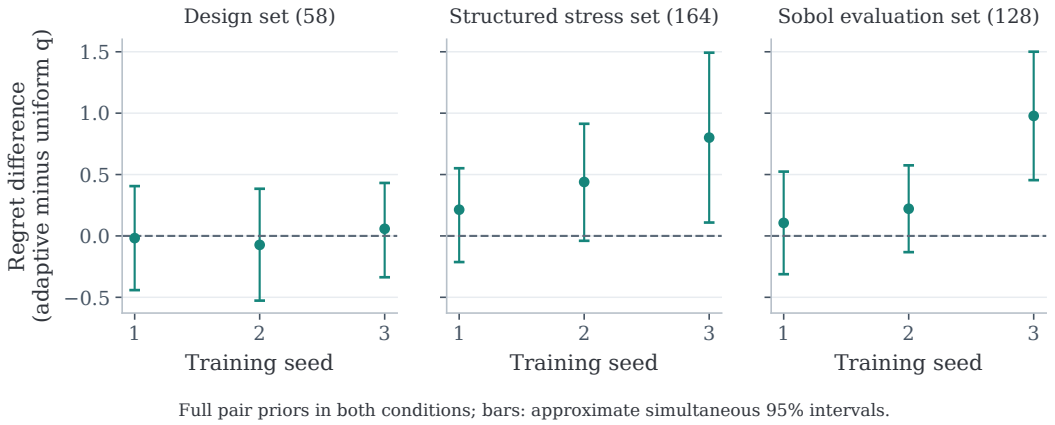

Figure 11: Effect of updating the environment mixture with full pair priors. Each point is adaptive-target minus uniform-target worst regret for one paired training seed; positive values favor the fixed uniform target. Error bars are approximate simultaneous 95% intervals conditional on the trained policies.

additional sets. Adaptive-minus-uniform differences for seed three are 0.801 and 0.978, with approximate intervals $[0.109, 1.493]$ and $[0.454, 1.501]$; the other four intervals include zero. Uniform-target Sobol set regrets are 3.144, 3.158 and 3.206, all unresolved against the best reference (3.183). Together with the training costs above, these results describe the target-update cost–performance tradeoff.

Response quality at the final strategy pair. The independent response improves final-q cost by 2.129–2.352 for compressed priors, with three strictly positive approximate intervals, and by 0.181–0.248 for full priors, with three intervals including zero (Tables 19–20). The feasible-deviation lower estimates in Equation (32) are respectively 2.190–2.406 and 0.323–0.470.

Complete deployment results. Table 18 combines all Beta candidate results. The RL training seeds are 923301, 923302, and 923303, each evaluated at both deployment-exploration probabilities. Compressed and full priors use adaptive targets unless labelled fixed q. On Beta design set, full adaptive priors use 101–121 CPU seconds per million decisions, versus 14–27 for compressed priors and 5.56 for MOSS. Costs weight complete component rollouts by deployment probabilities, including reward generation and evidence updates.

Table 18: Beta solver-capacity comparison on three environment sets. Entries are worst-regret points with approximate study-wide 95% intervals. All nine control-study mixtures and both benchmark policies are shown. The reference row displays the lowest-point reference separately on each set: MOSS, MOSS, and KL-UCB^{++}, respectively; its bands are for those individual policies. All 15 references remain in the comparison and uncertainty family.

Deployment	Design set (58)	Stress set (164)	Sobol set (128)
Point-selected reference	3.353 [3.120, 3.586]	3.213 [3.010, 3.416]	3.183 [2.939, 3.426]
Benchmark Grid	3.540 [3.259, 3.853]	3.526 [3.308, 3.828]	3.781 [3.432, 4.129]
Benchmark PBVI	3.845 [3.528, 4.161]	6.303 [6.056, 6.550]	6.631 [6.401, 6.861]
Compressed, seed 1	4.249 [4.210, 4.314]	6.288 [5.929, 6.646]	6.435 [6.045, 6.826]
Compressed, seed 2	4.266 [4.226, 4.406]	5.227 [5.093, 5.362]	5.197 [5.054, 5.518]
Compressed, seed 3	4.110 [4.048, 4.317]	6.352 [5.989, 6.715]	6.285 [5.883, 6.687]
Full, seed 1	2.974 [2.768, 3.180]	3.195 [3.028, 3.361]	3.250 [3.023, 3.476]
Full, seed 2	3.033 [2.816, 3.254]	3.388 [3.095, 3.681]	3.380 [3.258, 3.501]
Full, seed 3	3.063 [2.871, 3.254]	4.278 [3.894, 4.662]	4.183 [3.850, 4.517]
Full, fixed q, seed 1	2.992 [2.774, 3.209]	2.981 [2.810, 3.241]	3.144 [2.952, 3.335]
Full, fixed q, seed 2	3.106 [2.870, 3.343]	2.948 [2.767, 3.135]	3.158 [2.926, 3.390]
Full, fixed q, seed 3	3.005 [2.823, 3.208]	3.477 [3.169, 3.785]	3.206 [3.016, 3.395]

Seed	Pair prior	U_M	$R_T(\sigma, q)$	$R_T(\rho, q)$	Difference	Gap lower estimate
1	Compressed	4.249	4.106	1.914	2.192	2.335
1	Full	2.974	2.862	2.628	0.235	0.346
2	Compressed	4.266	4.213	1.860	2.352	2.406
2	Full	3.033	2.891	2.710	0.181	0.323
3	Compressed	4.110	4.049	1.920	2.129	2.190
3	Full	3.063	2.841	2.593	0.248	0.470

Table 19: Response checks at the six final Beta strategy pairs, on the design set. ρ is a separate full-prior response with belief capacity 512. Difference is $R_T(\sigma, q) - R_T(\rho, q)$; the last column estimates the gap lower bound $U_M - \min\{R_T(\sigma, q), R_T(\rho, q)\}$ from Equation (32).

Seed	Pair prior	Difference: approx. 95%	Gap lower estimate: approx. 95%
1	Compressed	[1.883, 2.502]	[2.137, 2.559]
1	Full	[-0.196, 0.665]	[0.000, 0.754]
2	Compressed	[2.059, 2.646]	[2.221, 2.691]
2	Full	[-0.214, 0.576]	[0.000, 0.760]
3	Compressed	[1.824, 2.434]	[1.971, 2.553]
3	Full	[-0.151, 0.647]	[0.084, 0.857]

Table 20: Approximate simultaneous intervals for the feasible response checks. They quantify these fixed deviations; their upper endpoints do not upper-bound the Nash gap.

C.12 Controlled resolution and coordinate study

The 19 ablation runs use $K = 3, T = 100$, a fixed pair-support cap of 16, at most 24 response rounds, and the same within-family environment set and fixed evaluation seed. Gaussian and Bernoulli use global clocks; Beta uses pair-local clocks. Only capacity changes within each implementation's three-resolution comparison. The Beta coordinate control changes the coordinate definition at fixed 17^3 resolution. Table 21 reports final regret, completed response rounds, and total wall-clock time. Runs stop when a response repeats or the 24-round budget is exhausted. Times include evaluation and the final repeated-response check, use one numerical-library thread per task, and were recorded with up to three concurrent tasks. They are complete-run costs, not matched per-response speed measurements. The support cap is fixed throughout this study.

At the shared round 16, mean-logit Beta Grid has regret 3.098621 and the square-root-count logit control has 3.416852. Their terminal comparison, 3.076257 versus 3.416852, additionally includes different stopping rounds (21 versus 16), so the final difference combines the coordinate change with additional iterations. Resolution changes both the generated policies and the induced environment-weight trajectories. Table 21 records the resulting regret and cost at each tested resolution.

C.13 Finite environment sets

An environment is specified by its vector of arm means. The benchmark domains are $[-4/\sqrt{T}, 4/\sqrt{T}]^K$ for unit-variance Gaussian rewards, $[0,1]^K$ for Bernoulli rewards, and $[10^{-5}, 1-10^{-5}]^K$ for concentration-two Beta rewards. The benchmark maximizes over stored finite subsets of these domains. Multi-arm rows are nondecreasing mean vectors; two-arm sets include both swapped orientations. Training, configuration selection, and scoring use these development sets. The nontrivial minimax problem includes hidden arm permutations, not only the sorted labelled rows: with the latter as the entire environment domain, always pulling the final arm would give zero regret. For multi-arm numerical comparisons we preserve the stored representative scores; no additional permutation average is inferred for a policy without the corresponding symmetry property.

Table 22 gives directly available support sizes and coordinate extrema for the representative ablation settings. The extrema do not define a Cartesian product. The sets include previously identified difficult environments; the available Gaussian constructors augment instances selected by high-regret screening with geometrically separated points. The exact stored mean vectors determine each score, rather than a new random draw of environments for each policy.

Some longer-horizon sets rescale existing environments. The saved two-arm Bernoulli construction for $T \in \{500, 1000\}$ preserves midpoint $m = (\mu_1+\mu_2)/2$ of each pair in a 400-row $T = 200$ source and uses signed gap $d_T = \sqrt{200/T}(\mu_2 - \mu_1)$, producing $(m - d_T/2, m + d_T/2)$. The Gaussian benchmark domain itself scales with $T^{-1/2}$. Horizon curves therefore include horizon-normalized instance sets, rather than measuring a fixed absolute mean domain.

C.14 Design-set evaluation and configuration selection

Every design-benchmark entry estimates expected cumulative pseudo-regret using 256 episodes per environment. Let $\widehat{R}_j(\pi)$ denote the estimate for stored environment j. For $K > 2$, the reported score is $\max_j \widehat{R}_j(\pi)$ over the stored ordered representatives. For $K = 2$, if j' is obtained by exchanging the two arms, the score is $\max_{\{j,j'\}}[\widehat{R}_j(\pi) + \widehat{R}_{j'}(\pi)]/2$. For mixtures, $\widehat{R}_j(\sigma) = \sum_i \sigma_i \widehat{R}_j(\pi_i)$ is formed before applying either aggregation. These conventions define the numerical benchmark; they do not constitute an exhaustive search over all continuous environments or every arm permutation for $K > 2$. For a permutation-equivariant policy, representative regret equals permutation-averaged regret in expectation. Otherwise, the representative score is not generally an unbiased estimate of that average. Equivariance must include initialization, tie handling, private randomization, and episode-level mixing; it is not assumed for every comparator.

The numerical response runs start from empty policy pools and do not import baseline policies or responses produced by another configuration. The reported checkpoint and representation were selected from the corresponding run history. Gaussian Grid uses a numerical Gaussian operator; two-arm Bernoulli Grid uses a labelled Bernoulli table; multi-arm Bernoulli Grid uses a normal

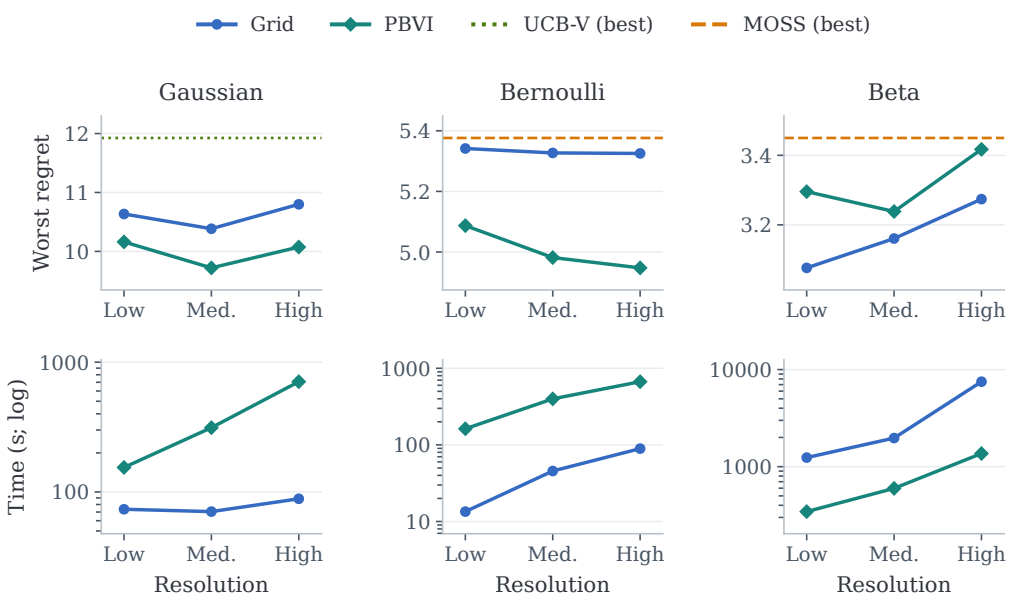

Figure 12: Separate representation-resolution study at $K = 3, T = 100$. Low, medium, and high resolution change grid sizes or posterior-point counts within each implementation. Regret need not improve monotonically. Times cover complete runs, including evaluation and stopping checks; different stopping rounds and concurrent load preclude a matched speedup interpretation.

Table 21: All 19 representation-control runs. The square-root-count logit control uses $L_a/\sqrt{N_a}$; all other Beta Grid entries use mean-logit coordinates.

Setting	Method	Resolution	Regret	Rounds	Seconds
G3-T100	Grid	17×33	10.635042	9	73.5
	Grid	33×65	10.384531	7	70.5
	Grid	65×129	10.799287	6	88.6
	PBVI	128	10.161188	10	154.5
	PBVI	256	9.720061	12	312.9
	PBVI	512	10.073703	11	707.5
B3-T100	Grid	17×33	5.341618	2	13.5
	Grid	33×65	5.326953	6	45.6
	Grid	65×129	5.325287	7	89.4
	PBVI	128	5.087138	14	162.3
	PBVI	256	4.981778	24	398.8
	PBVI	512	4.947716	21	668.3
C3-T100	Grid	17^3	3.076257	21	1240.1
	Grid	33^3	3.160659	17	1970.8
	Grid	$33 \times 65 \times 65$	3.274049	18	7503.0
	PBVI	128	3.295635	16	344.4
	PBVI	256	3.238413	18	597.7
	PBVI	512	3.417250	20	1366.1
	Grid (square-root-count logit)	17^3	3.416852	16	849.4

approximation. Beta Grid uses mean-logit coordinates. Thus the Grid label groups related response representations rather than identical likelihood operators. Most Gaussian and multi-arm Bernoulli Grid entries use 33×65 nodes and pair-support cap 16. Gaussian uses both global and pair-local clocks in selected settings. The two-arm Bernoulli tables use 4,913 or 35,937 nodes. Multi-arm Bernoulli uses cap 32 at B5–T500 and B8–T500; the latter also uses a pair-local clock. Beta Grid

Table 22: Stored $K = 3, T = 100$ environment sets. Extrema range over all arm coordinates.

Family	Environments	Minimum mean	Maximum mean
Gaussian	64	-0.4	0.4
Bernoulli	50	0.01339746	0.98660254
Beta	58	0.01339746	0.98660254

uses 17^3, 33^3, or $33 \times 65 \times 65$ nodes; multi-arm runs use pair-local clocks, usually cap 16, with cap 64 and 18-round budgets at C5–T500 and C5–T1000.

Gaussian PBVI uses 256 posterior points, cap 16, a global clock, and an 18-round maximum throughout. Bernoulli PBVI has the same default, except B3–T500 uses cap 32 and B2–T1000 uses cap 32 with 24 rounds. Beta PBVI uses setting-specific choices: 256 or 512 points, pair-support caps from 16 to 64 where applicable, global or pair-local clocks, and maximum budgets from 4 to 24 rounds. The main benchmark compares these selected configurations; the separate resolution study varies capacity within fixed implementations.

A reproducible reference configuration. The three-family matched-call study provides a fully specified reference configuration: PBVI with 256 posterior design points, labelled pair-support cap 16, a global clock, uniform initial environment weights, and six response calls from an empty pool. Its three training-seed pairs per family use the same construction and evaluation schedules (Appendix C.9). It provides a reproducible starting point at $K = 3, T = 100$. Response accuracy and support sensitivity are assessed by the separate controls above. Timings distinguish planning, rollout evaluation and outer-game updates; their costs vary across families and representations even with matched response-call budgets.

Baseline tuning uses the same finite environment sets as final reporting. The scalar search starts with 25 logarithmically spaced values in $[0.01, 4]$. If the optimum is at a boundary, the lower boundary is divided by 10 or the upper boundary multiplied by 4, adding five logarithmically spaced candidates, for at most four expansions and hard bounds $[10^{-4}, 64]$. Three subsequent rounds add geometric midpoints beside the current best value. Table 23 gives the episodes-per-environment schedule. Selected candidates are assessed on a tuning random stream; final reporting uses a fixed 256-episode stream. Among the search-stage and refined final evaluations, the lower score is retained for each method. These are selected development scores. Frozen-policy comparisons use the separate protocols in Appendices C.5 and C.6.

Table 23: Classical-baseline scalar-search rollout budgets per environment. Successive stages retain promising candidates; the final reporting budget is the same for every horizon.

Horizon	Successive search stages	Tuning confirmation	Final report
$T = 20$	$64 \to 128 \to 256$	1024	256
$T = 100$	$32 \to 64 \to 128$	512	256
$T \geq 200$	$16 \to 32 \to 64$	256	256

MOTS first searches its clipping-bonus coefficient at $\rho = 0.70$, then searches $\rho \in \{0.55, 0.60, \ldots, 0.95\}$ for the best and neighboring coefficients, followed by two midpoint refinements in ρ. Here $1/\rho$ inflates the Gaussian sampling variance; ρ is not the observation-noise standard deviation. This finite search does not establish a global hyperparameter optimum.

C.15 Baseline adaptations and interpretation

The comparisons use the following family-aware implementations of standard stochastic and adversarial bandit algorithms (Lattimore & Szepesvári, 2020). Exploration multipliers are tuned variants, so the guarantees for particular theoretical constants are not assumed to hold for every selected configuration.

Confidence-index methods. UCB uses empirical rewards and a tuned exploration coefficient, with known-noise scaling for Gaussian observations. MOSS uses $\bar{Y}_a + \alpha\sqrt{[\log(T/(KN_a))]_+/N_a}$. UCB-V multiplies its entire variance-adaptive bonus by a tuned scalar: $\alpha\{\sqrt{2\widehat{V}_a \log(t+1)/N_a} + 3\log(t+1)/N_a\}$. Gaussian UCB-V keeps this update, including the unit constant in its linear term, on unbounded, unclipped rewards. It is an empirical extension, not a confidence bound certified by its bounded-reward analysis. Gaussian KL-UCB++ uses the known-variance Gaussian divergence. Bernoulli uses Bernoulli KL. For Beta rewards, KL-UCB++ computes a bounded Bernoulli-KL upper index from the real-valued empirical mean using 48 bisection steps; it does not round rewards into successes and does not use the Beta-family likelihood divergence (Ménard & Garivier, 2017).

Randomized classical policies. Gaussian Thompson Sampling tunes a posterior-standard-deviation multiplier. Bernoulli Thompson Sampling tunes the symmetric $\mathrm{Beta}(a, a)$ prior. Beta-family Thompson Sampling uses the actual Beta observation likelihood with a numerical grid of 513 candidate means and tunes the symmetric mean-prior shape; it does not substitute a Bernoulli observation model. SoftElim uses real-valued sample means, with probability proportional to $\exp[-2N_a(\max_b \bar{Y}_b - \bar{Y}_a)^2/w^2]$ for bounded rewards. Its Gaussian-scaled variant replaces the exponent by $-N_a(\max_b \bar{Y}_b - \bar{Y}_a)^2/(2\sigma^2 w^2)$. MOTS scales both its Gaussian sampling standard deviation and clipping bonus by the known Gaussian noise scale; bounded families use a unit sub-Gaussian proxy. EXP3-IX uses $\eta = s\sqrt{2\log K/(KT)}$, implicit exploration $\gamma = \eta/2$, and the selected-arm loss estimate $(1-Y)/(p_a+\gamma)$. For Gaussian rewards, $1-Y$ can lie outside $[0,1]$; retaining this unclipped update does not inherit the bounded-loss guarantee.

Learned comparator. The Minimax-Bayes RL comparator is a sampled gradient descent–ascent adaptation of Buening et al. (2023), trained from a independent actor initialization for each setting. It uses 64 rounds, width 64, fixed learning-rate choices, and a checkpoint selected during tuning. Its inputs are public pull counts and reward statistics: raw reward sums for Gaussian and Bernoulli observations, and logit-reward sums for the Beta family. It does not observe the true environment or the instantaneous training prior at deployment. The training and checkpoint-selection protocol differs from the paper's output rule. Its smaller hyperparameter search and finite optimization budget are important comparison limits. The environment parameterization is variant-specific: the selected Gaussian and Beta $K = 3, T = 100$ actors use weights on finite environment supports, whereas the selected Bernoulli actor uses independent Beta distributions over the arms' means, with two shape parameters per arm. These are different environment-player representations, not interchangeable descriptions of the same finite-support update.

Why the strongest baseline varies. MOSS's horizon-aware exploration and UCB-V's empirical-variance term are plausible explanations for their frequent strong performance, but have not been causally isolated. In particular, Beta rewards have variance $\mu(1-\mu)/3$, lower than Bernoulli rewards at the same mean, while the selected UCB-V exploration coefficients can be small. These comparisons do not isolate the effects of variance adaptation or the untuned theoretical constants. UCB and KL-UCB++ are sometimes close to the strongest baseline, and Minimax-Bayes RL is strongest at G8–T20, C5–T20, and C8–T20. Both index-based and learned policies are competitive in this benchmark.

C.16 Controlled optimization budgets for the learned comparator

The budget control fixes four branches at $K = 3, T = 100$: Gaussian and Beta with finite-support environment weights, Bernoulli with its benchmark independent product-Beta environment distribution, and Bernoulli with a finite-support environment distribution as a separate alternative. Each branch uses three training seeds. The actor has width 64, Adam learning rate 0.0008, environment-player learning rate 0.03, and training exploration probability 0.1. The sampled policy-gradient update uses realized terminal regret. A single trajectory starts from an independently initialized random actor and saves its last iterates at 64, 256 and 1,024 updates; optimizers are not restarted at those intermediate budgets, and no trained actor or policy pool is imported.

Within a branch and seed, the budget comparison changes only the number of updates. Comparing the two Bernoulli branches also changes the environment representation, so it is not a budget-only effect. Likewise, comparing a fixed-budget last iterate with the benchmark's selected actor changes

the seed and output rule as well as computation. We evaluate every prescribed iterate without checkpoint selection. The protocol remains an adaptation of Buening et al. (2023), not their original discounted-bandit experiment or their random-paired-iterate output rule. Equal update counts do not imply equal total simulation: the finite-support branches evaluate every environment row when updating environment weights, whereas the product-Beta branch uses sampled environment updates. Both policy-gradient trajectories and all simulator transitions are reported.

Separating deployment exploration from training budget. The $\epsilon = 0.1$ deployment mixes its softmax probabilities with a uniform action distribution. If $z_a(h)$ is its logit for arm a at public history h, its action probability is

$$\pi_\epsilon(a \mid h) = (1-\epsilon)\frac{\exp z_a(h)}{\sum_{j=1}^{K} \exp z_j(h)} + \frac{\epsilon}{K}. \tag{33}$$

The deployment choice $\epsilon = 0.1$ imposes the budget-independent lower bound $R_T(\pi_\epsilon, \theta) \geq (\epsilon T/K)\sum_a[\max_j \mu_j(\theta) - \mu_a(\theta)]$. Its maxima on these panels are 5.333 for Gaussian and 4.619 for both bounded families. More training cannot remove this forced-exploration contribution. The deployment-exploration control was specified before inspecting its outcomes: evaluate each actor at both $\epsilon = 0.1$ and $\epsilon = 0$, keeping all network parameters unchanged. The latter preserves softmax randomness; it is not an argmax or a newly trained policy. Both forms of all fixed-budget iterates and all three benchmark actors give 78 deployments in total.

Evaluation uses the same fresh reward and action streams within each family, all six root assignments and the finite design panels. Episode means are averaged over assignments before the environment maximum. Means and sample standard deviations over three training seeds are descriptive; their variation also includes finite evaluation noise and, in some branches, runtime differences. Each within-seed budget contrast uses one unchanged numerical runtime. Separate paired episode-bootstrap intervals describe conditional Monte Carlo variation, not a training-population confidence statement.

Budget and deployment results. The evaluation covers all 12 trajectories through 1,024 updates and all 78 deployments. Table 24 summarizes every prescribed endpoint. Under $\epsilon = 0.1$ deployment, all 24 within-seed comparisons of 256 or 1,024 rounds against 64 rounds reduce point regret. The expanded comparator set therefore uses the 1,024-update actors. The Bernoulli product-Beta branch improves from 7.703 ± 0.056 to 6.791 ± 0.473; its finite-support alternative improves from 7.586 ± 0.078 to 6.830 ± 0.529.

Table 24: Learned-comparator budget control at $K = 3, T = 100$. Entries are mean $\pm$ sample SD of the three separate actors' worst-environment regrets, not the regret of an ensemble or a population confidence interval. Finite denotes weights on fixed instances; product-Beta denotes independent Beta distributions over Bernoulli arm means. Training uses $\epsilon = 0.1$ in every row; only the deployment probability is varied.

Reward family / environment strategy	ϵ_{eval}	64 rounds	256 rounds	1024 rounds
Gaussian: finite	0.1	14.540 ± 0.164	13.659 ± 0.115	12.815 ± 0.117
Gaussian: finite	0	13.675 ± 0.254	12.936 ± 0.899	11.739 ± 0.306
Bernoulli: finite	0.1	7.586 ± 0.078	6.942 ± 0.063	6.830 ± 0.529
Bernoulli: finite	0	8.026 ± 0.365	6.049 ± 0.096	6.194 ± 0.680
Beta: finite	0.1	5.809 ± 0.038	5.429 ± 0.004	5.410 ± 0.005
Beta: finite	0	5.246 ± 0.864	9.308 ± 0.230	9.748 ± 0.279
Bernoulli: product-Beta	0.1	7.703 ± 0.056	7.144 ± 0.268	6.791 ± 0.473
Bernoulli: product-Beta	0	8.177 ± 0.460	7.330 ± 0.943	7.016 ± 1.487

Table 24 retains both deployment forms at every budget. Their performance ordering depends on the reward family and training budget. All 1,024-update actors enter the frozen comparison in Appendix C.6 individually.

At 1,024 updates each trajectory uses 65,536 actor-gradient trajectories. Including environment-player evaluations, the Gaussian, Bernoulli and Beta finite-support branches use 426.394, 334.554

and 387.034 million simulator transitions per seed, respectively; the Bernoulli product-Beta branch uses 19.661 million. Counts and elapsed times are reported at every evaluation budget. These fixed-rate, fixed-width, last-iterate results establish budget and deployment sensitivity, not an optimization ceiling or convergence result for the policy class or the original Minimax-Bayes RL algorithms.

C.17 Selected response configurations

Table 25 gives the representation, rank cap, clock, and selected checkpoint for both methods in every main setting. Node counts describe axes within a time layer; they do not coarsen the finite horizon. The selected checkpoint is distinct from the maximum run budget; 33 Gaussian/Bernoulli Grid maximum budgets are unrecorded. Unrecorded maximum budgets are left unknown, rather than inferred from a selected checkpoint or a regenerated deployment.

Table 25: Selected configurations for all 60 main settings. Grid entries are state-axis node counts per time layer; P denotes retained posterior points. r is the pair-prior rank cap and R the selected policy-space response round, not the run maximum. Clocks: G, global; L, pair-local; E, equivalent at $K = 2$. A dash means no rank compression.

	Grid				PBVI			
Setting	Shape	r	Clock	R	P	r	Clock	R
G2-T20	33×65	–	E	2	256	16	E	12
G2-T100	33×65	–	E	2	256	16	E	9
G2-T200	33×65	–	E	8	256	16	E	5
G2-T500	33×65	–	E	5	256	16	E	3
G2-T1000	33×65	–	E	5	256	16	E	3
G3-T20	33×65	16	G	4	256	16	G	15
G3-T100	33×65	16	G	7	256	16	G	11
G3-T200	33×65	16	G	3	256	16	G	8
G3-T500	33×65	16	G	7	256	16	G	11
G3-T1000	33×65	16	G	3	256	16	G	6
G5-T20	33×65	16	G	4	256	16	G	7
G5-T100	33×65	16	G	3	256	16	G	6
G5-T200	33×65	16	G	5	256	16	G	6
G5-T500	33×65	16	G	4	256	16	G	4
G5-T1000	33×65	16	G	7	256	16	G	5
G8-T20	33×65	16	L	3	256	16	G	7
G8-T100	33×65	16	L	7	256	16	G	9
G8-T200	33×65	16	L	4	256	16	G	6
G8-T500	33×65	16	L	3	256	16	G	7
G8-T1000	33×65	16	L	5	256	16	G	8
B2-T20	$33 \times 33 \times 33$	–	E	7	256	16	E	10
B2-T100	$17 \times 17 \times 17$	–	E	19	256	16	E	8
B2-T200	$33 \times 33 \times 33$	–	E	24	256	16	E	18
B2-T500	$17 \times 17 \times 17$	–	E	8	256	16	E	18
B2-T1000	$33 \times 33 \times 33$	–	E	14	256	32	E	24
B3-T20	33×65	16	G	2	256	16	G	6
B3-T100	33×65	16	G	6	256	16	G	16
B3-T200	33×65	16	G	6	256	16	G	18
B3-T500	33×65	16	G	3	256	32	G	18
B3-T1000	33×65	16	G	5	256	16	G	18
B5-T20	33×65	16	G	2	256	16	G	10
B5-T100	33×65	16	G	3	256	16	G	10
B5-T200	33×65	16	G	3	256	16	G	18
B5-T500	33×65	32	G	8	256	16	G	18
B5-T1000	33×65	16	G	9	256	16	G	11
B8-T20	33×65	16	G	5	256	16	G	14
B8-T100	33×65	16	G	3	256	16	G	16
B8-T200	33×65	16	G	4	256	16	G	9
B8-T500	33×65	32	L	3	256	16	G	10
B8-T1000	33×65	16	G	5	256	16	G	17
C2-T20	$17 \times 17 \times 17$	–	E	12	256	–	E	12

Setting	Shape	r	Clock	R	P	r	Clock	R
C2-T100	$33 \times 33 \times 33$	–	E	12	256	–	E	12
C2-T200	$33 \times 33 \times 33$	–	E	12	256	–	E	24
C2-T500	$33 \times 33 \times 33$	–	E	12	256	–	E	6
C2-T1000	$33 \times 33 \times 33$	–	E	12	256	–	E	4
C3-T20	$17 \times 17 \times 17$	16	L	8	256	16	G	15
C3-T100	$17 \times 17 \times 17$	16	L	12	256	16	G	22
C3-T200	$33 \times 33 \times 33$	16	L	12	256	16	G	18
C3-T500	$33 \times 65 \times 65$	16	L	12	256	16	G	18
C3-T1000	$33 \times 65 \times 65$	16	L	12	256	32	L	18
C5-T20	$17 \times 17 \times 17$	16	L	8	256	32	G	12
C5-T100	$17 \times 17 \times 17$	16	L	12	256	32	G	18
C5-T200	$17 \times 17 \times 17$	16	L	12	256	32	G	18
C5-T500	$33 \times 33 \times 33$	64	L	18	512	64	G	18
C5-T1000	$33 \times 33 \times 33$	64	L	18	256	64	L	18
C8-T20	$17 \times 17 \times 17$	16	L	4	256	16	G	4
C8-T100	$17 \times 17 \times 17$	16	L	12	256	16	G	5
C8-T200	$17 \times 17 \times 17$	16	L	11	256	16	G	8
C8-T500	$33 \times 33 \times 33$	16	L	12	256	16	G	8
C8-T1000	$33 \times 33 \times 33$	16	L	12	256	16	G	6

C.18 Complete design-benchmark comparison

Table 26: Selected development outcomes: ordered representatives for $K > 2$, arm-swap averages for $K = 2$, not a common hidden-permutation game evaluation. Comparison is with the lowest regret b^* among nine scalable baselines. Gain is $100(b^* - \widehat{R})/b^*$. Counts use gain thresholds of 0% ("Lower") and 5%; these describe effect sizes, not statistical significance. Medians are over settings, not ratios of averages. The separate exact-response reference is excluded.

	Lower / 20		Gain $> 5\%$ / 20		Median gain (%)	
Family	Grid	PBVI	Grid	PBVI	Grid	PBVI
Gaussian ($\sigma = 1$)	18	20	7	19	3.21	10.79
Bernoulli	18	19	6	13	2.01	7.46
Beta ($c = 2$)	16	17	15	15	14.43	8.92
Total count / 60	52	56	28	47	—	—

Figure 4 shows all eleven methods in the 60-setting development benchmark.

Table 27 reports every main setting against the minimum of nine scalable baselines. The six decimal places preserve small differences in the recorded estimates, not six-decimal statistical precision. G, B, and C denote Gaussian, Bernoulli, and Beta families. At C3–T20, MOSS and KL-UCB++ tie at the reported baseline value.

Table 27: All 60 selected main results. Lower regret is better. The baseline column is the best of the nine scalable methods, excluding the separate B2–T20 exact-response reference.

Setting	Strongest baseline	Baseline	Grid	PBVI
G2-T20	UCB	2.838820	2.784645	2.603979
G2-T100	MOSS	6.871315	6.696860	5.914207
G2-T200	MOSS	10.010459	9.200918	8.525325
G2-T500	MOSS	16.277798	14.206263	14.583696
G2-T1000	UCB-V	22.369269	19.242557	19.868979
G3-T20	MOSS	4.970331	4.823199	4.588759
G3-T100	UCB-V	11.925000	10.384531	10.686074
G3-T200	UCB-V	17.015992	17.021594	15.326479
G3-T500	UCB-V	25.732461	24.747627	24.716082

Continued on next page

Table 27 (continued)

Setting	Strongest baseline	Baseline	Grid	PBVI
G3-T1000	MOSS	38.912172	34.844797	36.898057
G5-T20	MOSS	9.660512	9.270405	8.034988
G5-T100	MOSS	18.254844	17.622582	16.127806
G5-T200	UCB	25.552950	25.420459	22.028704
G5-T500	MOSS	40.158998	39.593984	37.933232
G5-T1000	UCB-V	58.605602	54.288864	54.075174
G8-T20	Minimax-Bayes RL	14.792987	13.891572	11.912805
G8-T100	MOSS	27.197500	26.723611	23.578369
G8-T200	KL-UCB++	36.164976	38.281338	32.867267
G8-T500	KL-UCB++	59.590830	59.419842	50.284354
G8-T1000	UCB-V	85.188244	83.576576	71.800695
B2-T20	MOSS	1.388364	1.333079	1.346503
B2-T100	MOSS	3.485930	3.000119	3.039889
B2-T200	MOSS	4.837406	4.381476	4.487094
B2-T500	UCB-V	7.552981	7.930764	7.656840
B2-T1000	MOSS	11.284392	11.241827	11.000905
B3-T20	MOSS	2.573416	2.571903	2.252656
B3-T100	MOSS	5.376292	5.326953	5.314536
B3-T200	MOSS	7.906422	7.520849	7.497599
B3-T500	MOSS	12.546953	12.379683	12.151415
B3-T1000	MOSS	18.651986	17.582019	17.693268
B5-T20	MOSS	4.339844	4.216797	3.370495
B5-T100	MOSS	9.073545	8.920688	8.289829
B5-T200	MOSS	12.718285	12.667640	11.496149
B5-T500	MOSS	20.359766	19.192867	20.226182
B5-T1000	MOSS	28.389345	26.716612	27.286837
B8-T20	MOSS	7.000000	6.571716	4.934369
B8-T100	KL-UCB++	12.966847	12.935156	11.045470
B8-T200	MOSS	17.898047	18.277310	16.243872
B8-T500	MOSS	27.062362	27.029869	24.983909
B8-T1000	MOSS	43.218453	42.212755	36.094818
C2-T20	MOSS	0.999980	0.747850	0.754506
C2-T100	MOSS	1.835312	1.644093	1.690546
C2-T200	UCB-V	2.645903	2.339336	2.188092
C2-T500	UCB-V	4.063526	3.249358	3.706889
C2-T1000	MOSS	6.038748	4.922375	5.099651
C3-T20	MOSS / KL-UCB++	1.999960	1.321402	1.295015
C3-T100	MOSS	3.449931	3.257555	3.169875
C3-T200	UCB-V	4.725324	4.961277	5.010225
C3-T500	UCB-V	7.325961	7.461705	7.853973
C3-T1000	UCB-V	9.678450	9.787103	9.306927
C5-T20	Minimax-Bayes RL	3.777268	2.243611	2.173442
C5-T100	UCB-V	5.375252	5.430289	5.076201
C5-T200	MOSS	7.627249	7.529563	7.151265
C5-T500	MOSS	12.327274	10.984744	12.101420
C5-T1000	UCB-V	16.250957	14.988024	16.356743
C8-T20	Minimax-Bayes RL	5.367080	3.691332	3.691332
C8-T100	MOSS	7.195088	4.719174	5.036970
C8-T200	MOSS	9.819141	7.546462	7.539456
C8-T500	MOSS	16.364416	10.514185	13.173233
C8-T1000	MOSS	21.184036	17.525380	19.264180

D REPRODUCIBILITY RESOURCES

The supplementary code and data distinguish response construction, deployment of saved policies, complete-state evaluation, and analysis of saved episode returns. They include all 60 design-environment arrays and 540 selected baseline configurations. The benchmark data contain 109 self-contained deployment entries: 99 method–setting combinations in the nine frozen-policy evaluation

settings, plus ten entries at Bernoulli $K = 2, T = 20$. Learned-policy entries retain positive-weight components or selected actors, not every search checkpoint.

Scope of reproducible computations. The code supports construction and deployment of the Grid representations in Appendix B.3, two-arm Bernoulli PBVI training at $T = 20$, and the three-family, six-response PBVI control in Appendix C.9, including single-seed and three-seed versions. A further support-cap check constructs and independently evaluates Bernoulli $K = 3, T = 100$ responses to uniform q at caps 16 and 32. Complete-state controls include joint and uncompressed pairwise planning, magnitude-versus-sign aggregation, and full policy-space iterations (Appendices C.3–C.4). For the calibrations in Appendices C.1 and C.2, the supplied code recomputes Bayes optima and evaluates saved action tables; it does not reproduce construction of those numerical responses.

The saved $K = 3, T = 100$ deployments include six Grid/PBVI mixtures and 27 baseline configurations across the three reward families. They can be evaluated on new reward trajectories using the observation conventions in Appendix C.5. Supplementary episode returns, training curves, configuration records, and timing measurements cover the frozen-policy comparisons, training-seed controls, response-capacity and coordinate controls, learned-comparator budgets, and complete Beta solver controls reported in Appendix C. The accompanying analysis code recomputes scores and uncertainty intervals with each study's stated aggregation and coverage family. Detailed coverage of trajectories, comparators and computational costs is specified with the code.

The supplementary resources also document an independent lower-budget $K = 3, T = 100$ evaluation stream. Its intervals form a separate family; its samples are not pooled with the primary 6,144-contribution comparison in Appendix C.6.

Regenerated tables and frozen-policy identity. Four entries use policy tables regenerated from saved run records: Gaussian $K = 3, T = 100$ Grid, Gaussian and Bernoulli $K = 5, T = 100$ Grid, and Beta $K = 5, T = 100$ PBVI. They are identified in the evaluation figures, appendix tables, and accompanying policy records. These tables were fixed before the independent evaluation; no response or mixture was selected using its outcomes. The policy records provide the configurations, construction records, and finite supports. Identity with the tables used to obtain the development scores has not been established, so frozen scores describe the supplied policies rather than an exact replay of those computations.

Remaining coverage limits. The supplied deployment subset does not include the full set of 120 selected Grid/PBVI numerical policies or every original benchmark search trajectory. The configuration table distinguishes selected checkpoints from maximum run budgets; 33 Gaussian/Bernoulli Grid maximum budgets remain unrecorded. These missing records are not inferred from checkpoints or regenerated policies.