Fast and Safe Opponent Modeling in Kuhn Poker

Stephen Kwak*
University of Pennsylvania
kwak7@seas.upenn.edu

Julian Weng*
University of Pennsylvania
jw0@wharton.upenn.edu

Jonathan Xu*
University of Pennsylvania
jonx@wharton.upenn.edu

Abstract

Nash equilibrium play guarantees safety but often leaves value on the table when opponents are exploitable. In this paper, we explore a consistent, optimization-based opponent model, Full Max A Posteriori (FMAP) proposed by Ganzfried [2025], and implement scalable, safe, and adaptive variants for Kuhn poker. FMAP treats opponent modeling as a constrained convex program in sequence form with a Dirichlet-form prior and sparse likelihood derived from partial observations. It satisfies consistency guarantees in terms of eventual convergence to the true opponent model that alternatives such as Bayesian Best Response do not. However, it may not be scalable in terms of game size and iteration count due to its usage of Projected Gradient Descent (PGD) relying on quadratic programming. To solve this, in addition to PGD, we implement a Frank-Wolfe (FW) and a stochastic Frank-Wolfe (SFW) variant with optional variance reduction inspired by Hazan and Luo [2016]. Additionally, we implemented two mechanisms for stability and adaptation (to potentially non-static opponents): an NE-BR mixture heuristic and a p-Restricted Nash Response (RNR) option [Johanson et al., 2007]. On 100-opponent Kuhn benchmarks, naive SFW reduces per-update cost by about $4\times$ relative to PGD (e.g., 24.7 ms vs. 102.6 ms) with a modest degradation in estimation error, while variance-reduced SFW achieves a $\sim$2.7$\times$ speedup (38.1 ms vs. 102.6 ms) while keeping profit and L2 error close to full-batch solvers. The safety mechanisms trade exploitation for bounded worst-case loss in non-stationary or adversarial settings.

1 Introduction

Poker has long served as a proving ground for decision-making under partial observability. Nash equilibrium (NE) strategies can guarantee at least the game value against any opponent but can be overly conservative when opponents make systematic mistakes. On the other hand, exploitative strategies earn more against such opponents but may suffer large losses when the opponent adapts. We focus on opponent modeling as a middle path: learn the opponent's behavior online and respond with calibrated aggression and safety.

We build on Full Max A Posteriori (FMAP), introduced by Ganzfried [2025], which frames opponent modeling as a convex optimization problem in sequence form. Instead of a black-box policy network, FMAP produces an interpretable posterior mode over opponent sequences given a Dirichlet-form prior and the set of sequences that are observationally consistent with each terminal history. This algorithm has the primary advantage of consistency: that is, as the number of hands played approaches infinity, the model's estimate of the opponent's strategy converges to their true (static) strategy, which is not necessarily a property of other common opponent modelling methods such as Bayesian Best Response.

However, the analytical solution provided by the FMAP algorithm, as given, has potential pitfalls if put into practice. Most real games are many times larger than Kuhn Poker, so the quadratic-problem projection step in PGD becomes expensive quickly. In addition, opponents could exhibit exploitable strategies while still not being completely static (or even being adversarial), requiring some method of tempering aggression when using the opponent model.

Our contributions are:

*Equal contribution

- **Optimization toolkit:** Implement PGD with exact projection, FW with an LP-based linear minimization oracle, and a stochastic FW (SFW) variant with optional variance reduction tailored to FMAP's sparse observations. In doing so, we expose clear trade-offs between projection cost, per-iteration complexity, and statistical efficiency.
- **Safety and adaptation:** Integrate discounted likelihood, entropy regularization, an NE-BR mixture heuristic, and a p-Restricted Nash Response (RNR) option [Johanson et al., 2007] to balance exploitation, robustness, and non-stationarity in a single framework.
- **Theoretical perspective:** Clarify that FMAP's statistical consistency for static opponents is *solver-agnostic*: any algorithm that approximately minimizes the FMAP objective (PGD, FW, SFW, or hybrids) inherits the same consistency guarantee, preserving FMAP's advantage over sampling-based Bayesian Best Response (BBR).
- **Empirical study:** Evaluate on Kuhn poker [Kuhn, 1950] with sequence-form constraints [Koller et al., 1996], comparing solvers and ablations (NE-BR mixture, p-RNR, switching opponents). SFW reduces average per-update latency from $\sim$102.6 ms (PGD) to as low as 24.7 ms; variance-reduced SFW achieves $\sim$38.1 ms while keeping profit and L2 error close to full-batch methods.

2 Background

Kuhn poker and sequence form. Kuhn poker [Kuhn, 1950] is a three-card, two-player imperfect-information game. Each player antes one chip; chance deals one card to each player; then players alternately bet or pass once. The resulting game tree has information sets corresponding to private card and action history.

Sequence form encodes realization weights for each action prefix. Let x denote player 1's realization vector and y the opponent's realization vector. Flow constraints enforce that the realization weight of a sequence equals the sum of its extensions, with root normalization:

$$Ex = e, \qquad Fy = f, \qquad x \geq 0,\ y \geq 0,$$

where E, F are sparse matrices reflecting the game tree and e, f are appropriate right-hand sides [Koller et al., 1996]. Payoffs are linear: given x and y, the value is $x^\top Ay$, where A encodes terminal utilities weighted by chance.

Observability. In repeated play, we observe terminal histories (cards, actions, and pot outcome) but not the opponent's internal information sets. Each terminal history ℓ is compatible with a subset of opponent sequences. An observability map $O(\ell)$ lists the indices of these sequences. Conditioned on ℓ, the chance-normalized probabilities $q_{t,j}$ act as likelihood weights over $j \in O(\ell_t)$. In FMAP, these sparse likelihoods are combined across hands with a Dirichlet-form prior over y.

Alternative approaches to opponent modeling. Before pursuing the FMAP-based exploitative approach described in this work, we initially explored Deep Counterfactual Regret Minimization (Deep CFR) [Brown et al., 2019]. Deep CFR combines neural network function approximation with counterfactual regret minimization to learn approximate Nash equilibrium strategies through self-play. This approach iteratively trains neural networks to represent both the current strategy and cumulative regrets across information sets, updating these networks via supervised learning on samples generated from self-play trajectories. As the algorithm progresses, the average strategy converges to an approximate Nash equilibrium of the game.

CFR-style self-play systems have achieved expert-level performance in large poker domains, producing strategies that are difficult to exploit in practice. The method requires training separate neural networks for each player position (typically multiple layers with hundreds of hidden units), accumulating regret statistics over millions of traversals of the game tree, and storing large replay buffers to stabilize training. In our preliminary experiments applying Deep CFR to Texas Hold'em poker with a limited action set, we found that even with bounding the game actions to a small set, the scale of the game demanded prohibitive computational resources: training required multiple hours of GPU time per experiment, large memory footprints for network parameters and regret storage, and careful hyperparameter tuning (network architecture, learning rates, reservoir sampling buffer sizes) to achieve stable convergence.

While Deep CFR's guarantees of approaching Nash equilibrium are theoretically appealing, the computational overhead proved to be discouraging for experimentation in our research setting. Moreover, Nash equilibrium strategies

are inherently defensive: they guarantee the game value but do not adapt to exploit suboptimal opponents. Given our goal of studying opponent modeling and exploitation in a interpretable framework, we attempted to pivot towards an exploitative modeling approach. Rather than learning a fixed Nash strategy through expensive self-play, we focus on learning opponent models from observed play and computing best responses (or safe mixtures) against those models. This shift trades the computational cost of Nash equilibrium computation for the statistical challenge of opponent inference, and it opens the door to earning more than the Nash value against weaker opponents. The FMAP framework we adopt provides an optimization-based method for this opponent modeling task with formal consistency guarantees, interpretable sequence-form posteriors, and update costs that scale more favorably than full equilibrium recomputation after each hand.

3 Related Work

Equilibrium computation. Regret minimization methods such as Counterfactual Regret Minimization (CFR) and its variants compute low-exploitability strategies by converging to approximate Nash equilibria in two-player zero-sum imperfect-information games [Zinkevich et al., 2007]. These methods provide strong worst-case guarantees but are not directly designed for opponent exploitation.

Safe exploitation. Safe exploitation has been studied through Restricted Nash Response (RNR) [Johanson et al., 2007] and Safe Opponent Exploitation [Ganzfried and Sandholm, 2015], which trade off worst-case guarantees against gains relative to an assumed opponent model.

FMAP and BBR. Ganzfried [2025] introduced FMAP in sequence form and proved that its MAP estimator is statistically consistent against static opponents drawn from a Dirichlet prior in imperfect-information games. The same work shows that sampling-based Bayesian Best Response (BBR), which restricts the opponent model to the convex hull of finitely many pre-sampled strategies, can be inconsistent: it need not converge to the true opponent even with infinite data.

Frank-Wolfe and stochastic variants. Frank-Wolfe (conditional gradient) methods provide projection-free optimization for smooth convex objectives over compact convex sets [Jaggi, 2013]. Stochastic conditional gradient methods and projection-free variance-reduced schemes handle large datasets and expensive gradients [Hazan and Luo, 2016, Mokhtari et al., 2020], and are relevant in our setting where FMAP inference requires repeated optimization over the sequence-form strategy polytope.

Poker AI. Poker AI systems such as Cepheus and Libratus demonstrate expert-to-superhuman performance via large-scale equilibrium computation, abstraction, and endgame refinements [Brown and Sandholm, 2018, Bowling et al., 2015]. Our focus is narrower: we study small games in which we can inspect the full sequence-form FMAP posterior and isolate the impact of different optimizers and safety mechanisms.

4 Methodology

4.1 FMAP objective

Let $y \in \mathbb{R}^n$ denote the opponent's sequence-form realization plan ($n = 13$ in Kuhn). We use a Dirichlet-form prior over realization weights on the sequence-form polytope, with log-density

$$\sum_{i=1}^{n} (\alpha_i - 1) \log y_i$$

(up to an additive constant) on the domain $\{y \geq 0 : Fy = f\}$. Following Ganzfried [2025], we assume $\alpha_i \geq 1$ (in our experiments $\alpha = 2$), which ensures the log-prior term is concave in y and the overall FMAP objective is concave over the sequence-form polytope.

Given observations $\{(\mathcal{O}_t, q_t)\}_{t=1}^{T}$ from repeated play, the likelihood of y is

$$L_T(y) \;\propto\; \prod_{t=1}^{T} \Big(\sum_{j \in \mathcal{O}_t} q_{t,j} y_j \Big),$$

because at each hand we only know that the opponent chose a sequence in $\mathcal{O}_t$, with chance-normalized weights $q_{t,j}$. The full log-posterior is therefore

$$\log p(y \mid \mathcal{D}_T) = \sum_{i=1}^{n} (\alpha_i - 1) \log y_i + \sum_{t=1}^{T} \log \Big(\sum_{j \in \mathcal{O}_t} q_{t,j} y_j \Big) + \text{const.}$$

FMAP [Ganzfried, 2025] defines the opponent model as the posterior mode in sequence form. The given FMAP objective is

$$\max_{y \geq 0,\; Fy = f} \quad \sum_{i=1}^{n} (\alpha_i - 1) \log y_i + \sum_{t=1}^{T} \log \Big(\sum_{j \in \mathcal{O}_t} q_{t,j} y_j \Big). \tag{1}$$

In implementation we minimize the *negative* of (1), so the gradient used for optimization is

$$\nabla_i = \frac{1 - \alpha_i}{y_i} - \sum_{t : i \in \mathcal{O}_t} \frac{q_{t,i}}{\sum_{j \in \mathcal{O}_t} q_{t,j} y_j}. \tag{2}$$

We also discuss discounted likelihood and entropy regularization as optional stability and adaptation mechanisms in Section 4.3, which modify the FMAP objective.

4.2 Solvers

Projected Gradient Descent (PGD). PGD performs a gradient step followed by projection onto $\{y \mid Fy = f, y \geq \epsilon\}$ using a quadratic program (QP) solved by Gurobi or SciPy. Backtracking line search enforces Armijo decrease. This provides stable convergence with standard guarantees from convex optimization, but projections are expensive in each iteration.

Frank-Wolfe (FW). FW replaces projection with a Linear Minimization Oracle (LMO) that solves

$$s_k \in \arg \min_{s:\, Fs = f,\; s \geq \epsilon} \langle \nabla f(y_k), s \rangle.$$

Updates take $y_{k+1} = (1 - \gamma_k) y_k + \gamma_k s_k$ with step sizes chosen according to standard Frank-Wolfe schedules [Jaggi, 2013]. Per-iteration cost can be lower than PGD because the LMO is a linear program, though FW may require more iterations to reach comparable accuracy.

Stochastic Frank-Wolfe (SFW) with variance reduction. To reduce gradient cost, we use mini-batch gradients over observations with an optional SVRG-style control variate. Our implementation is inspired by the projection-free variance-reduction framework of Hazan and Luo [2016]. In particular, Hazan and Luo [2016] show that variance reduction can improve stochastic-gradient complexity for smooth convex objectives (e.g., improving from $O(1/\varepsilon^3)$ to $O(1/\varepsilon^2)$ stochastic gradient evaluations in their SVRF setting), and they summarize further improvements under additional assumptions such as smoothness plus strong convexity. We do not claim that our streaming, game-specific implementation inherits every bound verbatim. Rather, we use these methods as guidance for practical speed-stability trade-offs.

4.3 Safety and adaptation

NE-BR mixture (heuristic). We use a simple convex combination of a Nash equilibrium anchor strategy x_{NE} and a best response to the current opponent model $x_{\text{BR}}(y)$:

$$x_{\text{mix}} = (1 - p)\, x_{\text{NE}} + p\, x_{\text{BR}}(y).$$

This provides a tunable exploitation-robustness trade-off via p. This mixture, while basic, provides a relatively predictable benchmark to compare against other safety mechanisms.

p-Restricted Nash Response (p-RNR). To obtain a principled safe-exploitation trade-off, we also implement Restricted Nash Response [Johanson et al., 2007] in sequence form. Given a fixed opponent model y_{fix}, we assume the opponent plays y_{fix} with probability p and otherwise plays adversarially (a best response). The corresponding p-RNR policy for Player 1 solves

$$\max_{x} \; p\, x^\top A y_{\text{fix}} \; + \; (1-p) \min_{y} \; x^\top A y,$$

which interpolates between a Nash security strategy ($p = 0$) and a best response to y_{fix} ($p = 1$). We compute this via the equivalent LP

$$\max_{x,q} \; p\,(A y_{\text{fix}})^\top x + (1-p)\, f^\top q \quad \text{s.t. } Ex = e,\; x \geq 0,\; -A^\top x + F^\top q \leq 0,$$

which is a standard sequence-form formulation of p-RNR.

Discounted likelihood. To handle non-stationary opponents, we can weight observation t by γ^{T-t} with $\gamma \in (0, 1]$, implemented via discount weights in the gradient and objective. This keeps FMAP responsive to recent play while retaining feasibility ($Fy = f$) at every update. Discounting intentionally sacrifices strict consistency with a fixed underlying strategy in exchange for adaptation.

Entropy regularization. Adding $\lambda \sum_i y_i \log y_i$ makes the optimization problem strictly convex for each fixed T (on the truncated domain) and empirically stabilizes stochastic gradients, reducing oscillations for SFW at larger batch sizes. This trades a small asymptotic bias toward higher-entropy posteriors for improved numerical behavior. This also comes at the sacrifice of formal consistency, but might suffice in practice.

4.4 Solver progression and scalability

PGD projects via a QP at every step. Projection cost grows poorly with game size, so we probed projection-free variants for scalability:

- **FW:** replaces projection with an LP LMO. In Kuhn, FW attains similar profit and error to PGD with comparable update-time scaling; in larger games, avoiding projections can be especially attractive when projection dominates runtime.
- **SFW:** uses mini-batch gradients to shrink per-step cost. This cut update time substantially but introduced higher variance, mildly hurting accuracy.
- **SFW + variance reduction:** adding SVRG-style control variates restored accuracy close to FW while keeping time low, delivering the best time/accuracy balance among stochastic variants.

4.5 Convergence and consistency

We now separate (i) optimization guarantees for fixed data and (ii) solver-agnostic statistical consistency as the number of hands grows. We prove that the former is in expectation correct for all solvers, with caveats for stochastic-based methods (that could be negligible in practice), while the latter applies to all solvers so maintains the primary advantage of the FMAP algorithm.

Let $\mathcal{D} = \{y \in \mathbb{R}^n : Fy = f,\; y_i > 0\}$ denote the (relative) interior of the sequence-form polytope. In implementation we enforce a numerical floor $y_i \geq \varepsilon$; one can interpret this as optimizing over $\mathcal{D}_{\varepsilon_T} = \{y : Fy = f,\; y_i \geq \varepsilon_T\}$ where $\varepsilon_T \downarrow 0$ as T grows.

For theoretical clarity we define the *minimization* form of the negative log-posterior (per sample):

$$F_T(y) \;=\; -\frac{1}{T}\sum_{t=1}^{T} \log\Big(\sum_{j \in \mathcal{O}_t} q_{t,j} y_j\Big) \;-\; \frac{1}{T}\sum_{i=1}^{n} (\alpha_i - 1)\log y_i.$$

Up to an additive constant and positive scalar multiple, minimizing F_T is equivalent to maximizing (1).

4.5.1 Optimization guarantees for fixed data

Definition 1 (Algorithmic convergence)**.** For a fixed dataset of size T, an optimization algorithm (PGD, FW, SFW, etc.) produces iterates y_k in $\mathcal{D}_\varepsilon$ (for some fixed numerical floor $\varepsilon > 0$). We say it *converges* if

$$F_T(y_k) \;\to\; F_T(\hat{y}_T)$$

as $k \to \infty$, where $\hat{y}_T = \arg\min_{y \in \mathcal{D}_\varepsilon} F_T(y)$ is the empirical minimizer. The mode of convergence (deterministic, in expectation, or almost surely) depends on the algorithm.

Proposition 1 (Strict convexity under $\alpha_i > 1$)**.** If $\alpha_i > 1$ for all i, then for each fixed T the objective $F_T(y)$ is strictly convex on $\mathcal{D}_\varepsilon$ and therefore has a unique minimizer $\hat{y}_T$.

Proof. For each t, the map $y \mapsto \log\left(\sum_{j \in \mathcal{O}_t} q_{t,j} y_j\right)$ is concave on $\{y \geq 0\}$, hence its negation is convex on $\mathcal{D}_\varepsilon$. The prior term contributes $-\sum_i (\alpha_i - 1)\log y_i$; when $\alpha_i > 1$ this is a strictly convex function of y on $\{y_i > 0\}$. A sum of a strictly convex function and a convex function is strictly convex, and restricting a strictly convex function to a convex set preserves strict convexity. Thus F_T is strictly convex on $\mathcal{D}_\varepsilon$ and has a unique minimizer. □

Proposition 2 (Algorithmic convergence of PGD and deterministic FW; expectation guarantees for stochastic variants)**.** Let $\hat{y}_T = \arg\min_{y \in \mathcal{D}_\varepsilon} F_T(y)$ be the unique empirical minimizer for a fixed sample size T (e.g., when $\alpha_i > 1$ for all i).

1. **PGD:** Under standard assumptions (Lipschitz gradients and suitable step sizes), projected gradient descent converges to $\hat{y}_T$ on compact convex sets.
2. **Deterministic FW:** Under smoothness, the Frank-Wolfe algorithm with appropriate step sizes satisfies $F_T(y_k) - F_T(\hat{y}_T) = O(1/k)$ on compact convex domains [Jaggi, 2013].
3. **Variance-reduced projection-free stochastic optimization:** Hazan and Luo [2016] provide projection-free stochastic methods with variance reduction and prove improved complexity in terms of the number of stochastic gradient evaluations needed to reach a target expected suboptimality for smooth convex objectives, and they summarize additional improvements under stronger assumptions (e.g., smoothness plus strong convexity). Our implementation is SVRG-style and empirically reduces gradient noise relative to naive mini-batching, but does not provide verbatim the worst-case guarantees of PGD / deterministic FW.

Proof. Part (1) follows from standard projected gradient descent theory on compact convex sets. Part (2) is a direct application of Jaggi [2013, Theorem 1] to F_T, which is smooth on the compact domain $\mathcal{D}_\varepsilon$ due to the floor ε. Part (3) follows from the projection-free variance-reduction framework of Hazan and Luo [2016] under their stated assumptions for finite-sum smooth convex objectives. □

Remark 1 (Almost sure convergence for SFW variants). Hazan-Luo's variance-reduced methods provide convergence guarantees in *expected* suboptimality. Other stochastic conditional gradient methods, such as those of Mokhtari et al. [2020], establish *almost sure* convergence of the objective value under suitable step-size conditions. Our experiments use an SVRG-style SFW implementation; while it is natural to expect almost sure convergence under appropriate hyperparameter schedules, this is not certain. This is in contrast to the guaranteed long-run statistical consistency described in the below section, which is more important for the general correctness of SFW.

4.5.2 Solver-agnostic statistical consistency

We now formalize the sense in which statistical consistency is independent of the specific convex solver.

Assumption 1 (Static opponent model). We assume the setting analyzed by Ganzfried [2025]:

1. The opponent's strategy $y^\star$ is drawn from the Dirichlet-form prior and held fixed across hands.
2. The game has perfect recall and admits a sequence-form representation; observations are generated by repeated play under $y^\star$, with exact observability mappings $\mathcal{O}_t$.
3. There is no discounting ($\gamma = 1$) and no entropy regularization in the FMAP objective.

Under Assumption 1, F_T is the negative log-posterior (up to constants) divided by T.

Theorem 3 (Solver-agnostic consistency of FMAP)**.** Suppose Assumption 1 holds. Let F_T be the FMAP objective defined on $\mathcal{D}_\varepsilon = \{y : Fy = f,\ y_i \geq \varepsilon\}$, and let $\hat{y}_T \in \arg\min_{y \in \mathcal{D}} F_T(y)$ denote the exact FMAP estimator. Let y_T be the (possibly random) output of any optimization algorithm (PGD, FW, SFW, or a hybrid) run on F_T such that for some sequence $\delta_T \downarrow 0$,

$$F_T(y_T) \ \leq \ \inf_{y \in \mathcal{D}} F_T(y) + \delta_T$$

in probability as $T \to \infty$ (e.g., by running enough inner iterations at each T). Then:

1. (**FMAP consistency**) The exact FMAP estimator satisfies $\hat{y}_T \to y^\star$ almost surely as $T \to \infty$.
2. (**Solver-agnostic consistency**) The approximate estimators y_T produced by any such algorithm also satisfy $y_T \to y^\star$ in probability as $T \to \infty$.

Proof sketch. Part (1) is the main consistency result of Ganzfried [2025]: under Assumption 1, the population objective has a unique minimizer at $y^\star$, and standard uniform convergence plus M-estimation arguments yield $\hat{y}_T \to y^\star$ almost surely.

For Part (2), standard approximate M-estimator results (e.g., Newey and McFadden, 1994) state that if (i) F_T converges uniformly to a population objective F on a compact domain, (ii) F has a unique minimizer $y^\star$, and (iii) $F_T(y_T)$ is within $o_p(1)$ of $\inf_y F_T(y)$, then $y_T \to y^\star$ in probability. In our setting, (i) and (ii) follow from the analysis in Ganzfried [2025], and (iii) is guaranteed by the assumed optimization accuracy $F_T(y_T) \leq \inf_y F_T(y) + \delta_T$ with $\delta_T \downarrow 0$ in probability. Thus any solver that produces approximate minimizers of the FMAP objective is statistically consistent for $y^\star$. □

Remark 2 (Implications for PGD, FW, and SFW). Proposition 2 shows that PGD and FW converge (for fixed T) to the same empirical minimizer under standard conditions, while SFW-style methods provide expectation-type convergence behavior. Theorem 3 then implies that, under the static-opponent assumptions, any such solver yields statistically consistent estimates of $y^\star$ as long as it approximately minimizes the FMAP objective to vanishing error as T grows.

4.5.3 Comparison with BBR and sampled methods

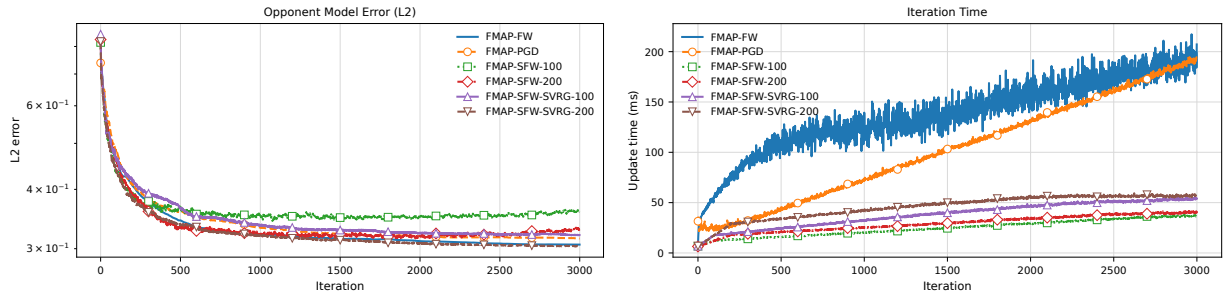
Ganzfried [2025] also analyzes sampling-based Bayesian Best Response (BBR), in which the opponent model is restricted to the convex hull of finitely many pre-sampled strategies from the prior. They show BBR is generally *inconsistent*. First, because BBR restricts the opponent model to the convex hull of finitely many pre-sampled strategies, if the true opponent strategy lies outside that convex hull then the model cannot converge to the truth no matter how much data is observed. Second, even in settings where the true strategy lies in the convex hull, BBR can still be inconsistent: the posterior mass can concentrate and the method can collapse onto a single sampled strategy rather than the true mixture (see their Proposition 2 argument in Rock-Paper-Scissors).

Our framework preserves FMAP's consistency advantage over BBR even when moving to stochastic, projection-free methods: FW and SFW merely change how we approximately minimize the FMAP objective. As long as they solve it well enough in the sense of Theorem 3, the resulting opponent model remains consistent in the static-opponent regime, whereas BBR and related sampled methods need not.

5 Experimental Setup

Environment. We evaluate in Kuhn poker with exact sequence-form matrices E, F, A. Opponents are sampled from Dirichlet priors ($\alpha = 2$ unless specified). Observations are generated by simulating cards and actions, then recording the set of opponent sequences consistent with the observed terminal history. All experiments are run on a single CPU machine (CPU-only, with parallel workers); unless otherwise noted we use a fixed base seed and average over many sampled opponents.

Metrics. We report (i) L2 error $\|y_{\text{model}} - y_{\text{true}}\|_2$ in sequence form, (ii) per-iteration expected profit $x^\top A y$, (iii) cumulative profit, and (iv) wall-clock update time per FMAP update. Baselines include a best response (upper bound on exploitation) and a Nash strategy (safety anchor).



(a) Opponent-model L2 error.

(b) Per-iteration update time.

Figure 1: Convergence and iteration-time curves for the main solver benchmark.

Presets and hyperparameters. Our main solver benchmark runs 3,000 hands against 100 opponents, comparing PGD, FW, and SFW with batch sizes 100 and 200, with and without variance reduction. Learning rates and batch sizes were chosen via coarse grid search to balance stability and speed; we enforce a small feasibility floor $y_i \geq \epsilon$ for numerical stability (solver defaults differ: $\epsilon = 10^{-6}$ for FW/SFW, 10^{-12} for PGD). Our safety benchmark runs 100 iterations against 30 opponents in both adversarial and typical (static opponent) scenarios, comparing pure exploitation, NE-BR mixing, p-RNR, and Nash security using SFW-SVRG.

6 Results

6.1 Solver accuracy vs. efficiency

Table 1 reports the final solver comparison (100 opponents, 3,000 hands; averages across runs). All solvers reach similar profit levels. Naive SFW is fastest but suffers higher estimation error, and SVRG-style variance reduction closes much of this gap while retaining low per-update latency.

Table 1: Solver comparison (final preset, Kuhn poker).

Model	$\mathbb{E}$[Avg profit / hand]	L2 error	Avg update time (ms)
FMAP-PGD	0.5523	0.315059	102.63
FMAP-FW	0.5549	0.305993	137.33
FMAP-SFW-100	0.5522	0.359125	24.66
FMAP-SFW-200	0.5547	0.329539	29.08
FMAP-SFW-SVRG-100	0.5504	0.320532	38.07
FMAP-SFW-SVRG-200	0.5526	0.303762	45.81

Across solvers, final profit is similar (Table 1), while naive SFW exhibits higher L2 error that is reduced by variance reduction (Figure 1). Final profit is similar (likely the same within a margin of error, which is why it appears uncorrelated with L2 error) as these solvers perform relatively similar to each other in practice. However, this could be masked by the extremely limited action set in Kuhn poker, so even substantial differences in opponent model could result in similar best-action. Hence, we decided to focus on minimizing L2 error which more directly represents model quality.

Quantitatively, SVRG reduces SFW's L2 error by 8-11% (from 0.359 to 0.321 for batch-100, from 0.330 to 0.304 for batch-200) while adding modest computational cost (54-58% time overhead respectively). A Pareto-efficient accuracy-efficiency trade-off is achieved by SFW-SVRG-200, which matches FW's accuracy (L2 $\approx$ 0.304) at one-third the update time (46 vs 137 ms). The per-iteration update time scales linearly as more observations accumulate for PGD, while FW-based methods exhibit sublinear scaling such that PGD timings approach that of vanilla-FW, which itself is still much slower than any form of stochastic FW.

6.2 Safety via mixing and p-RNR

To benchmark safety, we report both a worst-case adversarial scenario (the opponent best-responds to the agent's current strategy) and a typical static-opponent scenario (opponents sampled from a Dirichlet prior). Table 2 summarizes final profit per hand, final L2 error, and average update time in each scenario, along with a theoretical worst-case value computed by best-responding to the agent's initial policy (higher, i.e. less negative, is safer).

Table 2: Worst-case vs. typical trade-offs for safety mechanisms (30 opponents, 100 iterations; SFW-SVRG solver).

	Worst-case (adversarial BR)			Typical (Dirichlet $\alpha = 2$)			
Method	Final profit	Final L2	Time (ms)	Final profit	Final L2	Time (ms)	Worst-case value
Pure exploit	-0.2444	1.487180	24.41	0.5572	0.486461	19.22	-0.3333
NE-BR mixture ($p = 0.5$)	-0.1222	1.300923	19.01	0.3603	0.371078	18.28	-0.1944
p-RNR ($p = 0.5$)	-0.1074	1.484726	19.38	0.4429	0.467663	18.36	-0.1111
Nash security ($p = 0$)	-0.0556	1.259358	18.42	0.0587	0.596634	19.70	-0.0556

The results reflect a clear exploitation-safety trade-off. In the typical static-opponent scenario, pure exploitation achieves the highest profit, with p-RNR retaining substantial value while improving worst-case guarantees. In the worst-case adversarial scenario, all methods incur negative profit, but the ordering matches the theoretical guarantees: Nash has the best (least negative) worst-case value, pure exploitation the worst, and p-RNR provides a tunable middle ground that improves on the heuristic NE-BR mixture. The L2 error is larger in the adversarial scenario because the opponent's best response changes over time, so the model is tracking a moving target rather than a fixed strategy.

6.3 Adaptation and stability

Discounted likelihood ($\gamma = 0.95$) improves recovery after opponent switches by emphasizing fresh observations, at the cost of a small steady-state bias. Entropy regularization can reduce variance in stochastic updates and empirically lower oscillations in SFW at large batch sizes. Theoretically, these tools could further yield stable learning curves in the presence of non-stationary or adversarial opponents, but require further empirical study.

7 Discussion

Interpretability and transparency. FMAP outputs sequence probabilities that satisfy flow constraints, making it easy to inspect how the model views each information set (e.g., how often the opponent bluffs with a particular card). This contrasts with opaque policy networks and eases debugging and model criticism.

When to prefer each solver. PGD offers the cleanest and most classical convergence guarantees but is projection-heavy, making it less efficient with bigger games. FW replaces projections with an LP LMO, which can be advantageous in larger games where projection dominates cost; in our Kuhn experiments, FW and PGD exhibit similar update-time scaling once the per-update cost is dominated by processing the accumulated observation history. SFW is preferable once observations are plentiful or when full-gradient passes are expensive, delivering substantial latency reductions at the cost of higher variance, where adding variance reduction recovers much of the accuracy gap while keeping updates cheap.

Consistency vs. convergence. A key conceptual point is that FMAP's statistical consistency is a property of the *objective*, not the solver. Theorem 3 formalizes this: as long as PGD, FW, or SFW approximately minimize the FMAP objective for each T to vanishing error as T grows, they all converge to the same true opponent $y^\star$ in the static-opponent regime. This preserves FMAP's theoretical advantage over BBR even when we move to stochastic, projection-free methods.

Limitations. Kuhn poker is intentionally small; scaling FMAP to larger games might require abstraction, hierarchical solvers, or approximate sequence-form representations, which require care to maintain FMAP's consistency in practice and not defeat the point of its existence. We assume correct observability mappings and unbiased sampling;

extending to noisy or partial logging environments is an interesting avenue for future work. Finally, computing a Nash anchor in larger games is itself challenging and may necessitate approximate equilibrium computation.

In addition, due to time and scope constraints, we were not able to empirically validate our implementation of discounted likelihood or entropy reduction.

As mentioned above, a large motivation of our work is scaling FMAP to larger, imperfect information games such as Texas Hold'em. While we saw that through many iterations, our FW-based solvers exhibited favorable performance scaling relative to PGD-based methods, testing larger games and more iterations explicitly would be worthwhile to confirm our thesis.

8 Conclusion

We presented an optimization-driven framework for opponent modeling in imperfect-information games. By pairing FMAP with projection-free solvers, stochastic variance reduction, NE-BR mixing and p-RNR safety mechanisms, and discounting, we achieve fast updates, competitive exploitation, and improved robustness in Kuhn poker. Importantly, the use of FW and SFW does not alter FMAP's core consistency guarantee in the static-opponent setting: any solver that approximately minimizes the FMAP objective shares the same asymptotic target as the original PGD-based algorithm, while sampling-based BBR fundamentally does not. Future work includes integrating CFR-style regret updates, extending to larger poker variants, and deploying the model within real-time play.

References

Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit hold'em poker is solved. *Science*, 347(6218):145–149, 2015.

Noam Brown and Tuomas Sandholm. Superhuman AI for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424, 2018.

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*, pages 793–802. PMLR, 2019.

Sam Ganzfried. Consistent opponent modeling of static opponents in imperfect-information games. *arXiv preprint arXiv:2508.17671*, 2025.

Sam Ganzfried and Tuomas Sandholm. Safe opponent exploitation. *ACM Transactions on Economics and Computation*, 3(2):8:1–8:28, 2015. Special issue on selected papers from EC'12.

Elad Hazan and Haipeng Luo. Variance-reduced and projection-free stochastic optimization. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1263–1271. PMLR, 2016.

Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435, 2013.

Michael Johanson, Martin Zinkevich, and Michael H. Bowling. Computing robust counter-strategies. In *Advances in Neural Information Processing Systems*, volume 20, pages 721–728, 2007.

Daphne Koller, Nimrod Megiddo, and Bernhard von Stengel. Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14(2):247–259, 1996.

Harold W. Kuhn. A simplified two-person poker. In *Contributions to the Theory of Games*, volume 1, pages 97–103. Princeton University Press, 1950.

Aryan Mokhtari, Hamed Hassani, and Amin Karbasi. Stochastic conditional gradient methods: From convex minimization to submodular maximization. *Journal of Machine Learning Research*, 21(105):1–49, 2020.

Whitney K. Newey and Daniel McFadden. Large sample estimation and hypothesis testing. In *Handbook of Econometrics*, volume 4, pages 2111–2245. Elsevier, 1994.

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.