Maximizing Exploitation? Entropy Smoothing in Deep CFR

Max Oussoren

Computer Science and Mathematics, Stanford University

Problem / Experimental Setting

Question. Does an entropy-smoothed policy improve payoff against imperfect opponents relative to standard Deep CFR?

- **Environment:** Kuhn Poker in OpenSpiel; 3-card deck $\{J, Q, K\}$, one betting round, 12 information states.
- **Target metric:** cross-play expected value (EV) against bounded-rational opponents, not exploitability minimization alone.
- **Comparison:** baseline Deep CFR vs. SoftRM with $\lambda = 0.1$ and $\lambda = 0.3$ under identical training budget.

Setting	Value
Training budget	500 iterations; 200 traversals / iteration; 10 random seeds / method
Opponent ladder	MCCFR opponents trained for 50, 200, 1000, 5000 iterations
Evaluation	3000 episodes per method-opponent pairing; paired comparisons matched by training seed
Statistics	Paired bootstrap 95% CI with 20,000 resamples
Policy net	MLP $(256, 256)$
Regret / advantage net	MLP $(128, 128)$
Optimizer	Adam, learning rate 10^{-4}
Replay memory	Capacity 10^6; one policy and one advantage update / iteration

Data / State Representation

- No fixed offline dataset. Training samples are generated by self-play traversals.
- Each sample contains an information state together with regret- or strategy-related targets stored in replay memory.
- In Kuhn Poker, the information state is the acting player's private card plus public betting history.
- The model outputs a categorical distribution over legal actions at that state.

Models / Method

- Baseline CFR update

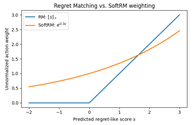
$$\sigma_{\mathrm{RM}}(a \mid I) \propto [\hat{R}(I,a)]_+.$$

- SoftRM update

$$\sigma_{\mathrm{SoftRM}}(a \mid I) = \frac{\exp(\lambda \hat{R}(I,a))}{\sum_{a' \in \mathcal{A}(I)} \exp(\lambda \hat{R}(I,a'))}.$$

- SoftRM changes only the final regret-to-policy map. Traversal generation, replay targets, network size, optimizer, and update schedule remain fixed.
- Entropy-regularized view

$$\pi^*(\cdot \mid I) = \arg\max_{\pi \in \Delta(\mathcal{A}(I))} \sum_a \pi(a \mid I)\hat{R}(I,a) + \tfrac{1}{\lambda} H(\pi(\cdot \mid I))$$

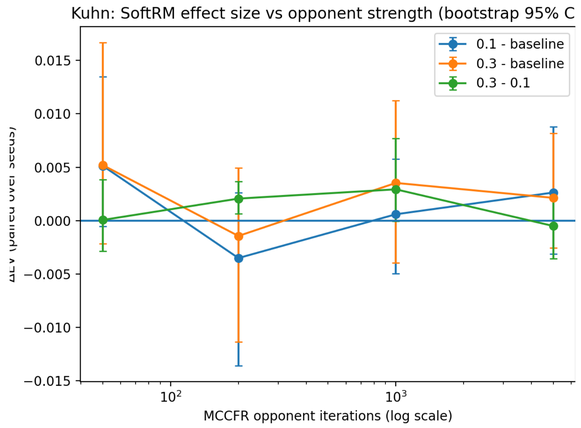


RM clips negative scores to zero. SoftRM keeps all actions active and compresses score gaps when $\lambda \hat{R}$ is small.

Key Result

SoftRM changes only the regret-to-policy decoder inside Deep CFR. In Kuhn Poker, $\lambda \in \{0.1, 0.3\}$ leaves cross-play EV nearly unchanged, paired bootstrap 95% CIs include zero at every opponent strength, policy entropy shifts only slightly ($0.690633 \to 0.691124/0.691128$), and direct entropy-loss regularization also shows no visible gain.

Results



Paired SoftRM effect vs. baseline. Error bars show bootstrap 95% CI.

Method	EV vs 5000	Entropy
Baseline	−0.1551	0.690633
SoftRM 0.1	−0.1525	0.691124
SoftRM 0.3	−0.1530	0.691128

Check	Outcome
Paired 95% CI	Includes zero at every opponent strength
Seed effect	Larger than method effect
Entropy shift	$\Delta H \approx 4.9 \times 10^{-4}$
Entropy-loss ablation	No visible EV / entropy gain

- 200-iteration MCCFR is harder than 5000-iteration MCCFR for all methods.
- Iteration count is only a rough opponent-strength proxy in this setting.

- Mean EV curves remain nearly overlapping across the MCCFR ladder.
- Example paired effects for SoftRM $\lambda = 0.1$: $+0.0051$ EV at MCCFR-50, $+0.0026$ EV at MCCFR-5000.
- Both entropy-smoothing routes leave the learned policy almost unchanged.

Discussion

- Predicted score magnitudes stay small, so $\lambda \hat{R}$ remains near zero for $\lambda \in \{0.1, 0.3\}$.
- Because $\lambda \hat{R}$ stays small, SoftRM keeps the same general action ranking but makes the policy less sharp.
- That change is too small to noticeably alter self-play, so the replay data and learned policy remain close to baseline.
- Later supervised targets remain aligned with the baseline trajectory, and the final average strategy stays close as well.
- In a game as small as Kuhn Poker, weak decoder-level smoothing may simply leave too little room for the training trajectory to separate from baseline.

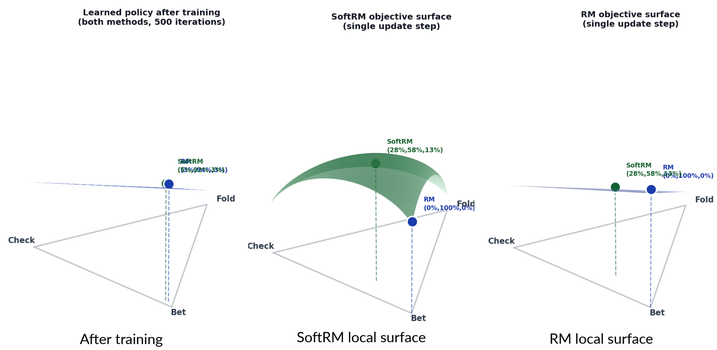
Ablation: Direct Entropy Regularization

$$L_{\mathrm{total}} = L_{\mathrm{strategy}} - \alpha H(\hat{p})$$

Adds an entropy bonus directly to the strategy-network objective, biasing outputs toward higher-entropy policies during supervised replay updates.

- More direct than SoftRM: changes the training loss, not just the final regret-to-policy decoder.
- Intended effect: larger spread in predicted action probabilities.
- Observed effect: minimal visible change in cross-play EV.
- Mean policy entropy also changed only marginally.
- Conclusion: modest entropy pressure was too weak to materially shift the training distribution.

Discussion: Optimization Landscape



After training | SoftRM local surface | RM local surface

Baseline RM can collapse toward a vertex when one action dominates positive-part regret. SoftRM flattens the plane when logits are small, so early updates diffuse across nearby actions rather than concentrating immediately.

- The regret scores stay small enough that SoftRM behaves almost linearly.
- It keeps the same rough action preferences, but softens the differences between them.
- That softening is not strong enough to meaningfully change what the policy does during self-play.
- Both methods therefore end up with nearly the same final average strategy.

Small-logit approximation

$$\mathrm{softmax}(\lambda \hat{R})_a \approx |\mathcal{A}(I)|^{-1} + \tfrac{\lambda}{|\mathcal{A}(I)|}\left(\hat{R}_a - \bar{\hat{R}}\right)$$

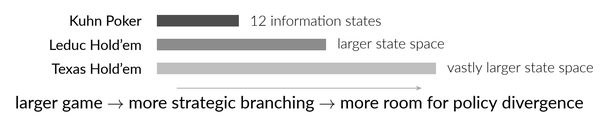
when $\|\lambda \hat{R}\| \ll 1$.

Causal chain

small $\hat{R}$ $\to$ small $\lambda \hat{R}$ $\to$ flatter local basin $\to$ similar replay distribution $\to$ similar final policy

Conclusion / Future Work

- The null result makes sense: with $\lambda \hat{R}$ staying small, SoftRM only slightly smooths the policy, so training samples stay close to baseline and the final learned strategy barely changes.
- In Kuhn Poker, the game is small enough that decoder-level smoothing may have limited room to produce different long-run behavior.



- Move to larger imperfect-information games such as Leduc, where small policy perturbations can change visitation more materially.
- Apply entropy pressure earlier in the learning pipeline, or use adaptive temperatures instead of fixed mild smoothing.

References

[1] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. *Deep counterfactual regret minimization.* ICML.
[2] Ryan D'Orazio, Dustin Morrill, James R. Wright, and Michael Bowling. 2020. *Alternative function approximation parameterizations for solving games: An analysis of f-regression counterfactual regret minimization.* AAMAS.
[3] Michael Johanson, Martin Zinkevich, and Michael Bowling. 2007. *Computing robust counter-strategies.* NeurIPS 20.
[4] Marc Lanctot, Kevin Waugh, Martin Zinkevich, and Michael Bowling. 2009. *Monte Carlo sampling for regret minimization in extensive games.* NeurIPS 22.
[5] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2007. *Regret minimization in games with incomplete information.* NeurIPS 20.