SAFE TEST-TIME REINFORCEMENT LEARNING FOR IMPERFECT INFORMATION GAMES

Ondřej Kubíček
Czech Technical University
Carnegie Mellon University

Viliam Lisý
Artificial Intelligence Center
Czech Technical University

Tuomas Sandholm
Carnegie Mellon University
Strategy Robot, Inc.
Strategic Machine, Inc.
Optimized Markets, Inc.

ABSTRACT

Using additional computation during inference improves performance across domains ranging from games to language models. In this work, we explore applying additional test-time training in imperfect-information games. We focus on two-player zero-sum games and show that modern policy-gradient algorithms, which converge to equilibria when applied to the full game, can produce highly exploitable strategies when applied locally at test time. This phenomenon, previously observed in tabular settings, persists in the function approximation regime. We extend safe subgame-solving techniques based on gadget games from the tabular setting to reinforcement learning and show that they can prevent this degradation. Scaling these methods to more complex domains may require learned generative models of the environment, as test-time training demands the ability to generate states and trajectories on the fly, and restarting a simulator from the current position may not always be feasible. More broadly, our findings are relevant beyond games, as LLM-based agents are increasingly trained via reinforcement learning, and similar safeguards may be necessary when trained in adversarial settings.

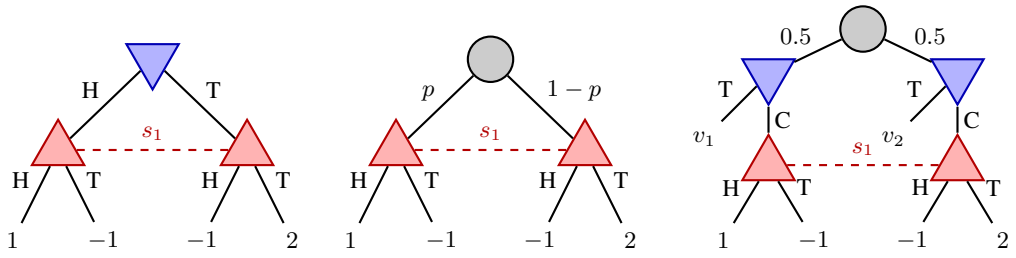
1 INTRODUCTION

Subgame solving is a technique for playing large games by decomposing them into smaller parts to be addressed separately. The major advantage of this is that compute can be focused in real time on positions that actually occur during play (Gilpin & Sandholm, 2006). In imperfect-information games, subgame solving is notoriously difficult, as players cannot observe the full state of the game (Gilpin & Sandholm, 2006; Burch et al., 2014; Ganzfried & Sandholm, 2015). Subgame solving methods must therefore consider a set of possible current world states, and the resulting strategy depends on the probability distribution over those states (Burch et al., 2014; Moravčík et al., 2016; Brown & Sandholm, 2017). Subgame solving techniques can be divided into two categories based on the world states they consider. When a player faces a decision, *knowledge-limited subgame solving (KLSS)* constructs a subgame from all world states that the player cannot distinguish (Zhang & Sandholm, 2021; 2026; Liu et al., 2023).[1] *Public state subgame solving (PSSS)* constructs a subgame from the set all world states, such that it is *common knowledge* between players, the game is in a state from this set. This is a superset of world states used for KLSS (Gilpin & Sandholm, 2006; Burch et al., 2014; Ganzfried & Sandholm, 2015; Moravčík et al., 2016; Brown & Sandholm, 2017; Moravčík et al., 2017; Brown & Sandholm, 2018; Schmid et al., 2023; Kubíček et al., 2024).

Recent research has produced policy-gradient algorithms that converge to various types of equilibria even in large imperfect-information games (Perolat et al., 2021; 2022; Sokota et al., 2023; 2025; Masaka et al., 2025; Kalogiannis & Farina, 2025). However, these algorithms assume that the player follows the trained strategy throughout play and do not provide a mechanism for using additional computation to improve decisions at specific points encountered during a match.

Recently, computation at test time has been introduced into policy-gradient algorithms for imperfect-information games. The *update-equivalence framework* simulates a single gradient step of the train-

[1] More generally, KLSS can construct a subgame from more world states



(a) Original game (b) Subgame of Player 1 (c) Resolving gadget game of Player 1

Figure 1: Modified version of Matching Pennies with Player 2 acting first.

ing algorithm to improve the strategy local to the current decision. It was used in the superhuman Stratego agent (Sokota et al., 2024; 2025). However, *its safety guarantees hold only for a single step and cannot be maintained beyond that*, because it assumes static opponent in the past. An alternative approach explicitly builds subgames, which are solved using tabular algorithms (Kubíček et al., 2024; Kubíček & Lisý, 2026). This approach involves complex training, and the tabular algorithm limits its scalability.

In this paper, we combine elements of both approaches: we use the same learning algorithm for training and testing, but adapt the sampling strategy at test time to mitigate the risk of converging to exploitable strategies. Our work is a first step toward a fully scalable subgame solving algorithm that uses only interaction trajectories generated by a simulator. It does not yet include nested or depth-limited subgame solving. Moreover, it requires constructing the entire game tree to obtain necessary statistics and relies on domain-specific properties. We discuss how each of these limitations can be addressed by future research in Section 5.

2 Example and Background on Gadget Games

We illustrate the well-documented problem of naive subgame solving on a modified version of Matching Pennies. In this game, two players each choose a face of a coin: Heads (H) or Tails (T). If the faces match, Player 1 wins; otherwise, Player 2 wins. In our modified version, if both players choose Tails, the reward is doubled, breaking the game's symmetry. We model this as a sequential game where Player 2 chooses first and Player 1 acts without observing Player 2's choice. The game tree is shown in Figure 1a.

This game has a unique Nash equilibrium in which both players play Heads with probability 0.6. Neither player wants to deviate, since the expected value of each action is 0.2. Now consider a subgame in which Player 1 faces a decision, knowing that Player 2 has already acted. The game tree of this subgame is shown in Figure 1b. Player 1's optimal strategy depends on the belief about Player 2's action. Let p denote the probability that Player 2 played Heads. If $p > 0.6$, Player 1's optimal strategy is always to play Heads. If $p < 0.6$, the optimum is always to play Tails. If $p = 0.6$, any strategy is optimal in the subgame, so the solution depends entirely on the algorithm used. LP-based methods will typically converge to a pure strategy, while regret-minimization methods converge to the uniform strategy (Koller et al., 1996; Zinkevich et al., 2007; Tammelin, 2014). Neither of these is a Nash equilibrium of the original game. For this reason, this approach is called *unsafe subgame solving*.

Safe subgame solving uses gadget games, which are based on a pretrained blueprint strategy for the resolving player, that is improved during gameplay. The gadget game modifies the subgame so that the opponent does not play a fixed strategy in the past but can instead choose any adversarial belief over the initial states. The resulting strategy is guaranteed to perform at least as well as the blueprint. In the worst case, the gadget game reconstructs the blueprint strategy; empirically, it often leads to significant gains (Moravčík et al., 2017; Brown & Sandholm, 2018; 2019; Schmid et al., 2023; Zhang & Sandholm, 2026; Kubíček et al., 2026).

In this work, we use the *resolving gadget game*, which adds one decision node for Player 2 before each initial node of the subgame. At these new nodes, Player 2 can either continue into the sub-

game (C) or terminate (T) and receive the value of its best response against the blueprint. The initial chance node probabilities are proportional to the product of Player 1's and chance's *reaches*, which are player's contribution to the probability of sampling the world state. The chance node does not carry information about Player 2's strategy. In any Nash equilibrium of the resolving gadget game, Player 1's subgame strategy is guaranteed to be at least as good as the blueprint.

Figure 1c shows the resolving gadget game for Player 1. The values v_1 and v_2 depend on the blueprint. For the uniform blueprint, $v_1 = 0$ and $v_2 = 0.5$, while for Nash, $v_1 = 0.2$ and $v_2 = 0.2$.

3 Learning in the Gadget Game

A common approach to reinforcement learning in games, which has been scaled to create strong agents in Stratego (Perolat et al., 2022; Sokota et al., 2025), is to sample entire trajectories and train in actor-critic architecture. The actor learns a policy, and the critic estimates the expected value of a player following that policy.

In general, algorithms designed to converge to optimal strategies in single-player settings do not converge to Nash equilibria in two-player zero-sum imperfect-information games. One option to guarantee convergence to Nash equilibria is to introduce a regularization policy (Perolat et al., 2021; Sokota et al., 2023) that modifies the *utility function* of the underlying game. The algorithms either adjust the regularization strength or modify the regularization policy throughout training.

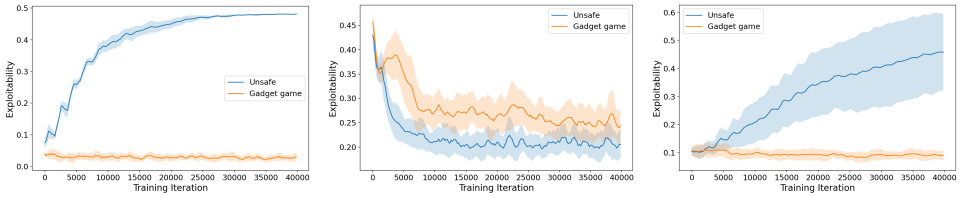
While these algorithms provably approximate Nash equilibria, they require sampling trajectories from the start of the game. Naively applying the same approach at test time does not guarantee that trajectories pass through the current decision point, limiting its effectiveness under a fixed compute budget. The update-equivalence framework improves upon this by applying KLSS. It samples world states consistent with the acting player's information based on the learned belief, then samples a trajectory from that state (Sokota et al., 2024; 2025). This approach simulates a single gradient step, which limits the potential improvement.

To enable more substantial improvements, we adopt the gadget games from the tabular setting. We simulate the resolving gadget game by introducing a third network that handles only Player 2's gadget decisions, while continuing to train the blueprint actor-critic on the subgame. We precompute the counterfactual best response values and reaches P_1, P_c to each initial world state w. On each iteration, we sample an initial world state proportional to the product of Player 1's and chance's reach probabilities, unroll the trajectory to the end of the game, and train all networks with the same policy-gradient algorithm used during training, with the only change that the loss is weighted by the probability p_c of the gadget actor predicting to continue. We use the CBV and the value estimate of the trajectory to train the gadget actor. This training mimics the tabular usage of the gadget game, ensuring that the discovered strategy cannot increase the exploitability. We cannot fully recover this guarantee because the strategy's parameters are shared, meaning that changing it at one point can affect it at all other points. We provide a training schema in Section A.

4 Experiments

We evaluate our approach on three games: Rock-Paper-Scissors, Imperfect-Information Goofspiel with 5 cards and Battleships on a 2×2 board with a single ship of size 2. We use Regularized Nash Dynamics as the policy-gradient algorithm (Perolat et al., 2021; 2022). For each game, we train 3 blueprint networks with different random seeds. Then, for selected training checkpoints, we train 3 subgame networks (again with different seeds), each trained in a subgame after one decision by each player.

Figure 2 shows the exploitability at a representative checkpoint. Additional checkpoints are provided in the Appendix. The results mirror those of tabular methods: in some games, such as Goofspiel, naive subgame solving produces strategies with lower exploitability than the blueprint, while in others, such as Battleship and Rock-Paper-Scissors, it converges to highly exploitable strategies. In contrast, the gadget game approach does not significantly increase exploitability. In the tabular setting, gadget games guarantee non-increasing exploitability; with function approximation, exploitability can increase slightly, but the degradation is far less severe than with the naive approach.



(a) Rock-Paper-Scissors (b) Goofspiel 5 (c) Small Battleship

Figure 2: Exploitability of the subgame-solving strategy using different subgame solving.

5 Discussion and Future Work

We have studied subgame solving in two-player zero-sum games by extending the gadget games to the reinforcement learning setting. In small games, we have shown that naively continuing training in a subgame can degrade performance, and that simulating the gadget game prevents this degradation. Our results are preliminary, as our current approach makes several simplifying assumptions that limit its generality and scalability. We discuss these limitations and potential solutions below.

Exact counterfactual best-response values. In the experiments, we compute exact counterfactual best-response values, which is intractable in large games. In on-policy sampling, the critic estimates the counterfactual value of an information set. However, this value represents an expected value if both players follow the actors strategy, which may be different than the best-response value during training. Nevertheless, it may serve as a reasonable approximation, consistent with the use of counterfactual values in nested depth-limited solving (Moravčík et al., 2017; Brown & Sandholm, 2018; Brown et al., 2018; Kubíček et al., 2026). We have further explored this in Section B.

Exact reaches and state enumeration. We exploit the small game sizes to extract exact reach probabilities from the blueprint actor and to enumerate all states in the subgame, from which we then sample. Neither extracting exact reaches nor enumerating states is tractable in large games. Belief generation in partially observable systems has been studied (Seitz et al., 2021; Solinas et al., 2025), and we believe these techniques can be extended to our setting, enabling sampling of states proportional to blueprint reach probabilities conditioned on the current observation.

Nested subgame solving. Both issues above become more pronounced in nested subgame solving, where each encountered subgame is solved by reusing strategies from previously solved subgames (Brown & Sandholm, 2017; Moravčík et al., 2017). The new solution influences every subsequent subgame through both the counterfactual values and the reach probabilities. One approach is to ignore changes from previous subgames and always assume the resolving player followed the blueprint, though this may lead to unsafe strategies. Alternatively, one could reuse both the critic and the belief model from previous subgame solves, but this requires correction terms since beliefs in the subgame may be biased because the gadget game does not directly represent the original game.

Depth-limited solving. We sample trajectories from the subgame to the end of the game. In games like Stratego, trajectories can span up to a thousand moves, which limits the number of samples within a fixed budget. Naively using critic at a depth limit may yield exploitable strategies (Kovařík et al., 2023). Value function in the form of reward continuation policies (Brown et al., 2018; Brown & Sandholm, 2019) has been trained alongside policy-gradient algorithms, but this modification loses all theoretical guarantees (Kubíček et al., 2024). Potential future work would be to develop a method which would be able to train this value function, while retaining theoretical guarantees.

Practical considerations. Based on our results, the naive approach does not immediately degrade strategy quality, so the safety provided by gadget games may not be useful in practical scenarios when using a limited compute budget. Also, the gadget game ensures that exploitability does not increase, but in head-to-head play against a fixed opponent, lower exploitability does not guarantee better performance.

ACKNOWLEDGMENTS

This work is supported by National Science Foundation grant RI-2312342 , the Vannevar Bush Faculty Fellowship ONR N00014-23-1-2876, the Czech Science Foundation GA25-18353S, the Grant Agency of the Czech Technical University in Prague (SGS23/184/OHK3/3T/13). The access to the computational infrastructure of the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 "Research Center for Informatics" is also gratefully acknowledged. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

REFERENCES

Noam Brown and Tuomas Sandholm. Safe and nested subgame solving for imperfect-information games. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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 and Tuomas Sandholm. Superhuman ai for multiplayer poker. *Science*, 365(6456): 885–890, 2019.

Noam Brown, Tuomas Sandholm, and Brandon Amos. Depth-limited solving for imperfect-information games. In , *Advances in Neural Information Processing Systems*, 2018.

Neil Burch, Michael Johanson, and Michael Bowling. Solving imperfect information games using decomposition. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pp. 602–608, 2014.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In , *Proceedings of the 35th International Conference on Machine Learning*, pp. 1407–1416, 2018.

Sam Ganzfried and Tuomas Sandholm. Endgame solving in large imperfect-information games. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 37–45, 2015.

Andrew Gilpin and Tuomas Sandholm. A competitive texas hold'em poker player via automated abstraction and real-time equilibrium computation. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 1007–1013, 2006.

Daniel Hennes, Dustin Morrill, Shayegan Omidshafiei, Rémi Munos, Julien Perolat, Marc Lanctot, Audrunas Gruslys, Jean-Baptiste Lespiau, Paavo Parmas, Edgar Duèñez Guzmán, and Karl Tuyls. Neural replicator dynamics: Multiagent learning via hedging policy gradients. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 492–501, 2020.

Fivos Kalogiannis and Gabriele Farina. Policy gradient methods converge globally in imperfect-information extensive-form games. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Vojtěch Kovařík, Dominik Seitz, Viliam Lisý, Jan Rudolf, Shuo Sun, and Karel Ha. Value functions for depth-limited solving in zero-sum imperfect-information games. *Artificial Intelligence*, 314 (C), 2023.

Ondřej Kubíček and Viliam Lisý. Look-ahead reasoning with a learned model in imperfect information games. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. Early version on arXiv, 2025.

Ondřej Kubíček, Neil Burch, and Viliam Lisý. Look-ahead search on top of policy networks in imperfect information games. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

Ondřej Kubíček, Viliam Lisý, and Tuomas Sandholm. Equilibrium refinements improve subgame solving in imperfect-information games, 2026. arXiv preperint.

Weiming Liu, Haobo Fu, Qiang Fu, and Yang Wei. Opponent-limited online search for imperfect information games. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 21567–21585, 2023.

Wataru Masaka, Mitsuki Sakamoto, Kenshi Abe, Kaito Ariu, Tuomas Sandholm, and Atsushi Iwasaki. On the power of perturbation under sampling in solving extensive-form games, 2025.

Matej Moravčík, Martin Schmid, Karel Ha, Milan Hladik, and Stephen Gaukrodger. Refining subgames in large imperfect information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Matej Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, 356(6337):508–513, 2017.

Julien Perolat, Remi Munos, Jean-Baptiste Lespiau, Shayegan Omidshafiei, Mark Rowland, Pedro Ortega, Neil Burch, Thomas Anthony, David Balduzzi, Bart De Vylder, Georgios Piliouras, Marc Lanctot, and Karl Tuyls. From poincaré recurrence to convergence in imperfect information games: Finding equilibrium via regularization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 8525–8535, 2021.

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(6623):990–996, 2022.

Martin Schmid, Matej Moravčík, Neil Burch, Rudolf Kadlec, Josh Davidson, Kevin Waugh, Nolan Bard, Finbarr Timbers, Marc Lanctot, G. Zacharias Holland, Elnaz Davoodi, Alden Christianson, and Michael Bowling. Student of games: A unified learning algorithm for both perfect and imperfect information games. *Science Advances*, 9(46):eadg3256, 2023.

Dominik Seitz, Nikita Milyukov, and Viliam Lisý. Learning to guess opponent's information in large partially observable games. In *Proceedings AAAI Workshop Reinforcement Learning in Games*, 2021.

Samuel Sokota, Ryan D'Orazio, J Zico Kolter, Nicolas Loizou, Marc Lanctot, Ioannis Mitliagkas, Noam Brown, and Christian Kroer. A unified approach to reinforcement learning, quantal response equilibria, and two-player zero-sum games. In *The Eleventh International Conference on Learning Representations*, 2023.

Samuel Sokota, Gabriele Farina, David J Wu, Hengyuan Hu, Kevin A. Wang, J Zico Kolter, and Noam Brown. The update-equivalence framework for decision-time planning. In *The Twelfth International Conference on Learning Representations*, 2024.

Samuel Sokota, Eugene Vinitsky, Hengyuan Hu, J. Zico Kolter, and Gabriele Farina. Superhuman ai for stratego using self-play reinforcement learning and test-time search, 2025.

Christopher Solinas, Radovan Haluska, David Sychrovsky, Finbarr Timbers, Nolan Bard, Michael Buro, Martin Schmid, Nathan R. Sturtevant, and Michael Bowling. Neural bayesian filtering, 2025.

Oskari Tammelin. Solving large imperfect information games using CFR+, 2014.

Brian Hu Zhang and Tuomas Sandholm. Subgame solving without common knowledge. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 2021.

Brian Hu Zhang and Tuomas Sandholm. General search techniques without common knowledge for imperfect-information games, and application to superhuman fog of war chess. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2026. Early version on arXiv, 2025.

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.

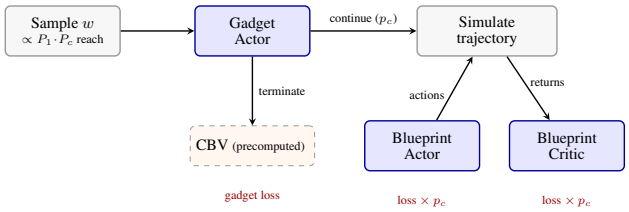


Figure 3: Schema for a single training trajectory. It starts with sampling the initial world state.

A Training schema

Figure 3 summarizes the subgame training, when using the additional gadget actor.

B Additional Experiments

In Figures 4 to 6 we show the exploitability curves for all the checkpoints trained for Rock-Paper-Scissors, Imperfect Information Goofspiel 5, and Battleship, respectively. We have also included 3 more versions of subgame solving. We have also included a max-margin gadget game, which is a different type of gadget (Moravčík et al., 2016). For both resolving gadget and max-margin gadget, we have also included a plot showing how the exploitability changes when we do not compute the exact counterfactual best response values but instead use the trained critic. Surprisingly, even though the trained critic lacks theoretical guarantees, its performance is on par with the exact values, suggesting that using the critic may be sufficient.

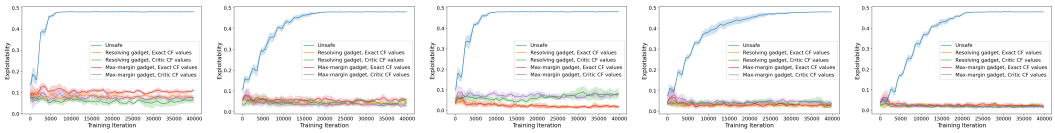


Figure 4: Exploitability of the subgame-solving strategy in Rock-Paper-Scissors using different checkpoints.

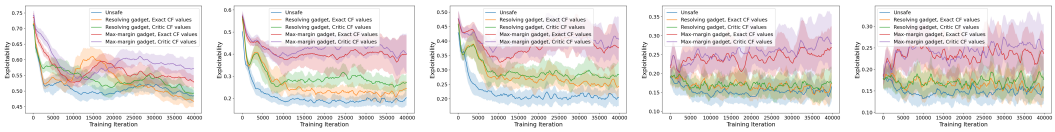


Figure 5: Exploitability of the subgame-solving strategy in Goofspiel 5 using different checkpoints.

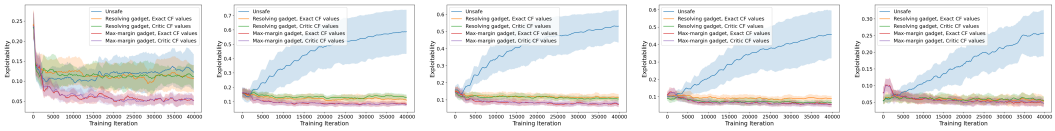


Figure 6: Exploitability of the subgame-solving strategy in Battleship 2×2 with ship of size 2 using different checkpoints.

C Additional experimental details

In our experiments we have used Regularized Nash Dynamics (RNaD) (Perolat et al., 2021; 2022) as the policy-gradient algorithm, which uses Neural Replicator Dynamics to train the actor (Hennes

et al., 2020), V-trace to predict the value estimate for training the critic (Espeholt et al., 2018). RNaD regularizes the reward by the KL-divergence between current policy and the regularization policy. It periodically sets the regularization policy to the current policy. We have used the same algorithm for training the Gadget critic. We have used a Multi-layered Perceptron for each neural network.

When training the actor-critic in the original game, we multiply the loss by the probability of gadging continuing into the subgame. We have clipped the probability to be at least $\epsilon \cdot P_2$, where P_2 is the opponent's reach to the sampled world state if it followed the blueprint and ϵ is a hyperparameter. This has been shown to provide substantial performance improvements in the tabular setting (Kubíček et al., 2026).

We provide all the used hyperparameters in Table 1

Parameter	**Value**
RNaD regularization η	0.2
RNaD regularization policy change	2000
NeuRD β clip	2.2
NeuRD avantage clip	1000
Continue clip ϵ	$5 \cdot 10^{-3}$
MLP hidden size	256
MLP hidden layers	2
Learning rate	$3 \cdot 10^{-4}$
Optimizer	ADAM
ADAM β_1, β_2	$0.0, 0.999$
V-trace λ	1.0
Discount Factor γ	1.0

Table 1: Hyperparameters.

D LLM USAGE

Throughout the writing of this paper we have used Large Language Models. We have checked and verified every output before using it. We have used these LLMs for these tasks:

- Claude Sonnet 4.5, Claude Opus 4.5 and Claude Opus 4.6 as a coding assistant.
- Gemini 3 pro and Claude Opus 4.6 for text formatting.