Zero-Shot Evaluation of Strategies by Playing Imitation Policies

Anonymous authors
Paper under double-blind review

Abstract

How a strategy handles an opponent it has never met, or a condition it has never seen, is not settled by running it against the opponents at hand, since the one that defeats it need not be among them. The strategies that most need an answer often cannot be run at will, leaving only a short record of past matches. Ratings and outcome predictors fit the results of those matches but take no condition as input, while a policy cloned from the record knows only the situations the record reached. The match can still be played, with a policy inferred from the record in place of the original. The simulator supplies the opponent, the condition and the dynamics, so the only unknown is how the strategy responds. We introduce PlayRead, an in-context imitation policy with a retrieval memory, trained on a population of runnable strategies. At each step of a new match, it retrieves recorded moments that resemble the current situation and the responses that followed them, while the match in progress is kept in its context. Training pairs the record of one strategy with its actions against an opponent absent from that record, and a new strategy is evaluated zero-shot by placing its record in the memory, whose values may then be tuned on that record. We evaluate PlayRead on a continuous-control game, Leduc poker, Overcooked and the iterated prisoner's dilemma. In the continuous-control game, playing the inferred policy is substantially more accurate than predicting payoff from the same memory when the record is short or the condition is unseen. In an unseen condition, PlayRead with one recorded episode predicts matchups more accurately than behavioral cloning fitted to eighteen. For most strategies the inferred policy also names the condition in which the strategy is weakest, and it yields an opponent that defeats it. With six open language models in the prisoner's dilemma, PlayRead predicted how they would play one another before they met.

1 Introduction

Can the strengths and weaknesses of an agent be known before it meets its opponents? Knowing them in advance would say which opponents it can be trusted against, under which conditions it breaks down, and which opponent would exploit it. Playing the matches would tell, but the agents for which this matters most usually cannot be run at will. A rival bot or a withdrawn model cannot enter new games, and calling a language model or a live trading rule is costly or risky. What remains is a short record of its behavior against other opponents and in other conditions.

The evaluation is therefore zero-shot, because none of the matches in question appears in the record. What must be recovered from the record is how the strategy responds to the moves of its opponent and to the conditions of the game. The outcome of an unplayed match follows from those responses and not from past results. Two strategies with the same results against the opponents on record can respond differently, and they then diverge against a new opponent or under a new condition. Ratings (Elo, 1978; Chen & Joachims, 2016; Balduzzi et al., 2018; Omidshafiei et al., 2019) and predictors trained on match outcomes (Grover et al., 2018) retain who won and discard how the strategy responded. Behavioral cloning (Pomerleau, 1989) keeps the responses, but only in the situations the record visited, and a new opponent leads the match into others, where its errors compound (Ross et al., 2011). If those responses can be recovered, the unplayed match can be played rather than predicted.

We infer from the record a runnable policy that stands in for the original. A population of runnable strategies provides the prior for the behavior the record does not cover, and payoffs are obtained

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.

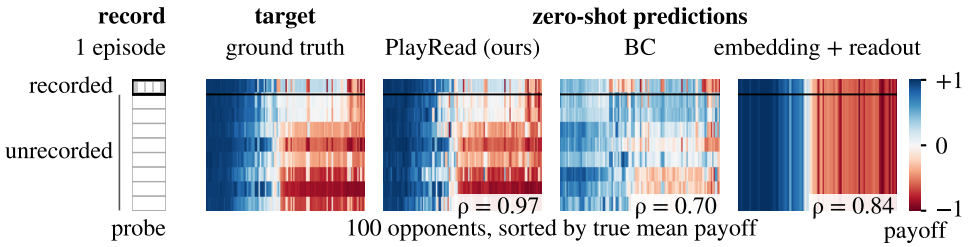

Figure 1: Zero-shot evaluation in the hill game. From one episode of a held-out strategy in one condition (top row), the task is to predict its payoffs against 100 opponents it never faced under all nine conditions. The readout takes no condition as input, so its rows repeat. ρ is the Spearman over the 900 cells for this strategy, chosen at the median gap between PLAYREAD and cloning. Medians over the 200 strategies are 0.95 for untuned PLAYREAD, 0.66 for cloning and 0.76 for the readout.

by playing the inferred policy in a simulator rather than by predicting them (Fig. 1). The simulator supplies the rules, the opponent and the condition, so changing the opponent or the condition changes what is simulated rather than what must be learned. We refer to a model that maps a record to such a policy as a reader. A reader is assessed by whether its policy reproduces the payoffs of the original rather than its actions, because policies whose actions differ little can obtain different payoffs. In the prisoner's dilemma, a single defection against a grudging opponent forfeits cooperation for good.

We introduce PLAYREAD, an in-context imitation policy with a retrieval memory (Duan et al., 2017; Garnelo et al., 2018; Humphreys et al., 2022; Schmied et al., 2025). It stores the record as short moments, each indexed by the situation that precedes it and holding the response that follows, and at every step of a new match retrieves the moments that resemble the current situation. The population is held in the shared parameters, the record in an external memory written at test time, and the match in progress in the context of a causal transformer. The parameters are trained once on a population of runnable strategies, with interactive imitation where the strategies can be queried (Ross et al., 2011; Wen et al., 2020), so a new strategy requires no update to the shared parameters. In a controlled game, playing the inferred policy outperforms scoring the same memory with a learned readout when the record is short or the condition is unseen. Our contributions are the following.

- A formulation of zero-shot evaluation from a short record, which holds out strategies, opponents and conditions separately from the length of the record and factorizes evaluation into what the simulator specifies and what the population and record supply (Sections 2 and 3).
- PLAYREAD, an in-context imitation policy with an external retrieval memory, trained with interactive imitation (Section 3).
- An empirical study across four domains, in which a controlled comparison shows that, with short records or unseen conditions, the advantage comes from playing the inferred policy rather than from a readout of the same memory, and the inferred policies build opponents that defeat a strategy and predict how language models play one another before they meet (Section 4).

2 ZERO-SHOT EVALUATION OF INTERACTING STRATEGIES

We consider two agents that interact in a partially observable stochastic game (Shapley, 1953; Hansen et al., 2004) that we can simulate. At step $t = 1, \dots, T$ of a match the game is in a state $\omega_t \in \Omega$. One agent receives an observation $o_t \in \mathcal{O}$ of that state and takes an action $a_t \in \mathcal{A}$, while its opponent takes a'_t, simultaneously or in turn. A *condition* $e \in \mathcal{E}$ parameterizes the game. It sets the distribution μ_e of initial states, the transition kernel $P_e(\omega_{t+1} \mid \omega_t, a_t, a'_t)$ and the scoring rule (Table 4). A *strategy* π maps the condition and the history $h_t = (o_1, a_1, a'_1, \dots, a_{t-1}, a'_{t-1}, o_t)$ to a distribution over actions, $a_t \sim \pi(\cdot \mid h_t, e)$. The payoff $y(\pi, \pi'; e)$ is the mean score of π against π' in condition e (Table 4).

The *record* of a strategy is a set $\mathcal{D} = \{\xi_1, \dots, \xi_k\}$ of k episodes. An episode $\xi = (e, (o_t, a_t, a'_t)_{t=1}^{T})$ records the condition of one match, what the strategy observed and what both agents did. It contains neither the states ω_t nor the score. A record therefore shows a strategy only at the histories its k matches reached, and only as one sampled action at each. At training time we have access to a population $\Pi_{\text{train}} = \{\pi_1, \dots, \pi_N\}$ of strategies that can be run against any opponent in any condition. At test time a *held-out* strategy $\pi^\star \notin \Pi_{\text{train}}$ is available to the evaluator only through its record $\mathcal{D}^\star$.

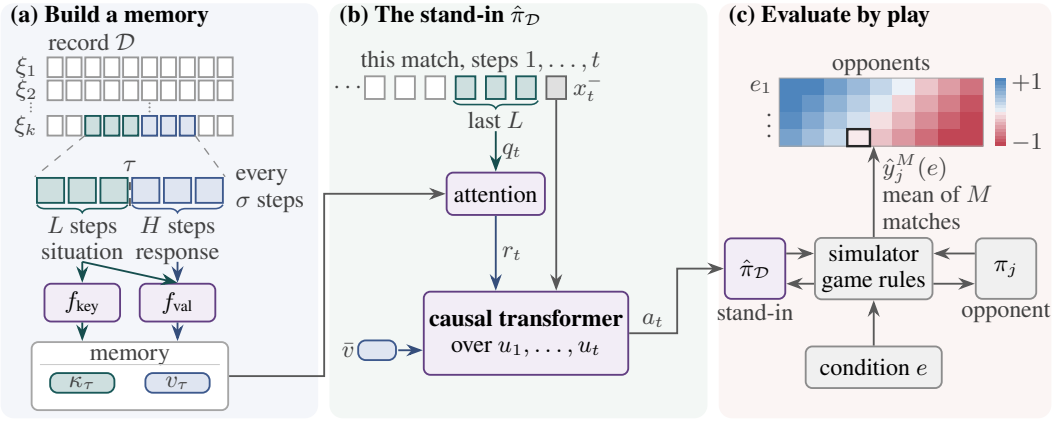

Figure 2: PLAYREAD. (a) Every σ steps a moment is cut from the record. The L steps before an anchor τ, the situation, are encoded into a key, and the same steps continued by the H steps after it, the response, into a value. (b) The last L completed steps query the memory, and a causal transformer combines the retrieved r_t with the current token x_t^- and the mean memory value $\bar{v}$ to emit a_t. (c) The simulator plays a_t against π_j under e, and the mean of M matches fills one cell. Gray is given by the simulator, purple learned, teal marks situations and keys, blue responses and values.

We pose two tasks on the payoffs of $\pi^\star$ against opponents absent from its record and under conditions that may also be absent. Both are defined on payoffs alone, and neither scores predicted actions. The first is matchup prediction. Given a pool $\mathcal{J}$ of runnable opponents that $\pi^\star$ never faced and a condition e that may be absent from $\mathcal{D}^\star$, predict $y(\pi^\star, \pi_j; e)$ for every $j \in \mathcal{J}$. A prediction is scored by rank correlation over the pool, so the task asks for the ordering of the opponents rather than their exact payoff values (Section 4.1). The second concerns exploitability, which a fixed pool does not reveal because it requires a search over opponents. Where the game is zero-sum, the exploitability of $\pi^\star$ within a parameterized family $\{\pi_\psi : \psi \in \Psi\}$ is $\varepsilon(\pi^\star; e) = \sup_\psi y(\pi_\psi, \pi^\star; e)$. Given the same record, the task is to search for ψ maximizing $y(\pi_\psi, \pi^\star; e)$. Any opponent found by this search gives a lower bound on $\varepsilon(\pi^\star; e)$ when played against the original. A stand-in for $\pi^\star$ serves both tasks to the extent that it reproduces the payoffs of the original. Let d be their outcome distance, the largest gap between their payoffs across opponents and conditions. If two opponents differ in payoff against the original by more than $2d$, the stand-in orders them as the original does. Moreover, in a zero-sum game and with d taken over the family, the best counter to the stand-in achieves against the original a payoff within $2d$ of the best counter in the family, both by the triangle inequality.

3 AN IMITATION POLICY WITH A MEMORY OF THE RECORD

PLAYREAD infers a runnable stand-in from a record, plays it, and searches for its best counter (Fig. 2).

3.1 A POLICY THAT READS ITS RECORD

PLAYREAD keeps the record as a retrieval memory of short moments rather than relying on one pooled vector of the record, which averages behavior across situations as a rating averages outcomes (Grover et al., 2018; Papoudakis et al., 2021; Wang et al., 2026). A step of a recorded episode becomes a token $x_t = [o_t, a_t, a'_t]$, and $x_{t_1:t_2}$ denotes the tokens of steps t_1 to $t_2 - 1$. Anchors $\Gamma(\mathcal{D})$ fall every σ steps along each episode. The moment at an anchor τ is stored as a key over the L steps that precede it, its situation, and a value over the same steps continued H past the anchor, the response. At step t of a match the last L completed steps form a query,

$$\kappa_\tau = f_{\text{key}}(x_{\tau-L:\tau}, e_\tau), \qquad v_\tau = f_{\text{val}}(x_{\tau-L:\tau+H}, e_\tau), \qquad q_t = f_{\text{key}}(x_{t-L:t}, e), \tag{1}$$

where e_τ is the condition of the episode containing τ and f_{key} and f_{val} are transformer encoders over the window, with the condition appended to every token (Appendix A). The memory holds one row per anchor, is fixed during a match, and the query attends over it (Vaswani et al., 2017),

$$r_t = \text{Attention}\big(q_t, \{\kappa_\tau\}_{\tau\in\Gamma(\mathcal{D})}, \{v_\tau\}_{\tau\in\Gamma(\mathcal{D})}\big). \tag{2}$$

Query and keys share an encoder, so the current situation is matched against situations in the record. A value covers the steps after its key, as in the external memory of Schmied et al. (2025), so r_t carries the response that a new match has to reproduce rather than the situation itself.

The match in progress. A key covers only the last L steps, one in the hill game, and forgets what the opponent did earlier, which is what a retaliating or adapting strategy responds to. The policy therefore passes its inputs through a causal transformer over the match in progress and a head,

$$u_t = W_x x_t^- + W_r r_t + W_c \bar{v}, \qquad g_t = \mathrm{Dec}\big(u_1, \dots, u_t\big), \qquad \hat{\pi}_{\mathcal{D}}(\cdot \mid h_t, e) = \mathrm{Head}(g_t), \tag{3}$$

where $x_t^- = [o_t, a_{t-1}, a'_{t-1}]$ holds the current observation and the previous actions of both sides, $\bar{v}$, the mean of the memory values, summarizes the whole record, and W_x, W_r and W_c are learned projections, with a position embedding added. We write $\hat{\pi}_{\mathcal{D}} = R_\theta(\mathcal{D})$, with θ shared by every strategy, so a new strategy requires no update to θ.

3.2 Training on a population

The reader is trained once on a population of strategies. Each example pairs a record with a target episode outside it, analogous to the context and target sets of a conditional neural process (Garnelo et al., 2018). The record $\mathcal{D}_i$ holds k episodes of a population strategy π_i against a fixed panel of probe opponents, with k drawn from a set $\mathcal{K}$ so that one reader serves several record lengths, and the reader minimizes

$$\mathcal{L}_{\mathrm{bc}}(\theta) = -\,\mathbb{E}_{i,\,\mathcal{D}_i,\,\tilde{\xi}}\Big[\textstyle\sum_t \log \hat{\pi}_{\mathcal{D}_i}\big(a_t \mid h_t, \tilde{e}\big)\Big], \tag{4}$$

where the target episode $\tilde{\xi}$ is played under a condition $\tilde{e}$ drawn independently of the record against a population opponent absent from $\mathcal{D}_i$ (Appendix B), and a_t and h_t are the action and the history of π_i at its step t. This trains the reader on behavior the record does not reveal.

The population as a prior. Over all measurable readers the minimizer of Eq. (4) is the posterior predictive distribution of behavior (Ortega et al., 2019; Mikulik et al., 2020), with p the distribution of strategies in the population,

$$R^\star(\mathcal{D})(a \mid h, e) \;=\; \mathbb{E}_{\pi \sim p(\cdot \mid \mathcal{D}, h, e)}\big[\,\pi(a \mid h, e)\,\big]. \tag{5}$$

The population thus supplies a prior over strategies, and the record updates it for the strategy at hand. The match in progress is handled separately, since in play it also holds the actions of the stand-in itself. The three sources enter the model separately, the population in the shared parameters, the record in a memory built per strategy and the match in progress in the context of the causal transformer, where a single context window would hold the last two together (Xu et al., 2022; Laskin et al., 2023).

Interactive imitation on the population. Behavior on the states a strategy visits does not determine behavior on the states its inferred policy visits, and the discrepancy compounds over a match (Fig. 3d, Ross et al., 2011; Wen et al., 2020). Training strategies, unlike held-out ones, can be queried anywhere, so we add DAgger-style interactive imitation on the states the inferred policy reaches,

$$\tilde{a}_t \sim \beta \pi_i(\cdot \mid h_t, e) + (1-\beta)\hat{\pi}_{\mathcal{D}_i}(\cdot \mid h_t, e), \quad \mathcal{L}_{\mathrm{ii}}(\theta) = -\,\mathbb{E}_{i,\,h_t \sim \nu_{\theta,i}}\big[\log \hat{\pi}_{\mathcal{D}_i}\big(a_t^i \mid h_t, e\big)\big], \tag{6}$$

where $\tilde{a}_t$ is played against a training opponent in a seen condition e, a_t^i is the action of the training strategy at the reached history, $\nu_{\theta,i}$ the distribution of histories these matches produce, and β decays from one to zero over the first half of training. The objective is $\mathcal{L}_{\mathrm{bc}} + \mathcal{L}_{\mathrm{ii}}$ (Algorithm 1).

3.3 Evaluating a held-out strategy

A strategy π playing π' in condition e induces a distribution over trajectories $\zeta = (\omega_{1:T+1}, a_{1:T}, a'_{1:T})$, which, unlike recorded episodes, hold the states of the game,

$$\mathbb{P}_{\pi,\pi',e}(\zeta) = \mu_e(\omega_1) \prod_{t=1}^{T} \pi\big(a_t \mid h_t, e\big)\, \pi'\big(a'_t \mid h'_t, e\big)\, P_e\big(\omega_{t+1} \mid \omega_t, a_t, a'_t\big), \tag{7}$$

where h'_t is the history seen by the opponent. Evaluating a held-out strategy composes this distribution from parts that come from different places,

$$\mathbb{P}(\zeta \mid e, \pi_j, \mathcal{D}) = \underbrace{\mu_e(\omega_1) \textstyle\prod_t P_e(\omega_{t+1} \mid \omega_t, a_t, a'_t)}_{\text{rules, known}} \underbrace{\textstyle\prod_t \pi_j(a'_t \mid h'_t, e)}_{\text{opponent, known}} \underbrace{\textstyle\prod_t \hat{\pi}_{\mathcal{D}}(a_t \mid h_t, e)}_{\text{strategy, inferred}}. \tag{8}$$

A condition and an opponent that never occurred together are therefore assembled by the simulator. Only the strategy factor is learned, and in the Bayes-optimal limit of Eq. (4) it is the posterior predictive of Eq. (5). The prediction against π_j is the payoff of the inferred policy, $\hat{y}_j(e) = y(\hat{\pi}_{\mathcal{D}}, \pi_j; e)$, estimated by the mean score $\hat{y}_j^M(e)$ of M simulated matches.

Tuning the memory on the record. A few gradient steps on the instance can reduce the amortization gap a forward pass leaves (Cremer et al., 2018; Kim et al., 2018). For memory tuning, part of the record is hidden from the memory, and the memory values v_τ are updated so that the policy predicts the hidden actions, with the keys and all parameters fixed. No outcome is used, and the settings are chosen on validation strategies per record budget (Appendix A).

3.4 Searching for an opponent

The stand-in can be run as often as needed, so opponent search does not require access to the original. Given a family $\{\pi_\psi : \psi \in \Psi\}$ chosen by the evaluator, let $F(\psi) = y(\pi_\psi, \hat{\pi}_{\mathcal{D}}; e)$ be the payoff of a candidate against the stand-in. We seek $\psi^\star \in \arg\max_{\psi\in\Psi} F(\psi)$, treat F as a black-box objective estimated from simulator rollouts, and approximate the maximizer by the cross-entropy method (de Boer et al., 2005) (Eq. (10)), or find it exactly by enumeration in the prisoner's dilemma and by best response in Leduc (Appendices A, O and P). To verify it, we play the returned opponent against the original, which the evaluation holds out as ground truth. In a zero-sum game its payoff there bounds the exploitability of the strategy within the family from below.

4 Experiments

We first ask whether a short record suffices to predict matchups in a game we control (Section 4.2), then where the advantage comes from (Section 4.3), then whether it holds in three standard domains (Section 4.4), and finally what the inferred policies permit beyond prediction (Section 4.5). Unless stated otherwise we report the Spearman correlation ρ between predicted and true payoffs, averaged over the held-out strategies and three training seeds.

4.1 Setup

Strategies and pools. We evaluate on four domains whose interactions differ in kind (Table 4). The hill game is a continuous-control game with nine conditions of damping and wind, and the only domain in which conditions can be withheld from every record and the population and the reader varied at will, so Sections 4.2, 4.3 and 4.5 use it. Overcooked shares the payoff between partners, Leduc poker has exact payoffs, and the iterated prisoner's dilemma is neither zero-sum nor cooperative. Each held-out strategy is scored against a pool it never faced, of 100 opponents in the first three domains and 215 published strategies in the last. True payoffs are exact in poker and estimated from played matches elsewhere, and the ceiling plays the originals through the same estimator (Appendices D and N to P). In the hill game (Lowe et al., 2017) the strategies are built around three archetypes that we made non-transitive, records come from six fixed probes outside the pool, and three conditions are unseen, absent from every record and training target (Appendix K).

Baselines. Ratings on a partial table (Elo, 1978; Chen & Joachims, 2016) and an embedding with a pairwise readout (Grover et al., 2018; Wang et al., 2026) build no policy. Cloning (Pomerleau, 1989) and a pretrained policy fine-tuned per strategy (Tang et al., 2025) build one without a population. Neural processes (Garnelo et al., 2018; Kim et al., 2019), a pooled in-context embedding (Jing et al., 2024a;b) and Prompt-DT (Xu et al., 2022) read a record as we do. Each baseline follows its published procedure with the same records, and the same population where it uses one (Appendix E).

4.2 Matchup prediction from a short record

Table 1 reports the hill game. Fitting a policy to a single strategy fails from one episode, since that episode constrains behavior only in the situations it visits and a new opponent drives the match into others. Population-trained readers recover much of the ordering from one episode, approaching the ceiling obtained by playing the originals through the same estimator. The budget of eight simulated matches and the noise of the 32-start truth explain the distance of that ceiling from one, and the reader explains the distance from the ceiling. A rating infers a position from outcomes, so we fit it to the true payoffs of $\pi^\star$ against 40 of the 100 opponents it must rank, which gives it part of the answer. A scalar rating cannot represent the cycle, while a two-vector model can (Chen & Joachims, 2016), but neither varies with the condition, and Blade–Chest degrades when the condition is unseen.

Table 1: Matchup prediction in the hill game, Spearman across 100 opponents that none of the 200 held-out strategies ever faced. A seen condition appears in the records and training targets of the population, though not necessarily in the record evaluated, and an unseen one in neither. Readers use 5000 training strategies and three seeds. Elo and Blade–Chest read no episodes and are given 40% of the results of the strategy instead. Bold marks the best method that sees no such results. For PlayRead a seed moves a cell by at most 0.007, and a 95 percent bootstrap interval is about 0.03.

	1 episode		18 episodes	
Method	seen	unseen	seen	unseen
outcome predictors, which need labeled matches				
Elo with 40% of its own results	0.519	0.533	0.519	0.533
Blade–Chest with 40% of its own results	0.939	0.801	0.939	0.801
embedding with a pairwise readout	0.832	0.729	0.952	0.805
the mean memory value of PlayRead with that readout, no play	0.796	0.735	**0.953**	0.815
alternatives that fit a policy to each strategy				
behavioral cloning, per strategy	0.484	0.531	0.906	0.845
pretrained policy, fine-tuned per strategy	0.426	0.464	0.873	0.796
in-context readers trained as published				
pooled in-context embedding	0.860	0.852	0.889	0.883
CNP reader	0.870	0.849	0.918	0.881
ANP reader	0.875	0.812	0.924	0.875
Prompt-DT	0.855	0.834	0.888	0.852
PlayRead without memory tuning	0.903	0.877	0.951	0.898
PlayRead	**0.904**	**0.882**	**0.953**	**0.900**
ceiling of the estimator	0.966	0.925	0.966	0.925

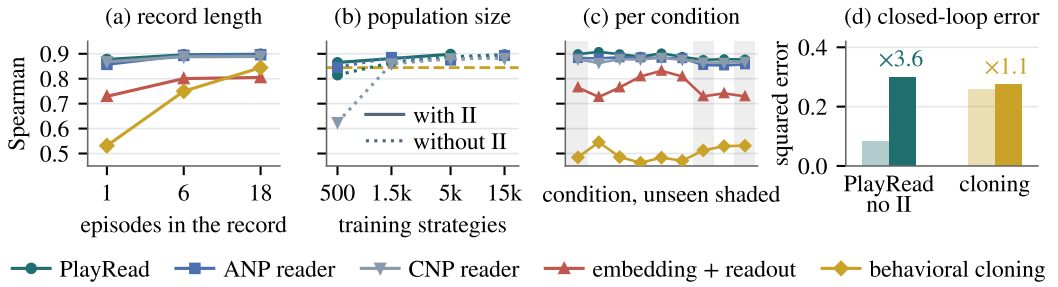

Figure 3: Why PlayRead works in the hill game, with the ANP and CNP readers trained with our recipe. (a) By record length, unseen condition. (b) By population size from 18 episodes, unseen condition, solid with interactive imitation (II), dotted without, dashed cloning. (c) From a one-episode record, accuracy in each of the nine evaluation conditions, unseen ones shaded. (d) Squared action error of PlayRead trained on 500 strategies without II and of cloning, on the states the original visits (light) and those the inferred policy reaches (dark), with their ratio (Appendix H).

4.3 Where the advantage comes from

Playing against predicting. The advantage could come from the memory or from playing the policy it yields. We freeze a trained PlayRead and train the pairwise readout of the outcome baseline on the mean memory value its policy receives, with the labeled matches that baseline is given, so neither side is short of supervision. As an input to that readout the mean memory value is about as accurate as the outcome-trained embedding. With eighteen episodes in a seen condition play and readout tie, so the memory holds the information the payoffs require. Without memory tuning, play is ahead by 0.083 when the condition changes, by 0.107 when the record shortens and by 0.142 when both happen (Table 1). A direct readout maps a pair of strategy summaries to one payoff and cannot vary with the condition, and it is trained on five million labeled matches. PlayRead instead plays the inferred policy in that condition, so changing the condition changes the simulated interaction.

Identity in the record, not the parameters. Given the record of a different held-out strategy, the reader tracks the strategy whose record it was given, and a wrong record leaves it further from the answer than no record at all (Table 2, Appendix G). Which episode the record holds matters more than any configuration in a search over the design of the reader (Appendices I and J).

Table 2: Components by regime, from one episode in the unseen condition, at 500 and 5000 training strategies and on 90 PPO agents outside the generator, with no memory tuning. Indented rows each remove one component from the reader without interactive imitation, and the last row gives PLAYREAD the record of another strategy, averaged over two such swaps in the hill game and one on the PPO agents (Table 6, Appendices G and M).

	500 strategies	5000 strategies	outside the generator
PLAYREAD	0.829	0.877	0.780
PLAYREAD without interactive imitation	0.780	0.859	0.779
without the history of the match in progress	0.785	0.830	0.345
without retrieval, mean memory value only	0.751	0.862	0.766
PLAYREAD given the record of another strategy	0.380	0.337	0.674

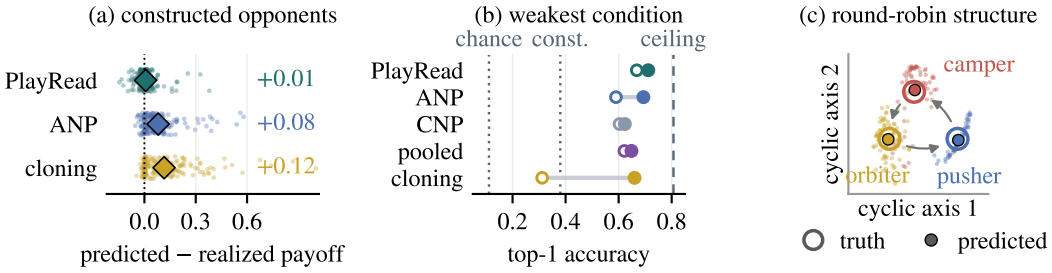

Figure 4: What play permits in the hill game. (a) Predicted minus realized payoff of opponents built against policies inferred from 18 episodes and evaluated in the unseen condition, mean as a diamond. (b) Top-1 accuracy in naming the weakest condition from one episode (hollow) and eighteen (filled). (c) The leading cyclic plane of the round robin among the 200 held-out strategies, both sides inferred from 18 episodes and played in the unseen condition. Rings are true archetype centers, filled markers inferred ones, dots inferred strategies and arrows the direction of the planted cycle (Appendix L).

The architecture against the procedure. Fitted as their papers specify, the readers of prior work lag PLAYREAD from one episode (Table 1). Retrained with our targets and interactive imitation, the ANP and CNP readers come within 0.02 of it from one episode, as does a context window in place of the memory (Table 7, Appendix I). A reader trained on recorded episodes alone learns a shortcut, since the match in progress identifies a strategy as well as the record does while the original is acting, and once the inferred policy plays only the record is left. Our reader without interactive imitation shows both effects, predicting actions better than cloning on the states the originals visit and worse on the states its own play reaches (Fig. 3d, Wen et al., 2020; Ross et al., 2011). Where the population is large the choice among readers therefore matters much less.

Which component matters, and where. The prior weakens when the population is small and when the evaluated agent lies outside the family it was drawn from, and Table 2 varies each. Retrieving the moments that resemble the current one, and retrieving the response rather than the situation, both matter with 500 training strategies and not with 5000 (Table 6). The history of the match in progress matters at 5000 strategies and most on the PPO agents, where removing it costs more than half of the accuracy, and there tuning the memory on the record also helps (Table 11). The record must provide what a weak prior cannot, and the match in progress what no prior covers. On those 90 agents PLAYREAD with its memory tuned still leads every reader given no outcomes, CFR poker strategies are read much less accurately, and both recover once agents of their kind enter the population (Appendices M and O).

4.4 COORDINATION, POKER AND LANGUAGE MODELS

Three standard domains have payoffs of different kinds, cooperative in Overcooked, zero-sum in Leduc poker and neither in the prisoner's dilemma (Fig. 5 and Table 3).

Coordination, where both sides share the payoff. In Overcooked (Carroll et al., 2019) some partners deliver more soups with any strategy, and a prior that predicts each partner by its average over all strategies already reaches 0.721. After removing the main effects of strategy and partner, what remains is how well a partner suits one strategy, and there PLAYREAD reaches 0.699 from one game against 0.474 for cloning. Asked to name a single teammate, PLAYREAD picks a partner 0.57 soups per game below the best available against 1.21 for cloning, and a published Prompt-DT matches

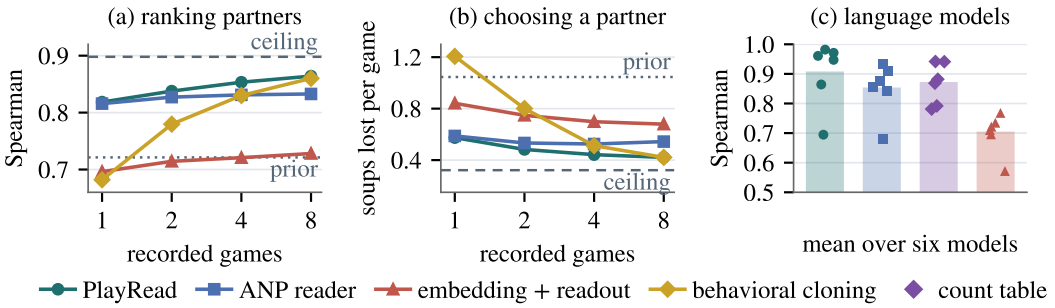

Figure 5: Beyond the hill game. (a, b) Overcooked, human proxies and PPO agents read from one to eight games, Spearman across 100 partners and soups lost by teaming with the partner predicted best, with the ceiling dashed and a prior that ignores the record dotted. (c) The prisoner's dilemma, each language model read from six probe matches, Spearman against 215 published strategies, bars the mean over six models and dots each model. The count table tallies moves after each recent history, the policy fitted to one strategy in this game (Appendix P).

Table 3: Leduc poker. Every method produces a table that is then played, so both quantities are exact. Payoff is the Spearman across the 100 pool opponents and exploitability the Spearman across the held-out strategies. *mlp* draws its 200 strategies like the population and *mixed* 50 from each family, including best responses. k counts recorded hands, bold the best per column (Table 14).

	payoff				exploitability			
	mlp		*mixed*		*mlp*		*mixed*	
Method	k=1	64	k=1	64	k=1	64	k=1	64
counting the recorded actions	0.865	0.889	0.760	0.808	0.415	0.837	0.300	0.662
projection on a population basis	0.866	0.969	**0.774**	0.960	0.490	0.931	**0.393**	0.843
PlayRead	**0.867**	**0.970**	0.765	**0.974**	**0.558**	**0.937**	0.330	**0.886**

it from one game. Eight games let cloning match PlayRead on the ranking, but not on how well a partner suits the strategy, which the population-trained reader draws from its prior (Appendix N).

Poker, where the truth needs no sampling. In Leduc hold'em (Lanctot et al., 2019; Wang et al., 2026) a strategy is a table over 936 situations, so its payoff and exploitability follow exactly, and every baseline also produces a table that is then played, so only the inference varies. The count baselines are the weakest, since 64 hands reach only about 75 situations. Fitting a low-dimensional basis learned from the training strategies to the recorded actions is the strongest baseline we found. PlayRead matches it on strategies drawn like the population and surpasses it on the mixed set in payoff from four hands and in exploitability at 64, and the basis is stronger there from a single hand (Table 3).

Language models, evaluated before they met. Six open language models had never played one another, so we read each from six matches, registered predictions for every pairing and then ran the matches. The predictions err by 0.17 points per round against 0.24 for a count table over recent moves, and the predicted exploiter of GLM-4-9B outscores it by 4.98 of a possible 5 points per round, as the live match confirmed. For ranking, each model plays the 221 published strategies of the Axelrod library (Knight et al., 2016) once, six of those matches form its record and the other 215 serve only as truth, so nothing a model did against the strategies it is ranked on is ever read. PlayRead outperforms the count table on held-out library strategies and on five of the six models. Llama-3.1-8B is the sole exception. It and Ministral-8B are the two models that do not reliably replay their records and whose predicted margins have the largest errors, so replaying the record flags in advance which predictions to trust (Appendix P).

4.5 What play permits

The second task asks which opponent of a family beats a strategy most. It is exact in Leduc, enumerated in the prisoner's dilemma and approximated in the hill game by searching the inferred policy (Section 3.4).

An opponent built against the inferred policy. The opponent returned by the cross-entropy search over each inferred policy is then played against the original. From 18 episodes in the unseen condition

it defeats all 200 held-out strategies with mean payoff $+0.84$, close to $+0.86$ for the same search run on the originals themselves (Fig. 6). A search on a wrong stand-in finds that error rather than a weakness of the strategy, so the overstatement measures the error. It is 0.007 for PLAYREAD, against 0.080 for an ANP reader and 0.115 for a clone (Fig. 4a).

The weakest condition of a strategy. Playing each inferred policy under all nine conditions names the condition where its mean payoff against the pool is lowest. From a single episode this is the true weakest condition for 0.67 of the held-out strategies, against 0.38 for always naming the condition most often weakest, chosen with the truth, and 0.11 for chance (Fig. 4b, Appendix K).

A round robin that never took place. Inferring a policy for each of the 200 held-out strategies and playing every pair fills a payoff matrix in which no original pair met, so no rating can be fitted. From 18 episodes PLAYREAD recovers it at 0.935 in the seen condition and 0.817 in the unseen one, against ceilings of 0.968 and 0.934, and in the seen condition keeps its transitive and cyclic parts in balance, which other readers that play shift toward the cyclic part (Fig. 4c, Appendix L, Balduzzi et al., 2019).

5 RELATED WORK

Evaluation from outcomes. Ratings (Elo, 1978; Bradley & Terry, 1952; Chen & Joachims, 2016; Balduzzi et al., 2018; Omidshafiei et al., 2019) and empirical game-theoretic analysis (Wellman, 2006; Lanctot et al., 2017) estimate payoffs from matches that were played, and learned outcome models predict the payoff between representations of two strategies (Grover et al., 2018; Wang et al., 2026). The first two need results for the strategy, and the last predicts without playing, so none can evaluate a condition absent from its data. Language models are likewise evaluated by playing repeated games against each other and against scripted strategies (Akata et al., 2025; Duan et al., 2024; Payne & Alloui-Cros, 2025). We infer behavior from a short record instead, and obtain the outcome of an unplayed match by playing the inferred policy.

Inferring agents. Opponent and agent models infer another agent, often in order to act against it (Rabinowitz et al., 2018; Papoudakis et al., 2021; Jing et al., 2024a;b; Rahmani et al., 2026), and in-context and retrieval-augmented policies infer behavior for the acting agent itself (Duan et al., 2017; Chen et al., 2021; Laskin et al., 2023; Xu et al., 2022; Humphreys et al., 2022; Schmied et al., 2025; Sridhar et al., 2025). Such models are judged by action prediction or by the return of the agent that uses them. PLAYREAD instead stands in for another strategy and is judged by whether it reproduces the payoffs of that strategy against new opponents and under new conditions.

Judging models by what they reproduce. Value-equivalent models are judged by the decisions they preserve rather than by likelihood alone (Farahmand et al., 2017; Grimm et al., 2020; 2021; Lambert et al., 2020; Dann et al., 2026), and adversarial imitation matches behavior on the states that interaction induces (Ho & Ermon, 2016; Song et al., 2018; Swamy et al., 2021; Tang et al., 2024). Our outcome distance applies this view to a model of another strategy, with opponents and conditions as the tests (De Nicola & Hennessy, 1984).

Searching against inferred agents. Adversarial policies are optimized against victims that can be run (Gleave et al., 2020; Wang et al., 2023) and against models of black-box agents (Dezfouli et al., 2020; Bui et al., 2023). We search against the inferred stand-in and verify the result on the original.

6 CONCLUSION

A strategy that cannot be run can still be evaluated from a short record, by inferring a runnable policy that stands in for the original, playing it in the simulator and judging it by the payoffs it reproduces rather than by the actions it predicts. The simulator supplies the rules, the opponent and the condition, so a new opponent or a new condition changes what is simulated rather than what has to be inferred. In a controlled game, playing the inferred policy is markedly more accurate than predicting the payoff from the same memory when the record is short or the condition is unseen, and the inferred policy also names the condition in which a strategy is weakest, yields an opponent that defeats it and recovers the payoff matrix among strategies that never met. The approach needs a simulator covering the conditions of interest, records in its observation space and a population covering the strategies evaluated. Outside that population the inferred policy falls back toward the population prior (Appendix M). All our experiments have two agents, and games with more players would need records and payoffs over the joint profiles of co-players.

Ethics Statement

Constructing an opponent from a few episodes of behavior could be used to exploit a deployed agent, such as a trading rule or a negotiating language model, without access to its weights. The same capability serves red-teaming, since a developer can search for the opponent that defeats an agent before deploying it and then verify that opponent by playing it. A constructed opponent is only as reliable as the inferred policy it was searched against (Section 2). Our experiments use simulated games, strategies that we generated or trained, the public Axelrod library, the public Overcooked human data, and open language models that we served ourselves, and no data were collected from people.

Reproducibility Statement

PlayRead is specified in Section 3 and Appendix A, with every hyperparameter of the hill-game reader. The environment, the population, the probes and the truth protocol are in Appendix D, the other domains in Appendices N to P, baselines and their tuning in Appendix E, and compute in Appendix F. Code, configuration files, seeds, the logs and summary files behind every number, and the files that redraw every figure are provided as supplementary material, with a README that maps each table and figure to its script and result files. Checkpoints and full prediction files exceed the size limit and will be released with the code.

AI Use Statement

The authors conceived and directed the research and made all scientific decisions, including the research question, the formulation, the method, the experiments and the claims. The authors used the Claude models of Anthropic (Claude Opus 4.8, Claude Opus 5, Claude Opus 5.5, Claude Fable 5 and Claude Fable 5.1) through the Claude Code agent as a coding and writing assistant, for implementation, running experiments, figure code, reference checks and editing of the text. The language models in the experiments are objects of study rather than assistants. The authors reviewed and verified all code, reported numbers and citations, and take full responsibility for the paper.

References

Elif Akata et al. Playing repeated games with large language models. *Nature Human Behaviour*, 2025.

David Balduzzi, Karl Tuyls, Julien Perolat, and Thore Graepel. Re-evaluating evaluation. In *NeurIPS*, 2018.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M Czarnecki, Julien Perolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *ICML*, 2019.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

The Viet Bui, Tien Mai, and Thanh H. Nguyen. Imitating opponent to win: Adversarial policy imitation learning in two-player competitive games. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2023.

Micah Carroll et al. On the utility of learning about humans for human-ai coordination. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, et al. Decision transformer: Reinforcement learning via sequence modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *WSDM*, 2016.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning (ICML)*, 2018.

Christoph Dann, Yishay Mansour, and Mehryar Mohri. Theoretical foundations and effective algorithms for policy-aware simulator learning. *arXiv preprint arXiv:2605.29032*, 2026.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, 2005.

Rocco De Nicola and Matthew Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34, 1984.

Amir Dezfouli, Richard Nock, and Peter Dayan. Adversarial vulnerabilities of human decision-making. *Proceedings of the National Academy of Sciences*, 2020.

Jinhao Duan et al. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations. *arXiv preprint arXiv:2402.12348*, 2024.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Arpad E Elo. *The Rating of Chessplayers, Past and Present*. Arco, 1978.

Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Marta Garnelo, Dan Rosenbaum, Chris J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and S M Ali Eslami. Conditional neural processes. In *ICML*, 2018.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Adam Gleave, Michael Dennis, Cody Wild, Neel Kant, Sergey Levine, and Stuart Russell. Adversarial policies: Attacking deep reinforcement learning. In *International Conference on Learning Representations*, 2020.

GLM Team. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*, 2024.

Aaron Grattafiori et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Christopher Grimm, André Barreto, Satinder Singh, and David Silver. The value equivalence principle for model-based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Christopher Grimm, André Barreto, Gregory Farquhar, David Silver, and Satinder Singh. Proper value equivalence. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Aditya Grover, Maruan Al-Shedivat, Jayesh K. Gupta, Yura Burda, and Harrison Edwards. Learning policy representations in multiagent systems. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1802–1811, 2018.

Eric A. Hansen, Daniel S. Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, pp. 709–715, 2004.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Peter C. Humphreys, Arthur Guez, Olivier Tieleman, Laurent Sifre, Théophane Weber, and Timothy Lillicrap. Large-scale retrieval for reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yuheng Jing, Kai Li, Bingyun Liu, et al. Towards offline opponent modeling with in-context learning. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yuheng Jing, Bingyun Liu, Kai Li, et al. Opponent modeling with in-context search. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024b.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *ICLR*, 2019.

Yoon Kim, Sam Wiseman, Andrew C. Miller, David Sontag, and Alexander M. Rush. Semi-amortized variational autoencoders. In *International Conference on Machine Learning (ICML)*, 2018.

Vincent Knight, Owen Campbell, Marc Harper, Karol Langner, James Campbell, Thomas Campbell, Alex Carney, Martin Chorley, Cameron Davidson-Pilon, Kristian Glass, et al. An open framework for the reproducible study of the iterated prisoner's dilemma. *Journal of Open Research Software*, 4(1):e35, 2016.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Learning for Dynamics and Control (L4DC)*, 2020.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2017.

Marc Lanctot, Edward Lockhart, Jean-Baptiste Lespiau, Vinicius Zambaldi, Satyaki Upadhyay, Julien Pérolat, Sriram Srinivasan, Finbarr Timbers, Karl Tuyls, Shayegan Omidshafiei, et al. OpenSpiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*, 2019.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, et al. In-context reinforcement learning with algorithm distillation. In *International Conference on Learning Representations (ICLR)*, 2023.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Vladimir Mikulik, Grégoire Delétang, Tom McGrath, Tim Genewein, Miljan Martic, Shane Legg, and Pedro A. Ortega. Meta-trained agents implement Bayes-optimal agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mistral AI. Un ministral, des ministraux: Introducing the world's best edge models. https://mistral.ai/news/ministraux, 2024.

Shayegan Omidshafiei, Christos Papadimitriou, Georgios Piliouras, Karl Tuyls, Mark Rowland, Jean-Baptiste Lespiau, Wojciech M Czarnecki, Marc Lanctot, Julien Perolat, and Rémi Munos. α-rank: Multi-agent evaluation by evolution. *Scientific Reports*, 9(1):9937, 2019.

Pedro A. Ortega, Jane X. Wang, Mark Rowland, Tim Genewein, Zeb Kurth-Nelson, Razvan Pascanu, Nicolas Heess, Joel Veness, Alex Pritzel, Pablo Sprechmann, Siddhant M. Jayakumar, Tom McGrath, Kevin Miller, Mohammad Azar, Ian Osband, Neil Rabinowitz, András György, Silvia Chiappa, Simon Osindero, Yee Whye Teh, Hado van Hasselt, Nando de Freitas, Matthew Botvinick, and Shane Legg. Meta-learning of sequential strategies. *arXiv preprint arXiv:1905.03030*, 2019.

Georgios Papoudakis, Filippos Christianos, and Stefano V. Albrecht. Agent modelling under partial observability for deep reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, 2021.

Kenneth Payne and Baptiste Alloui-Cros. Strategic intelligence in large language models: Evidence from evolutionary game theory. *arXiv preprint arXiv:2507.02618*, 2025.

Dean A. Pomerleau. ALVINN: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Neil C Rabinowitz, Frank Perbet, H Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International Conference on Machine Learning*, 2018.

Rahmani, Dziadzio, Strüber, Hernández-Gutiérrez, and Bethge. RevengeBench: Reverse engineering code-space policies from behavioral experiments. *arXiv preprint arXiv:2606.26094*, 2026.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Thomas Schmied, Fabian Paischer, Vihang Patil, Markus Hofmarcher, Razvan Pascanu, and Sepp Hochreiter. Retrieval-augmented decision transformer: External memory for in-context RL. In *Conference on Lifelong Learning Agents (CoLLAs)*, 2025.

Lloyd S. Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10): 1095–1100, 1953.

Jiaming Song, Hongyu Ren, Dorsa Sadeghi, and Stefano Ermon. Multi-agent generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Kaustubh Sridhar, Souradeep Dutta, Dinesh Jayaraman, and Insup Lee. REGENT: A retrieval-augmented generalist agent that can act in-context in new environments. In *International Conference on Learning Representations (ICLR)*, 2025.

Gokul Swamy, Sanjiban Choudhury, J. Andrew Bagnell, and Zhiwei Steven Wu. Of moments and matching: A game-theoretic framework for closing the imitation gap. In *International Conference on Machine Learning (ICML)*, 2021.

Jingwu Tang, Gokul Swamy, Fei Fang, and Zhiwei Steven Wu. Multi-agent imitation learning: Value is easy, regret is hard. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Zhenwei Tang, Difan Jiao, Eric Xue, Reid McIlroy-Young, Jon Kleinberg, Siddhartha Sen, and Ashton Anderson. Learning to imitate with less: Efficient individual behavior modeling in chess. *arXiv preprint arXiv:2507.21488*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Kevin Wang, Kevin Yang, Arjun Prakash, and Amy Greenwald. Towards learning representations of policies in two-player zero-sum imperfect-information games. *arXiv preprint arXiv:2607.01498*, 2026.

Tony T. Wang, Adam Gleave, Tom Tseng, Kellin Pelrine, Nora Belrose, Joseph Miller, Michael D. Dennis, Yawen Duan, Viktor Pogrebniak, Sergey Levine, and Stuart Russell. Adversarial policies beat superhuman go ais. In *International Conference on Machine Learning*, 2023.

Michael P Wellman. Methods for empirical game-theoretic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1552–1555, 2006.

Chuan Wen, Jierui Lin, Trevor Darrell, Dinesh Jayaraman, and Yang Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mengdi Xu, Yikang Shen, Shun Zhang, Yuchen Lu, Ding Zhao, Joshua B. Tenenbaum, and Chuang Gan. Prompting decision transformer for few-shot policy generalization. In *International Conference on Machine Learning (ICML)*, 2022.

A Outcome distance, architecture and training of PlayRead

The outcome distance of Section 2, $d_{\mathcal{J},\mathcal{E}}(\pi,\hat{\pi}) = \sup_{j\in\mathcal{J},\, e\in\mathcal{E}} |\, y(\pi,\pi_j;e) - y(\hat{\pi},\pi_j;e)\,|$, is the testing equivalence of concurrency theory (De Nicola & Hennessy, 1984) with opponents as tests, and proper value equivalence (Grimm et al., 2021) of the decision problem a fixed strategy induces for its opponent. It needs the original, so estimating it from sampled opponents gives a lower bound, which is why every constructed opponent is played against the original.

Writing c for a learned summary token, each encoder of Eq. (1) appends the condition to every token, embeds the tokens, prepends the summary, applies B transformer blocks, and returns the output at the summary position,

$$\begin{aligned} z_0 &= \big[\, c + \mathrm{pos}_0 \,;\, \mathrm{Emb}[x_{\tau-L};e] + \mathrm{pos}_1 \,;\, \ldots \,;\, \mathrm{Emb}[x_{\tau+H-1};e] + \mathrm{pos}_{L+H} \,\big], \\ z_\ell &= \mathrm{Block}_\ell(z_{\ell-1}), \qquad f(x_{\tau-L:\tau+H}, e) = \mathrm{Proj}\,\mathrm{LN}\big(z_B^{(0)}\big), \end{aligned} \tag{9}$$

where $\mathrm{pos}_0, \ldots, \mathrm{pos}_{L+H}$ are position embeddings, a learned segment embedding marks the steps before and after the anchor, LN is layer normalization, and Emb and Proj are linear. The key encoder applies the same form to $x_{\tau-L:\tau}$. The opponent search of Section 3.4 refines a random search by the cross-entropy method (de Boer et al., 2005), drawing in each round around each center $\bar{\psi}$ with per-parameter width η

$$\psi^{(1)}, \ldots, \psi^{(Q)} \sim \mathcal{N}\big(\bar{\psi}, \mathrm{diag}(\eta^2)\big), \quad \bar{\psi} \leftarrow \mathrm{mean}(\mathcal{S}), \quad \eta \leftarrow \max\big(\,\mathrm{std}(\mathcal{S}), \eta_{\min}\big), \tag{10}$$

where the samples are clipped to the unit box of the parameters of the family and the elite set $\mathcal{S}$ holds the m samples with the largest $\hat{F}^M$, the payoff estimated from M shared starts. In the continuous-control game $Q = 64$, $m = 8$, $M = 8$, the initial width is 0.15 and $\eta_{\min} = 0.02$.

Architecture (hill game). Tokens have 19 features (15 state, 2 own action, 2 opponent action). Memory anchors every $\sigma = 5$ steps with windows $L = 1$ and $H = 10$. Key and value encoders are one-layer transformers with a summary token (width 128, key dimension 64). The policy is a 4-layer causal transformer of width 128 with 4 heads. The retrieval is multi-head attention from the key of the last L steps of the match in progress over the memory. The head outputs a mean and a log standard deviation clamped to $[-4, 1]$. 1.31M parameters. A step-by-step player with a key–value cache reproduces the teacher-forced forward pass to 10^{-6} (unit test).

Training. AdamW, learning rate $3 \cdot 10^{-4}$, weight decay 0.01, one-cycle schedule with 5% warm-up, 300 epochs, 32 strategies per batch, 6 target episodes per strategy, record sizes $\mathcal{K} = \{1, 2, 3, 6, 12, 18\}$, gradient clipping at 1. Interactive imitation adds 2 matches per strategy per batch at 5000 strategies and 4 at 500, against random training opponents in a random seen condition, and the share of actions taken by the inferred policy rises linearly from 0 to 1 over the first half of training. The search uses 2048 candidates per strategy (half drawn like the population, half uniform over the parameter ranges), each candidate scored from 4 starts, four rounds of cross-entropy refinement around the best four, and the four final centers with the four best candidates re-scored from 32 starts. The play-time choices, a sampled action and a moment every five steps, were checked on the validation strategies. Training on the outcome distance itself, through opponents searched with gradients through the simulator, helped by 0.023 from scratch at 500 strategies (unseen condition, 18 episodes) but added nothing to interactive imitation, within 0.005 at 26 times the cost per epoch.

Tuning the memory on the record. Adam on the memory values only, with the steps, learning rate and spacing of moments chosen on validation strategies per record budget within a tie band of 0.002, the gentler setting among ties. Hill game, 30 steps with a hidden block of 20 steps at one episode and 300 steps at 6 and 18, learning rate 0.01 and 0.003. PPO agents, 100 steps with a hidden block of 20 and a moment every step at one episode, 1000 steps and a moment every step at 6, 3000 steps at 18, learning rate 0.01. Overcooked, no tuning at one game, 50 steps at 2 and 4 games and 300 at 8, learning rates 0.01 and 0.003. Poker, 300 steps at 64 probe hands only, learning rate 0.001. Language models, 100 steps, learning rate 0.01, chosen on 120 fresh synthetic strategies.

B What the training targets must contain

The default pairing plays every target episode against a population opponent the record does not contain, under an independently drawn training condition, so the condition of a target lies outside the record for 83 percent of one-episode records. Two variants change the pairing. The first draws each

Algorithm 1 Training PLAYREAD

for each batch of training strategies $\{\pi_i\}$ **do**
  draw $k \sim \mathcal{K}$ and records $\mathcal{D}_i$, build the memories, and set $\hat{\pi}_i \leftarrow R_\theta(\mathcal{D}_i)$
  $\mathcal{L}_{\text{bc}}$ on target episodes against opponents outside the record, in independently drawn conditions
  play matches mixing $\hat{\pi}_i$ and π_i, and compute $\mathcal{L}_{\text{ii}}$ on the actions of π_i in the reached states
  update θ on $\mathcal{L}_{\text{bc}} + \mathcal{L}_{\text{ii}}$
end for

Algorithm 2 Evaluating one held-out strategy with PLAYREAD

receive the record $\mathcal{D}^\star$ of $\pi^\star$, its matches against the probe panel
build the memory from $\mathcal{D}^\star$ and set $\hat{\pi} \leftarrow R_\theta(\mathcal{D}^\star)$, with θ fixed
optionally tune the memory values on a held-out part of $\mathcal{D}^\star$, with the keys and θ fixed
for each opponent π_j in the pool and each condition e **do**
  play M matches of $\hat{\pi}$ against π_j under e and average the scores into $\hat{y}_j^M(e)$
end for
report the rank correlation of $\hat{y}^M(e)$ against the true payoffs of $\pi^\star$
where the game is zero-sum, search the family for the opponent that maximizes $y(\pi_\psi, \hat{\pi}; e)$ and play it against $\pi^\star$

target from a condition the record contains and restricts interactive imitation to those conditions, the second replaces the targets by replays of the recorded matches from new starting states. With the flag off the modified code reproduces the stored runs bit for bit.

Tying the conditions of the targets to those in the record never helps. Paired against the default over the 200 test strategies with the plain objective at 500 strategies (five seeds), it loses 0.006 to 0.013 in the seen condition and 0.003 to 0.004 in the unseen one, and with interactive imitation the difference lies between -0.003 and $+0.003$ in every cell (three seeds at 500, one at 5000). Robustness to an unseen condition therefore comes from leaving the condition to the simulator, not from crossing conditions in the targets. What the targets must contain is other opponents, since training on replays of the recorded matches costs 0.097 from one episode and 0.051 from eighteen in the seen condition (plain objective, 500 strategies), although that variant also changes the number of target episodes, so its size is indicative.

C THE FOUR DOMAINS

Table 4: Domain summary and evaluation protocol. The objects of Section 2 in the four domains, with the number of conditions in parentheses and the observation and action spaces left to the appendix of each domain (Appendices D and N to P).

	hill game	Leduc poker	Overcooked	prisoner's dilemma
condition e	damping and wind (9)	fixed rules (1)	the layout (4)	the payoff matrix (1)
record	1 to 18 episodes of 100 steps	1 to 64 hands	1 to 8 games of 400 steps	6 matches of 200 rounds
payoff y	time closer to the hill	chips per hand	soups per game	points per round
strategies	scripted, around a cycle	networks, best responses	human proxies, PPO agents	Axelrod library, LLMs

D HILL GAME AND POPULATION

Dynamics. Two point agents start on opposite sides of the hill at a radius drawn uniformly from $[0.6, 0.9]$. Each observes 15 features, namely its own position and velocity, those of the opponent, the vector to the hill, both distances, the elapsed fraction of the episode, and the condition. Actions are 2-D forces clipped to $[-1, 1]$. Velocities update as $\text{vel} \leftarrow \text{vel}\,(1 - \text{damping}) + (a + \text{contact} + \text{wall} + (\text{wind}, 0))\,\Delta t$ with $\Delta t = 0.1$ and speed capped at 1, over $T = 100$ steps. Each step scores $+1$ for the agent closer to the hill and -1 for the other. The payoff is the mean. The conditions are damping $\in \{0.15, 0.25, 0.40\}$ and wind $\in \{-0.1, 0, 0.1\}$. Held out are $(0.40, 0.10)$, $(0.15, -0.10)$ and $(0.40, -0.10)$. We report the standard condition $(0.25, 0)$ as "seen" and $(0.40, 0.10)$ as "unseen".

Population. A strategy applies the force $w_{\text{goal}}(\text{goal} - \text{position}) - w_{\text{brake}}\,\text{vel} + w_{\text{ram}}\,\text{threat} \cdot \text{ram} + w_{\text{dodge}}\,\text{near} \cdot \text{closing} \cdot \text{dodge} + \text{noise}$, where the goal is the hill or a point on an orbit, ramming switches on when the opponent is nearer the hill or, for pushers, within range, and dodging when it closes in. Parameters are drawn around three archetypes that beat each other in a cycle. The families, in proportions 50/14/21/14%, are static, condition-dependent (w_{goal} scales with damping), adaptive (a running estimate of how aggressive the opponent is switches to orbit and dodge, and half also ram once rammed) and stochastic (action noise 0.15–0.4). 700 strategies split 500/200 by family, larger populations are independent draws with the same proportions, and the evaluation pool is 100 training strategies.

Records and truth. Six fixed probes (camper, pusher, orbiter, rammer, idler, wanderer) each play three episodes. The three episodes of a probe are played in three consecutive seen conditions, and the first of them shifts by one from each probe to the next, so each probe plays in three conditions and each condition hosts three probes. A budget of $k \leq 6$ episodes takes one from each of k probes and 18 takes all. Targets are 8 episodes per strategy against random training opponents in independently drawn conditions, and true payoffs average 32 random starts, whose two halves agree at a Spearman correlation of 0.970 in the seen condition and 0.936 in the unseen one. The memory of a strategy is built once from its record, the same inferred policy then plays every pool opponent under every condition, and nothing is refit per opponent or per match. A 2048-start reference over the same grid, drawn in four independent quarters, agrees with itself at 0.9985 in the seen condition and 0.9961 in the unseen one. Against that reference the 32-start truth reaches 0.988 and 0.974, which caps any method that predicts a payoff without playing, while the originals played through the $M = 8$ estimator reach 0.966 and 0.925 against the 32-start truth, the ceiling printed in Table 1. The methods of Table 1 that predict without playing are deterministic at scoring time, so the first of the two ceilings is theirs.

E Baselines

Behavioral cloning. A 2-layer MLP (width 64) from state to action per test strategy, full-batch AdamW, learning rate $3 \cdot 10^{-3}$. Steps in $\{1000, 2000, 3000\}$ and weight decay in $\{0, 0.1\}$ selected per record budget by matchup Spearman on 40 validation training strategies in two seen conditions. Selecting by held-out action error picked settings that ranked opponents worse, and a grid capped at 300 or 1000 steps selected its upper edge.

CNP and ANP. CNP uses a mean-pooled MLP encoder of (o, a) pairs and an MLP decoder of the pooled summary and the current observation, width 512 with 3 hidden layers each. ANP follows the deterministic path of Kim et al. (2019), with two self-attention layers over up to 1024 recorded (o, a) points and cross-attention from the current state, width 256 and 8 heads. Both have 1.59M parameters, are trained with the same record sizes on the same targets for 200 epochs, and play their mean action.

Outcome-based. The embedding with a readout uses a 2-layer transformer over each episode trained contrastively (two halves of one record embed close), mean over episodes, and an antisymmetric MLP readout of pairs trained on payoffs of every training strategy against 500 training strategies in the seen condition (2 starts each), 5000 steps. Its predictions are reused in the unseen condition, for which it has no labels. Blade–Chest is fitted by least squares on the payoffs of the 5000 training strategies against 500 of them plus a random 40% of the payoffs of each test strategy against the pool. A fifth of the observed cells selects among 24 configurations of steps, learning rate, weight decay and rank, without truth, and the chosen one is refitted on all observed cells before scoring. Elo is the same fit with rank 0. Its selection error agrees to four decimals across its whole sweep, so its accuracy reflects a scalar rating in a cyclic population rather than a failed optimization.

F Compute

Every run used one GPU, on a shared machine with eight RTX 4080 GPUs of 16 GB (a few baseline and design-search runs used a cloud RTX 4090), so the times in Table 5 overstate dedicated compute. Interactive imitation multiplies training time by about four to nine. The twelve readers trained with interactive imitation, four methods over three seeds, took 55.1 GPU-hours. Every played prediction in the hill game averages 8 simulated matches.

Table 5: Wall-clock cost on one shared RTX 4080 unless stated. II is interactive imitation. Training is per seed at 5000 strategies, mean over three seeds. Scoring is 200 held-out strategies against the 100-strategy pool in one condition, 160000 matches.

stage	method	cost
training, with / without II	PlayRead	8.73 h / 2.14 h
	ANP reader	3.54 h / 0.94 h
	CNP reader	3.00 h / 0.65 h
	pooled embedding	3.12 h / 0.34 h
reading 200 records into policies	PlayRead, ANP	0.1 to 0.5 s
scoring one condition, $k = 1$ to 18	PlayRead (idle GPU)	54 to 93 s
	ANP, CNP, pooled embedding	22 to 51 s
per-strategy fitting	behavioral cloning	1.7 to 2.6 s each
outcome baselines	embedding and readout	278 s, plus 179 s for 5.0M labeling matches
	Blade–Chest	52 s per fit, 1555 s to select
opponent search, per strategy	PlayRead	3.3 to 8.5 s
other domains	Leduc reader	25 to 26 min
	prisoner's dilemma reader, per fold	3.3 to 5 min
truth, 200 × 100 pairs, 32 starts	original strategies	10 to 11 s per condition

Table 6: Ablations in the unseen condition, Spearman from 1 and from 18 episodes over three training seeds (five for the reader without interactive imitation at 500 strategies). The indented rows remove one component from that reader.

Variant	500 strategies	5000 strategies
PlayRead	0.829 / 0.865	0.877 / 0.898
PlayRead without interactive imitation	0.780 / 0.815	0.859 / 0.888
without the history of the match in progress	0.785 / 0.843	0.830 / 0.880
without retrieval, mean memory value only	0.751 / 0.773	0.862 / 0.881
values from the past window only	0.749 / 0.783	0.864 / 0.889
half the width and depth	0.716 / 0.758	0.846 / 0.875

G Additional analyses of the hill game

Read with a record of zeros, PlayRead at 5000 strategies reaches 0.482 in the unseen condition from one episode, against 0.337 when it is given the record of another strategy.

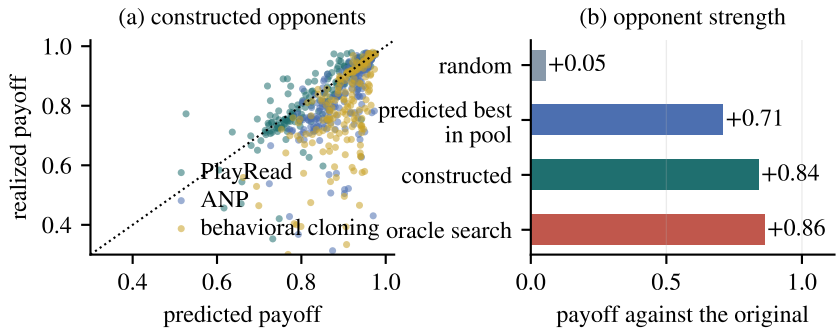

Figure 6: Opponents constructed against an inferred policy and then played against the original strategy, in the unseen condition from 18 episodes. (a) The payoff predicted against the inferred policy and the payoff realized against the original, one point per strategy, for PlayRead, the ANP reader and cloning. (b) The strength of the constructed opponent beside a random opponent, the pool opponent the inferred policy predicts to be strongest, and a search run directly on the original.

The planted cycle. Averaging the predicted payoffs by archetype gives a small matrix of how one kind of strategy fares against another. From 18 episodes it matches the truth to within 0.02 in mean absolute error with every sign correct in both conditions, against 0.04 and 0.063 for the ANP reader, which gets one sign wrong in the unseen condition, and 0.05 and 0.13 for cloning. From one episode PlayRead is within 0.05 and 0.06 with one sign wrong in each condition, against 0.08 and 0.09 for the ANP reader and 0.52 and 0.48 for cloning. The unseen condition changes the structure, since orbiting strategies beat pushing ones at standard damping by $+0.316$ and lose to them under heavy damping and wind by -0.149, where the inferred policies, which never saw that condition, place the

same entry at -0.152. The constructed opponents also describe the weaknesses in the same terms as an oracle search, ramming against pushers with weight 2.22 against 2.50 and barely ramming against campers.

Table 7: Spearman correlation for every reader, training population and budget in the continuous-control game. The seeds column gives the number of training seeds, and II is interactive imitation. Behavioral cloning fits each strategy on its own, so it has no population.

			seen condition			unseen condition		
reader	strategies	seeds	k=1	6	18	k=1	6	18
PlayRead	500	3	0.867	0.913	0.917	0.829	0.862	0.865
	5000	3	0.903	0.949	0.951	0.877	0.896	0.898
PlayRead, no II	500	5	0.816	0.872	0.873	0.780	0.813	0.815
	1500	1	0.867	0.916	0.919	0.837	0.864	0.865
	5000	3	0.890	0.939	0.941	0.859	0.886	0.888
	15000	1	0.905	0.949	0.951	0.869	0.894	0.896
PlayRead, memory tuned	5000	3	0.904	0.951	0.953	0.882	0.898	0.900
context-window reader, our recipe	500	3	0.865	0.911	0.912	0.817	0.858	0.866
	5000	3	0.908	0.949	0.949	0.878	0.899	0.903
memory and context in one reader, our recipe	5000	3	0.907	0.948	0.951	0.875	0.898	0.899
Prompt-DT, as published	500	3	0.762	0.819	0.813	0.737	0.769	0.754
	5000	3	0.855	0.892	0.888	0.834	0.860	0.852
ANP	500	3	0.832	0.879	0.888	0.790	0.835	0.845
	5000	3	0.884	0.928	0.933	0.858	0.894	0.896
ANP, no II	500	3	0.845	0.881	0.889	0.803	0.837	0.846
	1500	1	0.874	0.909	0.915	0.840	0.883	0.886
	5000	3	0.875	0.918	0.924	0.812	0.868	0.875
	15000	1	0.888	0.934	0.941	0.857	0.890	0.893
CNP	500	1	0.801	0.817	0.818	0.777	0.785	0.786
	5000	3	0.884	0.918	0.920	0.868	0.888	0.889
CNP, no II	500	3	0.689	0.705	0.704	0.620	0.624	0.625
	1500	2	0.834	0.868	0.871	0.834	0.856	0.860
	5000	3	0.870	0.914	0.918	0.849	0.879	0.881
	15000	1	0.887	0.930	0.931	0.845	0.878	0.881
pooled embedding	500	3	0.624	0.639	0.646	0.578	0.585	0.588
	5000	3	0.875	0.901	0.901	0.861	0.879	0.884
pooled embedding, no II	500	1	0.739	0.737	0.732	0.646	0.644	0.645
	5000	3	0.860	0.884	0.889	0.852	0.878	0.883
pretrained, fine-tuned	500	1	0.451	0.815	0.871	0.480	0.748	0.823
	5000	3	0.426	0.782	0.873	0.464	0.719	0.796
embedding + readout	500	1	0.783	0.917	0.931	0.671	0.786	0.800
	5000	3	0.832	0.942	0.952	0.729	0.801	0.805
behavioral cloning	—	3	0.484	0.826	0.906	0.531	0.750	0.845

All readers, populations and budgets. Table 7 lists every reader configuration behind Table 1 and Fig. 3. Most of the gain from a longer record arrives by the sixth episode, and at 5000 strategies PlayRead is the most accurate neural-process reader at every budget in both conditions, where interactive imitation improves every reader. At 500 strategies it improves PlayRead and the CNP reader, the latter from 0.625 to 0.786 from 18 episodes (one seed with interactive imitation), while the ANP reader is unchanged from 18 episodes and the pooled embedding loses 0.06.

A second population. Retrained without interactive imitation on a second population of 5000 strategies drawn with a different seed and scored against its own truth, PlayRead reaches 0.855 and 0.883 in the unseen condition from 1 and 18 episodes against 0.859 and 0.888 on the first, the ANP reader 0.844 and 0.871 and the CNP reader 0.801 and 0.841 (one training seed each), so PlayRead stays the most accurate reader on a second sampled population.

H Diagnosis of closed-loop error

We measured the compounding error that interactive imitation targets (Ross et al., 2011) directly, for the reader of Table 7 trained on 500 strategies without interactive imitation, over three training seeds, with 18-episode records in the unseen condition (Fig. 7). On the states the strategies themselves visit, the reader predicts actions better than per-strategy behavioral cloning, 0.072 to 0.094 against 0.258 in squared error. On the states the inferred policies reach when they drive the match, the same reader is 3.3 to 4.4 times worse, while cloning grows by 6 percent.

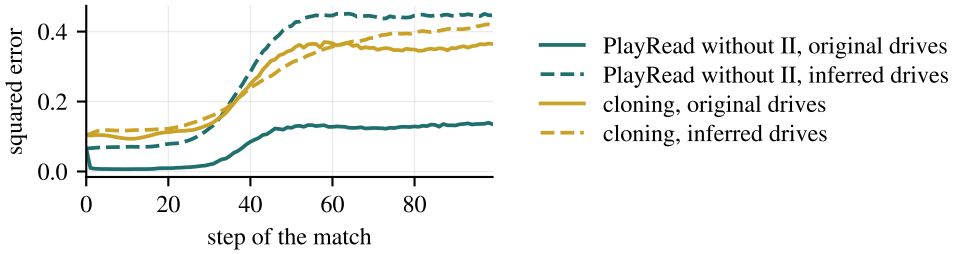

Figure 7: Squared error between the action of an inferred policy and the noise-free action of the strategy it was read from, along the match in the unseen condition, for readers trained on 500 strategies without interactive imitation (mean of three seeds) and for cloning, when the original drives the match (solid) and when the inferred policy does (dashed). Figure 3d averages these curves over the match.

The first ten steps separate the two failure modes. When the original drives, the error of a reader there is 0.012 to 0.014, an eighth of what cloning makes, because the history of the match in progress still holds the recent actions of the original. When the inferred policy drives, the same steps cost it 0.060 to 0.074, five times more, against 0.115 for cloning. Its advantage on the states of the original therefore rests partly on the actions it is fed, the copycat problem of Wen et al. (2020), and interactive imitation trains it on the histories it will actually produce.

I A reader with the record in its context

One may ask why PlayRead keeps the record in a retrieval memory rather than a context window. A context-window reader tests this. Its record enters one causal transformer as a prompt of at most 1024 steps and the match in progress attends to it, in the style of Prompt-DT (Xu et al., 2022), with no moments and no retrieval. Prompt-DT as published, with or without return tokens as validation chooses, is a baseline of Tables 1 and 11. The same architecture trained by our loop (1.21M parameters in six layers against 1.31M in four) is the recipe-equalized row of Table 7. It matches PlayRead on the full population and falls behind from one episode, where the evidence is thin, by 0.012 at 500 strategies in the unseen condition and 0.032 on agents outside the generator in the seen condition. With its prompt keys and values tuned as the memory is, on a setting chosen on validation, it leads on the PPO agents in the seen condition, 0.923 against 0.912 from 18 episodes, and trails in the unseen one, 0.869 against 0.873. A reader carrying both a memory and the raw record ties PlayRead (Table 7).

The two also differ in what they carry. At 18 episodes the memory holds 66k floats per strategy against 1548k for the prompt. With records of 72 episodes, four times the longest seen in training, both keep their accuracy in the unseen condition, 0.899 and 0.900, while the memory grows to 263k floats and the prompt keeps 1008 recorded steps, 56 percent of an 18-episode record and 14 percent of a 72-episode one.

A search over the design of PlayRead. Nine configurations, each trained for 60 epochs with one seed at width 144, were compared on 201 validation strategies drawn independently of the test strategies, with the criterion declared before any run, the mean Spearman over $k \in \{1, 18\}$ and the two conditions. They span 0.8802 to 0.8909, and no configuration beats the reader of the paper, at 0.8897, by more than the declared tie band of 0.002. A moment at every step is the only one above it, at 0.8909. Longer key windows and factorized keys are worse, 0.8802 to 0.8877, and depth alone does not help.

J Which episode the record holds

Every record in the hill game comes from a fixed panel of six probe opponents, and a one-episode budget takes one at random. Scoring each choice separately, with PlayRead of Table 1 without memory tuning and the first repeat of each probe, shows what that draw costs. The random draw the protocol makes gives 0.910 in the seen condition and 0.876 in the unseen one. The best single probe given to every strategy gives 0.941 and 0.894, and the best probe per strategy 0.944 and 0.909. The probe is chosen on one training seed and scored on another, over all six ordered pairs, so the choice is not made on the noise it is scored with. Choosing with the truth is not a method.

No probe serves every strategy. In the seen condition the rammer is best for 32 percent of the held-out strategies and the wanderer for 11, and averaged over strategies the probes range from 0.941 to 0.859. A single fixed probe therefore recovers most of the gain in the seen condition and about half in the unseen one, and the rest needs a choice per strategy. Three rules that pick a second episode from what the first one reveals, rollout spread, seed disagreement and memory coverage, took at most 13 percent of the 0.017 headroom of that choice in the seen condition and none of it in the unseen one.

K The three unseen conditions

The three are corners of the 3×3 grid, and each lies outside the convex hull of the six training conditions in the damping–wind plane, since the training conditions at damping 0.15 and 0.40 have no negative wind and the one at 0.40 no wind at all. A reader that reaches them therefore extrapolates rather than interpolates between settings it has seen. Reversing the wind gives the same game up to a half-turn of the arena, since walls, contacts, the hill and the start distribution are unchanged and every scripted rule is odd in positions and velocities and never reads the wind. Condition e0 therefore repeats the seen condition e2 in mirror image and e6 repeats e8, with per-strategy rank correlations between their truths of 0.975 and 0.957, close to the noise of the 32-start truth against 0.816 between e4 and e8. The main text reports e8, and Table 8 gives every method in all three conditions.

Table 8: Spearman correlation in the three unseen conditions, mean over three training seeds. The readers are trained with our procedure, and PlayRead is shown without memory tuning, so its e8 columns match the untuned row of Table 1. Starred methods return one prediction for every condition, scored against the truth of each condition.

	1 episode			18 episodes		
Method	e0	e6	e8	e0	e6	e8
PlayRead	**0.899**	**0.876**	**0.877**	**0.935**	**0.898**	**0.898**
ANP reader	0.883	0.855	0.858	0.914	0.893	0.896
CNP reader	0.877	0.865	0.868	0.899	0.888	0.889
pooled in-context embedding	0.874	0.858	0.861	0.889	0.881	0.884
behavioral cloning, per strategy	0.484	0.513	0.531	0.843	0.849	0.845
pretrained policy, fine-tuned	0.434	0.452	0.464	0.778	0.789	0.796
embedding with a pairwise readout*	0.765	0.729	0.729	0.857	0.805	0.805
Blade–Chest with 40% of own results*	0.846	0.801	0.801	0.846	0.801	0.801
Elo with 40% of own results*	0.475	0.532	0.533	0.475	0.532	0.533
ceiling of the estimator	0.955	0.925	0.925	0.955	0.925	0.925

The strongest and weakest conditions. For each held-out strategy the mean payoff against the pool is predicted under each of the nine conditions by playing its inferred policy, and compared with the same quantity for the original. A one-episode record covers one condition, so eight of the nine are new to it. Table 9 reports the profile correlation, the top-1 rate and the regret. Flipping

Table 9: Naming the weakest condition of a strategy by playing its inferred policy under all nine. Metrics as in Appendix K, chance top-1 0.111, constant the condition most often weakest.

	PlayRead		ANP reader		CNP reader		pooled embedding		cloning		constant	ceiling
episodes	1	18	1	18	1	18	1	18	1	18		
profile	**0.758**	**0.839**	0.704	0.790	0.682	0.755	0.690	0.746	0.402	0.748	—	0.956
top-1	**0.668**	**0.712**	0.590	0.693	0.603	0.623	0.622	0.648	0.312	0.660	0.380	0.807
regret	**0.031**	0.025	0.044	0.028	0.045	0.034	0.040	0.034	0.111	**0.022**	0.203	0.001

the wind is an exact symmetry of the game, so the six conditions with wind pair into mirror twins whose truths differ only by noise, and counting a twin as correct raises top-1 from 0.668 to 0.748 for PLAYREAD from one episode and from 0.312 to 0.368 for cloning, against a chance rate of 0.141. On the strongest condition rather than the weakest, PLAYREAD reaches 0.517 from one episode and 0.685 from eighteen, against 0.392 and 0.568 for the ANP reader and 0.135 and 0.402 for cloning, with regrets for PLAYREAD of 0.038 and 0.016, against a ceiling top-1 of 0.795 with regret 0.001.

A prediction that cannot vary with the condition costs 0.220 in regret, and always naming the condition most often weakest, chosen with the truth, costs 0.203. From 18 episodes cloning finds the weakest condition as well as PLAYREAD, 0.022 against 0.025, while PLAYREAD still ranks the conditions better, names the strongest one more often, 0.685 against 0.402, and finds the weakest pair of opponent and condition among 900 with less than half the regret, 0.077 against 0.168.

L A ROUND ROBIN AMONG HELD-OUT STRATEGIES

Each of the 200 held-out strategies is read from its own record and every pair of inferred policies is played with the estimator of the main experiment. The truth is the 200×200 matrix of payoffs among the originals over 32 starts, antisymmetric within Monte Carlo noise. No strategy in it ever played another, so ratings cannot be fitted. Table 10 reports the Spearman correlation over the 199 opponents of each strategy. The one-side column plays the inferred policy against the original opponents, isolating the cost of reading the opponent as well.

Table 10: Round robin among 200 held-out strategies, per-row Spearman. Readers average three seeds. The one-side column plays the inferred row strategy against the true column strategies (seed 0).

	seen condition			unseen condition		
Method	k=1	k=18	one side, 18	k=1	k=18	one side, 18
embedding with a pairwise readout	0.646	**0.942**	—	0.543	0.731	—
behavioral cloning, per strategy	0.191	0.835	0.906	0.233	0.738	0.849
CNP reader	0.733	0.863	0.919	0.681	0.786	0.893
ANP reader	0.733	0.890	0.937	0.652	**0.818**	0.904
PLAYREAD	**0.794**	0.935	0.952	**0.723**	0.817	0.908
ceiling of the estimator	0.968	0.968		0.934	0.934	

From a full record in the seen condition the round robin is read at 97 percent of its ceiling and the structure of the game survives. The two ceilings of Appendix D hold here as well, 0.990 and 0.979 for a method that does not play against 0.968 and 0.934 for one that does. From 18 episodes the outcome-trained embedding leads PLAYREAD by 0.007 in the seen condition in raw correlation and trails it by 0.014 as a fraction of its own ceiling. Decomposing the antisymmetrized matrices into a transitive part and a cyclic remainder (Balduzzi et al., 2019), from 18 episodes in the seen condition PLAYREAD preserves their balance, a transitive share of 0.453 against 0.452 in the truth, where the other readers make the game look more cyclic than it is, 0.344 to 0.402, and its ratings and leading cyclic plane are the closest to the truth among the readers that play, correlating at 0.982 and 0.991 against 0.895 to 0.942 and 0.951 to 0.966. That plane carries 48 percent of the energy of the matrix in the seen condition and 30 percent in the unseen one, and contains the planted cycle, which every population-trained reader recovers with the correct orientation. Figure 4c draws the plane of the unseen condition. Each center there averages the 62 to 75 strategies of one archetype, and the inferred plane is rotated onto the true one without reflection, since the orientation of the plane carries no information and a reflection would reverse the cycle. The unseen condition shows why playing matters. There the payoff of orbiting against pushing strategies turns negative, -0.134, and PLAYREAD predicts -0.123, whereas the outcome-trained embedding, which can only reuse its seen-condition prediction, gives $+0.288$.

Reading the opponent as well is not free. Relative to one-side play, which uses seed 0, the per-row Spearman of PLAYREAD at that seed falls from 0.952 to 0.935 in the seen condition and from 0.908 to 0.814 in the unseen one, because the errors of two inferred policies add. The same holds for every reader, and the margin of PLAYREAD over the CNP and ANP readers widens to 0.06 from one episode in the seen condition.

M AGENTS TRAINED BY REINFORCEMENT LEARNING

Every held-out strategy of the main experiment comes from the scripted generator of the training population. To test reading outside that generator, 90 agents were trained with PPO in the hill game and read with the readers of Table 1, none of which was retrained. Each agent is a small MLP with a Gaussian policy over the game state, for 53 of them extended with the last action of the opponent, trained against single scripted strategies, mixtures of 4, 16 or 64, or snapshots of itself, for 15 to 300 iterations and only in seen conditions. The 917 scripted training opponents include no pool opponent, probe or test strategy, and every agent beats a stationary and a random policy by at least 0.34. Records, truth and scoring follow the main protocol, and every method receives identical record tensors.

Table 11: Spearman on 90 agents trained by reinforcement learning, mean over three training seeds. The baselines are trained as their papers describe, and the same readers trained with our recipe give the values in parentheses in Table 12. The ceiling plays the true agents through the same estimator.

	1 episode		18 episodes	
Method	seen	unseen	seen	unseen
Blade–Chest with 40% of its own results	0.893	0.746	0.893	0.746
embedding with a pairwise readout	0.598	0.553	0.695	0.635
behavioral cloning, per strategy	0.654	0.619	0.856	0.780
pooled in-context embedding, as published	0.690	0.740	0.710	0.781
CNP reader, as published	0.722	0.740	0.742	0.766
ANP reader, as published	0.745	0.785	0.793	0.802
Prompt-DT, as published	0.613	0.686	0.745	0.771
PLAYREAD without memory tuning	0.809	0.780	0.811	0.784
PLAYREAD	**0.838**	**0.824**	**0.912**	**0.873**
ceiling of the estimator	0.933	0.885	0.933	0.885

Table 11 shows a partial transfer that tuning the memory repairs. The untuned PLAYREAD is the most accurate reader from one episode in the seen condition, 0.809 against 0.745 for the ANP reader and 0.654 for cloning, but it does not grow with the record, $+0.002$ with interval $[-0.012, 0.015]$ from 1 to 18 episodes, so cloning overtakes it at 18. Tuning the memory on the record lifts it to 0.838 and 0.912, ahead in every cell of every baseline that sees no results of the agent, and from 18 episodes within 0.021 of the ceiling. The outcome-trained methods fall again under the condition shift, Blade–Chest from 0.893 to 0.746 although it sees 40 percent of the results of each agent.

The diagnosis points to the population. Every agent lies farther from the training population, in the embedding, than the 95th percentile of the scripted test strategies, the error of the inferred mean action correlates at -0.77 with per-agent accuracy in the seen condition, and in the unseen condition the losses concentrate on agents trained against one opponent or a mixture of four, read at 0.69 to 0.82 against 0.86 to 0.89 for agents trained against mixtures of 16 or 64.

Covering the agents in the population. A second set of 300 PPO agents was trained with the same recipes, new seeds and new opponent draws, none sharing a recipe, opponent set and seed with a test agent. Of these 270 join the training population and 30 fix the number of fine-tuning epochs, so the 90 test agents are used for no choice. Every reader is fine-tuned from its own checkpoint with identical data and schedule, the outcome methods receive the labeled matches of the new agents, and a control fine-tunes PLAYREAD on scripted strategies alone.

Coverage closes most of the gap (Table 12). Without memory tuning, PLAYREAD gains 0.060 from one episode in the seen condition, interval $[0.047, 0.073]$, and 0.049 from 18 episodes in the unseen one, reaching 93 to 95 percent of the ceiling, while the control gains nothing, between -0.009 and $+0.003$, so the improvement comes from the added agents rather than from further training. Coverage also narrows the lead of the untuned reader, since a population that covers the agents gives every reader the prior that set PLAYREAD apart. From one episode it stays ahead of the neural-process readers in the seen condition, by 0.063 over the ANP reader and 0.045 over the CNP reader, and is level in the unseen one. From 18 episodes it trails every fine-tuned reader in the unseen condition, by up to 0.020, and the context-window reader with our recipe leads the untuned readers, 0.904 and 0.854. Tuning the memory on the record, with the settings chosen before expansion, restores the lead in every cell except the seen condition from one episode, where Blade–Chest, which sees 40 percent of the results of each agent, stays ahead, 0.899 against 0.890. From 18 episodes in the unseen

Table 12: Spearman on the 90 agents of Table 11 after the population is expanded with 270 other agents trained by reinforcement learning, with the value before expansion in parentheses. The neural-process and pooled readers are trained with our recipe. The memory of PLAYREAD is tuned with the settings of Table 11, chosen before expansion, and scored with the evaluation code of Table 11, which reproduces the untuned row to within 0.003.

	1 episode		18 episodes	
Method	seen	unseen	seen	unseen
Blade–Chest with 40% of its own results	**0.899** (0.893)	0.744 (0.746)	0.899 (0.893)	0.744 (0.746)
embedding with a pairwise readout	0.663 (0.598)	0.537 (0.553)	0.894 (0.695)	0.728 (0.635)
pooled in-context embedding	0.818 (0.672)	0.805 (0.734)	0.854 (0.717)	0.835 (0.781)
CNP reader	0.824 (0.750)	0.822 (0.756)	0.827 (0.783)	0.849 (0.780)
ANP reader	0.806 (0.757)	0.814 (0.791)	0.847 (0.789)	0.848 (0.805)
context-window reader, our recipe	0.876 (0.777)	0.815 (0.770)	0.904 (0.813)	0.854 (0.808)
PLAYREAD without memory tuning	0.869 (0.809)	0.821 (0.780)	0.882 (0.811)	0.834 (0.784)
PLAYREAD	0.890 (0.838)	**0.836** (0.824)	**0.925** (0.912)	**0.873** (0.873)
PLAYREAD, scripted fine-tuning only	0.817	0.772	0.821	0.777

condition the tuned reader reaches 0.873 with or without expansion, so there tuning alone recovers what coverage adds. Fine-tuning was capped at 60 epochs for every method, where the validation accuracy of most baseline runs was still rising, and on the scripted test strategies every reader loses at most 0.02 after expansion.

N OVERCOOKED WITH HUMAN-PROXY PARTNERS

Overcooked-AI (Carroll et al., 2019) is the standard benchmark for human-AI coordination, and its human proxies, behavioral clones of human play, are the usual stand-in for people, who cannot be replayed against new partners. We port the environment to torch, and the port reproduces every state and reward of the 86 recorded human games and of 100,000 random transitions of the original. A condition is one of four layouts, the score is soups delivered, and a record holds 1 to 8 games. The held-out strategies are the human proxies and PPO agents, each scored against 100 pool partners with the truth from 32 games. The pool holds 50 PPO checkpoints and 50 further human proxies trained on other subsets of the human data, and the training partners, the pool and the probes are disjoint. Besides the Spearman correlation across partners we report the correlation that remains once the main effects of strategy and partner are removed, since a prior that reads nothing already ranks partners by strength.

Table 13: Overcooked, the mean over four layouts and three training seeds. The readers other than PLAYREAD are trained as published, the prior that ignores the record predicts from partner strength alone, and the ceiling plays the original strategies through the same estimator.

record	PLAYREAD	ANP	CNP	pooled	Prompt-DT	embedding + readout	cloning	prior	ceiling
Spearman across partners									
1 game	**0.819**	0.816	0.805	0.811	0.818	0.696	0.682	0.721	0.898
2 games	**0.838**	0.827	0.815	0.820	0.831	0.714	0.780	0.721	0.898
4 games	**0.853**	0.831	0.819	0.820	0.836	0.721	0.830	0.721	0.898
8 games	**0.864**	0.833	0.821	0.824	0.836	0.728	0.860	0.721	0.898
soups per game lost by teaming with the partner predicted best									
1 game	0.57	0.59	0.68	0.63	**0.56**	0.84	1.21	1.05	0.32
8 games	**0.42**	0.54	0.54	0.57	0.51	0.68	**0.42**	1.05	0.32
Spearman after the main effects of strategy and partner are removed									
1 game	0.699	0.696	0.674	0.693	**0.704**	0.642	0.474	−0.010	0.821
8 games	**0.775**	0.732	0.708	0.717	0.733	0.699	0.724	−0.010	0.821

The picture is the one the hill game gives (Table 13). From one game every population-trained reader is far ahead of a policy cloned per strategy and of an outcome-trained embedding, PLAYREAD by 0.138 with interval $[0.122, 0.152]$ and by 0.122. From two games on, where the memory is tuned on the record, PLAYREAD leads every column of Table 13, with cloning tying it on the soups lost at eight games. From one game Prompt-DT as published matches PLAYREAD, 0.818 against 0.819, and slightly ahead on the soups lost and the interaction, 0.56 against 0.57 and 0.704 against 0.699, but it gains little from longer records and reaches 0.836 at eight games against 0.864. Cloning draws level

on the ranking by eight games, 0.860 against 0.864 with intervals $[0.851, 0.868]$ and $[0.855, 0.873]$, and on the soups lost to a wrong partner, but not on the error in soups, 0.58 against 0.42, nor on the interaction, 0.724 against 0.775. The human proxies are harder to read than the PPO agents, 0.731 against 0.906 from one game, and their ceiling is lower, 0.847 against 0.949. At eight games they and the asymmetric-advantages layout are the sub-cells where cloning ranks better, 0.804 against 0.794 and 0.852 against 0.839. Tuning was not adopted at one game, where validation showed no gain. The readers trained with our recipe change little here, the ANP reader by at most 0.003 in Spearman.

O Poker

Leduc hold'em is a two-round poker game with a six-card deck, implemented exactly by OpenSpiel (Lanctot et al., 2019). A strategy is a table over the 936 information states, so payoffs and exploitability follow from the sequence form rather than from sampled hands, which removes the Monte Carlo error of simulated matches. Our sequence-form payoffs agree with OpenSpiel to 1.1×10^{-15}.

The population holds 3900 strategies, 2500 softmax networks over the information state, 600 temperature variants, 600 per-state convex combinations of two networks and 200 best responses to random members, smoothed toward uniform play. Two held-out sets of 200 are drawn afterwards, *mlp* from the same generator of random networks as the population and *mixed* with 50 of each family, and the pool holds 100 members of the training population, disjoint from both held-out sets. A record is k nested hands against a fixed panel of six probes or against uniform pool opponents. A 64-hand record holds about 131 decisions and reaches about 75 of the 936 information states.

Three baselines estimate the table. Counts normalize the recorded action counts with a Laplace prior, shrunk counts pull them toward the population mean, and the population basis fits a 16-dimensional latent code of a low-rank model of action logits, learned on the training population, to the recorded actions under a Gaussian prior, available in Leduc because a strategy is a finite table. All three were refit on this split, and every method receives the identical array of recorded triples.

Table 14: Leduc poker, records from the probe panel. Left, the Spearman correlation between predicted and exact payoffs across the 100 pool opponents. Right, the correlation between predicted and exact exploitability. Each entry averages 200 held-out strategies. PlayRead averages three training runs, two of them from the same seed, and its memory is tuned on the record at 64 hands only, where validation showed a gain.

		payoff Spearman				exploitability Spearman			
test set	method	k=1	4	16	64	k=1	4	16	64
mlp	counts	0.865	0.869	0.874	0.889	0.415	0.608	0.759	0.837
	counts, shrunk	0.862	0.867	0.874	0.892	0.464	0.587	0.726	0.821
	population basis	0.866	**0.900**	0.932	0.969	0.490	0.762	**0.878**	0.931
	PlayRead	**0.867**	0.898	**0.938**	**0.970**	**0.558**	**0.765**	0.875	**0.937**
mixed	counts	0.760	0.766	0.777	0.808	0.300	0.512	0.573	0.662
	counts, shrunk	0.767	0.776	0.789	0.823	0.326	0.497	0.545	0.633
	population basis	**0.774**	0.883	0.919	0.960	**0.393**	**0.641**	**0.764**	0.843
	PlayRead	0.765	**0.894**	**0.933**	**0.974**	0.330	0.633	0.755	**0.886**

Table 14 separates two comparisons. Against the two count baselines both learned methods are far ahead, 0.97 against 0.89 at 64 hands on *mlp* and 0.97 against 0.81 on *mixed*. Against the population basis, a strong domain-specific baseline, PlayRead ties on *mlp*, where the differences of the table means run from -0.002 to $+0.006$, and leads once the held-out strategies leave the manifold the basis was built from. On *mixed* with probe records PlayRead leads by 0.011 to 0.014 from 4 hands on, and for a single untuned reader the paired intervals at 16 and 64 hands, $[0.001, 0.023]$ and $[0.006, 0.019]$, exclude zero. The advantage comes almost entirely from the best-response family, 0.969 against 0.933 at 64 hands while counting reaches 0.463. From a single hand PlayRead is behind the basis by 0.009.

Two caveats belong with these numbers. The reader is not bitwise reproducible across runs, and two independent runs of the pipeline differ by up to 0.004 in payoff Spearman and 0.010 in exploitability Spearman, the size of the ties on *mlp* and smaller than the advantage on *mixed*, while the baselines and the truth are identical to machine precision. The reader also sees which probe each hand was played against and from which seat, where the count baselines see only state and action.

Policies produced by learning algorithms. A second held-out set tests reading outside the generators of the population, with no retraining. It holds 184 strategies, the average and current strategies of CFR and CFR+ at twelve checkpoints (eleven for CFR+ after removing a duplicate), of external- and outcome-sampling Monte Carlo CFR with three seeds each, and of extensive-form fictitious play, together with exact best responses to ten pool opponents. Their median distance to the nearest training strategy is 0.753 against 0.545 for the *mlp* set.

Table 15: Payoff Spearman on 184 strategies produced by learning algorithms, probe records. The lower block restricts to exact best responses to ten pool opponents. PlayRead is one reader. The last column is the paired difference between PlayRead and the population basis with a 95% bootstrap interval over strategies.

strategies	k	counts	counts, shrunk	population basis	PlayRead	PlayRead minus basis
all 184	1	0.178	0.199	**0.219**	0.203	-0.015 $[-0.020, -0.010]$
	4	0.182	0.206	**0.301**	0.296	-0.005 $[-0.030, +0.020]$
	16	0.190	0.214	0.364	**0.370**	$+0.006$ $[-0.029, +0.042]$
	64	0.219	0.245	**0.433**	0.381	-0.052 $[-0.079, -0.024]$
best responses	16	0.154	0.204	0.830	**0.941**	$+0.111$ $[+0.050, +0.197]$
	64	0.234	0.289	0.905	**0.971**	$+0.066$ $[+0.017, +0.132]$

Every method ranks these strategies poorly, because a strategy near equilibrium beats every pool opponent by nearly the same amount. In the third with the lowest exploitability the median standard deviation of true payoffs across the pool is 0.079 against 0.577 in the third with the highest. Among the most exploitable third, which resembles the population, the two learned methods tie at 0.93. The reader loses to the basis at 64 hands on the Monte Carlo CFR iterates, by 0.074 and 0.088, and its inferred tables are more exploitable than the originals, 2.05 against 1.32, because unrecorded states are filled with population-typical play while the basis has a weak prior and follows the record further. On best responses, whose sharp play the population holds in smoothed form, the ordering reverses in favor of PlayRead.

Covering the algorithms in the population. A leave-one-family-out design tests whether coverage repairs this. For each of the five families of Table 15 the population is expanded with 756 to 1024 strategies of the other families, on denser checkpoint grids and with discounted and linear CFR as training-only families, dropping strategies numerically identical to a held-out one. The reader is retrained from scratch with two seeds and the basis and shrinkage prior are refitted on the same population. The best responses never enter training.

Table 16: Payoff Spearman on 174 held-out algorithm strategies, probe records, pooled over the five folds, each strategy scored by the models of the fold that holds out its family. Original and expanded refer to the training population. The last column is the paired difference between the expanded reader and the expanded basis.

	original population		expanded population			
k	basis	PlayRead	counts, shrunk	basis	PlayRead	PlayRead minus basis
1	0.220	0.205	0.226	**0.349**	0.239	-0.110 $[-0.131, -0.090]$
4	0.281	0.271	0.231	0.453	**0.501**	$+0.048$ $[+0.009, +0.086]$
16	0.337	0.336	0.237	0.586	**0.720**	$+0.134$ $[+0.094, +0.176]$
64	0.406	0.348	0.263	0.727	**0.832**	$+0.105$ $[+0.073, +0.140]$

Coverage lifts both population-based methods, and the reader more, by 0.48 at 64 hands against 0.32 for the basis, so its advantage returns at 4, 16 and 64 hands (Table 16). Pool records give the same ordering, and the advantage holds on four of the five families, a tie on fictitious play. It is not family recognition, since in the least exploitable third, where the mean of the added strategies ranks the pool at 0.24, the reader reaches 0.81, matching an oracle that copies the nearest added strategy. Two limits remain. From a single hand the expanded basis is ahead by 0.110, because most of the population is random networks and one hand says little beyond the prior. Drawing training rows from the added strategies more often raises both methods from a single hand and leaves the basis ahead there, while the lead of the reader at 64 hands narrows to 0.045. And CFR and CFR+ have near-twins among the other added CFR variants (CFR+, discounted and linear CFR), at median total variation 0.081 and 0.024, so those two folds are closer to interpolation than to a new family, whereas on the Monte Carlo families the reader still leads the basis by 0.087 and 0.111.

P Language models in the iterated prisoner's dilemma

Six open instruction-tuned models play the 200-round iterated prisoner's dilemma with payoffs 5 for defecting against cooperation, 3 for mutual cooperation, 1 for mutual defection and 0 for cooperating against defection. They are Qwen3-8B and Qwen3-14B, the latter in FP8 weights (Qwen Team, 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Ministral-8B-Instruct-2410 (Mistral AI, 2024), Gemma-2-9B-it (Gemma Team, 2024) and GLM-4-9B-0414 (GLM Team, 2024), each served by vLLM. Every move is a separate request whose system message (for Gemma, the start of the user message) states the rules, the payoffs, the length and the goal, and whose user message lists both histories as strings of C and D. Decoding is greedy and the move is the first standalone C or D within eight tokens. None of the 44200 requests per model failed.

Each model plays one match against each of the 221 published strategies of the Axelrod library (Knight et al., 2016). The six matches against Cooperator, Defector, Tit For Tat, Random, Grudger and Alternator form its record and the other 215 serve only as truth. A token holds one round.

The 24000 synthetic strategies are programs from seven families, 7150 tables over the last one to three joint moves, 4751 Moore machines, 2936 grudges with optional forgiveness, and the rest threshold rules, periodic patterns, tables that switch at a fixed round and probers. Entries mix fixed moves with random defection probabilities, and a quarter add move noise below 0.12. Each has six probe matches and eleven target matches. The published strategies form five random folds, and a held-out strategy never enters training as a record, a target or the opponent in a target match.

The reader simplifies that of Section 3.1. A two-layer causal transformer encodes each recorded match, and its output at every round is both key and value, so no memory entry sees a later round. The query at round t sums embeddings of the previous joint move and of t with a projection of the mean memory entry. Three decoder layers of causal self-attention, cross-attention and an MLP feed a sigmoid head for the probability of defection, with width 128, four heads and 1.26M parameters. Training minimizes binary cross-entropy on four target matches given $k \in \{1, \dots, 6\}$ probe records, without interactive imitation, for 8000 AdamW steps of 48 strategies at learning rate 6×10^{-4}.

The inferred policy of a held-out published strategy samples its moves against the real implementations of the other non-probe strategies, scored as the mean of three plays against a truth of three seeded matches. The ceiling of 0.975 scores the seed-0 matches of the original instead and is optimistic, since seed 0 is one of the truth seeds. The truth for a language model is its first match against each non-probe strategy, so no seed ceiling applies. The count table holds cooperation frequencies over the last joint move and how often the opponent defected, capped at two, played exactly against the table of each opponent. Playing the more likely move instead of sampling lowers the correlation to 0.835 on the library. The ANP reader and the embedding with a pairwise readout of Fig. 5c are trained on the same folds, population and records. The ANP reader takes single recorded rounds as its context and the state of the live round as its query, with 1.59M parameters against 1.26M for PlayRead and 1.7 times its training time, and its inferred policy is played and scored as that of PlayRead. The embedding encodes the six probe matches contrastively and fits an MLP readout on 88k observed scores, and it predicts a score without playing. On the library they reach 0.868 and 0.707 against 0.903 for PlayRead, and over the six models 0.849 and 0.700 against 0.903.

The constructed opponents are deterministic tables over the first move and the last one or two joint moves, 32 with memory one and 2^{21} with memory two, enumerated exactly against the greedy inferred policy of one reader per seed as a game tree that branches where a table entry is first used, then scored against the sampling policy. Margins at search time are inflated by selection, so Table 17 replays the best opponent of each seed with fresh draws.

GLM-4-9B and Qwen3-14B almost never defect, and their opponents defect for good within the first three rounds. Most opponents of Ministral-8B cooperate at first and defect for good within two rounds or after the first defection by the model. For Llama-3.1-8B, whose inferred policy is the least accurate, the readers disagree, with predicted memory-two margins from -0.68 to 4.28. Against Qwen3-8B and Gemma-2-9B exact enumeration finds no margin above 0.025. For the four models that reproduce their recorded moves the measured margins match the predictions to within 0.01 or exceed them, while Ministral-8B and Llama-3.1-8B fall short. Against Qwen3-8B a published strategy with a longer memory takes 0.93 points per round, which no table with memory one or two achieves, so the limit there lies in the family searched.

Live verification. The predictions were frozen before any live match and a hash of the prediction file is stored with the code. The six models were then served again on one A100 GPU with the same

Table 17: Language models read from six probe matches each. Coop. is the fraction of cooperative moves over all 221 matches, and replay the fraction of the recorded probe moves of a model that it reproduces when the six probe matches are played again live. The Spearman columns compare predicted and real scores across the 215 non-probe published strategies, with PlayRead averaged over five fold readers and three training seeds, its memory tuned on the six matches, and the untuned reader beside it. The margins are the score per round of the best deterministic opponent minus that of the model, predicted as the mean over 15 readers and, after the arrow, measured as the mean of two live matches. The last row averages the 221 held-out published strategies.

			Spearman			margin, predicted → live	
model	coop.	replay	PlayRead	untuned	count table	memory one	memory two
Qwen3-8B	0.566	1.000	**0.961**	0.950	0.869	0.03 → 0.03	0.00 → 0.00
Llama-3.1-8B	0.752	0.921	0.695	0.772	**0.792**	0.02 → −1.34	2.05 → 0.26
Ministral-8B	0.439	0.798	**0.864**	0.820	0.781	2.88 → 1.55	2.89 → 2.35
Gemma-2-9B	0.285	0.999	**0.947**	0.931	0.882	0.03 → 0.03	−0.00 → 0.01
GLM-4-9B	0.999	1.000	**0.982**	0.965	0.942	4.98 → 4.98	4.98 → 4.98
Qwen3-14B	0.996	1.000	**0.971**	0.933	0.941	4.43 → 4.76	4.41 → 4.93
held-out library	–	–	**0.903**	0.896	0.798	–	–

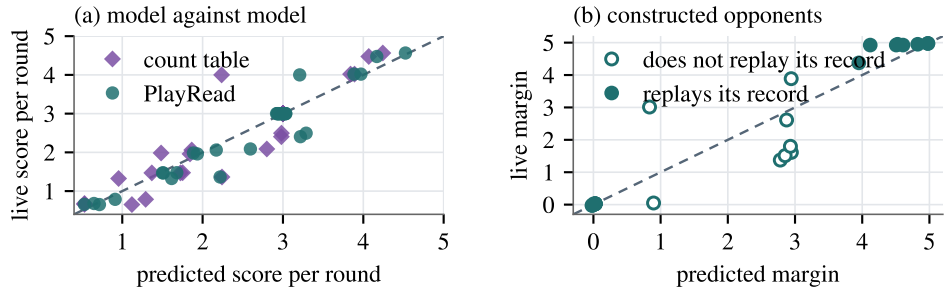

Figure 8: Language models. (a) Score per round of every ordered pair of models and of each model against itself, predicted before any live match against the live score, for PlayRead and the count table. (b) Margins of the opponents constructed against each model, predicted on the inferred policy against the live margin. Hollow markers are Llama-3.1-8B and Ministral-8B, which do not replay their own records.

vLLM version, decoding and prompts, each match twice, and none of the 35200 requests produced an unparseable reply. Replaying the six probe matches reproduces the recorded moves of Qwen3-8B, Gemma-2-9B, GLM-4-9B and Qwen3-14B almost exactly, while Llama-3.1-8B reproduces 92 percent and Ministral-8B 80 percent (Table 17, Fig. 8). Ministral-8B does not repeat itself between two identical matches on the same server, so greedy decoding sits near ties that batch composition and hardware flip and its record is one sample of a stochastic strategy. These two are also read least accurately, and Llama-3.1-8B is the one model that tuning the memory makes worse, 0.772 to 0.695. Interactive imitation was tried here with the synthetic strategies as oracles and lowered accuracy on the held-out library from 0.896 to 0.834, so the training procedure of Section 3.2 is established in the hill game and Overcooked rather than here.

The readers also predicted, before any live match and from the six probe matches alone, the score of every model against every other and against itself. Over two live matches per pair the sampled reader, registered as the primary prediction, errs by 0.17 points per round against 0.24 for the count table and correlates with the measured scores at 0.96 against 0.91. The greedy reader errs by 0.292. The count table ranks the pairs slightly better, 0.93 against 0.86, because 16 of the 36 pairs cooperate throughout at exactly 3 points per round, which the table predicts exactly and the reader places between 2.92 and 3.03.