KAIKA: REINFORCEMENT LEARNING FROM PAIRWISE COMPARISONS OF CONTINUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement-learning post-training reduces a model's behaviour to one scalar reward per response; Group Relative Policy Optimization (GRPO), the standard method, standardises these rewards within a group of responses to the same prompt. Many post-training outcomes are comparisons, a game won against an opponent or a puzzle solved where another attempt failed, and the scalar discards which continuation beat which. We ask whether a learner told only which of two continuations of the same state was better can post-train a model beyond GRPO. KAIKA is a population of policies that learns from such comparisons, with log-probabilities summed over each continuation: in games, a winner-vs-loser DPO term added to the policy gradient, pairing the played move with an alternative from the same position; on puzzles, where the comparisons replace the policy gradient, each successful attempt against each failed one. In chess, starting from a Stockfish-distilled 230M-parameter model, KAIKA gains $+156$ Elo over the base against GRPO's $+79$ at $T = 0.7$ (paired score difference $+0.0986$, 95% CI $[+0.0599, +0.1374]$, 5/5 seeds) and wins head-to-head on all 10 seeds; at $T = 0.3$ the paired difference is $+0.0404$ ($[-0.0098, +0.0906]$). The advantage over GRPO persists when every population member is an exact copy of the trained policy. On chess puzzles solved by 5M- and 32M-parameter language models, the comparison update exceeds GRPO with per-token-mean aggregation by $+0.0098$ at 5M (4/4 seeds) and $+0.0127$ at 32M (5/6 seeds), and by $+0.0167$ on three 32M seeds trained for 600 steps. Ablations locate part of the effect: replacing the summed log-probabilities with per-token means lowers accuracy on 3/3 seeds, by 0.0063 on average. Because an attempt ends at its first wrong move, failures are shorter than successes, and token averaging weights the correct opening moves a failure shares with a success more heavily in the failure; summing cancels their direct contribution. Neither population size (1 to 16 members), a curvature-based pair weight, nor a partial-credit reward adds measurable accuracy. Over training, the per-move rate of illegal moves rises by 41–91% under GRPO against 2–24% under KAIKA.

1 INTRODUCTION

Reinforcement-learning post-training reduces a model's behavior to a scalar reward per response. In Group Relative Policy Optimization (GRPO) (Shao et al., 2024), the standard method, the rewards of a group of responses to one prompt are standardised within the group, and each score is applied to its response's per-token mean log-probability. Many post-training outcomes are, however, comparisons: a game is won against a particular opponent, and a puzzle attempt succeeds where another from the same position fails. Reducing them to scalars discards which continuation beat which, and averaging over tokens gives the opening a failure shares with a success a larger negative weight in the shorter failure than the positive weight it receives in the success, so a correct step can receive a negative direct push (at initialisation, on 14–18% of chess puzzles with three or more moves).

This paper asks whether a learner that is told only which of two continuations of the same state was better can post-train a model further than GRPO given the same outcomes.

We introduce KAIKA, a population of policies that learns from comparisons between a better and a worse continuation of the same state, each scored by its log-probability summed over tokens. In

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.

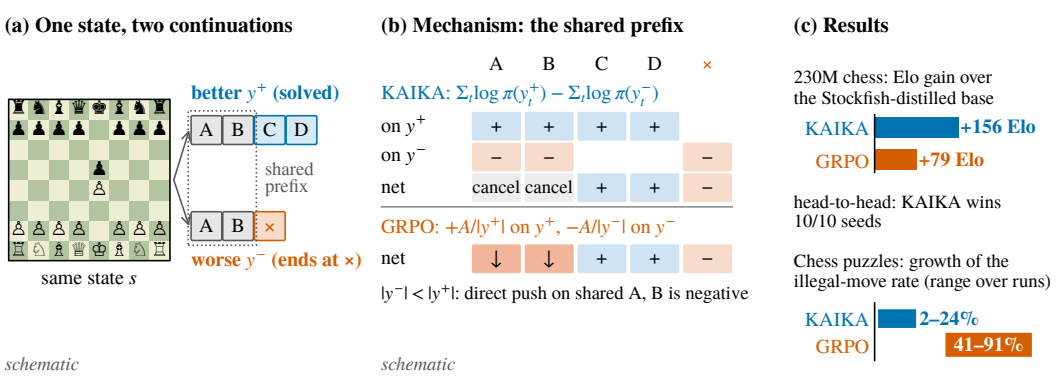

Figure 1: KAIKA learns from comparisons between two continuations of the same state. (a,b) Schematic. The pair is ordered by engine evaluation in chess and by the verifier on puzzles; in the summed difference $\log \pi(y^+) - \log \pi(y^-)$ the shared moves' direct contributions cancel, while GRPO's per-token mean weights the shorter failure by $1/|y^-| > 1/|y^+|$ and gives the shared correct moves a negative direct push. (b) is drawn for puzzles, where attempts end at their first wrong move; in chess games the pair is a single move. (c) 230M chess at $T = 0.7$ (5/5 seeds; head-to-head 10/10), and growth of the illegal-move rate from the first to the last 20 training steps (range over runs, 5M and 32M pooled).

games, a winner-vs-loser Direct Preference Optimization (DPO) term beside GRPO's policy gradient pairs the played move with an alternative legal move from the same position, and the members generate their training games by playing one another. On puzzles, where the comparisons replace the policy gradient, each successful attempt is paired against each failed one across the rollouts of all members. Summing over tokens makes a comparison cancel exactly the moves two continuations share, so the direct gradient falls only on the moves where they differ (Figure 1b).

In chess, starting from the Stockfish-distilled LFM2.5-230M-Chess model (Labonne, 2026), KAIKA gains $+156$ Elo over the base at evaluation temperature $T = 0.7$ against $+79$ for GRPO trained in the same recipe, and beats GRPO head-to-head on all 10 seeds (§5.1; per-seed scores in App. Figure 4); at $T = 0.3$ the paired difference is smaller, $+0.0404$ on 4/5 seeds. On chess puzzles solved by 5M- and 32M-parameter language models, the comparison update exceeds GRPO on 4/4 and 5/6 seeds (§5.2).

The method works; which of its parts carries the gain is only partly resolved. Summed aggregation accounts for part of the puzzle margin (§5.2; App. Table 11). The advantage over GRPO in chess persists when every population member is an exact copy of the trained policy, and on puzzles populations of 1 to 16 members reach the same accuracy, so a population of distinct members is not yet shown to add to it.

Contributions. **(1)** A comparison update between continuations of one state, scored by summed log-probabilities: a winner-vs-loser DPO term beside the policy gradient in games (§3) and a verifier-ordered pair loss across the members' rollouts on puzzles (§3.1); App. C states the full population objective. **(2)** Chess post-training of a distilled 230M model against GRPO, RLOO and REINFORCE in the same recipe, with GRPO's learning rate swept (§5.1). **(3)** Puzzle ablations of population size, pair weight, reward, token aggregation and a diversity term against same-seed GRPO (§5.2). Game beds and engine-free chess are in App. A.

2 Related Work

Preference learning. Preference optimization learns from pairwise comparisons (Christiano et al., 2017; Rafailov et al., 2023); GRPO baselines responses within a sampled prompt group (Shao et al., 2024). Online GRPO and online/semi-online DPO perform similarly on studied math and instruction tasks (Lanchantin et al., 2025); action-value distillation improves chess-puzzle GRPO over exact-move binary rewards (Hwang et al., 2025). In games, KAIKA adds to GRPO's policy gradient (PG) term a same-position winner-vs-loser DPO term (§3), a single-position form of the positive-versus-

negative contrast in GRPO's gradient (Wu et al., 2025). Direct Group Preference Optimization (DGPO) and GDPO replace the PG with a group DPO loss (Luo et al., 2026; Yi et al., 2026); the closest GRPO and DPO combinations add a pairwise term to GRPO on its own rollouts: Dynamic Policy Optimization's (DYPO) group alignment loss (Zhu et al., 2026) and SSR-GRPO's retrieval DPO over mined hard negatives (Song et al., 2026); Pref-GRPO turns preference-model win rates into GRPO rewards (Wang et al., 2025). General Preference Reinforcement Learning (GPRL) carries structured preferences into policy updates (Umer et al., 2026); Markov Chain from Human Feedback (MCHF) builds a Markov kernel from pairwise utilities (Koriyama & Liang, 2026); SPIRAL trains language models by self-play on zero-sum text games (Liu et al., 2025a).

Shared populations. Population learning is standard in non-transitive multi-agent games: AlphaStar trains against current and historical policies, Policy-Space Response Oracles (PSRO) expand a policy set, Neural Population Learning (NeuPL) encodes one in a conditional network, and Spinning Tops relate diversity to game structure and self-play cycles (Vinyals et al., 2019; Lanctot et al., 2017; Liu et al., 2022b;a; Czarnecki et al., 2020; Balduzzi et al., 2019). Multiplayer Nash Preference Optimization (MNPO) trains against several opponents and Population Based Training (PBT) replaces weak members; KAIKA replaces low-rated members with copies of the shared policy (an Elo screen) (Wu et al., 2026; Jaderberg et al., 2017). AZdb, the closest prior chess population, learns latent-code players in one AlphaZero network using Monte Carlo tree search (MCTS) and a diversity reward; no tested matchmaker beat self-play (Zahavy et al., 2024). AlphaZero also searches at each move; KAIKA does not (Silver et al., 2018).

Pairwise and population learning from verifiers. Several recent methods read GRPO with a binary verifier as a contrast between successes and failures. Mroueh (2025) derives it as an adaptive weighted contrastive loss with the per-question weight $\sqrt{p(1-p)}$, and It Takes Two reads it as an N-versus-M contrastive loss, where N and M denote successful and failed responses (Wu et al., 2025). DisCO drops that weight (Li et al., 2025); with the identity surrogate and summed-token scores, DisCO's discriminative objective J_1 is the single-policy form of DUEL, our puzzle update (§3.1). Dr. GRPO removes both GRPO's standard-deviation normalization and its per-response length normalization (Liu et al., 2025b); DUEL shares only the latter. GroupDPO trains on every correct–incorrect pair in a group (Lanchantin et al., 2025). Against this family, we add a matched-rollout comparison with GRPO at 5M and 32M parameters and the ablations of §5.2.

Populations of language-model learners share rollouts in Swarm sAmpling Policy Optimization (SAPO) (Amico et al., 2025) and Mutual RL (Liu et al., 2026), but each member still takes a GRPO update. Population-based training has used pairwise comparisons between agents to adapt hyperparameters (Bai & Cheng, 2024); here the comparison is the training signal itself. Diversity via Determinants (DvD) rewards diversity directly, maximizing the volume of the members' behavioural embeddings (Parker-Holder et al., 2020). Without such a term our members stay close to one another; with it at weight 0.1 they differ on every seed, but their mean accuracy does not rise (§5.2). None of these methods trains each member on verifier-ordered pairs across all members' rollouts with no policy-gradient term, as DUEL does; at this scale that population update is indistinguishable from the single-policy duel.

3 KAIKA AS TRAINED

KAIKA keeps $K = 8$ policy slots and learns from the games its members play against one another. The instance in this section is trained from the chess game outcomes reported below. App. C states the complete objective, in which each member keeps its own GRPO group, learns from the other members' responses through reward-gap-weighted DPO and descends from founders forked from one GRPO-trained parent (schematic: App. Figure 12); App. Table 2 lists the terms each result trains, and App. Figure 5 lists which components each version has.

Population. Slot 0 holds a shared policy θ, and the members are low-rank offsets of it, so every update of θ moves every member; outcomes update Elo standings, and low-rated members restart as jittered copies of θ (App. A.1).

One step. App. Figure 11 draws it. **(i) Play.** Slots duel in consecutive random perfect matchings with alternating seats. **(ii) Credit.** In engine-assisted chess each played move is credited according to the change in engine evaluation resulting from that move, at 2,000 nodes, and at every labelled branch an alternative legal move is sampled from $q(a' \mid s, a) = \pi(a' \mid s)/(1 - \pi(a \mid s))$, $a' \neq a$, the member's distribution with the played move a excluded; the engine's evaluation orders the pair. Where no engine is called, a learned move critic, trained on game outcomes over θ's features, credits each move in place of the engine. On Nim each move receives its game's outcome as credit, and one pair per ply compares the winner's and the loser's moves. **(iii) Train.** Each member learns from its own credited rows by a clipped PG whose advantage is the sign of the row's credit minus the step mean; the live members' mean gradient is subtracted from each member's. θ receives

$$\mathcal{L}_{\text{assist}}(\theta) = \mathcal{L}_{\text{batch}}(\theta) - \frac{1}{|\mathcal{E}|} \sum_{(s,a^+,a^-)\in\mathcal{E}} \log \sigma\big(0.1\,[r_\theta(s,a^+) - r_\theta(s,a^-)]\big)\,, \tag{1}$$

with $r_\theta(s,a) = \log \pi_\theta(a \mid s) - \log \pi_{\text{ref}}(a \mid s)$, summed over the response's tokens, against the frozen starting policy. The second term is the winner-vs-loser DPO term (Rafailov et al., 2023) over the step's pairs $\mathcal{E}$. $\mathcal{L}_{\text{batch}}$ is a policy gradient with REINFORCE++ global normalization across the step's distinct positions (Hu et al., 2025), reference KL and detached pre-write likelihoods; the ratio is one, the clip inactive and only the legal move token is trained, with sampling and scoring masked to legal moves (Carta et al., 2023; Tan et al., 2024; Huang & Ontañón, 2020). On fixed inputs its gradient matches TRL's `GRPOTrainer` (Hugging Face, 2026); the rollout group is a budget of self-play games and does not repeat samples per state.

3.1 Pair learning on chess puzzles

On puzzles, KAIKA's comparison takes the form of DUEL: the verifier orders every pair of attempts at one puzzle, pairs cross member boundaries, and the pair loss below replaces the PG.

Population and shared rollouts. A prompt x is a chess puzzle; a rollout y is the model's sequence of player moves, with the opponent's replies supplied by the environment; the verifier returns $r(y) = 1$ if every player move matches the solution line and $r(y) = 0$ otherwise. The population holds K policies $\pi_1, \ldots, \pi_K$ and a frozen reference π_{ref} (the pretrained model). For each prompt every member samples G rollouts, giving a shared pool $y_1, \ldots, y_N$ with $N = KG$ and owner $o(i) \in \{1, \ldots, K\}$. We use $K = 4$, $G = 4$, so $N = 16$ equals the group size of the single-policy GRPO baseline. For member k let $\ell_k(y) = \log \pi_k(y \mid x)$ summed over player tokens, $\bar{\ell}_k(y)$ its per-token mean, and $\text{sg}(\cdot)$ stop-gradient.

GRPO. The single-policy baseline standardizes the reward within its group of N rollouts and applies it to per-token mean log-probabilities,

$$\mathcal{L}^{\text{GRPO}} = -\frac{1}{N}\sum_{i=1}^{N} \hat{A}_i\, \bar{\ell}(y_i) + \frac{\lambda_{\text{G}}}{N}\sum_{i=1}^{N} \text{sg}\big(\bar{\ell}(y_i) - \bar{\ell}_{\text{ref}}(y_i)\big)\bar{\ell}(y_i), \qquad \hat{A}_i = \frac{r(y_i) - \bar{r}}{\text{std}(r) + 10^{-4}}, \tag{2}$$

with $\bar{r}$ and $\text{std}(r)$ taken over the group and $\lambda_{\text{G}} = 0.001$.

Pair loss. The pair set is $\mathcal{P} = \{(a,b) : r(y_a) > r(y_b)\}$ over all N rollouts, so pairs cross member boundaries. Member k scores every rollout in the pool with the implicit reward $s_k(y) = \text{sg}\big(\ell_k(y) - \ell_{\text{ref}}(y)\big)$, and $\sigma_{ab} = \sigma\big(\beta\,(s_k(y_a) - s_k(y_b))\big)$ is its Bradley–Terry probability that y_a beats y_b. The pair loss of member k is

$$\mathcal{L}_k^{\text{duel}} = -\sum_{(a,b)\in\mathcal{P}} \text{sg}(E_{ab})\,\frac{\ell_k(y_a) - \ell_k(y_b)}{c}, \qquad E_{ab} = \frac{1 - \sigma_{ab}}{\sigma_{ab}(1 - \sigma_{ab}) + \delta}\cdot\frac{1}{|\mathcal{P}|}, \tag{3}$$

where c is a constant normalizer ($c = 8$). The weight divides the pair's logistic residual $1 - \sigma_{ab}$ by its damped Bernoulli Fisher information, a damped Fisher-scoring step on the pair margin. Because s_k is detached, β enters the loss only through σ_{ab} inside this weight: as $\beta \to 0$, $\sigma_{ab} \to \frac{1}{2}$ and, with $\delta = \frac{1}{4}$, $E_{ab} \to 1/|\mathcal{P}|$ for every pair. At a one-nat margin $E_{ab}|\mathcal{P}|$ is 0.95 at $\beta = 0.1$ and 0.16 at $\beta = 3$. In a 5M run at $\beta = 0.1$, σ_{ab} stayed within $[0.46, 0.54]$ over the founding step and the first 10 updates, where $E_{ab}|\mathcal{P}|$ lies within $[0.92, 1.07]$. Peer rollouts enter without an importance ratio.

The pair loss at $E_{ab} = 1/|\mathcal{P}|$. With binary rewards, a puzzle with s successes among the N rollouts has $|\mathcal{P}| = s(N-s)$ pairs, and Eq. (3) with a constant weight reduces to

$$\mathcal{L}_k^{\text{duel}} = -\frac{1}{c}\Big(\frac{1}{s}\sum_{i:\, r(y_i)=1} \ell_k(y_i) - \frac{1}{N-s}\sum_{i:\, r(y_i)=0} \ell_k(y_i)\Big), \tag{4}$$

the difference between the mean summed log-probability of the successful and of the failed rollouts. Every decisive puzzle receives the same total weight $1/c$, whatever its pass rate $p = s/N$. GRPO with a group of N applies the same contrast to per-token mean log-probabilities with weight $\sqrt{p(1-p)}$. Restricted to one policy, Eq. (4) is the discriminative objective J_1 of DisCO with the identity surrogate (Li et al., 2025, Eq. 7), with summed-token rather than token-mean scores; the retain term below adds an anchor to the base model.

Shared prefixes. An episode ends at the first wrong move, so a failed rollout b shares its correct opening moves h with a successful rollout a and is shorter. Writing $a = (h, a')$ and $b = (h, b')$,

$$\ell_k(a) - \ell_k(b) = \log \pi_k(a' \mid x, h) - \log \pi_k(b' \mid x, h), \tag{5}$$

so the pair term puts no direct gradient on the tokens of h; updates to shared parameters can still change their probabilities. Under Eq. (2) a token of h receives weight $\hat{A}_a/|a| + \hat{A}_b/|b|$ from the two rollouts, which is negative when $\hat{A}_b < 0$ and $|b|$ is short enough relative to $|a|$. App. B gives the size of this effect at initialization and the per-token mean variant of the duel.

Retain term. Each member is anchored to the reference on its own rollouts, and the duel member's loss is

$$\mathcal{L}_k = \mathcal{L}_k^{\text{duel}} + \mathcal{L}_k^{\text{ret}}, \qquad \mathcal{L}_k^{\text{ret}} = \frac{\lambda}{G}\sum_{i:\, o(i)=k} \text{sg}\big(\bar{\ell}_k(y_i) - \bar{\ell}_{\text{ref}}(y_i)\big)\, \bar{\ell}_k(y_i), \tag{6}$$

where $\mathcal{L}_k^{\text{ret}}$ is a length-weighted reference regularizer: with the rollouts held fixed its gradient is that of $\frac{\lambda}{2G}\sum_{i:\, o(i)=k}\big(\bar{\ell}_k(y_i) - \bar{\ell}_{\text{ref}}(y_i)\big)^2$, which weights each rollout's summed log-ratio by $1/|y_i|^2$ for $|y_i|$ player tokens, so shorter rollouts are held closer to the reference. The KL term of Eq. (2) has the same form (App. B).

Three instantiations. DUEL trains each member on Eq. (6) only. SPEC keeps member k's own-group GRPO term $\mathcal{L}_k^{\text{grpo}} = -\frac{1}{G}\sum_{i:\, o(i)=k} \hat{A}_i\, \bar{\ell}_k(y_i)$, with $\hat{A}_i$ the reward standardized within k's own G rollouts, and adds a pair term over the same $\mathcal{P}$ with the DPO residual in place of the Fisher weight, $\mathcal{L}_k = \mathcal{L}_k^{\text{grpo}} + \lambda_{\text{pair}}\,\mathcal{L}_k^{\text{dpo}} + \mathcal{L}_k^{\text{ret}}$, where $\mathcal{L}_k^{\text{dpo}}$ is Eq. (3) with $E_{ab} = (1-\sigma_{ab})/|\mathcal{P}|$. UNIFIED mixes four terms for each member and puzzle: the reward standardized over all N rollouts and applied to the member's own rollouts, the DPO pair term of SPEC, a coverage term that rewards the member's successes on puzzles the other members fail, and the retain term; the mixing weights are shares computed from the member's rollout-level residual. Losses are averaged over the prompts of a batch and all members take one AdamW step together. A puzzle on which all N rollouts succeed or all fail has $\mathcal{P} = \emptyset$; its pair loss is zero, its retain term still applies, and it still counts in the average over the prompts of the batch (for GRPO such a group has $\hat{A}_i = 0$ and only the KL term acts).

Founding and evaluation. With $K > 1$, DUEL and SPEC start from a founding step: one GRPO update of a parent with 16 rollouts per prompt, then K copies of the parent, each updated on the parent's K rollouts per prompt with weight $(1 + \alpha\,[g = k])/(K + \alpha)$ on rollout g, where $[g = k]$ is an indicator equal to 1 when $g = k$ and 0 otherwise. We report greedy pass@1 on the 1164 pooled test puzzles, averaged over the K members, and compare each arm with GRPO on the same seed. Hyperparameters: $\beta = 0.1$, $\delta = 0.25$, $c = 8$, $\lambda = 0.05$, $\lambda_{\text{pair}} = 1$, $\alpha = 8$, founding learning rate 5×10^{-5}, learning rate 10^{-5}, gradient norm clipped at $\sqrt{K}$ over all members, 32 prompts per step, 200 steps. GRPO uses 16 rollouts per prompt, the same learning rate and a KL coefficient of 0.001. Its loss averages the token terms within each sequence (the `grpo` loss type of TRL), which TRL does not recommend because of its length bias; the TRL default normalizes over all tokens in the batch (Hugging Face, 2026). An ablation adds summed aggregation to GRPO: the token terms are summed and divided by the constant c (TRL's `dr_grpo` loss type; the default divides by the batch's token count instead). Unless named GRPO + summed aggregation, every comparison with GRPO in §5.2 is against the per-token mean. Runs are not bit-reproducible: same-seed reruns of one configuration differ by up to 0.0069.

4 EXPERIMENTAL SETUP

Chess. KAIKA post-trains LFM2.5-230M-Chess (Labonne, 2026), a Stockfish-distilled base, for 300 steps at 32 self-play games per step, with engine credit as in §3. A replaced member restarts from θ plus jitter of 10^{-3} per coordinate, norm 2.6 over the 6.75M low-rank coordinates (App. F.2).

Comparators. In chess and Nim, GRPO trains $\mathcal{L}_{\text{batch}}$ of Eq. (1) alone, with one advantage group per batch as in REINFORCE++ (Hu et al., 2025); in chess, on fixed inputs, its loss gradient matches TRL's `GRPOTrainer` (App. F.1). Its loss divides by the batch's token count, TRL's default `dapo` normalization, so it has no per-sequence length bias. In chess it runs at learning rate 10^{-5}, swept from 2.5×10^{-6} to 2×10^{-5} in App. F.2. It shares KAIKA's checkpoint, games per step, engine credit and branch moves, and trains 64 rows per step against KAIKA's 32; it trains the branch moves as independently credited policy-gradient rows drawn from q, so its update on them is not on-policy even with ratio one, while KAIKA uses them only in its pair term. Each arm trains on its own self-play games under the same game budget; GRPO puts its single policy in both seats (White/Black) (App. F.1). RLOO (Ahmadian et al., 2024) and REINFORCE run in the same recipe.

Scores. The score of policy A against policy B is A's wins plus half its draws, divided by the n games played with seats alternating; 0.5 is parity. Chess duels play $n = 1024$ games, Nim duels 128 and Goofspiel duels 200, at sampling temperature $T = 0.7$ or $T = 0.3$. A duel pairs the two arms' final policies from one training seed; "k/N_{seeds}" counts the seeds scoring above 0.5, intervals are 95% t intervals over seeds, and a seed's duels at both temperatures and at equalized entropy share one duel seed. Elo over the base converts a score s against the starting checkpoint into $400 \log_{10}[s/(1-s)]$; ladder Elo is a maximum-likelihood fit against Stockfish (The Stockfish developers, 2025) anchors (App. Table 18).

5 RESULTS

We report chess first, where the instance of §3 is trained against GRPO in a matched recipe (§5.1), and then chess puzzles, where the comparison update replaces the policy gradient and its components are removed one at a time against same-seed GRPO (§5.2). Scores, Elo and seed counts are as defined in §4.

5.1 CHESS AGAINST GRPO AND THE BASE

In direct duels at $T = 0.7$ against the starting checkpoint (5 seeds), KAIKA scores 0.7107 and GRPO 0.6121, +156 against +79 Elo over the base, about twice GRPO's gain; the paired difference is +0.0986 (95% CI $[+0.0599, +0.1374]$, 5/5 seeds). At $T = 0.3$ the paired difference is +0.0404 (95% CI $[-0.0098, +0.0906]$, 4/5 seeds), and the interval includes zero. Against a ladder of Stockfish anchors at $T = 0.7$, the gains over the base are +165 against +88 Elo, a paired +77 Elo difference (95% CI $[+53, +100]$) on 10/10 seeds. The base and GRPO score below 0.5 at the lowest rung (1320) on every seed, so their ratings extrapolate the fit below it; KAIKA's rating lies between two anchors on 5/10 seeds and is extrapolated on the rest. At $T = 0.3$, where sampling concentrates on each policy's most probable moves (the absolute-strength setting), KAIKA rates 1453 and GRPO 1425; their paired difference is +29 Elo (95% CI $[-5, +62]$), and under greedy decoding it is +11 Elo ($[-29, +51]$; App. Table 18).

Head-to-head at $T = 0.7$, KAIKA beats GRPO on all 10 seeds (+71 Elo by the same conversion; two-sided sign $p = 0.002$); on the planned five seeds KAIKA scores 0.6121 against GRPO (5/5) and on seeds 6–10 0.5896 (5/5). At $T = 0.3$ it scores 0.5268 (95% CI $[0.493, 0.560]$; 7/10 seeds). KAIKA's policy is sharper; at move entropy equalized on a fixed probe set it scores 0.5127 ($[0.479, 0.546]$; GRPO cooled) and 0.5234 ($[0.490, 0.557]$; KAIKA heated), above parity on 7/10 and 7/10 seeds (App. F.1). KAIKA also beats REINFORCE Leave-One-Out (RLOO) and REINFORCE in the same recipe on every seed at both temperatures (Table 1; App. Table 20). Across GRPO learning rates 2.5×10^{-6} to 2×10^{-5} and a KL coefficient of 0.01 (2 seeds; App. F.2), every two-seed mean score of KAIKA is above 0.5; the lowest individual-seed score is 0.4707, against the recipe at seed 2 and $T = 0.3$, and the strongest comparator is learning rate 5×10^{-6} at $T = 0.7$ (0.5974).

Table 1: Mean chess score of KAIKA's θ against single-policy comparators over seeds (§4), and seeds above 0.5. Engine-free rows: KAIKA trains 126–128 rows per step; per seed: App. Table 17.

Single-policy comparator	Seeds	$T = 0.7$		$T = 0.3$	
		Score	> 0.5	Score	> 0.5
GRPO	10	0.6009	10/10	0.5268	7/10
GRPO at KAIKA's entropy	10	0.5127	7/10	–	–
RLOO	5	0.6336	5/5	0.5437	5/5
REINFORCE	5	0.6187	5/5	0.5457	5/5
GRPO (engine-free, 64 rows) vs KAIKA without critic	5	0.5549	4/5	0.5567	4/5
GRPO (engine-free, 128 rows) vs KAIKA without critic	5	0.5036	3/5	0.5015	3/5

With exact copies of θ in the member slots, the winner-vs-loser DPO term adds +0.0689 at $T = 0.7$ (95% CI $[+0.0291, +0.1088]$; App. Table 4), so the gain does not require learned member diversity. A 300-step run costs 2.22 GPU-hours for KAIKA and 2.75 for GRPO with engine credit, and 2.34 and 1.33 engine-free at 64 rows (App. Table 29).

5.2 CHESS PUZZLES

App. Table 6 lists the results (models, data and protocol: App. B.1). At 5M, GRPO reaches 0.0649 over seeds 1–4 of one batch, from a base of 0.0567. DUEL is above GRPO by +0.0098 on 4/4 same-batch seeds (paired $t = 4.52$, df 3, $p = 0.020$; App. Table 6 adds seeds 5–8 against a second GRPO batch: +0.0030, 4/4). UNIFIED is above GRPO by +0.0061 on 3/3 seeds ($t = 19.10$) and SPEC by +0.0047 on 4/4 ($t = 4.60$); at 32M they are above it by +0.0043 and +0.0020 (4/6 each; $t = 1.17$ and 0.76). At 32M, GRPO reaches 0.1256 over 6 seeds, from a base of 0.0919. DUEL minus GRPO on seeds 1–6 is +0.0036, −0.0037, +0.0234, +0.0246, +0.0075 and +0.0208, a mean of +0.0127 (paired $t = 2.63$, df 5, $p = 0.047$). On seeds 3–6, which have per-puzzle outputs, the per-puzzle difference is +0.0191 (paired $t = 3.65$ over the 1,164 puzzles, seeds averaged per puzzle, df 1163, excluding seed-to-seed variance), and the served DUEL member alone solves 224 puzzle–seed pairs against 136 for GRPO alone ($p = 4.1 \times 10^{-6}$, exact binomial with puzzles independent within a seed). With three times the training (600 steps) at 32M, DUEL is above GRPO on all three seeds run (0.1463 against 0.1400, 0.1493 against 0.1229 and 0.1456 against 0.1280), by +0.0167 on average (seed-level $t = 2.87$, df 2, $p = 0.10$; per-puzzle $t = 3.15$, df 1163); the served DUEL member alone solves 159 puzzle–seed pairs against 100 for GRPO alone ($p = 3.0 \times 10^{-4}$). The advantage persists on these three seeds (+0.0078 at 200 steps).

App. Table 9 and Figure 3 collect the ablations; their 5M rows come from the batch of the main comparison. Same-seed reruns differ by up to 0.0069, so ablation gaps below that size are not separated by single runs. A single-policy duel without founding ($K = 1$, 16 rollouts) is above GRPO by +0.0086 (4/4). GRPO at learning rate 3×10^{-5} is below GRPO at 10^{-5} on both seeds, by 0.0027 on average, and at 10^{-4} it collapses to 0.0095 and 0.0112. With the KL coefficient lowered to $\lambda = 0.01$, DUEL is above GRPO by +0.0059 on 4/4 of seeds 1–4 ($t = 6.94$; learning-rate and KL sensitivity: App. Table 10).

Illegal moves. GRPO's rate of illegal moves rises much faster over training than the duel's, and the two ranges do not overlap at either scale (Figure 2, App. Table 12). From the first 20 to the last 20 of the 200 steps, the illegal-move rate per decision rises by +41.0 to +71.5% for GRPO at 5M (12 runs) and +62.0 to +91.2% at 32M (6 runs), against +2.4 to +24.3% and +8.4 to +22.9% for DUEL (10 and 6 runs). DUEL starts from a higher rate (5M: about 0.21 against 0.19; 32M: 0.13–0.14 against 0.12); its increase remains small. GRPO with the duel's anchor rises by +33.3% and +45.3% at 5M (two seeds; the lowest GRPO run is +41.0%), so part of what the anchor buys is keeping probability on legal moves. The pair loss, the summed scores and the stronger anchor each lower the growth (App. B).

Training reward and test accuracy. During training the duel's sampled solve rate is slightly below GRPO's (5M 0.0667 vs 0.0737; 32M 0.1197 vs 0.1230, App. Figure 10; over the last 50 steps 0.0706 vs 0.0749 at 5M and 0.1245 vs 0.1276 at 32M), yet under greedy decoding on 1,024

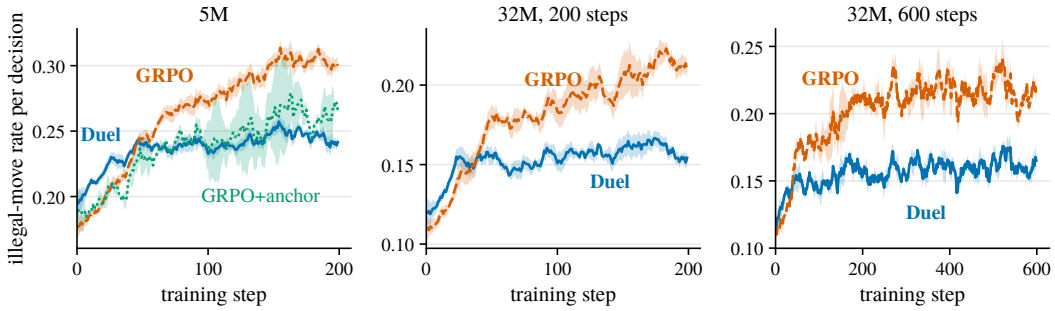

Figure 2: Illegal-move rate per decision over training, DUEL ($K = 4$) and GRPO; green: GRPO with KL coefficient 0.05 (5M, two seeds). Trailing 10-step mean per run; line: mean over runs; band: ± 1 standard error. Runs: 5M, 10 DUEL and 12 GRPO; 32M at 200 steps, 6 and 6; 32M at 600 steps, 3 and 3.

of each run's own training puzzles it is higher on all eight seeds tested, by $+0.0316$ at 5M (reruns of seeds 1–4) and $+0.0494$ at 32M (seeds 3–6, the runs with saved weights, whose bench gain is $+0.0191$), and on the held-out bench it is higher as well (App. Table 6). The duel's lower sampled reward therefore comes with higher greedy accuracy on the puzzles it trains on.

What the gain consists of. Increasing GRPO's learning rate at the tested settings did not recover the duel's gain. Giving GRPO the duel's KL anchor to the base model ($\lambda_G = 0.05$ instead of 0.001) changes it by $+0.0025$ and $+0.0000$ at 5M (seeds 1 and 2). At GRPO's own coefficient 0.001 the single-policy duel is above GRPO by $+0.0129$, $+0.0120$ and $+0.0068$ on seeds 1–3, and differs from the four-member duel at 0.05 by $+0.0084$, -0.0006 and -0.0013. The anchor therefore does not appear to account for the duel's gain. The pair weight is close to one at the default β, and making it active lowers accuracy (App. Table 7). A flat per-question weight alone does not exceed GRPO: Uniform-REINFORCE, which weighs all prompts equally, saturates at the accuracy of GRPO in Heckel et al. (2026).

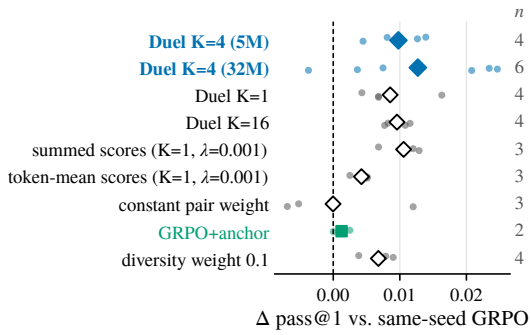

Figure 3: Final pass@1 minus same-seed GRPO per ablation (dots: seeds; diamond: mean); diversity weight 0.1 uses founding noise 0.03. All 19 arms: App. Figure 9.

GRPO scores a rollout by its mean token log-probability. The duel divides the summed log-probability by a constant ($c = 8$), which is the correction Dr. GRPO proposes for GRPO's response-length bias (Liu et al., 2025b). The ablation GRPO + summed aggregation applies this correction to GRPO (App. Table 11). It is above GRPO by $+0.0039$ at 5M (3/4 seeds) and $+0.0089$ at 32M (5/6), and DUEL is above it by $+0.0059$ (3/4) and $+0.0039$ (3/6). At 32M the single-policy summed duel is level with this ablation ($+0.0026$, 2/4 seeds) and above per-token-mean GRPO by $+0.0146$ on the same seeds, where the ablation is above it by $+0.0120$. Dr. GRPO's case is a long incorrect response. In chess a failed rollout stops at its first wrong move, so it is shorter than a successful one (mean length ratio 2.1–2.8 at initialization) and shares its correct opening moves. Per-token averaging then weights those shared correct moves more heavily in the failure than in the success and can make their direct push negative (Eq. (5); App. B); this penalization of shared correct tokens is documented for GRPO by Deng et al. (2025). The duel's pair term puts no direct gradient on a correct shared move; updates to shared parameters can still change its probability, and the retain term can move any token toward the base model. Replacing the summed log-probabilities with per-token means (App. B, Eq. (7)), with nothing else changed, lowers the single-policy duel by 0.0078, 0.0069 and 0.0043 (5M, three seeds, KL coefficient 0.001 as for GRPO). The summed-score duel is above GRPO by the values above and the mean-score duel by $+0.0051$, $+0.0051$ and $+0.0025$; this remaining difference is not the length

correction. The per-puzzle weights also differ: GRPO weighs the success-minus-failure contrast by $\sqrt{p(1-p)}$, Dr. GRPO by $p(1-p)$ and the duel uniformly (Eq. (4)); the constant-pair-weight row of App. Table 9 isolates this weight. With a constant pair weight ($\beta = 0$) and per-token scores, the single-policy duel differs from GRPO only in the flat per-puzzle weight; it differs from GRPO by -0.0052, -0.0069 and $+0.0120$ (seeds 1–3), and the mean-score duel differs from it by $+0.0103$, $+0.0120$ and -0.0095. The flat per-puzzle weight gives no consistent gain, and the remainder is not resolved at three seeds; only the aggregation term keeps its sign on every seed.

The population. At this scale the population has no measurable effect. On 5M seeds 1–4, DUEL with four members differs from the single-policy duel without founding by -0.0023, $+0.0083$, $+0.0013$ and -0.0024 (mean $+0.0012$, paired $t = 0.49$). Across $K = 1, 2, 4, 8, 16$ the mean pass@1 is 0.0735–0.0747, every K is above GRPO on every seed, and in only one run with $K > 1$ do the members differ from one another at $p < 0.05$ (permutation test, $p = 0.034$). After founding, the members differ by a relative parameter offset of 0.0018, so the cross-member pairs contrast near-copies of one policy and approximate that policy's group of 16. A determinant diversity term (Parker-Holder et al., 2020) at weight 0.1 makes the members differ on all four seeds (permutation $p \leq 0.041$, against one of four seeds without it) and changes their mean accuracy by -0.0021 against a control with the same founding noise ($t = -0.85$); the pass@1 of the members' token-level mixture changes by -0.0028 (App. Table 9; App. B.2 gives a theory of this).

6 DISCUSSION

In engine-credit chess the pair term raises the score against GRPO with copies of θ as members (App. Table 4), so the gain needs no learned member diversity. On puzzles, accuracy against GRPO is flat in the number of members from one to sixteen, and a diversity term that makes the members distinct does not raise it (§5.2). In both beds the update compares a better and a worse continuation of one state. On puzzles, a failure shares its correct opening moves with a success. Summed log-probabilities cancel their direct contribution to the pair score (Eq. (5)), and the ablations locate part of the gain over per-token-mean GRPO in that aggregation. The objective needs only a verifier or an outcome that orders two continuations, so it applies directly to language-model post-training, where correct and incorrect answers often share a long prefix; evaluating the full method there, with the population and the self-generated curriculum, is left to future work.

7 LIMITATIONS

Total chess compute is not matched (§5.1). The full objective of App. C, Eq. (15) (per-member GRPO, reward-gap-weighted cross-member DPO, founders) has not yet been trained to a working state in chess; the headline chess result trains a subset (App. Table 2).

8 CONCLUSION

We introduced KAIKA and evaluated its effectiveness for language-model post-training. In our chess experiments, KAIKA improves on the distilled base by approximately 156 Elo, compared with approximately 79 Elo for the recipe GRPO baseline, the only GRPO rated against the base, and outperforms that baseline in all 10 training-seed comparisons at $T = 0.7$ under the reported evaluation protocol. The advantage persists under controls that remove or alter several of the mechanisms initially proposed to explain it, without establishing which components account for the gain. We therefore report an empirical improvement without claiming a settled explanation. Summing log-probabilities over each attempt accounts for part of the puzzle gain (§5.2), and at $T = 0.7$ the pair term accounts for part of the chess margin over GRPO ($+0.0458$ with learning members, $+0.0689$ with copies of θ; at $T = 0.3$ it is not distinguishable from zero). The remaining gain, and whether distinct members contribute to either result, remain unresolved. Following Shazeer (2020), we provisionally attribute our success to divine benevolence.

REPRODUCIBILITY STATEMENT

An anonymous reproduction repository (MIT licence) provides one command per headline chess result: KAIKA training; matched GRPO training, with RLOO or REINFORCE selected by one flag; the base-versus-policy and GRPO-versus-KAIKA duels at $T{=}0.7$ and $T{=}0.3$ with $n{=}1024$ games; and the Stockfish ladder; the archive-bed results each have one command as well (App. G.1). It pins the public base model `mlabonne/LFM2.5-230M-Chess` by SHA-256 hash, Stockfish 18 and a locked requirements file for Python 3.12, and includes a CPU smoke test. Its README lists times measured on one H100 GPU: 300 KAIKA steps 8,521 s, 300 GRPO steps 8,667 s, one $n{=}1024$ duel 246 s and the ladder 164 s; the paper's own training runs are timed in App. Table 29. On H100 GPUs, the released code reproduces 51-step KAIKA and GRPO chess training runs bit for bit against the code that produced the paper's results: in two reference runs and one release run, each started from empty compiler caches, the final weights are identical, as is every captured training-state tensor of the KAIKA runs, and only logged diagnostics differ. `PARITY.md` in the repository lists the hashes and software versions; parity across GPU models is not claimed.

The archive and kernel studies provide seeded runners, per-seed results and timestamped designs and analyses (App. G.1). Neural-game tables are reconstructed from retained result files; the original language-model slate omitted trained weights. The head-to-head policies come from later checkpoint-saving repetitions and differ from the original slate. App. F.1 gives the headline comparison's per-seed scores and evaluation settings, and App. Table 29 its compute; App. E.2 gives the training protocol and settings (App. Table 16). Numerical reconstruction and replay of a historical trained policy are distinct checks. Recipes and protocol details appear in the corresponding appendix sections.

AI USE STATEMENT

We used generative AI tools to assist with formalising the conceptual framework, proposing and refining hypotheses, designing and reviewing experiments, implementing methods and experimental software, and analysing and interpreting results. This included code for simulation environments and synthetic-data generation. AI tools also assisted with literature retrieval and synthesis, manuscript drafting and editing, reference formatting, and code for figures and tables. Reported measurements come from executable experiments and retained result files. Checks of AI-assisted work included executable tests, comparisons against reference calculations, reconstruction of numerical tables from result files, and primary-source citation checks. The authors take responsibility for the final text, claims and artifacts, including material produced with AI assistance.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.

Jeffrey Amico, Gabriel Passamani Andrade, John Donaghy, Ben Fielding, Tristin Forbus, Harry Grieve, Semih Kara, Jari Kolehmainen, Yihua Lou, Christopher Nies, et al. Sharing is caring: Efficient LM post-training with collective RL experience sharing. *arXiv preprint arXiv:2509.08721*, 2025.

AoE4 World. API – AoE4 World. https://aoe4world.com/api, 2026.

W. Ross Ashby. *An Introduction to Cybernetics*. Chapman & Hall, London, 1956.

Hui Bai and Ran Cheng. Generalized population-based training for hyperparameter optimization in reinforcement learning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2024.

David Balduzzi, Karl Tuyls, Julien Pérolat, and Thore Graepel. Re-evaluating evaluation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

David Balduzzi, Marta Garnelo, Yoram Bachrach, Wojciech M. Czarnecki, Julien Pérolat, Max Jaderberg, and Thore Graepel. Open-ended learning in symmetric zero-sum games. In *International Conference on Machine Learning (ICML)*, 2019.

Quentin Bertrand, Wojciech Marian Czarnecki, and Gauthier Gidel. On the limitations of Elo: Real-world games are transitive, not additive. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2023.

Émile Borel. The theory of play and integral equations with skew symmetric kernels. *Econometrica*, 21(1):97–100, 1953. Translation of the 1921 original.

Herbie Bradley, Andrew Dai, Hannah Teufel, Jenny Zhang, Koen Oostermeijer, Marco Bellagente, Jeff Clune, Kenneth Stanley, Grégory Schott, and Joel Lehman. Quality-diversity through AI feedback. In *International Conference on Learning Representations (ICLR)*, 2024.

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. doi: 10.2307/2334029.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. *arXiv preprint arXiv:2302.02662*, 2023.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Singh Bedi, and Mengdi Wang. MaxMin-RLHF: Alignment with diverse human preferences. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 6116–6135, 2024. URL https://proceedings.mlr.press/v235/chakraborty24b.html.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30 (NeurIPS)*, 2017.

Rémi Coulom. Whole-history rating: A Bayesian rating system for players of time-varying strength. In *Computers and Games (CG 2008)*, Lecture Notes in Computer Science, 2008.

Antoine Cully. Multi-emitter MAP-elites: Improving quality, diversity and convergence speed with heterogeneous sets of emitters. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2021.

Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. Robots that can adapt like animals. *Nature*, 521(7553):503–507, 2015. doi: 10.1038/nature14422.

Wojciech M. Czarnecki, Gauthier Gidel, Brendan Tracey, Karl Tuyls, Shayegan Omidshafiei, David Balduzzi, and Max Jaderberg. Real world games look like spinning tops. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Pierre Dangauthier, Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill through time: Revisiting the history of chess. In *Advances in Neural Information Processing Systems 20*, pp. 337–344, 2007.

Roger R. Davidson. On extending the Bradley–Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329), 1970. doi: 10.1080/01621459.1970.10481082.

Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. *arXiv preprint arXiv:2505.18830*, 2025.

Li Ding, Jenny Zhang, Jeff Clune, Lee Spector, and Joel Lehman. Quality diversity through human feedback: Towards open-ended diversity-driven optimization. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

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

Maxence Faldor, Robert Tjarko Lange, and Antoine Cully. Discovering quality-diversity algorithms via meta-black-box optimization. *arXiv preprint arXiv:2502.02190*, 2025.

Matthew C. Fontaine, Julian Togelius, Stefanos Nikolaidis, and Amy K. Hoover. Covariance matrix adaptation for the rapid illumination of behavior space. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, 2020.

Roberto Gallotta, Antonios Liapis, and Georgios N. Yannakakis. Dynamic quality-diversity search. *arXiv preprint arXiv:2404.05769*, 2024.

Mark E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rStar-Math: Small LLMs can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. TextArena. *arXiv preprint arXiv:2504.11442*, 2025.

Reinhard Heckel, Mahdi Soltanolkotabi, and Christos Thramboulidis. Asymmetric prompt weighting for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2602.11128*, 2026.

Ralf Herbrich, Tom Minka, and Thore Graepel. TrueSkill: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19*, 2006.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. *arXiv preprint arXiv:2006.14171*, 2020.

Hugging Face. TRL – transformers reinforcement learning. https://huggingface.co/docs/trl/index, 2026.

Dongyoon Hwang, Hojoon Lee, Jaegul Choo, Dongmin Park, and Jongho Park. Can large language models develop strategic reasoning? post-training insights from learning chess. *arXiv preprint arXiv:2507.00726*, 2025.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. Population based training of neural networks, 2017.

Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1):203–244, 2011. arXiv:0811.1067.

Minbeom Kim, Kang-il Lee, Seongho Joo, Hwaran Lee, Thibaut Thonet, and Kyomin Jung. Drift: Decoding-time personalized alignments with implicit user preferences. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, 2025.

Hannah Rose Kirk, Alexander Whitefield, Paul Röttger, Andrew Bean, Katerina Margatina, Juan Ciro, Rafael Mosquera, Max Bartolo, Adina Williams, He He, Bertie Vidgen, and Scott A. Hale. The PRISM alignment dataset: What participatory, representative and individualised human feedback reveals about the subjective and multicultural alignment of large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS), Datasets and Benchmarks Track*, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Thomas Kleine Buening and Aadirupa Saha. ANACONDA: an improved dynamic regret algorithm for adaptive non-stationary dueling bandits. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 3854–3878, 2023.

Patrick Kolpaczki, Viktor Bengs, and Eyke Hüllermeier. Non-stationary dueling bandits. *arXiv preprint arXiv:2202.00935*, 2022.

Takuya Koriyama and Tengyuan Liang. A Markov chain approach to preference alignment. *arXiv preprint arXiv:2606.22652*, 2026.

Maxime Labonne. ChessLFM: 2000 Elo in 230M params. Substack, https://maximelabonne.substack.com/p/chesslfm-2000-elo-in-230m-params; model https://hf.co/mlabonne/LFM2.5-230M-Chess, 2026.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-DPO: Step-wise preference optimization for long-chain reasoning of LLMs. *arXiv preprint arXiv:2406.18629*, 2024.

Jack Lanchantin, Angelica Chen, Janice Lan, Xian Li, Swarnadeep Saha, Tianlu Wang, Jing Xu, Ping Yu, Weizhe Yuan, Jason E. Weston, Sainbayar Sukhbaatar, and Ilia Kulikov. Bridging offline and online reinforcement learning for LLMs. *arXiv preprint arXiv:2506.21495*, 2025.

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 (NeurIPS)*, 2017.

Allison Lau, Younwoo Choi, Vahid Balazadeh, Keertana Chidambaram, Vasilis Syrgkanis, and Rahul G. Krishnan. Personalized adaptation via in-context preference learning. *arXiv preprint arXiv:2410.14001*, 2024.

Gang Li, Ming Lin, Tomer Galanti, Zhengzhong Tu, and Tianbao Yang. DisCO: Reinforcing large reasoning models with discriminative constrained optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Shuyue Stella Li, Bhargavi Paranjape, Kerem Oktar, Zhongyao Ma, Gelin Zhou, Lin Guan, Na Zhang, Sem Park, Lin Chen, Diyi Yang, Yulia Tsvetkov, and Asli Celikyilmaz. HorizonBench: Long-horizon personalization with evolving preferences. *arXiv preprint arXiv:2604.17283*, 2026.

Xinyu Li, Ruiyang Zhou, Zachary C. Lipton, and Liu Leqi. Personalized language modeling from personalized human feedback. *arXiv preprint arXiv:2402.05133*, 2024.

Lichess. lichess.org open database. https://database.lichess.org/, 2026a.

Lichess. Chess opening names. https://github.com/lichess-org/chess-openings, 2026b.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025a.

Siqi Liu, Marc Lanctot, Luke Marris, and Nicolas Heess. Simplex neural population learning: Any-mixture Bayes-optimality in symmetric zero-sum games. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 13793–13806. PMLR, 2022a. URL https://proceedings.mlr.press/v162/liu22h.html. arXiv:2205.15879.

Siqi Liu, Luke Marris, Daniel Hennes, Josh Merel, Nicolas Heess, and Thore Graepel. NeuPL: Neural population learning. In *International Conference on Learning Representations (ICLR)*, 2022b. URL https://openreview.net/forum?id=MIX3fJkl_1. arXiv:2202.07415.

Xiangyu Liu, Hangtian Jia, Ying Wen, Yujing Hu, Yingfeng Chen, Changjie Fan, Zhipeng Hu, and Yaodong Yang. Towards unifying behavioral and response diversity for open-ended learning in zero-sum games. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 941–952, 2021.

Xiaoze Liu, Dhananjay Ram, Yuting Zhang, Zhaoyang Zhang, Wei Xia, and Stefano Soatto. Experience sharing in mutual reinforcement learning for heterogeneous language models. *arXiv preprint arXiv:2605.07244*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems 30*, 2017.

Chris Lu, Samuel Holt, Claudio Fanconi, Alex J. Chan, Jakob Foerster, Mihaela van der Schaar, and Robert Tjarko Lange. Discovering preference optimization algorithms with and for large language models. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Yihong Luo, Tianyang Hu, and Jing Tang. Reinforcing diffusion models by direct group preference optimization. In *International Conference on Learning Representations*, 2026.

Luke Marris, Paul Muller, Marc Lanctot, Karl Tuyls, and Thore Graepel. Multi-agent training beyond zero-sum with correlated equilibrium meta-solvers. In *International Conference on Machine Learning (ICML)*, 2021. arXiv:2106.09435.

Stephen McAleer, John Lanier, Kevin A. Wang, Pierre Baldi, and Roy Fox. XDO: A double oracle algorithm for extensive-form games. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. arXiv:2103.06426.

Michael D. McKay, Richard J. Beckman, and William J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, 1979. doi: 10.1080/00401706.1979.10489755.

Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, et al. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces. *arXiv preprint arXiv:2601.11868*, 2026. Benchmark and Harbor harness at https://www.tbench.ai.

Jean-Baptiste Mouret and Jeff Clune. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*, 2015.

Youssef Mroueh. Reinforcement learning with verifiable rewards: GRPO's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 36743–36768. PMLR, 2024. URL https://proceedings.mlr.press/v235/munos24a.html. arXiv:2312.00886.

Sewoong Oh and Devavrat Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems 27*, 2014.

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

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*, 2022.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: a review. *Neural Networks*, 113:54–71, 2019.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population based reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

pavelslab-nyu. Chess-Pretrain-Models. Hugging Face Hub, https://huggingface.co/pavelslab-nyu/Chess-Pretrain-Models, revision 23c484d, 2026.

Nicolas Perez-Nieves, Yaodong Yang, Oliver Slumbers, David H. Mguni, Ying Wen, and Jun Wang. Modelling behavioural diversity for learning in open-ended games. In *International Conference on Machine Learning (ICML)*, pp. 8514–8524, 2021.

Justin K. Pugh, Lisa B. Soros, and Kenneth O. Stanley. Quality diversity: A new frontier for evolutionary computation. *Frontiers in Robotics and AI*, 3:40, 2016. doi: 10.3389/frobt.2016.00040.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

Brian Roberson. The Colonel Blotto game. *Economic Theory*, 29(1):1–24, 2006. doi: 10.1007/s00199-005-0071-5.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems 32*, 2019.

Christopher D. Rosin and Richard K. Belew. New methods for competitive coevolution. *Evolutionary Computation*, 5(1):1–29, 1997.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

Aadirupa Saha and Shubham Gupta. Optimal and efficient dynamic regret algorithms for non-stationary dueling bandits. In *International Conference on Machine Learning (ICML)*, pp. 19027–19049, 2022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Jingyan Shen, Ang Li, Salman Rahman, Yifan Sun, Micah Goldblum, Matus Telgarsky, and Pavel Izmailov. Understanding reasoning from pretraining to post-training, 2026.

Idan Shenfeld, Felix Faltings, Pulkit Agrawal, and Aldo Pacchiano. Language model personalization via reward factorization. *arXiv preprint arXiv:2503.06358*, 2025.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404.

Seongho Son, William Bankes, Sayak Ray Chowdhury, Brooks Paige, and Ilija Bogunovic. Right now, wrong then: Non-stationary direct preference optimization under preference drift. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Guangxin Song, Xing Fang, Mingmin Jin, Jing Wang, Bokang Wang, Zhentao Song, Junjie Bai, and Jianbo Zhu. SSR-GRPO: Integrating supervision and semantic IDs into reinforcement learning for dense retrieval in e-commerce. *arXiv preprint arXiv:2608.19595*, 2026.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, Tim Althoff, and Yejin Choi. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 47345–47377. PMLR, 2024. URL https://proceedings.mlr.press/v235/swamy24a.html. arXiv:2401.04056.

Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. True knowledge comes from practice: Aligning LLMs with embodied environments via reinforcement learning. *arXiv preprint arXiv:2401.14151*, 2024.

The Stockfish developers. Stockfish: a free and strong UCI chess engine. https://stockfishchess.org, 2025.

Muhammad Umer, Muhammad Ahmed Mohsin, Ahsan Bilal, Arslan Chaudhry, Andreas Haupt, Sanmi Koyejo, Emily Fox, and John M. Cioffi. General preference reinforcement learning. *arXiv preprint arXiv:2605.18721*, 2026.

Oriol Vinyals, Igor Babuschkin, Wojciech M. Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H. Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019. doi: 10.1038/s41586-019-1724-z.

John von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, 1928. doi: 10.1007/BF01448847.

Ren-Jian Wang, Ke Xue, Yutong Wang, Peng Yang, Haobo Fu, Qiang Fu, and Chao Qian. Diversity from human feedback. *arXiv preprint arXiv:2310.06648*, 2023.

Yibin Wang, Zhimin Li, Yuhang Zang, Yujie Zhou, Jiazi Bu, Chunyu Wang, Qinglin Lu, Cheng Jin, and Jiaqi Wang. Pref-GRPO: Pairwise preference reward-based GRPO for stable text-to-image reinforcement learning. *arXiv preprint arXiv:2508.20751*, 2025.

Zhichao Wang. GIFT: Group-relative implicit fine-tuning integrates GRPO with DPO and UNA. *arXiv preprint arXiv:2510.23868*, 2025.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992. doi: 10.1007/BF00992696.

Fang Wu, Xu Huang, Weihao Xuan, Zhiwei Zhang, Yijia Xiao, Guancheng Wan, Xiaomin Li, Bing Hu, Peng Xia, Jure Leskovec, and Yejin Choi. Multiplayer Nash preference optimization. In *International Conference on Learning Representations (ICLR)*, pp. 61004–61024, 2026.

Huasen Wu and Xin Liu. Double thompson sampling for dueling bandits. In *Advances in Neural Information Processing Systems 29 (NeurIPS)*, 2016.

Yihong Wu, Liheng Ma, Lei Ding, Muzhi Li, Xinyu Wang, Kejia Chen, Zhan Su, Zhanguang Zhang, Chenyang Huang, Yingxue Zhang, Mark Coates, and Jian-Yun Nie. It takes two: Your GRPO is secretly DPO. *arXiv preprint arXiv:2510.00977*, 2025.

Qiaosi Yi, Shuai Li, Rongyuan Wu, Lingchen Sun, Zhengqiang Zhang, and Lei Zhang. GDPO-SR: Group direct preference optimization for one-step generative image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2177–2187, 2026.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The K-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012. doi: 10.1016/j.jcss.2011.12.028.

Tom Zahavy, Vivek Veeriah, Shaobo Hou, Kevin Waugh, Matthew Lai, Edouard Leurent, Nenad Tomasev, Lisa Schut, Demis Hassabis, and Satinder Singh. Diversifying AI: Towards creative chess with AlphaZero, 2024. URL https://arxiv.org/abs/2308.09175.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS), Datasets and Benchmarks Track*, 2023.

Taojie Zhu, Dongyang Xu, Ding Zou, Sen Zhao, Qiaobo Hao, Zhiguo Yang, and Yonghong He. Bridging SFT and RL: Dynamic policy optimization for robust reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026. arXiv:2604.08926.

Thomas P. Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. PersonalLLM: Tailoring LLMs to individual preferences. *arXiv preprint arXiv:2409.20296*, 2024.

A FURTHER CHESS, POPULATION AND GAME-BED RESULTS

A.1 THE POPULATION INSTANCE

Population. Slot 0 holds a shared policy θ, and the members are low-rank offsets $\theta_i = \theta + PC^{(i)}$ of it, so the backbone is shared and every update of θ moves every member. The member basis P is re-estimated from the members' gradient Gram matrix every 50 steps. No past policy is kept as an opponent.

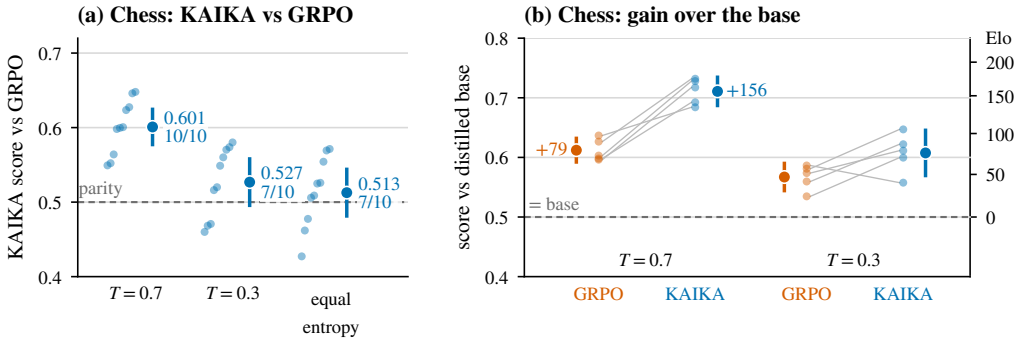

Figure 4: Scores (dots: seeds; bar: mean and 95% t interval; dashed: 0.5). (a) KAIKA's θ vs GRPO, 10 seeds, 1024 games each; the third column cools GRPO to KAIKA's entropy. (b) Each arm vs its distilled base, five seeds; right axis in Elo. What KAIKA trains here: Table 2.

Standings and replacement. Two steps of the loop in §3 act on the population alone. After play, **Compare.** Outcomes update Elo standings. After training, **Replace.** Elo screens the members; each eligible loser restarts as θ plus Gaussian jitter of σ_{mut} per coordinate, with fresh optimizer state.

Populations in post-training. Populations are already standard in post-training: GRPO scores each response against a group of sampled responses, and league training such as AlphaStar (Vinyals et al., 2019) keeps current and past agents as opponents. GRPO's group is discarded after each update, so it carries one policy between updates. KAIKA keeps a population of Elo-screened variants around a shared policy: it updates standings from outcomes and replaces low-rated members with jittered copies of the shared policy.

Full objective and instances. In the full objective, each of KAIKA's K members trains on its own rollouts with a clipped GRPO objective and, through a DPO term weighted by reward gaps, on how it ranks the other members' responses; founders are forked from one GRPO-trained parent by reweighting its group (App. C). The chess and Nim results come from a shared-policy instance whose members are low-rank offsets of one policy (§3); Table 2 lists the terms each result trains. Chess and the language-model games evaluate θ, while neural Blotto (Roberson, 2006) evaluates an LP mixture of the members. An earlier, gradient-free KAIKA archive is evaluated in App. H.

Contributions stated for the population instance. Table 3 maps contributions (2)–(4) to experimental objectives. **(1) A population objective.** Each member keeps an own-policy GRPO group, learns from the other members' responses through reward-gap-weighted DPO scored under itself, and descends from founders forked from one GRPO-trained parent; we state what these comparisons carry, since the gaps of N scalar rewards are fixed by $N - 1$ numbers (App. C). **(2) Shared-policy learning.** The shared-policy instance adds a same-position winner-vs-loser DPO term to a shared policy's batch policy gradient. **(3) Game beds.** On Nim trained from outcomes alone, and on Goofspiel and Blotto, we compare KAIKA with single-policy training (§A.5). **(4) Reinforcement learning from game outcomes alone.** On engine-free chess, where the only signal is the game result between policies over legal moves, KAIKA's engine-free arm, whose learned critic is part of KAIKA, loses to its critic-only ablation and beats a 128-row GRPO control on the same outcomes.

Table 2: What each result trains, against Eq. 15. θ: the update on θ ($\mathcal{L}_{\text{batch}}$; "all": the played rows of all eight slots at ratio one). Members: the member update (sign-PG: clipped PG on the sign of each row's credit minus the step mean, not Eq. 10). Pairs: θ-only pairs of Eq. 1 or cross-member pairs. Births: how members are replaced; founders follow Eq. 16. No chess result uses the diversity term ($\lambda_{\text{div}} = 0$ wherever the run configuration states it). Steps: KAIKA/comparator. n/s: not stated in the saved run configuration. [a]Per arm: pooled or same-position pairs, or none (founded arm with shared θ); member update z-scored, pooled or sign-PG.

Result	θ	Members	Pairs	Births	Credit	Steps	Seeds
Chess, engine credit	all	sign-PG	θ, same position	jitter 10^{-3}	engine	300/300	10
Chess, engine-free	all	sign-PG	θ, game/position	jitter 0.03, 10^{-3}	critic	300/300	5, 3
Chess population[a]	all	varies	cross-member	elites, founders	critic	100/300	1, 3
Full objective	own rows	own group	cross-member	founders	n/s	100/300	1
Nim, tic-tac-toe	all	sign-PG	θ, ply	jitter 0.03	outcome	300/300	5
Nim (end), Kuhn	all	sign-PG	θ, ply + NLL	jitter 10^{-3}	outcome	300/300	5
Goofspiel	consensus	n/s	n/s	jitter 10^{-3}, 10^{-2}	outcome	200/200	2
Neural Blotto	n/s	n/s	n/s	two lowest	outcome	200/n/s	3

	results on	games + puzzles	230M chess, 1.5B Nim	5M / 32M chess puzzles			32M puzzles
		GRPO K=1	KAIKA K=8	DUEL K=4	SPEC K=4	UNIFIED K=4	indep. pop. K=4
core	Population of K policies	○	●	●	●	●	●
	Pair term on summed log-probs	○	●	●	●	●	○
	KL anchor to the reference	●	●	●	●	●	●
extensions	Policy-gradient (GRPO) term	●	●	○	●	●	●
	Pairs across members	–	○	●	●	●	–
	Pair weight	–	DPO	Fisher	DPO	DPO	–
	Founders from a GRPO parent	○	○	●	●	○	○
	Shared θ, low-rank members	○	●	○	○	○	○
	Elo standings, jitter restarts	○	●	○	○	○	○
	Coverage term, gated mixing	○	○	○	○	●	○
	Learned move critic	○	◐	○	○	○	○

◐: engine-free games only. –: no pair term.

Figure 5: Components of each version of KAIKA (●: present, ○: absent, half-filled: partial), with GRPO and an independent population of GRPO learners as references. Every version keeps a population, a winner-vs-loser pair term on log-probabilities summed over tokens, and a KL anchor; the other rows vary by version, and the top band names the beds with results.

A.2 WHAT EACH RESULT TRAINS

Table 2 lists, for each result, which parameters train and by which update.

A.3 POPULATION ARMS, ENGINE-FREE CHESS AND GAME-BED SETUP

Population arms. The chess population arms (§A.6) replace the sign update of the members with per-member z-scored policy gradients, add cross-member DPO over pairs pooled per step (Eq. 13), or fork founders from a GRPO-trained parent by reweighting its group (Eq. 16); they replace members by jittered offspring of the elites every 10 steps.

Engine-free chess. Engine-free chess calls no engine: the environment supplies legal moves and outcomes, and the only training signal is the game result between policies. KAIKA's engine-free arm trains the learned critic and the same-game winner-vs-loser pair term, with jitter 0.03 per coordinate unscaled to the parent (newborn norm 77.94 against 2.6); a three-seed variant uses the position-pair term and jitter 10^{-3}. Engine-free chess adds a critic-only ablation, GRPO with KAIKA's learned critic, and KL-anchored GRPO controls of 64 and 128 rows per step given the same outcomes. On Nim both arms run at learning rate 10^{-5} and $\beta = 0.1$, and GRPO has no pair term and no learned critic. On Goofspiel and neural Blotto the comparator is an earlier GRPO implementation, which

Table 3: Experimental objectives and results, with each endpoint named in its row. Neural chess scores evaluate the shared policy. The appendix reader's guide indexes protocols and comparator access.

Objective	Setting / output	Reference	Result
Shared policy	Neural chess / θ	Starting policy; GRPO matched in game budget and engine supervision	Duel Elo vs base $+156/+79$; ladder gains $+165/+88$ (paired $+77$, 10/10); head-to-head 0.6009 at $T = 0.7$
Pair term	Chess factorial / θ	Same recipe without the pair term, copies of θ as members	$+0.0689$; CI $[+0.0291, +0.1088]$; 5/5 seeds
Game outcomes alone	Engine-free chess / θ	Critic-only ablation; 128-row GRPO on the same outcomes	0.3818 vs ablation (0/5); 0.5721 vs GRPO (5/5)
Language-model game	Nim / θ	GRPO on the same outcomes	0.5656 (3/5 seeds)
Goofspiel	Earlier configuration / θ	Earlier GRPO implementation (§4)	First fix 0.7475, 0.1850; second fix (jitter 10^{-2}) 0.7525, 0.6075 (2 seeds)
Mixture	Neural Blotto / exploitability	Earlier GRPO implementation, final policy; random archive	0.194–0.228 vs 0.651–0.888; random 0.051

on Goofspiel differs from KAIKA's configuration in learning rate (3×10^{-6} against 3×10^{-5}), KL weight, warm-up and batch.

Game beds. We post-train Qwen2.5-1.5B-Instruct (Qwen et al., 2025) on the TextArena Nim environment (Guertler et al., 2025) for 300 steps on five seeds, with no solver, engine or reward model. Each policy samples only among the position's legal moves, and each move is credited with its game's outcome. Both arms play 32 self-play games and train 64 policy-gradient rows per step, from two random opening plies (App. E.1). The training logs contain no engine call, solver label or branch row on any step. Two configured settings did not take effect in these runs: the likelihood anchor on the pair term and the sketch-based member gradient split; the members used the mean split of §3. Separate end-state runs on Nim and Kuhn poker train with the likelihood anchor in effect ($\alpha = 0.1$ on the pair winners' NLL), again with the mean split. In Blotto (Roberson, 2006) every pure allocation loses outright to some reply (Borel, 1953), so equilibria are mixed (von Neumann, 1928) and a policy trained to beat its opponents is beaten in turn (Balduzzi et al., 2019); the output is an LP mixture of members, which cannot worsen as its candidates grow (Liu et al., 2021). We post-train Qwen2.5-1.5B-Instruct on 20-troop, 3-field Blotto (231 allocations) for 200 steps and 64 rows per step on 3 seeds. Eight rank-64 members play 32 games per step; the two lowest-rated unfrozen members are replaced every 10 steps, with age 60 protected. A 19-run sweep freezes the trunk; a separate cohort trains it. Sampling and scoring are masked to legal allocations (Huang & Ontañón, 2020); App. F.5 details the LP readouts.

A.4 CHESS ABLATIONS

In the engine-credit arm on the two probed seeds (1234 and 2), the population collapses to one policy: losing slots are refilled with jittered copies of θ 384 and 393 times in 300 steps (a member lives about 5.5 and 5.3 steps), and two members choose the same move as often as one member agrees with itself (0.60 and 0.63 against 0.61–0.63 and 0.62–0.65; base 0.43). Without its KL anchor ($\beta = 0$, no reference model, learning rate 3×10^{-5}), an outcome-credit GRPO control falls below the base on all 5 seeds, by -152 to -17 Elo, and by -201 Elo in the batch-matched control (App. Table 19).

Table 4 separates the pair term from the members. At $T = 0.7$, the winner-vs-loser DPO term adds +0.0689 (95% CI $[+0.0291, +0.1088]$, excluding zero) with exact copies of θ in the member slots and +0.0458 with learning members, on every seed, so the pair term contributes without learned member diversity; the copies without the term score +0.0531 above parity. Learning members in

Table 4: Chess factorial at $T = 0.7$ and $T = 0.3$: each arm's θ against same-seed GRPO (mean, 95% t interval, seeds above 0.5). Seeds 1234, 3 and 4 of the headline recipe ran at an earlier commit than the other arms; its row is the first five seeds of the GRPO row of Table 1. Every paired effect: App. Table 21.

	$T = 0.7$		$T = 0.3$	
Arm (members, pair term)	Score [95% CI]	> 0.5	Score	> 0.5
learning, on (headline)	0.6121 [0.5729, 0.6513]	5/5	0.5314	4/5
learning, off	0.5663 [0.5434, 0.5892]	5/5	0.5376	5/5
copies of θ, on	0.6221 [0.5965, 0.6477]	5/5	0.5395	5/5
copies of θ, off	0.5531 [0.5342, 0.5721]	5/5	0.5333	5/5

place of the copies change the score by -0.0100 ($[-0.0301, +0.0102]$). Seeds 1234, 3 and 4 of the headline recipe ran at an earlier commit; on seeds 2 and 5, where all four arms share one commit, learning minus clone members with the pair term is -0.033 and $+0.009$ at $T = 0.7$ (mean -0.012) and -0.059 and -0.051 at $T = 0.3$ (mean -0.055). Unclamped member gradient norms reach 1.7×10^{16} and 3.3×10^{16} (seeds 1234 and 2; no non-finite value); clamping the member KL at 8 (active on 240 and 252 of 300 steps) changes the score against GRPO at $T = 0.7$ by -0.0220 and $+0.0122$ (App. Table 22).

Engine-free chess. Against its critic-only ablation the engine-free arm scores 0.3818 at $T = 0.7$ and 0.3990 at $T = 0.3$, below parity on all 5 seeds at both temperatures and at equalised entropy; a three-seed variant with the position-pair term and jitter 10^{-3} scores 0.4725 and 0.4691 against that ablation. Against a KL-anchored 128-row GRPO control given the same outcomes the arm scores 0.5721 and 0.5671 (5/5 seeds at both temperatures), and the variant scores 0.6914 and 0.6182 against the base (3/3).

A.5 Game beds

On Nim, KAIKA scores 0.5656 against GRPO at $T = 0.7$ (3/5 seeds above 0.5; $[0.227, 0.904]$) and 0.5719 at $T = 0.3$ (3/5; $[0.193, 0.951]$); 2 of 5 seeds score below 0.5 at both temperatures. On tic-tac-toe, KAIKA scores 0.4680 against GRPO at $T = 0.7$ and 0.5258 at $T = 0.3$, $2/5$ seeds above 0.5 at each. The end-state duels below come from separate training runs with the likelihood anchor in effect ($\alpha = 0.1$ on the pair winners' NLL); the sketch split again read the mean on every step.

End-state duels. The KAIKA shared policy θ plays the final GRPO policy of the same training seed over 1024 games per pairing at $T = 0.7$ and $T = 0.3$ (App. Table 25). Goofspiel, Blotto and Hex have no end-state duel.

Nim. With legal-move scoring and HF decoding, KAIKA scores 0.6598 against GRPO at $T = 0.7$ (4/5 seeds above 0.5; 95% t interval over seeds $[0.479, 0.841]$) and 0.6568 at $T = 0.3$ (4/5; $[0.451, 0.862]$), with 0.000 drawn and 0.000 ended at the ply limit and counted as draws; 1 of 5 seeds score below 0.5 at both temperatures.

Every Kuhn poker policy, the base model included, plays the same action with the jack, queen and king, so Kuhn end-state scores are omitted. On Tak the first update collapses play to the ply cap, and Tak is excluded. Both arms' end-state Nim and Kuhn poker training rollouts ran through a vLLM 0.20.1 top-p defect that sampled near-greedy (App. Table 25); chess, Goofspiel and all duels ran on HF Transformers.

On Goofspiel (Guertler et al., 2025), an earlier KAIKA configuration ran against the earlier GRPO implementation (§4) on 2 seeds at $T = 0.7$. The best of six scripted bidders scores at least 0.85 against every arm, the base model included, so these scores measure relative strength between the arms. With member jitter 10^{-3}, one seed's members became copies of θ and drew every duel from step 99, so no comparison carried signal and nothing was written for the remaining 101 steps; at the live seed the members' uniform mixture scores 0.5125 against its own θ. The seeds' θ score 0.8250 and 0.6775 against GRPO, the second frozen at step 98. A first fix sent a fraction 0.25 of the member duels against a random bidder; it kept both populations alive, and its θ scores 0.7475 and

Table 5: Chess population arms: θ against GRPO (training seed 1234 for every arm seed, so seeds 2 and 3 of the same-position arm are cross-seed pairings) and against the reference population; mean over seeds (seeds above 0.5). Every arm trains 100 steps against GRPO's 300, from game outcomes through the learned critic with no engine credit, and replaces members by jittered offspring of the elites every 10 steps. The learned critic trains on replayed positions rebuilt from the played games, and the pair term pools pairs across steps over a shared trunk. A same-seed rerun of the reference scores 0.5363 against it. Member reads: App. Table 24.

Arm	GRPO, T=0.7	GRPO, T=0.3	reference, T=0.3
per-member z-scored PG, cross-member DPO over step-pooled pairs	0.4609 (0/1)	0.5015 (1/1)	0.4219 (0/1)
per-member z-scored PG, step-pooled DPO, founded ($\eta_f = 24.5$)	0.4629 (0/1)	0.4907 (0/1)	0.4395 (0/1)
shared θ, founded ($\eta_f = 244.7$)	0.4673 (0/1)	0.5054 (1/1)	0.4316 (0/1)
per-member z-scored PG, step-pooled DPO, $\sigma_{\text{mut}} = 0.001$	0.4785 (0/1)	0.5293 (1/1)	0.4619 (0/1)
shared θ, founded ($\eta_f = 24.5$)	0.5254 (1/1)	0.5337 (1/1)	0.4756 (0/1)
shared θ, same-position outcome pairs	0.5658 (3/3)	0.5635 (3/3)	0.5037 (1/3)
reference (shared θ)	0.5522 (1/1)	0.5635 (1/1)	–

0.1850 against GRPO. With jitter 10^{-2} every step writes on both seeds, but the members are birth noise: on both seeds their norms are $0.707\,\sigma\sqrt{D}$ and their pairwise cosines lie within ± 0.0006, so they are nearly orthogonal random draws, and the mixture scores below do not test the population. θ scores 0.7525 and 0.6075, the members' mixture 0.6950 and 0.6275 against it and 0.5375 and 0.4650 against its own θ.

Neural Blotto. A random archive of 48 allocations, the number of members a run creates, reaches exploitability 0.051 under the same LP weighting, below the runs' snapshot-archive LP (0.082–0.155), whose 48 created members contribute 55–88 distinct probe-snapshot distributions; the random archive is a positive control for the LP weighting, and at that count learned selection is not needed. Frozen-trunk held-out exploitability is 0.141–0.315 versus 0.329 for uniform play. On the shared evaluation instrument, the frozen-trunk population's least-exploitable LP mixture scores 0.194–0.228 versus 0.651–0.888 for GRPO's final single policy; an LP over GRPO's three seed policies on the same instrument scores 0.532. Across the frozen-trunk, exogenous-only and trained-trunk runs, the least-exploitable mixture of the 8 live members (the live hull) scores 0.203–0.294, below the fifth percentile (0.333) of random 8-member archives.

On Nim, trained from outcomes alone, the spread over seeds ($[0.227, 0.904]$ at $T = 0.7$) is wider than the difference between the arms. On neural Blotto the members' least-exploitable mixture is compared with GRPO's final single policy, a different kind of object.

A.6 Chess population

Population arms. Table 5 adds per-member terms or founders to an engine-free population. The founded population with θ as a member is omitted: its duels loaded only the frozen trunk.

B Chess puzzles: further results

At 32M, SPEC differs from GRPO by $+0.0020$ $(4/6)$, UNIFIED by $+0.0043$ $(4/6)$ and the independent population by -0.0134 $(0/2)$.

Shared-prefix census. At initialisation, GRPO's push on the shared correct first move is 0.97 of the duel's pair-term push on puzzles with two player moves and 0.71–0.73 on puzzles with three or more, where it is negative on 14–18% of puzzles (5M, 32M and 50M models, seeds 1 and 2 pooled). On decisive puzzles a successful rollout is on average 2.1–2.8 times as long as a failed one (Fig. 6).

Table 6: Greedy pass@1 on 1164 pooled Lichess test puzzles, averaged over members. The GRPO row gives its mean and number of seeds; every other row gives the mean difference from GRPO on the same seeds (Δ), the seeds on which the arm is above GRPO out of the seeds run (w/n), and the paired t statistic. All arms use 16 rollouts per prompt, 32 prompts per step and 200 steps. The KAIKA arms (DUEL, SPEC, UNIFIED) use a KL coefficient of 0.05; GRPO and the Independent population, whose members each run the GRPO update, use 0.001; DUEL and SPEC with $K > 1$ also use a founding step. At 5M, every arm is paired with GRPO runs from the same batch on seeds 1–4, and the GRPO row gives the mean of those four runs; UNIFIED has seeds 1–3, its seed-4 run crashed before step 200. The two rows marked seeds 5–8 pair those seeds with a second GRPO batch; same-seed GRPO runs from different batches differ by up to 0.0069. Independent population was not run at 5M in this batch. Pretrained base: 0.0567 (5M), 0.0919 (32M). 32M seeds: DUEL, GRPO, SPEC and UNIFIED 1–6, Independent population 1–2.

	5M			32M		
Arm	Δ	w/n	t	Δ	w/n	t
GRPO (pass@1, n)	0.0649	4	–	0.1256	6	–
DUEL	+0.0098	4/4	4.52	+0.0127	5/6	2.63
UNIFIED	+0.0061	3/3	19.10	+0.0043	4/6	1.17
SPEC	+0.0047	4/4	4.60	+0.0020	4/6	0.76
DUEL, seeds 5–8	+0.0030	4/4	1.83	–	–	–
SPEC, seeds 5–8	+0.0020	3/4	1.02	–	–	–
Independent population	–	–	–	−0.0134	0/2	–

Table 7: Pair-weight sweep: DUEL minus GRPO (0.1203) at 32M, seed 1, as a function of β. The second row evaluates the weight $E_{ab}|\mathcal{P}|$ of Eq. (3) at a one-nat margin.

β	0.03	0.1	0.3	1.0	3.0
Δ pass@1	+0.0122	+0.0036	−0.0056	−0.0078	−0.0168
$E_{ab}\|\mathcal{P}\|$ at 1 nat	0.985	0.951	0.861	0.602	0.161

Retain term. The retain term $\mathcal{L}_k^{\text{ret}}$ of Eq. (6) is a length-weighted reference regulariser. With $|y|$ the number of player tokens of y, its gradient is

$$\nabla\mathcal{L}_k^{\text{ret}} = \frac{\lambda}{G}\sum_{i:\,o(i)=k}\big(\bar{\ell}_k(y_i) - \bar{\ell}_{\text{ref}}(y_i)\big)\nabla\bar{\ell}_k(y_i) = \frac{\lambda}{G}\sum_{i:\,o(i)=k}\frac{\ell_k(y_i) - \ell_{\text{ref}}(y_i)}{|y_i|^2}\nabla\ell_k(y_i),$$

the gradient of $\frac{\lambda}{2G}\sum_{i:\,o(i)=k}\big(\bar{\ell}_k(y_i) - \bar{\ell}_{\text{ref}}(y_i)\big)^2$ with the rollouts held fixed. With summed log-probabilities on both factors the summand would be the score-function estimate of $\lambda\,\nabla\mathrm{KL}(\pi_k \,\|\, \pi_{\text{ref}})$ over whole rollouts; the per-token means divide each rollout's term by $|y|^2$, so shorter rollouts are held closer to the reference. The KL term of Eq. (2) has the same form, with λ_{G} in place of λ and the group of N in place of the member's G rollouts.

Per-token mean variant. The per-token mean variant replaces only the log-probabilities in the pair loss of Eq. (3),

$$\frac{\ell_k(y_a) - \ell_k(y_b)}{c} \;\longrightarrow\; \bar{\ell}_k(y_a) - \bar{\ell}_k(y_b), \tag{7}$$

so that $\mathcal{L}_k^{\text{duel}} = -\sum_{(a,b)\in\mathcal{P}}\mathrm{sg}(E_{ab})\big(\bar{\ell}_k(y_a) - \bar{\ell}_k(y_b)\big)$. The weight E_{ab} and σ_{ab} are still computed from the summed margins s_k and are identical in both variants. The swap removes the cancellation of Eq. (5).

Table 7 varies β, which moves only the pair weight. On 32M seed 1 the difference from GRPO falls from +0.0122 at $\beta = 0.03$ and +0.0036 at the default $\beta = 0.1$ to −0.0056, −0.0078 and −0.0168 at $\beta = 0.3$, 1 and 3.

Table 8 splits the puzzles by the number of player moves. At 32M, DUEL minus GRPO is −0.0039 on puzzles with at most two player moves and +0.0379 on puzzles with three or more; at 5M it is +0.0057 and +0.0127. Within rating bin, the pretrained 32M model is above the pretrained 5M

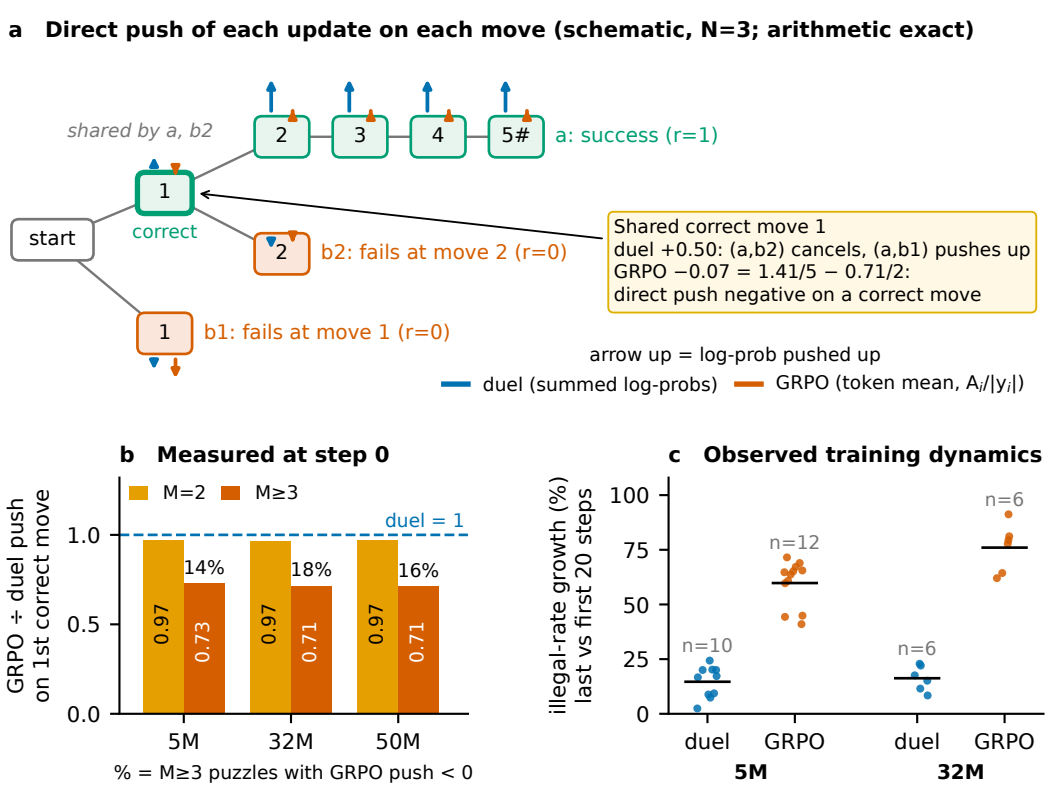

Figure 6: How shared moves enter each update. (a) Schematic of one mate puzzle with three rollouts. Rollout a succeeds in 5 moves. Rollout b1 fails at move 1. Rollout b2 plays the same correct move 1 and fails at move 2. Arrows give the sign and size of each update's direct push on each move's log-probability. For the duel (summed log-probabilities, pair contrast), the (a, b2) pair's direct contribution to the shared move 1 cancels exactly and the (a, b1) pair pushes it up ($+0.50$). For GRPO (per-token mean, weight $A_i/|y_i|$ with advantages $+1.41$, -0.71, -0.71), move 1 receives $+1.41/5 - 0.71/2 = -0.07$: its direct push is negative although it is correct. The lengths are illustrative ($|a|/|b2| = 2.5$, inside the measured 2.1–2.8 success/failure length ratio) and one token per move is assumed. (b) Measured at initialisation (5M, 32M, 50M models, seeds 1 and 2 pooled): GRPO's direct push on the first correct move divided by the duel's is 0.97 on puzzles with two player moves and 0.71–0.73 with three or more, and it is negative on 14–18% of the latter (9/65, 23/127, 31/191). (c) Growth over 200 training steps in the per-decision illegal-move rate (App. Table 12), last 20 vs first 20 steps, one dot per run, bar: mean. 5M: duel $+2.4$ to $+24.3\%$ ($n = 10$), GRPO $+41.0$ to $+71.5\%$ ($n = 12$). 32M: duel $+8.4$ to $+22.9\%$ ($n = 6$), GRPO $+62.0$ to $+91.2\%$ ($n = 6$).

model by $+0.049$ on puzzles with three or more player moves and by $+0.019$ on the rest (permutation $p = 0.064$), so a gain from model size splits the same way. In an fp32 CPU replicate of 5M seeds 1–4, 6 and 7, the aggregation term's effect (summed minus per-token scores) is $+0.0057$ with three or more player moves against $+0.0003$ on the rest (within-bin permutation $p = 0.33$).

For the duel itself, $\lambda = 0.01$ gives $+0.0059$ at 5M (seeds 1–4, against $+0.0098$ for $\lambda = 0.05$) and $+0.0135$ at 32M (one run, against $+0.0036$ and $+0.0066$ for $\lambda = 0.05$ in two runs). An fp32 CPU replicate at 5M (single policy, KL coefficient 0.001, seeds 1–4, 6 and 7) gives summed scores minus per-token-mean GRPO $+0.0097$ and summed minus per-token scores $+0.0029$, each on 6/6 seeds. With per-token scores the duel is above GRPO by $+0.0069$ (6/6), and with per-token scores and a constant pair weight ($\beta = 0$) by $+0.0056$ (4/6, two seeds at 0.0000); the two differ by $+0.0013$ (4/6), so the split between the pair weight and the rest of the per-token effect is unresolved. On every seed the illegal-move growth is lowest with summed scores, higher with per-token scores and highest with the constant pair weight, and with summed scores it is below GRPO's.

Table 10 lists every cell of a learning-rate and KL-coefficient sweep of the KAIKA arms.

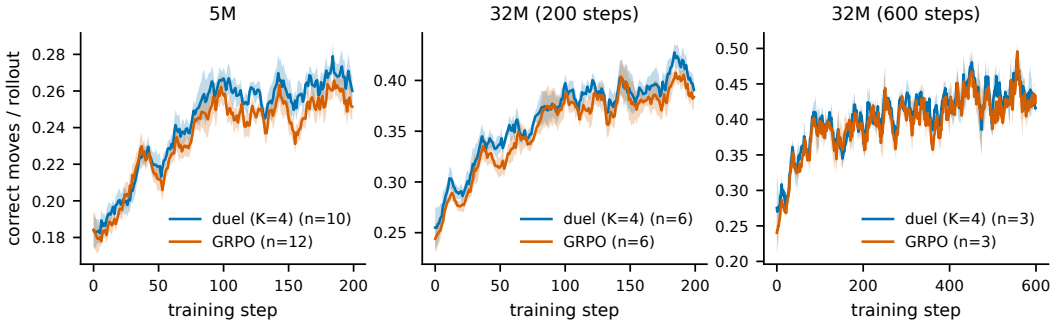

Figure 7: Correct player moves per training rollout (legal and on the solution line), DUEL ($K = 4$) and GRPO; smoothing, bands and runs as in Fig. 2.

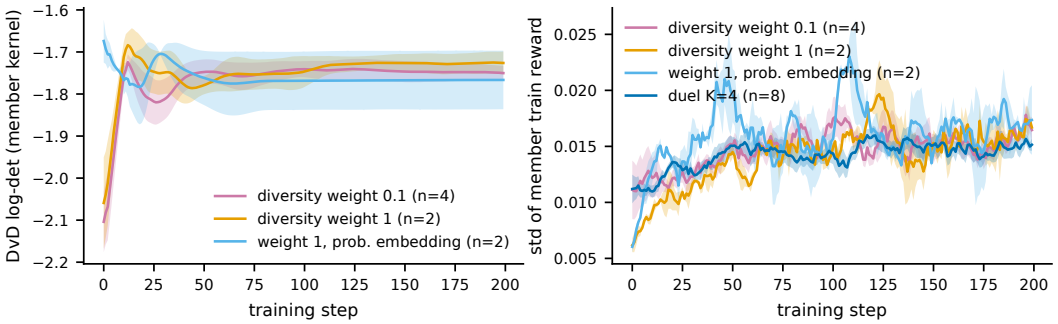

Figure 8: Member spread under a diversity term (Parker-Holder et al., 2020): (a) log-determinant of the members' behavioural kernel, (b) standard deviation of the training solve rate across members, for three diversity settings (weight 0.1: four seeds; the others: two seeds each) and DUEL with $K = 4$ (eight seeds, panel b).

Summed aggregation added to GRPO. Table 11 compares DUEL with GRPO and with the ablation GRPO+sum (GRPO + summed aggregation), which sums the token log-probabilities with a constant normaliser, as Dr. GRPO does. GRPO+sum is above published GRPO by $+0.0039$ at 5M (3/4 seeds) and $+0.0089$ at 32M (5/6, seed-level $t = 3.40$, df 5), and on 32M seeds 3–6 its served member alone solves 156 puzzle–seed pairs against 100 for GRPO alone ($p = 0.00056$). Against GRPO+sum, DUEL is higher by $+0.0059$ at 5M (3/4 seeds, $t = 2.66$, $p = 0.076$) and by $+0.0039$ at 32M (3/6, $t = 1.13$, df 5); on 32M seeds 3–6 the served members give 159 against 127 ($p = 0.067$).

Illegal-move growth by component. The pair loss, the summed scores and the stronger anchor each lower the growth: on 5M seeds 1 and 2, at GRPO's anchor the summed single-policy duel grows by $+25.0\%$ and $+38.9\%$ against GRPO's $+41.0\%$ and $+64.7\%$, per-token means raise it to $+35.9\%$ and $+46.7\%$, and the duel at $\lambda = 0.05$ grows by $+7.4\%$ and $+20.0\%$; this last contrast also changes the number of members from one to four.

B.1 CHESS-PUZZLE PROTOCOL AND COMPUTE

Models. Both base models are publicly released pretrained chess language models (`Chess-Pretrain-Models` on the Hugging Face Hub, revision `23c484d`, pavelslab-nyu, 2026), used without supervised fine-tuning. Each is a Qwen3-architecture decoder with tied input and output embeddings, an 81-token vocabulary and a 1,024-token context. The 5M model has 6 layers, width 256, feed-forward width 768 and 2 attention heads (5,158,144 parameters; 2.1B pretraining tokens); the 32M model has 8 layers, width 512, feed-forward width 1,536 and 8 query and 4 key–value heads (31,550,976 parameters; 1.0B pretraining tokens). The model cards give the token counts but do not describe the pretraining games. The SHA-256 digests of the `model.safetensors` files are

Table 8: DUEL minus GRPO by the number M of player moves in the puzzle's solution, pooled over the seeds with per-puzzle outputs (32M: seeds 3–6; 5M: seeds 1, 2, 4, 5). Δ: difference in pass@1 averaged over members. b/c: puzzle–seed pairs solved by the served DUEL member only and by GRPO only; p: exact two-sided binomial test on b/c, which treats puzzles as independent within a seed.

Size	Moves	Puzzles	Δ	b/c	p
32M	$M \leq 2$	523	-0.0039	$83/91$	0.60
	$M \geq 3$	641	$+0.0379$	$141/45$	1×10^{-12}
5M	$M \leq 2$	523	$+0.0057$	$65/54$	0.36
	$M \geq 3$	641	$+0.0127$	$59/27$	7.3×10^{-4}

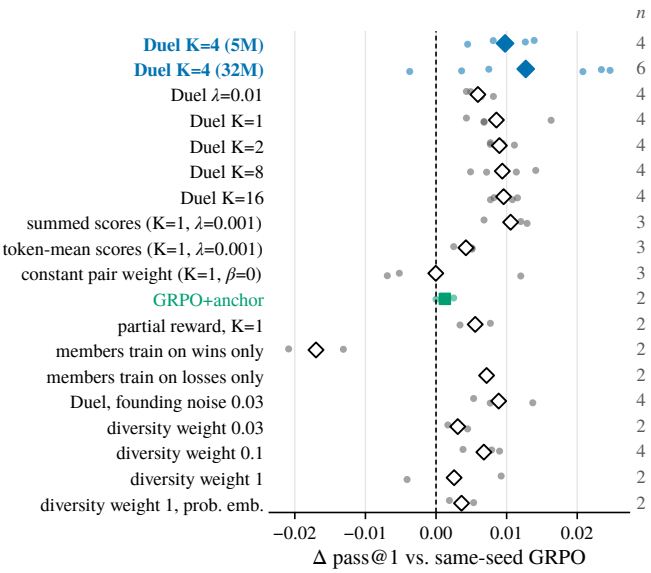

Figure 9: Final pass@1 of each ablation arm minus GRPO from the same batch and seed (dots: seeds; diamond: mean). 5M rows use seeds 1–4 of one batch, where DUEL is above GRPO by $+0.0098$; App. Table 6 adds seeds 5–8 paired with a second GRPO batch ($+0.0030$). The 32M row uses seeds 1–6, as App. Table 6. Win-only and loss-only arms: founding noise 0.03, KL coefficient 0.01; diversity-term arms and the row above them: founding noise 0.03; the partial-reward row subtracts GRPO trained on the partial reward; the constant-pair-weight row uses per-token scores and KL coefficient 0.001.

```
5M:  d41ebd6a6310fec89fab1e285f46b74d6d991ad5393155c314910a0e7ed59a0d
32M: cc8b38f48a7e9d1c2a35cd3d537b246d7e598bfe4833a971b526ce735e18e533
```

A move is a short token sequence: piece letter, origin square, an optional capture token, destination square, an optional promotion pair and an optional check or mate token, with castling as one token; move numbers are not tokenised. The prompt is the game history up to and including the opponent's move that sets up the puzzle, encoded in the same way.

Puzzles. Puzzles are Lichess puzzles in the format of the puzzle RL task of Shen et al. (2026). The training pool has $68{,}000$ puzzles in five rating bins, B1 (800–1199), B2 (1200–1599), B3 (1600–1999), B4 (2000–2399) and B5 (2400–2499), in proportions $40/30/15/10/5\%$. The $5{,}699$ puzzles without a game history are dropped, which leaves $62{,}301$. Each seed shuffles this pool; the last 256 puzzles of the shuffle form the held-out duel set used only to pick the served member, and training reads the rest in order, 32 puzzles per step, skipping puzzles whose prompt length plus 14 tokens per solution move reaches $1{,}000$. A 200-step run therefore sees $6{,}400$ distinct puzzles ($6{,}432$ with the founding batch). The test set has five bins of 310, 299, 267, 288 and 320 puzzles, with puzzle ratings 801–1198, 1200–1598, 1601–1999, 2000–2399 and 2400–3110; the $1{,}164$ test puzzles of the

Table 9: Ablations. Each row gives $A - B$ in pass@1 averaged over members, per seed where two are listed and the mean over seeds otherwise. 5M rows pair runs from the same batch and seed, as in Figure 9; there GRPO reaches 0.0649 on seeds 1–4 and DUEL is above it by $+0.0098$ (4/4). Same-seed GRPO reruns differ by up to 0.0069. The learning-rate row compares with the learning-rate sweep's own runs at 10^{-5}. The 230M row gives the score share of the final policy in 1,024 chess games against the same-seed GRPO with the pair term on, at $T = 0.7$ / 0.3 (Table 21). The β row compares with GRPO on 32M seed 1, where the default $\beta = 0.1$ gives $+0.0036$. K sweep: mean pass@1 0.0735 ($K = 1$), 0.0739 ($K = 2$), 0.0747 ($K = 4$), 0.0743 ($K = 8$) and 0.0745 ($K = 16$), seeds 1–4; every K is above GRPO on every seed. Summed and per-token scores: Eq. (7), one policy, KL coefficient 0.001. Win-only (loss-only) pairs: each member receives only the pairs its own rollout wins (loses). Partial credit: the training reward is the fraction of the solution's player moves played correctly before the episode ends; evaluation stays on the solve rate, and the $K = 1$ duel trained on it is below the $K = 1$ duel trained on the solve reward by -0.0017 and -0.0009. Constant pair weight: $\beta = 0$ with per-token scores, which differs from GRPO only in the flat per-puzzle weight. Diversity term: as in Parker-Holder et al. (2020), against the same arm without it; at weight 0.03 the members differ at $p < 0.05$ on one of two seeds.

A	B	Size, seeds	$A - B$
DUEL, K=4	K=1, no founding	5M, 4	$+0.0012$
DUEL, K=2, 4, 8, 16	K=1	5M, 4	flat in K
Learning members	Clone members	230M, 5	-0.0100 / -0.0080
DUEL, β=3 / β=1	GRPO	32M, 1	-0.0168 / -0.0078
GRPO, KL 0.05	KL 0.001	5M, 2	$+0.0025$ and $+0.0000$
DUEL, K=1, KL 0.001	DUEL, K=4, KL 0.05	5M, 3	$+0.0022$
DUEL, λ=0.01	GRPO	5M, 4	$+0.0059$
GRPO, lr 3×10^{-5} / 10^{-4}	lr 10^{-5}	5M, 2	-0.0027 / -0.0543
DUEL, K=1, partial credit	GRPO, partial credit	5M, 2	$+0.0077$, $+0.0034$
Win-only pairs, founding noise 0.03	DUEL, λ=0.01	5M, 2	-0.0290, -0.0174
Loss-only pairs, founding noise 0.03	DUEL, λ=0.01	5M, 2	-0.0008, $+0.0028$
Summed log-probabilities	Per-token means	5M, 3	$+0.0063$
Constant pair weight	GRPO	5M, 3	-0.0052, -0.0069, $+0.0120$
Diversity term, weight 0.1 / 0.03	DUEL, founding noise 0.03	5M, 4 / 2	-0.0021 / -0.0071, -0.0009

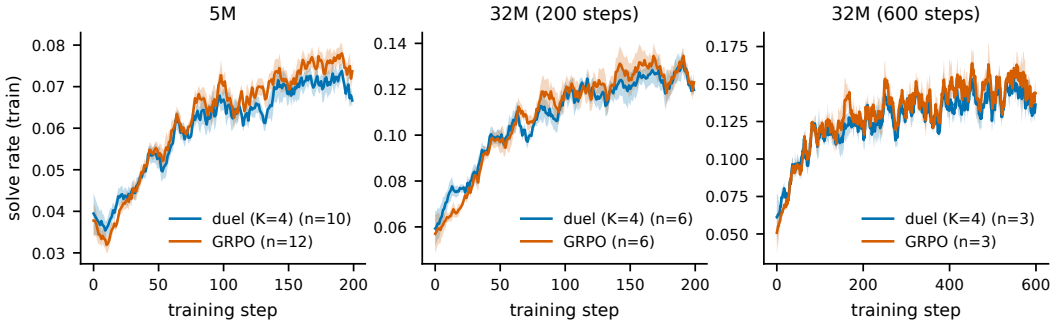

Figure 10: Solve rate of the sampled training rollouts, DUEL ($K = 4$) and GRPO; smoothing, bands and runs as in Fig. 2.

main text are bins B1–B4, and B5 is evaluated but excluded from that mean. No test puzzle shares a puzzle identifier, or a position and solution line, with a training puzzle, and no test puzzle comes from a game that also supplies a training puzzle. Every test puzzle has between 1 and 6 player moves.

Episodes and reward. The model plays every player move of the solution; after each correct move the environment appends the opponent's reply from the solution line. Each model turn decodes 7 tokens. A sampled move is accepted only if its tokens begin with the exact token sequence of a legal move, including the check or mate token; a sample that matches no legal move in this sense is counted as illegal. The episode ends at the first illegal or wrong move, and the reward is 1 if every

Table 10: Sensitivity to learning rate and KL coefficient: final pass@1 averaged over members minus GRPO on the same seed (Δ), and the seeds on which the cell is above GRPO (k/n). Defaults: learning rate 10^{-5}, KL coefficient 0.05, $N = 16$ rollouts; each row changes the settings named. The 5M cells come from a later batch than the GRPO runs they are paired with, and same-seed runs from different batches differ by up to 0.0069. Headline rows use the default settings; no row here was selected into the main text.

Size	Arm	Changed	Seeds	Δ	k/n
5M	DUEL	lr 5×10^{-6}	1, 2	+0.0095	2/2
5M	DUEL	$N = 8$	1, 2	+0.0064	2/2
5M	DUEL	KL 0.001	1, 2	+0.0073	2/2
5M	DUEL	KL 0	1, 2	+0.0074	2/2
5M	DUEL	lr 2×10^{-5}	1, 2	+0.0094	2/2
5M	DUEL	lr 4×10^{-5}	1, 2	−0.0087	0/2
5M	SPEC	KL 0.01	1, 2	+0.0036	2/2
5M	SPEC	KL 0.001	1, 2	+0.0030	1/2
5M	SPEC	lr 2×10^{-5}, KL 0.01	1, 2	+0.0004	2/2
5M	SPEC	lr 2×10^{-5}, KL 0.001	1, 2	−0.0018	1/2
5M	SPEC	lr 4×10^{-5}, KL 0.01	1, 2	−0.0169	0/2
5M	SPEC	lr 4×10^{-5}, KL 0.001	1, 2	−0.0158	0/2
5M	UNIFIED	KL 0.01	1, 2	+0.0083	2/2
5M	UNIFIED	KL 0.001	1, 2	+0.0004	1/2
5M	UNIFIED	lr 2×10^{-5}, KL 0.01	1, 2	+0.0019	2/2
5M	UNIFIED	lr 2×10^{-5}, KL 0.001	1, 2	+0.0061	1/2
5M	UNIFIED	lr 4×10^{-5}, KL 0.01	1, 2	−0.0079	0/2
5M	UNIFIED	lr 4×10^{-5}, KL 0.001	1, 2	−0.0110	0/2
32M	DUEL	lr 5×10^{-6}	1–6	+0.0103	5/6
32M	DUEL	lr 2×10^{-5}	1, 3	+0.0015	1/2
32M	DUEL	lr 2×10^{-5}	2[a]	–	–
32M	SPEC	lr 5×10^{-6}	1–3	+0.0046	2/3
32M	UNIFIED	lr 5×10^{-6}	1–3	+0.0041	2/3
32M	UNIFIED	KL 0.01	1–3	+0.0028	2/3

[a] –, out of memory at final eval (another process on the card).

Table 11: Ablation: GRPO + summed aggregation (GRPO+sum in this table), GRPO with summed, constant-normalised aggregation as in Dr. GRPO. Greedy pass@1 differences against GRPO with per-token mean aggregation (as published) and against GRPO+sum. Single runs on the same seeds; each cell gives the mean difference and the seeds above the comparator.

Arm	vs GRPO (token mean)	vs GRPO+sum	Seeds
5M			
GRPO+sum	+0.0039 (3/4)	–	1–4
DUEL	+0.0098 (4/4)	+0.0059 (3/4)	1–4
DUEL, K=1, KL 0.001	+0.0106 (3/3)	+0.0069 (2/3)	1–3
DUEL, founding noise 0.03	+0.0089 (4/4)	+0.0050 (3/4)	1–4
32M			
GRPO+sum	+0.0089 (5/6)	–	1–6
DUEL	+0.0127 (5/6)	+0.0039 (3/6)	1–6
DUEL, K=1, KL 0.001	+0.0146 (4/4)	+0.0026 (2/4)	3–6

player move matches the solution and 0 otherwise. For an illegal sample the loss sees the tokens up to and including the first one that departs from every legal move. Only the model's own move tokens enter the log-probabilities; prompt tokens and the opponent's replies are masked. Training rollouts are sampled at temperature 1 without truncation, and evaluation is a single greedy decode per puzzle (pass@1).

Optimisation. Every step uses 32 puzzles and $N = 16$ rollouts per puzzle: GRPO has one policy with $G = 16$; the population arms have $K = 4$ members with $G = 4$ each. All members are full copies of the model. The optimiser is AdamW with learning rate 10^{-5}, no weight decay and

Table 12: Growth of the illegal-move rate per decision (the fraction of rollouts that end in an illegal move, divided by the mean number of player decisions per rollout) from the first 20 to the last 20 of the 200 training steps, and the rate over the first 20 (start) and last 20 (end) steps: ranges over runs. The runs are those of Fig. 2, except the ablation rows (App. B).

Size	Arm	Seeds	Growth	Start	End
5M	GRPO	12	+41.0 to +71.5%	0.18–0.20	0.28–0.33
	GRPO, KL coefficient 0.05	2	+33.3% and +45.3%	0.19–0.20	0.26–0.27
	GRPO + summed aggregation (ablation)	4	+37.1 to +49.2%	0.18–0.20	0.26–0.28
	DUEL	10	+2.4 to +24.3%	0.21–0.22	0.22–0.26
32M	GRPO	6	+62.0 to +91.2%	0.11–0.13	0.21–0.22
	GRPO + summed aggregation (ablation)	6	+41.4 to +59.4%	0.12–0.13	0.17–0.19
	DUEL	6	+8.4 to +22.9%	0.13–0.14	0.15–0.17

gradient-norm clipping at $\sqrt{K}$ over all members, for 200 steps; the founding updates use learning rate 5×10^{-5} with a fresh optimiser state. The reference policy is a frozen bfloat16 copy of the base model. GRPO uses a KL coefficient of 0.001; the population arms use $\lambda = 0.05$ on each member's own rollouts, $\beta = 0.1$, $\delta = 0.25$, $c = 8$ and $\lambda_{\text{pair}} = 1$. Forward passes run under bfloat16 autocast with float32 weights. A seed sets the torch generator and the training-pool shuffle and is shared across arms, so seed s of every arm reads the same ordered puzzle stream (the founded arms spend its first batch on founding); GPU sampling is not bitwise deterministic. Table 13 lists the completed runs per arm. From the `experiments` directory, with the bundled python-chess 1.11.2 on the path, a run of the DUEL arm at 5M, seed 1, is launched as

```
PYTHONPATH=pylib python3 chess_bed.py --model models/chess/model_5m_2.1B
  --arm duel --found 1 --steps 200 --eval-every 50 --prompts 32
  --gen-bs 64 --lp-bs 128 --seed 1 --mem-frac 0.05 --out runs/chess/NAME
```

The other arms replace the arm flags by `--arm grpo` (no founding), `--arm spec --found 1` or `--arm unified`; at 32M the model is `model_32m_1.0B` with `--mem-frac 0.11` (`0.045` for two GRPO seeds).

Compute. Table 13 gives wall time per run on one GPU shared with other runs. Each process was capped at a fixed fraction of the card's memory (5% at 5M, 11% or 4.5% at 32M), several runs shared a card, and runs of different arms started within seconds of each other, so the times are contended. The logs do not name the GPU model.

Table 13: Compute per 200-step chess-puzzle run: median (range) over seeds. Wall: from the logged step-0 update to the final evaluation, which covers the evaluations at steps 50, 100 and 150, the final duel and the final evaluation, but not model loading, the step-0 evaluation, the founding step or step 0 itself. Train: the sum of the logged per-step times (rollouts, log-probabilities and backward passes). Memory: peak allocated tensor memory of the process.

Size	Arm	Runs	Wall (min)	Train (min)	Memory (GB)
5M	GRPO	4	11.4 (10.2–12.4)	10.6 (9.3–11.6)	0.89 (0.89–0.93)
5M	DUEL	8	21.9 (18.8–30.8)	19.6 (16.4–28.2)	1.86 (1.59–2.17)
5M	SPEC	8	18.8 (17.8–30.9)	16.5 (15.6–28.0)	1.86 (1.59–2.17)
5M	UNIFIED	3	33.7 (33.5–33.9)	31.1 (31.0–31.3)	1.10 (1.06–1.18)
32M	GRPO	6	11.8 (11.6–16.6)	10.8 (10.7–15.3)	3.00 (2.65–3.07)
32M	DUEL	6	22.0 (20.9–28.5)	19.1 (17.7–25.4)	6.00 (5.16–7.13)
32M	SPEC	6	22.9 (20.8–28.5)	19.7 (17.5–25.3)	6.00 (5.35–7.13)
32M	UNIFIED	6	28.3 (25.5–29.6)	24.5 (22.8–25.7)	4.39 (4.10–4.59)

B.2 WHY THE MEMBERS OF DUEL COLLAPSE ON PUZZLES

We explain why the member mean is flat in K (0.0735–0.0747, $K \in \{1, 2, 4, 8, 16\}$) and why a diversity term separates the members without raising their accuracy, for the member update of Eqs. (3) and (6).

Assumptions. (A1) Shared data: at step t all members see one pool $y_1, \ldots, y_N$ and one pair set $\mathcal{P}$, fixed by the verifier and independent of which member is learning. (A2) Shared anchor: every member is anchored to the same π_{ref} with the same λ. (A3) Small offsets: with $\bar{\theta} = \frac{1}{K}\sum_k \theta_k$ and $\delta_k = \theta_k - \bar{\theta}$, the update is linearised in δ_k; after founding the relative offset is 0.0018. (A4) A preconditioner P_t common to the members, an idealization: near-copies share weights to first order, but their AdamW states may differ, which adds a first-order term $\Delta P_k\, g$ that we neglect.

Proposition 1 (one map, K points). Under (A1)–(A2) member k takes the step $\theta_k \leftarrow \theta_k - \eta P_t\, g_t(\theta_k)$ with one random function g_t for all members, plus a zero-mean term ξ_k from the retain term's own rollouts. The pool depends on the members only through their mixture, which is symmetric in k, so its derivative enters every member's step identically and cancels in δ_k. Hence

$$\delta_k^{t+1} = (I - \eta P_t H_t)\,\delta_k^t - \eta\,(\xi_k - \bar{\xi}) + O(\eta\|\delta\|^2), \qquad H_t = \nabla g_t(\bar{\theta}). \tag{8}$$

Without own-rollout noise, identical members stay identical. No term separates the members: any growth of the spread would be an instability of the single-policy dynamics, shared by all nearby points, and not a difference between members. In a tabular softmax at one decision the constant-weight pair force of Eq. (4) is $-(e_{a^\star} - q)/c$ in the logits, where q is the failure distribution of the pool; it does not depend on θ_k, so the pair term adds nothing to H_t and only the anchor contracts the offsets (at $\pi_{\bar{\theta}} = \pi_{\text{ref}}$ its Hessian is the Fisher matrix). At our learning rate and λ this contraction is slow over 200 steps; the flat K sweep does not require it, since founding sets a small spread and no term separates the members further. Whether the offsets lie along directions that barely change the policy or decay, both fit the proposition; the per-member KL to the parent separates the two.

Proposition 2 (spread without different labels is second order). Let $J(\theta)$ be the expected solve rate of the sampled policy; the expansion is local and concerns this sampled rate, not greedy pass@1, which changes at argmax boundaries. Because $\sum_k \delta_k = 0$ the first-order term cancels:

$$\frac{1}{K}\sum_{k=1}^{K} J(\theta_k) = J(\bar{\theta}) + \frac{1}{2K}\sum_{k=1}^{K} \delta_k^\top \nabla^2 J(\bar{\theta})\,\delta_k + O(\|\delta\|^3). \tag{9}$$

A diversity term that leaves every member winning the same pairs changes the member mean in two ways: it may move the centre $\bar{\theta}$, with a sign the theory does not fix, and the spread it creates enters only at second order, with the sign of the curvature of J along the offsets, negative near a maximiser shared by all members. A mixture that draws one member per episode has solve rate equal to the member mean and never above the best member. The determinant term of Parker-Holder et al. (2020) at weight 0.1 makes the members differ on all four seeds (permutation $p \le 0.041$) and changes their mean by -0.0021 against a control with the same founding noise ($t = -0.85$), a change not distinguishable from zero, as expected of a second-order spread term plus a small centre shift; their served mixture, which averages the members' next-token probabilities, changes by -0.0028.

When a population should help. Let member k train on its own comparisons $\mathcal{P}_k$, $g_k = g(\theta_k; \mathcal{P}_k)$. Eq. (8) then gains the forcing $f_k = g(\bar{\theta}; \mathcal{P}_k) - \frac{1}{K}\sum_j g(\bar{\theta}; \mathcal{P}_j)$, and the offsets settle near $-H^{-1} f_k$, which is non-zero exactly when the members are credited for different comparisons. Two routes supply f_k. The first is member-specific credit on a shared reward, such as the coverage term of UNIFIED, which rewards a member's successes on puzzles the others fail; it can buy coverage of more puzzles, not a higher member mean. The second is curl. On puzzles the comparison of two rollouts is $r(y_a) - r(y_b)$ with a binary scalar r: an exact difference, whose Hodge curl is identically zero (Jiang et al., 2011). Every member faces one potential, so for the member mean and for a per-episode mixture no population scores above K copies of its best member. This bounds what the members can score; it does not decide which training procedure reaches the best member, and the lack of specialization we observe follows from (A1)–(A2), not from this bound. When the comparisons are outcomes between members, A_{ij}, and A has a non-zero curl part C, complementary policies can have a role, although curl alone does not rule out a single dominant policy (Balduzzi et al., 2019; Czarnecki et al., 2020). Every row of C sums to zero, so a per-member score contracted with C under uniform weights over opponents vanishes; the curl reaches the members only through member-specific rows (each member's pairs taken against its own opponents) or through selection. Population-based training treats the complementary, transitive case: its exploit step, and the pairwise guidance of weaker agents by stronger ones in Bai & Cheng (2024), move members toward the best one (Jaderberg et al., 2017), a contraction that serves a scalar objective, where the population is a parallel search worth its best member.

Numeric checks. With shared minibatch noise, identical members stay bit-identical over 500 steps of a quadratic loss, whose offset decay matches the finite-difference Jacobian to 5×10^{-12}. On a six-action softmax with the pair loss and a KL anchor the members' pair gradients agree exactly, the decay matches Eq. (8) to 2.4×10^{-4}, and Eq. (9) holds to a ratio of 0.93 at small offsets. A determinant term at a low-accuracy equilibrium (0.34) costs 6.8×10^{-3} against the members' own centre (quadratic prediction 6.3×10^{-3}) while moving that centre up by 8.7×10^{-3}; near the maximiser it lowers the member mean from 1.000 to 0.929. On rock–paper–scissors the pairwise total variation falls from 0.150 to 0.002 when members train against the shared field and rises to 0.999 when each trains against its own opponent.

The other beds. In the engine-credit chess arm the population collapses by replacement: losing slots are refilled with jittered copies of θ (App. A.4). On Goofspiel the members of the fixed arm differ from one another in parameters, by the birth draw (norm $0.707\,\sigma\sqrt{D}$ for birth-noise scale σ and D parameters, near-orthogonal; App. A.5), and not in anything learned; parameter distance is not a difference in which comparisons a member wins. There the members duel one another, so their comparisons can carry curl and the forcing term above can act once births are scaled to the parent. The puzzle null is the outcome the theory predicts for shared credit on a curl-free comparison. The population's claim to hold memory of the field's strategy structure is tested where the comparisons have curl and each member receives its own rows of them.

Appendix overview

App. A holds chess, population and game-bed results moved from the main text, and App. B further chess-puzzle results. App. C states the full KAIKA objective and its schematic (Fig. 12); App. D extends the related work. App. E gives the game-bed configurations (App. E.1) and the chess training protocol with its settings (App. E.2, Table 16). App. F gives the per-seed chess comparison, the Stockfish ladder and the chess ablations (App. F.1), the population arms (App. F.3), the end-state games (App. F.4) and the Blotto readouts (App. F.5). App. G covers reproducibility and compute, and App. H the earlier gradient-free KAIKA archive.

C The full objective

The main text (§3) states the instance behind the reported results. This section states the full objective of which it is a subset; Table 2 lists the terms each result trains.

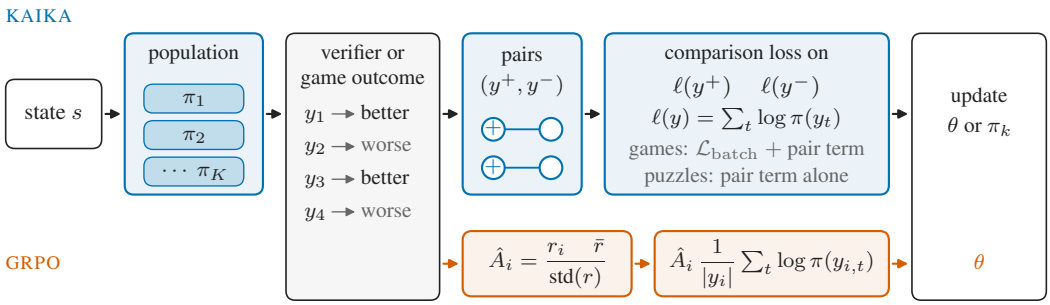

Figure 11: KAIKA and GRPO from the same state. KAIKA's population samples continuations, a verifier or the game outcome orders them, and ordered pairs (on puzzles every success against every failure, in games one pair per labelled move) are scored by the difference of their log-probabilities summed over tokens, a term added to the policy gradient $\mathcal{L}_{\text{batch}}$ in games (Eq. 1) and replacing it on puzzles (Eq. 3); the GRPO strip is the puzzle baseline of Eq. 2, which standardises one scalar reward within the group and weights each continuation's per-token mean log-probability by it.

Population and rollouts. KAIKA keeps K trainable policies $\pi_{\theta_1}, \ldots, \pi_{\theta_K}$ ($K = 8$). With independent parameter blocks the population state lies in the product $\mathcal{M}^K$ of the members' policy

manifolds (near regular points), whose Fisher metric is block-diagonal and only positive semidefinite in parameter coordinates; a shared trainable backbone couples the members. At iteration n each member's rollout policy $\bar{\pi}_k = \pi_{\theta_k^{(n)}}$ is frozen and samples G responses per prompt, $y_{kg} \sim \bar{\pi}_k(\cdot \mid x)$, with rewards $r_{kg} = R(x, y_{kg})$, so a prompt has $N = KG$ candidates. The number of policies, the number of responses per policy and the number of pair comparisons are different quantities.

Own-policy GRPO. Each member keeps a standard GRPO group (Shao et al., 2024),

$$A_{kg}^{\text{own}} = \frac{r_{kg} - \bar{r}_k}{s_k + \epsilon_A}, \tag{10}$$

with $\bar{r}_k$, s_k the mean and standard deviation of member k's G rewards; with $G = 1$ it is zero. Chess members play one move per position, so the population arms standardise per position: the moves members play or sample at x form one group, $A_a = (r_a - \bar{r}(x))/(s(x) + \epsilon_A)$, r_a being a played move's game outcome or a sampled move's mean rollout outcome; the baseline pools other members' outcomes, and a move alone at its position gets zero. With the token ratio $\rho_{kgt} = p_{\theta_k}(y_{kg,t} \mid x, y_{kg,<t}) / p_{\bar{\theta}_k}(y_{kg,t} \mid x, y_{kg,<t})$ and $C_\varepsilon(\rho, A) = \min(\rho A, \text{clip}(\rho, 1-\varepsilon, 1+\varepsilon)A)$, a batch of B prompts gives

$$\mathcal{L}_{\text{RL},k} = -\frac{1}{BG} \sum_x \sum_{g=1}^{G} \frac{1}{T} \sum_{t=1}^{T} C_\varepsilon\big(\rho_{kgt}, A_{kg}^{\text{own}}\big)\,, \tag{11}$$

Its denominator is the member's own behaviour policy, and clipping does not bound the update.

Cross-member DPO. For candidates $a = (k, g)$ and b of one prompt, $a \succ b$ when $r_a > r_b + \tau$ ($\tau \geq 0$), and $\mathcal{E}_{kj}(x)$ holds the pairs drawn from members k and j; pairs within one member ($k = j$) need $G \geq 2$. Each pair is weighted by its reward gap under the shared normalisation,

$$w_{ab} = \frac{r_a - r_b}{s_{\text{pop}} + \epsilon_A}, \tag{12}$$

where s_{pop} is the standard deviation of all KG rewards; separately normalised advantages (Eq. 10) can reverse a cross-member comparison. With $s_k(x, y) = \log \pi_{\theta_k}(y \mid x) - \log \pi_{\text{ref},k}(y \mid x)$ for a fixed full-support reference and $\beta > 0$, both responses are scored under the receiving member,

$$\begin{aligned} m_{k,ab} &= \beta\left[s_k(x, y_a) - s_k(x, y_b)\right], \\ \mathcal{L}_{\text{pair},k} &= -\frac{1}{B} \sum_x \frac{1}{Z_k(x)} \sum_{j=1}^{K} \sum_{(a,b) \in \mathcal{E}_{kj}(x)} w_{ab} \log \sigma(m_{k,ab})\,, \end{aligned} \tag{13}$$

with $Z_k(x) = \sum_j \sum_{(a,b) \in \mathcal{E}_{kj}(x)} w_{ab}$ and a zero contribution when $Z_k(x) = 0$. Member k thus learns how it ranks a winner that member j generated. Samples, rewards, graph, weights and references are frozen during the update, and the preference data are regenerated from the population at every iteration; with equal gaps, Eq. 13 is the DPO loss (Rafailov et al., 2023).

Joint objective. With a reference penalty $\mathcal{K}_k = \mathbb{E}_{c \sim \nu_n} D_{\text{KL}}(p_{\theta_k}(\cdot \mid c) \,\|\, p_{\text{ref},k}(\cdot \mid c))$ over the batch contexts ν_n and the members' disagreement on the same contexts,

$$\mathcal{D}_{\text{pop}} = \mathbb{E}_{c \sim \nu_n} \frac{1}{K} \sum_k D_{\text{KL}}(p_{\theta_k}(\cdot \mid c) \,\|\, \bar{p}(\cdot \mid c))\,, \qquad \bar{p} = \frac{1}{K} \sum_k p_{\theta_k}, \tag{14}$$

the population minimises

$$\mathcal{L}_n = \frac{1}{K} \sum_{k=1}^{K} \left[\mathcal{L}_{\text{RL},k} + \lambda_{\text{pair}} \mathcal{L}_{\text{pair},k} + \lambda_{\text{ref}} \mathcal{K}_k\right] - \lambda_{\text{div}} \mathcal{D}_{\text{pop}}, \tag{15}$$

with nonnegative coefficients. Disagreement is not automatically useful specialisation, and the gradient of $\mathcal{D}_{\text{pop}}$ vanishes when the members agree, so it cannot separate identical policies.

Founders. Members descend from one GRPO-updated parent θ_{parent}: with per-response clipped losses ℓ_g (the parent's advantages and behaviour probabilities) on a second batch $\mathcal{D}_1$ of K parent responses per prompt, founder k takes one step

$$\theta_k^{(0)} = \theta_{\text{parent}} - \eta_f \nabla_\theta \sum_{g=1}^{K} m_{kg}\, \ell_g(\theta_{\text{parent}}), \qquad m_{kg} = \frac{1 + \alpha \mathbf{1}[g = k]}{K + \alpha}, \quad \alpha \geq 0, \tag{16}$$

with $g_g = \nabla \ell_g(\theta_{\text{parent}})$ and mean $\bar{g}$,

$$\theta_k^{(0)} - \frac{1}{K} \sum_j \theta_j^{(0)} = -\eta_f \frac{\alpha}{K + \alpha} (g_k - \bar{g}) \,. \tag{17}$$

At $\alpha = 0$ the founders coincide; Eq. 17 is exact for SGD, not Adam. The founded chess population arms take one founder step with the η_f listed in Table 5.

What the comparisons carry. For N scalar rewards the gap matrix $M = r\mathbf{1}^\top - \mathbf{1}r^\top$ has rank at most 2 and satisfies $M_{ab} + M_{bc} + M_{ca} = 0$, so its N^2 entries are fixed by $N - 1$ numbers: many pairs do not add independent reward dimensions, and gaps of one scalar reward cannot be cyclic. Cyclic preferences need a matchup-dependent relation, such as the win rate W_{ij} of π_i against π_j that the game beds measure. For one policy trained on all reward-ordered pairs of equal-length responses, gap-weighted DPO and the centred policy gradient point in the same direction when the policy equals both the DPO reference and the GRPO behaviour policy and pair mass is equal across prompts, since $\sum_{A_a > A_b} (A_a - A_b)(h_a - h_b) = N \sum_a A_a h_a$ for centred advantages A and scores $h_a = \nabla_\theta \log \pi_\theta(y_a \mid x)$; with a fixed reference this holds at the first iteration only, and member-specific data, evolving margins and clipping separate the two afterwards. For binary rewards, GRPO's gradient contrasts positive with negative responses as DPO's does for one pair (Wu et al., 2025).

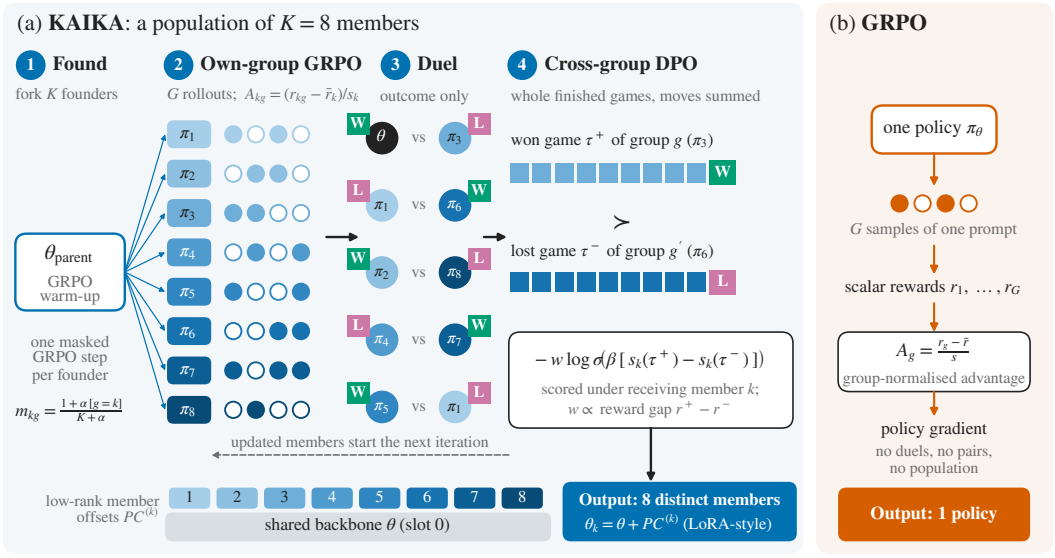

Figure 12: (a) The KAIKA population objective (Eq. 15). **1** One GRPO-trained parent forks $K = 8$ founders with one masked GRPO step each (Eq. 16). **2** Each member samples its own group of G rollouts and takes a GRPO step on group-normalised advantages (Eq. 10). **3** Members and θ duel; only the win or loss is observed. **4** A won game of one member's group and a lost game of another's form a DPO pair, scored under the receiving member and weighted by the reward gap (Eq. 13). Members are low-rank offsets $\theta_k = \theta + PC^{(k)}$ of a shared policy θ. (b) GRPO trains one policy from scalar rewards with a group-normalised advantage. Schematic; rollout rewards and duel outcomes are illustrative. The chess and Nim results come from the shared-policy instance of §3.

D EXTENDED RELATED WORK

None of KAIKA's primitives is new (Bradley–Terry and Elo (Bradley & Terry, 1952; Elo, 1978), MAP-Elites (Mouret & Clune, 2015), self-play curricula (Silver et al., 2018)); we position the combination by nearest family.

Preference and group-relative learning. In-weight preference learning optimises one policy against an evaluator's comparisons (Christiano et al., 2017; Ouyang et al., 2022; Rafailov et al., 2023). Step-DPO uses individual reasoning steps as preference units (Lai et al., 2024); rStar-Math trains a process preference model from verified search trajectories (Guan et al., 2025); and GIFT combines group-relative sampling with DPO-style implicit rewards (Wang, 2025). DGPO splits each GRPO group by the sign of the advantage into positive and negative subgroups and maximises a Bradley–Terry likelihood between them, dispensing with the policy-gradient framework (Luo et al., 2026, §3.1, Alg. 1); GDPO weights a Diffusion-DPO loss by group-relative advantages, and pairwise DPO is its case $G = 2$ (Yi et al., 2026, §4.2). Both replace the policy-gradient objective with a group DPO objective. Wu et al. (2025, §3–4) write GRPO as an N-vs-M contrastive loss whose gradient has DPO's positive-versus-negative form, and train with two rollouts per prompt (2-GRPO). Pref-GRPO runs the other direction: a pairwise preference reward model compares the images in a group, and each image's win rate becomes its GRPO reward (Wang et al., 2025, §IV-C). In games, KAIKA keeps its policy-gradient term and adds the DPO term on the same self-play population. By learning signal, DGPO and GDPO score samples with reward functions, 2-GRPO with verifiable rewards and Pref-GRPO with a preference model; KAIKA's pairs come from its own games, labelled by engine credit in assisted chess and by population-internal Hodge-derived comparisons in the engine-free variant. Online GRPO and online/semi-online DPO achieved similar performance and convergence on their verifiable math and non-verifiable instruction-following tasks (Lanchantin et al., 2025). On chess puzzles, action-value distillation improved GRPO relative to exact-move binary rewards (Hwang et al., 2025). Personalised alignment includes reward modelling, adaptation and inference-time steering (Li et al., 2024; Lau et al., 2024; Shenfeld et al., 2025; Kim et al., 2025). Other approaches track drift (Son et al., 2025), learn preference losses (Lu et al., 2024), or seek Nash-preference equilibria over pairwise utilities (Munos et al., 2024; Rosset et al., 2024; Swamy et al., 2024); group-relative advantages (Shao et al., 2024) baseline responses within a sampled group. Nash-MD, DNO and SPO remain unevaluated here. GPRL embeds responses in skew-symmetric subspaces and carries per-dimension group-relative advantages through the policy update (Umer et al., 2026); MCHF defines a Markov kernel from pairwise utilities and converges to a stationary distribution at a rate determined by their non-transitive component (Koriyama & Liang, 2026). These methods already treat preferences as structured relations; that representation and group-relative updates are shared prior art.

Shared and persistent populations. NeuPL is the nearest prior art: a population held in one conditional model, with transfer across members and convergence to a population of best-responses (Liu et al., 2022b); JPSRO, simplex populations, extensive-form double-oracle, PSRO and leagues also maintain strategic alternatives and train responses to a population (Marris et al., 2021; Liu et al., 2022a; McAleer et al., 2021; Lanctot et al., 2017; Vinyals et al., 2019; Omidshafiei et al., 2019; Balduzzi et al., 2019; Czarnecki et al., 2020; Balduzzi et al., 2018), behavioural-diversity objectives steering their variety (Rosin & Belew, 1997; Perez-Nieves et al., 2021; Liu et al., 2021). QD archives retain behaviour, including under reward drift (Mouret & Clune, 2015; Cully et al., 2015; Pugh et al., 2016; Fontaine et al., 2020; Cully, 2021; Faldor et al., 2025; Gallotta et al., 2024); preference-QD uses comparative feedback to guide diversity (Bradley et al., 2024; Ding et al., 2024; Wang et al., 2023), while mixture-preference and pluralistic-alignment models represent multiple reward hypotheses (Oh & Shah, 2014; Chakraborty et al., 2024; Sorensen et al., 2024). Our archive studies measure retained coverage under changing comparisons. The language-game neural studies train the shared policy from population experience. The chess study instead uses a direct same-position pair write, whose update remains active under an exact-copy population. NeuPL likewise shares parameters across its population.

Structured relational feedback and history. Hodge decomposition separates an additive rating from the cyclic residual of a comparison field (Jiang et al., 2011), and mElo-class corrections use it where real games are non-transitive (Balduzzi et al., 2018; Bertrand et al., 2023); this bridge

is prior work. Time-varying rating systems keep one drifting scalar per player (Glickman, 1999; Herbrich et al., 2006; Dangauthier et al., 2007; Coulom, 2008); continual learning treats replay and consolidation as safeguards against forgetting (Parisi et al., 2019; Kirkpatrick et al., 2017; Lopez-Paz & Ranzato, 2017; Rolnick et al., 2019); dueling bandits bound regret (Yue et al., 2012; Wu & Liu, 2016; Saha & Gupta, 2022; Kleine Buening & Saha, 2023; Kolpaczki et al., 2022) — DEX3.S is run and beaten at its published tuning (§H.2.2); ANACONDA (Kleine Buening & Saha, 2023) is the nearest unrun peer, named rather than claimed covered.

Public preference datasets and interactive-task benchmarks (Zollo et al., 2024; Li et al., 2026; Kirk et al., 2024; Chiang et al., 2024; Zheng et al., 2023; Merrill et al., 2026) provide different parts of this evaluation. We test retention, structured comparison learning and neural policy improvement separately. In the headline assisted chess comparison both arms receive the same engine credit and engine-valued branch moves, and a 300-step run costs 2.22 GPU-hours for KAIKA and 2.75 for GRPO (Table 29); the engine-free comparison uses outcome credit on both arms, and Table 29 accounts its 64-row control but not the wall clock of the 128-row control (§5.1).

E Experimental setup and hyperparameters

E.1 Language-model games against GRPO

Both arms start from Qwen2.5-1.5B-Instruct (Qwen et al., 2025) and play the TextArena Nim environment (Guertler et al., 2025). In training and in duels, a policy's sampler is masked to the position's legal moves. Each training step plays 32 games, credits every move with its game's outcome and trains 64 policy-gradient rows with a clipped surrogate and a clamped KL term to the base model. GRPO places its single policy in both seats and standardises the advantages over all of the step's rows. KAIKA's eight slots are θ and seven low-rank members. Each step schedules consecutive random perfect matchings among the slots, with seats alternated across rounds, and Elo standings update from every game. KAIKA's write to θ applies GRPO's estimator to the rows its slots play and adds the winner-vs-loser DPO term, one pair per ply of each decided game. The members learn at their own rate, and each step the members that lose the screen are replaced by jittered copies of θ. The jitter adds 0.03 per coordinate, unscaled by the parent, on the population side only; on the chess bed this scale places births at norm 77.9 against member norms of about 2.6, and on these beds the norms were not measured. Table 14 lists the settings, read from each run's configuration. Across the 10 training runs (3000 logged steps), the logs count no engine call, solver label, branch row or game against an outside opponent. The same campaign also trained tic-tac-toe; the bed was removed from the evaluated set because it has no established benchmark provenance.

Table 14: Language-model games: training settings, read from each run's configuration; every run of an arm has these values.

Setting	KAIKA	Batch PG
Training steps	300	300
Learning rate	1×10^{-5}	1×10^{-5}
Warm-up steps	20	20
KL coefficient β	0.1	0.1
KL clamp	8	8
Clip ϵ	0.2	0.2
Sampling temperature	0.7	0.7
Top-p	0.95	0.95
Population slots	8	–
Member rank	64	–
Member learning rate	1×10^{-4}	–
Member KL clamp	8	–
Offspring jitter σ	0.03	–
DPO β	0.1	–
Elo K	24	–
Random opening plies (Nim)	2	

Each duel pairs the two arms' final policies from the same training seed over 128 games, with seats alternating and top-p 0.95; duels start from two random opening plies. Table 15 gives KAIKA's

score for each training seed. On Nim, two members of a final population choose the same move with probability 0.301–0.482 (mean over member pairs; range over 5 seeds), a member agrees with itself with probability 0.551–0.677, and uniform play gives 0.220–0.250; each probe covers 178–221 positions from θ's self-play.

Table 15: KAIKA's score against GRPO for each training seed: wins plus half draws over 128 games.

		Training seed				
Game	T	1234	2	3	4	5
Nim	0.7	0.5312	0.9609	0.6875	0.2578	0.3906
Nim	0.3	0.6016	0.9844	0.7188	0.2188	0.3359

E.2 ASSISTED CHESS PROTOCOL

Protocol. Both arms start from LFM2.5-230M-Chess-clean (Labonne, 2026; one token per move over the 1,968 geometrically possible moves, illegal moves masked, a fixed 80-token board prompt; Stockfish-distilled SFT on 126M positions, then a GRPO polish; about 1450 Elo as a raw policy, 2004 with depth-3 search; every duel here plays the raw policy) and train 300 steps of 32 self-play games, no exogenous engine games, learning rate 10^{-5}, reference-KL 0.1. Action credit is the engine-evaluation change across the move (2,000 nodes, 300-centipawn scale). The assisted arm has eight rank-64 members and adds same-position branches with probability 0.5, the alternative sampled from the mover's legal distribution excluding the played move; engine labels train the sequence-sum, reference-relative pair term of Equation 1 ($\beta = 0.1$). The shared policy takes a full-parameter update; the historical Hodge chart write and winner-likelihood anchor are inactive. Member coordinates take the mean split: each member steps on its own credited coordinate gradient minus the mean of the live members' coordinate gradients. The recipe also names the sketch-B split, which needs a consensus chart that the full-parameter write does not build, so the trainer falls back to the mean split without a warning. The recipe enables the collapse guard, but the game path runs it only when the illegal-move threshold is positive; the threshold is zero, so the guard never evaluates on this bed and its per-step collapse-event list is empty by construction. The Elo screen keeps higher-ranked members and reinitialises eligible lower-ranked ones around the shared policy (jitter 0.001, three-step grace); `kaika_replace=0` means replacing all eligible screened losers, not disabling replacement. Evaluation restores 133 tensors over the base checkpoint for the final shared policy or GRPO checkpoint at step 299 and alternates seats over 512 paired environment seeds from 971000 (1,024 games per seed; temperature 0.7, top-p 0.95, Hugging Face runner, chunk 32). Every pairing completes all games with zero exhausted games, illegal moves or retries. Table 16 lists the realised chess settings.

Table 16: Chess training settings, read from each run's saved configuration (identical across seeds). All arms train 300 steps from LFM2.5-230M-Chess; engine-credit arms play 32 self-play games per step. Engine-credit members are rank-64 offsets of θ. GRPO's learning rate and β are swept in App. F.2. Evaluation: $T \in \{0.7, 0.3\}$, top-p 0.95, 1,024 games per seed.

Arm	Credit	lr	KL β	DPO β	PG rows / step	Adv. groups	Members: lr, jitter
KAIKA	engine	10^{-5}	0.1	0.1	32 of 64	4	10^{-4}, 10^{-3}
GRPO	engine	10^{-5}	0.1	–	64	1	–
KAIKA, engine-free	critic	10^{-5}	0.1	0.1	256 of 256	32	10^{-4}, 0.03
GRPO + critic (128 rows)	critic	10^{-5}	0.1	–	128	1	–

F ADDITIONAL RESULTS

F.1 ENGINE-LABEL-MATCHED CHESS COMPARISON

The closest completed engine-supervised chess control is GRPO, a batch-standardised policy-gradient (GRPO) comparator training 64 rows per step under the same engine-labelling rule as the

KAIKA theta write, with advantages standardised across the batch. Both learners pay for and receive the same per-move engine credit and engine-valued branch moves; they differ in how branch moves enter the update. The comparator subsamples its 64 rows uniformly from played and branch rows and credits each alone, so the two moves at a branch rarely share a step and are never compared. About a third of the comparator's row pool is branch moves, so roughly 21 of its 64 policy-gradient rows per step are branch moves with their own engine credit. KAIKA's policy-gradient term uses only its 32 played rows; it keeps each branch move beside the move played in that position and applies the pair term of Equation 1. Write objectives, member rollouts, selection and total compute also differ.

The rows are moves at distinct positions rather than repeated responses to one prompt. This is the global advantage normalisation of REINFORCE++ (Hu et al., 2025). The rollout group parameter sets the self-play game budget, not the number of samples per state. Grouping the played and branch move by position reduces every non-tied pair to fixed opposite advantages and diverged; for the engine-free comparator, grouping by opening had no consistent direction across 2 seeds.

Each comparator branch move is drawn from $q(a' \mid s, a) = \pi(a' \mid s)/(1 - \pi(a \mid s))$, the policy with the played move a excluded, credited independently by the engine and included in the policy-gradient and KL terms. This conditional draw is not an on-policy sample, so treating it as a policy-gradient row is a different update from on-policy policy gradient. KAIKA uses branch moves only in its pair term, and the same-credit comparator without branch rows also scores below KAIKA.

Both learners use the k_3 estimator of KL to the frozen base in the loss. The configured one-sided per-token log-ratio clamp removes the KL gradient beyond its bound; it is a looser form of the low-variance k_3 clamp used in `verl`. The reported divergence is unclamped. Comparators that grouped only the played and branch move of each position diverged with or without the clamp, whereas the batch-grouped comparator was trained with it. A comparator with eight fresh samples per position trained finitely without the clamp. A same-credit, batch-grouped comparator without branch rows trained finitely without the clamp and scored below KAIKA.

RLOO and REINFORCE use the same clipped objective, KL term and rows as the GRPO comparator and change only the advantage: the leave-one-out mean of all other rows for RLOO (Ahmadian et al., 2024), or the batch mean for REINFORCE with a constant baseline (Williams, 1992). The KL penalty therefore enters the loss rather than the reward as in the original RLOO.

With one optimiser step per rollout batch, the importance ratio is one and the clip is inactive, as in TRL's `GRPOTrainer` with a single iteration. Log-probabilities are taken at the sampling temperature over the legal moves. Only the move token is trained, so token-, sequence- and constant-normalised losses coincide. On fixed inputs the comparator's loss gradient matches TRL's within the audit tolerance; this validates the loss, not the row-sampling pipeline, and a ratio of one at the pre-update parameters does not make the branch rows on-policy. The recipe uses a constant learning rate with no warm-up, AdamW with its default weight decay and the configured gradient-norm clip. Both arms mask illegal moves in sampling and scoring. For the earlier pair write, KAIKA seeds 6 to 10 were added after the five-seed result was known. The same holds for the fixed-code KAIKA arm: seeds 1234 and 2 to 5 were the planned set, and seeds 6 to 10 were added afterwards to extend the comparison to ten seeds. Against GRPO at $T = 0.7$ its five-seed mean is 0.6121 (5 of 5 above parity) and its ten-seed mean 0.6009 (10 of 10). All arms ran 300 steps. Seed 5 of that arm has two training runs from two checkouts; the GRPO head-to-head uses one checkpoint, and the base-anchor, RLOO, REINFORCE, ladder and position-grouped duels use the other; each duel receipt names its seat checkpoints, so every evaluated policy is identified by run tag and checkpoint rather than by seed number. Observed engine evaluations per step (single steps, not constants): 2,732 at KAIKA step 190, 2,563 at matched GRPO step 148. Quality checks found no non-finite step fields, losses above 10^3 or illegal moves in evaluated games; member gradient norms are finite but reach 1.7×10^{16} in the unclamped seed-1234 run, with 172 steps above 10^6 (Table 22). Maximum raw KL: GRPO seed 5 4,639,201.50, RLOO 86,516,232, REINFORCE 50,970,088. Per-seed scores (Table 17) are read from the head-to-head result files into `tables/chess_matched.json`, which `gen_chess_matched.py` tabulates; `verify_chess_matched.py` checks every value against the stored duel summaries (`chess_matched_duels.json`). The RLOO and REINFORCE rows against the corrected pair write are read from `tables/chess_winmap.json`, the source of Table 20.

At $T = 0.7$ the move entropy of KAIKA's shared policy θ is 0.67–0.79 nats and that of GRPO 1.02–1.08 (the mean entropy of each policy's sampled legal-move distribution, top-p 0.95, over one fixed set of positions from the base policy's own play). Two further duels per seed equalise it at the $T = 0.7$ setting: GRPO at $T = 0.45$–0.54 against KAIKA at 0.7 (KAIKA scores 0.5127, 7/10 seeds above parity), and KAIKA at $T = 0.92$–1.07 against GRPO at 0.7 (0.5234, 7/10). Each temperature is read off that policy's entropy-temperature curve by linear interpolation (`chess_entropy_probe.json`); `verify_chess_matched.py` recomputes it.

Table 17: Per-seed shared-policy score share against each comparator. Position-grouped GRPO is an earlier comparator that groups advantages by position. Rows marked eq. equalise the two policies' move entropy at the $T = 0.7$ setting.

Comparator	T	s1234	s2	s3	s4	s5	s6	s7	s8	s9	s10
Batch PG, same information; KAIKA on the corrected pair write	0.7	0.6460	0.5640	0.6274	0.5996	0.6235	0.5522	0.5493	0.6006	0.6479	0.5981
Batch PG, same information; KAIKA on the corrected pair write	0.3	0.5801	0.4707	0.5703	0.5161	0.5200	0.4683	0.4600	0.5737	0.5601	0.5488
Batch PG cooled to equal entropy	eq.	0.5713	0.4619	0.5693	0.5059	0.5249	0.4272	0.4775	0.5088	0.5542	0.5259
Batch PG, KAIKA heated to equal entropy	eq.	0.5552	0.4521	0.5879	0.5264	0.5166	0.4639	0.4746	0.5347	0.5674	0.5557
RLOO, same information; KAIKA on the corrected pair write	0.7	0.6377	0.5928	0.6631	0.6108	0.6636	–	–	–	–	–
RLOO, same information; KAIKA on the corrected pair write	0.3	0.5151	0.5156	0.5962	0.5200	0.5718	–	–	–	–	–
REINFORCE, same information; KAIKA on the corrected pair write	0.7	0.6382	0.5850	0.6387	0.5850	0.6465	–	–	–	–	–
REINFORCE, same information; KAIKA on the corrected pair write	0.3	0.5767	0.5176	0.5332	0.5073	0.5938	–	–	–	–	–
Batch PG, same information; earlier pair write	0.7	0.6318	0.6147	0.5957	0.6069	0.6143	0.5859	0.5781	0.5767	0.6294	0.6133
Batch PG, same information; earlier pair write	0.3	0.5659	0.5010	0.5269	0.5347	0.4961	0.5283	0.4971	0.4995	0.5742	0.5474
RLOO, same information; earlier KAIKA checkpoint	0.7	0.6270	0.6094	0.6113	0.5923	0.6094	–	–	–	–	–
RLOO, same information; earlier KAIKA checkpoint	0.3	0.5522	0.5415	0.5405	0.5562	0.5098	–	–	–	–	–
REINFORCE, same information; earlier KAIKA checkpoint	0.7	0.6372	0.6460	0.5972	0.6128	0.5991	–	–	–	–	–
REINFORCE, same information; earlier KAIKA checkpoint	0.3	0.5703	0.5527	0.5176	0.5596	0.4917	–	–	–	–	–
GRPO, position groups (grpo_lt1 credit: no branch rows; KL unclamped on all rows)	0.7	0.5742	0.5889	0.6401	0.5620	0.6138	–	–	–	–	–
GRPO, position groups (grpo_lt1 credit: no branch rows; KL unclamped on all rows)	0.3	0.5107	0.5483	0.5923	0.5103	0.5664	–	–	–	–	–
No engine on either side: outcome-credit GRPO, 64 rows	0.7	0.4946	0.5718	0.6133	0.5835	0.5112	–	–	–	–	–
No engine on either side: outcome-credit GRPO, 64 rows	0.3	0.5229	0.5835	0.6094	0.5698	0.4980	–	–	–	–	–
No engine on either side: outcome-credit PG grouped by opening (R1b; two seeds)	0.7	0.5513	0.5073	–	–	–	–	–	–	–	–
No engine on either side: outcome-credit PG grouped by opening (R1b; two seeds)	0.3	0.5591	0.5186	–	–	–	–	–	–	–	–

What the controls attribute. The paired factorial in Table 21 isolates the pair term from member learning against same-seed GRPO. At the higher temperature the pair effect is positive with learning and exact-copy members, while both learning-member contrasts and the interaction interval span zero; at the lower temperature all factorial-effect intervals span zero. Exact-copy members without the pair term remain +0.0531 above parity at the higher temperature and +0.0333 at the lower temperature, each on all 5 seeds.

Table 18: Stockfish ladder at 12000 nodes per move. Rungs are Stockfish `UCI_Elo`, 1320 is the floor, and the base falls below the lowest rung at $T = 0.7$ and $T = 0.3$ and is bracketed under greedy decoding ($T = 0$). At $T = 0.7$ GRPO also scores below 0.5 at the lowest rung on every seed, and KAIKA is bracketed on 5/10 seeds. Fitted Elo below the floor is extrapolated; scores against the shared 1320 and 1500 rungs are the model-free comparison.

T	Arm	Fitted Elo [95% interval]	Bracketed/n	Score at 1320	Score at 1500
0.7	base (LFM2.5-230M-Chess-clean)	1195 [1120, 1260]	0/1	0.297	0.102
0.7	KAIKA θ	1360 [1341, 1378]	5/10	0.485	0.240
0.7	GRPO	1283 [1268, 1298]	0/10	0.419	0.202
0.7	RLOO (GRPO recipe)	1256 [1197, 1315]	0/5	0.386	0.170
0.7	REINFORCE (GRPO recipe)	1259 [1231, 1287]	0/5	0.363	0.197
0.7	paired difference (KAIKA − GRPO)	+77 [+53, +100], 10/10			
0.3	base (LFM2.5-230M-Chess-clean)	1355 [1295, 1405]	0/1	0.469	0.289
0.3	KAIKA θ	1453 [1421, 1485]	10/10	0.608	0.345
0.3	GRPO	1425 [1395, 1454]	10/10	0.591	0.294
0.3	RLOO (GRPO recipe)	1426 [1407, 1445]	4/5	0.536	0.350
0.3	REINFORCE (GRPO recipe)	1416 [1400, 1432]	4/5	0.570	0.306
0.3	paired difference (KAIKA − GRPO)	+29 [−5, +62], 8/10			
0	base (LFM2.5-230M-Chess-clean)	1400 [1340, 1455]	1/1	0.539	0.250
0	KAIKA θ	1459 [1437, 1481]	10/10	0.634	0.361
0	GRPO	1448 [1419, 1477]	9/10	0.609	0.332
0	paired difference (KAIKA − GRPO)	+11 [−29, +51], 5/10			

Engine-free comparator reproduction. The `pKout_batch` rerun reproduces `grpoout_lt1`: all 5 final checkpoint hashes and all 300 loss and KL values per seed agree, and the evaluation result files give the earlier means 0.5549 and 0.5567 at $T = 0.7$ and 0.3 (4 and 4 of 5 above parity). These are repeated outputs for the same seeds, not independent replications. Batch grouping was already the default, the informative-row filter has no shadow rows on this path, and the added KL setting leaves the traces unchanged. Both learners use outcome credit without engine supervision; KAIKA also keeps population-internal Hodge-derived pair labels, so objectives and compute differ. The remote checkpoint and game files were not available for independent local inspection. Table 19 reports an earlier engine-free KAIKA variant against a control without a KL anchor.

F.2 Chess component ablations and held-out openings

Member jitter scale. A birth adds σ_{mut} times standard Gaussian noise to each of the 6,750,208 low-rank coordinates, unscaled by the parent, so its norm is $\sigma_{\text{mut}}\sqrt{6{,}750{,}208}$: 2.6 at 10^{-3} and 77.9 at 0.03, against member norms of about 2.6. The full recipe and its pair-off arm run at 10^{-3} and the exact-copy arms at 0, and the engine-free arm of Table 1 at 10^{-3}. The full recipe run at 0.03 scores 0.4707 against the same recipe at 10^{-3} at $T = 0.7$, above parity on $1/5$ seeds; against GRPO it scores 0.5867 at $T = 0.7$ and 0.5313 at $T = 0.3$, above parity on $5/5$ seeds each. The scale acts on the population side of a comparison only; GRPO has no members. The Nim population (§A.5) and the learned-critic population arms below also realise 0.03 on every seed.

Learning-rate bracket. Against the recipe comparator at seed 2 and $T = 0.3$, KAIKA scores 0.4707, the only score below 0.5 in the learning-rate and β sweep; with each seed's two temperatures averaged, it scores 0.6130 at seed 1234 and 0.5173 at seed 2. The GRPO comparator's stated learning rate is the recipe value, not a selected optimum. Averaged over the 2 seeds, KAIKA's score share against GRPO at learning rates 2.5×10^{-6}, 5×10^{-6}, 10^{-5} (the recipe) and 2×10^{-5} is 0.6545, 0.5974, 0.6050 and 0.6453 at $T = 0.7$, and 0.5608, 0.5442, 0.5254 and 0.5640 at $T = 0.3$. A lower share means a stronger comparator: the strongest comparator sits inside the bracket, at 5×10^{-6}, at $T = 0.7$, and at the recipe rate at $T = 0.3$. Every two-seed mean is above parity.

Member-KL clamp and clone count. Without a clamp on the members' KL term, the largest member gradient norm reaches 1.7×10^{16} at seed 1234 and 3.3×10^{16} at seed 2, above 10^6 on 172 and 177 of 300 steps. Clamping that term at 8 (F1) removes the growth at both seeds (maxima

Table 19: An earlier engine-free KAIKA variant against an outcome-credit GRPO control without a KL anchor, training seeds 1234, 2345, 3, 4 and 5: 1,024 games per duel, temperature 0.7, top-p 0.95. W, D and L are the first-named seat's rates; the final column gives the score share of the policy named beside it. KAIKA carries three terms tied to the base (reference KL $\beta = 0.1$, pair NLL weight 0.1, base-KL cap 0.1) at learning rate 10^{-5}; the control trains with $\beta = 0$ and no reference model at 3×10^{-5} and scores below the base on every seed. Illegal-move rate 0.0 on every seat.

Duel (A vs. B)	W	D	L	Share	of
Seed 1234					
KAIKA θ vs. outcome-credit GRPO	0.7305	0.1230	0.1465	0.7920	KAIKA
Starting checkpoint vs. KAIKA θ	0.3262	0.1650	0.5088	0.5913	KAIKA
Starting checkpoint vs. outcome-credit GRPO	0.6113	0.1895	0.1992	0.2939	GRPO
Seed 2345					
KAIKA θ vs. outcome-credit GRPO	0.4863	0.1865	0.3271	0.5796	KAIKA
Starting checkpoint vs. KAIKA θ	0.3652	0.1445	0.4902	0.5625	KAIKA
Starting checkpoint vs. outcome-credit GRPO	0.4199	0.2090	0.3711	0.4756	GRPO
Seed 3					
KAIKA θ vs. outcome-credit GRPO	0.5801	0.1621	0.2578	0.6611	KAIKA
Starting checkpoint vs. KAIKA θ	0.3809	0.1514	0.4678	0.5435	KAIKA
Starting checkpoint vs. outcome-credit GRPO	0.5166	0.2051	0.2783	0.3809	GRPO
Seed 4					
KAIKA θ vs. outcome-credit GRPO	0.6885	0.1572	0.1543	0.7671	KAIKA
Starting checkpoint vs. KAIKA θ	0.3750	0.1719	0.4531	0.5391	KAIKA
Starting checkpoint vs. outcome-credit GRPO	0.6162	0.1709	0.2129	0.2983	GRPO
Seed 5					
KAIKA θ vs. outcome-credit GRPO	0.4658	0.1953	0.3389	0.5635	KAIKA
Starting checkpoint vs. KAIKA θ	0.3457	0.1826	0.4717	0.5630	KAIKA
Starting checkpoint vs. outcome-credit GRPO	0.4180	0.2188	0.3633	0.4727	GRPO
Seed 1234*, batch-matched control*					
KAIKA θ vs. batch-matched GRPO	0.7334	0.1494	0.1172	0.8081	KAIKA
Starting checkpoint vs. batch-matched GRPO	0.6699	0.1826	0.1475	0.2388	GRPO

Table 20: Chess round robin, five training seeds, 1,024 games per duel: the row policy's mean score against the column policy (seeds in brackets). RLOO and REINFORCE are the GRPO (GRPO) recipe with only the advantage estimator changed. KAIKA's seed-5 policy in the KAIKA–GRPO pairing is a second training copy of the same recipe.

$T = 0.7$	base	kaika (full recipe)	GRPO	RLOO (GRPO recipe)	REINFORCE (GRPO recipe)
base	–	0.289 [5]	0.388 [5]	0.400 [5]	0.405 [5]
kaika (full recipe)	0.711 [5]	–	0.612 [5]	0.634 [5]	0.619 [5]
GRPO	0.612 [5]	0.388 [5]	–	0.515 [5]	0.511 [5]
RLOO (GRPO recipe)	0.600 [5]	0.366 [5]	0.485 [5]	–	0.494 [5]
REINFORCE (GRPO recipe)	0.595 [5]	0.381 [5]	0.489 [5]	0.506 [5]	–

$T = 0.3$	base	kaika (full recipe)	GRPO	RLOO (GRPO recipe)	REINFORCE (GRPO recipe)
base	–	0.392 [5]	0.433 [5]	0.442 [5]	0.430 [5]
kaika (full recipe)	0.608 [5]	–	0.531 [5]	0.544 [5]	0.546 [5]
GRPO	0.567 [5]	0.469 [5]	–	0.493 [5]	0.497 [5]
RLOO (GRPO recipe)	0.558 [5]	0.456 [5]	0.507 [5]	–	0.506 [5]
REINFORCE (GRPO recipe)	0.570 [5]	0.454 [5]	0.503 [5]	0.494 [5]	–

Table 21: Paired factorial effects against same-seed GRPO (with minus without), over 1024 games for each of 5 training seeds. Seeds 1234, 3 and 4 of the full arm ran at an earlier commit; seeds 2 and 5 ran at the commit of the other three arms, where learning minus clone members with the pair term is -0.033 and $+0.009$ at $T = 0.7$ and -0.059 and -0.051 at $T = 0.3$.

	$T = 0.7$			$T = 0.3$		
paired effect (with minus without)	effect	[t95]	above 0	effect	[t95]	above 0
pair label, learning members	$+0.0458$	$[+0.020, +0.072]$	$5/5$	-0.0062	$[-0.052, +0.039]$	$2/5$
pair label, clone members	$+0.0689$	$[+0.029, +0.109]$	$5/5$	$+0.0062$	$[-0.016, +0.028]$	$4/5$
learning members vs clones, pair on	-0.0100	$[-0.030, +0.010]$	$1/5$	-0.0080	$[-0.071, +0.055]$	$2/5$
learning members vs clones, pair off	$+0.0132$	$[-0.016, +0.042]$	$3/5$	$+0.0043$	$[-0.033, +0.041]$	$4/5$
interaction (pair effect, learning minus clones)	-0.0231	$[-0.053, +0.007]$	$1/5$	-0.0123	$[-0.058, +0.034]$	$1/5$
clone members, pair off, minus 0.5	$+0.0531$	$[+0.034, +0.072]$	$5/5$	$+0.0333$	$[+0.013, +0.053]$	$5/5$

3.6×10^2 and 1.2×10^3). Against the same-seed GRPO at $T = 0.7$, F1 changes the share by -0.0220 at seed 1234 and +0.0122 at seed 2 (Table 22). F2 adds the elite-offspring transition to the clamp and F3 also a diversity term, so their changes combine clamp and transition. Direct same-seed duels of each clone arm's theta against the eight-clone arm's theta at $T = 0.7$ read 0.5200, 0.4844, 0.4795 and 0.4819 for the two-clone arm (seeds 1234, 2, 3, 4); 0.4932 for the sixteen-clone arm (seed 2), against a cross-seed eight-clone-versus-eight-clone yardstick of 0.4819 to 0.5049 (3 duels), with 3 of the 5 direct duels inside that band; against the same-seed GRPO at $T = 0.7$ the arms' shares are 0.6003 for two clones (mean of 4 seeds), 0.6221 for eight clones (mean of 5 seeds), 0.6143 for sixteen clones (one seed); at $T = 0.3$ they are 0.5003 for two clones (mean of 3 seeds, 1 above parity), 0.5395 for eight clones (mean of 5 seeds, 5 above parity), 0.5654 for sixteen clones (one seed, 1 above parity), all over 1,024 games per duel.

Table 22: Member-KL clamp ladder: largest member gradient norm over 300 steps, steps above 10^6, clamp activity (mean clamped-token share), and shares over 1024 games against the same-seed GRPO and against the unclamped full recipe.

								vs unclamped	
Arm	$\max \|g\|$	steps $> 10^6$	clamp steps (share)	GRPO T=0.7	Δ	T=0.3	Δ	T=0.7	T=0.3
unclamped full recipe (seed 1234)	1.7×10^{16}	172	–	0.6460	–	0.5801	–	–	–
F1 (member KL clamp 8, seed 1234)	3.6×10^2	0	240 (0.034)	0.6240	-0.0220	0.5386	-0.0415	0.4785	0.4590
F2 (clamp 8 + elite-offspring transition, seed 1234)	4.8×10^2	0	256 (0.026)	0.5464	-0.0996	0.5347	-0.0454	0.4111	0.4692
F3 (clamp 8 + transition + diversity $\lambda = 1$, seed 1234)	4.7×10^2	0	252 (0.020)	0.6069	-0.0391	0.5552	-0.0249	0.4038	0.4746
unclamped full recipe (seed 2)	3.3×10^{16}	177	–	0.5640	–	0.4707	–	–	–
F1 (member KL clamp 8, seed 2)	1.2×10^3	0	252 (0.032)	0.5762	$+0.0122$	0.5146	$+0.0439$	0.4839	0.4795

Held-out openings. From 128 opening lines disjoint from the eight training lines (Table 23; held-out lines start 6 to 12 plies deep, training lines at 4) the corrected θ scores 0.6264 at $T = 0.7$ (5/5 above parity) and 0.5459 at $T = 0.3$ (4/5), against 0.6121 and 0.5314 on the training lines; the paired held-out-minus-training difference at $T = 0.7$ is $+0.0143$ (95% CI $[-0.0049, 0.0335]$). Both are relative to the comparator; with no duel against the starting checkpoint on either bank, the held-out result does not establish improvement over the base.

GRPO given more steps. GRPO trained for 375 and 600 steps (1.25 and 2 times the population's 300; the median 300-step GRPO run already takes longer than the population's, App. Table 29) is

dueled against the same 300-step θ on seeds 1234 and 2. On seed 1234, θ scores 0.6460, 0.6099 and 0.5889 at $T = 0.7$ against GRPO at 300, 375 and 600 steps, and 0.5801, 0.5566 and 0.5327 at $T = 0.3$. On seed 2, θ scores 0.5479 and 0.5488 at $T = 0.7$ against GRPO at 375 and 600 steps, and 0.4893 and 0.4614 at $T = 0.3$, below parity. At $T = 0.7$ θ stays above parity on both seeds against GRPO given twice the steps; at $T = 0.3$ it is above parity on seed 1234 and below on seed 2.

Table 23: Held-out openings: score share of the corrected KAIKA θ against GRPO of the same seed on the eight training lines and on 128 held-out lines.

Seed	train, $T = 0.7$	held-out, $T = 0.7$	train, $T = 0.3$	held-out, $T = 0.3$
1234	0.6460	0.6606	0.5801	0.5908
2	0.5640	0.5591	0.4707	0.4897
3	0.6274	0.6636	0.5703	0.5654
4	0.5996	0.6196	0.5161	0.5234
5	0.6235	0.6289	0.5200	0.5601
mean	0.6121	0.6264	0.5314	0.5459

F.3 CHESS POPULATION ARMS

Table 24: Chess population arms. Scores are means over seeds; "seeds" counts seeds whose mean of the two GRPO scores exceeds 0.5. Member reads from the training artifacts: steps with no member write, pre-step pair margin (first half $\rightarrow$ last half of training), label agreement on re-rolled branches, fraction of position groups with $Z_k > 0$, trunk unchanged after founding.

arm	GRPO 0.7	GRPO 0.3	ref. 0.3	seeds	no-write	margin	agree	$Z>0$	trunk
shared θ, same-position outcome pairs	0.5658	0.5635	0.5037	3/3	0	$0.0030 \rightarrow 0.0106$	0.5183	0.99	–
per-member z-scored PG, cross-member DPO over step-pooled pairs	0.4609	0.5015	0.4219	0/1	0	$-0.0085 \rightarrow -0.0016$	–	–	–
per-member z-scored PG, step-pooled DPO, $\sigma_{\mathrm{mut}} = 0.001$	0.4785	0.5293	0.4619	1/1	0	$0.0010 \rightarrow -0.0000$	–	–	–
per-member z-scored PG, step-pooled DPO, founded ($\eta_f = 24.5$)	0.4629	0.4907	0.4395	0/1	0	$-0.0596 \rightarrow -0.0557$	–	–	–
shared θ, founded ($\eta_f = 24.5$)	0.5254	0.5337	0.4756	1/1	0	–	–	–	–
shared θ, founded ($\eta_f = 244.7$)	0.4673	0.5054	0.4316	0/1	0	–	–	–	–
reference (shared θ)	0.5522	0.5635	–	1/1	–	–	–	–	–

F.4 END-STATE GAMES

Table 25: End-state games: KAIKA's score against GRPO (1024 games per pairing; a game ended at the ply limit is a draw), mean over training seeds, 95% t interval over seeds, and seeds scoring above 0.5; the fractions of all games drawn and ended at the ply limit (counted as draws). Moves are scored among the legal moves (legal) or generated as free text (free); duels decode with HF Transformers or vLLM. Each training seed and temperature draws its games from its own block of environment seeds. Both arms of every Nim and Kuhn poker run here were trained before a fix to vLLM top-p sampling over legal moves, which sampled closer to greedy; on base-model positions at $T = 0.7$ it lowers policy entropy from 0.169 to 0.000 on Kuhn poker and from 0.257 to 0.111 on Nim.

		$T = 0.7$			$T = 0.3$				
Game	Moves, engine	Score	95% CI	> 0.5	Score	95% CI	> 0.5	Drawn	Ply limit
Nim	legal, HF	0.6598	[0.479, 0.841]	4/5	0.6568	[0.451, 0.862]	4/5	0.000	0.000

F.5 HELD-OUT BLOTTO TRANSFER AND POPULATION DIAGNOSTICS

The held-out cohort evaluates the central trunk, the highest-final-training-Elo non-θ member selected without held-out outcomes, and matched-seed GRPO at training seeds 1234, 2, 3, plus a base control. Each condition requests 128 greedy games arranged as 64 reversed-seat blocks against one fixed exogenous opponent tape. The uniform-stage tape is uniform over the condition's legal allocations, but it is fixed during evaluation and is itself exploitable: its canonical one-round best-response value is 0.329, versus 0 for an exact stage-game Nash mixture. The equal-split tape is one pure allocation, not a chance or equilibrium policy.

Table 26 reports `pair_score_mean`, the mean of win-minus-loss over complete two-seat blocks. This differs from `score_mean`, which averages all terminal games, whenever only one seat of a block terminates. For example, the budget-16 base control has terminal-game score 0.1261 but complete-block score 0.1364 after nine unpaired terminal games are excluded; the budget-24 values are 0.2598 and 0.2540 after one exclusion. All policy rows in Table 26 have 64 complete blocks. Their tape-relative intervals are individual-arm intervals, not intervals for a paired population–GRPO contrast: the required cross-arm block vectors were not retained.

LP serving uses member action distributions; archives pool probe snapshots; history controls fit one opponent leg and evaluate the other two. Across the 19 frozen-trunk runs, the served population mixture spans 0.141–0.315 held-out exploitability, compared with the uniform-play reference 0.329. On the evaluation instrument, the frozen-trunk, exogenous-only and trained-trunk served mixtures span 0.194–0.228, 0.181–0.260 and 0.344–0.372; GRPO spans 0.651–0.888 on that same instrument. Separately, leave-one-leg-out held-out LPs over GRPO histories span 0.433–0.535, with diet-matched values 0.322/0.502/0.956. For the trained trunk, separately constructed held-out served-mixture values are 0.338/0.200/0.359, own-θ history LP values are 0.259/0.285/0.268, and current-policy θ values on the evaluation instrument are 0.365/0.523/0.700, all ordered by seeds 309/311/313. Their prompt and sample constructions differ, so no per-seed dominance claim is made.

At the 8-member live size, the 9 probe runs span 0.203–0.294, below the random archive's fifth percentile 0.333 and median 0.514. Each run creates 48 members and pools 55–88 distinct member distributions across probe snapshots. The population archive LP reaches 0.082–0.155. The median over 200 archives of 48 random pure allocations is 0.051; this null matches the created member count, rather than the larger LP candidate count. Its best-eight greedy-plus-swap oracle median is 0.143. Among measurable members that moved, TV distance from birth at step 199 spans 0.174–0.967. Mean live-hull exploitability falls from 0.453 at step 0 to 0.277 at step 40 and remains within 0.238–0.277 thereafter.

The unchanged base measures 0.575 on the evaluation instrument and 0.546–0.646 on the probe; the maximum absolute gap, 0.071, remains unresolved.

Table 28 summarizes the admitted population and gate-withheld rows.

Table 26: Held-out nine-round Colonel Blotto complete-block scores against the fixed uniform-stage tape. K is the central trunk, M the training-Elo-selected non-θ member (slots 1, 6, and 5 for seeds $1234, 2, 3$), and G matched-seed GRPO. Budget 16 and 24 change the total units and therefore the action set to 153 and 325 allocations; $(2, 3, 4)$ uses the explicit weighted-payoff twin because the shipped environment stores but does not apply field values. Negative results are retained. † marks a request-stream invalidity rate above 0.2; its observed terminal score remains shown, while conditional stage diagnostics are withheld.

Training seed	Readout	Canonical	Budget 16	Budget 24	Weights $(2, 3, 4)$
1234	K (θ)	$+0.0781$	$+0.1016$	$+0.1406$	$+0.3438$
	M (1)	$+0.0781$	$+0.0703$	$+0.1406$	$+0.3438$
	G	$+0.0703$	$+0.1719$	$+0.2891$	-0.1328
2	K (θ)	$+0.0625$	$+0.1562$	$+0.2656$	-0.1250
	M (6)	$+0.0625$	$+0.1562$	$+0.2656$	-0.1250
	G	$+0.1875$	$-1.0000^{\dagger}$	$+0.2656$	-0.0781
3	K (θ)	$+0.1953$	-0.1562	$+0.2656$	-0.0781
	M (5)	$+0.1953$	-0.1875	$+0.2656$	-0.0781
	G	$+0.1406$	-0.1719	$+0.2656$	-0.1406

Table 27: Blotto population exploitability; lower is better. Each row names its evaluation, probe or held-out instrument. The trained-trunk rows report separately constructed served-mixture and history-LP held-out readouts and the current-policy θ on the evaluation instrument. Mixture and archive rows use the same LP meta-solver; random-archive entries report the stated quantile over independent draws.

Readout	Cohort	Exploitability
Frozen-trunk LP-served mixture, evaluation instrument	3 seeds	0.194–0.228
Exogenous-only served mixture, evaluation instrument	3 seeds	0.181–0.260
Trained-trunk served mixture, evaluation instrument	3 seeds	0.344–0.372
GRPO final single policy, evaluation instrument	3 seeds	0.651–0.888
LP over GRPO snapshots, leave-one-leg-out held out	3 seeds	0.433–0.535
Diet-matched GRPO snapshot LP, held out	seeds 309/311/313	0.322/0.502/0.956
LP over GRPO seed policies	3 policies	0.532
Live-hull LP, 8 members, step 199, probe	9 runs, 3 cohorts	0.203–0.294
Snapshot-archive LP (55–88 candidates)	9 runs, 3 cohorts	0.082–0.155
Unselected random-8 LP, 5th pct./median	1000 draws	0.333 / 0.514
Created-member-count-matched random-48 LP, median	200 draws	0.051
Oracle best-8-of-48 random, median	30 draws	0.143
Trained-trunk readouts, separately labelled by instrument		
Served mixture, separately constructed held out	Seed 309	0.338
Own θ history LP, separately constructed held out	Seed 309	0.259
Own θ, evaluation instrument	Seed 309	0.365
Served mixture, separately constructed held out	Seed 311	0.200
Own θ history LP, separately constructed held out	Seed 311	0.285
Own θ, evaluation instrument	Seed 311	0.523
Served mixture, separately constructed held out	Seed 313	0.359
Own θ history LP, separately constructed held out	Seed 313	0.268
Own θ, evaluation instrument	Seed 313	0.700

The sampled follow-up contains ten files and 100 arm cells. Generation seed 0 covers the three independently trained populations; seven additional decoder seeds, $1, \ldots, 7$, re-evaluate only the fixed seed-1234 population and are nested evaluation-RNG variation, not training replicates. Across all eight decoder seeds, every seed-1234 population slot retains support one, entropy zero, conditional stage BR 1.0, and the same terminal score. Seed 3 repeats this collapse. For seed 2, all eight population slots are present in the checkpoint contract but each records 128 legal and 128 unparsed requests, 128 forgiven retries, and no terminal invalid game. The native one-error rule implies an invalid-first, legal-retry sequence, but ordered response texts were not retained; that ordering is

Table 28: One-round population readouts against the fixed uniform-stage tape. Scores are complete-block terminal scores. Population support and entropy describe the seven-member equal-weight marginal; mean TV is over its 21 member pairs; BR is its exact conditional stage-game best-response value. Hodge entries give gradient/curl energy fractions and total squared energy E for the estimated endpoint-policy payoff matrix, conditional on the empirical marginals. Fractions are undefined and omitted when $E = 0$. "Gate" means all eight population slots exceeded the request-stream invalidity threshold, so no survivor-renormalized diversity, mixture, or Hodge result is reported. † retains a terminal score from an arm that crossed that gate.

Training seed	Decode	θ score	GRPO score	Status	Support	Entropy	Mean TV	Mixture BR	Hodge grad/curl
1234	Greedy	$+0.2656$	-0.0938	admitted	1	0	0	1.0	$1/0\ (E = 16)$
2	Greedy	$+0.0156^{\dagger}$	$+0.0625$	gate		withheld			gate
3	Greedy	$+0.0625$	$+0.0625$	admitted	1	0	0	1.0	$E = 0$
1234	Sampled, gen. 0	$+0.2656$	-0.2109	admitted	1	0	0	1.0	$1/0\ (E = 8)$
2	Sampled, gen. 0	$+0.0156^{\dagger}$	$-0.9219^{\dagger}$	gate		withheld			gate
3	Sampled, gen. 0	$+0.0625$	-0.0312	admitted	1	0	0	1.0	$E = 0$

source-derived rather than witnessed. The unchanged request-stream invalidity gate is $0.5 > 0.2$, so complete membership is not complete admission and no legal-survivor mixture is substituted.

In this historical nine-round cohort (Table 28; not the frozen-trunk, trained-trunk, live-hull or snapshot-archive readouts of Table 27), for the two admitted populations, both seven-member and all-eight equal-weight marginals are the same pure action. Their best-response value is 1.0 with whole-game bootstrap interval $[1, 1]$, their improvement over the best constituent is zero, and their TV distance from the exact Nash mixture, which has support 13, is 1.0 with no support overlap. These are failures to obtain behavioural diversity or mixed-strategy protection, not evidence that the learned policies are Nash. The exact stage calculation does not measure exploitability of the history-conditioned nine-round policy.

G Compute and Reproducibility

G.1 Reproducibility details

The archive beds (the Lichess decade, its $16\times$-budget follow-up, the chess self-play loop, coevolution drift, the capacity study and the payoff-league comparison) sample parents with replacement. The release exposes this as the option `Archive(parent_sampling="with_replacement")`, whose default is `"distinct"`, and the paper's archive runners set it. With this setting the release reproduces the Lichess decade exactly (100/100 per-arm rows; anti-collapse KAIKA minus vanilla 0.0293, Holm-adjusted 0.0315, with its prespecified check passing), coevolution drift (125/125 per-seed scalars) and the payoff-league comparison (20/20 metric pairs). The Lichess preregistration's pinned hash is updated by a dated amendment that keeps the original pin. The two Lichess runners pin NumPy 2.5.1 and run in a separate environment; every other command uses the locked requirements. The Lichess follow-up, the capacity study and the chess self-play loop reproduce all but a few rows at the release (99/100, 68/72 and 10/12). None of the differing rows involves parent sampling: the four capacity scalars that differ belong to one oracle-league run whose artifact did not reproduce at its own original commit either, and the other three rows reproduce at their original commits but not after later code changes. The README lists each command with the check it prints.

The archive and chess-kernel studies ship seeded CPU runners (the LLM-agent harness makes external calls) and per-seed JSON artifacts indexed by a claim index; the two provenance-only exceptions are the screen's calibration magnitudes and one development-loop crossing. Each frozen-preregistration study ships its preregistration document, public-beacon block derivation, void-attempt history and independent analyzer output; the factorials pin their runner SHA-256 in their

Table 29: Compute per 300-step training run: median GPU-hours over seeds, with the median wall seconds per step in parentheses. Each run trains on one NVIDIA H100 80GB of the campaign node; GPU-hours are the artifact's wall clock. The engine-credit chess step is bound by serial Stockfish evaluations, not by the GPU. Seeds per arm: chess engine credit KAIKA 10, GRPO 10, RLOO 5, REINFORCE 5; chess without engine KAIKA 5, GRPO 5; language games 5 per arm. The policy columns are KAIKA and GRPO on chess, and KAIKA-H and GRPO on the language games.

		GPU-hours (s/step)			
Bed	Policy	KAIKA	Batch PG / GRPO	RLOO	REINFORCE
Chess, engine credit	LFM2.5-230M-Chess	2.22 (26.0)	2.75 (27.5)	2.41 (24.8)	2.49 (25.8)
Chess, no engine, 64 rows	LFM2.5-230M-Chess	2.34 (27.3)	1.33 (12.1)	–	–
Kuhn poker	Qwen2.5-1.5B	0.56 (6.0)	0.21 (2.4)	0.21 (2.4)	0.21 (2.5)
Nim	Qwen2.5-1.5B	0.55 (5.7)	0.23 (2.5)	0.26 (3.0)	0.22 (2.2)
Kuhn poker	Qwen2.5-3B	1.02 (11.0)	0.54 (6.3)	0.54 (6.3)	0.54 (6.2)
Nim	Qwen2.5-3B	1.04 (11.1)	0.57 (6.6)	0.49 (5.6)	0.54 (6.3)

preregistration documents and assert Python 3.12 / NumPy 2.5.1 at runtime. Amendment histories sit beside the code; RFC-3161 timestamps cover the later code freezes (earlier ones are git-ordered only). Re-execution is the primary check, with robustness across the knob grid; preregistration certifies only that designs and decision rules preceded outcomes. Seven attempts were voided pre-unblinding by mechanical gates and quarantined unopened, across four studies. Two defects found by post-hoc audit were repaired: one study's never-executed analyzer equivalence branch (erratum with a zero-freedom recompute) and another study's bed module, verified at run time but left uncommitted (bit-identical clean-clone check). The guarantees are as strong as this audit trail.

Neural-game tables are rebuilt from retained run result files; the original language-model slate saved no weights, so its later checkpoint-saving repetitions support the head-to-head evaluations, not an exact replay of the original policies. `tables/chessmodel_assisted.json` holds the earlier assisted configuration's per-seed scores, evaluation settings and immutable source pointers. Retraining from a seed and reproducing a reported table are distinct checks.

G.2 Compute

Compute. Table 29 gives the cost of one 300-step training run for every arm. Each run uses one GPU, so GPU-hours equal the artifact's wall clock.

H The KAIKA archive

Retention under changing feedback. MAP-Elites preserves behavioural variety (Mouret & Clune, 2015); Dynamic MAP-Elites tracks reward drift (Gallotta et al., 2024). Quality-Diversity through AI Feedback (QDAIF) uses AI-scored quality and diversity (Bradley et al., 2024), while Quality Diversity through Human Feedback (QDHF) learns descriptors from human similarity feedback (Ding et al., 2024). The archive studies below compare retained coverage and serving quality against vanilla and Dynamic MAP-Elites and an implementation of NS-DPO (Son et al., 2025). Hodge decomposition separates additive ratings from cyclic comparison structure (Jiang et al., 2011; Balduzzi et al., 2018). The archive uses the residual as a descriptor; the decomposition itself is prior work.

H.1 The archive model

Comparison structure. The competitive studies represent pairwise advantage by a time-varying skew-symmetric kernel,

$$A_t(x, y) = -A_t(y, x), \qquad \Pr_t(x \succ y) = \sigma(A_t(x, y)) \,, \tag{18}$$

with logistic link σ for binary comparisons, so A_t is a logit and the expected signed payoff is $M_t(x, y) = 2\sigma(A_t(x, y)) - 1$. In the scalar-rated special case, $A_t(x, y) = s_t(x) - s_t(y)$; Bradley–

Terry worths correspond to $s_t = \log w_t$ (Bradley & Terry, 1952). General kernels also express cycles. Hodge decomposition separates additive ratings from residual comparison structure (§H.2.5). The structured successor's minimax serving solves the zero-sum game on an estimated payoff matrix $\hat{M}$, a model payoff blended with the pair's empirical signed outcomes and clipped to $[-1, 1]$; the neural Blotto LP uses the members' action distributions.

The KAIKA archive. The archive studies of §H.2 use a separate, gradient-free learner. Online Bradley–Terry/Elo standings (Bradley & Terry, 1952; Elo, 1978) estimate comparative quality from pairwise outcomes; a MAP-Elites grid (Mouret & Clune, 2015) keeps one lineaged elite per behaviour-descriptor niche, with retained variety as the objective (Ashby, 1956); and retained elites are drawn as self-play opponents (Silver et al., 2018), so the opponent distribution changes with the population. The standings control retention, pairing and serving without backpropagating into candidate parameters. A structured successor adds regime detection, per-regime low-rank kernel estimates, founder protection, active pairing and minimax serving.

H.2 LEARNING, MEMORY AND CURRICULUM IN THE KAIKA ARCHIVE

This section tests the archive on candidate sets: whether accumulated comparisons predict better after change (§H.2.1), whether bounded populations retain more alternatives and convert capacity into coverage (§H.2.2–H.2.3), whether a live population works as a curriculum (§H.2.4), and whether real chess comparisons carry structure that additive ratings discard (§H.2.5). The appendix reader's guide indexes the protocols.

H.2.1 ACCUMULATING COMPARATIVE EXPERIENCE

After a hidden preference changes, the accumulated comparison standard improves held-out accuracy over a restart by 0.0336 during early adaptation (95% CI $[0.0261, 0.0410]$, 110 usable blocks). On the chess calibration ladder, duel–Elo recovers the hidden strength order with Spearman 0.997, versus 0.914 for a scalar probe that rates each candidate by its score against a single fixed median-strength anchor (5 seeds). This ranking reverses under unbiased noisy scalar feedback in all 18 tested conditions. On the PersonalLLM pool (Zollo et al., 2024), the accumulated standard improves serving quality by 0.114 over random with design-of-experiments (DoE) memory (McKay et al., 1979) and the same archive structure.

H.2.2 RETAINING ALTERNATIVES UNDER CHANGE

On public drifting preferences, the KAIKA archive improves mode retention over a binary-comparison adaptation of Dynamic MAP-Elites (Gallotta et al., 2024) by $+0.399$ at matched comparison and memory budgets (96 blocks; one-sided exact sign-flip test, $p \approx 2 \times 10^{-20}$), and by $+0.108$ ($[+0.039, +0.176]$) when Dynamic MAP-Elites also draws the same random proposals. In this study the archive retains and serves from a bounded duel posterior, without inherited Elo standings. A larger study finds retention gains of 0.072 over NS-DPO (95% CI $[0.0375, 0.1066]$, 384 blocks, Holm $p = 3.3 \times 10^{-5}$) and 0.075 over vanilla MAP-Elites (Holm $p = 6.8 \times 10^{-6}$). NS-DPO receives full reward-model vectors for gradient fitting, while the archive receives binary outcomes; NS-DPO serves 0.083 better (§7).

On observed *Age of Empires IV* matchups from AoE4 World (AoE4 World, 2026), Nash-support retention improves by 0.0617 over DEX3.S (Saha & Gupta, 2022) at its published tuning (95% CI $[0.0212, 0.1022]$, 20 blocks, Holm $p = .018$). On 227 Lichess chess openings over 120 months (Lichess, 2026b;a), the anti-collapse gain over vanilla MAP-Elites is 0.029 (Holm $p = .032$), and $+0.012$ in the deeper $1{,}408$-line universe.

In a shared-descriptor study, served exploitability is 0.021 for the KAIKA archive versus 0.084 and 0.067 for two payoff-solving leagues (lower is better). In discovery, against random with DoE, open-support coverage improves by 0.0875 (20 blocks).

H.2.3 CONVERTING CAPACITY INTO COVERAGE

The fresh-seed capacity study measures regional uplift $A = G_H - G_L$: the gain per capacity doubling in the higher-capacity region minus that in the lower region (a rate per doubling, that is per

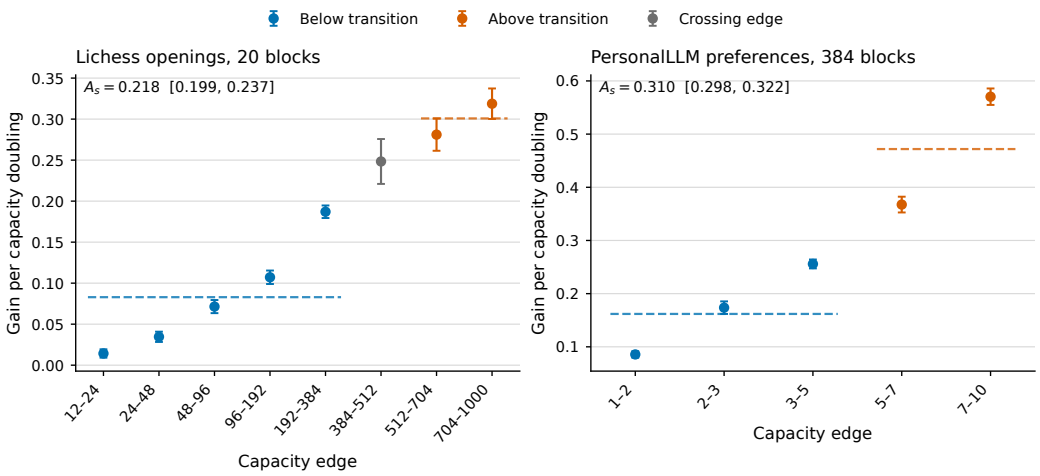

Figure 13: Conversion gain per capacity doubling on chess openings and public preferences. Points are means across blocks; whiskers are descriptive 95% t intervals. Dashed segments show the log-capacity-weighted regional means, whose difference defines uplift. The transition-crossing Lichess edge is excluded from both regions.

unit of log capacity, not per added slot). Uplift is 0.2179 on Lichess (95% CI $[0.1992, 0.2366]$, 20 blocks) and 0.3100 on PersonalLLM ($[0.2980, 0.3221]$, 384 blocks; Figure 13). Both exceed their prespecified domain margins (Holm $p \leq 5 \times 10^{-6}$). A separate SpinningTop contrast measures *between-system* acceleration: the KAIKA archive exceeds the fair same-information league by 0.2354 ($[0.2271, 0.2438]$, 320 blocks, Holm $p = 1.5 \times 10^{-5}$). Against the fair sample-PSRO league, the archive has lower coverage regret (one minus the best retained candidate's true value) in all 12 SpinningTop cells, and in $9/12$ with matched caps.

H.2.4 Learning against an evolving population

On a real chess policy space, a live archive of multilayer-perceptron (MLP) heads beats its frozen snapshot in 14 of 16 matched comparisons (Wilcoxon $p = 8.4 \times 10^{-4}$; mean $+0.1286$). Fixed-teacher calibration separately finds saturation at 258–683 Elo above the teacher caps. In the synthetic Elo-logistic match model, the retention advantage is zero at transitivity by construction and grows with non-transitivity (0.458 versus 0.280 for naive retention at $\beta = 2$). Self-play reaches 3307 Elo, anchored to the Stockfish calibration ladder of §H.2.1, at 260 generations and is still improving at the cutoff.

H.2.5 Relational structure in real chess

Projecting engine self-play comparisons onto additive-rating flows leaves a residual of 0.114 of total flow energy, against 0.082 under a fitted Davidson null (additive per-seed abilities plus colour and draw terms; Davidson, 1970), 3.2 points; the excess relative to the null's mean residual is 0.3828 across 60 shards (Monte Carlo $p = 10^{-4}$). On the natural chess kernels, Hodge curl carries 0.53–0.58 of the squared comparison-flow energy per regime on these banks, compared with 0.25 after uniform regime pooling and 0.10–0.15 on the synthetic planted beds, whose equilibria are exact by construction. The residual carries information that additive ratings discard.

In a 12-seed semi-natural screen, revealing the regime gives return-regime exploitability 0.320 (lower is better). Removing protection worsens it by 0.040 on $11/12$ seeds, the scalar estimator-and-serving package by 0.068 on $11/12$, and passive pairing by 0.048 on $9/12$. On one fully natural development bank (a fixed chess-policy set with an exhaustively measured pairwise kernel), the structured archive improves from 0.610 at 16,384 labels to 0.511 at 65,536; the exact-kernel blind-Nash reference is 0.369, best-fixed is 0.500 and uniform play is 0.648. The separate five-arm factorial compares per-regime with pooled mElo-class estimation: mean $+0.0228$ on 12 seeds, one-sided 80% lower bound -0.0047 against a 0.05 criterion, which its sole primary does not meet.

Inherited standing cuts served exploitability from 0.305 to 0.050.

Limitations of the archive studies. On PersonalLLM, NS-DPO serves 0.083 better than the KAIKA archive, and the archive minus random with DoE memory is -0.028 (95% CI $[-0.098, 0.042]$) on contextual retention. On the $1{,}408$-line Lichess universe, uniform proposals keep 0.019 more family entropy (Holm $p = .0002$). A same-information league with twice the memory is equivalent within ± 0.010 on retention. The fair sample-PSRO league has lower coverage regret in all 12 Blotto cells and, on the earlier capacity endpoint, scales faster ($+0.0134$). In the closed world, panel-ranking AUC and serving quality are 0.035 and 0.119 below random with DoE (10 blocks). Plain Bradley–Terry learning exceeds QD retention in the weak-to-strong test at every tested $\gamma > 0$.