INTERFACE-CONTROLLED EVALUATION AND SIZE TRANSFER OF LANGUAGE-MODEL EQUILIBRIUM SOLVERS ON PROCEDURAL ZERO-SUM MATRIX GAMES

Anonymous authors
Paper under double-blind review

ABSTRACT

We study language models as tool-free equilibrium solvers that read a serialized payoff matrix and write a strategy pair, and ask whether they can learn to solve unseen zero-sum games and whether named-game evaluations of such models measure the game or the prompt. Every answer is scored by an exploitability certificate that needs no equilibrium label. First, in an interface-controlled audit of Qwen3.5-9B on ten named base games with 12,600 released observations, the correct-label advantage is 24.6 points under raw prompting but 1.6 points under a matched chat template, and the interaction stays positive when base-game identities are resampled. Second, training on 2×2–3×3 games transfers to unseen sizes under a fixed short-answer protocol: at 5×5, the best-of-four success rate is 61% for LP-label SFT, 37% for a residual-reward GRPO recipe that uses no equilibrium labels, 16% for a chat-template base that answers validly in 93% of samples, and 15% for a format-only SFT control; SFT reaches 33% at pass@1. Third, the dominated-action padding control used to probe size effects needs calibration: a payoff-blind policy passes 45 of 50 padded 12×12 games under range-normalized scoring but 6 of 50 at the embedded game's precision. We release the records, the calibrated control, and the scoring code.

1 INTRODUCTION

Can large language models compute equilibria of matrix games, and can an evaluation distinguish that computation from prompt-interface effects? LLMs prompted with structured few-shot chain-of-thought demonstrations choose best-response actions in classic games such as the Prisoner's Dilemma and Chicken with high accuracy (Gandhi et al., 2023), while systematic analyses find persistent gaps in formal strategic reasoning (Fan et al., 2024). Named-versus-anonymous comparisons, however, often change semantic cues, prompt wrappers, output contracts, and matrix classes at once.

What the task measures. Zero-sum matrix games are solvable by linear programming, and a model with tool access should call a solver; simple iterative solvers also pass every evaluation game within 32 iterations, as Section A.4 shows. Our interest is measurement and learning, not numerical efficiency, and we study the tool-free, direct-answer setting for two reasons. First, named-game evaluations of LLM strategic behavior score answers in this setting, so whether their named-versus-anonymous and small-versus-large comparisons measure the game is a question about this protocol. Second, the LP structure gives every answer an exact certificate that needs no label, which makes interface audits, format controls, and padding calibrations checkable. Our measurements therefore concern computing an equilibrium in context from a displayed matrix, not strategic reasoning in natural-language interactions.

We first separate a cue manipulation from its interface. In an interface-controlled greedy pass@1 audit of Qwen3.5-9B over ten named base games and three affine grids, the correct-label minus anonymous effect at 1,024 tokens is 24.6 percentage points under raw prompting and 1.6 points under the matched chat interface; the 23.0-point interaction remains positive when the ten base-game identities, rather than cases, are resampled, and under raw prompting any label, correct or wrong,

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.

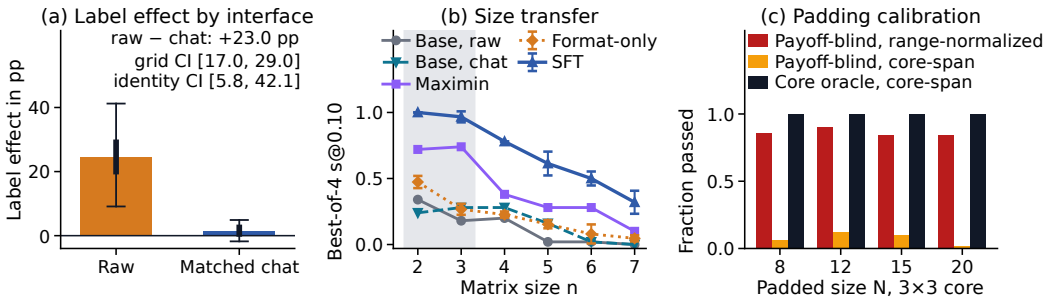


Figure 1: **Interface audit, size transfer, and padding calibration. (a)** Correct-label minus anonymous pass@1 effect under raw and matched-chat prompting at a 1,024-token budget, with 900 paired cases per interface. Thick bars: 95% CIs from the grid-then-case bootstrap; thin bars: CIs from resampling the ten base-game identities. **(b)** SFT trains on the shaded sizes 2×2–3×3; the curve shows the mean and sample s.d. of verifier-selected best-of-four success across three training seeds on 50 fixed games per size. The base model is shown under the raw prompt and under its chat template on the same games and decoding, with the format-only SFT control trained on random valid targets with three seeds; maximin is the pure maximin strategy. Table 2 lists all rows, including VERGE. **(c)** Reference policies on dominated-padding games, 50 per size: a payoff-blind policy that plays uniformly on the first three actions, scored with the range-normalized residual of the padded matrix and with the core-span residual, and the core oracle, which pads the embedded game's exact solution with zeros and passes every game under both scores.

raises the rate of parseable answers by 25–38 points, as Figure 1 and Section 4.1 show. All 12,600 observations are released with a self-contained recomputation script.

This separation leaves the question of computation: *can LLMs learn to compute equilibria on newly sampled matrices?* Every training step presents a freshly sampled payoff matrix, and every answer is scored by an exact exploitability certificate computable from the matrix and the answer alone. We compare two learning signals:

- **SFT** (Supervised Fine-Tuning): the model is trained on payoff matrices paired with their Nash strategies, computed by linear programming.
- **VERGE** (Verifiable Equilibrium-Regret GRPO): the model receives only an exploitability reward, a scalar certificate measuring distance from Nash equilibrium, with no access to the equilibrium itself.

After training exclusively on 2×2–3×3 games, verifier-selected best-of-four success, meaning normalized exploitability below 0.10, at 5×5 is 61% for SFT and 37% for VERGE, and pass@1 is 33% and 21%. The comparison that matters is a base model that already answers in the required format: under its chat template, on the same games, prompt text, and decoding, it answers validly in 93% of 5×5 samples and reaches 16%, or 6% at pass@1. Game by game, SFT also solves every 5×5 game that the payoff-reading maximin rule solves, and 21 more.

Size effects are often probed by embedding a small game in a large matrix through dominated actions, which leaves the equilibrium unchanged. We show that this control needs calibration: a payoff-blind policy that always plays the first three actions uniformly passes 45 of 50 padded 12×12 games when precision is normalized by the padded payoff range, but only 6 of 50 when it is normalized by the embedded game's range, whereas the embedded game's exact solution passes all 50 under both scores, as Figure 1c and Section 4.3 show. We release the calibrated control, with the embedded block at a random position and core-span scoring.

Contributions. For a language-model solver scored by one label-free, shift- and scale-invariant certificate, we provide:

1. a fully released audit showing that the named-game label effect is interface-conditioned, with identity-level uncertainty, in Section 4.1;
2. size transfer of LP-label SFT and of a label-free residual-reward recipe from 2×2–3×3 to 7×7, ahead of a format-competent chat-template base and a format-only SFT control on the same

protocol and, for SFT, of an LP-label prior and, with best-of-four, of maximin on paired games, in Section 4.2;
3. a calibration showing that dominated-action padding must be scored at the embedded game's precision, where a payoff-blind policy passes 45/50 versus 6/50, with a released control in Section 4.3.

Theorem 1 bounds how display rounding moves the score, and Theorem 2 shows that role-merged GRPO carries no on-policy gradient, which motivates our cooperative residual reward.

2 Related Work

Learned equilibrium solvers. Neural equilibrium solvers operate on payoff tensors: Duan et al. (2023a) train Nash approximators with a label-free maximum-regret loss, Marris et al. (2022) train label-free equivariant solvers whose parameters are independent of game size and transfer zero-shot to larger games, NfgTransformer (Liu et al., 2024) minimizes the NE gap with an equivariant transformer, Duan et al. (2023b) study when equivariant approximators help, and Wu & Lisser (2022) regress LP saddle-point values of zero-sum games with a CNN. Our learner is instead a pre-trained language model that reads a serialized matrix and writes a JSON strategy pair. Interface, numeric rendering, and output length, which these architectures fix by design, thereby become measurable variables, and a format-competent base model becomes the natural comparator in Section 4.2. Theorem 1 adapts the Lipschitz argument of Duan et al. (2023a) to zero-sum NashConv to bound how display rounding moves the score.

LLMs on matrix games and prompt sensitivity. Evaluations of LLM strategic behavior (Gandhi et al., 2023; Fan et al., 2024; Duan et al., 2024; Wang et al., 2024; Zheng et al., 2025b) mostly use named or story-framed games. Several works modify familiar games. Lorè & Heydari (2024) contrast game structure with contextual framing, Silva (2025) find that mixed-strategy play degrades under slight modifications, and Georgousis et al. (2026) alter the payoffs and action labels of the Prisoner's Dilemma and Rock–Paper–Scissors. Lekeas & Stamatopoulos (2026) trace a late-layer override of the Nash action. None of them crosses the label cue with a raw versus chat interface on the same variants. More generally, meaning-preserving format changes can move LLM accuracy by tens of points (Sclar et al., 2024). Our audit crosses the two factors. On the same ten base games and payoff-preserving variants, the correct-label minus anonymous effect is 24.6 points under raw prompting and 1.6 under a matched chat template, as Section 4.1 reports. The named-versus-anonymous gap, which is read as evidence of memorized game knowledge, is thus itself interface-conditioned. Concurrently, a revised audit finds that parser truncation and framing change game-theoretic LLM measures (Pechon-Elkins & Chun, 2026), and GENSTRAT (Shadarevian et al., 2026) generates zero-sum card games procedurally to avoid contamination.

Training LLMs on games. SPIRAL (Liu et al., 2026) and MARSHAL (Yuan et al., 2026) train LLMs by self-play on games to improve general reasoning, GIFT (Lyu et al., 2026) applies GRPO to nine classic matrix games with a pure-action Nash reward and payoff sign flips and ± 100 offsets as augmentation, Zhou et al. (2025) compare SFT with RL from verifiable rewards on curated economics problems, and the concurrent Wang et al. (2026) compress poker solver mixed strategies into decision-tree rules that are given to frozen LLMs in the prompt. The first four train on fixed games or problems and report downstream ability. We sample a fresh matrix at every step and size, score mixed-strategy pairs with an exact certificate, and compare LP-label SFT with a residual-reward GRPO variant (Shao et al., 2024) under one generator and output protocol, and we evaluate transfer to sizes never seen in training. Theorem 2, proved in Section E, concerns role-merged scoring of one output in both roles; role-conditioned self-play such as SPIRAL and population methods such as PSRO (Lanctot et al., 2017) lie outside its scope.

3 Method and Theory

3.1 Procedural Game Testbed

We study two-player zero-sum matrix games with n actions per player, defined by a payoff matrix $A \in \mathbb{R}^{n \times n}$, where the row player receives A_{ij} and the column player receives $-A_{ij}$ when row i and

column j are played. A *mixed strategy* is a probability distribution over actions: $\mathbf{p} \in \Delta^n$ for the row player, $\mathbf{q} \in \Delta^n$ for the column player.

Definition 1 (Exploitability / NashConv)**.** *For strategies* $(\mathbf{p}, \mathbf{q})$ *in a zero-sum game with payoff matrix* A*, the* exploitability *is:*

$$\begin{aligned}\mathrm{Exploit}(A, \mathbf{p}, \mathbf{q}) = &\underbrace{\max_i (A\mathbf{q})_i - \mathbf{p}^\top A\mathbf{q}}_{\textit{row regret}} \\ &+ \underbrace{\mathbf{p}^\top A\mathbf{q} - \min_j (\mathbf{p}^\top A)_j}_{\textit{column regret}}.\end{aligned} \tag{1}$$

$\mathrm{Exploit} = 0$ *if and only if* $(\mathbf{p}, \mathbf{q})$ *is a Nash equilibrium. We report average per-player regret normalized by the payoff range:* $\overline{\mathrm{Exploit}} = \mathrm{Exploit}/(2(\max A - \min A))$*, and define the* exploitability reward *as* $r = 1 - \overline{\mathrm{Exploit}}$*. Thus* $\overline{\mathrm{Exploit}} < 0.10$ *is equivalent to NashConv below* 20% *of the payoff range; for a valid strategy in a nonconstant zero-sum game,* $\overline{\mathrm{Exploit}} \in [0, 1/2]$*. Success is the strict inequality* $r > 0.90$*, evaluated on the unrounded matrix with the numerical conventions of Section B.1. The normalized residual* $\overline{\mathrm{Exploit}}$ *is invariant to positive rescaling and additive shifts of* A*, which preserve equilibria. Invalid or unparsable outputs receive reward* -0.25*.*[1]

Game generation. Each training step samples a fresh $n \times n$ matrix with integer entries from $[-9, 9]$ and rescales it to payoff span 2 by $A \leftarrow 2A/(\max A - \min A)$, without re-centering the entries. Training uses $n \in \{2, 3\}$; evaluation extends to $n \in \{2, \ldots, 7\}$. Training matrices are drawn independently of the fixed evaluation sets. Evaluation games at $n \geq 4$ are therefore new instances, since the integer support has 19^{16} matrices at 4×4, whereas at 2×2, with 19^4 matrices, exact coincidences with training draws are possible, which is one reason we treat 2×2–3×3 as in-distribution. None of the 200 fixed evaluation games at 2×2–5×5 is isomorphic to a catalogued named zero-sum game or skew-symmetric, and beyond the training sizes most games have no pure-strategy saddle point: one occurs in 62%, 50%, 10%, 8%, 4%, and 0% of the games at $n = 2, \ldots, 7$, as detailed in Section A.5.

Prompt and output format. The model receives the matrix with entries printed to one decimal, while LP labels and scores use the unrounded matrix. Each displayed entry is within 0.05 of the scored one, so by Theorem 1 the normalized residual of any answer moves by at most 0.025, and scoring baselines or saved full responses on the displayed matrix changes at most 4 of 50 flags per arm and size, as Section B.1 reports. The model outputs a JSON object whose two fields list the mixed strategies of the row and the column player; the verbatim prompt appears in Section B. The prompt states the game and the output contract but does not name the solution concept; the equilibrium target enters only through the SFT labels and the VERGE reward, so training conveys both which strategy pair is scored and how to compute it. Strategies are normalized to the probability simplex after parsing. Emitting both strategies keeps the certificate computable from the response alone, without a chosen opponent, since NashConv is defined on a pair.

Cooperative exploitability formulation. In both training methods, SFT and VERGE, the model proposes a complete strategy pair $(\mathbf{p}, \mathbf{q})$ per turn, receiving the exploitability reward $r = 1 - \overline{\mathrm{Exploit}}(A, \mathbf{p}, \mathbf{q})$: a single-agent cooperative formulation in which each response is scored independently on its distance to a Nash equilibrium. Role-merged self-play, where each response is scored as both row and column player, cancels the on-policy score-function gradient by Theorem 2; the cooperative formulation avoids this by construction.

3.2 Two Teaching Methods

LP-label SFT. For each generated game, we compute the Nash equilibrium, which exists by the minimax theorem, using linear programming. The model is trained with cross-entropy loss on matrix–equilibrium pairs, using LoRA with $r = 32$ and $\alpha = 64$ on the query, key, value, and output projections of the full-attention layers and on the MLP projections. Each run interleaves 2×2 and 3×3 games within shuffled minibatches.

[1]When $\max A = \min A$, a constant matrix, we perturb one entry by $+1$ to avoid division by zero; see Section B.

VERGE. The model generates strategy proposals and receives the exploitability reward $r = 1 - \bar{\text{Exploit}}$ as feedback. No Nash equilibrium labels are provided. Each outer step samples two games, each of a size drawn uniformly from $\{2, 3\}$, with $G = 8$ responses per game, and normalizes the 16 rewards jointly into advantages $\hat{a}_k$. With $\bar{\ell}_\theta(y_k)$ the mean token log-probability of response y_k given its prompt, the per-response loss is

$$\begin{aligned}\mathcal{L}_k = & -\min\big(\rho_k \hat{a}_k,\ \text{clip}(\rho_k, 0.80, 1.28)\,\hat{a}_k\big) \\ & + 0.02\,\big(\bar{\ell}_{\text{old}}(y_k) - \bar{\ell}_\theta(y_k)\big),\end{aligned} \tag{2}$$

with $\rho_k = \exp\big(\bar{\ell}_\theta(y_k) - \bar{\ell}_{\text{old}}(y_k)\big)$, a sequence-level ratio of length-normalized likelihoods as in GSPO (Zheng et al., 2025a) rather than GRPO's per-token ratio, with the asymmetric clip range of DAPO (Yu et al., 2025). The anchor $\bar{\ell}_{\text{old}}$ is recomputed by a forward pass after sampling and held fixed, so the second term has gradient $-0.02\,\nabla_\theta \bar{\ell}_\theta(y_k)$ and raises the likelihood of every sampled response. The 16 responses are processed in micro-batches of two with four-step accumulation, giving two optimizer updates per outer step, as described in Section C.1. This objective differs from standard GRPO in its ratio, its grouping across two games, and its anchor term; VERGE results refer to this recipe as a whole, including its reward, validity penalty, and update rule.

Theoretical comparison. SFT and VERGE optimize fundamentally different objectives. SFT imitates a specific equilibrium selector $T(A) = (\mathbf{p}^*, \mathbf{q}^*)$ determined by the LP solver. VERGE minimizes the exploitability residual $\text{Exploit}(A, \mathbf{p}, \mathbf{q})$. The following result, which adapts the argument of Lemma 5.2 of Duan et al. (2023a) to zero-sum NashConv, the sum of both regrets rather than the maximum deviation, characterizes their distinct stability properties and bounds the effect of display rounding on the score:

Theorem 1 (Residual stability vs. selector instability)**.** *For any fixed strategies* $(\mathbf{p}, \mathbf{q}) \in \Delta^n \times \Delta^n$*:*

$$|\text{Exploit}(A, \mathbf{p}, \mathbf{q}) - \text{Exploit}(B, \mathbf{p}, \mathbf{q})| \le 2\|A - B\|_\infty. \tag{3}$$

That is, $\text{Exploit}(\cdot, \mathbf{p}, \mathbf{q})$ *is Lipschitz in the payoff matrix under the max-entry norm* $\|A\|_\infty = \max_{i,j} |A_{ij}|$.

Conversely, for any $\delta > 0$*, there exist games* A, A' *with* $\|A - A'\|_\infty < \delta$ *such that* $\|T(A) - T(A')\|_1 = \Omega(1)$*, where* T *is any equilibrium selector that returns a vertex solution at degeneracy, as common deterministic LP tie-breaking rules do.*

The first part follows because the bilinear terms of Equation (1) cancel and each remaining extremum moves by at most $\|A - B\|_\infty$; the second uses a scaled matching-pennies family that reaches the all-zeros matrix. The full proof is in Section D.

Consequence. For a fixed strategy pair, VERGE's signal is Lipschitz in the payoffs, whereas a deterministic solver label can jump at support-boundary degeneracies; the transfer of trained models is measured in Section 4.2. Since all nonconstant matrices are rescaled to span 2, the training reward is a fixed affine transform of the residual, and the theorem does not by itself control a range-normalized reward as the payoff range approaches zero.

Why not zero-sum self-play? In competitive self-play each response would be scored as both row and column player with antisymmetric rewards $(r_i, -r_i)$. We prove in Section E that such *role-merged* GRPO, with both rewards in one normalization group, yields exactly zero on-policy, unclipped strategic gradient for every realization; clipping outside its interval and KL terms lie outside the statement. We therefore train the cooperative formulation, in which each response is scored once.

4 Experiments

Experiments use Qwen3.5-9B (Qwen Team, 2026) with LoRA of rank $r = 32$ and $\alpha = 64$. SFT and VERGE train for 2000 steps with learning rates 5×10^{-5} and 10^{-5}. Evaluation uses 50 games per size and four samples per game at temperature 0.7 and top-p 0.95, since best-of-four requires stochastic decoding, with a 128-token generation cap. We report $s@0.10$: the fraction of games where the *best-of-4* sample achieves $\bar{\text{Exploit}} < 0.10$. The selection uses only the certificate, computable from the matrix and the four candidates without the equilibrium, at a cost of four generations per

Table 1: Interface audit at a 1,024-token budget over three grids, with 900 paired cases per interface. Effects are correct-label minus anonymous pass@1 in percentage points, with 95% CIs from the grid-then-case bootstrap and from the bootstrap over the ten base-game identities, each with 20,000 replicates.

Contrast	Effect	Grid-then-case CI	Identity CI
Raw interface	24.6	$[19.1, 30.0]$	$[9.1, 41.2]$
Chat interface	1.6	$[-0.3, 3.4]$	$[-1.8, 4.9]$
Raw minus chat	23.0	$[17.0, 29.0]$	$[5.8, 42.1]$

game, so the main table also reports pass@1 on the first sample for every model row. This threshold is NashConv below 20% of the payoff range; on the 3×3 arms the base-versus-SFT separation holds at every threshold from 0.05 to 0.20, as Section 4.2 shows. All evaluation games use seed 99. The uniform row in Table 2 calibrates a non-adaptive mixed-strategy null: 4% success at both 5×5 and 7×7. The worst-case binomial standard error is 7.1 points at $N=50$; $\pm$ values are s.d. across training seeds; game-level intervals are in Table 3. Table 9 lists the protocol of every experiment, and Section B.2 states which numbers are re-scored from full responses. The audit re-scoring, the padding calibration, and the chat-template base run follow a registration of endpoints and decision rules, summarized in Table 11, written after the original training runs and audit grids and before these analyses; the game-by-game and threshold comparisons are post-hoc descriptive analyses.

4.1 Interface-Controlled Label Audit

We evaluate the public Qwen3.5-9B checkpoint on ten named base games under anonymous, correct-label, and wrong-label conditions, with raw and chat interfaces and greedy pass@1 decoding. The raw arm gives the user prompt to the model without a template; the chat arm renders the same user prompt with the model's chat template, a one-line system instruction, and thinking disabled, so the interface contrast comprises these three elements together; generation is greedy, and the parser scores the last valid JSON strategy object in the response, as detailed in Section B. Three affine and permutation grids fixed before evaluation hold 30 payoff-preserving variants per identity; each grid contains 4,200 observations. Re-scoring all 12,600 released prompts and responses from scratch reproduces every stored validity and success flag, with a maximum reward difference of 2.4×10^{-7}.

At 1,024 tokens, the correct-label effect is 24.6 points under raw prompting, 95% CI $[19.1, 30.0]$, and 1.6 points under matched chat, CI $[-0.3, 3.4]$, with chat validity of 99.7–100%; the raw-minus-chat interaction in Table 1 is 23.0 points with CI $[17.0, 29.0]$, and it is 18.3, 23.3, and 27.3 points in the three grids separately. These intervals resample grids and then paired cases. Because the grids share the ten identities, we also resample identities: the interaction interval becomes $[5.8, 42.1]$, so the interface dependence holds at the level of base games, and leaving out any single identity keeps the interaction between 17.5 and 27.4 points. Under raw prompting, labels mainly change whether the model answers, with valid-output rates of 17–24% anonymous and 42–58% with the correct label: a registered decomposition in Table 14 assigns 24.0 of the 24.6 points to validity, with identity CI $[10.3, 38.8]$, and 0.5, with CI $[-7.2, 6.5]$, to the success of valid answers, whereas on Qwen3-8B the interaction is carried by the success of valid answers, 4.6 of 4.8 points with CI $[0.8, 8.9]$. On the six canonical zero-sum games alone, the label effect is 30.6 points raw and 1.3 chat, an interaction of 29.3 with CI $[20.6, 37.6]$ and identity CI $[6.3, 53.5]$, as listed in Table 13; per-identity effects are in Section B. A fourth completed grid that reuses the seed-99 affine values with new permutations gives the same picture, with a four-grid interaction of 19.8 points, CI $[12.3, 26.8]$, positive in each grid. A registered replication on a second checkpoint, Qwen3-8B, with the same grids and protocol, gives the same direction at a smaller magnitude: 4.8 points with CI $[2.2, 7.4]$ and identity CI $[0.9, 9.3]$, as listed in Table 12.

4.2 Out-of-Distribution Generalization

Table 2 shows the core result. Both SFT and VERGE improve upon the base model at every size beyond training: 5×5 success rises from 2% for the raw base and 16% for the chat-template base to 61% for SFT and 37% for VERGE, whose seeds reach 58%, 40%, and 14%, and SFT reaches 32%

Table 2: OOD generalization on zero-sum games, training on sizes 2–3 and testing on 2–7. All rows use the seed-99 evaluation games, 50 per size; Table 5 covers VERGE seed 123. s@0.10 is best-of-4 success with exploitability selection; pass@1 uses only the first sampled output. Fine-tuned rows report mean±sample s.d. over 3 training seeds; SFT and format-only SFT valid-output rates are 1.00 in every cell, and the few-shot base has valid rates 0.22/0.26/0.14/0.07 and pass@1 0.02/0.00/0.00/0.00. Raw: plain prompt; chat: the model's chat template with thinking disabled, all else equal. Dashes: not evaluated. Analytic rows use the strict rule of Section B.1.

Method	**2×2**	**3×3**	**4×4**	**5×5**	**6×6**	**7×7**
Base, raw s@0.10	0.34	0.18	0.20	0.02	0.02	0.00
Base, raw pass@1	0.10	0.16	0.04	0.00	0.00	0.00
valid-output rate	*0.98*	*0.75*	*0.56*	*0.27*	*0.08*	*0.07*
Base, chat s@0.10	0.24	0.28	0.28	0.16	0.02	0.00
Base, chat pass@1	0.10	0.06	0.10	0.06	0.00	0.00
valid-output rate	*1.00*	*0.99*	*1.00*	*0.93*	*0.81*	*0.74*
Few-shot, 3 examples	0.10	0.02	0.00	0.00	–	–
Uniform	0.16	0.04	0.16	0.04	0.08	0.04
Pure maximin	0.72	0.74	0.38	0.28	0.28	0.10
SFT s@0.10, 3 seeds	**1.00**±.00	**0.97**±.04	**0.78**±.00	**0.61**±.09	**0.50**±.05	**0.32**±.09
SFT pass@1	**1.00**±.00	**0.84**±.09	0.49±.01	0.33±.07	0.23±.03	0.12±.03
Format-only SFT s@0.10	0.47±.05	0.27±.04	0.23±.02	0.15±.03	0.08±.07	0.05±.02
Format-only SFT pass@1	0.15±.02	0.07±.04	0.04±.02	0.05±.05	0.01±.01	0.00±.00
VERGE s@0.10, 3 seeds	0.85±.12	0.66±.22	0.50±.24	0.37±.22	0.29±.26	0.18±.09
VERGE pass@1	0.77±.21	0.50±.26	0.31±.20	0.21±.18	0.11±.10	0.07±.03
valid-output rate	*0.87*	*0.78*	*0.83*	*0.79*	*0.49*	*0.40*
LP Oracle	1.00	1.00	1.00	1.00	1.00	1.00

at 7×7, where every base arm and the uniform null are at or below 4%. On the first sample alone, SFT reaches 0.33±0.07 at 5×5 and 0.12±0.03 at 7×7, VERGE 0.21±0.18 and 0.07±0.03, and the base 0.00 raw and 0.06 and 0.00 with the chat template, so the transfer does not depend on verifier selection. The SFT and VERGE rows are per-seed summaries of the four-sample evaluation script; a seeded re-run on the three SFT adapters reproduces every SFT cell, and Section B.2 gives the record type behind each number.

Format-competent and payoff-reading comparators. The raw-prompted base model's valid-output rate in Table 2 falls from 0.98 at 2×2 to 0.27/0.08/0.07 at 5×5–7×7, so a raw base-to-SFT gain would combine adaptation to the JSON-only output contract with strategy learning. We therefore compare SFT with comparators that need no format adaptation, on the same protocol and the same games, in Table 3. Under its chat template the base model answers validly in 93% of 5×5 samples; averaged over the three training seeds, SFT exceeds it by 45 points with best-of-four, 0.61 versus 0.16, and by 27 points at pass@1, 0.33 versus 0.06. Game by game, a re-evaluation of SFT seed 42 with per-game records solves 35 of the 50 5×5 games with best-of-four and 20 at pass@1, against 8 and 3 for the chat-template base, with 28 versus 1 and 18 versus 1 discordant games. Of the 46 5×5 games without a pure-strategy saddle point, counted in Section A.5, the adapter solves 31. The payoff-reading maximin rule solves 14 of all 50 games and 10 of those 46; SFT solves 21 that maximin misses and maximin none that SFT misses, and at 3×3 SFT is ahead at pass@1 as well, 45 versus 37. At the stricter threshold 0.05, the seed-42 adapter still solves 15 5×5 games with best-of-four and 11 at pass@1, against 1 and 0 for the chat-template base and 6 for maximin, as listed in Table 7. A registered format-only control, the same SFT recipe trained on random valid strategy pairs with three seeds, answers validly in every sample and reaches 0.15 at 5×5 and 0.05 at pass@1; by the registered rule, SFT's gain beyond the better format-competent comparator is 45 points with best-of-four and 27 at pass@1, and each SFT seed in Table 5 exceeds the chat-template base and every format-only seed at every size in both readings. Because random targets lack the sparsity of LP labels, we also score a registered payoff-blind policy that answers each game with LP equilibria of other games from the same generator: at 5×5 it reaches 0.11 with best-of-four and 0.03 at pass@1, with 95% ranges over 1,000 draws of $[0.04, 0.20]$ and $[0.00, 0.08]$ listed in Table 13, so this prior does not explain SFT's 0.61 and 0.33.

Table 3: Game-by-game comparison on the same 50 seed-99 games, a post-hoc descriptive analysis: games solved by the re-evaluated SFT seed-42 adapter and by the comparator, discordant games, exact two-sided sign test, and 95% interval of the difference from a game bootstrap with 20,000 replicates. The seed-42 re-evaluation solves 35/20 5×5 games where its evaluation in Table 2 gave 31/17; SFT values are the stored per-sample rewards of that run, and the chat-template base is re-scored from its full responses.

Comparator	Size	Reading	SFT	comp.	SFT only / comp. only	p	difference
chat base	3×3	pass@1	45	3	43 / 1	5.1×10^{-12}	$+0.84$ $[0.72, 0.94]$
chat base	3×3	best-of-4	50	14	36 / 0	2.9×10^{-11}	$+0.72$ $[0.60, 0.84]$
chat base	5×5	pass@1	20	3	18 / 1	7.6×10^{-5}	$+0.34$ $[0.20, 0.48]$
chat base	5×5	best-of-4	35	8	28 / 1	1.1×10^{-7}	$+0.54$ $[0.38, 0.68]$
maximin	3×3	pass@1	45	37	10 / 2	0.039	$+0.16$ $[0.04, 0.28]$
maximin	3×3	best-of-4	50	37	13 / 0	2.4×10^{-4}	$+0.26$ $[0.14, 0.38]$
maximin	5×5	best-of-4	35	14	21 / 0	9.5×10^{-7}	$+0.42$ $[0.28, 0.56]$

Table 4: Reference policies on the padding game streams: games passed out of 50 at $\bar{\text{Exploit}} < 0.10$ under the range-normalized residual of the padded matrix / the core-span residual.

Condition	**Policy**	8×8	12×12	15×15	20×20
Dominated $3\to N$	P1 payoff-blind	43 / 3	45 / 6	42 / 5	42 / 1
	P2 core oracle	50 / 50	50 / 50	50 / 50	50 / 50
	P3 uniform over N	0 / 0	0 / 0	0 / 0	0 / 0
Random $3\to N$	P1 payoff-blind	0 / 0	0 / 0	0 / 0	0 / 0
	P2 core oracle	1 / 1	0 / 0	0 / 0	0 / 0
	P3 uniform over N	3 / 1	6 / 4	5 / 2	8 / 3

Sample and threshold sensitivity. On the 3×3 re-evaluation in Table 6, the SFT seeds solve 34–38 of 50 games on all four samples, while the base model solves 40 of 50 on none. The calibrated-control run saved the full responses of all three SFT adapters on 50 further unpadded 3×3 games; re-scored from the text, they solve 49, 49, and 49 games with best-of-four and 44, 45, and 41 at pass@1, and no flag in Table 13 changes when they are scored on the displayed matrix. At 5×5 the seed-42 adapter stays ahead of maximin and of the chat-template base at every threshold from $\tau = 0.05$ to 0.15 in both readings of Table 7; at $\tau = 0.20$ uniform play already passes 41 of 50 games, so looser thresholds do not discriminate between policies.

4.3 Calibrating the Dominated-Padding Control

A common way to separate the size of a game from its difficulty is to embed a small game into an $N \times N$ matrix by adding actions that are irrelevant under iterated dominance: new rows have payoffs below the original minimum and are strictly dominated for the row player, and on the original row support new columns have payoffs above the original maximum and are dominated for the column player, who minimizes the row player's payoff. The Nash equilibrium is unchanged and only the dimension grows. As a contrast, the same 3×3 block is placed in the top-left corner of an otherwise random $N \times N$ game, which in general changes the equilibrium. In both constructions the embedded block occupies the first three rows and columns.

We regenerate the padded game streams exactly, with seed 99, 50 cores per size, and the generation order of the released scripts, and score three policies that need no model in Table 4: *P1*, a payoff-blind policy, plays $(1/3, 1/3, 1/3, 0, \ldots, 0)$ for both players; *P2*, the core oracle, solves the embedded block and pads its LP equilibrium with zeros, reading the core payoffs and nothing else; *P3* plays uniformly over all N actions. Each is scored with the range-normalized residual of the padded matrix and with the core-span residual, which divides the same NashConv of the full padded game by twice the span of the embedded 3×3 block. On every dominated game, P2 has NashConv below 10^{-15}, confirming that the construction preserves the embedded equilibrium.

The padded entries lie outside the embedded game's range, with rows below its minimum by 1.1–1.5 and columns above its maximum by the same margin, so the padded matrix has a payoff span of about 5 against 2 for the embedded game, and the range-relative threshold $\mathrm{Exploit} < 0.10$ is about 2.5 times looser in absolute terms. Under this score, the payoff-blind policy passes 84–90% of dominated games and none of the random ones, reproducing the dominated-versus-random contrast without reading a single payoff. Scored at the embedded game's precision, it passes 1–6 of 50, while the core oracle passes all 50 at every size. A second game stream, generated by the released dominated-padding script at $N = 8, 10, 12, 15, 20$, gives the same pattern: P1 passes 42–47 of 50 under the range-normalized score and 2–5 under the core-span score. A padding control that isolates solving of the embedded game therefore needs the core-span score and the embedded block at a random position, so that a policy anchored to fixed actions cannot pass. The calibrated control permutes the rows and columns of the padded matrix, and the permutation matters for trained models as well: under core-span scoring with best-of-four, SFT seeds 42, 123, and 7 solve 45, 43, and 40 of 50 dominated 12×12 games with the embedded block in the first three positions and 14, 14, and 21 after permutation. On the permuted control, the three SFT adapters pass 49 of 150 12×12 games at the embedded game's precision, against 6 of 50 for uniform play on the core positions, and pooled within game they are ahead on 23 games and behind on 5, a registered game-level difference of 20.67 points with 95% interval $[6.67, 34.67]$.

5 Discussion

Three findings hold across the controls. First, the named-game label advantage of Qwen3.5-9B is a property of the raw interface: it shrinks from 24.6 to 1.6 points under a matched chat template, and the interaction remains positive when base-game identities are resampled and in a registered replication on a second checkpoint, Qwen3-8B, where it is 4.8 points with 95% CI $[2.2, 7.4]$. Second, procedural training on 2×2–3×3 games transfers to 5×5–7×7 under a fixed direct-answer protocol, for LP-label SFT and for a residual-reward recipe that uses no equilibrium labels. SFT exceeds a chat-template base that answers validly, a format-only SFT control, and a payoff-blind LP-label prior, and with best-of-four it solves every paired 5×5 game that maximin solves plus 21 more. Third, dominated-action padding is passed by a payoff-blind policy under range-normalized scoring, not under core-span scoring.

Implications for evaluation. Cue manipulations of named games should hold the prompting interface fixed and report uncertainty over base-game identities, as in Section 4.1. Direct-answer scores should report pass@1 next to best-of-k, include a base model that already satisfies the output contract, and compare game by game with payoff-reading heuristics such as maximin. Padding controls should score at the embedded game's precision and randomize the embedded block's position, as in Section 4.3.

Reading named-game results. Compare labels kept and removed under the benchmark and matched-chat interfaces, resampling base-game identities. Persistence across interfaces and identities supports a game-knowledge interpretation; a gap confined to one interface measures a cue–prompt interaction.

Beyond zero-sum games. The certificate extends beyond the zero-sum case: for a general-sum bimatrix game, the sum of both players' best-response gains remains a label-free score, but LP labels and the game value no longer exist, so training targets and payoff-reading references would need redefinition.

Scope. Training uses 2×2–3×3 zero-sum games with payoffs rendered on a span-2 scale, LoRA adapters of Qwen3.5-9B without full-parameter fine-tuning, three seeds per method, and VERGE as in Equation (2), whose results belong to the recipe as a whole. The audit covers two Qwen checkpoints and ten named base games and concerns interface dependence, not training-data exposure. Training gains are for the prompt as posed in Section 3.1, which states the output contract but not the solution concept, and the size-transfer claim concerns freshly generated games, not invariance of the answer to relabeling the actions. The conclusions concern the tool-free direct-answer protocol with open checkpoints.

AI USE STATEMENT

In this work, we used generative AI tools to design and improve experiments and to implement methods. We have not used generative AI tools to formulate questions for surveys or interviews or to transcribe recordings, and qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools to edit the paper to improve readability and to create or edit software code. We have reviewed all AI-assisted work: AI-assisted code was tested before use, and every reported number is computed from recorded experimental outputs by analysis scripts. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This study involves no human subjects, crowdsourcing, or personal data: all games are procedurally generated or are textbook matrix games, and all models are public checkpoints. The released data consist of synthetic payoff matrices, prompts, model outputs, and scores, and they contain no information about individuals; fairness considerations tied to demographic groups do not arise. The work concerns evaluation and training of equilibrium solvers on synthetic games, not a deployed strategic agent. Its main benefit is sharper tests that separate procedural performance from cue and interface effects. A possible misuse is the deployment of stronger strategic-reasoning systems in negotiation, persuasion, or planning without human oversight; to limit this risk, the supplementary material provides code, synthetic generators, and records rather than trained checkpoints or a user-facing decision system.

REPRODUCIBILITY STATEMENT

The testbed, the certificate, and the success rule are defined in Section 3.1 and Section B.1; the training recipes are in Section 3.2 and Section C, with the training and evaluation budget in Table 10; the protocol of every experiment is in Table 9, with verbatim prompts in Section B; the record type behind each reported number is in Section B.2; complete proofs are in Sections D and E; and the registered analyses are in Section F. The anonymous supplementary material contains the game generator, training, evaluation, and analysis code, the released records, and scripts that recompute the reported numbers from them.

REFERENCES

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the strategic reasoning capabilities of LLMs via game-theoretic evaluations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 28219–28253, 2024. doi: 10.52202/079017-0885.

Zhijian Duan, Wenhan Huang, Dinghuai Zhang, Yali Du, Jun Wang, Yaodong Yang, and Xiaotie Deng. Is Nash equilibrium approximator learnable? In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pp. 233–241. IFAAMAS, 2023a. URL https://www.ifaamas.org/Proceedings/aamas2023/pdfs/p233.pdf.

Zhijian Duan, Yunxuan Ma, and Xiaotie Deng. Are equivariant equilibrium approximators beneficial? In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8747–8778, 2023b. URL https://proceedings.mlr.press/v202/duan23d.html.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. Can large language models serve as rational players in game theory? A systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17960–17967, 2024. doi: 10.1609/aaai.v38i16.29751.

Kanishk Gandhi, Dorsa Sadigh, and Noah D. Goodman. Strategic reasoning with language models. *arXiv preprint arXiv:2305.19165*, 2023.

Dimitrios Georgousis, Maria Lymperaiou, Angeliki Dimitriou, Giorgos Filandrianos, and Giorgos Stamou. Evaluating counterfactual strategic reasoning in large language models. In *Proceedings of the Fifth Workshop on Generation, Evaluation and Metrics (GEM)*, pp. 309–354, San Diego, California, USA, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.gem-main.31. URL https://aclanthology.org/2026.gem-main.31/.

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*, volume 30, 2017.

Paraskevas V. Lekeas and Giorgos Stamatopoulos. How a cooperative-override circuit suppresses Nash play in large language models. *arXiv preprint arXiv:2604.27167*, 2026.

Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, 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. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=7Yayy5fNLg.

Siqi Liu, Luke Marris, Georgios Piliouras, Ian Gemp, and Nicolas Heess. NfgTransformer: Equivariant representation learning for normal-form games. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=4YESQqIys7.

Nunzio Lorè and Babak Heydari. Strategic behavior of large language models and the role of game structure versus contextual framing. *Scientific Reports*, 14:18490, 2024. doi: 10.1038/s41598-024-69032-z.

Nuoyan Lyu, Bingbing Xu, Xueyun Tian, Weihao Meng, Yige Yuan, Yang Zhang, Zhiyong Huang, Tat-Seng Chua, and Huawei Shen. GIFT: Games as informal training for generalizable LLMs. *arXiv preprint arXiv:2601.05633*, 2026.

Luke Marris, Ian Gemp, Thomas Anthony, Andrea Tacchetti, Siqi Liu, and Karl Tuyls. Turbocharging solution concepts: Solving NEs, CEs and CCEs with neural equilibrium solvers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5586–5600, 2022. doi: 10.52202/068431-0404.

Mateo Pechon-Elkins and Jon Chun. Auditing game-theoretic measures of strategic reasoning in LLMs. *arXiv preprint arXiv:2603.10029*, 2026.

Qwen Team. Qwen3.5-9B model card. https://huggingface.co/Qwen/Qwen3.5-9B, 2026.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *International Conference on Learning Representations (ICLR)*, 2024.

Vartan Shadarevian, Kia Ghods, Alex Kenich, and Anany Kotawala. GENSTRAT: Toward a science of strategic reasoning in large language models. *arXiv preprint arXiv:2605.23238*, 2026.

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

Alonso Silva. Large language models playing mixed strategy Nash equilibrium games. In *Network Games, Artificial Intelligence, Control and Optimization (NETGCOOP 2024)*, Lecture Notes in Computer Science, pp. 142–152. Springer, 2025. doi: 10.1007/978-3-031-78600-6_13.

Han Wang, Philippe Beardsell, Boning Li, Aaron Sasmita, Shuai Li, Hongyuan Zha, and Baoxiang Wang. Solver-guided reasoning for mixed-equilibrium strategies. *arXiv preprint arXiv:2608.06741*, 2026.

Haochuan Wang, Xiachong Feng, Lei Li, Yu Guo, Zhanyue Qin, Dianbo Sui, and Lingpeng Kong. TMGBench: A systematic game benchmark for evaluating strategic reasoning abilities of LLMs. *arXiv preprint arXiv:2410.10479*, 2024.

Dawen Wu and Abdel Lisser. Using CNN for solving two-player zero-sum games. *Expert Systems with Applications*, 204:117545, 2022. doi: 10.1016/j.eswa.2022.117545.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244, 2025. doi: 10.52202/085713-3775.

Huining Yuan, Zelai Xu, Zheyue Tan, Xiangmin Yi, Mo Guang, Kaiwen Long, Haojia Hui, Boxun Li, Xinlei Chen, Bo Zhao, Xiao-Ping Zhang, Chao Yu, and Yu Wang. MARSHAL: Incentivizing multi-agent reasoning via self-play with strategic LLMs. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=GCd5v3ehmr.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.

Kehan Zheng, Jinfeng Zhou, and Hongning Wang. Beyond Nash equilibrium: Bounded rationality of LLMs and humans in strategic decision-making. *arXiv preprint arXiv:2506.09390*, 2025b.

Yufa Zhou, Shaobo Wang, Xingyu Dong, Xiangqi Jin, Yifang Chen, Yue Min, Kexin Yang, Xingzhang Ren, Dayiheng Liu, and Linfeng Zhang. Reasoning like an economist: Post-training on economic problems induces strategic generalization in LLMs. *arXiv preprint arXiv:2506.00577*, 2025.

A Additional Results

The re-evaluations of Tables 3, 6 and 7 and the base re-run of Section A.3 use the trained adapters, SFT seeds 42, 123, and 7 and VERGE seed 7, on the same seed-99 evaluation games. Paired tests are exact two-sided sign tests over games. Intervals for pooled SFT differences come from a crossed bootstrap that resamples the 50 games and the three seeds independently with 20,000 replicates.

A.1 Per-Seed Values

Table 5: Per-seed values behind the fine-tuned rows of Table 2 on the seed-99 set of 50 games; each cell gives $s@0.10$ / pass@1 / sample-level valid rate. VERGE seed 123 was evaluated one size at a time with the generator re-seeded to seed 99 for each size, so its games coincide with the shared set at 2×2 and are fresh draws from the same generator at 3×3–7×7.

Seed	2×2	3×3	4×4	5×5	6×6	7×7
SFT 42	1.00 / 1.00 / 1.00	0.98 / 0.80 / 1.00	0.78 / 0.50 / 1.00	0.62 / 0.34 / 1.00	0.56 / 0.26 / 1.00	0.36 / 0.14 / 1.00
SFT 123	1.00 / 1.00 / 1.00	1.00 / 0.94 / 1.00	0.78 / 0.48 / 1.00	0.70 / 0.40 / 1.00	0.48 / 0.20 / 1.00	0.38 / 0.14 / 1.00
SFT 7	1.00 / 1.00 / 1.00	0.92 / 0.78 / 1.00	0.78 / 0.50 / 1.00	0.52 / 0.26 / 1.00	0.46 / 0.22 / 1.00	0.22 / 0.08 / 1.00
VERGE 42	0.98 / 0.98 / 1.00	0.90 / 0.78 / 1.00	0.76 / 0.54 / 1.00	0.58 / 0.38 / 0.97	0.58 / 0.22 / 0.83	0.28 / 0.10 / 0.39
VERGE 7	0.80 / 0.56 / 0.62	0.48 / 0.26 / 0.36	0.44 / 0.22 / 0.52	0.40 / 0.24 / 0.59	0.20 / 0.08 / 0.22	0.16 / 0.04 / 0.43
VERGE 123	0.76 / 0.76 / 1.00	0.60 / 0.46 / 1.00	0.30 / 0.16 / 0.96	0.14 / 0.02 / 0.80	0.08 / 0.02 / 0.43	0.10 / 0.06 / 0.38
Format-only 42	0.50 / 0.16 / 1.00	0.22 / 0.04 / 1.00	0.24 / 0.06 / 1.00	0.12 / 0.04 / 1.00	0.16 / 0.02 / 1.00	0.06 / 0.00 / 1.00
Format-only 123	0.42 / 0.16 / 1.00	0.28 / 0.12 / 1.00	0.24 / 0.02 / 1.00	0.16 / 0.10 / 1.00	0.06 / 0.02 / 1.00	0.02 / 0.00 / 1.00
Format-only 7	0.50 / 0.12 / 1.00	0.30 / 0.06 / 1.00	0.20 / 0.04 / 1.00	0.18 / 0.00 / 1.00	0.02 / 0.00 / 1.00	0.06 / 0.00 / 1.00

A.2 Sample Distribution at 3×3

Table 6 gives the four-sample distribution and the threshold sweep of the small-size re-evaluation. The distribution is bimodal: the SFT seeds solve 34–38 of 50 games on all four samples, while the base model solves 40 of 50 on none, so between 68% and 81% of what best-of-4 credits to SFT is a game solved on every draw. Pooled over the three SFT seeds, best-of-four exceeds maximin at 3×3 by 23 points, with a crossed game-by-seed 95% interval of $[11, 36]$.

Table 6: Small-size re-evaluation on 50 seed-99 3×3 games. Left: number of games on which exactly k of the four samples succeed at $\mathrm{Exploit} < 0.10$. Right: games succeeding at alternative thresholds τ on the same arms, given as best-of-4 / pass@1.

	games with k successful samples					successes at threshold τ				
Arm	k=0	1	2	3	4	0.05	0.10	0.15	0.20	0.30
Base	40	6	3	1	0	4/1	10/3	27/10	39/21	49/34
SFT seed 42	1	3	4	5	37	45/41	49/45	50/48	50/50	50/50
SFT seed 123	0	0	5	11	34	46/38	50/47	50/50	50/50	50/50
SFT seed 7	3	4	3	2	38	42/33	47/40	50/48	50/49	50/50
VERGE seed 7	23	12	8	5	2	21/9	27/11	30/13	35/15	36/15

A.3 Base-Model Evaluation Variability

The raw base row of Table 2 is a single evaluation run at temperature 0.7. The small-size re-evaluation of Table 6 includes an independent re-run of the base model on the same protocol at 2×2 and 3×3, which gives 0.26 and 0.20 against 0.34 and 0.18 in Table 2, with valid rates of 0.97 and 0.78 versus 0.98 and 0.75. These differences of four games and one game out of 50 lie within the binomial standard error of 7 points. An earlier evaluation of the base model on the same games gives $0.36/0.24/0.14/0.04/0.00/0.00$ at $n = 2, \ldots, 7$.

A.4 Iterative Payoff-Reading Solvers

As a cost reference for the direct-answer protocol, a registered descriptive analysis runs two standard iterative solvers on the seed-99 evaluation games: fictitious play, with simultaneous updates from

Table 7: Games solved out of 50 at thresholds τ on the seed-99 3×3 and 5×5 games, given as best-of-4 / pass@1; a post-hoc descriptive analysis. SFT seed 42 values are stored per-sample rewards of its re-evaluation; the chat-template base is re-scored from full responses; maximin and uniform emit one strategy, so both readings coincide.

Size	Arm	$\tau = 0.05$	0.10	0.15	0.20
3×3	SFT seed 42	49 / 38	50 / 45	50 / 50	50 / 50
	Base, chat template	3 / 1	14 / 3	30 / 8	40 / 18
	Maximin	29	37	43	44
	Uniform	0	2	12	33
5×5	SFT seed 42	15 / 11	35 / 20	49 / 31	50 / 47
	Base, chat template	1 / 0	8 / 3	18 / 6	36 / 17
	Maximin	6	14	27	34
	Uniform	0	2	19	41

uniform play and strategies given by the empirical averages of best responses, and simultaneous regret matching with average strategies. One iteration is one matrix–vector product per player, and each solver returns one strategy pair per game, scored in exact rational arithmetic with the strict success rule of Section 3.1. Table 8 gives the success rates. Regret matching reaches the 5×5 and 7×7 best-of-four rates of SFT after four iterations and fictitious play after 16; both pass every game at every size within 32 iterations.

Table 8: Success rate of iterative payoff-reading solvers after T iterations on the seed-99 evaluation games, with 50 games per size and one strategy pair per game, under the exact strict rule. With the evaluation harness's span constant, which admits residuals exactly at 0.10, three cells shown here are one game higher: fictitious play at 3×3 for $T{=}1$ and 4 and at 5×5 for $T{=}1$.

Solver	Size	$T{=}1$	2	4	8	16	32	64
Fictitious play	3×3	0.52	0.68	0.84	0.94	1.00	1.00	1.00
	5×5	0.20	0.26	0.32	0.56	0.98	1.00	1.00
	7×7	0.02	0.04	0.08	0.28	0.72	1.00	1.00
Regret matching	3×3	0.04	0.22	0.88	0.94	1.00	1.00	1.00
	5×5	0.04	0.22	0.66	0.84	0.98	1.00	1.00
	7×7	0.04	0.16	0.42	0.82	1.00	1.00	1.00

A.5 Degeneracy and Isomorphism to Named Games

Pure-strategy saddle points. Of the 50 seed-99 evaluation games per size, 31/25/5/4/2/0 admit a pure-strategy saddle point at $n = 2, \ldots, 7$, that is, 62%, 50%, 10%, 8%, 4%, and 0%, and only 2×2 games contain a constant row or column, 6 of 50. Beyond the training sizes most evaluation games therefore have no pure-strategy equilibrium.

Isomorphism to catalogued named games. We matched every evaluation game against a catalogue of seven named zero-sum games: the two exemplars used in the few-shot prompt, rock–paper–scissors, rock–paper–scissors with a well, O'Neill's 4×4 game, two-finger Morra, and rock–paper–scissors–lizard–Spock. Closed under player swap, the catalogue has nine classes, which we matched up to row and column permutation and positive affine rescaling, using exact rational arithmetic. None of the 200 evaluation games at 2×2–5×5 matches any class, and none is skew-symmetric. Over the exhaustive 2×2 generator support of $19^4 = 130{,}321$ matrices, 990 or 0.76% are affine copies of matching pennies or of the few-shot exemplars, and fresh 2,000-game samples match 13 times at 2×2 and never at 3×3–5×5.

B Experimental Setup and Evaluation Protocols

Game generation details. Integer payoffs are drawn uniformly from $\{-9, \dots, 9\}$ and rescaled to span 2: $A \leftarrow 2A/(\max A - \min A)$. Entries are not re-centered, so a matrix with entries in $\{5, \dots, 9\}$ maps into $[2.5, 4.5]$. If all entries are equal, one is perturbed by +1 to ensure non-degeneracy.

Procedural OOD evaluation protocol. For each game size n and the evaluation set with seed 99:

- generate 50 random games, or 30 for intermediate checkpoints;
- for each game, generate 4 samples at temperature 0.7 and top-p 0.95 with at most 128 new tokens;
- parse the JSON output and normalize each strategy to the probability simplex;
- report the best-of-4 exploitability reward and first-sample pass@1;
- $s@0.10$ is the fraction of games whose best reward exceeds 0.90.

Table 9: Protocols of all experiments. "2-sided" is the residual of Definition 1. Unless stated, the model is Qwen3.5-9B and games come from the integer generator with seed 99.

Experiment	Models	Games	Decoding	Tokens	Score
Interface audit, Section 4.1	base	10 identities × 30 variants × 3 conditions, 3 grids	greedy, raw / chat	128, 1,024	2-sided, pass@1
Main OOD, Table 2	base with raw prompt and with chat template, SFT ×3, format-only SFT ×3, VERGE ×3	50 per size, $n = 2$–7	4 samples, T=0.7, top-p 0.95	128	2-sided, best-of-4, pass@1
Paired re-evaluation, Table 3	SFT seed 42	50 at 3×3 and 5×5	as main	128	2-sided, both
Small-size re-evaluation, Table 6	base, SFT ×3, VERGE seed 7	50 at 2×2 and 3×3	as main	128	2-sided, both
Padding calibration, Section 4.3	none; reference policies	50 cores per N, $N = 8$–20	–	–	range-normalized, core-span
Calibrated padding control, Table 11	SFT ×3	50 cores per N, $N = 8$–20, rows and columns permuted	as main	$64+12N$, ≤ 304	range-normalized, core-span

Verbatim prompts. No system message is used in the main evaluation. The plain-JSON user prompt is shown below with the matrix printed row by row to one decimal; the first matrix row is an example, and the ellipses stand for the remaining rows and, in the last line, for the strategy entries the model fills in.

```
You are given a zero-sum matrix game.
Row player receives the listed payoff.  Column player
receives the negative.
Return mixed strategies for both players as JSON only.
Matrix A:
[
[0.4, -1.2, 0.8],
...
]
Return:  {"row":[...],"col":[...]}
```

The few-shot variant of Table 2 prepends three solved games in the same format. The chat-template base wraps the identical user prompt in the model's chat template with thinking disabled and no system message. All prompts were fixed before evaluation; none was selected by performance.

Interface-controlled audit protocol. The audit uses the public Qwen3.5-9B checkpoint. Ten named base identities are split into six canonical zero-sum games and four conventional general-sum games evaluated under an explicit zero-sum recast. Each identity appears in 30 payoff-preserving variants per grid, each a positive affine map of the payoffs combined with independent row and column permutations. The three grids of the main analysis, with seeds 99, 456, and 789, use mutually disjoint affine values and were fixed before evaluation; they reuse the ten identities, so they are holdout grids rather than independent model replications. Each grid evaluates anonymous,

correct-label, and wrong-label conditions with 300 named cases per condition, plus 150 anonymous procedural matrices, under raw and chat interfaces at 128- and 1,024-token budgets, with greedy pass@1 decoding: $4 \times (900 + 150) = 4{,}200$ observations per grid. The raw arm tokenizes the user prompt with the tokenizer's special tokens and no template; the chat arm wraps the same user prompt in the model's chat template with the system message "Solve the displayed zero-sum matrix game. Return JSON only." and thinking disabled. The interface contrast therefore comprises the template, the system instruction, and the thinking setting together. The generation driver is released with the rows. The released script re-parses the displayed matrix and every response, re-scores all observations, and recomputes each contrast with two bootstraps of 20,000 replicates each: grids and then paired cases within grid, and the ten identities with all their variants pooled across grids. Valid-output rates under raw prompting are 17.0–24.3% for anonymous prompts, 41.7–57.7% with the correct label, and 44.3–62.0% with the wrong label across grids, and 99.7–100% for every chat condition. The raw label effect is heterogeneous across identities, from -11 to $+72$ points, which is why the identity intervals are wide; the chat effect lies between -9 and $+11$ points for every identity; it is 1.3 points for the six canonical zero-sum games and 1.9 points for the four general-sum games recast as zero-sum. A fourth completed grid with seed 123 reuses the affine values of the seed-99 grid with new permutations and is therefore reported as a sensitivity analysis; its interaction is 10.0 points, against 18.3, 23.3, and 27.3 for the seed-99, 456, and 789 grids. Leaving out any single identity from the three main grids keeps the interaction between 17.5 and 27.4 points, a descriptive analysis added after registration.

Second checkpoint. The registered replication runs the public Qwen3-8B checkpoint, a dense model of the previous Qwen generation, on the released case files of the three main grids, unchanged, with the same four arms, prompts, system message, thinking setting, and greedy decoding: 12,600 further observations, re-scored with the released script. The registration was written before any Qwen3-8B generation; its primary endpoint is the $1{,}024$-token raw-minus-chat interaction with the grid-then-case interval, and its rule that the point estimate is above 0 and the interval excludes 0 is met, as listed in Table 11. The secondary quantities are in Table 12. The replication confirms the direction of the interaction at a smaller magnitude than for Qwen3.5-9B.

B.1 Numerical Conventions

Threshold rule. Success is the strict inequality $r > 0.90$. The evaluation harness adds 10^{-8} to the span, which matters only for residuals exactly at the threshold. Re-scoring the analytic baselines on the 300 seed-99 games in exact rational arithmetic changes one cell of Table 2: one uniform-play $2{\times}2$ game has normalized residual exactly 0.10, so uniform $2{\times}2$ success is $8/50$, where the harness constant gives $9/50$; the maximin and LP rows are unchanged. Model rows are the values of the evaluation harness. The re-evaluation records of Tables 3, 6 and 7 keep each sample's reward and the final 120 characters of its response, so the SFT and VERGE values in these tables aggregate the stored per-sample rewards of the harness, and the retained tails are not used to validate them; the supplement accounts for every retained tail against its stored reward. The chat-template base and the audit keep full responses and are re-scored from them. In the Qwen3.5-9B interface audit, exact re-scoring of every response changes no success flag. In the Qwen3-8B audit, exact rational scoring changes four stored success flags: two wrong-label cases, one in each of two grids, each at both chat budgets, have normalized residual exactly 0.10 and therefore fail the strict rule, whereas their stored floating-point reward exceeds 0.90; the supplement lists these cases. The supplement keeps the original records unchanged and adds the exact strict scores and the summaries regenerated from them; no registered estimate or interval changes.

Displayed versus scored matrix. Displayed entries differ from the scored ones by at most 0.05, so by Theorem 1 the normalized residual of any answer moves by at most

$$\frac{2 \cdot 0.05}{2 \cdot 2} = 0.025$$

on the span-2 matrices. Scoring the analytic baselines on the displayed instead of the unrounded matrix changes 0–3 of 50 games per cell; maximin at $5{\times}5$ goes from 14 to 11 games. The registered analysis of Table 13 applies the same comparison to the saved full responses of the chat-template base, the format-only control, and the SFT adapters on the unpadded $3{\times}3$ and permuted $12{\times}12$ conditions: 0–4 of 50 best-of-4 flags and 0–1 pass@1 flags change per arm and size for the chat-template base

and the format-only control, none for the three SFT adapters on the unpadded 3×3 games, and 1–3 best-of-4 flags for the SFT adapters on the permuted 12×12 padding control under the core-span score. The registered 5×5 quantity of Table 13 subtracts the displayed-matrix rate of the better comparator from the main-table SFT means, which have no saved responses and are not re-scored; it is therefore not a paired displayed-matrix comparison, and the paired full-response checks for SFT are the unpadded 3×3 and permuted 12×12 ones above. The task is defined by the unrounded matrix, and a strategy pair that is exact for the displayed matrix has residual at most 0.025 on it.

B.2 Record Provenance

Numbers re-scored from saved full responses are the interface audit and its second-checkpoint replication, with every prompt and response; the chat-template base in Tables 2, 3 and 7; the format-only control in Tables 2 and 5; the calibrated padding control of the SFT adapters and of VERGE seed 7 in Tables 11 and 12; and the analyses of Table 13. The SFT and VERGE rows of Tables 2 and 5 are per-seed summaries written by the four-sample evaluation script, not by the trainers' internal progress checks, which use other settings; the script stores summary rates, so these rows are reported as summaries. Re-running the script with its fixed seeds on the three SFT adapters reproduces every per-seed SFT value of Table 5, and hence every SFT cell of Table 2; the supplement contains the script, its outputs, and a comparison script. This identifies the SFT rows as outputs of the stated protocol, and answer-level analyses of the SFT adapters use the re-evaluations and full-response records listed in this subsection. Numbers that aggregate stored per-sample rewards of the evaluation code are the SFT and VERGE re-evaluations of Tables 3, 6 and 7, whose records keep each sample's reward and only the last 120 characters of its response. The analytic rows for uniform, maximin, and LP are recomputed from the regenerated games. Table 2 and the re-evaluations are separate evaluation runs with fresh sampling, so the same seed can differ between them: SFT seed 42 at 5×5 solves 31/17 games in Table 2 and 35/20 in Table 3. The VERGE mean in Table 2 includes seed 123, whose 3×3–7×7 games are fresh draws from the same generator, as noted in Table 5; the $\pm$ values are therefore across training seeds and, for VERGE, also across these two game sets.

C Method and Implementation Details

LoRA configuration. All experiments use LoRA with $r = 32$, $\alpha = 64$, and dropout 0.05, applied to the query, key, value, and output projections and to the gate, up, and down projections, seven target projection types in total. In the hybrid Qwen3.5 architecture these match the projections of the full-attention layers and the MLPs; the input and output projections of the linear-attention layers are not adapted. The base checkpoint is the public Qwen3.5-9B. The adapters have 58.2M trainable parameters against 9.0B in the base model, or 0.65%.

Optimizer and schedule. We use AdamW with weight decay 0.01, linear warmup followed by cosine decay, and learning rates 5×10^{-5} for SFT and 1×10^{-5} for VERGE. VERGE optimizes Equation (2): a clip of $[1-0.20, 1+0.28]$ on the ratio of length-normalized sequence likelihoods and a 0.02 penalty on the change of the mean token log-probability relative to the anchor. No separate frozen reference policy is used.

C.1 VERGE Implementation

Each outer step draws two games, each of a size sampled uniformly from $\{2, 3\}$, so both sizes occur in about half of the steps, permutes each game's rows and columns, and samples $G = 8$ responses per game at temperature 0.9, top-p 0.95, and at most 128 new tokens from the raw prompt. The 16 rewards are normalized jointly by the mean and standard deviation over the step. The anchor $\bar{\ell}_{\text{old}}(y_k)$ is then computed in a separate forward pass in evaluation mode, and the update of Equation (2) runs in training mode with LoRA dropout active on micro-batches of two responses with four-step accumulation and gradient-norm clipping at 1.0, giving two optimizer updates per outer step. Rollouts are sampled in the module mode left by the preceding update, that is, in training mode with LoRA dropout active, except at the first step and at the step after each periodic evaluation, which sample in evaluation mode; the anchor pass always runs in evaluation mode and the update in training mode. An invalid output receives -0.25 and any valid strategy pair at least 0.5, so the reward ranks every valid response above every invalid one. Both passes score the decoded response appended to the prompt

and a newline, that is, the retokenized text, so the anchor is the likelihood of that text recomputed after sampling. The ratio ρ_k compares length-normalized likelihoods of the same text under the current weights in training mode and the pre-update weights in evaluation mode, so it also reflects LoRA dropout of 0.05 from the first micro-batch on. The format-only control of Table 2 uses the SFT recipe unchanged except that each training target is a random valid strategy pair, each vector drawn from a uniform Dirichlet distribution and printed to three decimals like the LP targets, so it learns the output format without equilibrium information. SFT trains for 2,000 steps with the same LoRA configuration. The per-seed transfer in Table 5 is the result of this implementation.

Training and evaluation budget. Table 10 counts the training signal and updates of each recipe and the evaluation cost per game. The two recipes have the same number of outer steps; VERGE takes twice as many optimizer updates and samples 32,000 responses, so the SFT–VERGE comparison compares the two recipes at equal outer steps.

Table 10: Training and evaluation budget per run of one training seed. Training games are freshly generated 2×2 or 3×3 matrices; responses are at most 128 new tokens.

Recipe	Outer steps	Training games	Optimizer updates	Sampled responses
LP-label SFT	2,000	8,000, batch 4	2,000	0
Format-only SFT	2,000	8,000, batch 4	2,000	0
VERGE	2,000	4,000, 2 per step	4,000	32,000, 16 per step

Evaluation, every model row: 4 generations of at most 128 new tokens and 4 certificate evaluations per game; pass@1 uses the first generation only; 50 games per size.

D Proof of Residual Stability, Theorem 1

Proof. Fix $(\mathbf{p}, \mathbf{q}) \in \Delta^n \times \Delta^n$.

Step 1: row-regret term.

$$
\begin{aligned}
|\max_i (A\mathbf{q})_i - \max_i (B\mathbf{q})_i| &\le \max_i |(A\mathbf{q})_i - (B\mathbf{q})_i| && (4)\\
&= \max_i |((A-B)\mathbf{q})_i| && (5)\\
&\le \max_i \sum_j |A_{ij} - B_{ij}| \cdot q_j && (6)\\
&\le \|A-B\|_\infty \cdot \|\mathbf{q}\|_1 \\
&= \|A-B\|_\infty, && (7)
\end{aligned}
$$

where the last step uses $\|\mathbf{q}\|_1 = 1$.

Step 2: cancellation of the bilinear term. Expanding Equation (1), the two bilinear terms $\mathbf{p}^\top A\mathbf{q}$ cancel:

$$\mathrm{Exploit}(A, \mathbf{p}, \mathbf{q}) = \max_i (A\mathbf{q})_i - \min_j (\mathbf{p}^\top A)_j.$$

Step 3: Lipschitz bound. By Step 2 and the triangle inequality, and then by Step 1 applied to each term:

$$
\begin{aligned}
&|\mathrm{Exploit}(A, \mathbf{p}, \mathbf{q}) - \mathrm{Exploit}(B, \mathbf{p}, \mathbf{q})| \\
&\quad\le |\max_i (A\mathbf{q})_i - \max_i (B\mathbf{q})_i| + |\min_j (\mathbf{p}^\top A)_j - \min_j (\mathbf{p}^\top B)_j| && (8)\\
&\quad\le \|A-B\|_\infty + \|A-B\|_\infty \\
&\quad= 2\|A-B\|_\infty. && (9)
\end{aligned}
$$

The second term follows by the same argument applied to the rows of $(A-B)^\top$ with $\|\mathbf{p}\|_1 = 1$. No dimension factor n appears because the simplex constraints absorb it.

Step 4: selector instability. Let M be the matching-pennies matrix

$$M = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$$

and consider the scaled family $A_\epsilon = \epsilon \cdot M$. For any $\epsilon > 0$, the unique Nash equilibrium is $\mathbf{p}^* = \mathbf{q}^* = (1/2, 1/2)$, so

$$T(A_\epsilon) = ((1/2, 1/2), (1/2, 1/2)).$$

At $\epsilon = 0$, the payoff matrix is all zeros, exploitability is identically zero, and *every* mixed strategy is a Nash equilibrium. Any vertex-returning selector, including common deterministic LP tie-breaking rules, returns a pure strategy, for example $T(A_0) = ((1, 0), (1, 0))$. Thus

$$\|T(A_\epsilon) - T(A_0)\|_1 \geq 1,$$
$$\|A_\epsilon - A_0\|_\infty = \epsilon \to 0,$$

confirming an $\Omega(1)$ discontinuity for vertex-returning selectors. □

E GRPO Gradient Cancellation in Zero-Sum Self-Play

This section proves that *role-merged* GRPO, where the same generated output is scored in both row and column roles within a single normalization group, yields exactly zero on-policy, unclipped score-function gradient. The statement excludes KL regularization and PPO-style clipping outside its inactive region. A clipped surrogate can produce a nonzero correction once the current-to-behavior likelihood ratio crosses a clipping boundary because positive and negative advantages select different branches; this does not contradict the on-policy cancellation proved below. The statement concerns the implementation in which one output serves both roles simultaneously; it does not apply to self-play architectures with separate policy copies or asymmetric normalization groups. The result motivates the cooperative exploitability reward of Section 3.1.

Theorem 2 (On-policy score-gradient cancellation in symmetric zero-sum self-play)**.** *Consider a zero-sum game where a shared policy π_θ plays both roles. GRPO generates G responses, each evaluated as row player with reward r_i and as column player with reward $-r_i$. Under role-merged group normalization, the unclipped score-function gradient evaluated on-policy, at likelihood ratio* 1*, is exactly* **0** *for every realization. Equivalently, cancellation holds wherever both members of each pair use the same differentiable likelihood-ratio branch. The theorem does not assert cancellation of an off-policy clipped surrogate outside the clipping interval.*

Proof. Step 1: reward group. Under symmetric zero-sum self-play, each game instance k produces a row reward r_k and a column reward $-r_k$. The group of $2G$ rewards is

$$\mathcal{R} = \{r_1, \ldots, r_G, -r_1, \ldots, -r_G\}.$$

Step 2: group mean. The group mean is

$$\bar{r} = \frac{1}{2G} \sum_{i=1}^{G} (r_i + (-r_i))$$
$$= 0$$

identically, that is, exactly zero for every realization rather than in expectation.

Step 3: group standard deviation. Since $\bar{r} = 0$, the group standard deviation is

$$\sigma = \sqrt{\frac{1}{2G} \sum_{i=1}^{G} (r_i^2 + r_i^2)}$$
$$= \sqrt{\frac{1}{G} \sum_{i=1}^{G} r_i^2}.$$

If $\sigma = 0$, then all $r_i = 0$ and all normalized advantages are zero, so the gradient is zero. Otherwise $\sigma > 0$ and the normalized advantages below are well-defined.

Step 4: advantages. The advantages of the row-role and the column-role response i are

$$\hat{A}_i^{\text{row}} = r_i/\sigma,$$
$$\hat{A}_i^{\text{col}} = -r_i/\sigma.$$

Step 5: per-response cancellation. Since each response k proposes a complete strategy pair and is evaluated in both roles, the *same* output y_k appears in both gradient terms. By Step 4 the two advantages sum to zero, so the GRPO gradient contribution from response k is:

$$\begin{aligned}\hat{A}_k^{\text{row}} \nabla \log \pi_\theta(y_k) + \hat{A}_k^{\text{col}} \nabla \log \pi_\theta(y_k) \\ = \big(\hat{A}_k^{\text{row}} + \hat{A}_k^{\text{col}}\big) \nabla \log \pi_\theta(y_k) \\ = 0 \cdot \nabla \log \pi_\theta(y_k) \\ = \mathbf{0}.\end{aligned} \tag{10}$$

Step 6: summation and clipping. Summing over all G responses, the on-policy unclipped strategic gradient is $\mathbf{0}$ exactly, for every realization, not only in expectation. At likelihood ratio one the PPO clipping operation is inactive, so the same statement holds for the initial clipped-surrogate gradient. After an update moves a ratio outside the clipping interval, the positive- and negative-advantage terms can select different branches and need not cancel. A KL regularization term is also outside this cancellation statement. □

F Registered Analyses

Before re-scoring the released audit records, computing the padding calibration, and running the chat-template base, we wrote a registration with inputs, endpoints, and decision rules. It was written after the training runs, the evaluation games, and the audit grids had been fixed and run, and it covers these re-analyses and new runs; the grids were fixed before evaluation by the audit's own run protocol. The calibrated padding control, the format-only control, and the second-checkpoint audit were registered in addenda before any of their generations, and the record analyses of Table 13 in a further addendum before they were computed. Later addenda register, before any of their computations, the iterative reference solvers of Table 8, a decomposition of the audit label effects in Table 14, and its sensitivity analyses in Table 15. Table 11 lists the confirmatory primary endpoints. The paired and threshold comparisons of Tables 3 and 7 and the leave-one-identity-out analysis are post-hoc descriptive analyses.

Table 11: Registered primary endpoints. Audit effects are correct-label minus anonymous pass@1 in percentage points, with 95% CIs from the grid-then-case bootstrap. Padding counts are games passed out of 50 by the payoff-blind policy. OOD values are $s@0.10$ / pass@1 at 5×5.

Analysis	**Endpoint**	**Result**
Audit, primary, 3 grids, 1,024 tokens	raw / chat / raw minus chat	24.6 $[19.1, 30.0]$ / 1.6 $[-0.3, 3.4]$ / 23.0 $[17.0, 29.0]$
Padding calibration, primary	dominated 12×12, range-normalized / core-span	45 / 6
Calibrated padding control, primary	pooled SFT over 3 seeds, permuted dominated 12×12, core-span; comparator: uniform on the core positions	0.33, 49/150; comparator 0.12, 6/50
Format comparators, primary	SFT / chat-template base / format-only SFT	0.61, 0.33 / 0.16, 0.06 / 0.15, 0.05
Second-checkpoint audit, primary	Qwen3-8B, raw minus chat, 3 grids, 1,024 tokens	4.78 $[2.22, 7.44]$

Table 12 lists the registered secondary quantities of the second-checkpoint audit and the registered game-level test of the calibrated padding control. The game-level aggregation, which compares the mean best-of-4 core-span success over the three SFT seeds within each game with the comparator, drops ties, and uses a 50-game cluster bootstrap with 20,000 replicates, was fixed in an addendum to the registration after the responses had been generated and before this analysis was run. The VERGE seed-7 arm of this control is secondary and is not pooled with SFT.

Table 13 lists the registered analyses of existing records: re-scoring the saved full responses of the chat-template base, the format-only control, and the SFT adapters on unpadded 3×3 and permuted 12×12 games against the displayed one-decimal matrix, a payoff-blind reference policy that answers with LP equilibria of other games, and the descriptive interface interaction on the six canonical zero-sum identities. No new generation was run for them.

Table 12: Registered secondary quantities in percentage points unless counts, with 95% CIs. Qwen3-8B audit: 1,024 tokens, three grids, 900 paired cases per interface; label effects are correct-label minus anonymous pass@1. Calibrated padding control: permuted dominated 12×12, core-span score, best-of-4.

Quantity	Result
Qwen3-8B raw label effect	2.11; grid-then-case $[0.44, 3.78]$; identity $[-0.33, 6.00]$
Qwen3-8B chat label effect	-2.67; grid-then-case $[-5.33, -0.56]$; identity $[-7.00, 0.67]$
Qwen3-8B raw minus chat	4.78; identity $[0.89, 9.33]$; per grid 6.00 / 3.00 / 5.33 for seeds 99 / 456 / 789; post-hoc leave-one-identity-out range 3.21–5.56
Padding control, game-level test, SFT seeds pooled within game vs. uniform on the core positions	games SFT higher / comparator higher / tied: 23 / 5 / 22; exact two-sided sign test $p = 0.00091$; difference 20.67 $[6.67, 34.67]$
Padding control, registered core-solving rule: success ≥ 0.50, advantage ≥ 30 points, and game-level $p < 0.01$	success 0.33; advantage 20.67 points; game-level $p = 0.00091$
Padding control, VERGE seed 7, secondary	$5/50$

Table 13: Registered analyses of existing records with exact rational scoring and strict threshold 0.10. Flag changes are games whose best-of-4 / pass@1 success changes when the same responses are scored on the displayed instead of the unrounded matrix. LP-label prior: mean and 2.5–97.5% range over 1,000 draws of four labels per game from 1,000 LP-solved games of the same size and generator. Audit subset: 1,024 tokens, three grids, percentage points with 95% CIs.

Quantity	Result
Displayed matrix, chat-template base, flag changes at $n = 2, \ldots, 7$	1/0, 1/0, 3/1, 0/1, 1/0, 1/0
Displayed matrix, format-only seeds 42 / 123 / 7, flag changes at $n = 2, \ldots, 7$	2/0, 1/0, 2/1, 1/0, 0/0, 3/1 / 0/0, 4/1, 1/0, 1/0, 2/0, 1/0 / 2/0, 0/0, 0/0, 0/0, 1/0, 1/0
Displayed matrix, SFT seeds 42 / 123 / 7, unpadded 3×3	0/0 each; solved 49/44, 49/45, 49/41 under both matrices
Displayed matrix, SFT seeds 42 / 123 / 7, permuted dominated 12×12, core-span	3/0, 1/0, 2/1
Displayed matrix, 5×5: main-table SFT means, not re-scored, minus the better comparator re-scored on the displayed matrix	45.0 best-of-4 / 28.3 pass@1 points; rule that both are ≥ 20 and SFT 3×3 changes are ≤ 3: met
LP-label prior, best-of-4 at $n = 3, \ldots, 7$	0.35 $[0.22, 0.46]$, 0.20 $[0.10, 0.30]$, 0.11 $[0.04, 0.20]$, 0.06 $[0.00, 0.14]$, 0.04 $[0.00, 0.10]$
LP-label prior, pass@1 at $n = 3, \ldots, 7$	0.10 $[0.02, 0.20]$, 0.06 $[0.00, 0.12]$, 0.03 $[0.00, 0.08]$, 0.02 $[0.00, 0.06]$, 0.01 $[0.00, 0.04]$
LP-label prior, rule at 5×5: upper range < 0.41 best-of-4 and < 0.23 pass@1	met
Audit, six canonical zero-sum identities, raw minus chat	29.3; grid-then-case $[20.6, 37.6]$; identity $[6.3, 53.5]$; label effect raw 30.6, chat 1.3

Table 14 lists a further registered analysis of the audit records: an exact decomposition of each label effect into a validity part and a part due to the success of valid answers,

$$\Delta(vc) = \Delta v \, \frac{c_{\mathrm{c}} + c_{\mathrm{a}}}{2} + \Delta c \, \frac{v_{\mathrm{c}} + v_{\mathrm{a}}}{2},$$

where v is the valid rate and c the success rate among valid answers, and the subscripts c and a denote the correct-label and anonymous conditions. Its registered rule, that for Qwen3.5-9B under raw prompting the validity part exceeds the quality part and its identity interval excludes 0, is met. All quantities of Table 14 and the sensitivity analyses of Table 15 are descriptive analyses of existing records: the valid answers compared under the two conditions come from different subsets of cases, so the decomposition describes where the effect appears rather than isolating a causal effect on answer quality.

Table 14: Registered decomposition of the audit label effects at 1,024 tokens over three grids, with 900 paired named cases per interface, in percentage points with 95% identity-bootstrap intervals from 20,000 replicates. Valid and success-among-valid rates are anonymous → correct label.

Checkpoint	Contrast	Valid rate, %	Success among valid, %	Effect	Validity part	Quality part
Qwen3.5-9B	raw	$19.8 \to 50.2$	$78.1 \to 79.6$	24.6	24.0 [10.3, 38.8]	0.5 [−7.2, 6.5]
	chat	$99.8 \to 99.9$	$53.3 \to 54.8$	1.6	0.1 [0.0, 0.2]	1.5 [−1.9, 4.8]
	raw minus chat			23.0	24.0 [10.2, 38.8]	−1.0 [−9.2, 6.0]
Qwen3-8B	raw	$23.7 \to 24.8$	$15.0 \to 22.9$	2.1	0.2 [−0.1, 1.0]	1.9 [−0.3, 5.0]
	chat	$100 \to 100$	$51.1 \to 48.4$	−2.7	0.0	−2.7 [−7.0, 0.7]
	raw minus chat			4.8	0.2 [−0.1, 1.0]	4.6 [0.8, 8.9]

Table 15: Registered descriptive sensitivity analyses of Table 14 in percentage points with 95% identity-bootstrap intervals from 20,000 replicates. Both-valid pairs: paired cases whose anonymous and correct-label answers are both valid, with the success rates of the two conditions on them. Leave-one-out: range of each part over the ten single-identity removals. Identities with higher raw valid rate: identities whose raw-interface valid rate is higher with the correct label than anonymously.

Quantity	Qwen3.5-9B	Qwen3-8B
Both-valid pairs, raw: pairs; success	127; $80.3 \to 70.1$, −10.2 [−39.1, 16.9]	194; $12.9 \to 14.9$, 2.1 [−1.3, 7.6]
Both-valid pairs, chat: pairs; success	898; $53.3 \to 54.8$, 1.4 [−1.9, 4.8]	900; $51.1 \to 48.4$, −2.7 [−7.1, 0.7]
Leave-one-out, raw validity / quality part	20.4–27.3 / −1.1–3.8	0.0–0.3 / 0.4–2.3
Leave-one-out, interaction validity / quality part	20.3–27.3 / −2.8–2.1	0.0–0.3 / 3.0–5.3
Identities with higher raw valid rate	8 of 10	4 of 10