Breaking the 1/2 Barrier for Well-Supported Nash Equilibria in Win–Lose Games

Shiva Kintali*

October 18, 2026

Abstract

We give two deterministic polynomial-time algorithms for well-supported Nash equilibria in win–lose bimatrix games. An elementary witness-accumulation argument achieves error 0.4817282. A capped primal–dual linear program and chronological payoff peeling improve the guarantee to $(\sqrt{13}-1)/6 \approx 0.434259$. Both algorithms extend to rational nonnegative games satisfying $R_{ij} + C_{ij} \leq 1$ for every outcome. The stronger algorithm returns rational strategies attaining the exact algebraic bound; its analysis uses a bilinear potential to control support survival during deletion.

Keywords: bimatrix games; win–lose games; well-supported Nash equilibrium; approximation algorithms; linear programming.

1 Introduction

For $\varepsilon \geq 0$, an ε-well-supported Nash equilibrium (ε-WSNE) is a pair of mixed strategies in which every pure action played with positive probability has expected payoff within ε of that player's best-response payoff against the opponent's strategy. This bounds each player's expected regret, the probability-weighted average of the corresponding supported-action gaps; a small average alone can conceal a poor supported action. We give two polynomial-time constructions below 1/2 for win–lose bimatrix games (payoffs in $\{0, 1\}$). Both extend to a subclass of normalized games (payoffs in $[0, 1]$).

Related work. Kontogiannis and Spirakis [11] gave polynomial-time WSNE guarantees of 2/3 for general normalized bimatrix games and 1/2 for win–lose games. General guarantees improved to approximately 0.6608 using probability shifting and small subgames [10], then to 0.6528 using separate zero-sum optimizations [5]. Deligkas, Fasoulakis, and Markakis [8] obtain $(1/2 + \eta)$ for every fixed $\eta > 0$ in polynomial time. Liu's preprint [14] claims an exact 1/2 guarantee by support contraction; our proofs do not depend on that claim. Related results treat symmetric games [6] and quasipolynomial near-half approximation [9].

Sampling yields logarithmic-support approximations [13]; for WSNE, sampling and support enumeration give quasipolynomial approximation schemes [11]. Win–lose constructions rule out universal constant support bounds below 2/3 [3], and indeed at every fixed WSNE error below one [2]. Our algorithms allow supports to grow with the input. Computing an exact equilibrium, equivalently a 0-WSNE, is PPAD-complete even for win–lose games [1, 4]; this does not itself locate a constant-error hardness threshold. Expected-regret bounds in [7, 12] do not alone imply the same WSNE bounds.

*Email: shiva.kintali@gmail.com Homepage: shivakintali.github.io

Results. Let $m, n \geq 1$ be integers, and let $J = \mathbf{1}_m \mathbf{1}_n^\top$ denote the $m \times n$ all-ones matrix. Both constructions use the class

$$R, C \in \mathbb{Q}^{m \times n}, \qquad R \geq 0, \quad C \geq 0, \quad R + C \leq J. \tag{1.1}$$

All matrix inequalities are entrywise, so these hypotheses imply normalized payoffs. Complexity is measured in the total binary input length. Define

$$E_0 = \frac{2408641}{5000000} = 0.4817282, \qquad \varepsilon_* = \frac{\sqrt{13} - 1}{6} = 0.434258545910664\ldots. \tag{1.2}$$

Theorem 1.1 (Stronger guarantee)**.** *There is a deterministic algorithm that, given any game satisfying* (1.1)*, returns rational mixed strategies forming an ε_*-WSNE in time polynomial in the binary input length. It uses at most two zero-sum security computations and one capped primal–dual LP, followed by at most $m + n$ whole-action deletions.*

Theorem 1.2 (Elementary guarantee)**.** *There is a deterministic algorithm that, given any game satisfying* (1.1)*, returns rational mixed strategies forming an E_0-WSNE in time polynomial in the binary input length. It uses at most seventeen zero-sum matrix-game computations, each on at most m rows and n columns up to transposition.*

Corollary 1.3. *Both guarantees hold for every win–lose bimatrix game.*

Proof. If a binary input has a cell with $R_{ij} = C_{ij} = 1$, its pure profile is an exact Nash equilibrium: each player receives its maximum possible payoff. Otherwise every cell has payoff sum at most one, so (1.1) holds. This preprocessing is standard [11, Theorem 3]. □

Table 1 compares the two constructions.

Table 1: The two guarantees and their mechanisms. Optimization counts are worst-case upper bounds; both algorithms return rational strategies.

Construction	WSNE error	Main mechanism	Optimization and postprocessing
Elementary	$E_0 = 0.4817282$	Accumulate and average separating witnesses; condition both supports.	At most 17 zero-sum computations, including security; averaging and conditioning.
Stronger	$\varepsilon_* \approx 0.434259$	A capped primal–dual LP and chronological joint charging.	At most two zero-sum computations and one capped LP; at most $m + n$ deletions.

Contributions and proof strategy. Both proofs start with two zero-sum security computations and exploit $R + C \leq J$. The elementary proof accumulates separating column witnesses. Its key invariant allows each original row to have high payoff against at most one stored witness; averaging and simultaneous support conditioning then give E_0. The stronger proof replaces this sequence with a capped primal–dual LP. Complementary slackness gives a direct construction in the low-witness branch. In the high-witness branch, we delete actions sequentially and charge each pair of deleted actions once, at its later deletion. A bilinear potential proves support survival. For $1/3 \leq E < 1/2$, the resulting uniform sufficient condition is $3E^2 + E - 1 \geq 0$, whose positive root is ε_*. Strict separation of the remaining rational security value from ε_* permits an input-dependent

rational tolerance below ε_*, keeping the LP coefficients and output strategies rational. Throughout, deviations range over all original actions, including those assigned zero probability.

The elementary proof isolates an independent witness invariant and averaging mechanism. Our results answer the question posed by Kontogiannis and Spirakis [11, Section 6]: whether a well-supported Nash equilibrium with a fixed approximation error strictly below $1/2$ can be computed in polynomial time for every win–lose bimatrix game. Neither algorithm establishes a universal below-half guarantee for arbitrary normalized games. The instance-dependent bound of [11, Theorem 4],

$$\frac{1}{2}\left(\max_{i,j}(R_{ij}+C_{ij})-\min_{i,j}(R_{ij}+C_{ij})\right),$$

can still be better on individual inputs, and constant-sum games are exactly solvable by linear programming.

Organization. Section 2 develops security preprocessing and conditioning estimates. Section 3 gives the elementary construction. Section 4, readable directly after the preliminaries, develops the stronger construction through the capped LP and chronological peeling. Section 5 implements the algebraic endpoint with rational arithmetic and proves polynomial bit complexity. Section 6 compares the constructions and discusses extensions and limitations. Appendix A contains the worked examples.

2 Preliminaries

An $m \times n$ bimatrix game consists of real payoff matrices (R, C), where $m, n \geq 1$ are integers. It is *normalized* if every payoff belongs to $[0, 1]$, and *win–lose* if every payoff belongs to $\{0, 1\}$. Throughout the proofs we assume (1.1), except where a lemma explicitly states weaker hypotheses. Polynomial time means polynomial in the total binary encoding length of the rational input.

For a positive integer r, write $[r] = \{1, \ldots, r\}$ and

$$\Delta_r = \{x \in \mathbb{R}^r_{\geq 0} : \mathbf{1}^\top x = 1\}.$$

The support of a probability vector is $\operatorname{supp}(x) = \{i : x_i > 0\}$. For a nonempty $S \subseteq [m]$, let $\Delta(S) = \{x \in \Delta_m : \operatorname{supp}(x) \subseteq S\}$; equality of the support with S is not required. A mixed profile is a pair $(x, y) \in \Delta_m \times \Delta_n$. The row and column players' expected payoffs are $x^\top R y$ and $x^\top C y$, respectively. Denote row i of R by R_i and column j of C by C_j. A profile is rational when all coordinates of both strategies are rational; its approximation tolerance may be irrational.

The pure-action gaps, measured against *all original actions*, are

$$D_i(y) = \max_{r \in [m]} R_r y - R_i y, \tag{2.1}$$

$$F_j(x) = \max_{s \in [n]} x^\top C_s - x^\top C_j. \tag{2.2}$$

For normalized payoffs these gaps lie in $[0, 1]$. For $\varepsilon \geq 0$, the profile (x, y) is an ε-WSNE precisely when

$$D_i(y) \leq \varepsilon \quad (i \in \operatorname{supp}(x)), \qquad F_j(x) \leq \varepsilon \quad (j \in \operatorname{supp}(y)).$$

An action of probability zero imposes no supported-action condition, but remains included in the

maxima in (2.1) and (2.2). The ordinary expected regrets are

$$\max_r R_r y - x^\top R y = \sum_i x_i D_i(y),$$
$$\max_s x^\top C_s - x^\top C y = \sum_j y_j F_j(x).$$

Thus a small average gap need not bound every positive-probability gap, whereas every ε-WSNE has expected regret at most ε for each player. In particular, a 0-WSNE is exactly a Nash equilibrium.

The fixed targets E_0 and ε_* are defined in (1.2). A branch's working tolerance E is fixed at E_0 in the elementary construction and is a rational parameter in Section 4. The endpoint algorithm preprocesses at ε_*, then uses $E = e < \varepsilon_*$ in its capped stage (Section 5). Table 2 collects the recurring quantities; additional parameters are defined where they enter the algorithms.

Table 2: Notation for a branch after security preprocessing and any player exchange. In a nonterminal branch, $E < a < U$.

Quantity	Meaning
E	Working WSNE tolerance; the endpoint algorithm's capped stage uses the input-dependent rational value $e < \varepsilon_*$.
$U = 1 - E$	Complement of the working tolerance.
$a = \mathrm{val}(R)$	Row-security value in the current player orientation.
$z = U - a$	Deficit budget for the elementary fallback or chronological peeling.
c	Slack in the all-row cap $U + c$; positive in the elementary fallback and zero in the stronger construction.

2.1 Security preprocessing

For a real $r \times s$ matrix A, with $r, s \geq 1$, the zero-sum row value is

$$\mathrm{val}(A) = \max_{x \in \Delta_r} \min_{j \in [s]} (x^\top A)_j = \min_{y \in \Delta_s} \max_{i \in [r]} (Ay)_i.$$

We use minimax duality and polynomial-time rational linear programming in their standard forms [15]. For rational data, fix a deterministic exact LP procedure returning rational optima of polynomial encoding length. The payoff arguments hold for every optimal solution.

A row-security saddle $(p, q_0) \in \Delta_m \times \Delta_n$ for R, of value $a = \mathrm{val}(R)$, satisfies

$$(p^\top R)_j \geq a \quad (j \in [n]), \qquad R_i q_0 \leq a \quad (i \in [m]), \qquad R_i q_0 = a \quad (i \in \mathrm{supp}(p)). \tag{2.3}$$

The support equalities follow from $p^\top R q_0 = a$: all terms $R_i q_0$ are at most a, so every term with positive weight p_i must equal a.

For the column-payoff security game define

$$\sigma = \max_{y \in \Delta_n} \min_{i \in [m]} (Cy)_i = \min_{x \in \Delta_m} \max_{j \in [n]} x^\top C_j = \mathrm{val}(C^\top).$$

Let $(u, v) \in \Delta_m \times \Delta_n$ be its saddle, with u the row minimizer and v the column maximizer. Then

$$u^\top C_j \leq \sigma \quad (j \in [n]), \qquad (Cv)_i \geq \sigma \quad (i \in [m]), \qquad u^\top C_j = \sigma \quad (j \in \mathrm{supp}(v)). \tag{2.4}$$

The supported-column equalities follow similarly from $u^\top C v = \sigma$. For normalized payoffs, $a, \sigma \in [0, 1]$. These are separate zero-sum computations for the two payoff matrices. Throughout, R_i denotes a row and C_j a column; we write $(p^\top R)_j$ and $(Cq)_i$ for the opposite orientations.

Lemma 2.1 (Preliminary branches)**.** *Assume* (1.1) *and the security identities* (2.3)–(2.4)*. Let* $0 < E < 1/2$ *and* $U = 1 - E$*. Apply the following tests in order, returning at the first applicable rule; each output is an* E*-WSNE:*

1. *If* $a \geq U$*, return* (p, q_0)*.*
2. *If* $\sigma \geq U$*, return* (u, v)*.*
3. *If* $a \leq E$ *and* $\sigma \leq E$*, return* (u, q_0)*.*

If none applies, exchanging the roles of the two players when necessary gives a row-security saddle whose value satisfies $E < a < U$*.*

Proof. In the first branch, every supported row has payoff a, the maximum against q_0, by (2.3). For every original column j,

$$p^\top C_j \leq 1 - (p^\top R)_j \leq 1 - a \leq E. \tag{2.5}$$

Column payoffs are nonnegative, so every supported column has gap at most E, regardless of its probability.

In the second branch, every supported column is an exact best response by (2.4). Every original row satisfies $R_i v \leq 1 - (Cv)_i \leq 1 - \sigma \leq E$, and its payoff is nonnegative. Thus every supported row has gap at most E.

In the third branch, every original row payoff against q_0 is at most $a \leq E$, and every original column payoff against u is at most $\sigma \leq E$. Nonnegativity again bounds both supported-action gaps by E.

If none applies, both $a, \sigma < U$ and at least one is greater than E. If $a \leq E$, replace (R, C) by $(C^\top, R^\top)$. The new row-security saddle is the already computed pair (v, u), of value $\sigma \in (E, U)$, so no further zero-sum computation is needed. If $a > E$, no exchange is needed. The transformation preserves (1.1); a profile returned in the exchanged game is converted back by exchanging its two strategies. □

2.2 Conditioning estimates

The following conditioning estimate keeps the normalization denominators explicit. Both deletion decisions use the original, unconditioned profile.

Lemma 2.2 (Conditioning)**.** *Let* $R, C \in [0,1]^{m \times n}$*, and let* (x, y) *be a mixed profile. For thresholds* $t_R, t_C \geq 0$*, remove from* x *every positive coordinate with* $D_i(y) > t_R$ *and from* y *every positive coordinate with* $F_j(x) > t_C$*. Assume the removed masses satisfy* $0 \leq \alpha, \beta < 1$*, respectively, and let* x', y' *be the conditioned distributions. Then every positively supported action in* (x', y') *has gap at most*

$$\frac{t_R + \beta}{1 - \beta} \quad \textit{for the row player,} \qquad \frac{t_C + \alpha}{1 - \alpha} \quad \textit{for the column player.} \tag{2.6}$$

In particular, for $E \geq 0$*, the profile* (x', y') *is an* E*-WSNE if*

$$t_R + (1 + E)\beta \leq E, \qquad t_C + (1 + E)\alpha \leq E. \tag{2.7}$$

Threshold equalities are retained.

Proof. Let A and B be the deleted row and column sets. Then

$$x'_i = \frac{x_i \mathbf{1}_{i \notin A}}{1 - \alpha}, \qquad y'_j = \frac{y_j \mathbf{1}_{j \notin B}}{1 - \beta}.$$

The denominators are positive by hypothesis, and both vectors sum to one. Fix $i \in \operatorname{supp}(x')$ and any original row $r \in [m]$. Retention gives $(R_r - R_i)y \le t_R$. Since $R_{rj} - R_{ij} \ge -1$ for every j,

$$(R_r - R_i)y' = \frac{(R_r - R_i)y - \sum_{j \in B} y_j (R_{rj} - R_{ij})}{1 - \beta} \le \frac{t_R + \beta}{1 - \beta}.$$

Maximizing over every original r proves the row assertion. For $j \in \operatorname{supp}(y')$ and any original column $s \in [n]$, retention gives $x^\top (C_s - C_j) \le t_C$, and $C_{is} - C_{ij} \ge -1$. Hence

$$(x')^\top (C_s - C_j) = \frac{x^\top (C_s - C_j) - \sum_{i \in A} x_i (C_{is} - C_{ij})}{1 - \alpha} \le \frac{t_C + \alpha}{1 - \alpha}.$$

Maximizing over every original s proves the column assertion, including deviations to deleted or originally unplayed actions. Zero coordinates remain zero after conditioning. Finally, multiplying by the positive denominators shows that (2.7) is equivalent to both bounds in (2.6) being at most E. □

Lemma 2.3 (Mass bound from a support cap)**.** *Let $R \in \mathbb{R}^{m \times n}$, let $(x, y) \in \Delta_m \times \Delta_n$, and suppose $M \in \mathbb{R}$ and $c, A \ge 0$ satisfy*

$$\max_{r \in [m]} R_r y \le M + c, \tag{2.8}$$

$$R_i y \le M \quad (i \in \operatorname{supp}(x)), \tag{2.9}$$

$$\sum_{i \in \operatorname{supp}(x)} x_i (M - R_i y) \le A. \tag{2.10}$$

For any $t > c$, the total x-mass of actions with $D_i(y) > t$ is at most $A/(t - c)$.

Proof. Let $B = \{i \in \operatorname{supp}(x) : D_i(y) > t\}$ and $\gamma = \sum_{i \in B} x_i$. For every $i \in B$,

$$R_i y < \max_r R_r y - t \le M + c - t, \qquad M - R_i y > t - c > 0.$$

Every deficit $M - R_i y$ on $\operatorname{supp}(x)$ is nonnegative by (2.9). Therefore

$$(t - c)\gamma \le \sum_{i \in B} x_i (M - R_i y) \le \sum_{i \in \operatorname{supp}(x)} x_i (M - R_i y) \le A.$$

Division by $t - c > 0$ proves the claim. The same argument covers an empty set B and $A = 0$; in the latter case $\gamma = 0$. □

The support cap M need not bound an unplayed deviation. Using only the resulting expected-regret estimate

$$\sum_i x_i D_i(y) \le M + c - x^\top R y \le A + c$$

would instead give a bad-action mass bound of $(A + c)/t$. The separate support cap yields the mass bound $A/(t - c)$ used in the elementary fallback.

3 The elementary construction

We prove Theorem 1.2 under (1.1), with $E = E_0$. If preprocessing does not return a profile, the oriented row-security saddle satisfies $E < a < 1 - E$. A separation computation on the current row set either supplies a row strategy capping every column payoff by E, or produces a column witness. In the first case we stop, combining that row strategy with a mixture of the security opponent and stored witnesses. In the second, we store the witness and, before the next separation, admit rows with high payoff against it.

The key invariant allows each original row to have high payoff against at most one witness. It controls row gaps at an early stop and the average payoffs used in the fallback. After a bounded number of witnesses, the fallback conditions both strategies on actions whose gaps do not exceed fixed thresholds. Section 3.2 verifies these two outcomes, and Section 3.3 completes the proof and complexity analysis.

3.1 Algorithm and parameters

Fix the rational constants

$$\begin{aligned} E &= \frac{2408641}{5000000} = 0.4817282, \qquad U = 1 - E = \frac{2591359}{5000000} = 0.5182718, \\ h &= U + \frac{1}{1000000} = \frac{647841}{1250000} = 0.5182728. \end{aligned} \tag{3.1}$$

In particular, $0 < E < U < h < 1$ and $h - U = 10^{-6}$.

Preprocessing and parameter choice. Apply Lemma 2.1 with the rational tolerance $E = E_0$. If it returns a profile, the elementary guarantee is already proved. Otherwise apply the player exchange from that lemma if needed, write R, C, m, n for the resulting game and its dimensions, and retain the row-security saddle (p, q_0) with $E < a < U$. Record any exchange and reverse it when returning either of the profiles below.

The subsequent estimates use the closed interval $E \le a \le U$; the actual nonterminal branch lies in its interior. Choose parameters according to Table 3, and put $T = K - 1$. In both branches $0 < d < E$ and $w > 0$.

Table 3: Exact parameters. Here K is the fallback witness count, and T is the last possible early-stop stage.

Security-value interval	K	T	d	w
$E \le a \le 121/250$	15	14	6/25	3/50
$121/250 < a \le U$	13	12	1/4	71/1000

Accumulation and early stopping. Initialize

$$S_0 = \{i \in [m] : R_i q_0 \ge a - d\}, \qquad S = S_0. \tag{3.2}$$

By (2.3), $\operatorname{supp}(p) \subseteq S_0$, so S is nonempty. For any nonempty current set S, define

$$c(S) = \min_{x \in \Delta(S)} \max_{j \in [n]} x^\top C_j = \max_{q \in \Delta_n} \min_{i \in S} (Cq)_i. \tag{3.3}$$

The equality is zero-sum minimax duality on the restricted row set and the *full original* column set. Here the row distribution minimizes column payoffs, while q maximizes their minimum over S.

For $k = 0, 1, \ldots, K-1$, with $q_1, \ldots, q_k$ already stored, the current set is $S = S_0 \cup \bigcup_{t=1}^{k} H_t$; the indexed union is empty when $k = 0$. Perform the following steps:

1. Compute $c(S)$, a minimizing distribution $x_S \in \Delta(S)$, and a maximizing distribution for (3.3).

2. If $c(S) \le E$, return
$$x = x_S, \qquad y = (1 - kw)q_0 + w \sum_{t=1}^{k} q_t. \tag{3.4}$$
For $k = 0$, the sum is empty and $y = q_0$.

3. If $c(S) > E$, store a maximizer q_{k+1} satisfying
$$(Cq_{k+1})_i \ge c(S) > E \quad \text{for every } i \in S. \tag{3.5}$$
If $k + 1 = K$, use the fallback immediately. Otherwise form
$$H_{k+1} = \{i \in [m] : R_i q_{k+1} \ge h\}, \qquad S \leftarrow S \cup H_{k+1}, \tag{3.6}$$
and continue to the next iteration, even if H_{k+1} is empty. There is no extra test after the Kth witness and no need to form H_K.

For every possible early stop, $1 - kw \ge 1 - Tw > 0$; the two lower bounds are $4/25$ and $37/250$. Hence (3.4) is a probability distribution.

Fallback and conditioning. After K witnesses have been stored, set
$$\bar{q} = \frac{1}{K} \sum_{t=1}^{K} q_t, \qquad V = h + \frac{1-h}{K}, \qquad c = V - U, \qquad t_R = \frac{E + c}{2}, \quad t_C = \frac{E}{2}. \tag{3.7}$$
Starting from $(p, \bar{q})$, use the full-game gaps (2.1)–(2.2) to define
$$I_R = \{i \in \mathrm{supp}(p) : D_i(\bar{q}) \le t_R\}, \tag{3.8}$$
$$I_C = \{j \in \mathrm{supp}(\bar{q}) : F_j(p) \le t_C\}. \tag{3.9}$$
Let $\alpha = \sum_{i \notin I_R} p_i$ and $\beta = \sum_{j \notin I_C} \bar{q}_j$. Return
$$x_i = \begin{cases} p_i/(1-\alpha), & i \in I_R, \\ 0, & i \notin I_R, \end{cases} \qquad y_j = \begin{cases} \bar{q}_j/(1-\beta), & j \in I_C, \\ 0, & j \notin I_C. \end{cases} \tag{3.10}$$
We prove below that $0 \le \alpha, \beta < 1$, so this step is well defined. The deletions in (3.8)–(3.9) are simultaneous decisions made from the unconditioned profile $(p, \bar{q})$. Appendices A.1 and A.2 illustrate an early stop after admitting a new row and a fallback with an actual deletion.

3.2 Correctness

Witness invariant.

Lemma 3.1 (At most one high-payoff witness)**.** *Whenever a witness q_t is generated, every row already in the current set S satisfies $R_i q_t < U$. Every original row has payoff at least h against at most one of the stored witnesses. Moreover, every row of S_0 has payoff below U against every stored witness.*

Proof. For a row already in S, (1.1) and (3.5) give

$$R_i q_t \le 1 - (Cq_t)_i < 1 - E = U < h. \tag{3.11}$$

If a row first attains payoff at least h at a witness q_s with $s < K$, it is added to S immediately and stays there. At every subsequent witness its payoff is below U by (3.11). Before that first witness, its payoff is below h by definition. If the first such event occurs at q_K, there is no later witness. Thus there cannot be two such events for any original row. Rows in S_0 belong to S from the start, giving the final assertion. □

Corollary 3.2. *If $1 \le k \le K$ witnesses have been generated, then every original row satisfies*

$$\sum_{t=1}^{k} R_i q_t \le 1 + (k-1)h. \tag{3.12}$$

For $i \in S_0$ the stronger bound $\sum_{t=1}^{k} R_i q_t \le kU$ holds. If a row in the current set S is outside S_0, it has a birth index $s \le k$ with $R_i q_s \ge h$.

Proof. At most one summand in (3.12) can be at least h; that summand is at most one and all others are below h. If there is no such summand, the sum is at most $kh \le 1 + (k-1)h$. The S_0 assertion follows directly from Lemma 3.1. A row outside S_0 can enter S only through (3.6), proving the birth assertion. □

Early stopping.

Lemma 3.3. *Every profile returned by (3.4) is an E-WSNE for the parameters in Table 3.*

Proof. All original column payoffs against x_S are at most $c(S) \le E$. They are also nonnegative, so every positively supported column of y has gap at most E.

If $k = 0$, all supported rows belong to S_0 and have payoff at least $a - d$ against q_0. Every original row has payoff at most a against q_0. Hence the supported row gaps are at most $d < E$, proving this case.

Suppose now $1 \le k \le T$. All coefficients in (3.4) are nonnegative. The supported row classes have the following lower bounds:

$$L_0 = (1 - kw)(a - d) \qquad \text{for supported rows in } S_0, \tag{3.13}$$

$$L_1 = wh \qquad \text{for supported rows in } S \setminus S_0. \tag{3.14}$$

The first follows from (3.2) and nonnegativity of R. For the second, Corollary 3.2 gives a birth witness of weight w against which the row earns at least h; the remaining terms are nonnegative.

Independently, all original deviations split into two classes with upper bounds

$$A_0 = (1 - kw)a + kwU \qquad \text{for original rows in } S_0, \tag{3.15}$$

$$A_1 = (1 - kw)(a - d) + w[1 + (k-1)h] \qquad \text{for original rows outside } S_0. \tag{3.16}$$

For (3.15), use $R_i q_0 \le a$ and the stronger S_0 witness bound. For (3.16), use $R_i q_0 < a - d$ outside S_0 and (3.12). These are bounds on all original rows, whether or not they have entered S.

The four differences that can bound a supported gap are therefore

$$G_1(k, a) = A_0 - L_0 = (1 - kw)d + kwU, \tag{3.17}$$

$$G_2(k, a) = A_0 - L_1 = (1 - kw)a + kwU - wh, \tag{3.18}$$

$$G_3(k, a) = A_1 - L_0 = w[1 + (k-1)h], \tag{3.19}$$

$$G_4(k, a) = A_1 - L_1 = (1 - kw)(a - d) + w[1 + (k-2)h]. \tag{3.20}$$

For clarity, the complete pairing of classes is

	deviation in S_0	deviation outside S_0
supported row in S_0	G_1	G_3
supported row in $S \setminus S_0$	G_2	G_4

so it suffices to bound all four G's by E. Empty classes require no separate argument.

Their differences as k increases by one are respectively

$$w(U-d), \qquad w(U-a), \qquad wh, \qquad w(h-a+d), \tag{3.21}$$

all nonnegative on the parameter intervals. Only G_2 and G_4 depend on a, and their coefficient of a is $1-kw>0$. Consequently the maximum in each parameter branch occurs at $k=T$ and the upper endpoint of its security-value interval. Direct rational substitution gives Table 4, where each terminating decimal is exact.

Table 4: Exact upper bounds for all early stopping stages. Every entry is strictly below $E = 0.4817282$.

Parameter branch	G_1	G_2	G_3	G_4
$K=15$	0.473748312	0.481691944	0.464252784	0.472196416
$K=13$	0.4785675736	0.4814744312	0.4757710568	0.4786779144

For example, the largest entry is $60211493/125000000 = 0.481691944$; its distance below E is $1133/31250000 > 0$. Thus every positively supported row has gap below E, completing the proof. □

Fallback.

Lemma 3.4. *Suppose K witnesses are generated. Set $\bar{q}, V, c$ as in (3.7), and put $z = U - a > 0$. Then*

$$\max_{r\in[m]} R_r\bar{q} \le V = U + c, \tag{3.22}$$

$$R_i\bar{q} < U \quad (i \in \mathrm{supp}(p)), \tag{3.23}$$

$$\sum_{i\in\mathrm{supp}(p)} p_i(U - R_i\bar{q}) \le z, \tag{3.24}$$

$$\sum_{j\in[n]} \bar{q}_j F_j(p) \le z. \tag{3.25}$$

Proof. By (3.12), every original row has average payoff at most $[1+(K-1)h]/K = V$. Since $\mathrm{supp}(p) \subseteq S_0$, Lemma 3.1 proves (3.23). Also (2.3) implies $p^\top R\bar{q} \ge a$. Hence

$$\sum_{i\in\mathrm{supp}(p)} p_i(U - R_i\bar{q}) = U - p^\top R\bar{q} \le U - a = z,$$

with nonnegative summands.

For every original column j, the payoff-sum condition and (2.3) give $p^\top C_j \le 1 - (p^\top R)_j \le 1 - a$. On the other hand, every witness satisfies $(Cq_t)_i > E$ on every row in $\mathrm{supp}(p) \subseteq S_0$. Thus $p^\top Cq_t > E$ and $p^\top C\bar{q} > E$. Consequently

$$\sum_j \bar{q}_j F_j(p) = \max_j p^\top C_j - p^\top C\bar{q} \le 1 - a - E = U - a = z.$$

This proves (3.25). □

Proposition 3.5 (Sufficient fallback inequality)**.** *Suppose K witnesses have been generated, $0 < c < E$, and*

$$4(1+E)(U-a) \le E(E-c). \tag{3.26}$$

Then the fallback profile (3.10) is well defined and is an E-WSNE.

Proof. Here $t_R - c = (E-c)/2 > 0$ and $t_C = E/2 > 0$. Apply Lemma 2.3 to (3.22)–(3.24) with $M = U$ and $A = z$. It gives

$$\alpha \le \frac{z}{t_R - c} = \frac{2z}{E-c}, \qquad \beta \le \frac{z}{t_C} = \frac{2z}{E}. \tag{3.27}$$

For the column estimate, each deleted positive coordinate has gap greater than t_C, while all gaps are nonnegative. Thus

$$t_C\beta \le \sum_{j \notin I_C} \bar{q}_j F_j(p) \le \sum_j \bar{q}_j F_j(p) \le z,$$

which gives the second bound in (3.27).

Inequality (3.26) first ensures that neither distribution is completely deleted, since

$$\alpha \le \frac{E}{2(1+E)} < 1, \qquad \beta \le \frac{E-c}{2(1+E)} < 1. \tag{3.28}$$

It also gives both target inequalities of Lemma 2.2. By (3.27), it suffices to verify

$$\frac{E+c}{2} + (1+E)\frac{2z}{E} \le E,$$
$$\frac{E}{2} + (1+E)\frac{2z}{E-c} \le E,$$

and each of these inequalities is equivalent to (3.26). Lemma 2.2 therefore bounds every supported gap after conditioning by E. All deviations in that lemma range over the original action sets. □

Parameter verification. For fixed K, the number c is constant and $z = U - a$ decreases with a. It suffices to check (3.26) at the lower endpoint of each security-value interval, including the excluded lower endpoint of the second interval as a conservative bound. Put

$$z_{\max} = \begin{cases} U - E = 91359/2500000, & K = 15, \\ U - 121/250 = 171359/5000000, & K = 13. \end{cases}$$

Table 5 gives exact values. Its final column is the positive difference between the right and left sides of (3.26) at $z = z_{\max}$.

Proposition 3.5 applies throughout both security-value intervals. In particular, the fallback requires neither a further separation computation nor a numerical approximation to any constant.

The tight lower-interval check also explains the choice of the target. Writing $\delta = h - U = 10^{-6}$, the branch $K = 15$ has $c = (E + 14\delta)/15$ and $z_{\max} = 1 - 2E$. Its fallback margin is

$$E(E-c) - 4(1+E)z_{\max} = \frac{134}{15}E^2 + \left(4 - \frac{14\delta}{15}\right)E - 4. \tag{3.29}$$

The positive root of this quadratic is approximately 0.4817281940863629, slightly below the rational target $E = 0.4817282$. The exact positive margin in Table 5, rather than this decimal approximation to the root, certifies the inequality. This calculation concerns the selected certificate and does not establish optimality of the algorithm or of the WSNE bound.

Table 5: Exact fallback certificates. In both branches $0 < c < E$ and the displayed margin is strictly positive.

K	c	$z_{\max}$	$E(E-c)-4(1+E)z_{\max}$
15	2408711/75000000	91359/2500000	436831/5859375000000
13	2408701/65000000	171359/5000000	900633200481/81250000000000

3.3 Completion and complexity

Proof of Theorem 1.2. Lemma 2.1 proves correctness of the preliminary branches. If none applies, it supplies the orientation required by the accumulation algorithm. The set S is always nonempty, since it contains $\operatorname{supp}(p)$. At every separation computation either an early stop is taken or a witness is stored by minimax duality. There are at most K such computations. An early stop is correct by Lemma 3.3; otherwise K witnesses are available and the fallback is correct by Proposition 3.5 and Table 5. Thus the algorithm always returns an E-WSNE in the oriented game. If the players were exchanged, swapping the output strategies back preserves these gap bounds in the original game.

There are two initial zero-sum computations and at most $K \le 15$ separation computations, for a total of at most seventeen. Each uses at most m rows and n columns up to transposition; exchanging the players exchanges these two dimensions. For completeness, a separation computation can be implemented by the linear program and its dual

$$\text{minimize } \lambda \quad \text{subject to } x \in \Delta(S), \quad x^\top C_j \le \lambda \ (j \in [n]), \tag{3.30}$$

$$\text{maximize } \mu \quad \text{subject to } q \in \Delta_n, \quad (Cq)_i \ge \mu \ (i \in S). \tag{3.31}$$

Here a zero-sum computation means obtaining the common optimal value and optimal strategies; primal and dual may be solved together or separately. Equivalently, feasibility of both constraint systems with $\lambda = \mu$ is one rational linear program. The count of seventeen refers to matrix-game computations, not to an implementation-dependent count of separately executed primal and dual LP solves. The constant number of solves places no restriction on support cardinalities.

Rational linear programs of polynomial encoding length admit optimal rational strategies with polynomial encoding length, computable in polynomial bit time by standard exact linear programming. The input coefficients in (3.30)–(3.31) are selected from the original rational payoff matrices; the row restrictions only remove variables. Membership tests for S_0 and H_t are exact rational comparisons. Since the number of solves and witnesses is constant, the sums, scalar products, and mixtures defining the output have polynomial encoding length. Conditioning in (3.10) uses positive rational denominators and preserves this property: a ratio of polynomial-bit rationals again has polynomial binary encoding length. All gaps used for deletion and final verification are computable by polynomially many rational arithmetic operations. This proves the claimed deterministic polynomial-time bound. □

Boundary cases. The following conventions are part of the algorithm, and their correctness follows from the non-strict estimates proved above:

- If $a = U$ or $\sigma = U$, a preliminary high-value branch returns. If both values are at most E, the low-value branch returns. The value $a = 121/250$ uses the first parameter branch.
- An action with $R_i q_0 = a - d$ belongs to S_0. An action with $R_i q_t = h$ belongs to H_t whenever that set is formed. The strict bound $R_i q_t < U < h$ on accumulated rows guarantees that equality

at h cannot recur for an already admitted row.

- Equality $c(S) = E$ takes the early stopping branch. An empty birth set does not prevent storing its witness or advancing the loop. At the final witness no new stopping test is needed.
- Equality $D_i(\bar{q}) = t_R$ or $F_j(p) = t_C$ retains the action. The estimates on deleted mass use strict inequality only for the deleted actions. They remain valid as weak upper bounds when the deleted set is empty.
- No positive lower bound on individual strategy coordinates is assumed. Every positive output row belongs to one of the two row classes in the early proof or to the retained set I_R in the fallback. Every positive output column is covered by its corresponding column argument. Exactly zero coordinates are excluded from support tests; their pure actions remain possible deviations.

4 The stronger construction

This section depends only on Section 2. Fix a rational tolerance E with $1/3 \le E < 1/2$. If preprocessing does not return a profile, it supplies an oriented row-security saddle with $E < a < U = 1 - E$. We solve a primal–dual LP that caps every row payoff at U while maximizing a column-payoff floor b on $\mathrm{supp}(p)$. If $b \le E$, the dual multipliers identify two row classes. Their combined weights, paired with a mixture of the security opponent and capped strategy, give an E-WSNE. If $b > E$, chronological peeling gives the guarantee provided $(1 + E)(U - a) \le E^2$.

The peeling proof is independent of the LP. A neutral-cut inequality accounts for the shared deficit budget, and a bilinear potential shows that neither support can disappear. The retained-action inequalities then bound every supported gap. Section 4.3 derives the uniform approximation threshold and states the guarantee using the actual security value; Section 5 implements the algebraic endpoint with rational arithmetic.

4.1 Capped LP and dual construction

For the LP construction fix a rational E with $1/3 \le E < 1/2$, put $U = 1 - E$, and suppose that preprocessing has left $E < a < U$. Keep the row-security strategy p and its saddle opponent q_0, and set $S = \mathrm{supp}(p)$.

Solve the following rational LP:

$$\begin{aligned} \text{maximize} \quad & b \\ \text{subject to} \quad & q \ge 0, \quad \mathbf{1}^\top q = 1, \\ & R_k q \le U \qquad (k = 1, \dots, m), \\ & (Cq)_i \ge b \qquad (i \in S). \end{aligned} \tag{P}$$

Here b is free. The pair $(q_0, 0)$ is feasible because $a < U$ and $C \ge 0$. Also $b \le 1$, since S is nonempty and every $(Cq)_i \le 1$. The optimum is attained: equivalently, maximize the continuous function $\min_{i \in S}(Cq)_i$ over the nonempty compact capped simplex.

Its dual is

$$\begin{aligned} \text{minimize} \quad & \nu + U \textstyle\sum_{k=1}^{m} \mu_k \\ \text{subject to} \quad & \lambda_i \ge 0 \ (i \in S), \quad \textstyle\sum_{i \in S} \lambda_i = 1, \\ & \mu_k \ge 0 \ (k = 1, \dots, m), \quad \nu \in \mathbb{R}, \\ & \lambda^\top C_j - (\mu^\top R)_j \le \nu \qquad (j = 1, \dots, n). \end{aligned} \tag{D}$$

Extend λ by zero outside S. To see the dual explicitly, the Lagrangian for a primal maximization over $b \in \mathbb{R}$ and $q \geq 0$ is

$$b + \sum_{i \in S} \lambda_i((Cq)_i - b) + \sum_k \mu_k(U - R_k q) + \nu(1 - \mathbf{1}^\top q).$$

A finite supremum forces $\sum_i \lambda_i = 1$ and $\lambda^\top C_j - (\mu^\top R)_j \leq \nu$ for every column, leaving the objective in (D).

Choose rational optimal solutions (q, b) and (λ, μ, ν) of polynomial encoding length, and put $M = \sum_k \mu_k$. Strong duality and complementary slackness give

$$b = \nu + UM, \qquad \mu_i > 0 \Longrightarrow R_i q = U. \tag{4.1}$$

For completeness, the duality gap is the sum of the following nonnegative terms:

$$\sum_{i \in S} \lambda_i((Cq)_i - b) + \sum_k \mu_k(U - R_k q) + \sum_j q_j(\nu - \lambda^\top C_j + (\mu^\top R)_j).$$

This sum equals $\nu + UM - b$. At optimum it is zero, which proves (4.1) without any nondegeneracy assumption. Obtaining an optimal primal–dual pair counts as one capped LP computation: it can be implemented by a single rational feasibility LP containing both constraint systems and the linear equality $b = \nu + U \sum_k \mu_k$. Feasibility follows from strong duality, and weak duality makes every feasible pair optimal.

The low-witness branch.

Lemma 4.1 (Dual construction)**.** *If the optimum in* (P) *satisfies $b \leq E$, define*

$$x = \frac{\lambda + \mu}{1 + M}, \qquad w = \frac{a}{a + U}, \qquad y = (1 - w)q_0 + wq. \tag{4.2}$$

Then (x, y) is an E-WSNE.

Proof. The vectors in (4.2) are probability distributions: $\sum_i \lambda_i = 1$, $\sum_i \mu_i = M \geq 0$, and $0 < w < 1$. For every original column j, $R + C \leq J$ and dual feasibility give

$$\begin{aligned}(\lambda + \mu)^\top C_j &\leq \lambda^\top C_j + M - (\mu^\top R)_j \\ &\leq \nu + M = b + EM \leq E(1 + M).\end{aligned}$$

Thus $x^\top C_j \leq E$ for every original j. Since all column payoffs are nonnegative, every positive column of y is E-optimal.

Set $L = aU/(a + U)$. Every original row k has

$$R_k y \leq (1 - w)a + wU = 2L.$$

The choice of w balances the two supported-row guarantees: $(1 - w)a = wU = L$. If $\lambda_i > 0$, then $i \in \operatorname{supp} p$ and $R_i q_0 = a$, so $R_i y \geq (1 - w)a = L$. If $\mu_i > 0$, (4.1) gives $R_i q = U$, so $R_i y \geq wU = L$. Every $x_i > 0$ lies in at least one of these two classes, including their possible overlap. Hence its pure-action gap against any original row is at most

$$L = \frac{aU}{a + U} \leq \frac{U}{2} \leq E.$$

The first inequality uses $a < U$; the second is exactly the condition $E \geq 1/3$. This proves both supported-action requirements, including $M = 0$, $b = E$, and arbitrary zero coordinates of q or q_0. □

Appendix A.3 gives explicit primal–dual certificates, including overlapping row classes and the boundary cases $b = E$ and $M = 0$.

4.2 Chronological payoff peeling

The following lemma allows real parameters and strategies, with a cap slack c. The stronger algorithm uses rational data and $c = 0$.

Lemma 4.2 (Peeling lemma)**.** *Suppose* (1.1) *holds. Let* $0 < E < 1$*, put* $U = 1 - E$*, and suppose* $p \in \Delta_m$*,* $q \in \Delta_n$*,* $a \in [0, U]$*, and* $c \in [0, E)$ *satisfy*

$$\max_k R_k q \le U + c, \tag{4.3}$$

$$(Cq)_i \ge E \qquad (i \in \operatorname{supp} p), \tag{4.4}$$

$$(p^\top R)_j \ge a \qquad (j = 1, \dots, n). \tag{4.5}$$

Define

$$r_i = U - R_i q \quad (i \in \operatorname{supp} p), \qquad d_j = 1 - a - p^\top C_j \quad (j \in \operatorname{supp} q), \tag{4.6}$$

$$z = U - a, \qquad A = E - c, \qquad B = E, \qquad D = 1 + E. \tag{4.7}$$

If

$$Dz \le AB, \tag{4.8}$$

the sequential deletion procedure below returns an E*-WSNE. It terminates after at most* $|\operatorname{supp} p| + |\operatorname{supp} q|$ *deletions, in any sequential order of currently eligible actions. Each player's retained original probability mass is greater than* $1/(1 + E)$*.*

All deficits in (4.6) are nonnegative. Indeed, $R_i q \le 1 - (Cq)_i \le U$ on $\operatorname{supp} p$, and $p^\top C_j \le 1 - (p^\top R)_j \le 1 - a$ for every column. For definiteness set $r_i = 0$ when $p_i = 0$ and $d_j = 0$ when $q_j = 0$; these auxiliary values are used only in terms multiplied by zero. The two total deficits can sum to more than z. Lemma 4.3 accounts explicitly for their overlap through the nonnegative matrix $J - R - C$.

Start with deleted sets $\mathcal{I} = \mathcal{J} = \varnothing$. Use the original, unconditioned probabilities throughout the procedure. A deterministic implementation fixes a total order of the original actions and always chooses the first eligible action. The proof permits any sequential choice. Repeatedly apply either available strict rule:

$$r_i + \sum_{j \in \mathcal{J}} q_j (R_{ij} + E) > A \quad \Longrightarrow \quad \mathcal{I} \leftarrow \mathcal{I} \cup \{i\}, \qquad i \in \operatorname{supp} p \setminus \mathcal{I}, \tag{4.9}$$

$$d_j + \sum_{i \in \mathcal{I}} p_i (C_{ij} + E) > B \quad \Longrightarrow \quad \mathcal{J} \leftarrow \mathcal{J} \cup \{j\}, \qquad j \in \operatorname{supp} q \setminus \mathcal{J}. \tag{4.10}$$

An equality never triggers deletion. When no strict rule applies, set

$$\alpha = \sum_{i \in \mathcal{I}} p_i, \qquad \beta = \sum_{j \in \mathcal{J}} q_j, \qquad x_i = \frac{p_i \, \mathbf{1}_{i \notin \mathcal{I}}}{1 - \alpha}, \qquad y_j = \frac{q_j \, \mathbf{1}_{j \notin \mathcal{J}}}{1 - \beta}. \tag{4.11}$$

The proof shows that both denominators are positive. Appendix A.4 illustrates a strict deletion and checks the final gaps against every original action.

Lemma 4.3 (Fractional neutral-cut inequality)**.** *Under the hypotheses of Lemma 4.2, let* $Z = J - R - C$*. For arbitrary* $u_i, v_j \in [0, 1]$ *on the positive supports, extended by zero outside the respective supports,*

$$\sum_i p_i u_i r_i + \sum_j q_j v_j d_j \le z + \sum_{i,j} p_i q_j u_i v_j Z_{ij}. \tag{4.12}$$

Proof. On $\operatorname{supp} p$ put $\xi_i = (Cq)_i - E \geq 0$, and put $\eta_j = (p^\top R)_j - a \geq 0$ on $\operatorname{supp} q$. Since $R + C + Z = J$,

$$r_i = \xi_i + (Zq)_i, \qquad d_j = \eta_j + (p^\top Z)_j.$$

The displayed identities are used on the respective positive supports. Extend ξ_i and η_j by zero outside those supports in weighted sums. The total available budget is exactly

$$\begin{aligned}\sum_i p_i\xi_i + \sum_j q_j\eta_j + p^\top Zq &= p^\top Cq - E + p^\top Rq - a + p^\top Zq \\ &= 1 - E - a = z.\end{aligned}$$

Using nonnegativity and $u_i + v_j \leq 1 + u_iv_j$ gives

$$\begin{aligned}\sum_i p_iu_ir_i + \sum_j q_jv_jd_j &= \sum_i p_iu_i\xi_i + \sum_j q_jv_j\eta_j + \sum_{i,j} p_iq_j(u_i + v_j)Z_{ij} \\ &\leq \sum_i p_i\xi_i + \sum_j q_j\eta_j + p^\top Zq + \sum_{i,j} p_iq_ju_iv_jZ_{ij},\end{aligned}$$

which is (4.12). Equivalently, the right side of (4.12) minus its left side equals

$$\sum_i p_i(1 - u_i)\xi_i + \sum_j q_j(1 - v_j)\eta_j + \sum_{i,j} p_iq_j(1 - u_i)(1 - v_j)Z_{ij} \geq 0.$$

This identity also covers fractional and zero deletion coordinates. All sums with $p_i = 0$ or $q_j = 0$ vanish. □

Charging each deleted pair once.

Proof of Lemma 4.2. The empty deletion prefix has $\alpha = \beta = 0$, strictly below the positive barriers B/D and A/D. A prefix is an initial segment of the chosen sequential deletion order. Consider now a nonempty deletion prefix. For the proof only, allow the last action in the prefix to be deleted by a fraction $\theta \in (0, 1]$. All earlier deletions remain whole; thus at most one deletion coordinate lies strictly between zero and one. Let u_i, v_j be the resulting deletion fractions, and let

$$\alpha = \sum_i p_iu_i, \qquad \beta = \sum_j q_jv_j.$$

Multiply each strict trigger by the original probability mass deleted at that step and sum. For a partially deleted last action use its partial mass. Its opposite player's deleted set is fixed during that step, so the same strict trigger remains valid. Every multiplier used for an actual or partial deletion is positive, so summing preserves strict inequality.

For a pair (i, j) whose two actions have positive deletion fractions in this prefix, only the later deletion charges the pair. Write $P_{ij} = R_{ij}$ if the row was deleted later and $P_{ij} = C_{ij}$ if the column was deleted later. For all other pairs, choose P_{ij} arbitrarily from $\{R_{ij}, C_{ij}\}$; their product u_iv_j is zero. The sum of triggers therefore yields

$$A\alpha + B\beta < \sum_i p_iu_ir_i + \sum_j q_jv_jd_j + \sum_{i,j} p_iq_ju_iv_j(P_{ij} + E). \tag{4.13}$$

This identity of charges includes a fractional last action: each such pair simply has its charge multiplied by that action's fraction.

Since $Z_{ij} + R_{ij} = 1 - C_{ij} \le 1$ and $Z_{ij} + C_{ij} = 1 - R_{ij} \le 1$, combining (4.12) and (4.13) gives the strict potential bound

$$F(\alpha, \beta) := A\alpha + B\beta - D\alpha\beta < z. \tag{4.14}$$

There is no charge for an undeleted pair and no pair is charged twice.

Interpolate every whole deletion continuously by varying just that action's deletion fraction. This produces an axis-parallel continuous path from $(0, 0)$, and (4.14) holds at every nonempty point of the path. If this path first reached either boundary

$$\alpha = \frac{B}{D} \qquad \text{or} \qquad \beta = \frac{A}{D},$$

then on that boundary $F(\alpha, \beta) = AB/D$. The boundary point is nonempty because both barriers are positive. This contradicts (4.14) and $z \le AB/D$. Consequently every deletion prefix satisfies

$$\alpha < \frac{B}{D} < 1, \qquad \beta < \frac{A}{D} < 1. \tag{4.15}$$

Since $A \le E$ and $B = E$, it follows in particular that $1 - \alpha > 1/(1 + E)$ and $1 - \beta > 1/(1 + E)$. Thus neither support can disappear, even when a whole action has probability larger than one of these barriers. The partial deletion is a proof device; the algorithm deletes whole actions.

It remains to verify all original deviations after conditioning. For every original row k, nonnegativity and (4.3) imply

$$R_k y \le \frac{U + c}{1 - \beta}.$$

For a retained positive row i,

$$R_i y = \frac{U - r_i - \sum_{j \in \mathcal{J}} q_j R_{ij}}{1 - \beta}.$$

Since its strict deletion rule no longer applies,

$$r_i + \sum_{j \in \mathcal{J}} q_j R_{ij} + E\beta \le E - c.$$

Subtracting the preceding payoff bounds gives, for every original k,

$$(R_k - R_i)y \le \frac{c + r_i + \sum_{j \in \mathcal{J}} q_j R_{ij}}{1 - \beta} \le E. \tag{4.16}$$

Similarly, nonnegativity and $p^\top C_\ell \le 1 - a$ give $x^\top C_\ell \le (1 - a)/(1 - \alpha)$ for every original column ℓ. For a retained positive column j,

$$x^\top C_j = \frac{1 - a - d_j - \sum_{i \in \mathcal{I}} p_i C_{ij}}{1 - \alpha}, \qquad d_j + \sum_{i \in \mathcal{I}} p_i C_{ij} + E\alpha \le E.$$

The inequality is precisely its nonviolated deletion rule. Hence

$$x^\top (C_\ell - C_j) \le \frac{d_j + \sum_{i \in \mathcal{I}} p_i C_{ij}}{1 - \alpha} \le E. \tag{4.17}$$

The denominators are positive by (4.15), so (4.16)–(4.17) prove the WSNE conditions. Every iteration deletes a new positive-support action, which proves the stated termination bound. □

4.3 Completion of the guarantee

The high-witness branch.

Lemma 4.4 (Witness followed by peeling)**.** *If $b > E$ and*

$$(1+E)(1-E-a) \le E^2, \tag{4.18}$$

chronological peeling applied to (p, q) returns an E-WSNE.

Proof. Primal feasibility gives $\max_k R_k q \le U$ and $(Cq)_i \ge b > E$ for every $i \in \operatorname{supp} p$. The security saddle gives $(p^\top R)_j \ge a$ for every original j. These are exactly the hypotheses of Lemma 4.2 with $c = 0$, $A = B = E$, and $z = 1 - E - a$. Its budget condition is (4.18). □

Origin of the approximation threshold. In a nonterminal branch, $1/3 \le E < 1/2$ and $E < a < 1 - E$. The high-witness condition (4.18) has the exact slack

$$E^2 - (1+E)(1-E-a) = 3E^2 + E - 1 + (1+E)(a-E). \tag{4.19}$$

Thus $3E^2 + E - 1 \ge 0$ makes this slack strictly positive for every remaining security value a. The polynomial $3E^2 + E - 1$ has positive root $\varepsilon_* = (\sqrt{13} - 1)/6$. Together with the low branch's restriction $E \ge 1/3$, this yields the uniform sufficient bound $E \ge \varepsilon_*$ for the two branches. The corollary below takes rational $E > \varepsilon_*$. The instance-dependent guarantee in Proposition 4.6 also permits smaller E when the actual-security condition holds; Section 5 uses this flexibility to attain the endpoint with a rational working tolerance $e < \varepsilon_*$.

Corollary 4.5 (Fixed rational tolerances)**.** *For every fixed rational E with $1/3 \le E < 1/2$ and $3E^2 + E - 1 > 0$, a rational E-WSNE of every game satisfying (1.1) is computable deterministically in polynomial time. In particular, $E = 217129273/500000000 = 0.434258546$ is valid.*

Proposition 4.6 (Instance-dependent guarantee)**.** *Assume (1.1), and let $1/3 \le E < 1/2$ be rational. Given a rational row-security saddle satisfying (2.3) with $E < a < 1 - E$, if (4.18) holds, the capped-LP algorithm returns a rational E-WSNE in polynomial bit time. Here the running time is polynomial in the combined encoding length of R, C, E and the supplied saddle (p, q_0, a). This statement uses the actual rational security value a; it does not require $3E^2 + E - 1 > 0$.*

Proof. Solve (P) and (D). If $b \le E$, use Lemma 4.1; if $b > E$, use Lemma 4.4. These cases exhaust all optimal values and cover equality at $b = E$. The LP coefficients have polynomial encoding length in the supplied data, and exact rational LP returns optima of polynomial encoding length. The low-branch formulas and at most $m + n$ whole-action deletions use polynomially many rational operations. Peeling uses original weights, so the test sums and final quotients have polynomial encoding length. A fixed action order makes the procedure deterministic. Further bit-complexity details appear in Section 5. □

Proof of Corollary 4.5. First apply Lemma 2.1. If no preliminary branch returns a profile, orient the game so $E < a < U = 1 - E$. Then

$$(1+E)(U-a) < (1+E)(1-2E).$$

The assumed inequality gives

$$E^2 - (1+E)(1-2E) = 3E^2 + E - 1 > 0,$$

so (4.18) holds. Apply Proposition 4.6 and exchange players back if needed. For the displayed rational example, direct integer arithmetic gives

$$3E^2 + E - 1 = \frac{80525587}{2500000000000000000} > 0.$$

The initial $(1, 1)$ check proves the win–lose consequence. □

5 Attaining the algebraic endpoint

5.1 Rational implementation and complete algorithm

To attain ε_* with rational LP coefficients, we use separation of the rational security value a from ε_* to select a slightly smaller rational working tolerance. The preceding section already provides the approximation guarantee whenever the actual-security inequality holds. Here we ensure that inequality using only polynomial-bit rational computations. Preprocessing still uses the target ε_*; only the subsequent capped stage uses the smaller tolerance. Appendix A.5 gives an exact calculation.

Lemma 5.1 (Rational separation)**.** *Let $a = P/Q \in [0, 1]$ be in lowest terms with $Q \geq 1$ and $a > \varepsilon_*$. Then*

$$a - \varepsilon_* \geq \frac{1}{7Q^2}. \tag{5.1}$$

Proof. Let $h(t) = 3t^2 + t - 1$. On $[0, 1]$ it is strictly increasing, $h(\varepsilon_*) = 0$, and $h'(t) = 6t + 1 \leq 7$. The positive rational

$$h(a) = \frac{3P^2 + PQ - Q^2}{Q^2}$$

has an integer numerator at least one. The mean value bound $h(a) - h(\varepsilon_*) \leq 7(a - \varepsilon_*)$ proves (5.1). □

For the next lemma and the endpoint example in Appendix A, write

$$f(t, a) = 2t^2 + at + a - 1.$$

Lemma 5.2 (A rational tolerance below the endpoint)**.** *Suppose $\varepsilon_* < a < 1 - \varepsilon_*$, with $a = P/Q$ in lowest terms and $Q \geq 1$. Set*

$$k = 7 + 2\lceil \log_2 Q \rceil, \qquad T = 2^k, \qquad e = \frac{\lfloor (\mathrm{isqrt}(13T^2) - T)/6 \rfloor}{T}, \tag{5.2}$$

where $\mathrm{isqrt}(N) = \lfloor \sqrt{N} \rfloor$*. Then*

$$\frac{1}{3} < e < \varepsilon_* < \frac{1}{2}, \qquad e < a < 1 - e, \tag{5.3}$$

$$(1 + e)(1 - e - a) < e^2. \tag{5.4}$$

Proof. For any real s and integer T, taking the integer floor before division by six does not change the final floor: $\lfloor (\lfloor s \rfloor - T)/6 \rfloor = \lfloor (s - T)/6 \rfloor$. Indeed, write $\lfloor s \rfloor - T = 6\ell + r$ with $0 \leq r < 6$; adding $s - \lfloor s \rfloor \in [0, 1)$ does not change the quotient's integer part. Since $\sqrt{13T^2} = T\sqrt{13}$, formula (5.2) gives $e = \lfloor T\varepsilon_* \rfloor / T$. Also $T = 128(2^{\lceil \log_2 Q \rceil})^2 \geq 128Q^2$. Since ε_* is irrational,

$$0 < \varepsilon_* - e < \frac{1}{T} \leq \frac{1}{128Q^2} < \frac{1}{64Q^2}. \tag{5.5}$$

Since $3(5/12)^2 + 5/12 - 1 = -1/16 < 0$, we have $\varepsilon_* > 5/12$. Together with $T \geq 128$, (5.5) therefore gives $e > 5/12 - 1/128 > 1/3$. Also $e < \varepsilon_* < a < 1 - \varepsilon_* < 1 - e$.

Because $3\varepsilon_*^2 + \varepsilon_* - 1 = 0$,

$$f(\varepsilon_*, a) = (1 + \varepsilon_*)(a - \varepsilon_*) \geq \frac{1 + \varepsilon_*}{7Q^2}.$$

For $e \leq t \leq \varepsilon_*$, one has $\partial f/\partial t = 4t + a \leq 4\varepsilon_* + a < 1 + 3\varepsilon_* < 3$, using $a < 1 - \varepsilon_*$ and $\varepsilon_* < 1/2$. The mean value bound and (5.5) yield

$$f(e, a) > \frac{1 + \varepsilon_*}{7Q^2} - \frac{3}{64Q^2} > 0.$$

The final positivity follows from $1/7 - 3/64 = 43/448 > 0$. Finally,

$$f(e, a) = e^2 - (1 + e)(1 - e - a),$$

so $f(e, a) > 0$ is precisely (5.4). □

Proof of Theorem 1.1. For a win–lose input first return a pure $(1, 1)$ cell if one exists; otherwise (1.1) holds. For a general input satisfying (1.1), start directly with the security computations. Compute the two rational security saddles and perform Lemma 2.1 at threshold ε_*. Exact comparisons need no algebraic arithmetic in an LP: for a rational $t \in [0, 1]$, the sign of $t - \varepsilon_*$ is the sign of the rational $3t^2 + t - 1$. Comparisons with $1 - \varepsilon_*$ use $1 - t$ instead. Any preliminary output is already a rational ε_*-WSNE.

Otherwise exchange players if needed so that the rational security value $a = P/Q$ satisfies $\varepsilon_* < a < 1 - \varepsilon_*$. Compute the rational e in Lemma 5.2. Use this e as the tolerance in the capped LP and its two branches, keeping the already computed row security saddle. Conditions (5.3) and (5.4) verify Proposition 4.6. It returns a rational e-WSNE, which is an ε_*-WSNE because $e < \varepsilon_*$. Exchange strategies back if the players were swapped. Proposition 5.3 below proves polynomial bit time. □

Complete algorithm. The following lists all decisions in the endpoint implementation. When multiple deletions qualify, choose the first in the fixed order row 1 through row m, followed by column 1 through column n.

1. On a win–lose input, scan for a $(1, 1)$ cell and return it if one exists. In the remaining case require $R, C \geq 0$, $R + C \leq J$.
2. Compute rational security saddles $(p, q_0), a$ for R and $(u, v), \sigma$ for C, as in (2.3)–(2.4).
3. If $a \geq 1 - \varepsilon_*$, return (p, q_0). Otherwise if $\sigma \geq 1 - \varepsilon_*$, return (u, v). Otherwise if both $a, \sigma \leq \varepsilon_*$, return (u, q_0).
4. If $a \leq \varepsilon_* < \sigma$, make the simultaneous assignment

   $$(R, C, p, q_0, a, m, n) \leftarrow (C^\top, R^\top, v, u, \sigma, n, m)$$

   and record that the players were exchanged. Subsequent action indices and the deletion order use these updated dimensions. Now $\varepsilon_* < a < 1 - \varepsilon_*$.
5. Reduce $a = P/Q$ to lowest terms. Compute e by (5.2) and put $U = 1 - e$.
6. Solve the capped primal–dual pair (P)–(D) with $E = e$ and this U.

7. If $b \le e$, form the strategies in (4.2). Otherwise run (4.9)–(4.10) with $E = e$, $c = 0$, and original weights p, q, and form (4.11).

8. If Step 4 exchanged the players, exchange these two strategies back. Return the resulting pair in the original player order.

All maximizations in the proof and every LP deviation constraint range over the original action sets. No intermediate normalization is used during peeling.

5.2 Polynomial bit complexity

Proposition 5.3. *The endpoint algorithm above is deterministic and polynomial in the binary input length, and its output strategies have polynomial binary encoding length.*

Proof. The two zero-sum games have rational LP formulations with a polynomial number of variables and inequalities. Exact rational LP algorithms return primal and dual optima of polynomial encoding length in polynomial time [15]. In particular, the denominator Q of the selected rational security value has polynomially many bits.

Comparisons with ε_* or $1-\varepsilon_*$ use signs of rational quadratic expressions, as described above. All remaining tests are rational comparisons. The integer k in (5.2) is $O(1+\log Q)$. Thus $T = 2^k$ and $13T^2$ have polynomially many bits. Integer square root, rational reduction, and the computation of e are polynomial-time integer operations. The coefficients of the capped LP and its dual therefore also have polynomial encoding length, so exact rational LP remains polynomial at this stage. No LP with irrational coefficients is used. In particular, $\lceil \log_2 Q \rceil$ is computed exactly as the least nonnegative integer ℓ with $2^\ell \ge Q$: it is the binary bit length of $Q-1$, with the bit length of zero defined as zero.

For the low branch, all mixtures and normalizations in (4.2) involve polynomially many rational numbers of polynomial bit length. Their denominators $1+M$ and $a+U$ are positive. For the high branch, at most $m+n$ actions are deleted. Each trigger is a sum of original rational weights times input payoffs or e. Maintaining these sums or recomputing them directly both require polynomially many rational arithmetic operations. A common denominator can be obtained from products of the denominators of the input payoffs, selected LP strategies, and e. There are polynomially many such factors, each of polynomial bit length, so their product has polynomial bit length. The procedure keeps the original weights until final conditioning.

At the final normalization, Lemma 4.2 gives $1-\alpha > 1/(1+e)$ and $1-\beta > 1/(1+e)$. Taking ratios of polynomial-bit rationals with nonzero denominators again yields polynomial-bit rationals. No loop length depends on inverse slack, a smallest positive probability, or a floating-point tolerance. Every exact comparison, construction, and output therefore has polynomial bit complexity. □

Boundary cases.

- The preliminary rules use weak inequalities at the threshold and its complement. For rational input, the rational security values cannot equal the irrational numbers ε_* or $1-\varepsilon_*$; their comparisons are decided by exact quadratic signs.
- In the capped stage $E = e$. Equality $b = e$ takes the low-witness branch. Complementary slackness is valid for every optimal primal–dual pair.
- A coordinate with original probability zero is never a deletion candidate. It remains an admissible original deviation.

- Strict deletion triggers retain ties. The peeling proof permits equality in its budget condition $Dz \le AB$.
- Fractional deletions are used only to prove that a whole-action deletion cannot cross a mass barrier. The algorithm never splits an action or performs infinitesimal steps.
- Arbitrary overlap between $\operatorname{supp}\lambda$ and $\operatorname{supp}\mu$ is permitted. The case $\mu = 0$ uses the same low-branch formula.
- A one-row or one-column game and degenerate LP optimal faces require no separate support-size assumption. The arguments use only nonempty probability supports and the displayed inequalities.

6 Comparison and scope

6.1 Comparison of the constructions

With $U = 1 - E$ and $z = U - a$, the elementary fallback uses $E = E_0$ and a positive cap slack c. Proposition 3.5 gives the sufficient condition

$$4(1 + E)z \le E(E - c).$$

The witness average supplies c, while separate removed-mass estimates produce the factor four. The stronger high-witness branch combines the exact cap $c = 0$ from primal feasibility with chronological joint charging. Lemma 4.4 gives the sufficient condition

$$(1 + E)z \le E^2.$$

The resulting uniform bound is derived in (4.19). Section 5 attains its algebraic endpoint using an input-dependent rational tolerance satisfying the actual-security condition. These sufficient conditions establish no optimality or lower-bound claim for either construction or for arbitrary algorithms.

6.2 Extensions and limitations

Corollary 6.1 (A controlled excess above total payoff one)**.** *Let $R, C \in \mathbb{Q}^{m\times n}$ be normalized and suppose $R + C \le MJ$ for a supplied rational $M \ge 1$. For either $\kappa \in \{E_0, \varepsilon_*\}$, a deterministic algorithm returns rational mixed strategies forming an $(M\kappa)$-WSNE in time polynomial in the encoding length of R, C, M.*

Proof. Set $\widetilde{R} = R/M$ and $\widetilde{C} = C/M$. These rational nonnegative matrices satisfy $\widetilde{R} + \widetilde{C} \le J$. Apply Theorem 1.2 or Theorem 1.1 to obtain a rational κ-WSNE (x, y) of the scaled game. For every pair of original rows r, i and columns s, j,

$$(R_r - R_i)y = M(\widetilde{R}_r - \widetilde{R}_i)y, \qquad x^\top(C_s - C_j) = Mx^\top(\widetilde{C}_s - \widetilde{C}_j).$$

Since $M > 0$, taking maxima over all original deviations multiplies each supported-action gap by M. The strategies and their supports are unchanged. Forming the scaled matrices has polynomial bit cost and yields polynomial encoding length in R, C, M, so the stated running time follows from the cited algorithms. □

In particular, $M < 1/(2\kappa)$ gives a below-1/2 guarantee for this class. The payoff-sum assumption is used both to transfer security and witness bounds between players and to control chronological pair charges. Normalization alone yields $R + C \le 2J$; taking $M = 2$ in the corollary gives $2\kappa > 1/2$ for both stated constants.

For the rounding reduction of Daskalakis, Mehta, and Papadimitriou [7, Lemma 5], define

$$\widehat{R}_{ij} = \mathbf{1}_{\{R_{ij} \ge 1/2\}}, \qquad \widehat{C}_{ij} = \mathbf{1}_{\{C_{ij} \ge 1/2\}}.$$

An e-WSNE of $(\widehat{R}, \widehat{C})$ is a $(1+e)/2$-WSNE of (R, C), using the same strategies. This transfer bound is at least 1/2 for every $e \ge 0$. Applying it with either present win–lose guarantee therefore does not establish a universal below-1/2 guarantee for arbitrary normalized games.

A Worked examples

The first two examples illustrate the elementary algorithm at its fixed tolerance $E = E_0$. The next two use the stronger algorithm with $E = 7/16$ and $U = 9/16$; indeed $3E^2 + E - 1 = 3/256 > 0$. The first four games have rational payoffs satisfying (1.1); the last is a win–lose game illustrating the algebraic endpoint. Where an LP optimum is not unique, we specify valid optimal solutions to describe an admissible execution. The reported *gap pair* is

$$\left(\max_{i \in \mathrm{supp}(x)} D_i(y), \ \max_{j \in \mathrm{supp}(y)} F_j(x) \right),$$

with deviations ranging over all original actions. All calculations are exact. These examples illustrate the algorithms; the universal guarantees follow from the preceding proofs.

A.1 An elementary early stop after admitting a new row

Consider

$$R = \begin{pmatrix} 1/2 & 1 & 0 \\ 1/2 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad C = \begin{pmatrix} 1/2 & 0 & 0 \\ 1/2 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

These matrices satisfy (1.1). Use the row-security saddle

$$p = (1/2, 1/2, 0)^\top, \qquad q_0 = (0, 1/2, 1/2)^\top, \qquad a = 1/2.$$

Direct multiplication gives $p^\top R = (1/2, 1/2, 1/2)$ and $Rq_0 = (1/2, 1/2, 0)^\top$, verifying the certificate. The column-security value is $\sigma = 0$, since row 3 of C is zero. No preliminary branch applies. Table 3 selects $K = 13$, $d = 1/4$, and $w = 71/1000$, and $S_0 = \{1, 2\}$.

The first separation value is $c(S_0) = 1/2$, with unique witness $q_1 = (1, 0, 0)^\top$. Its row-payoff vector is $(1/2, 1/2, 1)^\top$, so exactly row 3 meets the birth threshold h. After adding that row, $c(S) = 0$, and the unique minimizing row distribution is $x = (0, 0, 1)^\top$. The early output is therefore

$$y = (71/1000, 929/2000, 929/2000)^\top.$$

The original payoff vectors are

$$Ry = (1/2, 1/2, 71/1000)^\top, \qquad x^\top C = (0, 0, 0).$$

Thus the gap pair is exactly $(429/1000, 0)$, both entries below E_0. The supported row is newly admitted, while its best deviations are in the original set S_0.

A.2 An elementary fallback with repeated witnesses and a deletion

Consider

$$R = \begin{pmatrix} 0 & 1 \\ 51/100 & 49/100 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 0 \\ 49/100 & 0 \end{pmatrix}.$$

The row-security certificate is

$$p = (1/51, 50/51)^\top, \qquad q_0 = (1/2, 1/2)^\top, \qquad a = 1/2,$$

since $p^\top R = (1/2, 1/2)$ and $Rq_0 = (1/2, 1/2)^\top$. The column-security value is $\sigma = 49/100$, certified by $u = (0, 1)^\top$ and $v = (1, 0)^\top$. No preliminary branch applies, and again the parameters are $K = 13$, $d = 1/4$, and $w = 71/1000$.

Here $S_0 = \{1, 2\}$. At every separation computation the value is $49/100 > E_0$, and the unique witness is $(1, 0)^\top$. Both entries of $R(1, 0)^\top = (0, 51/100)^\top$ are below h, so the birth set stays empty. The algorithm correctly stores thirteen copies of the same witness and then uses $\bar{q} = (1, 0)^\top$. The row gaps are $(51/100, 0)^\top$; the only supported column has gap zero. Since $t_R = 16860517/65000000 < 51/100$, the fallback removes exactly row 1, of original mass $1/51$, and removes no column. It returns $x = (0, 1)^\top$, $y = (1, 0)^\top$. The original payoff vectors are $(0, 51/100)^\top$ and $(49/100, 0)$, respectively. Both supported gaps are zero. This also illustrates why an empty birth set does not terminate witness accumulation early.

A.3 A low witness with two distinct supported-row classes

Consider

$$R = \begin{pmatrix} 0 & 9/10 \\ 1/2 & 2/5 \end{pmatrix}, \qquad C = \begin{pmatrix} 1 & 1/10 \\ 0 & 3/5 \end{pmatrix}.$$

The payoff-sum matrix has rows $(1, 1)$ and $(1/2, 1)$, so (1.1) holds. Direct multiplication verifies the security certificates

$$p = (1/10, 9/10)^\top, \quad q_0 = (1/2, 1/2)^\top, \quad a = 9/20,$$

and $u = (2/5, 3/5)^\top$, $v = (1/3, 2/3)^\top$, $\sigma = 2/5$ for C. Both security equalities hold coordinatewise. Hence no preliminary branch applies: $E < a < U$, while $\sigma < E$.

An optimal capped primal solution and an optimal dual solution are

$$q = (3/8, 5/8)^\top, \quad b = 3/8, \qquad \lambda = (0, 1)^\top, \quad \mu = (2/3, 0)^\top, \quad \nu = 0.$$

Indeed, $Rq = (9/16, 7/16)^\top$ and $Cq = (7/16, 3/8)^\top$, proving primal feasibility. Both dual column inequalities hold with equality, and the dual objective is $(9/16)(2/3) = 3/8 = b$, proving optimality. Since $b < E$, the low branch applies. Here the two supported-row classes are disjoint.

Formula (4.2) gives

$$M = 2/3, \qquad w = 4/9, \qquad x = (2/5, 3/5)^\top, \qquad y = (4/9, 5/9)^\top.$$

The original payoff vectors are

$$Ry = (1/2, 4/9)^\top, \qquad x^\top C = (2/5, 2/5).$$

Thus the gap pair is $(1/18, 0)$, with both entries below E.

For a boundary variant, replace C_{21} by $1/6$. Its column-security saddle is

$$u' = (13/40, 27/40)^\top, \qquad v' = (3/8, 5/8)^\top, \qquad \sigma = E.$$

Indeed, $(u')^\top C = (E, E)$ and $Cv' = (E, E)^\top$. The same capped strategy $q = (3/8, 5/8)^\top$ attains $b = E$. An optimal dual certificate is

$$\lambda = (1/10, 9/10)^\top, \qquad \mu = (1/3, 0)^\top, \qquad \nu = 1/4.$$

Both dual column constraints are equalities and $\nu + U\sum_i \mu_i = E$. Row 1 belongs to both positively supported classes. The low branch gives $x = (13/40, 27/40)^\top$, with the same $y = (4/9, 5/9)^\top$ and gap pair $(1/18, 0)$, since $x^\top C = (E, E)$ and Ry is unchanged. The same primal optimum also admits the dual certificate $\lambda = x$, $\mu = 0$, $\nu = E$, which gives exactly the same output (x, y). Thus the formulas explicitly cover $b = E$, overlapping classes, and $M = 0$.

A.4 A high witness followed by an actual deletion

Keep R from the preceding example and replace C by

$$C = \begin{pmatrix} 1 & 0 \\ 1/2 & 2/5 \end{pmatrix}.$$

Again $R + C \le J$. The same $(p, q_0), a$ is an R-security saddle; for C, the pair $u = (0, 1)^\top$, $v = (1, 0)^\top$ has value $\sigma = 1/2$. Neither value reaches U, and $a > E$, so preprocessing again leaves the capped stage.

The capped optimum is $q = (1, 0)^\top$, $b = 1/2$: it is feasible and achieves this floor on $\operatorname{supp} p$, while the second row of C has maximum entry $1/2$, ruling out a larger floor. Thus $b > E$. The peeling budget holds because

$$z = U - a = 9/80, \qquad (1 + E)z = 207/1280 < 245/1280 = E^2.$$

The row deficits are $r_1 = 9/16$ and $r_2 = 1/16$. Initially the only positive column has deficit $d_1 = 0$. Rule (4.9) therefore deletes row 1, of original probability $1/10$. The other row is retained, and the column's updated trigger is

$$\frac{1}{10}(C_{11} + E) = \frac{23}{160} < E.$$

No further deletion applies. Conditioning gives $x = (0, 1)^\top$, $y = (1, 0)^\top$. The original row payoff vector is $(0, 1/2)^\top$, and the original column payoff vector is $(1/2, 2/5)$. The output is an exact Nash equilibrium. In particular, the deleted row and the zero-probability second column remain among the original deviations: the full gap vectors are

$$(D_1(y), D_2(y)) = (1/2, 0), \qquad (F_1(x), F_2(x)) = (0, 1/10).$$

Their positive entries belong only to unplayed actions.

A.5 The dyadic endpoint tolerance on a win–lose game

Let

$$R = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

The security values are $a = \sigma = 1/2$, and all four security strategies are uniform. Preprocessing at ε_* reaches the capped stage. For $a = P/Q = 1/2$, Lemma 5.2 gives

$$k = 9, \qquad T = 512, \qquad e = 111/256.$$

Exact arithmetic verifies

$$3e^2 + e - 1 = -157/65536 < 0, \qquad f(e, 1/2) = 3041/32768 > 0.$$

Thus this tolerance lies below the universal endpoint, while the actual-security sufficient condition still holds. The capped optimum is uniform, with $b = 1/2 > e$. Here $U = 1 - e = 145/256$, $r_1 = r_2 = 1/2 - e = 17/256 < e$, and both column deficits are zero, so no deletion is needed. The output is the uniform exact equilibrium. This illustrates why the endpoint proof uses the actual value a rather than requiring $3e^2 + e - 1 > 0$.

Use of AI: The author prompted Claude and ChatGPT interactively. The author verified and expanded all the proofs manually.

References

[1] T. Abbott, D. Kane, and P. Valiant. On the complexity of two-player win-lose games. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, pages 113–122, 2005. doi:10.1109/SFCS.2005.59.

[2] Y. Anbalagan, H. Huang, S. Lovett, S. Norin, A. Vetta, and H. Wu. Large supports are required for well-supported Nash equilibria. In *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques (APPROX/RANDOM 2015)*, LIPIcs 40, pages 78–84, 2015. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.78.

[3] Y. Anbalagan, S. Norin, R. Savani, and A. Vetta. Polylogarithmic supports are required for approximate well-supported Nash equilibria below 2/3. In *Web and Internet Economics (WINE 2013)*, LNCS 8289, pages 15–23, 2013. arXiv:1309.7258.

[4] X. Chen, S.-H. Teng, and P. Valiant. The approximation complexity of win-lose games. In *Proceedings of the Eighteenth Annual ACM–SIAM Symposium on Discrete Algorithms (SODA 2007)*, pages 159–168, 2007. https://www.cs.purdue.edu/homes/pvaliant/epsilon-nash.pdf.

[5] A. Czumaj, A. Deligkas, M. Fasoulakis, J. Fearnley, M. Jurdziński, and R. Savani. Distributed methods for computing approximate equilibria. *Algorithmica*, 81:1205–1231, 2019. doi:10.1007/s00453-018-0465-y.

[6] A. Czumaj, M. Fasoulakis, and M. Jurdziński. Approximate well-supported Nash equilibria in symmetric bimatrix games. In *Algorithmic Game Theory (SAGT 2014)*, LNCS 8768, pages 244–254, 2014. arXiv:1407.3004.

[7] C. Daskalakis, A. Mehta, and C. H. Papadimitriou. A note on approximate Nash equilibria. In *Web and Internet Economics (WINE 2006)*, LNCS 4286, pages 297–306, 2006. https://people.csail.mit.edu/costis/papers/camera-81.pdf.

[8] A. Deligkas, M. Fasoulakis, and E. Markakis. A polynomial-time algorithm for 1/2-well-supported Nash equilibria in bimatrix games. *SIAM Journal on Computing*, 52(5):1083–1096, 2023. doi:10.1137/23M1549237. Preprint: arXiv:2207.07007.

[9] M. Fasoulakis and E. Markakis. An improved quasi-polynomial algorithm for approximate well-supported Nash equilibria. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):1926–1932, 2019. doi:10.1609/aaai.v33i01.33011926.

[10] J. Fearnley, P. W. Goldberg, R. Savani, and T. B. Sørensen. Approximate well-supported Nash equilibria below two-thirds. *Algorithmica*, 76(2):297–319, 2016. doi:10.1007/s00453-015-0029-3.

[11] S. C. Kontogiannis and P. G. Spirakis. Well supported approximate equilibria in bimatrix games. *Algorithmica*, 57:653–667, 2010. Published online in 2008. doi:10.1007/s00453-008-9227-6.

[12] H. Li, D. Li, and X. Deng. Discovering expert-level Nash equilibrium algorithms with large language models. *Nature Communications*, 17:7168, 2026. doi:10.1038/s41467-026-74003-1.

[13] R. J. Lipton, E. Markakis, and A. Mehta. Playing large games using simple strategies. In *Proceedings of the Fourth ACM Conference on Electronic Commerce (EC 2003)*, pages 36–41, 2003. doi:10.1145/779928.779933.

[14] Z. Liu. Support contraction for well-supported Nash equilibria. Preprint, 2026. arXiv:2608.00453v1.

[15] A. Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, 1986.