An exact classical threshold in the binary triangle network

Abstract

We determine exactly when the distribution $p_t(a,b,c) = [1 + t(ab + bc + ca)]/8$, with $a, b, c \in \{-1, 1\}$ and $0 \le t \le 1$, can be generated by three independent classical sources in a triangle network. It is realizable if and only if $t \le t_*$, where $t_* = 0.3621418794\ldots$ is the unique root in $(0.36, 0.37)$ of $3t^4 + 28t^3 + 66t^2 - 36t + 3 = 0$. This proves the optimality of the attaining model of Gisin et al. (Nat. Commun., 2020), also used by da Silva, Pozas-Kerstjens, and Parisio (Phys. Rev. A, 2025) in their conjectured classical boundary. More generally, the same threshold bounds the average pair correlation whenever $\mathbb{E}(A + B + C) = \mathbb{E}(ABC) = 0$, without assuming symmetric responses, identical source laws, or finite source spaces. The proof first reduces this constrained optimization to deterministic responses with at most three values per source. Exact rational inequalities then bound every response-table orbit except the attaining one by $9/25 < t_*$. For the remaining orbit, the constrained stationary equations force identical source laws and determine the quartic. Thus a symmetric optimizer is obtained from the unrestricted problem, rather than assumed. The appendix provides the complete exact-arithmetic program that generates and verifies the finite inequalities, without external data or certificate files.

Note. This paper was generated entirely by AI, including MiMo, using an automated research pipeline developed by Chenghua Liu and Hanyu Li.

1 Introduction

Independent sources impose correlation constraints that are absent from the usual Bell-local model with a single shared source. Fritz's correlation-scenario framework [1] makes this distinction explicit: networks can exhibit nonlocality even without measurement inputs. The triangle is the smallest cycle of pairwise sources, with three parties and no common source. Renou et al. [2] exhibited genuine quantum nonlocality in this geometry using entangled states and joint measurements. Characterizing its classical correlations is a prerequisite for deciding what can be explained by independent classical resources alone.

In a classical triangle model, three independent sources X, Y, Z have probability laws μ_X, μ_Y, μ_Z, and the output distribution has the form

$$p(a,b,c) = \int d\mu_X(x)\, d\mu_Y(y)\, d\mu_Z(z)\, P_A(a \mid y,z) P_B(b \mid z,x) P_C(c \mid x,y). \tag{1}$$

Throughout this paper, $a, b, c \in \{-1, 1\}$. The source probability spaces are arbitrary, and the response kernels are measurable probability distributions. They may use independent private randomness. A distribution admitting (1) is called *triangle-local*. Neither the response kernels nor the source laws are assumed to coincide.

Even with binary outputs, the classical set is difficult to describe. A shared random choice between two triangle models introduces a common source and need not produce another triangle model. In particular, symmetry of an observed distribution does not justify averaging its hidden-variable realizations or restricting them to symmetric ones. The inflation technique gives necessary compatibility conditions through consistency relations among copies of latent variables [3]. Its hierarchy is asymptotically complete [4], but this does not by itself yield an exact algebraic threshold at a finite level.

We consider the one-parameter family

$$p_t(a,b,c) = \frac{1+t(ab+bc+ca)}{8}, \qquad 0 \le t \le 1. \tag{2}$$

Its one-body and three-body moments vanish, while each pair moment is t. Gisin et al. [5] constructed a triangle-local model on this line with $t \approx 0.36214$. Their hexagon-based no-signaling constraints imply the necessary bound $t \le \sqrt{2}-1$, which does not exploit the vanishing three-body moment and does not establish optimality of that model. The later study of da Silva, Pozas-Kerstjens, and Parisio [6] uses the same construction within a proposed boundary for symmetric binary distributions. The matching universal bound at this point remained conjectural. We prove it, allowing arbitrary asymmetric classical models.

The result is stronger than a characterization of (2). For an arbitrary binary distribution, define

$$M_1 = \frac{\mathbb{E}A+\mathbb{E}B+\mathbb{E}C}{3}, \qquad M_2 = \frac{\mathbb{E}AB+\mathbb{E}BC+\mathbb{E}CA}{3}, \qquad M_3 = \mathbb{E}ABC. \tag{3}$$

Only two scalar constraints are needed.

Theorem 1 (Sharp aggregate bound). *Every classical triangle model with binary outputs satisfying $M_1 = M_3 = 0$ obeys*

$$M_2 \le t_*, \qquad t_* = 0.36214187940926894258544455999088 7\ldots, \tag{4}$$

where t_* is the unique root in $(0.36, 0.37)$ of

$$3t^4 + 28t^3 + 66t^2 - 36t + 3 = 0. \tag{5}$$

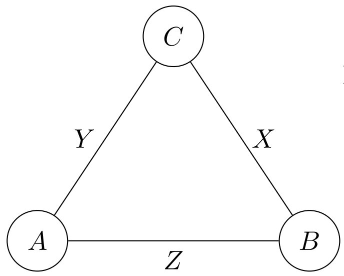
The bound is attained.

Corollary 2. *The distribution (2) is triangle-local if and only if $0 \le t \le t_*$.*

Theorem 1 permits unequal individual means and unequal pair correlations. This aggregate formulation also enables the proof. Rosset, Gisin, and Wolfe [7] established finite hidden-variable cardinality bounds for general networks and the closed semialgebraic nature of their classical correlation sets. We use the same source-wise convexity principle, but preserve only three aggregate moments to obtain compactness. At a constrained maximum, normalization and the two zero-moment constraints then reduce each source to three values. This is a reduction of an optimizing model, not a claim that every binary triangle distribution has such a realization.

The resulting deterministic response tables have 61,872 symmetry orbits. Exact affine bounds on products of source-probability simplexes separate every orbit except that of the attaining table from the optimum: their constrained pair averages are at most 9/25. The remaining orbit is optimized analytically. The finite gap excludes its boundary, and its interior stationary equations force identical source laws. Thus the argument proves the existence of a symmetric optimizer without imposing symmetry on the original optimization. It does not assert uniqueness of unrestricted hidden-variable representations.

Our result concerns an exact classical compatibility boundary, not quantum realizability or the rest of the boundary proposed in [6]. Recent work of Don et al. [8] addresses a different binary family, the noisy W distributions, combining nonlocality certificates with quantum models that reproduce selected distributions to machine precision.



Independent sources X, Y, Z

$$A = A(Y, Z)$$
$$B = B(Z, X)$$
$$C = C(X, Y)$$

Figure 1: The classical triangle. Private randomness can be absorbed into independent sources, making the responses deterministic. The upper bound allows different source laws and different response tables.

2 Attainment of the threshold

We first verify the attaining construction and identify the table used in the upper-bound argument. The model appears in [5, Supplementary Note 2, Eq. (25)], up to exchanging the second and third source values and converting response probabilities into signs. We use the ordering of [6, Appendix B.1].

Let the three sources be independent and identically distributed on $\{1, 2, 3\}$, with probabilities (x, y, z), and let every party use the symmetric response table

$$F = \begin{pmatrix} -1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \qquad A = F_{YZ}, \quad B = F_{ZX}, \quad C = F_{XY}. \tag{6}$$

Choose y to be the root in $(3/8, 2/5)$ of

$$4y^4 - 8y + 3 = 0, \tag{7}$$

and put

$$x = \frac{1 - 2y^2}{4y}, \qquad z = 1 - x - y. \tag{8}$$

The polynomial in (7) is strictly decreasing on $(0, 1/\sqrt{2})$, since its derivative is $16y^3 - 8 < 0$ there. Its values at $3/8$ and $2/5$ have opposite signs, so the stated root exists and is the only root in $(0, 1/\sqrt{2})$. Moreover, $x > 0$, and

$$x + y = \frac{1}{4y} + \frac{y}{2} < \frac{1}{4(3/8)} + \frac{3/8}{2} = \frac{41}{48} < 1,$$

where the strict inequality follows because $1/(4y) + y/2$ decreases on this interval. Thus $x, y, z > 0$. Numerically,

$$(x, y, z) \approx (0.4544225743510722,\ 0.3861128955012451,\ 0.1594645301476828). \tag{9}$$

Expanding the moments of (6) gives

$$\begin{aligned} M_1 &= -1 + 4xy + 2y^2, \\ M_2 &= 1 - 8xy + 4x^2y - 4y^2 + 12xy^2 + 4y^3, \\ M_3 &= -1 + 12xy - 12x^2y + 6y^2 - 12xy^2 - 4y^3. \end{aligned} \tag{10}$$

After substituting (8), these become

$$M_1 = 0, \qquad M_2 = \frac{1}{4y} - 1 + 2y - y^3, \qquad M_3 = 2 - \frac{3}{4y} - y^3. \tag{11}$$

Equation (7) therefore implies

$$M_1 = M_3 = 0, \qquad M_2 = \frac{1}{y} + 2y - 3. \tag{12}$$

The source laws coincide and F is symmetric, so the output distribution is invariant under all party permutations. Its individual means consequently vanish and its pair moments coincide. Expanding

$$p(a,b,c) = \frac{1}{8}\mathbb{E}\big[(1+aA)(1+bB)(1+cC)\big]$$

shows that it is p_t with $t = 1/y + 2y - 3$.

To identify this value, eliminate y from $2y^2 - (t+3)y + 1 = 0$ and (7). Set $u = t+3$. Reduction of the latter polynomial using the quadratic gives

$$(u^3 - 4u - 16)y = u^2 - 8.$$

Multiplying the quadratic by $(u^3 - 4u - 16)^2$ and using this identity yields

$$\begin{aligned} 0 &= 2(u^2-8)^2 - u(u^2-8)(u^3-4u-16) + (u^3-4u-16)^2 \\ &= 2(3u^4 - 8u^3 - 24u^2 + 192) \\ &= 2(3t^4 + 28t^3 + 66t^2 - 36t + 3). \end{aligned} \tag{13}$$

Exact rational evaluations place y in $(0.3861, 0.3862)$, and hence $1/y + 2y - 3$ in $(0.36, 0.37)$. The derivative $12t^3 + 84t^2 + 132t - 36$ is positive throughout this last interval. Thus the attained value is precisely t_* from (5).

Every $t \in [0, t_*]$ is also attainable. Let $R_A, R_B, R_C \in \{-1, 1\}$ be mutually independent private signs, independent of the model, with common mean $\eta = \sqrt{t/t_*}$. Replacing the outputs by $R_A A, R_B B, R_C C$ multiplies the one-body, pair, and three-body moments by η, η^2, η^3, respectively. The resulting moments therefore give exactly p_t. This local postprocessing requires no shared random choice between models.

3 The ternary reduction and finite separation

For the upper bound, we first make the aggregate optimization compact. We then reduce a maximizing model to three values per source and separate its finitely many response-table types. The source probabilities remain continuous variables throughout.

Lemma 3 (Finite barycentres)**.** *Let g be a bounded measurable map from a probability space to $\mathbb{R}^d$. Its mean is a convex combination of at most $d+1$ values of g. The same conclusion holds with the values chosen from any prescribed full-measure subset of the domain.*

Proof. Restrict to the prescribed full-measure subset, if any, and let $m = \int g$. The point m belongs to the closed convex hull K of the range. If m lies on the relative boundary of K, choose a supporting affine functional ℓ that is nonnegative on K, vanishes at m, and is nonconstant on the affine hull of K. Since $\mathbb{E}\ell(g) = 0$, we have $\ell(g) = 0$ almost surely. Restricting the domain to this full-measure set decreases the dimension of the affine hull. After at most d such restrictions, m lies in the relative interior of the closed convex hull of the remaining range.

A point in this relative interior belongs to the convex hull itself. Indeed, in positive affine dimension, choose a small simplex around m inside the closed convex hull. Approximate its vertices by points of the convex hull of the range closely enough that m remains inside the simplex. In affine dimension zero the assertion is immediate. Carathéodory's theorem now expresses m as a convex combination of at most $d+1$ values of g. $\square$

Lemma 4 (Ternary optimizing reduction)**.** *The maximum of M_2 over all classical triangle models satisfying $M_1 = M_3 = 0$ is attained by a deterministic model with at most three values of each source.*

Proof. First make the responses deterministic. Introduce independent uniform random variables U_A, U_B, U_C on $[0,1]$, independent of all sources, and set $A = 1$ exactly when $U_A \leq P_A(1 \mid Y, Z)$, with analogous rules for B, C. Attach U_A to Z, U_B to X, and U_C to Y; the other recipient of each seed ignores it. The enlarged sources remain mutually independent, and each output is now a measurable deterministic function of its two incident sources.

For a fixed deterministic model, define

$$g_X(x) = \int d\mu_Y(y)\, d\mu_Z(z)\, \left(\frac{A+B+C}{3}, \frac{AB+BC+CA}{3}, ABC \right),$$

where the outputs are evaluated at (x,y,z). This is a bounded measurable map whose mean is (M_1, M_2, M_3). Lemma 3 replaces X by a source supported on at most four values without changing these moments. Apply the same operation to Y and then Z, retaining the earlier finite supports. Thus every attainable aggregate triple has a deterministic realization with at most four values per source.

Pad smaller alphabets with zero-probability values. There are finitely many deterministic response tables on three four-valued sources, and for each table the aggregate moments depend continuously on three compact probability simplexes. Their finite union is therefore the entire compact set of attainable aggregate triples. The constraint set $M_1 = M_3 = 0$ is closed and nonempty, so a global maximum of M_2 exists.

Take a deterministic four-valued realization of a global maximum. With the responses and the other two source laws fixed, optimization over one source law is a linear program with three equality constraints: normalization, $M_1 = 0$, and $M_3 = 0$. An extreme point of its feasible polytope has at most three positive coordinates. To see this, a support of size greater than three would admit a nonzero vector in the kernel of the three constraint rows, supported on those coordinates. Both sufficiently small positive and negative perturbations along it would remain feasible, contradicting extremality. Choose an optimal extreme point. Its objective value is still the global maximum. Repeating this step for the other two sources preserves the already reduced supports and yields the claim. $\square$

For three-valued sources, a deterministic response table is a signing of the 27 edges of the complete tripartite graph $K_{3,3,3}$. The three vertex classes correspond to the sources, and the sign of an edge specifies the response to its two endpoint values. Independent source laws assign probability weights to the three vertex classes.

Permuting values within each source, permuting the three parties, and flipping all output signs preserve the constrained optimization. The global sign flip negates M_1, M_3 and leaves M_2 unchanged. The symmetry group has order

$$(3!)^3\, 3!\, 2 = 2592.$$

Its action on the 2^{27} signings has 61,872 orbits, as verified by both enumeration and Burnside's lemma in Appendix A.

Lemma 5 (Certified finite gap)**.** *For every deterministic ternary response-table orbit except the orbit of* (6)*, and every choice of the three independent source probability laws,*

$$M_1 = M_3 = 0 \quad\Longrightarrow\quad M_2 \le \frac{9}{25}. \tag{14}$$

Zero source probabilities are allowed.

Computer-assisted proof. Write the source probability vectors as $q, r, s \in \Delta_2$, where

$$\Delta_2 = \{v \in \mathbb{R}^3_{\ge 0} : v_1 + v_2 + v_3 = 1\}.$$

For a fixed response table, define

$$G(q,r,s) = \big(\mathbb{E}(A+B+C), \mathbb{E}(ABC), \mathbb{E}(AB+BC+CA)\big) = (3M_1, M_3, 3M_2). \tag{15}$$

Independence makes G affine in each source probability vector separately. Let $Q, R, S \subseteq \Delta_2$ be triangles with vertices q_i, r_j, s_k, respectively. If $q = \sum_i \alpha_i q_i$, $r = \sum_j \beta_j r_j$, and $s = \sum_k \gamma_k s_k$ are their barycentric representations, then

$$G(q,r,s) = \sum_{i,j,k=1}^{3} \alpha_i \beta_j \gamma_k G(q_i, r_j, s_k).$$

Hence G throughout the product cell $Q \times R \times S$ lies in the convex hull of its 27 corner values v_ℓ.

A rational pair (λ, μ) satisfying

$$(v_\ell)_3 \le \frac{27}{25} + \lambda (v_\ell)_1 + \mu (v_\ell)_2 \qquad (\ell = 1, \dots, 27) \tag{16}$$

therefore proves (14) on that cell. Starting from Δ_2^3, the program either verifies such a pair or bisects an edge of one source triangle. The two resulting product cells cover the parent, including its boundary. A finite binary tree whose leaves all satisfy (16) proves the bound on the whole domain.

The complete program in Appendix A enumerates the response-table orbits and generates the exceptional orbit directly from (6). For each of the other 61,871 orbits, it constructs and checks a covering tree. All 27 inequalities at every accepted leaf are checked by exact integer arithmetic. An unaccepted cell encountered at a stopping limit causes failure, not acceptance. The full run terminates successfully with the values in Table 1. Consequently every nonexceptional orbit, with all continuous source laws included, satisfies (14). □

The strict inequality $9/25 < t_*$ isolates the attaining orbit. No claim that $9/25$ is the optimal bound for the other orbits is needed.

4 Optimality on the exceptional orbit

It remains to optimize (6) over three possibly different source laws. Write these as

$$(x_i, y_i, z_i), \qquad z_i = 1 - x_i - y_i, \qquad i = 1, 2, 3,$$

Quantity	Verified value
Response-table orbits, including the exceptional orbit	61,872
Orbits certified by (14)	61,871
Nodes in all product-simplex partition trees	5,103,027
Affine leaf witnesses	2,582,449
Maximum bisection depth	27

Table 1: Exhaustive exact verification for Lemma 5. The appendix program generates all inputs and checks all leaf inequalities without floating-point arithmetic.

with nonnegative coordinates, and put $u_i = x_i + y_i$. Define

$$\begin{aligned} E &= \sum_{i<j} (u_i u_j - x_i x_j), \\ T &= y_1 y_2 y_3 + \sum_i x_i y_j y_k, \\ S &= \sum_i y_i x_j x_k, \qquad V = T + S = u_1 u_2 u_3 - x_1 x_2 x_3. \end{aligned} \tag{17}$$

In each sum indexed by i, the indices j, k are the other two elements of $\{1, 2, 3\}$.

These expressions follow from the positive entries of F. The quantity E is the sum of the three probabilities of a positive output, and T is the probability that all three outputs are positive. A source value equal to 3 forces both incident outputs to be negative. If all source values lie in $\{1, 2\}$, at least two values equal to 2 give three positive outputs; exactly one value equal to 2 gives two. Thus the sum of the three probabilities of two specified positive outputs is $3T + S$. Expanding the products of the indicators $(1 + A)/2, (1 + B)/2, (1 + C)/2$ now gives

$$\begin{aligned} M_1 &= \frac{2}{3}E - 1, \\ M_2 &= 1 - \frac{4}{3}E + \frac{4}{3}(3T + S), \\ M_3 &= -1 + 2E - 4V. \end{aligned} \tag{18}$$

In particular,

$$M_1 = M_3 = 0 \iff E = \frac{3}{2}, \qquad V = \frac{1}{2}. \tag{19}$$

We first analyze stationary points in the interior $x_i, y_i, z_i > 0$. The finite gap will then exclude a boundary maximizer.

Lemma 6 (Regularity)**.** *The gradients of M_1 and M_3, with respect to the six coordinates $(x_i, y_i)_{i=1}^3$, are linearly independent whenever $x_i > 0$ and $y_i > 0$ for all i.*

Proof. By (18), it suffices to consider E, V. The gradient of E is nonzero, since $\partial E/\partial y_i = u_j + u_k > 0$. Suppose $\nabla V = \rho \nabla E$. The y_i derivatives give

$$u_j u_k = \rho(u_j + u_k), \qquad i = 1, 2, 3.$$

Thus $\rho > 0$, and subtraction of the equations $1/u_j + 1/u_k = 1/\rho$ gives $u_1 = u_2 = u_3 = u$ and $\rho = u/2$. The x_i derivatives would then imply

$$x_j x_k = \frac{u}{2}(x_j + x_k).$$

But $0 < x_j, x_k < u$, so $1/x_j + 1/x_k > 2/u$, contradicting this equality. □

Lemma 7 (Stationary laws coincide)**.** *Every interior stationary point of M_2, subject to $M_1 = M_3 = 0$, for the table* (6) *satisfies $x_1 = x_2 = x_3$ and $y_1 = y_2 = y_3$.*

Proof. Lemma 6 permits Lagrange multipliers λ, μ. With $L = M_2 - \lambda M_1 - \mu M_3$, (18) gives

$$L = 1 + \lambda + \mu + cE + bT + aS, \tag{20}$$

where

$$a = \frac{4}{3} + 4\mu, \qquad b = 4 + 4\mu, \qquad c = -\frac{4}{3} - \frac{2}{3}\lambda - 2\mu. \tag{21}$$

The six derivative equations are

$$0 = c(y_j + y_k) + by_jy_k + a(x_jy_k + y_jx_k), \tag{22}$$
$$0 = c(x_j + y_j + x_k + y_k) + b(y_jy_k + x_jy_k + y_jx_k) + ax_jx_k. \tag{23}$$

Suppose first that $a = 0$, so $b = 8/3 > 0$. Dividing (22) by y_jy_k gives $c(1/y_j + 1/y_k) = -b$, hence $c \neq 0$ and $y_1 = y_2 = y_3 = y > 0$. Then $c = -by/2$, and (23) reduces to $0 = (by/2)(x_j + x_k) > 0$, a contradiction. Therefore $a \neq 0$.

Dividing (22) by y_jy_k now gives three equations

$$\frac{ax_j + c}{y_j} + \frac{ax_k + c}{y_k} = -b.$$

The three quantities $(ax_i + c)/y_i$ must therefore all equal $-b/2$. Set $h = -c/a$ and $d = b/(2a)$. Then

$$x_i = h - dy_i, \qquad i = 1, 2, 3. \tag{24}$$

Substitution into (23), followed by division by a, yields

$$-h^2 + h(2d - 1)(y_j + y_k) + d(2 - 3d)y_jy_k = 0. \tag{25}$$

If $h = 0$, positivity of x_i, y_i forces $d < 0$, whereas (25) forces $d = 0$ or $d = 2/3$. Consequently $h \neq 0$.

Put $P = h(2d - 1)$ and $Q = d(2 - 3d)$. Subtracting pairs of (25) gives

$$(y_i - y_j)(P + Qy_k) = 0. \tag{26}$$

Assume that the y_i are not all equal. If $Q = 0$, an unequal pair in (26) gives $P = 0$, and then (25) gives $h = 0$, a contradiction. Thus $Q \neq 0$. If all three y_i were distinct, (26) would force all of them to equal $-P/Q$. Therefore exactly two coincide. Relabel so that $y_1 = y_2 = v \neq y_3$. Applying (26) to the pairs $(1, 3)$ and $(2, 3)$ gives $v = -P/Q$. The equation for $(j, k) = (1, 2)$ in (25) then implies

$$0 = h^2Q + P^2 = h^2\left[d(2 - 3d) + (2d - 1)^2\right] = h^2(d - 1)^2. \tag{27}$$

Hence $d = 1$. But (25) now reads $-x_jx_k = 0$, again contradicting positivity. All y_i coincide, and (24) gives the same conclusion for all x_i. □

Proof of Theorem 1. By Lemma 4, a global maximizer has deterministic responses and at most three values per source. Pad smaller alphabets to three values. Section 2 gives a feasible value $t_* > 9/25$, so Lemma 5 places the maximizing response table in the orbit of (6). Apply the corresponding relabelings and, if necessary, the global output flip to put it in that form.

Every source value has positive probability. Otherwise flip one response entry incident to a zero-probability source value. This changes no observed probability. The three copies of F together have nine positive entries; the modified table has eight or ten. Every table in the exceptional orbit has either nine or eighteen positive entries, because its symmetries only permute entries or complement all signs. The modified table is therefore nonexceptional, contradicting Lemma 5 and the value $M_2 > 9/25$.

The maximizing laws are thus interior, and Lemmas 6 and 7 show that all three are $(x, y, 1-x-y)$. Equations (19) become

$$(x+y)^2 - x^2 = \frac{1}{2}, \qquad (x+y)^3 - x^3 = \frac{1}{2}.$$

The first gives (8). Substituting it into the second gives (7). Since $x, y > 0$, we have $0 < y < 1/\sqrt{2}$, where that quartic has exactly one root. The maximum is therefore t_*, and its attainment was proved in Section 2. $\square$

Proof of Corollary 2. For p_t, the aggregate triple is $(M_1, M_2, M_3) = (0, t, 0)$. Theorem 1 gives necessity of $t \leq t_*$, and the construction and local postprocessing in Section 2 give sufficiency. $\square$

References

[1] T. Fritz, *Beyond Bell's theorem: correlation scenarios*, New Journal of Physics **14**, 103001 (2012), doi:10.1088/1367-2630/14/10/103001.

[2] M.-O. Renou, E. Bäumer, S. Boreiri, N. Brunner, N. Gisin, and S. Beigi, *Genuine quantum nonlocality in the triangle network*, Physical Review Letters **123**, 140401 (2019), doi:10.1103/PhysRevLett.123.140401.

[3] E. Wolfe, R. W. Spekkens, and T. Fritz, *The inflation technique for causal inference with latent variables*, Journal of Causal Inference **7**(2), 20170020 (2019), doi:10.1515/jci-2017-0020.

[4] M. Navascués and E. Wolfe, *The inflation technique completely solves the causal compatibility problem*, Journal of Causal Inference **8**(1), 70–91 (2020), doi:10.1515/jci-2018-0008.

[5] N. Gisin, J.-D. Bancal, Y. Cai, P. Remy, A. Tavakoli, E. Zambrini Cruzeiro, S. Popescu, and N. Brunner, *Constraints on nonlocality in networks from no-signaling and independence*, Nature Communications **11**, 2378 (2020), doi:10.1038/s41467-020-16137-4. The attaining model is in Supplementary Note 2, Eq. (25).

[6] J. M. da Silva, A. Pozas-Kerstjens, and F. Parisio, *Local models and Bell inequalities for the minimal triangle network*, Physical Review A **112**, L030403 (2025), doi:10.1103/cg9v-wv2t; arXiv:2503.16654v2. Appendix references in the text refer to this arXiv version.

[7] D. Rosset, N. Gisin, and E. Wolfe, *Universal bound on the cardinality of local hidden variables in networks*, Quantum Information and Computation **18**, 910–926 (2018), arXiv:1709.00707.

[8] E. Don, J. Bavaresco, P. Lipka-Bartosik, N. Gisin, N. Brunner, and A. Pozas-Kerstjens, *The minimal example of quantum network Bell nonlocality*, preprint (2026), arXiv:2605.00981v1.

A Exact verification of the finite gap

This appendix specifies the finite calculation used in Lemma 5 and gives its complete executable implementation. The inputs are the response table (6), the rational bound 9/25, and the enumeration and subdivision rules below. No external orbit list, certificate, data file, or numerical witness is required.

Number source values by $0, 1, 2$. Encode a response table by a 27-bit word, with bit one denoting output -1. The bit positions are

$$C(i,j) : 3i + j, \qquad B(k,i) : 9 + 3i + k, \qquad A(j,k) : 18 + 3j + k.$$

The three copies of (6) give word 127388645, whose least image under all symmetries is 2889227.

For an edge permutation π arising from source-value and party permutations, let $c(\pi)$ be its number of cycles on the 27 edges. It fixes $2^{c(\pi)}$ signings. Combining a permutation with the global sign flip fixes a signing only when every edge cycle has even length; this is impossible on an odd number of edges. The cycle counts for the 1296 edge permutations are

$c(\pi)$	3	5	7	9	11	15	17	21	27
Multiplicity	144	540	36	398	36	105	27	9	1

Burnside's lemma consequently gives

$$\frac{1}{2592}\sum_{\pi} 2^{c(\pi)} = \frac{160372224}{2592} = 61872.$$

The program constructs all 1296 permutations explicitly and checks this sum. It also scans all 2^{27} words in increasing order. On encountering an unmarked word, it marks all its permutation images and their complements. The word is then the least representative of a new orbit. This scan covers every signing and independently checks the orbit count. The exceptional orbit is computed from (6), not read from a supplied list.

For each nonexceptional representative, the initial cell is the product of three full source simplexes. At total bisection depth d, store its corner values as $v_\ell = f_\ell/2^d$, with integral f_ℓ, and set

$$a_\ell = 25(f_\ell)_1, \qquad b_\ell = 25(f_\ell)_2, \qquad c_\ell = 25(f_\ell)_3 - 27\,2^d. \tag{28}$$

Thus (16) is equivalent to the 27 half-plane constraints $a_\ell\lambda + b_\ell\mu \geq c_\ell$.

The routine `bound_cell` constructs a rational point satisfying these constraints, if it finds one, by processing the half-planes in order. When the current point violates a new constraint, the preceding constraints are restricted to the new boundary line. They give lower and upper bounds on one coordinate. The routine chooses zero if it is allowed, and otherwise an interval endpoint. The resulting point is an axis intercept or an intersection of two boundary lines. It accepts a cell only after independently rechecking *all* 27 inequalities by cross multiplication with a positive denominator. The validity of an accepted cell therefore depends on these exact comparisons, not on any completeness claim about the witness search.

An unaccepted cell is subdivided along the longest edge among its three source triangles, measured in Euclidean barycentric coordinates. Ties are resolved in source and vertex order. Replacing one endpoint and then the other by the midpoint gives two triangles whose union is the original triangle. At a replaced endpoint the moment numerators are added, while all other moment numerators are doubled, producing the common denominator 2^{d+1}. Vertex numerators in `V` are updated by the same rule, using a separate denominator for each source. Induction from the root therefore verifies both the stored corner values and coverage of the source-probability domain.

A depth-first traversal processes both children of each subdivision. If a cell cannot be accepted and either two million nodes have been visited for that orbit or its depth is 42, the run fails. These limits never authorize acceptance. In the completed run the maximum depth is 27, and the forest counts obey

$$5{,}103{,}027 = 2 \cdot 2{,}582{,}449 - 61{,}871.$$

Together with the orbit scan and all leaf tests, exhaustion of every subdivision stack establishes Lemma 5.

All arithmetic is integral, with the following bounds ensuring that fixed-width operations are exact. At every permitted depth $d \leq 42$, moment numerators have magnitude at most $3 \cdot 2^d$, and every coefficient in (28) has magnitude at most $102 \cdot 2^d < 2^{49}$. Two-term determinants therefore have magnitude less than 2^{99}, within signed 128-bit range. Comparisons of two determinant ratios use products of magnitude less than 2^{198}; the final plane evaluations are smaller still. For a source at depth d_i, its vertex coordinates have numerators between zero and 2^{d_i}. Squared-distance numerators are less than 2^{86}, and denominator alignment in distance comparisons uses shifts of at most 84 bits. All these comparisons fit in 256-bit integers. The 64-bit stored numerators and plane coefficients also stay within their signed ranges.

Save the following complete listing as `triangle.cpp` and compile it with a C++17 compiler supporting `__int128_t` and the header-only Boost.Multiprecision library:

```
c++ -O3 -std=c++17 triangle.cpp -o triangle
./triangle
```

The program takes no parameters, reads no files, and uses no random seed or floating-point arithmetic. The listing was compiled with GCC 14.2.0 and executed in full. It returned

```
PASS orbits=61872 certified=61871 nodes=5103027
     leaves=2582449 depth=27
```

with the actual output on one line. Progress messages are written to standard error. Successful completion requires every orbit to be covered, every nonexceptional orbit to pass its exact cell tests, and all reported counts to agree with the checks at the end of the program.

```
#include <algorithm>
#include <array>
#include <cstdint>
#include <iostream>
#include <stdexcept>
#include <vector>
#include <boost/multiprecision/cpp_int.hpp>
using I128=__int128_t;
using I256=boost::multiprecision::int256_t;
constexpr int64_t HNUM=27, HDEN=25;
struct Plane {int64_t a,b,c;};
struct Point {I128 x=0,y=0,d=1;};
struct Bound {I128 n=0,d=1;int plane=-1;bool set=false;};
struct Witness {uint8_t kind=0,j=0,k=0;};
static bool lessfrac(const Bound&a,const Bound&b) {
    return I256(a.n)*I256(b.d)<I256(b.n)*I256(a.d);
}
static bool satisfies(const Plane&a,const Point&p) {
    return I256(a.a)*I256(p.x)+I256(a.b)*I256(p.y)>=I256(a.c)*I256(p.d);
}
```

```
static Point intersect(const Plane&a,const Plane&b) {
    Point p;
    p.d=I128(a.a)*b.b-I128(a.b)*b.a;
    p.x=I128(a.c)*b.b-I128(a.b)*b.c;
    p.y=I128(a.a)*b.c-I128(a.c)*b.a;
    if(p.d<0){p.d=-p.d;p.x=-p.x;p.y=-p.y;}
    return p;
}
// Exact feasibility of 27 half planes in two dimensions. A feasible point is
// an exact affine upper certificate for the convex hull of this product cell.
static bool bound_cell(const int64_t F[27][3],int depth,Witness* witness=nullptr) {
    Plane P[27];const int64_t den=int64_t(1)<<depth;
    for(int j=0;j<27;j++) P[j]={HDEN*F[j][0],HDEN*F[j][1],HDEN*F[j][2]-HNUM*den};
    Point p;Witness w;
    for(int j=0;j<27;j++) {
        if(satisfies(P[j],p))continue;
        if(P[j].a==0 && P[j].b==0)return false;
        const bool use_x=P[j].a!=0;
        const int64_t A=use_x?P[j].a:P[j].b;
        const int sign=A>0?1:-1;
        Bound lo,hi;
        for(int k=0;k<j;k++) {
            I128 coefficient, rhs;
            if(use_x){
                coefficient=sign*(I128(P[k].b)*P[j].a-I128(P[k].a)*P[j].b);
                rhs=sign*(I128(P[k].c)*P[j].a-I128(P[k].a)*P[j].c);
            }else{
                coefficient=sign*(I128(P[k].a)*P[j].b-I128(P[k].b)*P[j].a);
                rhs=sign*(I128(P[k].c)*P[j].b-I128(P[k].b)*P[j].c);
            }
            if(coefficient==0){if(rhs>0)return false;continue;}
            Bound q{rhs,coefficient,k,true};
            if(q.d<0){q.d=-q.d;q.n=-q.n;}
            if(coefficient>0){if(!lo.set || lessfrac(lo,q))lo=q;}
            else {if(!hi.set || lessfrac(q,hi))hi=q;}
        }
        if(lo.set && hi.set && lessfrac(hi,lo))return false;
        int active=-1;
        if(lo.set && lo.n>0)active=lo.plane;
        else if(hi.set && hi.n<0)active=hi.plane;
        if(active>=0){p=intersect(P[j],P[active]);
        if(p.d==0)return false;
        w={3,uint8_t(j),uint8_t(active)};}
        else if(use_x){p={P[j].c,0,P[j].a};w={1,uint8_t(j),0};}
        else{p={0,P[j].c,P[j].b};w={2,uint8_t(j),0};}
        if(p.d<0){p.d=-p.d;p.x=-p.x;p.y=-p.y;}
    }
    // Independent final exact check, so a feasibility-algorithm bug cannot
    // create a false pruning decision.
    for(int j=0;j<27;j++) if(!satisfies(P[j],p))return false;
    if(witness)*witness=w;
    return true;
}
struct Cell {
    int64_t F[27][3];
    int64_t V[3][3][3];
    int depths[3]={0,0,0};
    int depth=0;
};
struct Result {int status=0,depth=0;uint64_t nodes=0,leaves=0;};
static int coord(int point,int axis){
    return axis==0?point/9:axis==1?(point/3)%3:point%3;}
static int otherpoint(int point,int axis,int value){
```

```
    return point+(value-coord(point,axis))*(axis==0?9:axis==1?3:1);}
static std::array<int,3> split_choice(const Cell&c){
    I128 best=-1;int bestdepth=0;std::array<int,3> out={0,0,1};
    for(int a=0;a<3;a++)for(int i=0;i<3;i++)for(int j=i+1;j<3;j++){
        I128 d=0;
        for(int k=0;k<3;k++){I128 e=c.V[a][i][k]-c.V[a][j][k];d+=e*e;}
        if(best<0 || (I256(d)<<(2*bestdepth))>
            (I256(best)<<(2*c.depths[a]))){best=d;
            bestdepth=c.depths[a];
            out={a,i,j};}
    }
    return out;
}
static Cell child(const Cell&c,int axis,int replaced,int other){
    Cell d=c;d.depth++;d.depths[axis]++;
    for(int p=0;p<27;p++)for(int k=0;k<3;k++)
        d.F[p][k]=coord(p,axis)==replaced?
            c.F[p][k]+c.F[otherpoint(p,axis,other)][k]:
            2*c.F[p][k];
    for(int i=0;i<3;i++)for(int k=0;k<3;k++)
        d.V[axis][i][k]=i==replaced?
            c.V[axis][i][k]+c.V[axis][other][k]:
            2*c.V[axis][i][k];
    return d;
}
static Cell root(uint32_t pattern){
    Cell c{};
    for(int a=0;a<3;a++)for(int i=0;i<3;i++)c.V[a][i][i]=1;
    for(int i=0;i<3;i++)for(int j=0;j<3;j++)for(int k=0;k<3;k++){
        const int z=1-2*int((pattern>>(i*3+j))&1u);
        const int y=1-2*int((pattern>>(9+i*3+k))&1u);
        const int x=1-2*int((pattern>>(18+j*3+k))&1u);
        int p=i*9+j*3+k;
        c.F[p][0]=x+y+z;c.F[p][1]=x*y*z;c.F[p][2]=x*y+y*z+z*x;
    }
    return c;
}
static Result verify_pattern(uint32_t pattern,uint64_t limit,int maxdepth){
    std::vector<Cell> todo;todo.push_back(root(pattern));Result r;
    while(!todo.empty()){
        Cell c=todo.back();todo.pop_back();
        r.nodes++;
        r.depth=std::max(r.depth,c.depth);
        Witness w;
        if(bound_cell(c.F,c.depth,&w)){
            r.leaves++;
            continue;
        }
        if(r.nodes>=limit){r.status=1;return r;}
        if(c.depth>=maxdepth){r.status=2;return r;}
        auto s=split_choice(c);
        // Prefix order: the child replacing the lower-numbered vertex first.
        todo.push_back(child(c,s[0],s[2],s[1]));
        todo.push_back(child(c,s[0],s[1],s[2]));
    }
    return r;
}

using Perm=std::array<int,3>;
using Lookup=std::array<std::array<uint32_t,256>,4>;
constexpr uint32_t N=uint32_t(1)<<27, MASK=N-1;
void require(bool ok) {
    if(!ok) throw std::runtime_error("verification failed");
```

```
}
int edge(int a,int i,int b,int j) {
    if(a>b){std::swap(a,b);std::swap(i,j);}
    int block=(a==0 ? (b==1 ? 0 : 1) : 2);
    return 9*block+3*i+j;
}
std::vector<Lookup> symmetries() {
    std::vector<Perm> ps;
    Perm p={0,1,2};
    do {ps.push_back(p);} while(std::next_permutation(p.begin(),p.end()));
    std::vector<Lookup> out;
    uint64_t fixed=0;
    for(auto parts:ps) for(auto p0:ps)
    for(auto p1:ps) for(auto p2:ps) {
        Perm m[3]={p0,p1,p2};
        std::array<int,27> image;
        for(int a=0;a<3;a++) for(int b=a+1;b<3;b++)
        for(int i=0;i<3;i++) for(int j=0;j<3;j++)
            image[edge(a,i,b,j)]=edge(parts[a],m[a][i],parts[b],m[b][j]);
        bool visited[27]={}; int cycles=0;
        for(int i=0;i<27;i++) if(!visited[i]) {
            cycles++;
            for(int j=i;!visited[j];j=image[j]) visited[j]=true;
        }
        fixed+=uint64_t(1)<<cycles;
        Lookup L{};
        for(int byte=0;byte<4;byte++) for(int v=0;v<256;v++)
        for(int bit=0;bit<8 && 8*byte+bit<27;bit++)
            if((v>>bit)&1) L[byte][v]|=uint32_t(1)<<image[8*byte+bit];
        out.push_back(L);
    }
    require(out.size()==1296 && fixed==160372224);
    return out;
}
uint32_t transform(uint32_t v,const Lookup& L) {
    return L[0][v&255]|L[1][(v>>8)&255]|
           L[2][(v>>16)&255]|L[3][v>>24];
}
int main() {
    auto group=symmetries();
    uint32_t reference=0;
    for(int block=0;block<3;block++)
    for(int i=0;i<3;i++) for(int j=0;j<3;j++) {
        bool positive=(i==0 && j==1)||(i==1 && j==0)||(i==1 && j==1);
        if(!positive) reference|=uint32_t(1)<<(9*block+3*i+j);
    }
    uint32_t exceptional=MASK;
    for(const auto& L:group) {
        uint32_t w=transform(reference,L);
        exceptional=std::min(exceptional,std::min(w,MASK^w));
    }
    require(reference==127388645 && exceptional==2889227);
    std::vector<uint64_t> seen(N/64,0);
    uint64_t orbits=0,certified=0,nodes=0,leaves=0;
    int depth=0;
    for(uint32_t v=0;v<N;v++) {
        if((seen[v/64]>>(v%64))&1) continue;
        orbits++;
        for(const auto& L:group) {
            uint32_t w=transform(v,L), z=MASK^w;
            seen[w/64]|=uint64_t(1)<<(w%64);
            seen[z/64]|=uint64_t(1)<<(z%64);
        }
```

```
        if(v==exceptional) continue;
        Result r=verify_pattern(v,2000000,42);
        require(r.status==0);
        certified++; nodes+=r.nodes; leaves+=r.leaves;
        depth=std::max(depth,r.depth);
        if(certified%5000==0) std::cerr<<certified<<" cases checked\n";
    }
    require(orbits==61872 && certified==61871);
    require(nodes==5103027 && leaves==2582449 && depth==27);
    require(nodes==2*leaves-certified);
    std::cout<<"PASS orbits="<<orbits<<" certified="<<certified
             <<" nodes="<<nodes<<" leaves="<<leaves
             <<" depth="<<depth<<"\n";
}
```