Does quantum mechanics use unfair sampling?

Pierre Leroy

Abstract

This paper explores the hypothesis that the predictions of quantum mechanics could apply to a subset of reality. Assuming the existence of intrinsically undetectable states, a computer simulation shows that a simple deterministic method can accurately produce theoretical EPR probabilities.

1. Introduction

The various attempts at ontological interpretations of quantum mechanics remain problematic to this day. They attempt to reconcile the non-local appearance of the theory with a logically consistent explanation.This results in difficult-to-conceive hypotheses, such as superdeterminism, retrocausality, and multiple universes.

Among the hypotheses attempting to explain the non-local appearance of EPR experimental results, the one associated with the detection loophole was assumed from the start because it is ontologically the simplest.
However, for this loophole to have an explanatory effect on non-local experimental results, it implies a non-random selection mechanism of possible measurements.

Such a mechanism would then cause a certain number of photons to disappear from the measurements under specific conditions, inevitably influencing the detection rate.

However, such a mechanism is not observed experimentally, and with the high-efficiency detectors used in recent experiments, even if this mechanism were active, the quantity of unmeasured photons would not allow us to explain the values of violation of Bell's inequalities.

It then appears that the detection loophole is definitively closed.

This is true, but within a restricted framework.
It assumes that detectors can detect all real particles, or in other words, that there are no intrinsically undetectable states (similar to 'dark matter'). This implies that unfair sampling will cause the detection rate to vary under certain conditions.

The local method simulated in this article, assuming the existence of intrinsically undetectable states, allows us to escape from this framework.

2. Description and properties of the local method

It is used to simulate a conventional EPR experiment in order to define the probabilities of coincidence-type measurements.
It must perform the following operations:

- Identify the detectable and undetectable states of photons.
  This state is defined by a local variable at the emission of an EPR pair.

- Define a rule that toggles a state from detectable to undetectable.
  Indeed, if the undetectable states were permanent, as if belonging to a separate universe, they would have no influence on the experimental results.
  In the local model this state can be reversed at the polarizers.
- Maintain a constant detection rate.
  The number of photons in a detectable and undetectable state must remain balanced, not producing fluctuations in the detection rate of the measurable part.
- Define the specific properties of entangled pair emission.

2.1 Description of the local model

For computer simulation the following local model is used.

<u>Local variables of the photon:</u>

The photon simulation uses three local variables denoted p,q,e with:

- p: polarization angle $[-\pi..\pi]$
- q: stochastic variable [-1..1] (described further)
- e: 0: undetectable 'dark' state, 1: detectable 'bright' state.

2.2 Polarizer simulation

It produces two effects which are the selection of an output O or E for the photon, and the possible inversion of the detectability state e of the photon.

<u>Selecting the O or E output of the polarizer</u>:

It is determined by the following function which evaluates a value w as a function of 3 parameters (β, p, q) with β the angle of the polarizer.

$w = \cos(2*(\beta - p) + \text{asin}(q))$

- if $w \geq 0$, output O is selected
- if $w < 0$, output E is selected

The polarization of the photon is then aligned with the polarization corresponding to the selected O or E output.

This method produces exactly Malus's law.

<u>Inversion of the photon e state</u>:

It depends on the difference in polarization of the photon between its entry and its exit from the polarizer and on a constant noted Ta.

Noting rp this difference with :

rp = (outgoing polarization) - (incoming polarization) $[-\pi/2 .. \pi/2]$

The inversion of e is done according to the rp value with the following test:

If |rp| > $\pi/4$ + Ta, the value of e is inverted ($0 \leftrightarrow 1$).
The constant Ta shifts the test threshold.

2.3 Source simulation

It defines the local variables values for the two photons in a pair.

The polarizations are defined based on the type of simulated source: |HH⟩+|VV⟩ or |HV⟩+|VH⟩.
The mixing rate of the first state with the second is adjustable for non-entangled states and is defined from the entanglement coefficient r for entangled states.

<u>For a pair of unentangled AB photons</u>:

The variables q and e of the two photons are initialized with independent random variables.

Ae = random 0/1
Aq = random [-1..1]
Be = random 0/1
Bq = random [-1..1]

<u>For a pair of entangled AB photons</u>:

It is assumed that entanglement is produced by interference between the two photons at emission which produces the following effects:

- The detectability e states are defined identically for both photons.
- The polarization of the two photons is randomized with a common value r_pol [-PI/4..PI/4]
- The state q and the randomization direction of p are defined as identical or reversed depending on the type of parametric conversion simulated.

Ap = Ap + r_pol
Aq = random [-1..1]
Ae = random 0/1
Be = Ae

For an entangled state |HH⟩+|VV⟩
Bp = Bp + r_pol
Bq = Aq
For an entangled state |HV⟩+|VH⟩
Bp = Bp - r_pol
Bq = -Aq

The interference probability is parameterized as a function of the classical entanglement level r [0..1].

2.4 Simulation of detection

It is simulated that only one of the states e is detectable. (0 or 1)
An undetectable state does not trigger a measurement event, regardless of the detector's efficiency.
The measurable state can be chosen to be 0 or 1, this does not change the measurement results.
State 1 is chosen in the simulations.

3.0 EPR simulation and comparison with quantum mechanical predictions

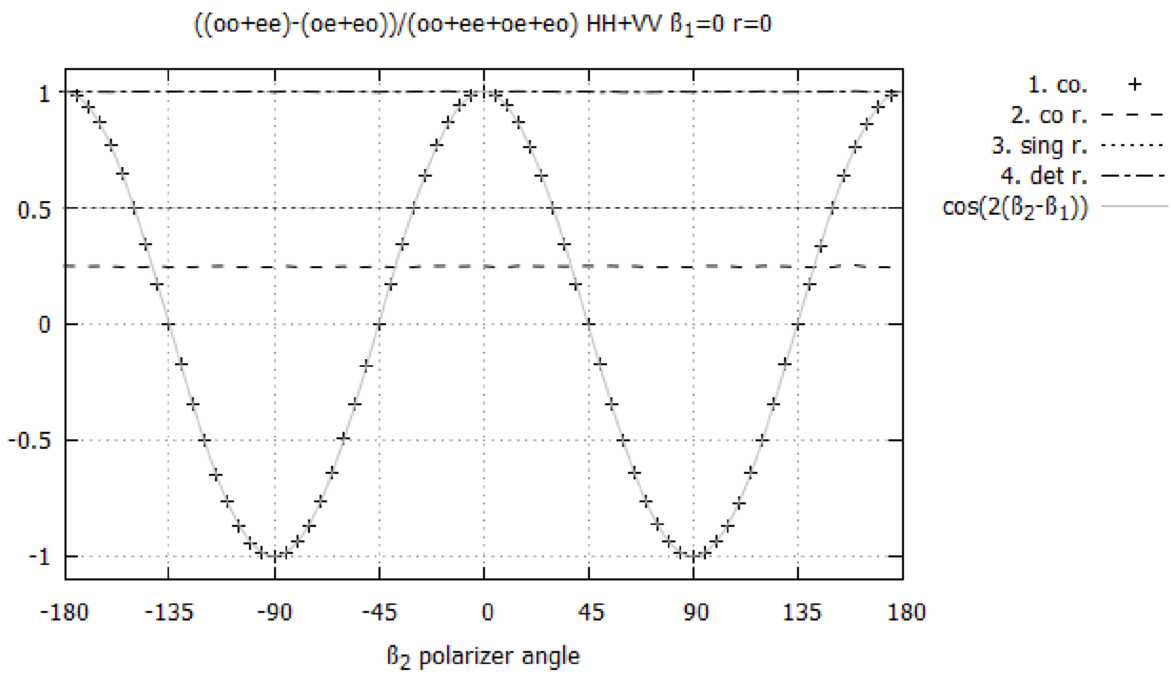
The computer simulation aims to verify that the unfair sampling produced by the undetectable states has no influence on the detection rate of the total number of photons, and that it allows the production of the theoretical detection probabilities of EPR coincidences.

3.1 Non-entangled states

A unentangled state is simulated.
The source randomly generates pairs of polarized photons $|HH\rangle$ or $|VV\rangle$ with the same probability.
By denoting the state of e associated with each photon as 0 or 1, this defines 8 different types of pairs:

H0H0 H1H1 V0V0 V1V1 H0H1 H1H0 V0V1 V1V0

The possible measurements are, by detecting state 1, the pairs H1H1 and V1V1 and the single measurements H1 and V1. The detected states are not necessarily those emitted, because the state e 0/1 can change depending on the angles of the polarizers.



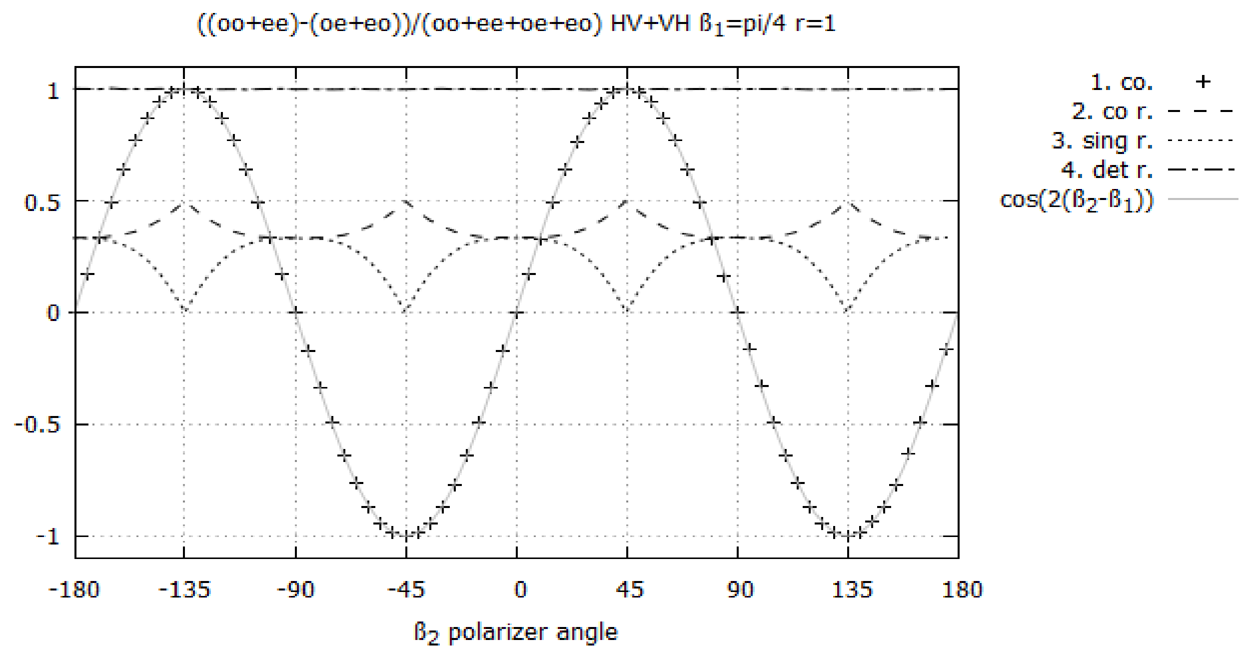
Graph 1.a: Coincidences of measurements of non-entangled states. β1=0 is constant, β2 varies [-π/π].

- Curve 1: Detection coincidences ((oo+ee)-(oe+eo))/(oo+ee+oe+eo).
  The angle β1 is aligned with the source, resulting in a visibility of 1.
- Curve 2: The coincidence measurement rate is 0.25 relative to the real number of pairs emitted. Among 8 possible combinations, the measurable pairs are: H1H1 V1V1, or 1/4 of the emitted pairs.
- Curve 3: The single measurement rate is 0.5. It is generated by pairs in which only one of the two photons is measurable: H0H1 H1H0 V0V1 V1V0
  The H0H0 and V0V0 pairs produce no measurements (uu)
- Curve 4: The apparent total photon detection rate:
  This corresponds to the number of photons measurable directly at the source (1/2 of the real rate), relative to the number of photons measurable after the polarizers, which remains 1/2 of the real rate even if initially measurable photons have become non-measurable, and vice versa. This rate remains constant at 1, indicating that the possible inversion effect of the e states does not affect the photon detection rate.

With non-entangled states, unfair sampling has no detectable effect on the measurements.

3.2 Entangled states

A maximal entangled state |Ψin⟩ = (1/√2)(|HV⟩+|VH⟩) is simulated
The measurement is made in the diagonal HV basis (β1 = π/4) allowing us to verify that the visibility remains at 1.



Graph 1.b: Coincidences of measurements of entangled states. β1= π/4 is constant, β2 varies [-π/π].

<u>Pair measurement rate:</u>

We now observe that the pair measurement rate is no longer constant and varies between 1/3 and 1/2 of the real emitted rate. However, the total photon detection rate (curve 4) remains constant because the single measurement and pair measurement rates vary in opposition (curves 2 and 3).

Single measurement rate:

The model necessarily generates single measurements regardless of detector efficiency unless the polarizers are aligned or offset by $\pi/2$.
This single measurement rate is equal to twice the pair measurement reduction rate, which reduces by a maximum of 1/3 for an angle offset of $\pi/4$ between the polarizers.

Non-measurement rate uu:

It is produced by pairs in which neither photon is detectable.
This rate is exactly identical to the pair measurement rate and corresponds to the pairs that would be detectable if the detected state were reversed. (0 instead of 1 in the simulation)

3.3 Study of the probabilities of coincidences

The following graphs compare the coincidence probability measured with the local model and the theoretical ones predicted by quantum mechanics.

Two types of parametric sources are simulated: type 1 and type 2.

These probabilities predicted by quantum mechanics for the different types of detection coincidences OO OE EO EE are as follows:

SPDC type 1: $|\psi r\rangle = (|HH\rangle + r|VV\rangle) / \sqrt{(1+r^2)}$

$$POO = [\ \cos\alpha\cos\beta + r\sin\alpha\sin\beta\ ]^2 / (1+r^2)$$
$$POE = [-\cos\alpha\sin\beta + r\sin\alpha\cos\beta\ ]^2 / (1+r^2)$$
$$PEO = [-\sin\alpha\cos\beta + r\cos\alpha\sin\beta\ ]^2 / (1+r^2)$$
$$PEE = [\ \sin\alpha\sin\beta + r\cos\alpha\cos\beta\ ]^2 / (1+r^2).$$

SPDC type 2: $|\psi r\rangle = (|HV\rangle + r|VH\rangle) / \sqrt{(1+r^2)}$

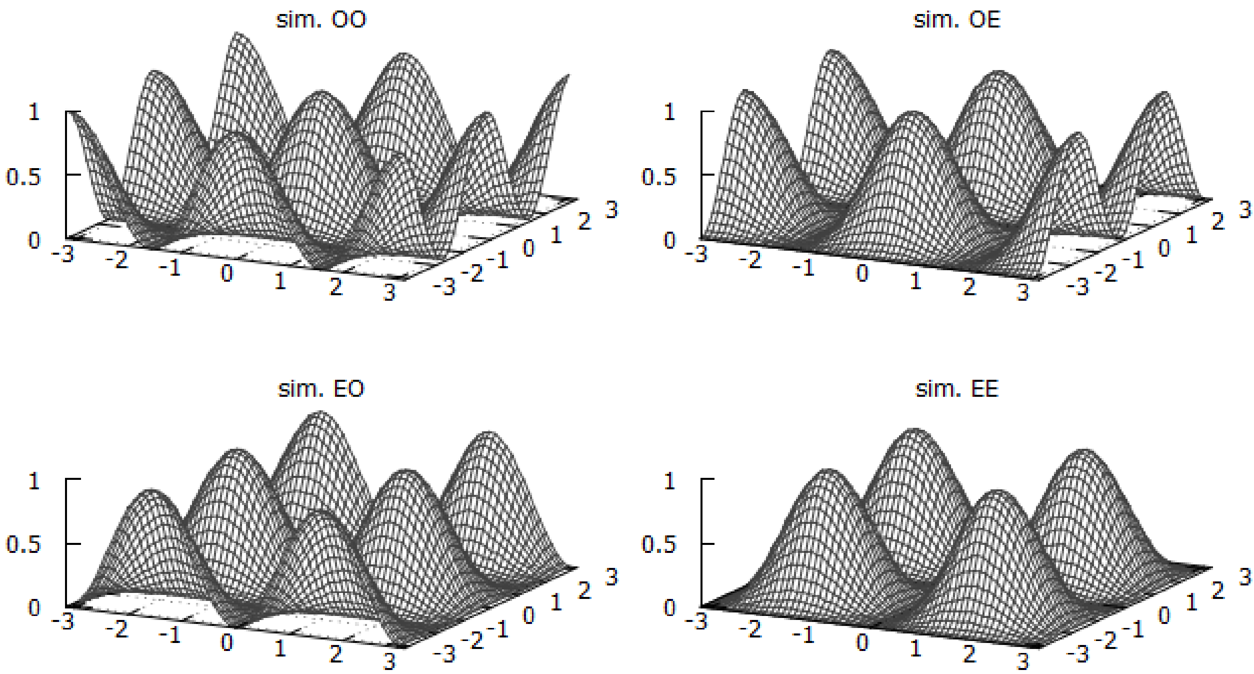
$$POO = [\ \cos\alpha\sin\beta + r\sin\alpha\cos\beta\ ]^2 / (1+r^2)$$
$$POE = [\ \cos\alpha\cos\beta - r\sin\alpha\sin\beta\ ]^2 / (1+r^2)$$
$$PEO = [-\sin\alpha\sin\beta + r\cos\alpha\cos\beta\ ]^2 / (1+r^2)$$
$$PEE = [\ \sin\alpha\cos\beta + r\cos\alpha\sin\beta\ ]^2 / (1+r^2)$$

With α and β being the angle between the polarizers, and r being the entanglement coefficient between 0 and 1.

The 3D graphs display the measured coincidence rates as a function of the polarizer angles α and β varying between $-\pi$ and π.

Coincidences r = 0

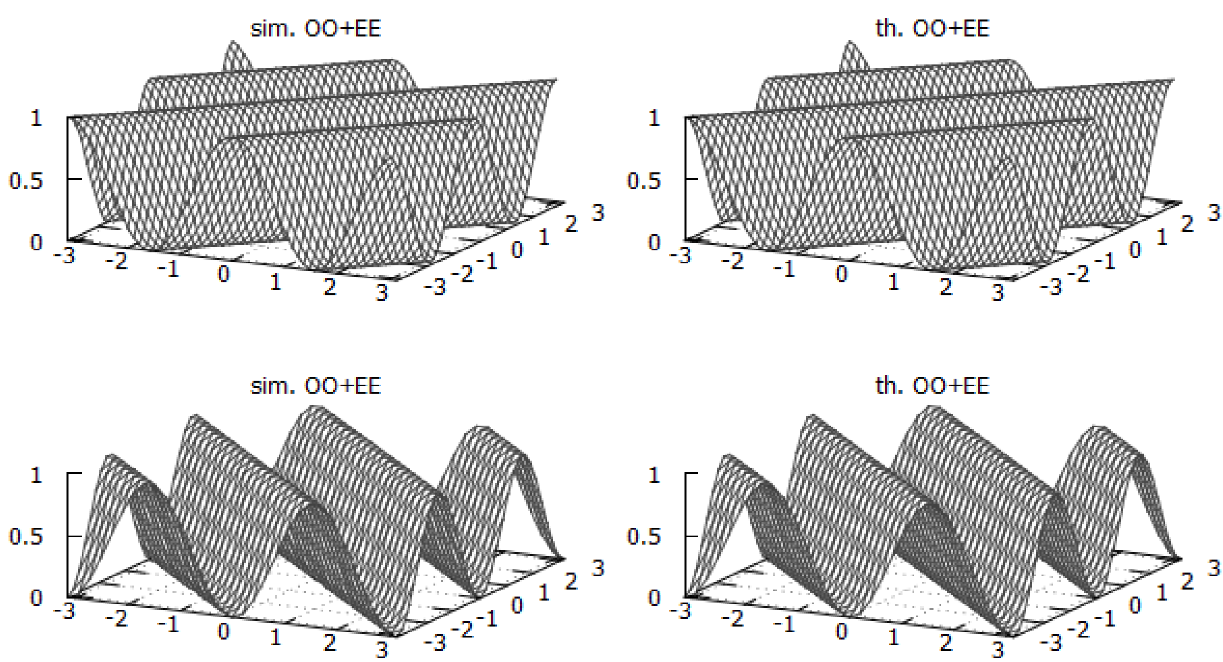
A type 1 SPDC with r = 0 is simulated, which produces an unentangled state composed only of $|HH\rangle$ pairs and allows classical coincidences to be verified.



Graph 2a. Non-entangled state |HH〉

<u>Coincidences r = 1</u>

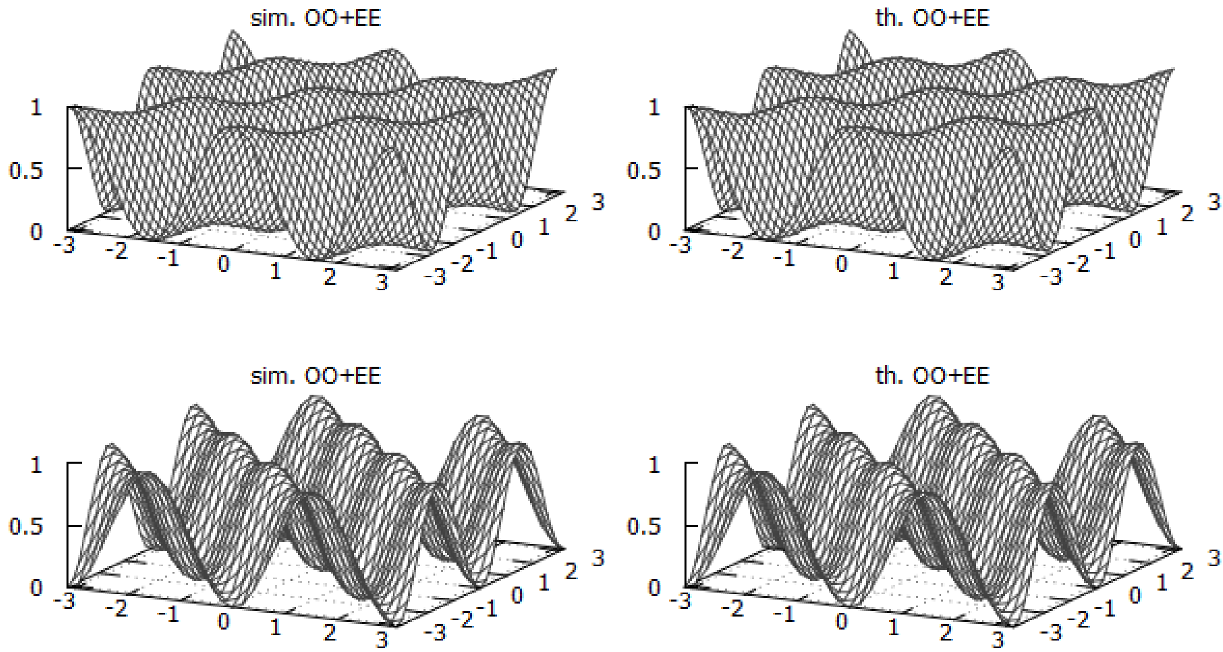
The maximal entanglement states are simulated for a type 1 and 2 SPDC (graph top/bottom).
The graphs on the left are produced by the simulation, those on the right with theoretical equations.



Graph 2b. Maximum entanglement r = 1

Coincidences r = 0.5

Partial entanglement states are also simulated for both types of SPDC.



Graph 2c. Example with partial entanglement r = 0.5

We see from these graphs that the local model accurately produces the quantum correlations for both fully entangled states as well as partially entangled ones..

4. Value of CHSH and Eberhard's inequality

CHSH.

For unentangled states, the CHSH value converges to $\sqrt{2}$ as a function of the number of simulated measurements.
For maximally entangled states, it converges to $2\sqrt{2}$.

For partially entangled states, the simulation results are also consistent with the theoretical values, reflecting the fact that the coincidence probability amplitudes are the same, as can be seen from the graphs.

Eberhard's inequality.

The local model does not allow for a violation of Eberhard's inequality.
It can only approach the violation limit value with J tending towards 0.
If is simulated the non-detection of single measurements, a violation occurs as soon as the simulated entanglement level is non-zero ($r > 0$).

Discussion:

Does the fact that no violation is possible with inequalities that take into account single measurements imply that the described local model is invalid?

It seems possible, or even necessary, that the violation of these inequalities has a different origin than that producing the violation of CHSH.

The very clear violation of CHSH, observable experimentally very close to the theoretical maximum value of $2\sqrt{2}$ [1], results solely from the amplitude of the correlations produced by the entangled states. For these amplitudes to be possible, it seems necessary for a non-random process to come into play at the polarizers. If this process is non-local, it conflicts with special relativity because it requires an exchange of information.
It then seems necessary that the violation of CHSH results from a local process.

Violation of Eberhard's inequality is much more difficult to produce experimentally, and the violations obtained are very weak. [2]. It is not directly related to the amplitude of the correlations but results from any alteration of the measurements on one arm that is dependent on the setting of the other arm, or in other words, if the experiment is sensitive to specific polarizer alignment configurations.

This can then be produced by an influence having only a random effect on the measurements.
Such an effect, not requiring an exchange of information, can remain compatible with relativity.
It can be simulated on the local model by randomly acting on the detectability state e of the photons as a function of the angle configurations between the polarizers.

5. Local model ontology

This section describes the physical assumptions made for the algorithmic development of the local model.

Polarizer:

The first simulation step involves finding a rule for directing a photon toward the polarizer's O or E output.

The model uses the classical Malus law, replacing $\cos^2(\theta)$ with the equivalent $\cos(2\theta)$ to obtain the sign of the cosine, which will then determine the selected output.
However, in this form, with a result dependent only on θ, representing the angle difference between the photon's polarization and the polarizer's angle, the generated transmission law operates using periodic thresholds of $\pi/2$ and generates a rectangular transmission law for the polarizer's O or E outputs.

To obtain a continuous law, it is necessary to add a stochastic variation to θ, modulating the choice of the selected output.

To obtain exactly the cosine Malus law, it is mathematically necessary that the random distribution of values added to θ be produced by the function asin(r), with r a random value uniformly distributed between -1 and 1.

What does this distribution generated by asin(r) mean?
It 'encodes' in the angle θ a stochastic value defined from a r value uniformly distributed.

In the local model, r is replaced by the local variable q of the photon.
The assumption is that the value q then represents a linear spatial position of the particle.
The value asin(q) would then encode the position of the particle in the phase of a local wave.

This interpretation makes sense in a pilot wave model.
Given that asin(q) varies between –π/2 and π/2, the position in the phase could be that on a wavefront.
The non-uniform distribution produced by the asin(q) function would then seem to indicate that the particles tend to be positioned in the middle of the wavefronts.

These ideas are similar to those of Louis de Broglie's initial pilot-wave model, in which the pilot wave would be the electromagnetic potential, considered as a real entity.
Since the local model presented here does not require nonlocality to produce the correlations of quantum mechanics, it does not require the nonlocal quantum potential field proposed by David Bohm.

Two sets of particles:

The selected output of the polarizer then depends on two parameters, θ and asin(q), that is, on a polarization defined by a wave and a particle position in this wave.

It is then assumed that there is a possible tension between these two decision elements, and that the choice of output will depend on a variational principle of least action.

When the choice of an output simultaneously minimizes the necessary action on the wave (polarization change) and that on the particle (position or trajectory change), the particle's detectability state remains unchanged.

In the opposite case, it is assumed that reversing the detectability state can minimizes the total action.
However, it must be considered that reversing this state also requires an action that must be taken into account in the total calculation.

In the simulation method, the action threshold triggering an inversion of the e state corresponds to the action required to change the photon's polarization by an amplitude of π/4 + the action value to invert the e state, defined by the constant Ta (Toggle action).

This mechanism automatically generates two sets of particles of balanced size with an inverted e detectability state.

Entanglement:

It is assumed that this is produced at the source by an interference between the two photons.

One effect of this interference is to define an identical detectability state for both photons in a pair.
The consequence will be that the pair will either be detectable or produce no measurement (uu), thus limiting the possibility of coincidence measurements to only pairs emitted in a detectable state where neither photon has changed state, and those initially emitted in an undetectable state but where both photons have changed state.

Since inversions of e states are done at the polarizers and are angles dependent, this results in unfair sampling of possible coincidence measurements.

The second assumed effect of the interference is to partially randomize the polarization of the two photons in the pair, with a common value between $-\pi/4$ and $\pi/4$.

In the framework of a pilot wave, it is assumed that the particle is not passively guided by the wave but that it interacts with it, as in hydrodynamic models.
The proximity of the two particles can then partially alter the local pilot wave, which would be the electromagnetic potential defining the polarization.
This alteration would also have an effect on the relative position of the photons in the pair, producing a supposed bunching or anti-bunching effect depending on the type of SPDC.

The partial randomization of the polarization produces the superposition effect of quantum mechanics.
Although the photons in the pair remain globally oriented in their H or V emission basis, their real polarizations are altered, and the outputs taken from the polarizers will no longer depend strictly on the emission polarizations but on the altered values of the local variables p and q.

6. Experimental evaluation of the model

With entangled states, even if the photon detection rate remains constant, variations in the pair and single measurement rates are observed depending on the angle difference between the polarizers.
If undetectable states are real, with a maximally entangled state, the variation should be 1/3 between the maximum and minimum coincidence measurement rates, produced by polarizers aligned and offset by an angle of $\pi/4$.

Without undetectable states, this variation should theoretically be zero, which should be observed experimentally if the EPR experiments used four detectors.
However, since most experiments use two detectors, this variation is at least 1/2 due to undetected photons.

Nevertheless, a difference should still be observable with two detectors using maximal entanglement.
Without undetectable states, the pair measurement rate should reduce by a factor of $(1 – 1/2) = 0.5$
With undetectable states, the reduction should be $(1 - (1/2)*(2/3)) = 2/3 \approx 0.66$
There should be a difference of ≈ 16.7% observable.

Conclusion

The presented local model shows that with the assumption of intrinsically undetectable states, the correlations predicted by quantum mechanics can be obtained with an algorithmically simple model based on a few elementary interactions.

Assuming that the predictions of quantum mechanics apply only to a measurable subset of reality could simply explain its nonlocal appearance.
This would not change the theory, only its interpretation, and would perhaps also allow it to be extended to a non-visible part.

Considering undetectable states shifts the ontological difficulty of nonlocality, seemingly logically insoluble, to a problem that can be addressed with local realism.

Références

[1] Detection-Loophole-Free Test of Quantum Nonlocality, and Applications.
arXiv:1306.5772v2 26 Sep 2013
[2] Significant-Loophole-Free Test of Bell's Theorem with Entangled Photons.
arXiv:1511.03190v2 20 Dec 2015

Code source

The following C code simulates the local model, plots the graphs shown in this document, and evaluates CHSH values.
A version can be downloaded directly from this link: [qm_co_2e.c]

```
// EPR correlations simulation using dark states
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PI 3.14159265358979323846
#define DEG_TO_RAD(d) (((d)*PI)/180.0)

// ----------------------------------------------
// RNG

// RNG seed
static unsigned long r_seed = -123;
#define UPD_SEED (r_seed = (r_seed * 214013L + 2531011L))

// return random value in [0..1[ range
double rand1u(void)
{
  return (double)((UPD_SEED >> 16) & 0x7fff)*(1.0/0x8000);
}

// return signed random value in [-1..1[ range
double rand1s(void)
{
  return ((short)(UPD_SEED >> 16))*(1.0/0x8000);
}

// return random boolean state 0/1
int rand_bool(void)
{
  return (short)(UPD_SEED >> 16) >= 0;
}

// ------------------------------------------------
// simulate polarizer

#define OUT_U 0              // coded value for not detectable (dark)
```

```
#define OUT_O 1              // coded value for o output
#define OUT_E 2              // coded value for e output

// define photon local variables type
struct lv_t
{
  double p;                  // polarization
  double q;                  // stochastic variable -1..1
  int e;                     // 0/1 ensemble id
};

#define Ta 0.025             // see paper

static int sim_polarizer(double beta, struct lv_t *lv)
{
  int out;
  double p0 = lv->p;         // save photon initial polarization
  double w = cos(2*(beta - lv->p) + asin(lv->q));
  double rp;                 // photon polarization change

  // select polarizer ouput and update photon polarization
  if (w >= 0) { out = OUT_O; lv->p = beta; }
  else        { out = OUT_E; lv->p = beta + PI/2; }

  // toggle e state if s(out) > s(rp(pi/4)) + Ta
  rp = fabs(fmod(lv->p - p0, PI));  // |rp| polarization delta modulus pi
  if ((rp >= (PI/4 + Ta)) && (rp < (3*PI/4 - Ta)))
    lv->e = !lv->e;

  return out;
}

// ---------------------------------------------
// EPR simulation

#define E_DETECT 1     // detected bright state 0 or 1

// polarisation source type
enum ep_mode
{
  pol_ni = 0,          // 0: missing init => exit program
  pol_hv_vh,           // HV + VH
  pol_hh_vv            // HH + VV
};

// experiment config
struct exp_conf_t
{
  double r;            // r entanglement theoretical value
  double gamma_k;      // ratio of 1st polarization term
  double gamma_v;      // entanglement probability
  enum ep_mode p_mode; // polarization mode p_mode value
```

```
};

// define some non-entangled states
// kA [0..1] ratio of 1st polarization term
void init_exp_no_ent_kA(struct exp_conf_t *conf, enum ep_mode p_mode, double kA)
{
  conf->r = 0;
  conf->gamma_k = kA;  // mix 1 st term
  conf->gamma_v = 0;   // no entanglement
  conf->p_mode = p_mode;
}

// define entangled states using r 0..1
void init_exp_ent_r(struct exp_conf_t *conf, enum ep_mode p_mode, double r)
{
  // note: this basic init follows r fairly closely, but would require
  // a theoretical SPDC model for k/v initialization.
  conf->r = r;               // save theoretical init value
  conf->gamma_k = (1+cos(r*PI/2))/2;
  conf->gamma_v = sin(r*PI/2);
  conf->p_mode = p_mode;
}

// EPR counters
struct ctr_t
{
  int oo, oe, eo, ee;   // coincidences
  int ou, uo, eu, ue;   // singles
  int uu, n_pulse;
};

// angles for H/V
#define H_POL 0
#define V_POL (PI/2)

// simulate EPR 1 pair emission, update coincidence counter
static void epr_sim(const struct exp_conf_t *conf, double beta_a, double beta_b,
struct ctr_t *ctr)
{
  // simulate emission, init Alice/Bob photons local variables
  struct lv_t a_lv, b_lv;
  int a_out, b_out, a_det, b_det;

  if (conf->p_mode == pol_ni)     // pol not initialized
    exit(-1);

  // Init Alice q,e
  a_lv.e = rand_bool();           // random ensemble 0/1
  a_lv.q = rand1s();              // random q

  // init polarization
  if (rand1u() < conf->gamma_k)   // use 1st term
```

```
{
  if (conf->p_mode == pol_hh_vv)
    { a_lv.p = H_POL; b_lv.p = H_POL; } // HH
  else
    { a_lv.p = H_POL; b_lv.p = V_POL; } // HV
}
else                             // use 2nd term
{
  if (conf->p_mode == pol_hh_vv)
    { a_lv.p = V_POL; b_lv.p = V_POL; } // VV
  else
    { a_lv.p = V_POL; b_lv.p = H_POL; } // VH
}

// Init Bob q,e
if (rand1u() < conf->gamma_v)    // entangled state prob
{
  double d_pol = rand1s()*(PI/4); // pol randomize +/- PI/4
  b_lv.e = a_lv.e;               // set same ensemble for Bob
  a_lv.p += d_pol;               // randomize pol Alice

  // type of SPDC
  if (conf->p_mode == pol_hv_vh) // type 2: HV + VH
  {
    b_lv.p -= d_pol;             // opposite randomization
    b_lv.q = -a_lv.q;            // anti bunching ?
  }
  else                           // type 1: HH + VV
  {
    b_lv.p += d_pol;             // same randomization
    b_lv.q = a_lv.q;             // bunching ?
  }
}
else                             // not entangled state
{
  b_lv.e = rand_bool();
  b_lv.q = rand1s();
}

// simulate polarizers
a_out = sim_polarizer(beta_a, &a_lv);
b_out = sim_polarizer(beta_b, &b_lv);

// detection 'bright' photons only
a_det = (a_lv.e == E_DETECT) ? a_out : OUT_U;
b_det = (b_lv.e == E_DETECT) ? b_out : OUT_U;

// update counters
ctr->n_pulse++;                  // num trials
if (a_det == OUT_O)
{
  if      (b_det == OUT_O)  ctr->oo++;
```

```
    else if (b_det == OUT_E)  ctr->oe++;
    else                      ctr->ou++;
  }
  else
  if (a_det == OUT_E)
  {
    if      (b_det == OUT_O)  ctr->eo++;
    else if (b_det == OUT_E)  ctr->ee++;
    else                      ctr->eu++;
  }
  else
  if (a_det == OUT_U)
  {
    if      (b_det == OUT_O)  ctr->uo++;
    else if (b_det == OUT_E)  ctr->ue++;
    else                      ctr->uu++;
  }
}

// repeat N epr tests, update counters
static void epr_test_N(struct exp_conf_t *conf, double beta_a, double beta_b,
struct ctr_t *ctr, int N)
{
  int i;
  for (i=0; i<N; i++)
    epr_sim(conf, beta_a, beta_b, ctr);
}

// --------------------------------------------------------
// gnuplot graphs
// reset;set grid;set xrange [-180:180];set yrange [-1.1:1.1];set xtics 45;plot
"co_2d.txt" using 1:2 with point title "1. co.","co_2d.txt" using 1:3 with lines
title "2. co r.","co_2d.txt" using 1:4 with lines title "3. sing r.","co_2d.txt"
using 1:5 with lines title "4. det r."
void gen_graph_co_4_det(const char *file_name, struct exp_conf_t *conf, double
an_base)
{
  FILE *f = fopen(file_name, "wt");
  if (f)
  {
    int b;
    double beta_a = an_base;
    printf("Define 2D correlations graph %s, please wait..\n", file_name);
    for (b=-180; b<180; b+=5)
    {
      struct ctr_t ctr = { 0 };
      double beta_b = DEG_TO_RAD(b);
      epr_test_N(conf, beta_a, beta_b, &ctr, 500*1000);
      {
        int n_eq = ctr.oo + ctr.ee;
        int n_nq = ctr.oe + ctr.eo;
        int n_co = n_eq + n_nq;
```

```
        int n_si = ctr.ou + ctr.uo + ctr.eu + ctr.ue;
        double E = n_co ? (double)(n_eq - n_nq)/n_co : 0;
        double C = (double)n_co/ctr.n_pulse;      // coincidence ratio / pulse
        double S = (double)n_si/ctr.n_pulse;      // single ratio / pulse
        double dr = (double)(n_co*2 + n_si)/ctr.n_pulse; // apparent detected
photons ratio
        fprintf(f, "%d %.3f %.3f %.3f %.3f %.3f %.3\n", b, E, C, S, dr);
      }
    }
    fclose(f);
  }
}

// return x²
double pow2(double x) { return x*x; }

// QM theoreticals probabilities for r
// |HH> + r|VV>
double p_oo_hh(double x, double y, double r) { return
(1.0/(1.0+r*r))*pow2( cos(x)*cos(y) + r*(sin(x)*sin(y))); }
double p_oe_hh(double x, double y, double r) { return (1.0/(1.0+r*r))*pow2(-
cos(x)*sin(y) + r*(sin(x)*cos(y))); }
double p_eo_hh(double x, double y, double r) { return (1.0/(1.0+r*r))*pow2(-
sin(x)*cos(y) + r*(cos(x)*sin(y))); }
double p_ee_hh(double x, double y, double r) { return
(1.0/(1.0+r*r))*pow2( sin(x)*sin(y) + r*(cos(x)*cos(y))); }
// |HV> + r|VH>
double p_oo_hv(double x, double y, double r) { return
(1.0/(1.0+r*r))*pow2( cos(x)*sin(y) + r*(sin(x)*cos(y))); }
double p_oe_hv(double x, double y, double r) { return
(1.0/(1.0+r*r))*pow2( cos(x)*cos(y) - r*(sin(x)*sin(y))); }
double p_eo_hv(double x, double y, double r) { return (1.0/(1.0+r*r))*pow2(-
sin(x)*sin(y) + r*(cos(x)*cos(y))); }
double p_ee_hv(double x, double y, double r) { return
(1.0/(1.0+r*r))*pow2( sin(x)*cos(y) + r*(cos(x)*sin(y))); }

// set grid;set hidden3d;set xyplane at 0.0;set xrange [-pi:pi];set yrange [-
pi:pi];set zrange [0:1.0]
// splot "co_3d.txt" with lines

// draw coincidence probability in 3d
// oo/oe/eo/ee: added prob if 1
// N_sim -1:QM theoreticals results, N emit pulses for local simulation
void gen_graph_co_3D(const char *file_name, struct exp_conf_t *conf, int oo, int
oe, int eo, int ee, int N_sim)
{
  FILE *f = fopen(file_name, "wb");
  if (f)
  {
    double beta_a;
    printf("Define 3D correlations graph %s, N=%d, please wait..\n", file_name,
N_sim);
```

```
    #define D_AN (PI/30)             // num steps in PI for graph

    for (beta_a = -PI; beta_a <= PI; beta_a += D_AN)
    {
      double beta_b;
      for (beta_b = -PI; beta_b <= PI; beta_b += D_AN)
      {
        double z = 0;                // result prob
        if (N_sim == -1)             // theoretical graph
        {
          double r_th = conf->r;
          if (conf->p_mode == pol_hh_vv)
          {
            if (oo) z  = p_oo_hh(beta_a, beta_b, r_th);
            if (oe) z += p_oe_hh(beta_a, beta_b, r_th);
            if (eo) z += p_eo_hh(beta_a, beta_b, r_th);
            if (ee) z += p_ee_hh(beta_a, beta_b, r_th);
          }
          else
          {
            if (oo) z  = p_oo_hv(beta_a, beta_b, r_th);
            if (oe) z += p_oe_hv(beta_a, beta_b, r_th);
            if (eo) z += p_eo_hv(beta_a, beta_b, r_th);
            if (ee) z += p_ee_hv(beta_a, beta_b, r_th);
          }
        }
        else                         // simulate model
        {
          struct ctr_t ctr = { 0 };  // counters
          int n_co;
          epr_test_N(conf, beta_a, beta_b, &ctr, N_sim);

          n_co = ctr.oo + ctr.oe + ctr.eo + ctr.ee;
          if (n_co != 0)
          {
            if (oo) z  = (double)ctr.oo;
            if (oe) z += (double)ctr.oe;
            if (eo) z += (double)ctr.eo;
            if (ee) z += (double)ctr.ee;
            z /= n_co;
          }
        }
        fprintf(f, "%.2f, %.2f, %.2f\n", beta_a, beta_b, z);        // save
result
      }
      fprintf(f, "\n");
    }
    fclose(f);
  }
}
```

```
// ----------------------------------------------------
// eval CHSH

// get E from counters for CHSH
double get_E(const struct ctr_t *ctr)
{
  int n_eq = ctr->oo + ctr->ee;
  int n_nq = ctr->oe + ctr->eo;
  int n_tot = n_eq + n_nq;
  double E = n_tot ? (double)(n_eq - n_nq)/n_tot : 0;
  return E;
}

// get E theoretical
double get_E_th(struct exp_conf_t *conf, double a, double b)
{
  double E, r = conf->r;
  if (conf->p_mode == pol_hh_vv)
        E = (p_oo_hh(a,b,r) + p_ee_hh(a,b,r)) - (p_oe_hh(a,b,r) +
p_eo_hh(a,b,r));
  else  E = (p_oo_hv(a,b,r) + p_ee_hv(a,b,r)) - (p_oe_hv(a,b,r) +
p_eo_hv(a,b,r));
  return E;
}

// CHSH: https://en.wikipedia.org/wiki/CHSH_inequality
void eval_chsh(struct exp_conf_t *conf, int N)
{
  double S, S_th;
  double a1 = DEG_TO_RAD(0.0);
  double a2 = DEG_TO_RAD(45.0);
  double b1 = DEG_TO_RAD(22.5);
  double b2 = DEG_TO_RAD(67.5);

  struct ctr_t a1_b1 = { 0 };
  struct ctr_t a1_b2 = { 0 };
  struct ctr_t a2_b1 = { 0 };
  struct ctr_t a2_b2 = { 0 };

  printf("Eval CHSH, please wait.. ");
  epr_test_N(conf, a1, b1, &a1_b1, N);
  epr_test_N(conf, a1, b2, &a1_b2, N);
  epr_test_N(conf, a2, b1, &a2_b1, N);
  epr_test_N(conf, a2, b2, &a2_b2, N);

  // S = E(a1b1) - E(a1b2) + E(a2b1) + E(a2b2)
  // simulation result
  S = get_E(&a1_b1) - get_E(&a1_b2) + get_E(&a2_b1) + get_E(&a2_b2);

  // theoretical result
  S_th = get_E_th(conf, a1,b1) - get_E_th(conf, a1,b2) + get_E_th(conf, a2,b1) +
get_E_th(conf, a2,b2);
```

```
  printf("CHSH: %.5f (theoretical:%.5f)\n", fabs(S), fabs(S_th));
}

// generate graphs for paper and eval CHSH
int main(void)
{
  struct exp_conf_t conf = { 0 };
  int N = 20*1000;                    // N sim for 3D graphs
  r_seed = -123;                      // init RNG seed

#if 1
  // graph1, not entangled coincidences H base
  init_exp_no_ent_kA(&conf, pol_hh_vv, 0.5);  // no entanglement HH + VV 50%
  gen_graph_co_4_det("g1_a.txt", &conf, DEG_TO_RAD(0.0));  // aligned H vis 1

  // graph2, entangled coincidences HV diagonal base
  init_exp_ent_r(&conf, pol_hv_vh, 1.0f);      // max entanglement HV + VH
  gen_graph_co_4_det("g1_b.txt", &conf, DEG_TO_RAD(45.0)); // HV diagonal
#endif

#if 1
  // 3d probs graphs
  // not entangled correlations HH VV, OO OE EO EE
  init_exp_ent_r(&conf, pol_hh_vv, 0.0f);
  gen_graph_co_3D("g2_a1.txt", &conf, 1, 0, 0, 0, N); // OO
  gen_graph_co_3D("g2_a2.txt", &conf, 0, 1, 0, 0, N); // OE
  gen_graph_co_3D("g2_a3.txt", &conf, 0, 0, 1, 0, N); // EO
  gen_graph_co_3D("g2_a4.txt", &conf, 0, 0, 0, 1, N); // EE

  // entangled sim + theoretical HH VV, HV VH OO+EE
  init_exp_ent_r(&conf, pol_hh_vv, 1.0f);
  gen_graph_co_3D("g3_a1.txt", &conf, 1, 0, 0, 1,  N); // OO + EE sim
  gen_graph_co_3D("g3_a2.txt", &conf, 1, 0, 0, 1, -1); // OO + EE theoretical
  init_exp_ent_r(&conf, pol_hv_vh, 1.0f);
  gen_graph_co_3D("g3_a3.txt", &conf, 1, 0, 0, 1,  N); // OO + EE sim
  gen_graph_co_3D("g3_a4.txt", &conf, 1, 0, 0, 1, -1); // OO + EE theoretical

  // partial entanglemement sim + theoretical HH VV, HV VH OO+EE
  init_exp_ent_r(&conf, pol_hh_vv, 0.5f);
  gen_graph_co_3D("g4_a1.txt", &conf, 1, 0, 0, 1,  N); // OO + EE sim
  gen_graph_co_3D("g4_a2.txt", &conf, 1, 0, 0, 1, -1); // OO + EE theoretical
  init_exp_ent_r(&conf, pol_hv_vh, 0.5f);
  gen_graph_co_3D("g4_a3.txt", &conf, 1, 0, 0, 1,  N); // OO + EE sim
  gen_graph_co_3D("g4_a4.txt", &conf, 1, 0, 0, 1, -1); // OO + EE theoretical
#endif

#if 1
  // eval CHSH
  init_exp_ent_r(&conf, pol_hh_vv, 0.0f); eval_chsh(&conf, 10*1000*1000); //
1.41..
  init_exp_ent_r(&conf, pol_hh_vv, 1.0f); eval_chsh(&conf, 10*1000*1000); //
2.82..
```

```
  init_exp_ent_r(&conf, pol_hv_vh, 0.0f); eval_chsh(&conf, 10*1000*1000); //
1.41..
  init_exp_ent_r(&conf, pol_hv_vh, 1.0f); eval_chsh(&conf, 10*1000*1000); // 0
(need CHSH angles changes for HV+VH)
#endif
  return 0;
}
```

Version 1.0 May 18 2025