

# A Python tool for assessing compatibility of observations with arbitrary causal structures with classical or quantum latent variables

Emanuel-Cristian Boghiu<sup>1</sup> and Alejandro Pozas-Kertjens<sup>2</sup>

<sup>1</sup>ICFO-The Institute of Photonic Sciences, The Barcelona Institute of Science and Technology, Castelldefels (Barcelona), Spain

<sup>2</sup>Instituto de Ciencias Matemáticas (CSIC-UAM-UC3M-UCM), Madrid, Spain

## BACKGROUND

### CAUSAL INFERENCE

Finding which causes are behind observed correlations in nature is at the core of the scientific discipline. Observed correlations are represented by the joint probability distribution of a set of classical random variables. One can ask whether certain causal relationships between variables are compatible with the observed correlations. This is known as the **causal compatibility problem**. Bayesian causal networks, in the form of directed acyclic graphs (DAGs), give the tools to formalise such questions. Each causal hypothesis can be encoded as a DAG, with causal influence between variables being represented as directed edges.

### QUANTUM CAUSAL SCENARIOS

In quantum theory one cannot always assign a deterministic output to measurement results: the theory is inherently probabilistic. Thus, measurement results are represented as random variables. This naturally leads to ask which cause-and-effect relationships can explain observed measurement statistics. A simple experiment is two spacelike-separated parties performing local measurements on a shared entangled quantum state, Fig. 1 a). The **quantum causal compatibility problem** is to find whether some observed correlations are compatible in a general DAG where we also have sources of quantum correlations, and not only classical [1].

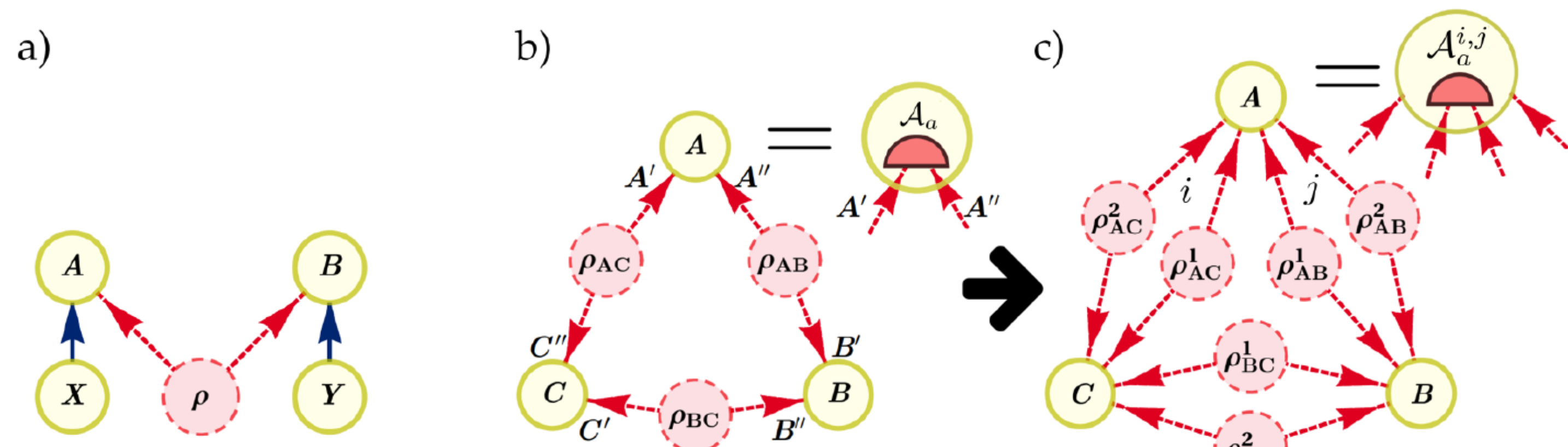


Figure 1. Different DAGs. In a) we have the standard Bell scenario, in b) we have the triangle network with no inputs and in c) we have an inflation of the triangle network.

### QUANTUM INFLATION

Quantum inflation [2] is a technique that can tackle the **quantum causal compatibility problem**. The main challenge lies in solving the problem comes from the statistical independence of the sources. Consider the triangle network in Fig. 1 b). Inflation relaxes the condition of independence in Fig 1 b) to symmetries on a larger “inflated” graph with copied sources, as in Fig 1 c). Symmetries are imposed on a further SDP relaxation of the problem based on the NPA hierarchy. Such SDPs look like the following toy example:

find real numbers  $v_1, v_2, v_3,$

$$\text{s.t. } \Gamma = \begin{pmatrix} \mathbb{1} & \mathcal{A}^{1,1} & \mathcal{A}^{1,2} & \mathcal{A}^{2,1} & \mathcal{A}^{2,2} & \mathcal{B}^{1,1} \mathcal{C}^{1,1} \\ \mathcal{A}^{1,1} & \langle \mathcal{A}^{1,1} \rangle & \langle \mathcal{A}^{1,2} \rangle & \langle \mathcal{A}^{2,1} \rangle & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle \\ \mathcal{A}^{1,2} & \langle \mathcal{A}^{1,2} \rangle & v_1 & \langle \mathcal{A}^{2,1} \rangle & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{A}^{1,1} \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle \\ \mathcal{A}^{2,1} & \langle \mathcal{A}^{2,1} \rangle & \langle \mathcal{A}^{1,2} \rangle & v_1 & v_1 & v_2 \\ \mathcal{A}^{2,2} & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{A}^{1,2} \rangle & \langle \mathcal{A}^{2,1} \rangle & v_1 & v_3 \\ \mathcal{B}^{1,1} \mathcal{C}^{1,1} & \langle \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{A}^{2,2} \rangle & \langle \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle \end{pmatrix} \succeq 0$$

$P_{\text{observed}}(a=0) = \langle \mathcal{A}^{1,1} \rangle = \langle \mathcal{A}^{1,2} \rangle = \langle \mathcal{A}^{2,1} \rangle = \langle \mathcal{A}^{2,2} \rangle$   
 $P_{\text{observed}}(b=0, c=0) = \langle \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle, \quad P_{\text{observed}}(a=0, b=0, c=0) = \langle \mathcal{A}^{1,1} \mathcal{B}^{1,1} \mathcal{C}^{1,1} \rangle$   
 $\Gamma_{ij} = \text{tr } \rho^{\text{inflated DAG}} \mathcal{O}_i^\dagger \mathcal{O}_j, \quad \mathcal{O} = \{ \mathbb{1}, \mathcal{A}^{1,1}, \mathcal{A}^{1,2}, \mathcal{A}^{2,1}, \mathcal{A}^{2,2}, \mathcal{B}^{1,1} \mathcal{C}^{1,1} \}.$

Trustees:

## WHY, AND FEATURES

Inflation methods are, in general, complex to implement due to the large amount of symbolic computations a priori needed to calculate the SDP relaxation. Current implementations are restricted to small, specific networks and particular inflation DAGs (see, e.g., the computational appendices of [3, 4]). QINFLATION is the first open-source implementation of the quantum inflation technique that applies to any DAG, input-output configuration, and nature of the latent variables (classical or quantum). Computations are compiled in a just-in-time procedure via the Numba package [5], speeding up the runtime. The first version of the package, along with an accompanying paper explaining its documentation and exemplifying different use cases, will be released shortly.

## HOW TO USE THE PACKAGE

The package follows an object-oriented design. The focus is put on user experience, so only a small amount of basic parameters is needed to address compatibility or optimisation problems. The following shows how to implement the inflation from Fig 1 c):

```
# Triangle network, Figure 1 b), as a hypergraph.
# Entry (i,j) specifies if source i connects with party j.
causal_network = [[1, 1, 0], [0, 1, 1], [1, 0, 1]]

# Two copies of each source, network in Figure 1 c).
inflation_level = [2, 2, 2]

# For each source, if we treat it as quantum mechanical or classical.
if_source_classical = [False, False, True]

# Number of outputs for each party. Optional: specify how many inputs "a la Bell"
output_cardinalities, input_cardinalities = [2, 2, 2], [1, 1, 1]

problem = QInflation.scenario(causal_network, inflation_level, if_source_classical,
                             output_cardinalities, input_cardinalities)

A11, A12, A21, ... = problem.measurement_operators()

problem.sdp_relaxation([1, A11, A12, A21, A22, B11*C11])
```

```
Option 1: do a causal compatibility test
p = ... # p is an ndarray indexed as # [a,b,c,...,x,y,z,...]
problem.set_probability(p)
problem.solve()
problem.incompatibility_certificate()

Option 2: optimize a polynomial function
1 objective = ... # Symbolic polynomial # of the measurements
2
3
4 problem.set_objective(objective)
5
6 problem.solve() # Maximises by default
7
8 print(problem.objective_value)
```

## REFERENCES

[1] A. Tavakoli et al., Bell nonlocality in networks, Reports on Progress in Physics 85, 056001 (2022).  
[2] E. Wolfe, A. Pozas-Kerstjens, M. Grinberg, D. Rosset, A. Acín, and M. Navascués, Quantum inflation: A general approach to quantum causal compatibility, Physical Review X 11, 021043 (2021).  
[3] A. Pozas-Kerstjens, N. Gisin, and A. Tavakoli, Phys. Rev. Lett. 128, 010403 (2022).  
[4] A. Pozas-Kerstjens, N. Gisin, and M.-O. Renou, arXiv:2203.16543 (2022).  
[5] S. K. Lam et al., in Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15 (Association for Computing Machinery, New York, NY, USA, 2015).

Supported by:

Member of: