Numerical simulation of randomly generated states for two-qubit system according to the LHVM paradigm and entropy measure

Department of physics, Faculty of science, Ferdowsi university of Mashhad

Under the guidance of **Dr. S. J. Akhtarshenas**

By **Arash T. Jamshidi**

September 2024

- *State space*
- *Hilbert-Schmidt form of density matrix*
- *Two-qubit system and defining s_j r_i t_{ij}*
- *Generating random uniform states*
- *Problems that we have faced!*
- *Separability and entanglement*
- *EPR – Bell inequalities*
- *Horodecki's theorem*
- *Results*
- *Entropic Manifestation of Entanglement*
- *MEMS and Results*

“ *State Space*

$$S(\mathcal{H}) : \begin{cases} Hermition \\ Trace = 1 \\ Posituvity \end{cases}$$

Real coefficients

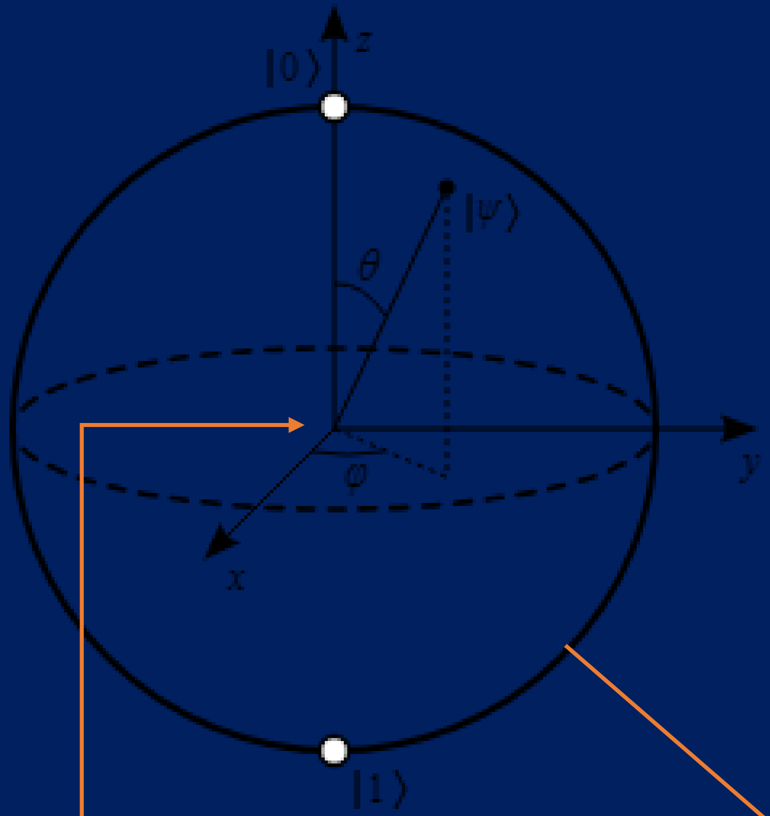
$$\rho = \frac{1}{2}(\mathbb{I} + \vec{r}.\vec{\sigma})$$

- $\vec{r}$: *Bloch vector*
- $\vec{\sigma}$: *Pauli matrices*

$$\sqrt{x^2 + y^2 + z^2} \leq 1$$

Bloch sphere

“ Bloch sphere



$r = 0$

$\rho = \frac{1}{2}\mathbb{I}$

Totally mix states

$\rho = |\Psi\rangle\langle\Psi|$

Pure states

“ Hilbert-Schmidt

Qudit !

$$\rho = \frac{1}{d}\left(\mathbb{I} + \vec{r}.\vec{\lambda}\right)$$

- $\vec{\lambda}$: *Gell-Mann matrices*

$2^d - 1$

For two-qubit system : $\mathcal{H} = \mathbb{C}^2 \otimes \mathbb{C}^2$

$$\rho = \frac{1}{4}\left(\mathbb{I} \otimes \mathbb{I} + (\vec{r}.\vec{\sigma}) \otimes \mathbb{I} + \mathbb{I} \otimes (\vec{s}.\vec{\sigma}) + t_{ij}(\vec{\sigma}_i \otimes \vec{\sigma}_j)\right)$$

$$\rho = \frac{1}{4}\left(\mathbb{I} \otimes \mathbb{I} + (\vec{r}.\vec{\sigma}) \otimes \mathbb{I} + \mathbb{I} \otimes (\vec{s}.\vec{\sigma}) + t_{ij}(\vec{\sigma}_i \otimes \vec{\sigma}_j)\right)$$

$Tr\left(\sigma_i \sigma_j\right) = 2\delta_{ij}$

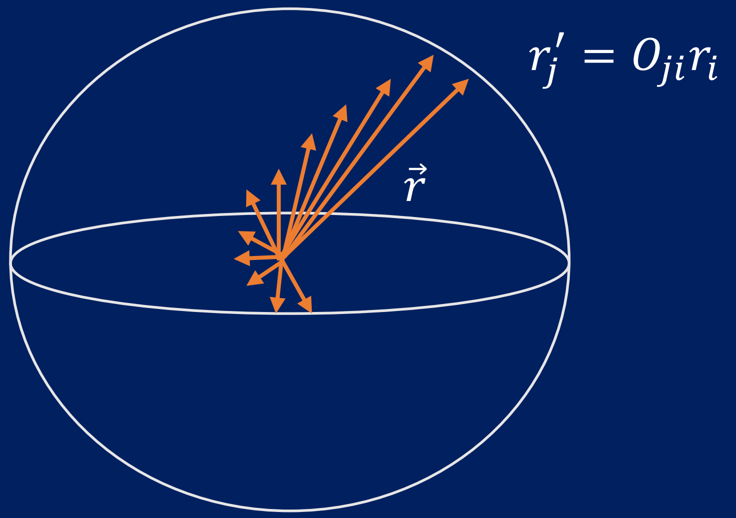
$$r_i = Tr[\rho(\vec{\sigma}_i \otimes \mathbb{I})]$$

$$t_{ij} = Tr[\rho\left(\vec{\sigma}_i \otimes \vec{\sigma}_j\right)]$$

$$s_j = Tr[\rho\left(\mathbb{I} \otimes \vec{\sigma}_j\right)]$$

“ How we generate random states

$$\rho = \frac{1}{d}(\mathbb{I} + \vec{r}.\vec{\lambda}) \qquad SO(3) \leftrightarrow SU(2)$$



$$\rho' = \frac{1}{d}(\mathbb{I} + \vec{r}'.\vec{\lambda}) \rightarrow \rho' = \frac{1}{d}\left(\mathbb{I} + \sum_{i,j} O_{ji} r_i \lambda_j\right)$$

$$u\lambda_i u^\dagger = \sum_{j=1}^{N^2-1} O_{ji}\lambda_j \qquad \rho' = \frac{1}{d}\left(\mathbb{I} + \sum_{i,j} u\lambda_i u^\dagger r_i\right)$$

$$\rightarrow \rho' = \frac{1}{d}(\mathbb{I} + u(\vec{r}.\vec{\lambda})u^\dagger) = u\{\frac{1}{d}(\mathbb{I} + (\vec{r}.\vec{\lambda}))\}u^\dagger$$

$$\rho \rightarrow \rho' = u\rho u^\dagger$$

“ How we generate random uniform probability

$$\rho' = u\rho_d u^\dagger$$

$$\rho_d = \begin{bmatrix} P_1 & 0 & 0 & 0 \\ 0 & P_2 & 0 & 0 \\ 0 & 0 & P_3 & 0 \\ 0 & 0 & 0 & P_4 \end{bmatrix} \qquad \sum_i P_i = 1$$

*Which P_i are randomly generated with **Dirichlet distribution**, from **NumPy** package in python.*

it worth to notice that this function uniformly generate initial probabilities.

```
for i in range(num_denmet):
    def generate_diagonal_elements(d):
    # Generate n random numbers following Dirichlet distribution
        random_numbers = np.random.dirichlet(np.ones(d))
        return random_numbers

    diagonals = generate_diagonal_elements(d)

    set_ro_d.append(diagonals*np.identity(d))
```

“ *How we generate random Unitary matrix*

$$\rho' = u\rho u^{\dagger}$$

- *Uniform , random unitary matrix*

“steps

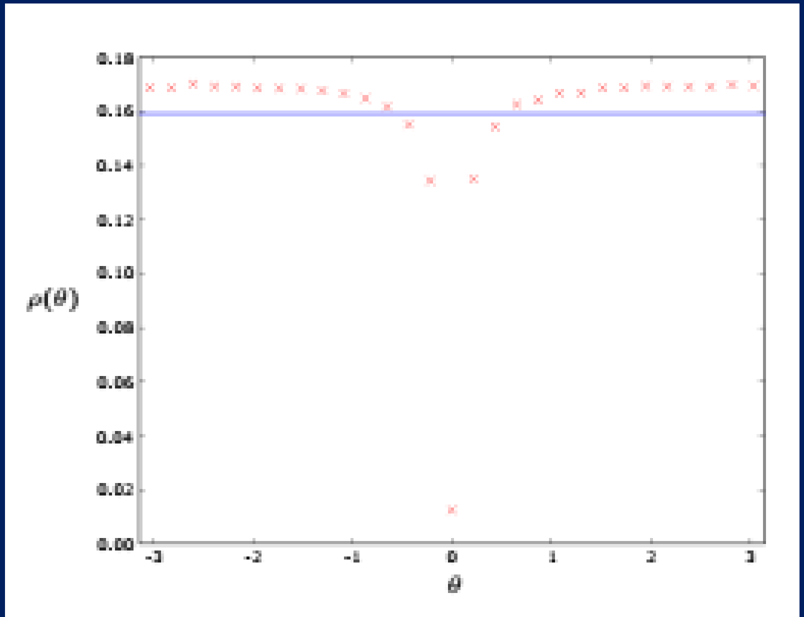
1. First we generate a random complex matrix
2. We use QR decomposition to decompose our random complex matrix to one unitary matrix (Q) and one upper-triangular (R)

- But the output is not distributed with Haar measure.

in the case of unitary matrices distributed according to the Haar measure:

(large matrices)
random matrix theory

1. eigenvalues tend to repel each other (level repulsion)
2. average density of eigenvalues around the unit circle is approximately constant



(a) Eigenvalue density

- F. Mezzadri, "How to generate random matrices from the classical compact groups," Notices of the American Mathematical Society, 54 (5), 592-604 (2007).

“ *Our problem with random unitary matrix distribution*

The main problem is that QR decomposition is not unique !

$Z \epsilon\ GL(N, \mathbb{C})\ ,\ Z = QR$ *which* Q *is unitary and* R *is upper triangular matrices.*

$Z = Q\Lambda\Lambda^{-1}R$

$$let\ \Lambda = \begin{pmatrix} e^{i\theta_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{i\theta_N} \end{pmatrix} = diag\left(e^{i\theta_1}, \dots, e^{i\theta_N}\right) \qquad \Lambda\ \epsilon\ \Lambda(N)$$

"unitary diagonal matrices group "

$Q' = Q\Lambda$ *and* $R' = \Lambda^{-1}R$ *are still unitary and upper riangular respectively.*

QR decomposition defines a multi valued map: $\qquad Z = QR = Q'R'$

$$QR: GL(N, \mathbb{C}) \ \to U(N) \times T(N)$$

- F. Mezzadri, "How to generate random matrices from the classical compact groups," Notices of the American Mathematical Society, 54 (5), 592-604 (2007).

“ Our problem with random unitary matrix distribution

$QR: GL(N,\mathbb{C}) \to U(N) \times T(N)$ $\qquad$ $QR = Q'R'$

$\Lambda(N) = U(N) \cap T(N)$ $\qquad$ *so we should consider one–to–one map*:

$$\overline{QR}: GL(N,\mathbb{C}) \to U(N) \times \Gamma(N)$$

where $\Gamma(N) = T(N)/\Lambda(N)$ *is the right coset space of* $\Lambda(N)$ *in* $T(N)$

can be chosen by fixing the arguments of the elements of the main diagonal of R ϵ T(N)

$$\Lambda = \begin{pmatrix} \frac{r_{11}}{|r_{11}|} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{r_{NN}}{|r_{NN}|} \end{pmatrix}$$

r_{jj}

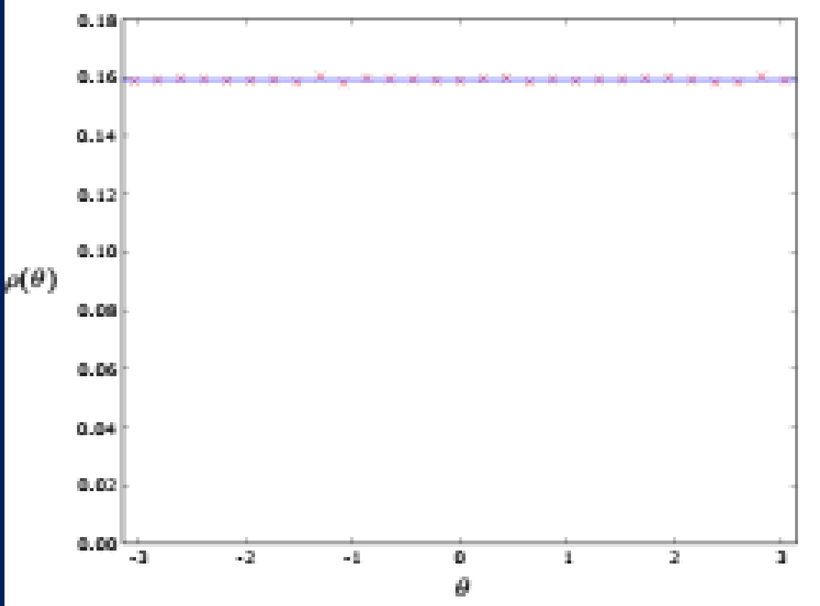
- F. Mezzadri, "How to generate random matrices from the classical compact groups," Notices of the American Mathematical Society, 54 (5), 592-604 (2007).

“ *How we generate random Unitary matrix*

$$\rho' = u\rho u^{\dagger}$$

$The\ Q' = Q\Lambda$ *is distributed with Haar measure*

```
for i in range(num_state):
    def random_unitary(d):
        # Generate a random complex matrix
        X = np.random.randn(d, d) + I * np.random.randn(d, d)

        # QR decomposition
        Q, R = np.linalg.qr(X)

        # Create a diagonal matrix with phase factors
        D = np.diag(np.diag(R) / np.abs(np.diag(R)))

        # Construct the unitary matrix
        U = np.dot(Q, D)

        return U

    set_state.append(random_unitary(d))
```

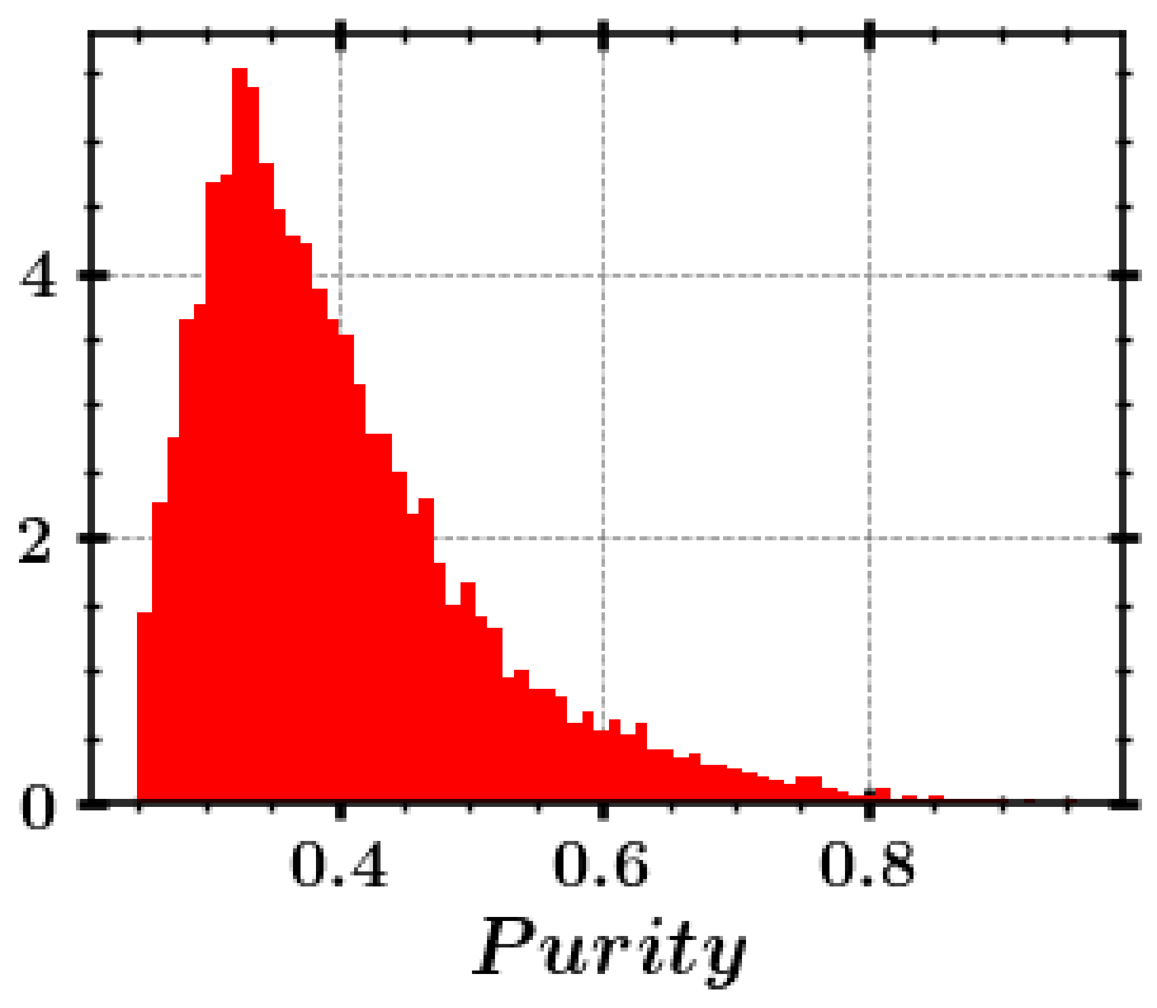


(a) Eigenvalue density

- small fluctuations due to the repulsion effect

- F. Mezzadri, "How to generate random matrices from the classical compact groups," Notices of the American Mathematical Society, 54 (5), 592-604 (2007).

" *Purity of randomly generated states*



“ Separability and Entanglement

$$\rho = \sum P_k(\rho_A^k \otimes \rho_B^k)$$

If it can’t be written as convex sum of pure states, ρ is Entangled !

“ *EPR - Bell - CHSH*

1935

1964

1969

$$P(a,b) = \int A(a,\lambda)B(b,\lambda)\rho(\lambda)d\lambda$$

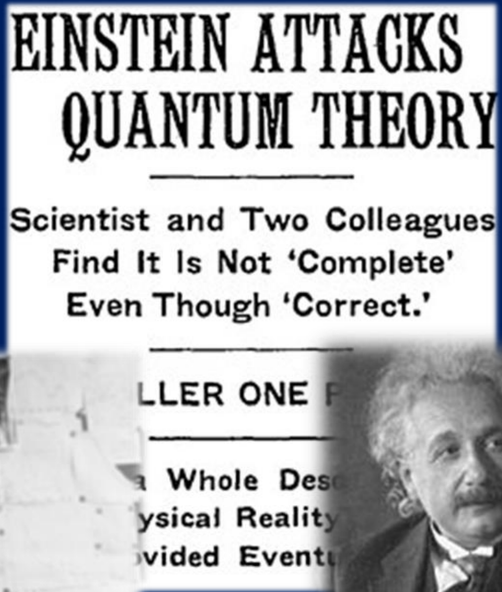
$$\langle B\rangle \leq \beta_{LR}$$

“ *CHSH*

$$\boldsymbol{B_{CHSH}} = \mathbf{A} \otimes [\boldsymbol{B} + \boldsymbol{B}'] + \boldsymbol{A}' \otimes [\boldsymbol{B} - \boldsymbol{B}']$$

$$\langle \boldsymbol{B_{CHSH}} \rangle = Trac[\rho \boldsymbol{B_{CHSH}}]$$

$$|\langle \boldsymbol{B_{CHSH}} \rangle| \leq 2$$



EINSTEIN ATTACKS QUANTUM THEORY

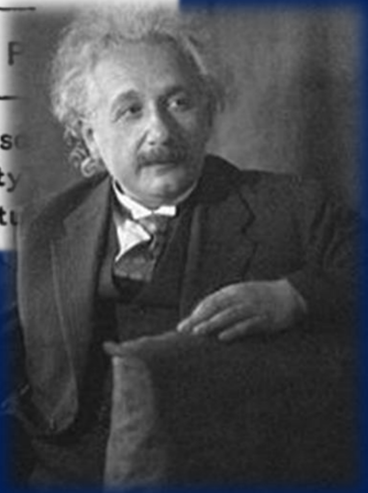
Scientist and Two Colleagues Find It Is Not 'Complete' Even Though 'Correct.'

LLER ONE

a Whole Des

ysical Reality

vided Event



- *Locality*
- *Realism*

- A. Einstein, B. Podolsky, and N. Rosen, "Can quantum-mechanical description of physical reality be considered complete?" Physical Review, 47 (10), 777-780 (1935)
- J. S. Bell, "On the Einstein Podolsky Rosen Paradox," Physics Physique Физика, 1 (3), 195-200 (1964).
- J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," Physical Review Letters, 23 (15), 880-884 (1969)

$$\boldsymbol{B_{CHSH}} = \mathbf{A} \otimes [\boldsymbol{B} + \boldsymbol{B'}] + \boldsymbol{A'} \otimes [\boldsymbol{B} - \boldsymbol{B'}]$$

$$\rho = \frac{1}{4}\left(\mathbb{1} \otimes \mathbb{1} + (\vec{r}.\vec{\sigma}) \otimes \mathbb{1} + \mathbb{1} \otimes (\vec{s}.\vec{\sigma}) + t_{ij}(\vec{\sigma}_i \otimes \vec{\sigma}_j)\right)$$

$$\langle \boldsymbol{B_{CHSH}} \rangle = Trac[\rho \boldsymbol{B_{CHSH}}]$$

- $A' = \sigma.a'$
- $A = \sigma.a$
- $B' = \sigma.b'$
- $B = \sigma.b$

$$Tr(\sigma_i \sigma_j) = 2\delta_{ij} \rightarrow \cancel{r_i} \;\&\; \cancel{s_j} \qquad \langle \boldsymbol{B_{CHSH}} \rangle \propto t_{ij}$$

$$\langle \boldsymbol{B_{CHSH}} \rangle = \boldsymbol{Tr}\left[\frac{\mathbf{1}}{\mathbf{4}} t_{ij}(\vec{\sigma}_i \otimes \vec{\sigma}_j)[\boldsymbol{B_{CHSH}}]\right] \rightarrow \langle \boldsymbol{B_{CHSH}} \rangle = \sum a_i \, t_{ij}(b + b')_j + a'_i t_{ij}(b - b')_j$$

$$t_{ij} = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

"Correlation tensor"

“ Horodecki’s measure of maximum Bell’s operator expectation value

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

→ *We cannot necessarily diagonalize the matrix T but we can calculate the matrix $T^{\dagger}T$ which is both diagonal and positive.*

Horodecki’s theorem : *The maximum expectation value of the Bell operator is equal to:*

$$Max\{\langle \boldsymbol{B_{CHSH}} \rangle\} = 2\sqrt{u_1 + u_2}$$

Where u_i are the non-negative eigenvalues of the positive and real matrix $T^{\dagger}T$, such that :

$$u_1 \geq u_2 \geq u_3 \geq 0$$

- R. Horodecki, P. Horodecki, and M. Horodecki, "Violating Bell inequality by mixed spin-1/2 states: necessary and sufficient condition," Physical Review Letters, 74 (20), 414-417 (1995)

“ *steps*

$$random\ 4 \times 4\ \rho_i$$

$$\rho = \frac{1}{4}\left(\mathbb{I} \otimes \mathbb{I} + (\vec{r}.\vec{\sigma}) \otimes \mathbb{I} + \mathbb{I} \otimes (\vec{s}.\vec{\sigma}) + t_{ij}(\vec{\sigma}_i \otimes \vec{\sigma}_j)\right)$$

$$t_{ij} = Tr[\rho(\vec{\sigma}_i \otimes \vec{\sigma}_j)]$$

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} \\ t_{21} & t_{22} & t_{23} \\ t_{31} & t_{32} & t_{33} \end{bmatrix}$$

$$eig(T^\dagger T)$$

$$u_1 \geq u_2 \geq u_3 \geq 0$$

$$Max\{\langle \boldsymbol{B_{CHSH}} \rangle\} = 2\sqrt{u_1 + u_2}$$

$$|\langle \boldsymbol{B_{CHSH}} \rangle| \leq 2$$

$$if\ \sqrt{u_1 + u_2} \geq 1\ \rightarrow\ \rho \text{ is Entagled}$$

“ Main loop for checking the Bell’s inequality

```
for q in range(num_denmet):
    for j in range(num_state):

        TT =[]
        ro = set_state[j]@set_ro_d[q]@Dagger(set_state[j])
        density_matrix.append(ro)
        puty = np.trace(density_matrix[q]@(density_matrix[q].conj().T))
        purity.append(abs(puty))
        for i in range(len(sig_prdct)):
            t = np.trace(density_matrix[q]@sig_prdct[i])
            TT.append(t)
        T = np.array([[TT[0],TT[1],TT[2]],
          [TT[3],TT[4],TT[5]],
          [TT[6],TT[7],TT[8]]])

        eigval , eigvec = np.linalg.eig(Dagger(T)@T)
        eeg = abs(eigval)
        g = np.trace(np.diag(eeg))-min(eeg)

        max_Exp = 2 * np.sqrt(g)
        Expectation_sum.append(max_Exp)
        if max_Exp > 2:
            Entangled_state.append(set_state[j])
            purity_ents = np.trace(abs(density_matrix[q]@density_matrix[q].conj().T))
            good_ro_purity.append(purity_ents)
            Expectation_sum_notBell.append(g)

        else:
            Separable_state.append(set_state[j])
            Expectation_sum_Bell.append(g)
```

“ Main loop for checking the Bell’s inequality

```
for q in range(num_denmet):
    for j in range(num_state):

        TT =[]
        ro = set_state[j]@set_ro_d[q]@Dagger(set_state[j])
        density_matrix.append(ro)
        puty = np.trace(density_matrix[q]@(density_matrix[q].conj().T))
        purity.append(abs(puty))
        for i in range(len(sig_prdct)):
            t = np.trace(density_matrix[q]@sig_prdct[i])
            TT.append(t)
        T = np.array([[TT[0],TT[1],TT[2]],
          [TT[3],TT[4],TT[5]],
          [TT[6],TT[7],TT[8]]])

        eigval , eigvec = np.linalg.eig(Dagger(T)@T)
        eeg = abs(eigval)
        g = np.trace(np.diag(eeg))-min(eeg)

        max_Exp = 2 * np.sqrt(g)
        Expectation_sum.append(max_Exp)
        if max_Exp > 2:
            Entangled_state.append(set_state[j])
            purity_ents = np.trace(abs(density_matrix[q]@density_matrix[q].conj().T))
            good_ro_purity.append(purity_ents)
            Expectation_sum_notBell.append(g)

        else:
            Separable_state.append(set_state[j])
            Expectation_sum_Bell.append(g)
```

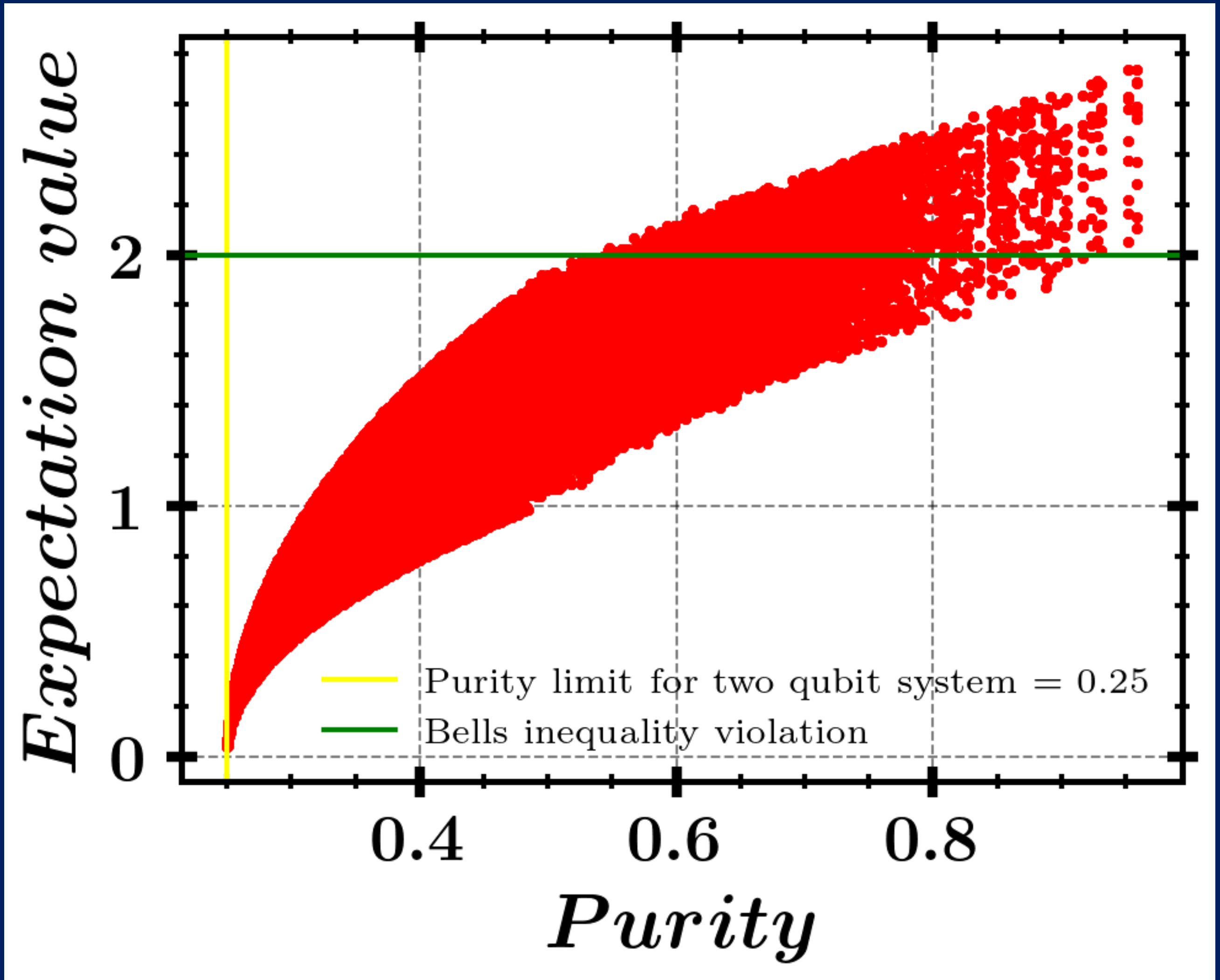
```
sigma_x = np.array([[0,1],
                    [1,0]])
sigma_y = np.array([[0,-I],
                    [I,0]])
sigma_z = np.array([[1,0],
                    [0,-1]])

sigma = [sigma_x ,sigma_y ,sigma_z]
#-----------------
for i in range(0,3):
    for j in range(0,3):
        tt = np.kron(sigma[i],sigma[j])
        sig_prdct.append(tt)
```

$$t_{ij} = Tr[\rho(\vec{\sigma}_i \otimes \vec{\sigma}_j)]$$

- ρ_d : 100000
- u_i : 10

Example run

Bell's inequality as witness suggest that we have 27880 Entangled state out of 696700 possible ones.



4.002 % of the randomly generated states are Entangled state as far as Bell's inequality can tell.

“ *Entropic Manifestation of Entanglement*

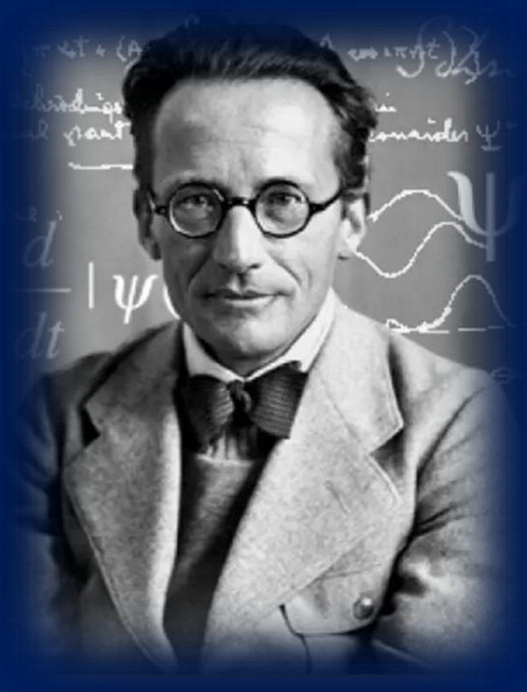
$$Classic : I\,(x|y) = I(x,y) - I(y) \geq 0$$

$$QM : I\,(A|B) = S(\rho_{AB}) - S(\rho_A)\ \ !$$

If ρ_{AB} is separable then $\rightarrow I\,(A|B)\ \geq 0$

$$\rho = \sum P_k(\rho_A^k \otimes \rho_B^k)$$

$$\begin{cases} S(\rho_{AB}) \geq S(\rho_A) \\ S(\rho_{AB}) \geq S(\rho_B) \end{cases}$$



“The best possible knowledge of a whole does not include the best possible knowledge of its parts and this is what keeps coming back to haunt us”

- Horodecki and Horodecki, 1996: R. Horodecki and M. Horodecki, "Information-theoretic aspects of quantum inseparability of mixed states," Physical Review A, 54 (3), 1838-1843 (1996)
- R. Horodecki et al., 1996: R. Horodecki, M. Horodecki, and P. Horodecki, "Separability of mixed states: necessary and sufficient conditions," Physics Letters A, 223 (1-2), 1-8 (1996)
- Terhal, 2002: B. M. Terhal, "Detecting quantum entanglement," Theoretical Computer Science, 287 (1), 313-335 (2002)
- Vollbrecht and Wolf, 2002: K. G. H. Vollbrecht and M. M. Wolf, "Conditional entropies and their relation to entanglement criteria," Journal of Mathematical Physics, 43 (9), 4299-4306 (2002)

“ calculating the partial trace

```
#  calculating partial trace
density_A = []
density_B = []
for i in range(len(density_matrix)):
    BB = np.split(density_matrix[i], 2, axis=1)
    BB0 = np.split(BB[0] , 2, axis=0)
    BB1 = np.split(BB[1] , 2, axis=0)
    B00 = BB0[0]
    B01 = BB1[0]
    B10 = BB0[1]
    B11 = BB1[1]
    P_Tr_B = np.array([[np.trace(B00),np.trace(B01)],
                       [np.trace(B10),np.trace(B11)]])
    density_A.append(P_Tr_B)
    P_Tr_A = np.array([[B00[0][0]+B11[0][0],B00[0][1]+B11[0][1]],
                       [B00[1][0]+B11[1][0],B00[1][1]+B11[1][1]]])
    density_B.append(P_Tr_A)
```

“ Calculating the Von Neumann entropy

```
#calculating of Von Neumann entropy
import math
Entropy_A = []
Entropy_AB =[]
for i in range(len(density_A)):
    eigval , eigvec = np.linalg.eig(density_A[i])
    eg = abs(eigval)
    s = -(eg[0]*(math.log(eg[0])) + eg[1]*(math.log(eg[1])))
    Entropy_A.append(s)

for i in range(len(density_matrix)):
    eigval , eigvec = np.linalg.eig(density_matrix[i])
    eg = abs(eigval)
    s = -( eg[0]*(math.log(eg[0])) + eg[1]*(math.log(eg[1])) + eg[2]*(math.log(eg[2])) + eg[3]*(math.log(eg[3])))
    Entropy_AB.append(s)
```

```python
# calculating entropy witness
separables_S = []
entangleds_S = []
for i in range(len(Entropy_AB)):
    if Entropy_AB[i]>=Entropy_A[i]:
        separables_S.append(density_matrix[i])
    else:
        entangleds_S.append(density_matrix[i])
```

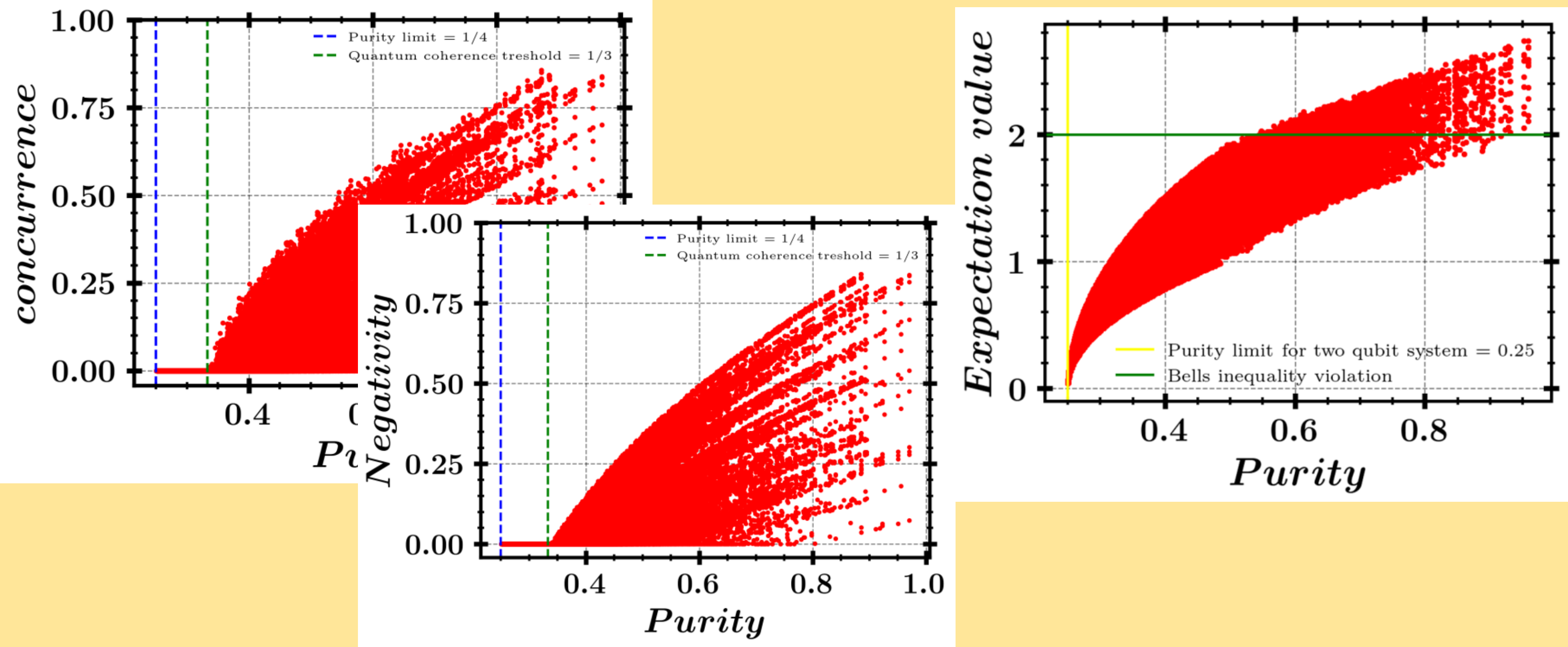
Entropy as witness suggest that we have 16535 Entangled state.

1.6535 % of the generated random states are Entangled state as far as Entropy as witness can tell.

Thanks for your attention !
To be continued …

As we saw in previous presentation we investigate different entanglement measures

different entanglement measures can give different orderings for pairs of mixed states.

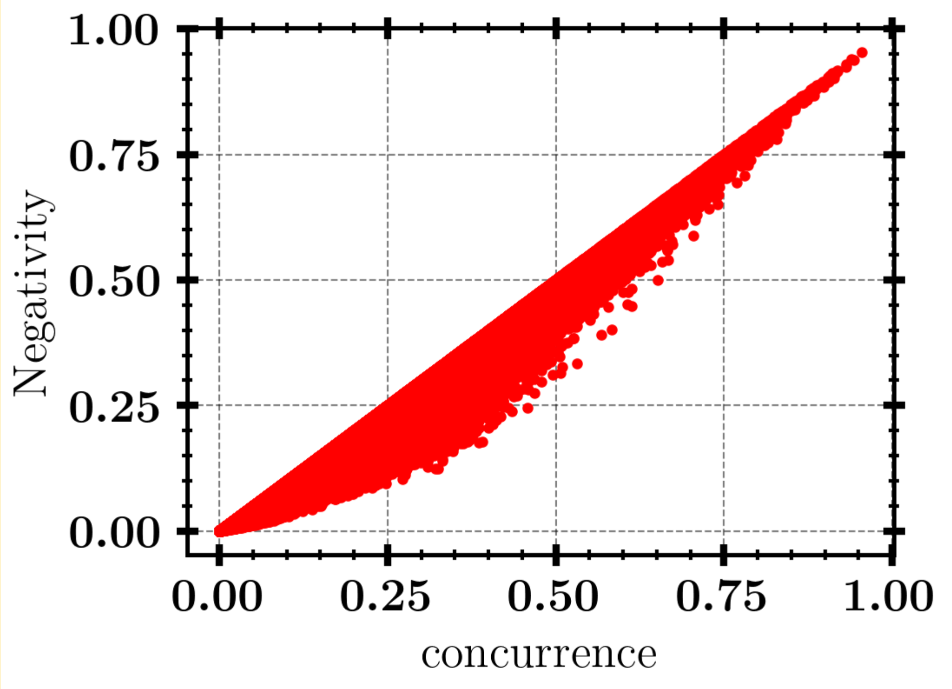


Now we investigate the highest possible entanglement for a given level of mixedness

"MEMS"

- G. Vidal and R. F. Werner, 2002: G. Vidal and R. F. Werner, "A computable measure of entanglement," Physical Review Letters, 89 (17), 170401 (2002)
- K. Zyczkowski, P. Horodecki, A. Sanpera, and M. Lewenstein, 1998: K. Zyczkowski, P. Horodecki, A. Sanpera, and M. Lewenstein, "Volume of the set of separable states," Physical Review Letters, 85 (17), 2961-2964 (1998)
- A. Peres, 1996: A. Peres, "Separability criterion for density matrices," Physical Review Letters, 77 (8), 1413-1415 (1996)
- M. Horodecki, P. Horodecki, and R. Horodecki, 1996: M. Horodecki, P. Horodecki, and R. Horodecki, "Separability of mixed states: necessary and sufficient conditions," Physics Letters A, 223 (1-2), 1-8 (1996)

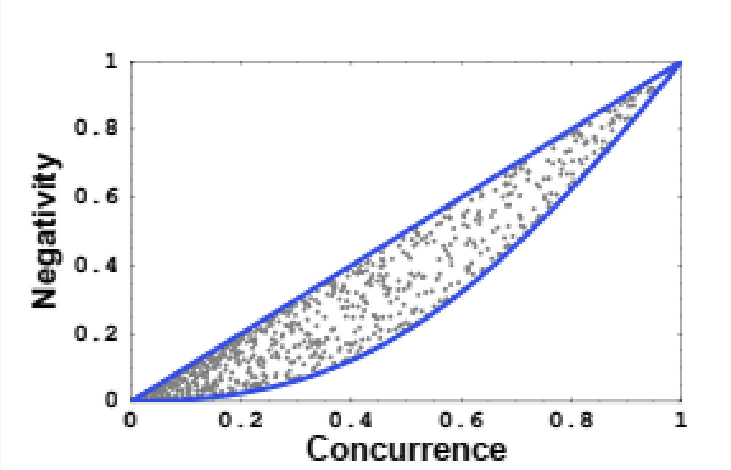
“ *Concurrence vs Negativity*



- Maximum negativity for a given value of concurrence using **Lagrange optimization techniques**.

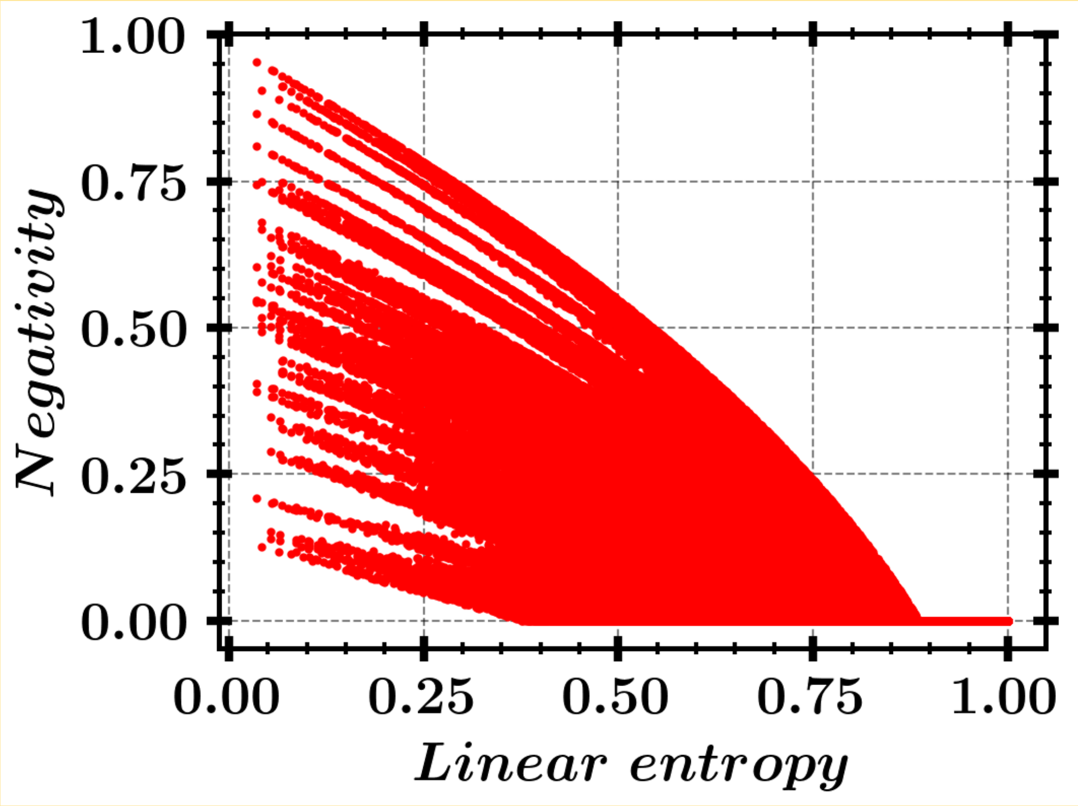
$$N(\rho) = \frac{\|\rho^{T_B}\|_1 - 1}{2}$$

$$\mathcal{L} = N(\rho) + \lambda(C(\rho) - C_0)$$



- F. Verstraete, K. Audenaert, J. Dehaene, and B. De Moor, 2001: F. Verstraete, K. Audenaert, J. Dehaene, and B. De Moor, "A comparison of the entanglement measures negativity and concurrence," Journal of Physics A: Mathematical and General, 34 (47), 10327-10332 (2001)
- Tzu-Chieh Wei et al.: T.-C. Wei, K. Nemoto, P. M. Goldbart, P. G. Kwiat, W. J. Munro, and F. Verstraete, "Maximal entanglement versus entropy for mixed quantum states," Physical Review A, 67 (2), 022110 (2003)

“ *Negativity vs Linear Entropy*



$$S_L(\rho) = \frac{N}{N-1}[1 - Tr(\rho^2)]$$

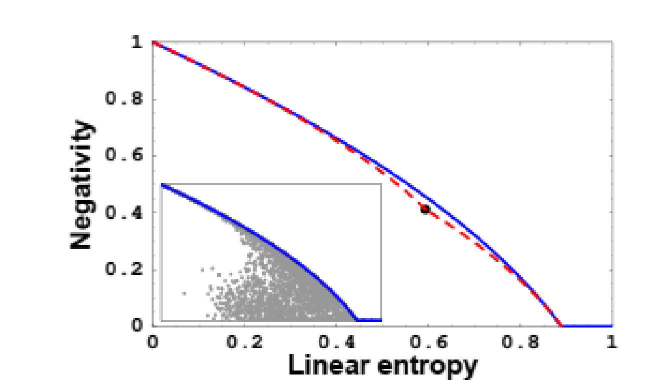
$$N(\rho) = \frac{\|\rho^{T_B}\|_1 - 1}{2}$$

$$\mathcal{L} = N(\rho) + \lambda(S_L(\rho) - S_0)$$

Example (Werner states):

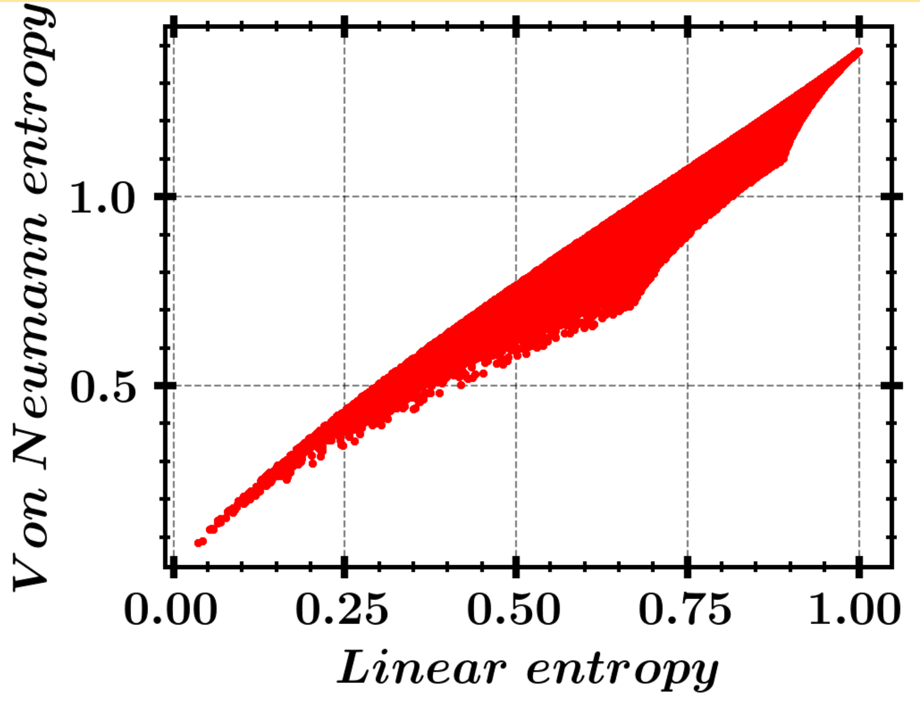
$$\rho_W(P) = P|\Psi^-\rangle\langle\Psi^-| + (1-p)\frac{I}{4}$$

$$S_L(\rho_W(P)) = \frac{3P^2}{2} \quad N(\rho) = Max(0, \frac{P}{2})$$



- Tzu-Chieh Wei et al.: T.-C. Wei, K. Nemoto, P. M. Goldbart, P. G. Kwiat, W. J. Munro, and F. Verstraete, "Maximal entanglement versus entropy for mixed quantum states," Physical Review A, 67 (2), 022110 (2003)

“ Von Neumann Entropy vs Linear Entropy



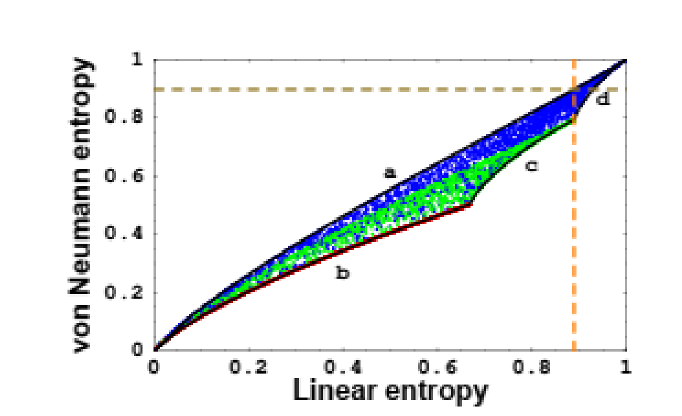
$$S_L(\rho) = \frac{N}{N-1}[1 - Tr(\rho^2)]$$

$$S_{VN}(\rho) = -Tr(\rho Log(\rho))$$

$$\mathcal{L} = -Tr(\rho Log(\rho)) + \lambda(1 - Tr(\rho^2) - S_L)$$

Maximize von Neumann entropy for a fixed linear entropy using Lagrange multipliers

$$\mathcal{L} = -\sum_i \lambda_i Log(\lambda_i) + \lambda(1 - \sum_i {\lambda_i}^2 - S_0)$$



- Tzu-Chieh Wei et al.: T.-C. Wei, K. Nemoto, P. M. Goldbart, P. G. Kwiat, W. J. Munro, and F. Verstraete, "Maximal entanglement versus entropy for mixed quantum states," Physical Review A, 67 (2), 022110 (2003)

Thanks for your attention