Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Transmitter and receiver implementation for a free-space QKD system

Author: Andrea Vanghetti
Instructors: Prof. Giuseppe Vallone, Prof. Nicola Laurenti

A.Y. 2024-2025

1 Introduction

Quantum Random Number Generators (QRNGs) exploit the inherent unpredictability of quantum phenomena to produce sequences of random bits with provable entropy. Unlike classical pseudo-random number generators, which rely on deterministic algorithms, QRNGs are rooted in the fundamental probabilistic nature of quantum mechanics and are essential for applications in cryptography, simulation, and secure communications.

This report presents an experimental comparison of three QRNG protocols with increasing levels of trust assumptions: Trusted, Semi-Device-Independent based on Entropic Uncertainty Principle (SDI-EUP), and Semi-Device-Independent based on Quantum State Tomography (SDI-TOMO). The goal is to evaluate and contrast their performance in terms of min-entropy extraction and output length, under various input states and measurement settings.

The experimental setup is based on a spontaneous parametric down-conversion (SPDC) source using non-linear crystals to generate entangled photon pairs. Polarization controllers and projective measurements are employed to realize and test the different protocols. By measuring coincidences in selected polarization bases and analyzing the resulting statistics, we estimate the entropy of the system and extract randomness using appropriate post-processing techniques.

This study aims to assess the trade-off between security assumptions and randomness generation efficiency, highlighting the practical implications of adopting weaker device assumptions in QRNG protocols.

2 Objective

The main objective of this experiment is to experimentally evaluate and compare the performance of four Quantum Random Number Generation (QRNG) protocols that differ in their level of trust assumptions and measurement requirements:

1. **Trusted QRNG**: assumes full knowledge and control of the source and measurement devices.

2. **Semi-Device-Independent QRNG based on the Entropic Uncertainty Principle (SDI-EUP)** [2]: requires minimal assumptions on the devices, relying on the uncertainty relations of quantum measurements.

3. **Semi-Device-Independent QRNG based on Quantum State Tomography (SDI-TOMO)** [3]: exploits the reconstruction of the density matrix to bound entropy.

4. **Semi-Device-Independent QRNG using Generalized Measurements (SDI-POVM)** [1]: employs Positive Operator-Valued Measures (POVMs) to enhance entropy estimation beyond projective measurements.

These protocols are tested under a variety of input quantum states (e.g., rectilinear, diagonal, and circular polarizations), generated through spontaneous parametric down-conversion and manipulated via polarization optics. By analyzing the statistical properties of the measurement outcomes, we estimate the extractable entropy and compare the protocols in terms of security level, output bit rate, and implementation complexity.

This comparative study aims to provide insight into the practical trade-offs between randomness generation efficiency and the level of device trust required in each QRNG model.

3 Experimental Setup

The experimental setup is based on a continuous-wave (CW) laser emitting horizontally polarized light. A Half Wave Plate (HWP), rotated by $+\frac{\pi}{8}$ from its zero angle, is placed after the laser to transform the polarization state into the diagonal state $|D\rangle$. The beam then passes through a nonlinear beta-barium borate (BBO) crystal, where spontaneous parametric down-conversion (SPDC) occurs, generating pairs of entangled photons that are split into two distinct spatial modes.

Each beam is reflected by a mirror and directed towards a Single-Photon Detector (SPD). On Alice's arm, between the mirror and the SPD, an additional sequence of optical elements is inserted: a HWP, a Quarter Wave Plate (QWP), and a Polarizing Beam Splitter (PBS), enabling projective polarization measurements in various bases. Detection events from the SPDs are processed by a Time-to-Digital Converter (TDC), which records arrival times and allows coincidence detection and statistical analysis.

3.1 State Preparation

Four different quantum states were prepared during the experiment by modifying the optical components along Alice's path:

- **Mixed state:** No additional elements were inserted, and the system remains in the mixed state resulting from SPDC and default polarization (see Figure 1).
- **Diagonal state ($|D\rangle$):** A linear polarizer (POL) is placed before the HWP to prepare the beam in a well-defined $|D\rangle$ state (Figure 7a).
- **Right-circular state ($|R\rangle$):** Starting from the diagonal configuration, a QWP rotated by $+\frac{\pi}{4}$ is inserted after the polarizer to generate the right-circular polarization (Figure 7b).
- **Psi state ($|\psi\rangle$):** For the preparation of the $|\psi\rangle$ state, the same setup of the right-circular state (Figure 7b) is used, but the waveplates are rotated at a random degree of our choice.

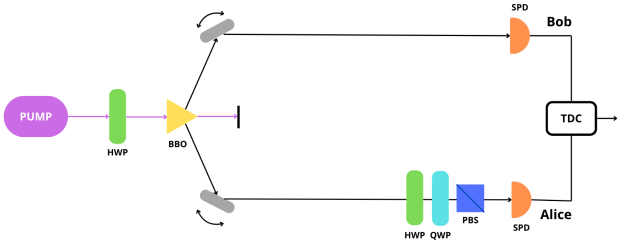


FIGURE 1: Setup for the mixed state.

3.2 Measurements

Polarization measurements are performed by rotating the waveplates to specific angles from their respective zero positions, reported in Table 1.

Waveplate	Zero Position Angle
HWP	36°
QWP	38°

TABLE 1: Reference zero positions of the waveplates.

To perform projective measurements in different polarization bases, the HWP and QWP were rotated accordingly. Table 2 reports the angular settings for each measurement projector.

$\hat{P}$	HWP Angle	QWP Angle
$\|H\rangle\langle H\|$	0	0
$\|V\rangle\langle V\|$	$+\pi/4$	0
$\|D\rangle\langle D\|$	$+\pi/8$	0
$\|A\rangle\langle A\|$	$-\pi/8$	0
$\|R\rangle\langle R\|$	$+\pi/8$	$+\pi/4$
$\|L\rangle\langle L\|$	$-\pi/8$	$-\pi/4$

TABLE 2: Waveplate angles for projective polarization measurements.

4 Results

4.1 Coincidences

To analyze the experimental outcomes, we processed a set of 24 raw `.txt` files, each containing two columns: the first represents the photon *time tags* (with a resolution of approximately 81 ps), and the second indicates the detection channel associated with each event. The files are organized such that each of the four prepared quantum states (*mixed*, $|D\rangle$, $|R\rangle$, and $|\psi\rangle$) is associated with six measurement outcomes, corresponding to projections onto the $\{|H\rangle, |V\rangle, |D\rangle, |A\rangle, |R\rangle, |L\rangle\}$ bases.

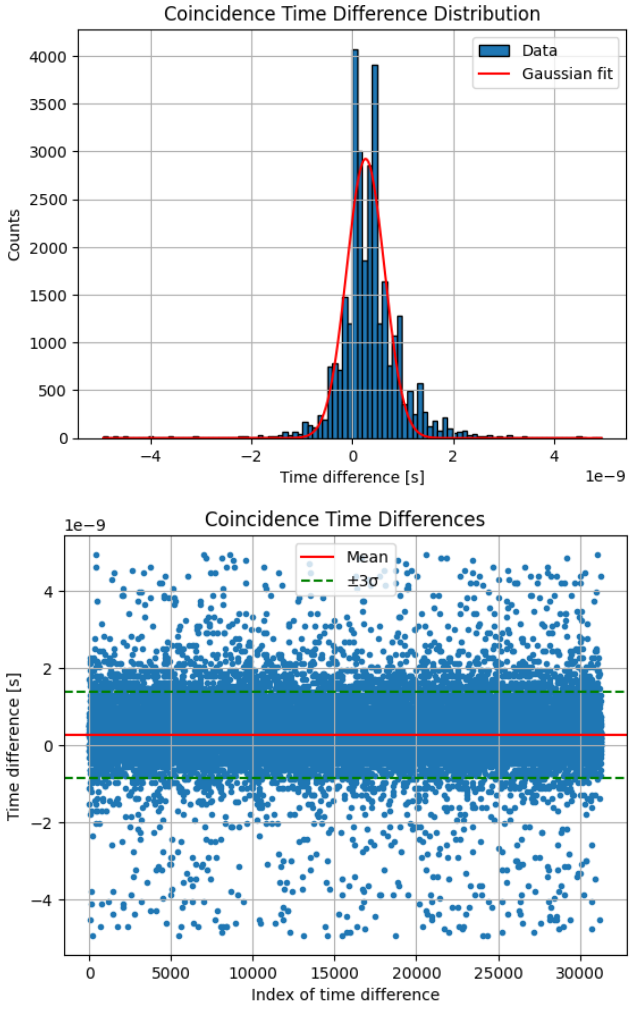


FIGURE 2: Coincidence Time Differences for diagonal prepared state measured on H

In Figure 2 an example of the distributions of time arrivals is shown (more examples are shown in Appendix B). In order to identify valid photon coincidences between the two detection channels, we defined a coincidence time window of $5 \cdot 10^{-10}$ seconds. Only detection events within this time interval on distinct channels were considered as coincidences. Additionally, for consistency across datasets, we selected a fixed time duration of 40 s for all coincidence counts.

During the data analysis, some inconsistencies were observed between the expected and recorded results. In particular, the prepared diagonal and right-circular states yielded a higher number of coincidences in their orthogonal measurement bases (antidiagonal and left-circular, respectively). This anomaly suggested a possible

file mismatch. To address this issue, we verified and swapped the files labeled `diagonal_measured_on_D.txt` with `diagonal_measured_on_A.txt`, and similarly `right_measured_on_R.txt` with `right_measured_on_L.txt`, to ensure alignment between measurement outcomes and state preparations.

The coincidence counts for each prepared state are reported in Tables 3–6.

Basis	Coincidences
$\|H\rangle$	18170
$\|V\rangle$	17609
$\|D\rangle$	34353
$\|A\rangle$	52
$\|L\rangle$	16808
$\|R\rangle$	17839

TABLE 3: Coincidence counts for prepared state $|D\rangle$.

Basis	Coincidences
$\|H\rangle$	9186
$\|V\rangle$	22728
$\|D\rangle$	12143
$\|A\rangle$	19965
$\|L\rangle$	2699
$\|R\rangle$	30258

TABLE 4: Coincidence counts for prepared state *mixed*.

Basis	Coincidences
$\|H\rangle$	18819
$\|V\rangle$	13665
$\|D\rangle$	15476
$\|A\rangle$	15926
$\|L\rangle$	215
$\|R\rangle$	31390

TABLE 5: Coincidence counts for prepared state $|R\rangle$.

Basis	Coincidences
$\|H\rangle$	12932
$\|V\rangle$	31973
$\|D\rangle$	16925
$\|A\rangle$	27831
$\|L\rangle$	3810
$\|R\rangle$	42620

TABLE 6: Coincidence counts for prepared state $|\psi\rangle$.

4.2 Probabilities

To evaluate the probability of each measurement outcome, we compute the conditional probability that a photon is detected in a given polarization state $|\psi\rangle$, using the relation:

$$P_{|\psi\rangle} = \frac{N_{|\psi\rangle}}{N_{|\psi\rangle} + N_{|\psi^{\perp}\rangle}} \tag{1}$$

where $N_{|\psi\rangle}$ is the number of coincidences corresponding to the projection onto state $|\psi\rangle$, and $N_{|\psi^{\perp}\rangle}$ is the number corresponding to its orthogonal complement. This definition assumes binary projective measurements.

The computed probabilities for each measurement basis and prepared quantum state are reported in Tables 7–10.

Basis	Probabilities
$\|H\rangle$	0.508
$\|V\rangle$	0.492
$\|D\rangle$	0.998
$\|A\rangle$	0.002
$\|L\rangle$	0.485
$\|R\rangle$	0.515

TABLE 7: Table for $|D\rangle$ state probabilities

Basis	Probabilities
$\|H\rangle$	0.503
$\|V\rangle$	0.497
$\|D\rangle$	0.493
$\|A\rangle$	0.506
$\|L\rangle$	0.503
$\|R\rangle$	0.497

TABLE 8: Table for *mixed* state probabilities

Basis	Probabilities
$\|H\rangle$	0.579
$\|V\rangle$	0.421
$\|D\rangle$	0.493
$\|A\rangle$	0.507
$\|L\rangle$	0.007
$\|R\rangle$	0.993

TABLE 9: Table for $|R\rangle$ state probabilities

Basis	Probabilities
$\|H\rangle$	0.288
$\|V\rangle$	0.712
$\|D\rangle$	0.378
$\|A\rangle$	0.622
$\|L\rangle$	0.082
$\|R\rangle$	0.918

TABLE 10: Table for $|\psi\rangle$ state probabilities

4.3 Min-Entropies

The min-entropy, denoted H_{∞}, represents the worst-case unpredictability of a random variable. In the context of quantum random number generation, it provides a bound on the amount of extractable randomness that remains secure against an adversary, based on either classical or quantum assumptions about the system.

We consider two scenarios for entropy evaluation: (i) a **trusted device** scenario where all components are assumed ideal and fully characterized, and (ii) a **semi-device-independent** scenario where the source may be untrusted and entropy is bounded via the entropic uncertainty principle.

4.3.1 Trusted QRNG Protocol

The Trusted QRNG protocol assumes full knowledge and control of the quantum system, including the source and mea-

surement apparatus. Under this assumption, if the source emits a pure state ρ_A, the min-entropy can be estimated directly from the maximum observed probability as follows [2]:

$$H_{\min}(P) = -\log_2 \left(\max_i P_i \right) \tag{2}$$

which is equivalent to the common form:

$$H_\infty(X) = -\max_x \left(\log_2 P_x \right) \tag{3}$$

This measure represents the maximal information an adversary could gain about the outcome by guessing the most probable result. Table 11 shows the classical min-entropy values obtained from the experimental probabilities for each prepared quantum state.

Prepared State	$H_{\infty,TRUSTED}$
$\|D\rangle$	0.978
Mixed	0.992
$\|R\rangle$	0.788
$\|\psi\rangle$	0.490

TABLE 11: Classical min-entropy values computed for each prepared state.

4.3.2 Quantum Conditional Min-Entropy via the Entropic Uncertainty Principle

In scenarios where the source is not fully trusted, we use the framework of quantum conditional min-entropy to evaluate the amount of extractable randomness that remains secure even in the presence of a potential adversary with side information (e.g., Eve). The entropic uncertainty relation provides a lower bound for this quantity.

Let $\mathbb{X}$ and $\mathbb{Z}$ be two mutually unbiased bases (MUBs), such as the $\{|H\rangle, |V\rangle\}$ and $\{|D\rangle, |A\rangle\}$ bases. The uncertainty principle in terms of min- and max-entropies reads [2]:

$$H_{\min}(Z|E)_\rho + H_{\max}(X|B)_\rho \geq \log_2 d \tag{4}$$

For dimension $d = 2$, the inequality simplifies to:

$$H_{\min}(Z|E) \geq 1 - H_{\max}(X) \tag{5}$$

where the max-entropy $H_{\max}(X)$ (also known as Rényi entropy of order 1/2) is defined as:

$$H_{\max}(X) = 2\log_2 \left(\sum_x \sqrt{P_x} \right) \tag{6}$$

This bound quantifies the minimum unpredictability of the measurement outcome Z, given Eve's knowledge. The results obtained for each state are shown in Table 12.

Prepared State	$H_{\min,EUP}(Z\|E)$
$\|D\rangle$	0.892
Mixed	$76.16 \cdot 10^{-5}$
$\|R\rangle$	0.009
$\|\psi\rangle$	0.070

TABLE 12: Quantum conditional min-entropy values using the entropic uncertainty principle.

These values demonstrate the significantly reduced entropy estimates when trust in the source is relaxed. In particular, states that appear highly random under the trusted model (e.g., $|D\rangle$ or $|L\rangle$) exhibit very low entropy bounds when assessed under the entropic uncertainty framework. This highlights the importance of trust assumptions in QRNG implementations and the trade-off between device independence and extractable randomness.

4.3.3 Quantum Conditional Min-Entropy via the Full Tomography Method

An alternative method for estimating the quantum conditional min-entropy relies on quantum state tomography. In this approach, the density matrix ρ of the quantum state is reconstructed from the measured Stokes parameters, and the min-entropy is computed directly from the state's purity.

According to the method described in [3], the min-entropy is given by:

$$H_\infty(\hat{\rho}) = -\log_2 \left(\frac{1 + \sqrt{1 - |S_1 - iS_2|^2}}{2} \right) \tag{7}$$

where S_1 and S_2 are components of the Stokes vector, defined in terms of normalized intensities $I_{|\psi\rangle}$ as follows:

$$S_0 = I_H + I_V \tag{8}$$
$$S_1 = I_H - I_V \tag{9}$$
$$S_2 = I_D - I_A \tag{10}$$
$$S_3 = I_R - I_L \tag{11}$$

Each $I_{|\psi\rangle}$ is the number of coincidences measured in the corresponding polarization basis, normalized with respect to S_0 for that state.

The Stokes parameters calculated for each prepared state are summarized in Table 13.

Prepared State	S_0	S_1	S_2	S_3
$\|D\rangle$	1	0.016	0.997	0.030
Mixed	1	0.005	-0.013	-0.005
$\|R\rangle$	1	0.159	-0.014	0.986
$\|\psi\rangle$	1	-0.424	-0.244	0.836

TABLE 13: Stokes parameters for each prepared state.

Substituting the values of S_1 and S_2 into Eq. (7), we obtain the min-entropy values reported in Table 14.

Prepared State	$H_{\infty,TOMO}$
$\|D\rangle$	0.894
Mixed	$7.18 \cdot 10^{-5}$
$\|R\rangle$	0.423
$\|\psi\rangle$	0.777

TABLE 14: Quantum conditional min-entropy values via full tomography.

The reconstructed density matrix ρ for each state is derived using the relation:

$$\rho = \frac{1}{2} \begin{bmatrix} 1 + S_3 & S_1 - iS_2 \\ S_1 + iS_2 & 1 - S_3 \end{bmatrix} \tag{12}$$

The explicit density matrices obtained from the experimental data are:

- $\rho_{|D\rangle}$:

$$\begin{bmatrix} 0.51488 + i0 & 0.00784 + i0.49849 \\ 0.00784 - i0.49849 & 0.48512 + i0 \end{bmatrix}$$

- ρ_{mixed}:

$$\begin{bmatrix} 0.49749 + i0 & 0.00268 + i0.00653 \\ 0.00268 - i0.00653 & 0.50251 + i0 \end{bmatrix}$$

- $\rho_{|\psi\rangle}$:

$$\begin{bmatrix} 0.91811 + i0 & -0.21216 + i0.12181 \\ -0.21216 - i0.12181 & 0.08189 + i0 \end{bmatrix}$$

- $\rho_{|R\rangle}$:

$$\begin{bmatrix} 0.99320 + i0 & 0.07933 - i0.00717 \\ 0.07933 + i0.00717 & 0.00680 + i0 \end{bmatrix}$$

Each matrix satisfies the conditions for physical density operators:

- $\rho = \rho^\dagger$ (Hermiticity),
- $\rho \geq 0$ (positive semidefinite),
- $\text{Tr}(\rho) = 1$ (normalized trace).

These results validate the tomographic reconstruction process and confirm the quantum physicality of the prepared states. The min-entropy values obtained using this method align closely with theoretical expectations, while revealing the impact of coherence and purity on randomness extraction potential.

4.3.4 Quantum Conditional Min-Entropy via POVM Measurements

As demonstrated in [1], the use of Positive Operator-Valued Measures (POVMs) can enhance the amount of extractable randomness in a QRNG protocol. In particular, the implementation of symmetric informationally complete measurements allows for tighter entropy bounds under minimal assumptions.

In our analysis, we consider two POVM configurations:

- A 4-element POVM (F^4), defined on the $\{|H\rangle, |V\rangle, |D\rangle, |A\rangle\}$ basis:

$$F^4 = \left\{ \frac{1}{2} |k\rangle \langle k| \right\}_{k=H,V,D,A}$$

- A 6-element POVM (F^6), defined on the full polarization set $\{|H\rangle, |V\rangle, |D\rangle, |A\rangle, |R\rangle, |L\rangle\}$:

$$F^6 = \left\{ \frac{1}{3} |k\rangle \langle k| \right\}_{k=H,V,D,A,R,L}$$

The min-entropy in this context is computed from the guessing probability P_g, which quantifies the maximum probability that an adversary correctly guesses the measurement outcome. According to [1], this probability can be estimated using:

$$P_g = \frac{1}{N} + \frac{1}{N} \sum_k f_N(\vec{r} \cdot \vec{u}_k, \alpha)\, \theta\left(\vec{r} \cdot \vec{u}_k - \cos\alpha\right) \qquad (13)$$

Prepared State	$P_{g,\text{4-POVM}}$	$P_{g,\text{6-POVM}}$
$\|A\rangle$	0.500	0.33333
Mixed	0.500	0.33333
$\|L\rangle$	0.500	0.33332
$\|\psi\rangle$	0.500	0.33254

TABLE 15: Guessing probabilities for each prepared state using 4-POVM and 6-POVM.

Prepared State	$H_{\infty,\text{4-POVM}}$	$H_{\infty,\text{6-POVM}}$
$\|D\rangle$	1.000	1.5850
Mixed	1.000	1.5850
$\|R\rangle$	1.000	1.5850
$\|\psi\rangle$	1.000	1.5884

TABLE 16: Min-entropy values for each prepared state using POVM-based analysis.

where: - $f_N(x, \alpha) = x\cos\alpha + \sqrt{1-x^2}\sin\alpha$; - $\theta(\cdot)$ is the Heaviside function; - $\vec{r}$ is the Bloch vector of the reconstructed state; - $\vec{u}_k$ are the POVM direction vectors; - $\alpha = \frac{\pi}{4}$ for the 4-POVM and $\alpha = \arccos\left(\frac{1}{\sqrt{3}}\right)$ for the 6-POVM.

When the point $\vec{r}$ lies inside the geometry defined by the POVM (a square in the ZX plane for F^4, and an octahedron in the Bloch sphere for F^6), the guessing probability reaches the lower bound:

$$P_g = \frac{2}{N}$$

However, during the analysis we observed cases where $\vec{r}$ lies outside the POVM geometry. When $\vec{r}$ lies outside multiple faces of the 6-POVM octahedron, the formula in Eq. (13) may return an invalid (overestimated) value for P_g. Since no analytical correction is yet available for this condition, we conservatively adopt the worst-case scenario, i.e., the maximum obtained guessing probability.

Figures 3 and 4 show the positions of the Bloch vector $\vec{r}$ with respect to the POVM geometry for each prepared state. Table 15 summarizes the resulting guessing probabilities, and Table 16 reports the corresponding min-entropy values.

These results demonstrate that the use of generalized measurements (POVMs) can significantly increase the extractable randomness compared to standard projective measurement models. In particular, the 6-POVM configuration consistently yields higher min-entropy values due to its increased symmetry and information completeness.

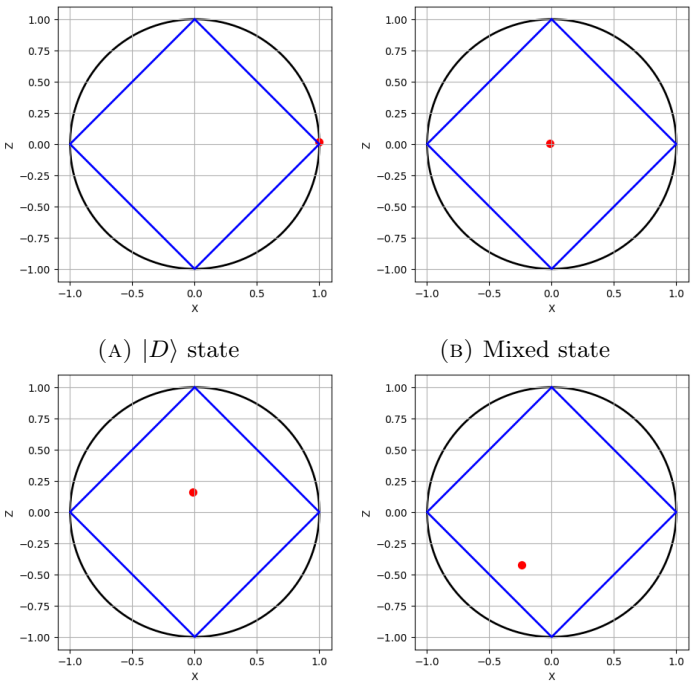
5 Randomness Extractor

5.1 Methodology

Having estimated the min-entropy for each protocol and prepared quantum state, we are now able to extract a secure string of random bits using a seeded randomness extractor.

Bit assignment. For each prepared state, we begin by assigning a binary value to every detection event based on the measurement outcome. The following convention is used:

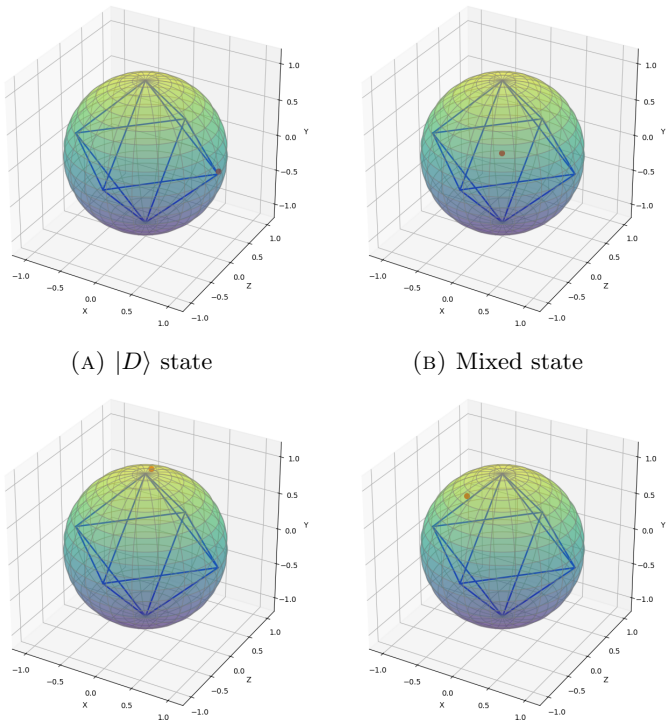
$$|H\rangle, |D\rangle, |R\rangle \to 0$$
$$|V\rangle, |A\rangle, |L\rangle \to 1$$



(A) $|D\rangle$ state (B) Mixed state

(C) $|R\rangle$ state (D) $|\psi\rangle$ state

FIGURE 3: 4-POVM square (blue) and Bloch vectors $\vec{r}$ (red).



(A) $|D\rangle$ state (B) Mixed state

(C) $|R\rangle$ state (D) $|\psi\rangle$ state

FIGURE 4: 6-POVM octahedron (blue) and Bloch vectors $\vec{r}$ (red).

The binary strings corresponding to each measurement basis are then merged together and sorted by their associated photon arrival time to generate the raw bit string r.

Extraction process. To extract uniform random bits from the non-uniform string r, we use a seeded extractor based on multiplication with a *Toeplitz matrix* T. The secure output string s is computed as:

$$s = T \cdot r \tag{14}$$

Security analysis via Leftover Hashing Lemma. The security of the extracted string is quantified using the Leftover Hashing Lemma [4], which relates the min-entropy of the input to the security parameter Δ and the output length ℓ. Specifically:

$$\Delta = \frac{1}{2}\sqrt{2^{\ell - H_{\min}}} \tag{15}$$

This parameter Δ defines the statistical distance between the extracted string and an ideal uniform string that is completely independent of any adversary's knowledge.

Alternatively, for a chosen level of security Δ, we can compute the maximum allowed output length as:

$$\ell = \left\lfloor H_{\min}(X|E) - 2\log_2 \frac{1}{\Delta} + 2 \right\rfloor \tag{16}$$

Performance visualization. Figure 5 shows the relationship between the output length ℓ and the desired security parameter Δ, while Figure 6 displays the corresponding bitrate (i.e., output bits per input bit).

5.2 Randomness Extraction Results

As a case study, we apply the randomness extractor to the raw data corresponding to the **diagonal** prepared state. The extraction was performed using a custom Python implementation (see Appendix A for code details).

From Figure 5, we observe that as the security parameter Δ becomes smaller (i.e., the requirement for secrecy becomes stronger), the output length ℓ decreases, as expected. Likewise, in Figure 6, the bitrate also reduces with increasing security requirements.

Consistent with the findings in [1], the 4-POVM and 6-POVM protocols achieve the highest extraction performance across all security levels, producing longer output strings with lower Δ. The tomography-based approach (SDI-TOMO) performs slightly better than the SDI-EUP scenario but remains less efficient than the Trusted model. These trends highlight the fundamental trade-offs between device trust assumptions and achievable randomness throughput.

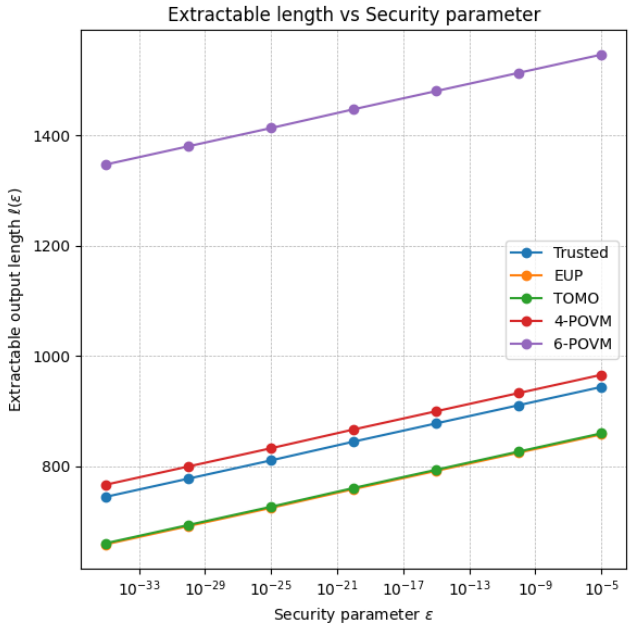


FIGURE 5: Output length ℓ as a function of security parameter Δ.

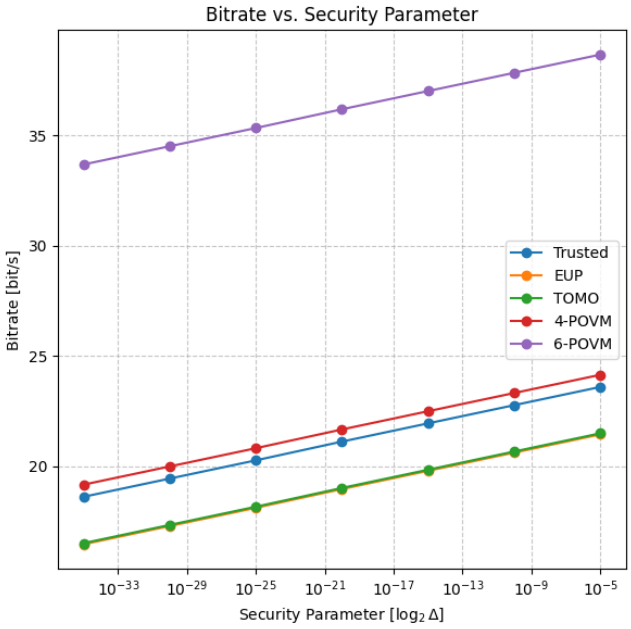


FIGURE 6: Bitrate as a function of security parameter Δ.

References

[1] Massimiliano Avesani, Hadi Tebyanian, Paolo Villoresi, et al. Unbounded randomness from uncharacterized sources. Communications Physics, 5:273, 2022.

[2] Harold Ollivier, Jean-Pierre Tillich, and Robert Zeier. Quantum simplicial complexes and the coloring problem. Physical Review A, 90:052327, 2014.

[3] Mary Beth Ruskai, Stanislaw Szarek, and Elisabeth Werner. An analysis of completely-positive trace-preserving maps on m_2. Physical Review A, 75:032334, 2007.

[4] Marco Tomamichel, Christian Schaffner, Adam Smith, and Renato Renner. Leftover hashing against quantum side information. IEEE Transactions on Information Theory, 2011.

A Randomness Extractor Code

```
    import pandas as pd
import numpy as np
from pathlib import Path
import math
from scipy.linalg import toeplitz
from typing import Sequence, Union, Optional

################################################################################

### FUNCTIONS ###

def generate_toeplitz_matrix(
    n: int,
    m: int,
    *,
    random_state: Optional[Union[int, np.random.Generator]] = None
) -> np.ndarray:
    # Set up RNG
    if isinstance(random_state, (int, np.integer)):
        rng = np.random.default_rng(random_state)
    elif isinstance(random_state, np.random.Generator):
        rng = random_state
    else:
        rng = np.random.default_rng()

    # first column and first row
    c0 = rng.integers(0, 2, size=n, dtype=np.uint8)
    r0 = rng.integers(0, 2, size=m, dtype=np.uint8)
    # enforce consistency at (0,0)
    r0[0] = c0[0]

    # build Toeplitz
    toepl = toeplitz(c0, r0).astype(np.uint8)
    return toepl

def extract_random_bits(
    raw_bits: Sequence[int],
    toeplitz_matrix: np.ndarray
) -> np.ndarray:
    raw = np.asarray(raw_bits, dtype=np.uint8)
    _, m = toeplitz_matrix.shape
    if raw.ndim != 1 or raw.size != m:
        raise ValueError(f"raw_bits must be length {m}, got {raw.size}")
    # matrix multiplication mod 2
    product = toeplitz_matrix.dot(raw)
    return np.mod(product, 2).astype(np.uint8)

def max_extracted_length(
    h_min: float,
    data_length: int,
    epsilon: float
) -> int:
    if not (0 < epsilon < 1):
        raise ValueError("epsilon must be in (0,1)")
    raw_entropy = h_min * data_length
    subtract = 2 * math.log2(1 / epsilon)
    l = math.floor(raw_entropy - subtract)
    if l < 0:
        raise ValueError("Parameters yield negative extractable length")
    return l
```

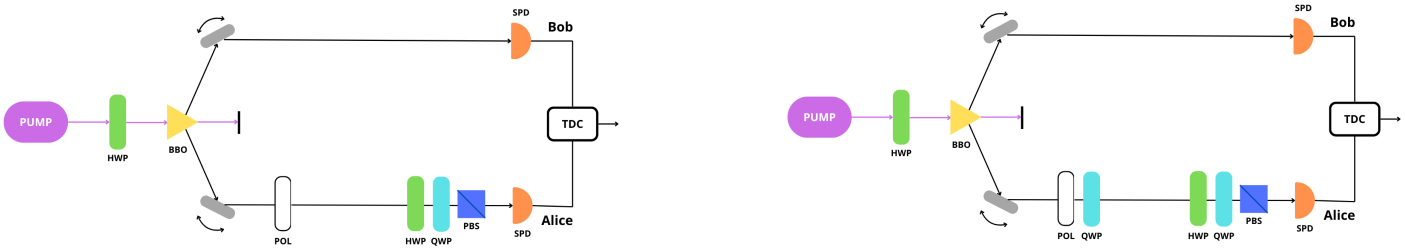
```python
def dataframe_creation(filename, resolution, duration):
    df = pd.read_csv(filename, delimiter=';', comment='#', names=['Time','Channel'], header=None)
    df['Time'] -= df['Time'][0]
    df['Time'] = df['Time'] * resolution
    df = df[df['Time'] <= duration]
    return df

def find_coincidence(df, delta_mean, delta_std, coincidences_num, coincidence_window):

    # Create two different arrays for 'Time' and 'Channel'
    # time_tags = df['Time']
    # channels = df['Channel']

    channel_changes = df["Channel"] != df["Channel"].shift()
    delta_time = df.loc[channel_changes, "Time"].diff()
    delta_time = (delta_time * df["Channel"].apply(lambda x: 1 if x == 3 else -1)).dropna()

    # Calculate mean and std of
    delta_mean.append(np.mean(delta_time))
    delta_std.append(np.std(delta_time))

    coincidence_mask = (delta_time >= -coincidence_window) & (delta_time <= coincidence_window)

    coincidences_num.append(coincidence_mask.sum())

    # Extract filtered time tags based on the coincidence mask
    filtered_time_tags = df.loc[df.index[channel_changes], "Time"][1:][coincidence_mask]

    # Return results as a DataFrame
    return filtered_time_tags.to_frame(name="Coinciding Time Tags")

def find_coincidence_for_list(file_list,
                              df_names,
                              resolution,
                              coincidence_window,
                              duration = 60):

    df_list = {}

    for file, df_name in zip(file_list, df_names):
        print(f"Creating dataframe for {file}")
        df_list[df_name] = dataframe_creation(file, resolution, duration)

    # Define the output list
    coincidences_list = {}

    # Define the output vectors
    delta_mean = []
    delta_std = []
    coincidences_num = []

    # Iterate through the DataFrames (values in df_list)
    for df_name, df in df_list.items():
        print(f"Searching for coincidences on in dataframe {df_name}")
        timed_coincidences = find_coincidence(df,
                                              delta_mean,
                                              delta_std,
                                              coincidences_num,
                                              coincidence_window=coincidence_window)
        coincidences_list[df_name] = timed_coincidences

    return coincidences_list, coincidences_num, delta_mean, delta_std
```

```
##########################################################################

### CONSTANTS ###
RESOLUTION = 80.955e-12  # [ps]
DURATION = 40.0 #60.0  # Max window of duration (s)
COINCIDENCE_WINDOW = 5e-10  # Window of coincidences in s
RAW_DATA_LENGTH = 1000

### FOLDERS ###
BASE_DIR = Path.cwd()
DATA_PATH = BASE_DIR / 'data'

### FILE'S NAME ###
projectors = ['H', 'V', 'D', 'A', 'L', 'R']
generated_states = ['diagonal'] #, 'mixed', 'psi', 'right']

files = []
df_names = []
for generated_state in generated_states:
    for projector in projectors:
        filename = f'{generated_state}_measured_on_{projector}.txt'
        files.append(DATA_PATH / filename)
        df_names.append(f'{generated_state}_on_{projector}')

coincidences_list, coincidences_num, _, _ = find_coincidence_for_list(files,
                                                                      df_names,
                                                                      RESOLUTION,
                                                                      COINCIDENCE_WINDOW
                                                                      )

prepared_state = 'diagonal'

# Encoding the outcomes
coincidences_list[f'{prepared_state}_on_H']['Measure'] = '0'
coincidences_list[f'{prepared_state}_on_V']['Measure'] = '1'
coincidences_list[f'{prepared_state}_on_D']['Measure'] = '0'
coincidences_list[f'{prepared_state}_on_A']['Measure'] = '1'
coincidences_list[f'{prepared_state}_on_R']['Measure'] = '0'
coincidences_list[f'{prepared_state}_on_L']['Measure'] = '1'

# Merging encoded data and sorting by Time Tags
merged_dataframe_1 = pd.concat([coincidences_list[f'{prepared_state}_on_H'],
                                coincidences_list[f'{prepared_state}_on_V']
                                ])
merged_dataframe_1.sort_values(by='Coinciding Time Tags',
                               inplace=True,
                               ascending=True)

merged_dataframe_2 = pd.concat([coincidences_list[f'{prepared_state}_on_H'],
                                    coincidences_list[f'{prepared_state}_on_V'],
                                    coincidences_list[f'{prepared_state}_on_D'],
                                    coincidences_list[f'{prepared_state}_on_A']
                                    ])
merged_dataframe_2.sort_values(by='Coinciding Time Tags',
                                   inplace=True,
                                   ascending=True)

merged_dataframe_3 = pd.concat([coincidences_list[f'{prepared_state}_on_H'],
                                    coincidences_list[f'{prepared_state}_on_V'],
                                    coincidences_list[f'{prepared_state}_on_D'],
                                    coincidences_list[f'{prepared_state}_on_A'],
                                    coincidences_list[f'{prepared_state}_on_R'],
```

```
                                        coincidences_list[f'{prepared_state}_on_L']
                                        ])
merged_dataframe_3.sort_values(by='Coinciding Time Tags',
                                       inplace=True,
                                       ascending=True)

raw_data_1 = np.array(merged_dataframe_1['Measure'])
raw_data_1 = np.fromiter(''.join(raw_data_1), dtype=int)

raw_data_2 = np.array(merged_dataframe_2['Measure'])
raw_data_2 = np.fromiter(''.join(raw_data_2), dtype=int)

raw_data_3 = np.array(merged_dataframe_3['Measure'])
raw_data_3 = np.fromiter(''.join(raw_data_3), dtype=int)

raw_data_short_1 = raw_data_1[:RAW_DATA_LENGTH]
raw_data_short_2 = raw_data_2[:RAW_DATA_LENGTH]
raw_data_short_3 = raw_data_3[:RAW_DATA_LENGTH]

# Min-entropies
min_entropies = {
    'classical_Hm': 0.978,
    'quantum_Hm': 0.892,
    'quantum_tomo_Hm': 0.894,
    'POVM_square_Hm': 1,
    'POVM_octhaedron_Hm': 1.58,
}

raw_data = [
    raw_data_short_1,
    raw_data_short_2,
    raw_data_short_3,
    raw_data_short_2,
    raw_data_short_3
]

epsilon_array = [
    1e-5,
    1e-10,
    1e-15,
    1e-20,
    1e-25,
    1e-30,
    1e-35
    ]

output_length_vector = []
for H_min, data in zip(list(min_entropies.values()), raw_data):
    output_length_vector.append(max_extracted_length(H_min, len(data), 1e-6))

outputs = []
for i, (output_length, data) in enumerate(zip(output_length_vector, raw_data)):

    toeplitz_matrix = generate_toeplitz_matrix(output_length, len(data))
    extracted_bits = extract_random_bits(data, toeplitz_matrix)
    outputs.append(extracted_bits)

# Etichette leggibili per ciascun dataset
labels = [
    "Classical",
    "Quantum",
    "Quantum Tomography",
    "Square POVM",
```

```
    "Octahedron POVM"
]

# Stampa ordinata dei risultati
for label, output in zip(labels, outputs):
    print(f'{label:<30}: {output}')
```

B Plots



(A) Setup for $|D\rangle$. (B) Setup for $|R\rangle$.

FIGURE 7: Optical setups used for the preparation of the states $|D\rangle$ and $|R\rangle$.

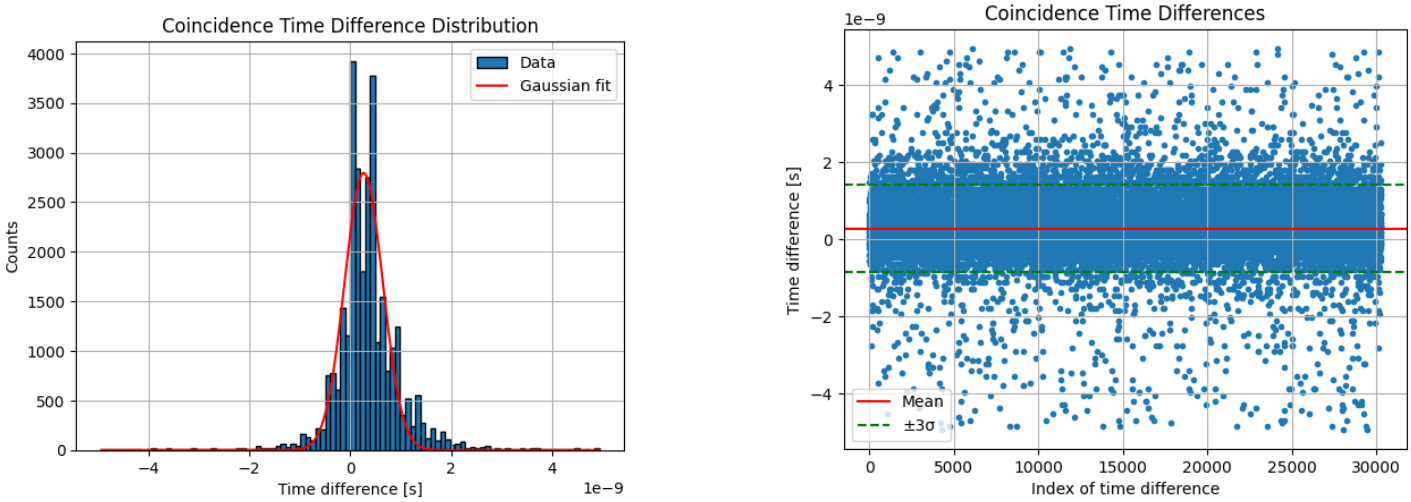


FIGURE 8: Coincidence time differences for $|D\rangle$ prepared state measured on $|V\rangle$

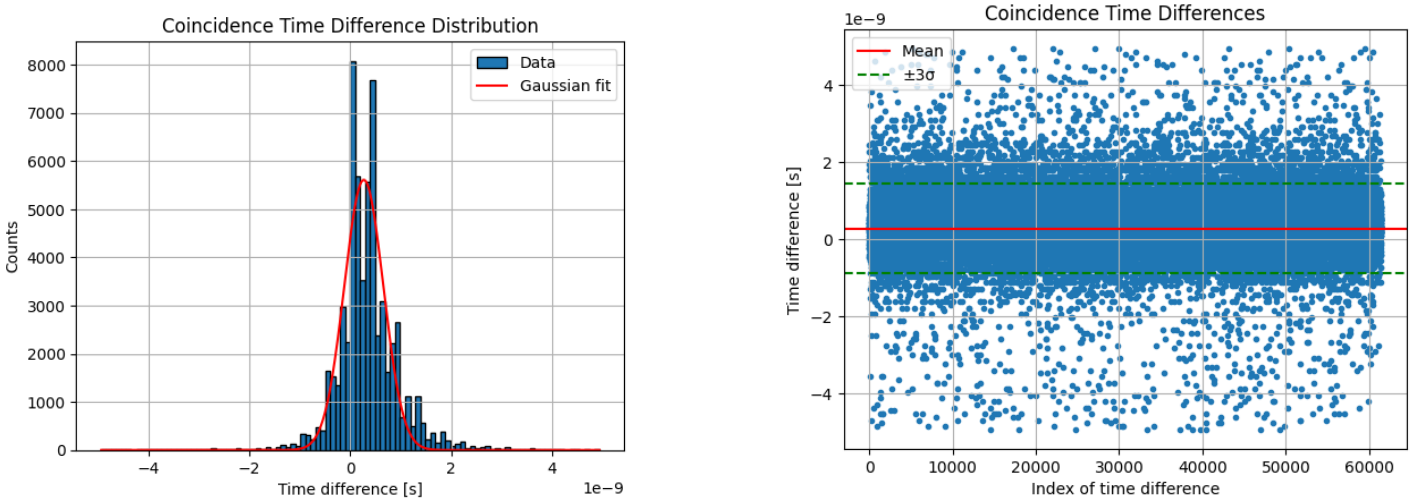


FIGURE 9: Coincidence time differences for $|D\rangle$ prepared state measured on $|D\rangle$

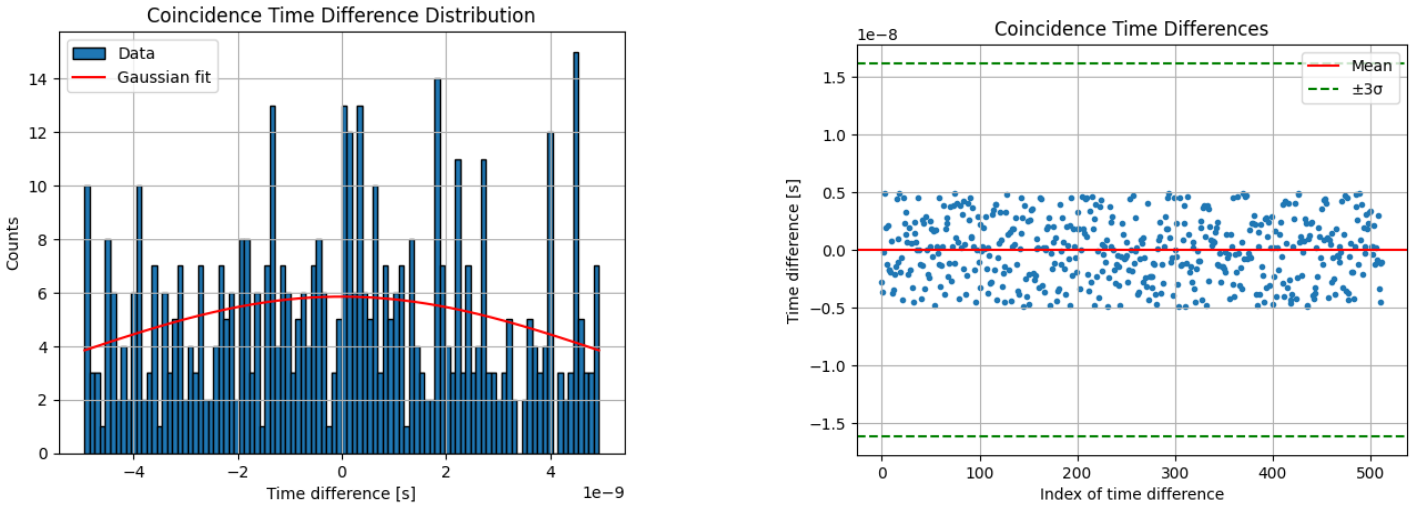


FIGURE 10: Coincidence time differences for $|D\rangle$ prepared state measured on $|A\rangle$

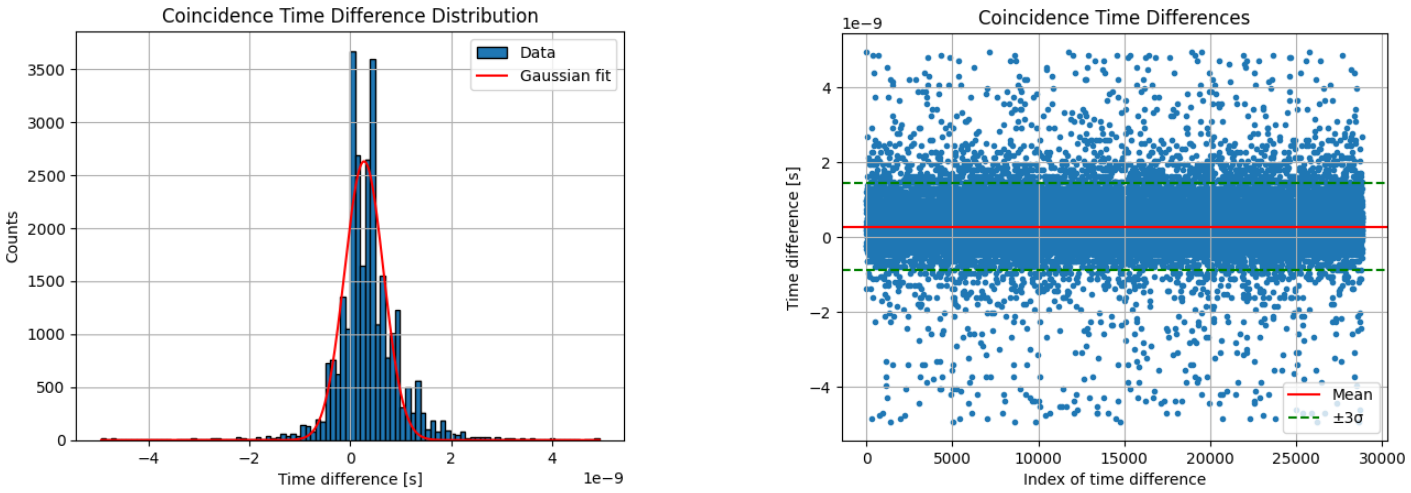


FIGURE 11: Coincidence time differences for $|D\rangle$ prepared state measured on $|R\rangle$

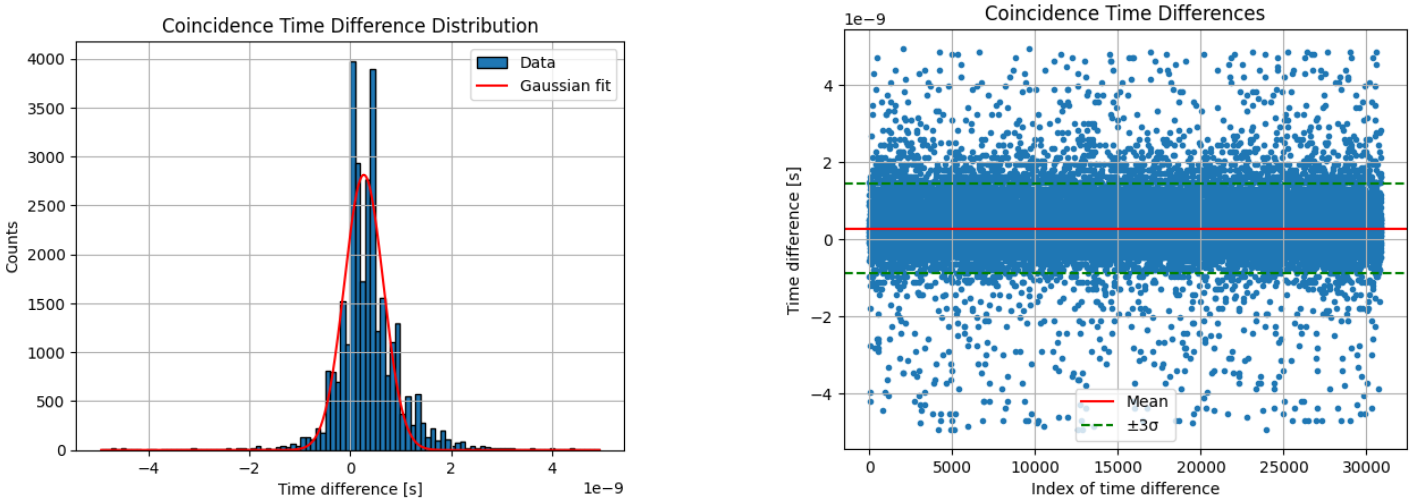
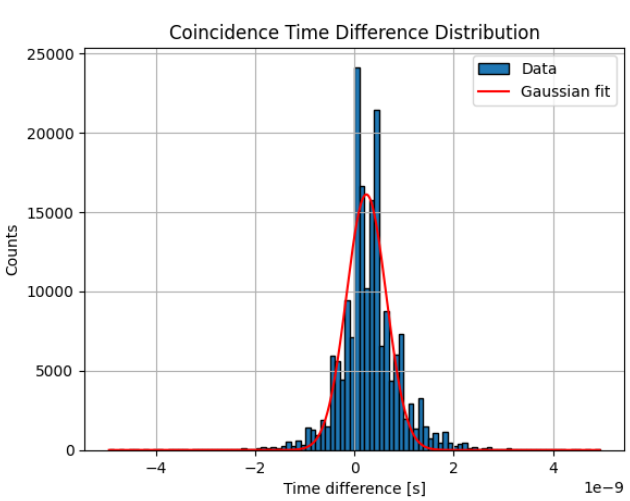


FIGURE 12: Coincidence time differences for $|D\rangle$ prepared state measured on $|L\rangle$



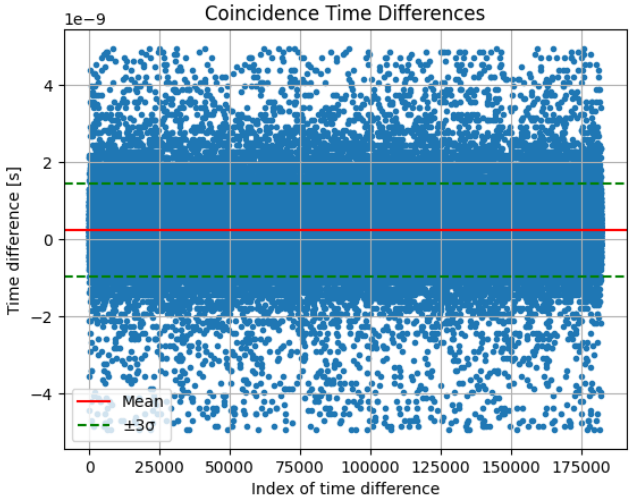
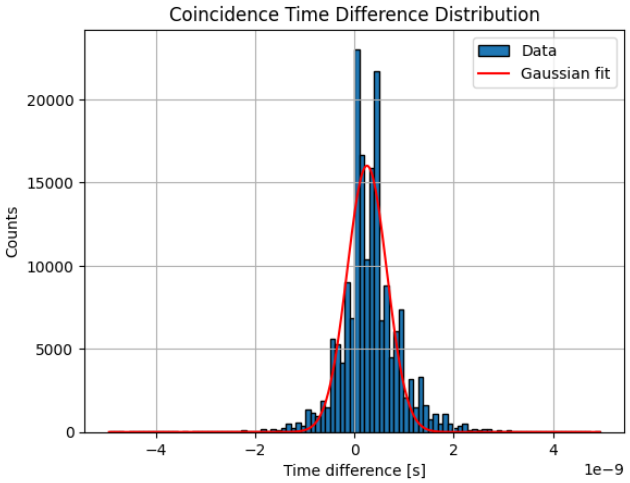


FIGURE 13: Coincidence time differences for *mixed* prepared state measured on $|H\rangle$



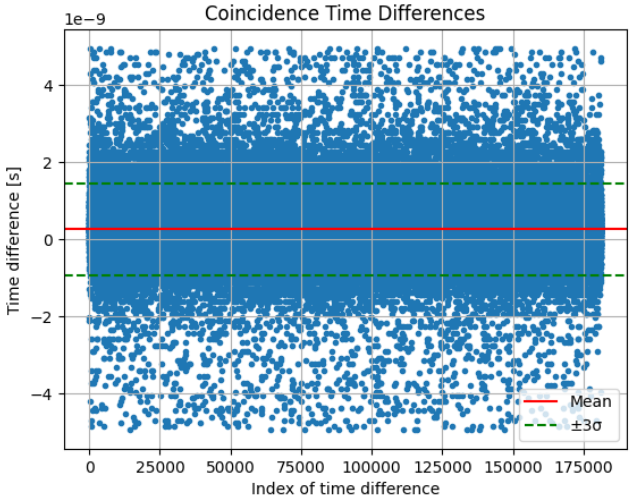
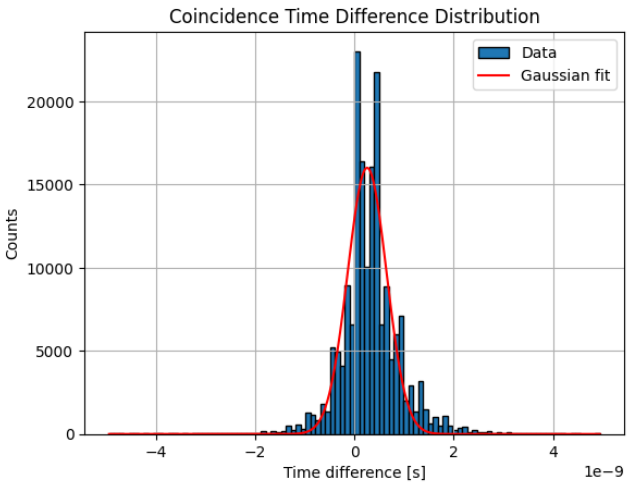


FIGURE 14: Coincidence time differences for *mixed* prepared state measured on $|V\rangle$



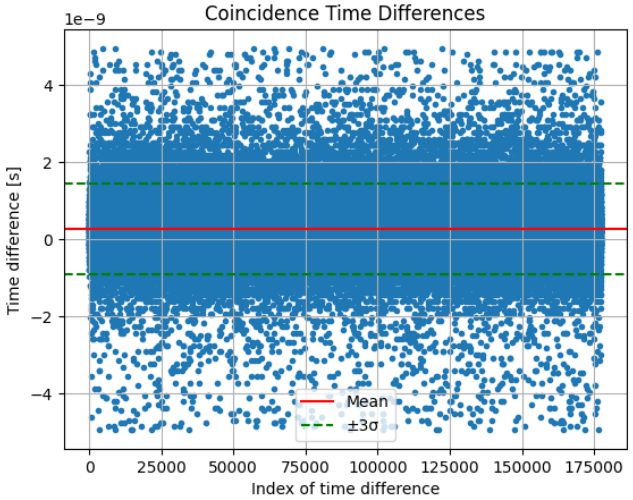


FIGURE 15: Coincidence time differences for *mixed* prepared state measured on $|D\rangle$

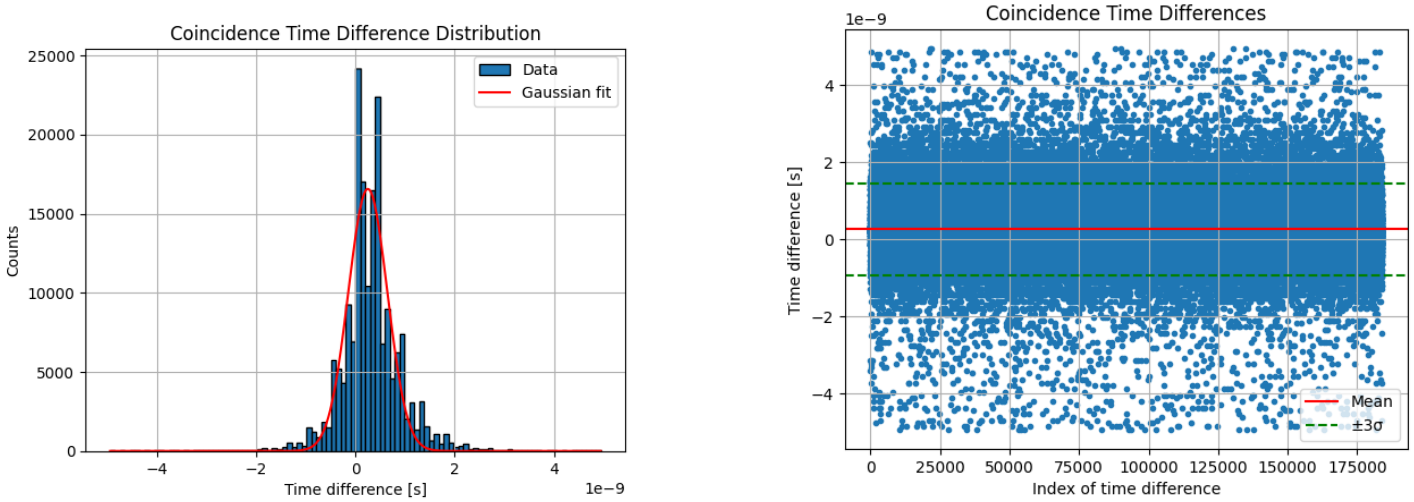


FIGURE 16: Coincidence time differences for *mixed* prepared state measured on $|A\rangle$

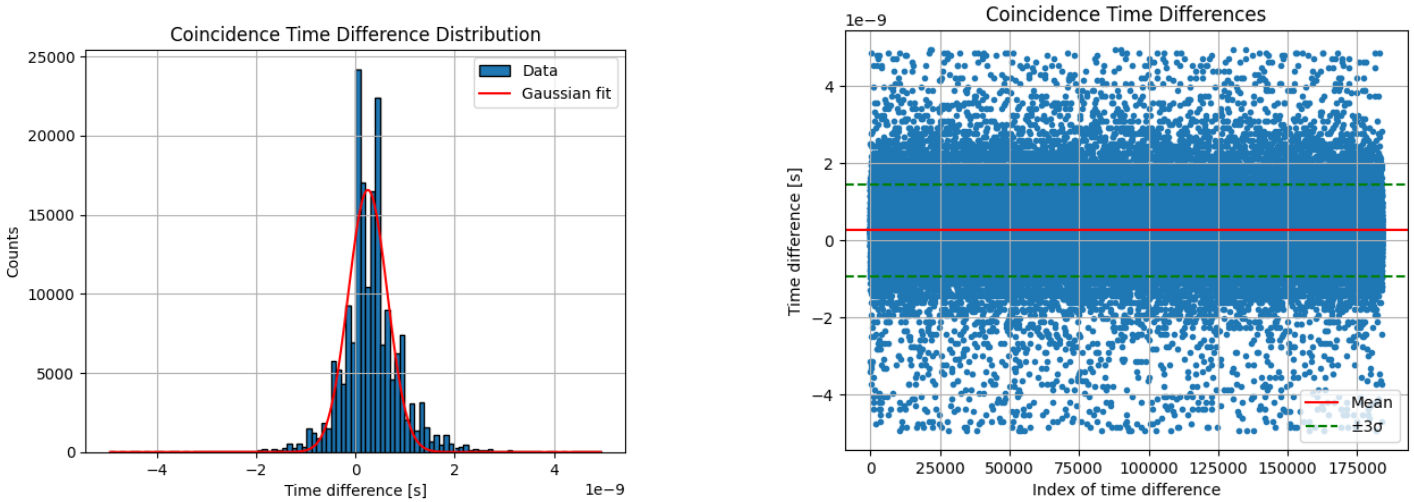


FIGURE 17: Coincidence time differences for *mixed* prepared state measured on $|R\rangle$

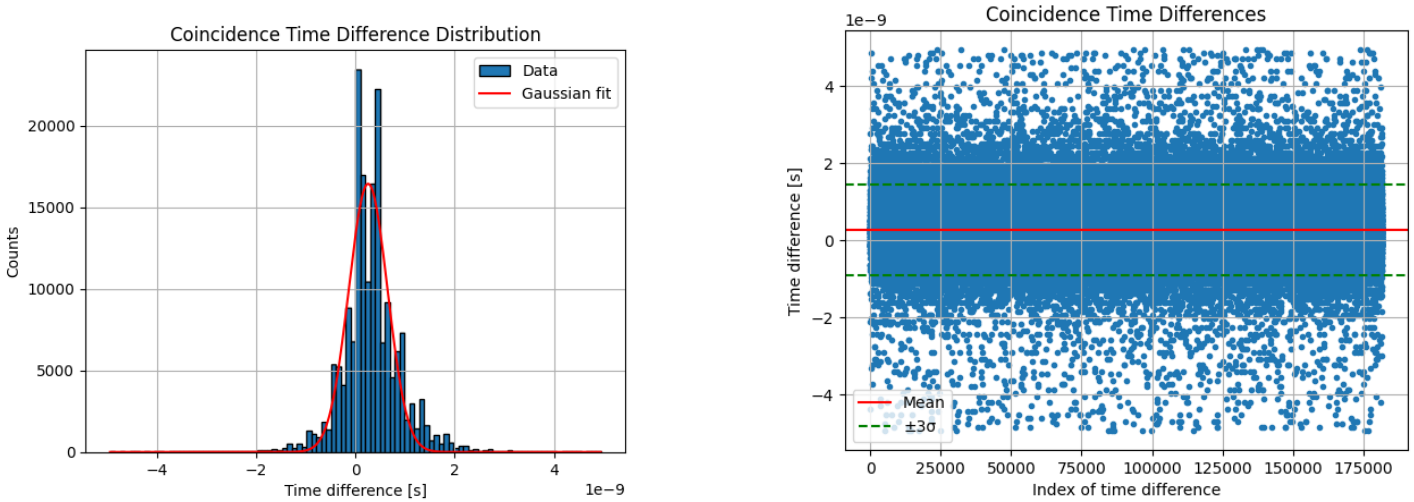
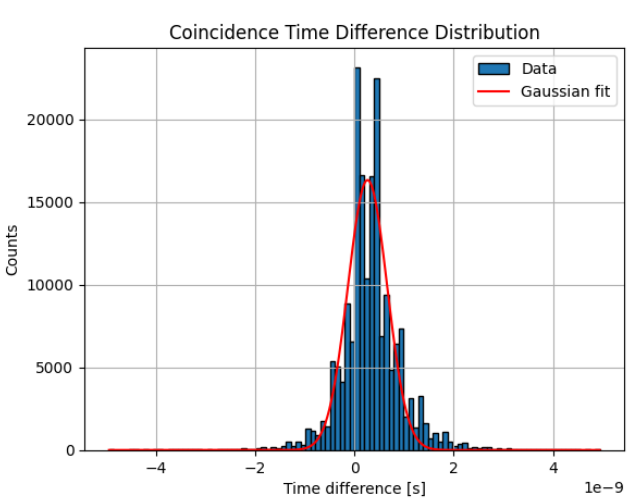


FIGURE 18: Coincidence time differences for *mixed* prepared state measured on $|L\rangle$



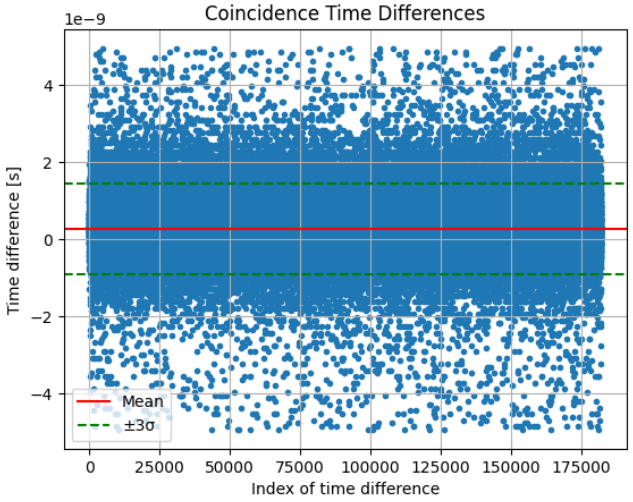
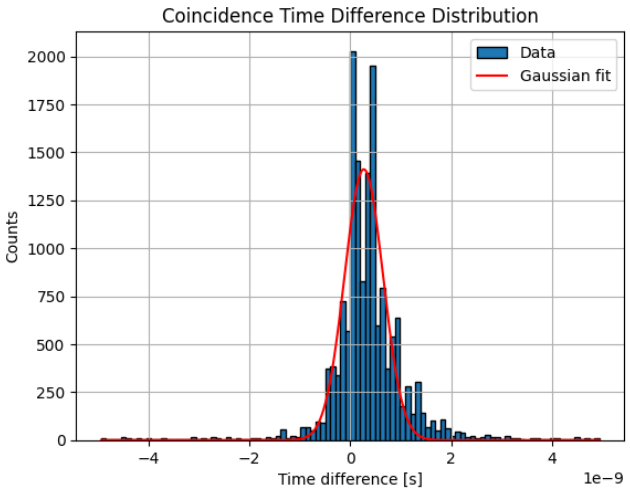


FIGURE 19: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|H\rangle$



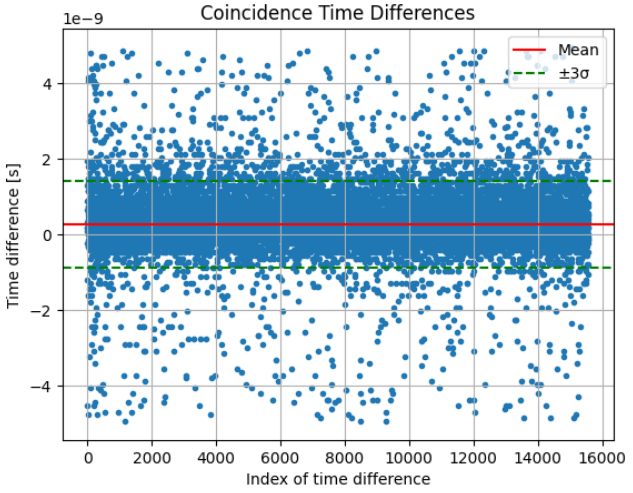
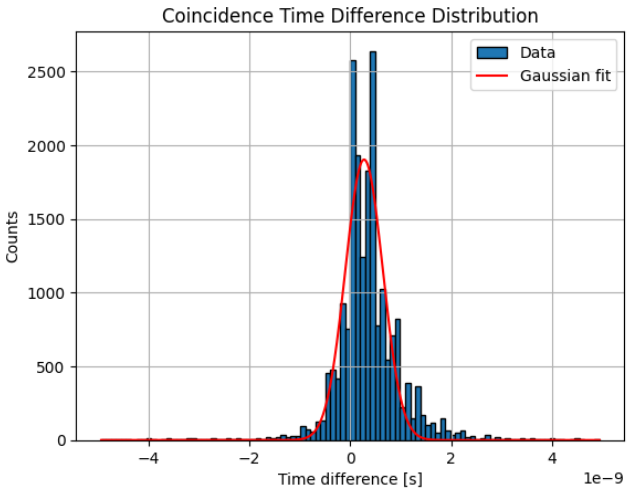


FIGURE 20: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|V\rangle$



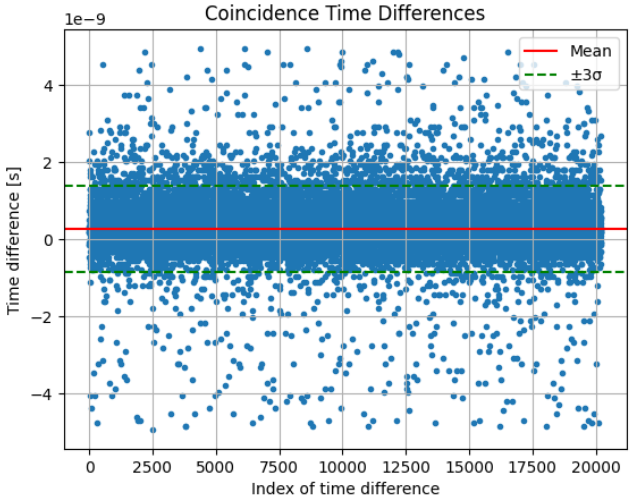
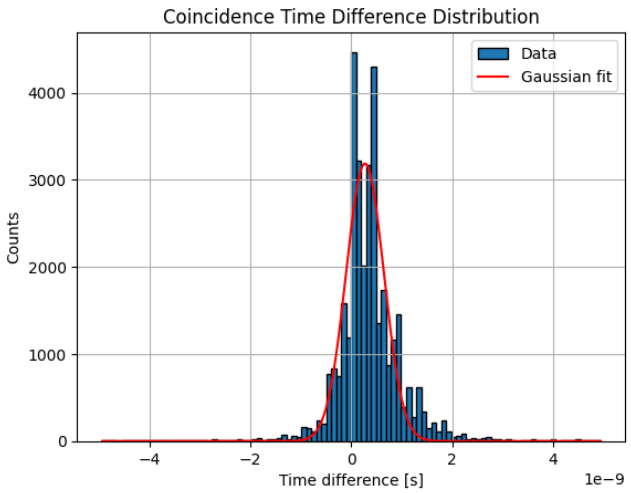


FIGURE 21: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|D\rangle$



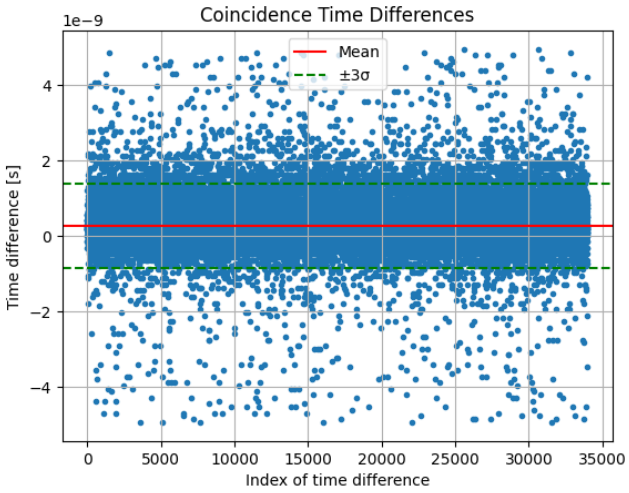
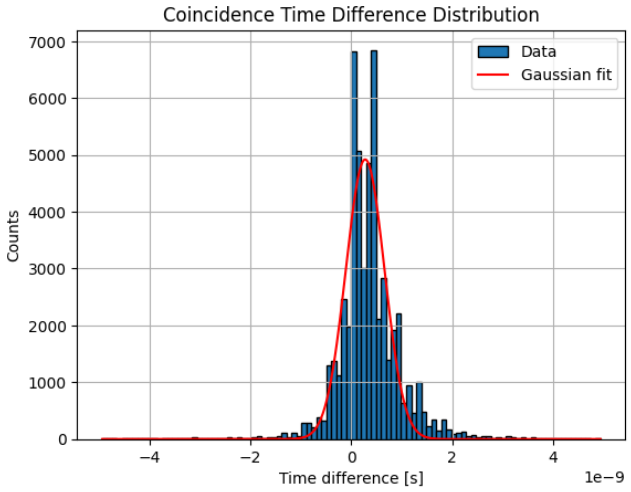


FIGURE 22: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|A\rangle$



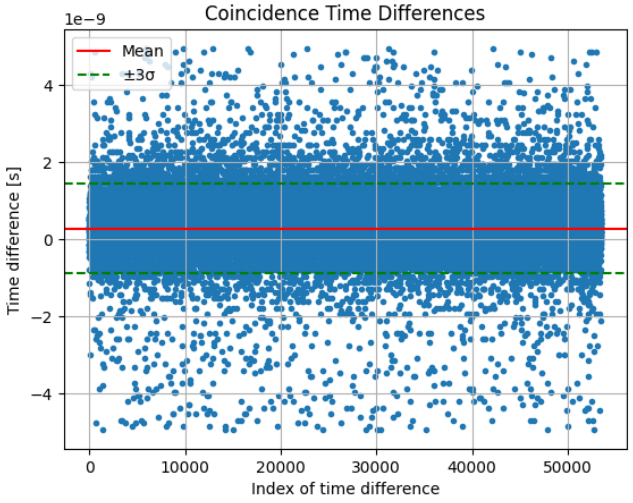
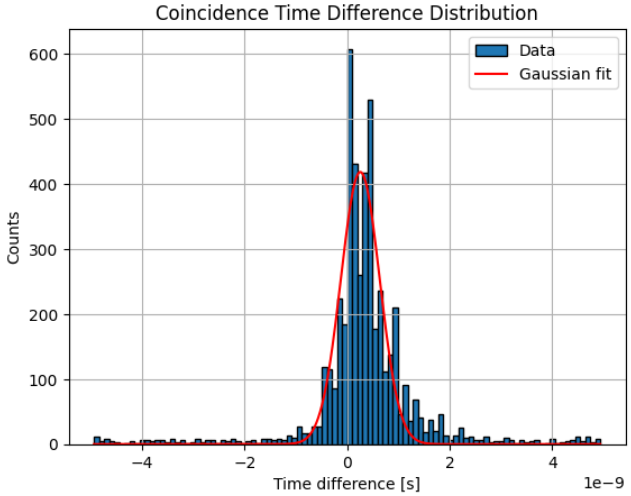


FIGURE 23: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|R\rangle$



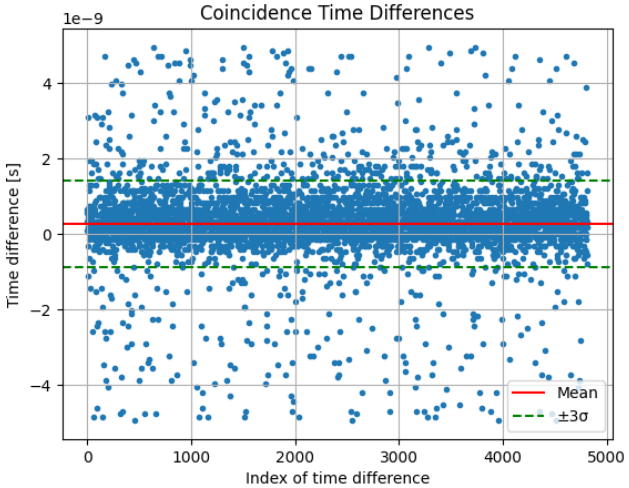


FIGURE 24: Coincidence time differences for $|\psi\rangle$ prepared state measured on $|L\rangle$

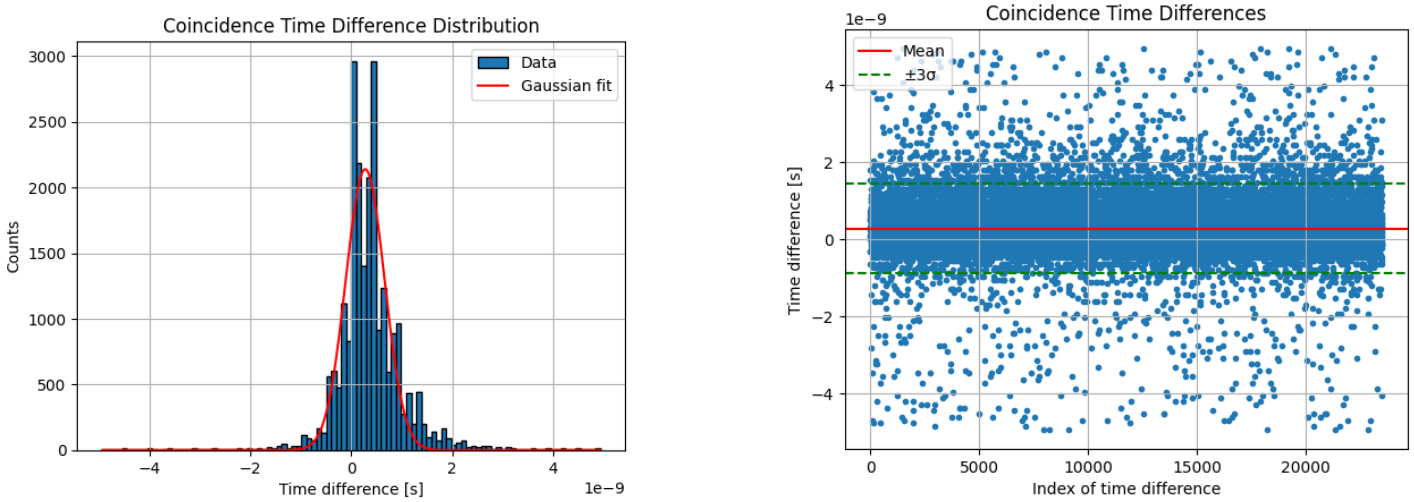


FIGURE 25: Coincidence time differences for $|R\rangle$ prepared state measured on $|V\rangle$

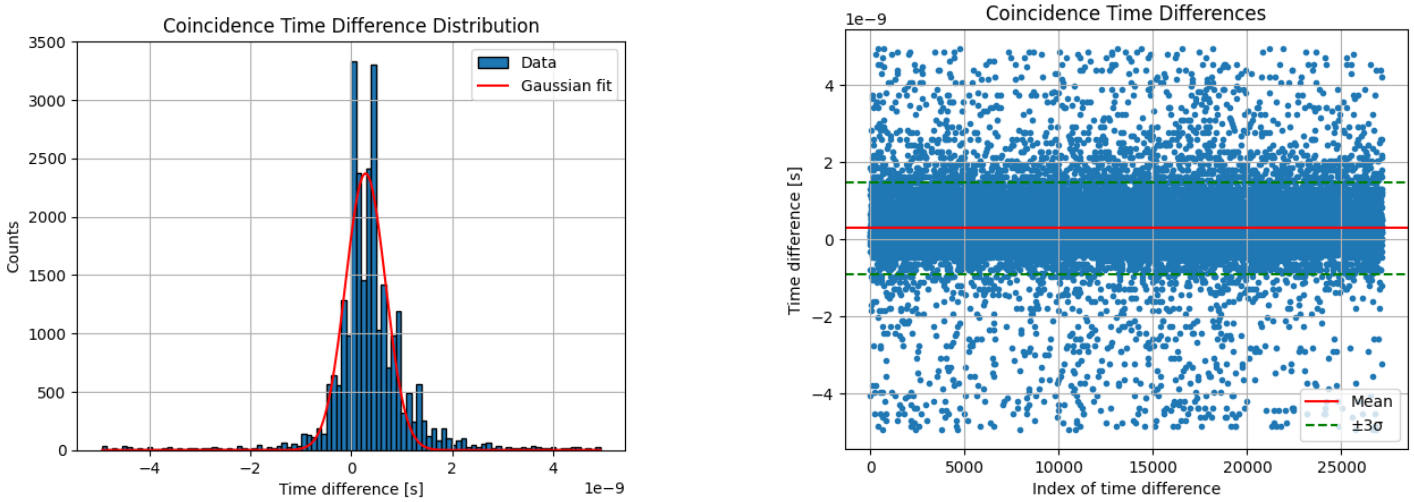


FIGURE 26: Coincidence time differences for $|R\rangle$ prepared state measured on $|D\rangle$

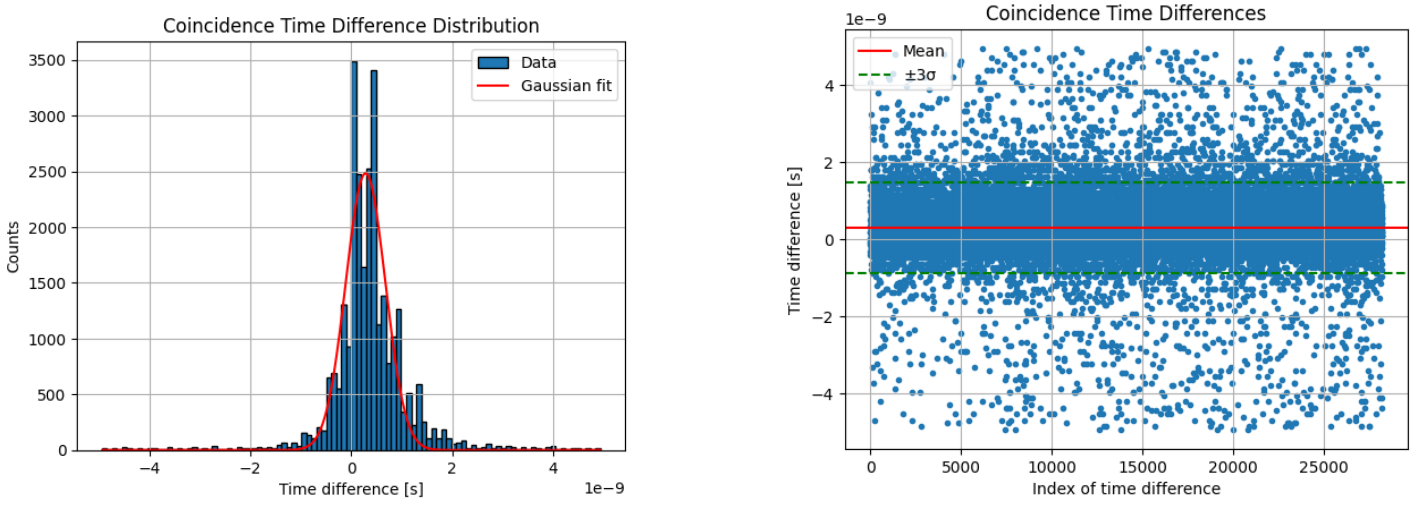
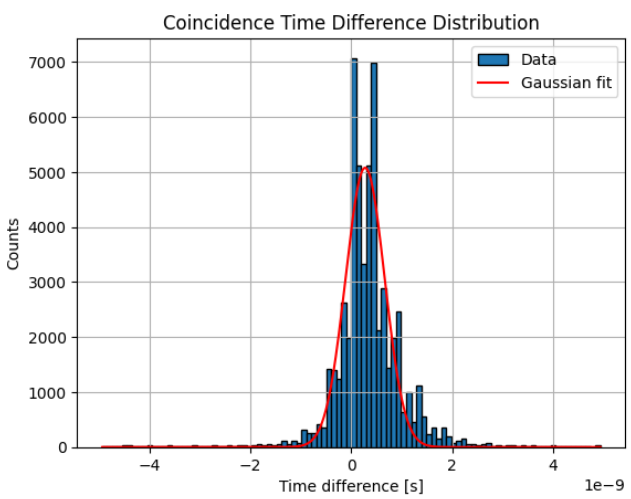


FIGURE 27: Coincidence time differences for $|R\rangle$ prepared state measured on $|A\rangle$



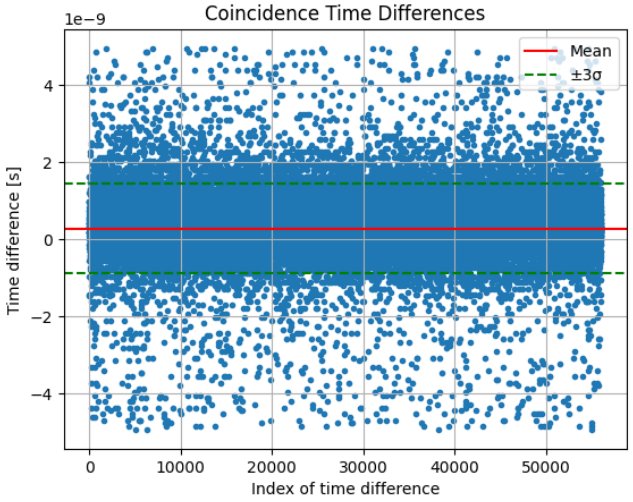
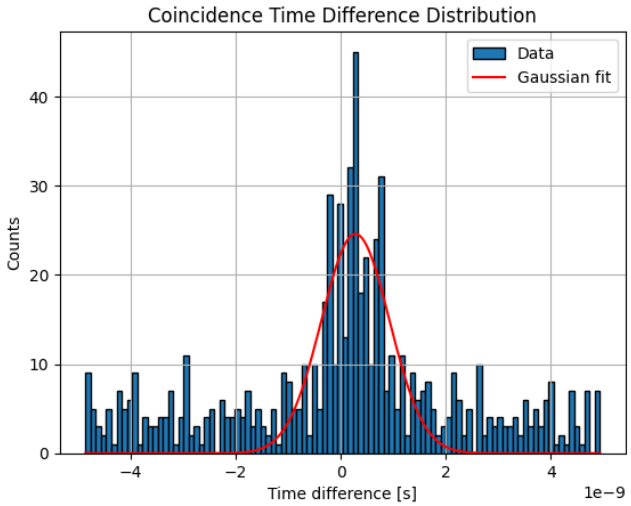


FIGURE 28: Coincidence time differences for $|R\rangle$ prepared state measured on $|R\rangle$



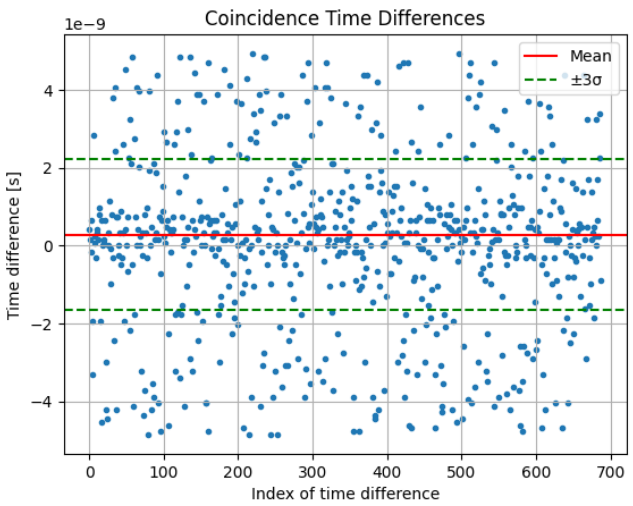


FIGURE 29: Coincidence time differences for $|R\rangle$ prepared state measured on $|L\rangle$