Spin-Network Quantum Reservoir Computing with Distributed Inputs: The Role of Entanglement

Sareh Askari,[1,2,3,*] Youssef Kora,[1,2,3,4] and Christoph Simon[1,2,3,†]

[1] *Department of Physics and Astronomy, University of Calgary, Calgary, Alberta, Canada*
[2] *Institute for Quantum Science and Technology (IQST), University of Calgary, Calgary, Alberta, Canada*
[3] *Hotchkiss Brain Institute (HBI), University of Calgary, Calgary, Alberta, Canada*
[4] *Department of Physics, University of New Brunswick, Fredericton, Canada*

(Dated: November 10, 2025)

Reservoir computing is a promising neuromorphic paradigm, and its quantum implementation using spin networks has shown some advantage when entanglement is present. Here, we consider a distributed scenario in which two distinct input time series are injected into separate qubits of a spin-network reservoir. We investigate how the overall entanglement, as well as its localization in the system, influence the performance of the reservoir. Focusing on bilinear memory tasks that require computing the product of the two inputs, we evaluate the short term memory capacity and correlate it with logarithmic negativity as a measure of bipartite entanglement. We find that short term memory capacity reaches its maximum at relatively small coupling strengths. In contrast, average entanglement peaks at larger couplings. Analyzing entanglement across all bipartitions, we find that the entanglement between the two input qubits is consistently the strongest and most relevant for task performance. In the small coupling strength regime where the short term memory capacity is maximized, the reservoir exhibits an extended memory tail: performance remains high for a long time. Finally, a pronounced dip in performance at zero time delay, observed across frequencies, indicates that information requires a finite propagation time through the reservoir before it can be effectively recalled. In summary, our results show that moderate entanglement—particularly between the two input qubits—plays a key role in enhancing short term memory performance.

I. INTRODUCTION

Inspired by biological systems, neuromorphic approaches aim to mimic brain-like information processing, achieving substantial reductions in energy consumption without major compromises in performance[1–7]. Among these approaches, reservoir computing (RC) stands out for its simplicity and physical realizability. The concept of reservoir computing originated in the early 2000s with the development of Echo State Networks (ESNs)[8] and Liquid State Machines (LSMs) [9, 10]. Unlike traditional neural networks, RC does not require training within the hidden layers; only the output layer weights are optimized [11, 12]. This allows RC to be implemented directly on physical systems without tuning internal parameters. An additional advantage of RC is its model-free, data-driven nature—it can infer the underlying dynamics of a system directly from time series data, without relying on an explicit model [13, 14].

Building on these foundations, quantum reservoir computing (QRC) was introduced in 2017, proposing a quantum reservoir processor capable of emulating nonlinear dynamical systems—including classical chaotic behavior—through complex quantum dynamics [15]. Subsequently, Ghosh et al. presented a quantum reservoir processing platform designed to perform quantum tasks on quantum inputs [16]. Further work demonstrated that even small, noisy quantum systems can successfully perform nonlinear temporal information-processing tasks without error-mitigation techniques, marking the first experimental realization of QRC on near-term gate-model hardware [17].

Subsequent research has demonstrated several potential advantages of QRC [18]. Notably, tasks such as prediction and reconstruction that would require thousands of neurons in classical reservoirs can be performed by only a few entangled qubits in a quantum reservoir [19, 20]. QRC has been explored on various physical platforms, including spin networks [21–23], Rydberg atom arrays [23–26], Bose–Hubbard model[27] and nonlinear oscillators [28, 29], each offering distinct trade-offs in terms of scalability, tunability, and interaction dynamics.

It has been shown that an entanglement advantage can arise in an Ising spin network when a single time series is injected into one qubit of a four-qubit system. Specifically, they showed that the presence of entanglement enhances short-term memory when the dissipation timescale exceeds that of the input signal, highlighting its beneficial role in the system's information-processing performance [21].

While previous work has focused on single-input scenarios, the dynamics of multi-input quantum reservoirs—especially how they depend on entanglement—remain underexplored. Recent studies have begun to address multivariate time-series settings, [30, 31]. However, these works primarily focus on forecasting performance and hardware feasibility, without analyzing how entanglement within the reservoir contributes to

* sareh.askari@ucalgary.ca
† christoph.simon@ucalgary.ca

computational capability. On the other hand, entanglement provides advantages for distributed quantum information processing, particularly in communication-complexity tasks [32, 33]. These results suggest that quantum correlations can enhance the exchange and integration of information between subsystems [34], providing a broader motivation for exploring multi-input quantum reservoirs. A recent study introduced a gate-based multivariate QRC (MTS-QRC) that assigns *d injection* qubits (one per feature) and evolves them under a Hamiltonian; the approach was validated on IBM's Heron-R2 NISQ processor [30]. Another recent related study applied a similar gate-based QRC framework to real-world financial data, using a quantum reservoir to forecast a real-world financial time-series data index [31]. The approach was presented as a proof-of-concept for leveraging quantum reservoirs in finance and econometrics.

In this work, we investigate the case of two input channels by injecting two distinct time series simultaneously into two separate qubits of a four-qubit Ising spin network, and we study the effect of entanglement and its structure on the short-term memory (STM) capacity [35], denoted as C_{STM}. The system is governed by a Hamiltonian with randomly sampled coupling strengths, drawn from a uniform distribution. The width of this distribution serves as a hyperparameter that allows us to vary the amount of entanglement in the reservoir. Our focus is on bilinear memory tasks, in which the reservoir is trained to compute the product of the two input signals at various time delays τ. These tasks are bilinear and require the reservoir to capture correlations between the two input streams.

Our results indicate that C_{STM}, increases as the average entanglement across all bipartitions of the system grows from zero. However, the maximum C_{STM} is reached at relatively small coupling strengths, where the entanglement is only moderate and has not yet peaked. When analyzing different bipartitions of the system, we find that the two input qubits are more strongly entangled with each other than with the rest of the network, or compared to the entanglement between non-input qubits. Importantly, reservoir performance is optimal in the regime where this input–input entanglement dominates.

Moreover, we find that in this regime, the memory profile as a function of time delay τ differs from other regimes. The memory decays more slowly, consistent with information being partially localized between the two input qubits due to their stronger mutual entanglement. We also observe a pronounced dip at zero time delay in the memory profile, suggesting that information requires a finite time to propagate through the network before it can be effectively recalled.

The remainder of this paper is organized as follows. Section 2 outlines the methodology, including the simulation of open-system dynamics, evaluation of entanglement and memory capacity, and the definition of computational tasks. Section 3 presents the numerical results, highlighting how memory capacity depends on coupling strength and entanglement. Finally, Section 4 discusses the broader implications, emphasizing the role of input–input entanglement and contrasting weak and strong coupling regimes.

II. METHODOLOGY

Our approach involves simulating a spin-based quantum reservoir governed by a generalized transverse-field Ising Hamiltonian. The goal is to evaluate how entanglement influences memory performance in a distributed-input setting. Below, we describe the reservoir configuration, input injection method, simulation details, and performance evaluation techniques (Figure 1

We study spin networks consisting of 4 qubits. We model the dynamics of an open quantum spin network governed by the Lindblad master equation[36]:

$$\frac{d\rho}{dt} = -i[H,\rho] + \sum_i \left(L_i \rho L_i^\dagger - \frac{1}{2}\left\{ L_i^\dagger L_i, \rho \right\} \right), \qquad (1)$$

where ρ is the density matrix of the system, H is the system Hamiltonian, and L_i are the Lindblad operators associated with local dissipation on each qubit.

The Hamiltonian H consists of local Z-field terms and pairwise XX interactions between qubits [37]:

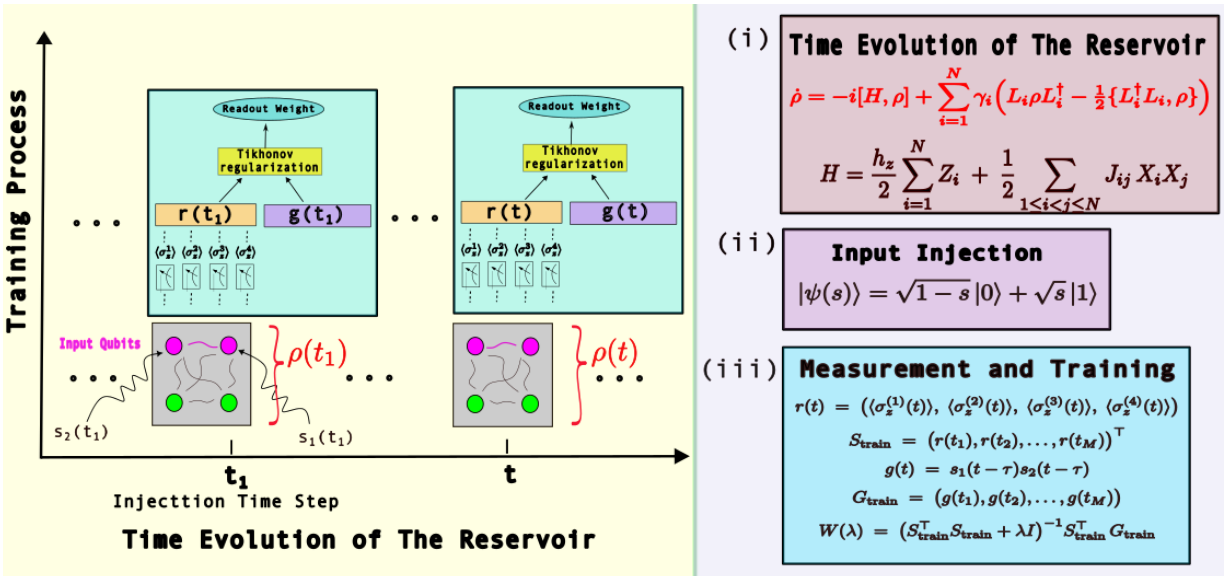
$$H = \sum_{i=1}^{4} \frac{h_z}{2} Z_i + \sum_{i<j} \frac{J_{ij}}{2} X_i X_j, \qquad (2)$$

where Z_i and X_i denote the Pauli operators acting on the ith qubit. We fix the local field strength to $h_z = 1.5$, which allows us to explore a broad range of ratios between the magnetic field and the interaction strength, h_z/J_s. This ensures that the simulations cover regimes where the dynamics are dominated by the local Z-field as well as those where the pairwise XX couplings become the leading term. J_{ij} are the coupling coefficients between qubits i and j. The values of J_{ij} are randomly sampled from a uniform distribution in the interval $[-J_s/2, J_s/2]$, where J_s is the coupling strength parameter. The coupling matrix J_{ij} determines the connectivity and interaction strength within the network.

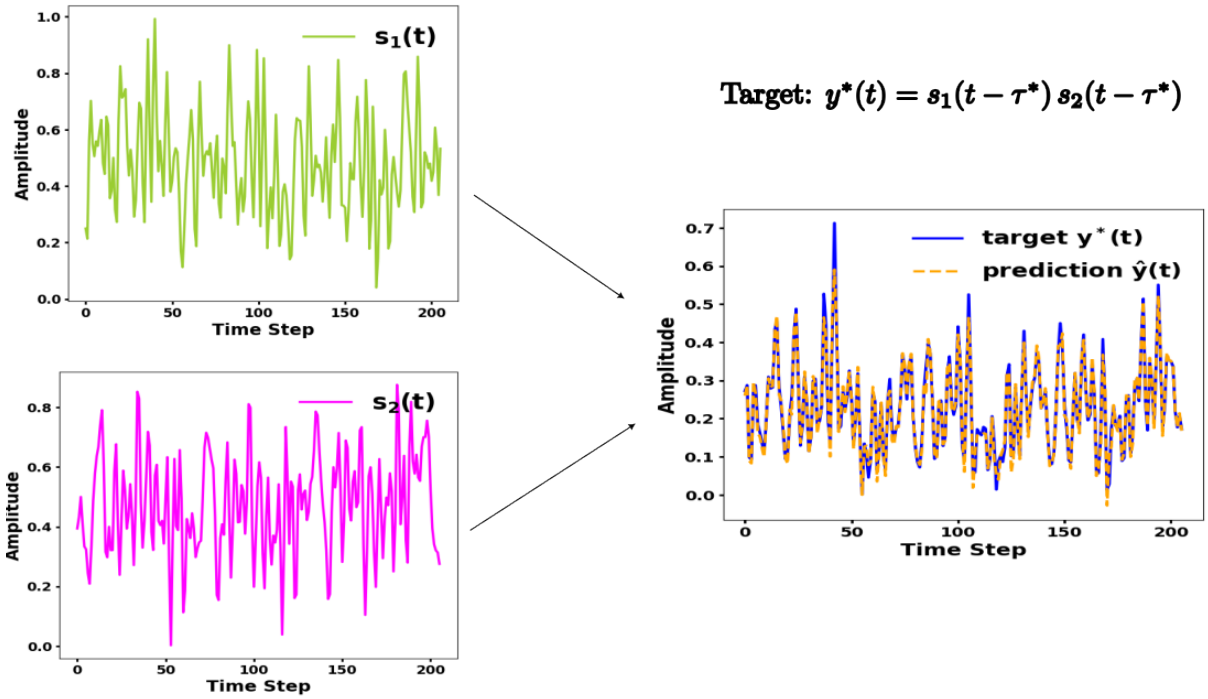
Dissipation is introduced through local Lindblad operators of the form:

$$L_i = \sqrt{\Gamma} \cdot \frac{1}{2}(Z_i + iY_i), \qquad (3)$$

where $\Gamma = 0.01$ is the dissipation rate. The chosen value is small compared to both the local field ($h_z = 1.5$) and the interaction strengths ($J_{ij} \in [-J_s/2, J_s/2]$), ensuring that dissipation acts as a weak perturbation rather than dominating the coherent dynamics. Y_i is the Pauli-Y operator on qubit i. These non-Hermitian terms model the interaction of each spin with its local environment.



(a)



(b)

Figure 1: *Distributed quantum reservoir computing pipeline and task performance.* **(a)** Along the horizontal axis, the reservoir—modeled as a system of four spin qubits—evolves in time under the Ising Hamiltonian [box (i)] together with environmental interactions described by the Lindblad master equation [box (i)]. At discrete injection steps t_k, the two input sequences $s_1(t_k)$ and $s_2(t_k)$ are injected *simultaneously but into separate qubits*, reflecting the distributed nature of the input encoding described in box (ii). The reservoir then evolves from state $\rho(t_1)$ to $\rho(t)$ until the next injection, at which point the cycle repeats. The vertical line separates the physical evolution (lower part) from the learning stage (upper part). In the measurement-and-training step [box (iii)], the expectation values of all four qubits are concatenated at each time step to form a reservoir state vector $\mathbf{r}(t)$. Stacking these vectors across all times yields the training matrix R_{train}. The target sequence, defined as $g(t)$, is collected across all time steps to form G_{train}. This target is mapped to R_{train}, and the readout weights $W(\lambda)$ are obtained via Tikhonov regularization. **(b)** Example at the optimal delay τ^* (selected by maximizing $C_{\text{STM}}(\tau)$) for coupling $J_s = 0.1$ and input frequency $f = 2$. Top left: input $s_1(t)$. Bottom left: $s_2(t)$. Top right: target $y^*(t) = s_1(t-\tau^*)\, s_2(t-\tau^*)$. Bottom right: test prediction $\hat{y}(t)$ overlaid with target $y^*(t)$.

We numerically integrate the evolution using a fourth-order Runge-Kutta (RK4) method [38]. The time derivative $\frac{d\rho}{dt}$ is computed at each step by evaluating both the coherent unitary evolution governed by H and the non-unitary dissipative dynamics from the Lindblad terms.

A. Distributed Input Injection

a. Generating input sequences Similar to standard practices in time–frequency benchmarking studies [39], we generate multi-component sinusoidal inputs composed of several frequency components with randomized phases to probe the reservoir's temporal response. We generate 9 distinct time series, each composed of 20 sinusoidal components with frequencies $f_k \in \left[\frac{f_0}{5000}, \frac{f_0}{50}\right]$, linearly spaced across this range. Independent phase offsets $\phi_k \sim \mathcal{U}(0,1)$ are used to introduce variability across sequences and components. The time variable $t \in [0,1]$ is sampled at $N = 1000$ discrete time steps. Each sequence is constructed according to:

Each input sequence is constructed as a sum of sinusoidal components,

$$x_q(t) = \sum_{k=1}^{20} \sin\Big(2\pi f_k t + 2\pi\phi_k^{(i)}\Big),$$

and is then normalized to the range $[0,1]$ using min–max normalization:

$$s_q(t) = \frac{x_q(t) - \min_t\big(x_q(t)\big)}{\max_t\big(x_q(t)\big) - \min_t\big(x_q(t)\big)}.$$

Here the subscript q indicates the qubit to which the signal is injected (e.g., $q = 1, 2$ for input qubits 1 and 2).

b. Input Injection. The total evolution time was set to $t_{\text{final}} = 2750$ (in dimensionless units), and input signals were injected periodically every $\Delta t = 7.5$ time units. This choice yielded the best overall memory performance across the tested coupling strengths and input frequencies, indicating an optimal trade-off between information retention and refresh of input signals. We inject classical inputs s_1 and s_2 from two independent sequences by resetting two designated qubits (typically qubits 1 and 2) to pure states that encode the current input values. Each input is mapped to a single-qubit state defined as:

$$|\psi(s)\rangle = \sqrt{1-s}\,|0\rangle + \sqrt{s}\,|1\rangle \tag{4}$$

where $s \in [0,1]$ is the normalized input value at the current time step.

To inject the new input states, we first remove the original states of the input qubits by computing the partial trace:

$$\rho_{34} = \mathrm{Tr}_{12}\,(\rho)$$

We then construct the updated density matrix by taking the tensor product of the reduced reservoir state with the new input states:

$$\rho' = \rho_1(\mathrm{s}_1) \otimes \rho_2(\mathrm{s}_2) \otimes \rho_{34}. \tag{5}$$

where $\rho_1(\mathrm{s}_1) = |\psi(\mathrm{s}_1)\rangle\langle\psi(\mathrm{s}_1)|$ and similarly for $\rho_2(\mathrm{s}_2)$.

B. Bilinear short-term memory (STM) task

We evaluate the reservoir's capability to retain and process temporal information using a bilinear short-term memory (STM) task. At each time step k, the target output is defined as the element-wise product of the two input signals with a temporal delay τ:

$$\bar{y}_k = s_1(k-\tau)s_2(k-\tau). \tag{6}$$

This task requires the reservoir to store past information from both input streams and to reproduce their temporal correlations at a later time, thus serving as a measure of its memory capacity.

a. Training the reservoir From each input sequence, we sample values at fixed intervals. These sampled input values are then used to construct the target functions for training and evaluation. The reservoir outputs—given by the expectation values $\langle\sigma_z^i\rangle$ from all qubits i —are collected at each injection time, after the reservoir has reached its steady state, and assembled into a feature matrix. A linear regression model is then trained to map the reservoir outputs to the corresponding target function, with or without delay, thereby enabling the reservoir to retain and process short-term temporal information.

b. Testing the reservoir's output In the testing phase, we evaluate the generalization performance of the reservoir using two previously unseen input sequences. As described previously, after feeding the test sequences into the reservoir and allowing it to reach a steady state, we collect the $\langle\sigma_z\rangle$ expectation values from the readout qubits and assemble them into a test feature matrix.

c. Ridge regression and the regularization parameter λ To obtain the optimal output weights, we use ridge regression (also known as Tikhonov regularization). Given the training data R (reservoir states) and target outputs G, the regression seeks a linear map A that minimizes the regularized loss function

$$\mathcal{L}(A_{out}) = \|G - A_{out}R\|^2 + \lambda\|A_{out}\|^2, \tag{7}$$

where λ controls the trade-off between fitting accuracy and weight regularization.

The analytical solution is given by

$$A_{out} = GR^{\mathrm{T}}\left(RR^{\mathrm{T}} + \lambda I\right)^{-1}, \tag{8}$$

where I is the identity matrix. A larger λ suppresses large weight amplitudes and prevents overfitting, while a smaller λ allows more faithful fitting [40].

The performance of the model is quantified using the squared Pearson correlation coefficient between the predicted and true outputs. To evaluate the overall memory

capacity of the reservoir, we define the short-term memory (STM) capacity at each τ as[41]:

$$C_{\mathrm{STM}}(\tau) = \frac{\mathrm{cov}^2(y, \bar{y}_\tau)}{\sigma_y^2 \, \sigma_{\bar{y}_\tau}^2}, \tag{9}$$

where y is the predicted output vector, $\bar{y}_\tau$ is the delayed target sequence, $\mathrm{cov}(y, \bar{y}_\tau)$ denotes their covariance, and σ represents the standard deviation. The optimal regularization parameter λ for ridge regression is selected by maximizing the test-time memory performance over a logarithmically spaced range of values. This ensures robust generalization while avoiding overfitting. To quantify the reservoir's memory performance, we compute the total short-term memory capacity as

$$C_{\mathrm{STM}} = \sum_{\tau=0}^{\infty} C_{\mathrm{STM}}(\tau). \tag{10}$$

In practice, the summation is truncated once $C_{\mathrm{STM}}(\tau)$ becomes negligible. $C_{\mathrm{STM}}(\tau)$ has already decayed to zero by $\tau = 24$ in all cases considered here, so we restrict the calculation to this range.

C. Entanglement Analysis

To quantify bipartite entanglement within the reservoir, we use the logarithmic negativity. For a bipartition between subsystem A and the remainder of the system, it is defined as

$$E_A(\rho) = \log \left\| \rho^{T_A} \right\|_1, \tag{11}$$

where ρ^{T_A} denotes the partial transpose of the density matrix with respect to subsystem A and logarithm is taken in base 2. Here, A can correspond either to a single qubit i or to a pair of qubits ij. We denote by E_i the entanglement between qubit i and the rest of the system, and by E_{ij} the entanglement between the pair of qubits $\{i, j\}$ and the remaining qubits, both corresponding to the general definition $E_A(\rho)$ for subsystem A. All logarithmic negativities are computed from the steady-state density matrix ρ. During the simulation, we evolve the full density matrix $\rho(t)$ of the open quantum system under Ising hamiltonian and Lindblad dynamics and extract it at regular intervals. At each sampled time step, we compute the logarithmic negativity for multiple bipartitions of the four-qubit network. At each time step, the corresponding entanglement values are denoted by $E_i(t)$ or $E_{ij}(t)$ for single–vs.–rest and pair–vs.–rest partitions, respectively. These quantities are then averaged over time after the reservoir has reached its steady state.

This quantity provides a single scalar metric for tracking how entanglement is distributed throughout the network and how it evolves with coupling strength J_s. In the results section, we focus primarily on the behavior of single–vs.–rest partitions (E_1–E_4), pair–vs.–rest partitions (E_{12}, E_{13}, E_{14}), to examine how different forms of entanglement correlate with memory capacity and task performance.

III. RESULTS

A. Memory capacity and average entanglement in different coupling regimes.

Figure 2a shows that the optimal performance is reached during the initial rise of memory capacity, before the peak of average entanglement, rather than at the maximum entanglement point. Specifically, the memory capacity increases going from $J_s \approx 0$ to $J_s \approx 0.3$. In this regime, the entanglement also grows. From $J_s \approx 0.3$ to $J_s = 6$, the memory capacity decreases again, while entanglement continues to grow, reaching its maximum value. Beyond this point, further increases in the coupling strength reduce both entanglement and memory capacity.

Figure 2b reveals that E_1 and E_2 are consistently larger than E_3 and E_4.

Figure 2c further shows that E_{12} is larger than E_{13} and E_{14}. Combined with the observations from Fig. 2b, this establishes that the entanglement between qubits 1 and 2 dominates over other types of entanglement in the system.

When examining Fig. 2a–c, it appears that the memory performance peaks around $J_s \approx 0.3$, coinciding with regions where the entanglement associated with the input qubits (E_1, E_2) differs most strongly from that of the remaining qubits (E_3, E_4), and where the pairwise input entanglements (E_{13}, E_{14}) become more pronounced relative to E_{12}. To quantify this observation, we compute the following differences and ratios of entanglement measures. The differences $(E_1 + E_2) - (E_3 + E_4)$ and $(E_{13}+E_{14})-(2E_{12})$ both reach their maxima at $J_s \approx 0.5$, which is near the coupling strength corresponding to the peak of C_{STM} ($J_s \approx 0.32$), whereas the overall entanglement peaks at $J_s \approx 6$ (where the performance is not maximum). The trends of these differences closely follow that of C_{STM} versus J_s, particularly for larger couplings ($J_s > 6$) (Fig. 2d).

Similarly, the balance ratios,

$$\frac{E_1 + E_2}{E_3 + E_4} \quad \text{and} \quad \frac{E_{13} + E_{14}}{2E_{12}},$$

exhibit a maximum at approximately the same coupling strength—again close to the point of highest C_{STM} (Fig. 2e). All these quantities reach their maxima within approximately the same coupling regime— an order of magnitude weaker than the coupling strength at which the entanglements attain their peak. Taken together, these results indicate that memory capacity peaks when the entanglement between the two input qubits (1 and 2) dominates over other partitions, suggesting that this

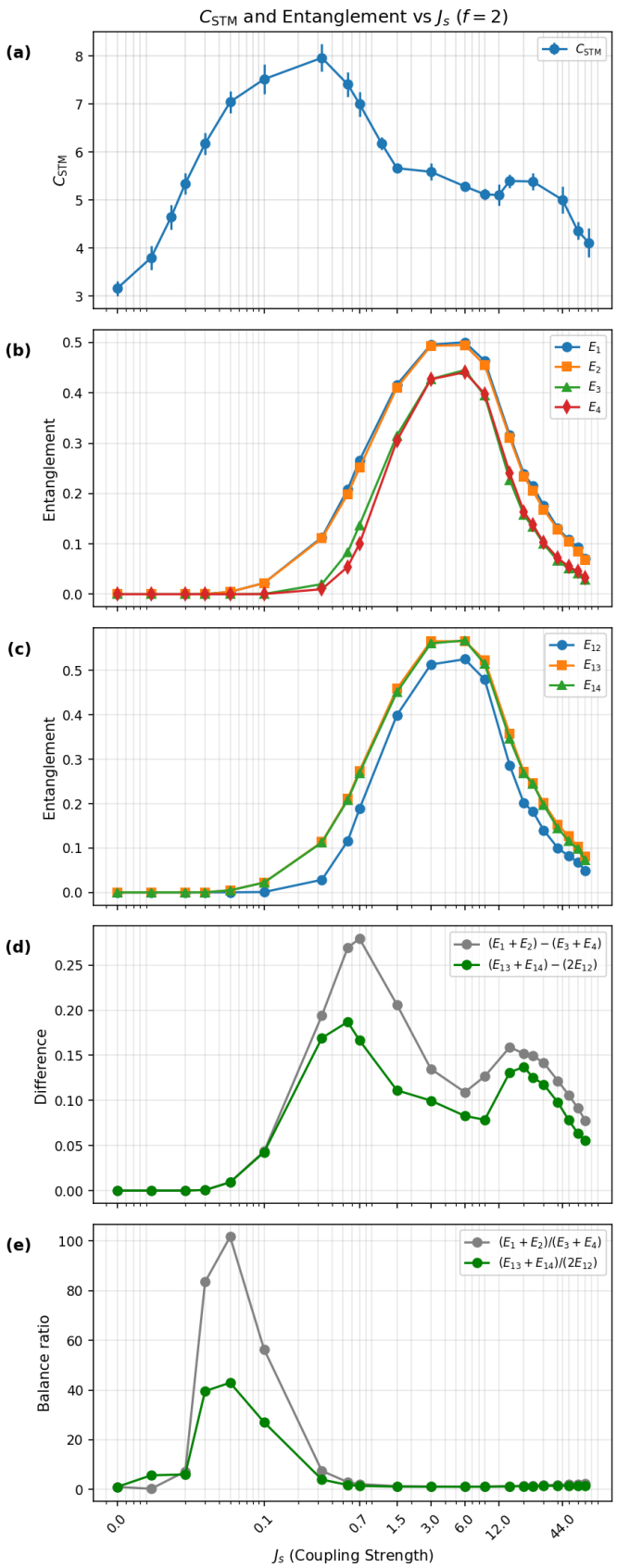
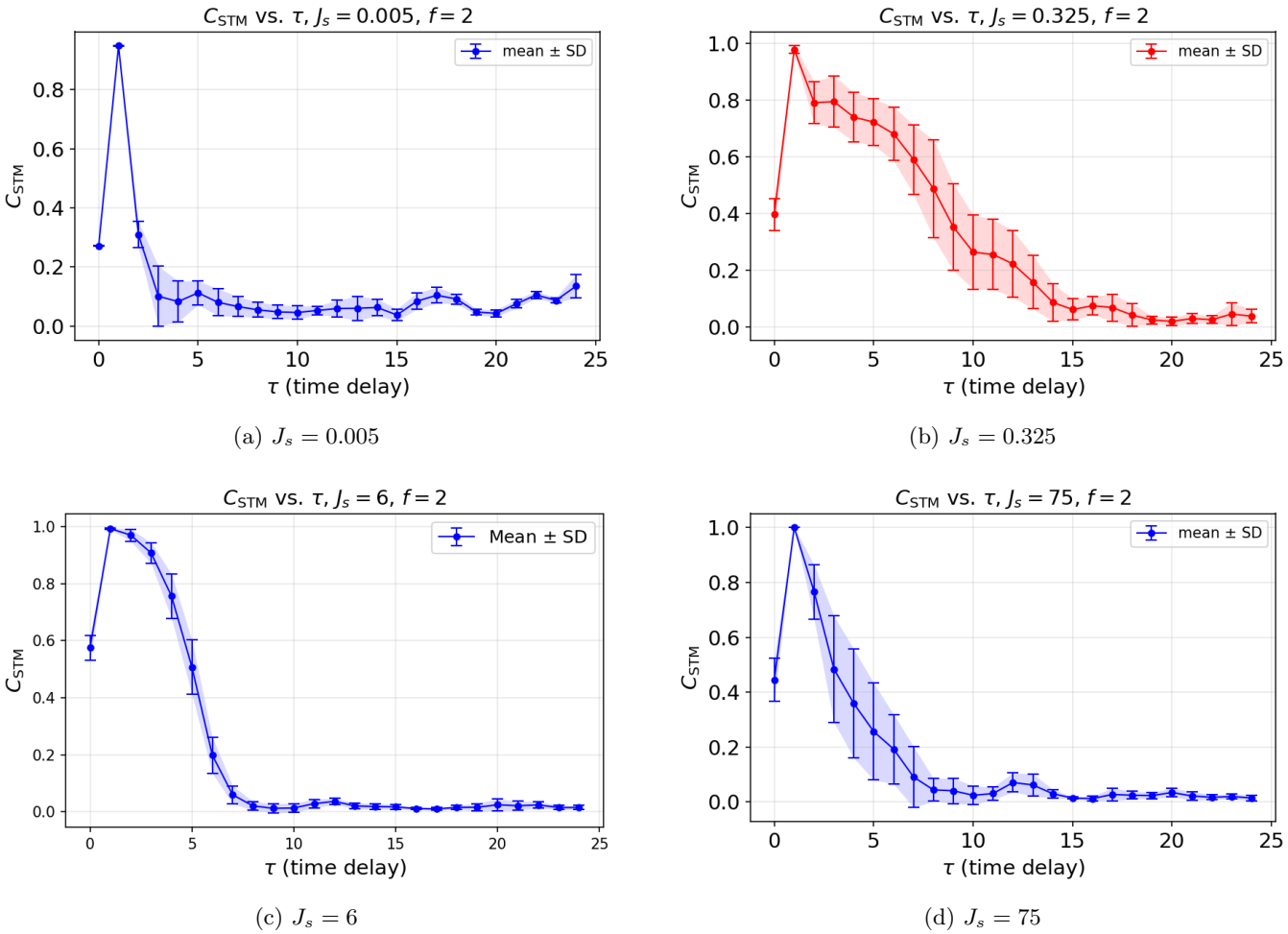


Figure 2: *Memory capacity, average entanglement, and input–input entanglement in different coupling regimes.* **(a)** Memory capacity vs. J_s peaks at intermediate couplings ($J_s \approx 0.3$) before decaying. **(b)** Curves show the logarithmic negativity E, averaged over ten random connectivity matrices. Entanglements correspond to different partitions of the system: $E_1 : \{1\}|\{2,3,4\}$, $E_2 : \{2\}|\{1,3,4\}$, $E_3 : \{3\}|\{1,2,4\}$, $E_4 : \{4\}|\{1,2,3\}$. (E_1–E_4) show similar non-monotonic behavior with a maximum near $J_s \approx 6$. **((c)** Pair–vs.–rest partitions are $E_{12} : \{1,2\}|\{3,4\}$, $E_{13} : \{1,3\}|\{2,4\}$, $E_{14} : \{1,4\}|\{2,3\}$, following the same trend with modest differences. At $f = 2$, larger E_1 and E_2 together with $E_{12} < E_{13}, E_{14}$ indicate that entanglement is concentrated within the input pair $\{1,2\}$. **((d)** Gray curve: difference between $E_1 + E_2$ and $E_3 + E_4$; green curve: difference between $E_{13} + E_{14}$ and $2E_{12}$. Both exhibit maxima around $J_s \approx 0.3$–0.7, following a trend similar to panel (a). **((e)** Balance ratios: $(E_1 + E_2)/(E_3 + E_4)$ and $(E_{13} + E_{14})/(2E_{12})$, both peaking around $J_s \approx 0.05$. Overall, the optimum performance occurs along the initial rise before the entanglement peak, where the entanglement between qubits 1 and 2 dominates relative to other partitions—suggesting that this form of entanglement is the most beneficial for the task.



(a) $J_s = 0.005$

(b) $J_s = 0.325$

(c) $J_s = 6$

(d) $J_s = 75$

Figure 3: Short-term memory capacity $C_{\mathrm{STM}}(\tau)$ versus delay τ at three coupling strengths. At very small coupling ($J_s = 0.005$), the memory profile resembles that of the moderate and high-coupling case ($J_s = 6$ and $J_s = 75$), with a fast decaying memory. By contrast, the weak-coupling case ($J_s = 0.325$) exhibits a long, low-amplitude tail (slow fading memory). These distinct decay behaviors correlates with Fig. 2: in figure Fig. 2, the performance peaks at $J_s \approx 0.3$, here we can see that at this regime the memory profile is not similar to other regimes, exhibiting a slower decay. The difference can be understood as information becoming trapped between the two input qubits at $J_s = 0.325$, consistent with Fig. 2, which shows that the input qubits are more strongly entangled with each other than with the rest of the network.

particular form of entanglement is especially relevant for the computational task.

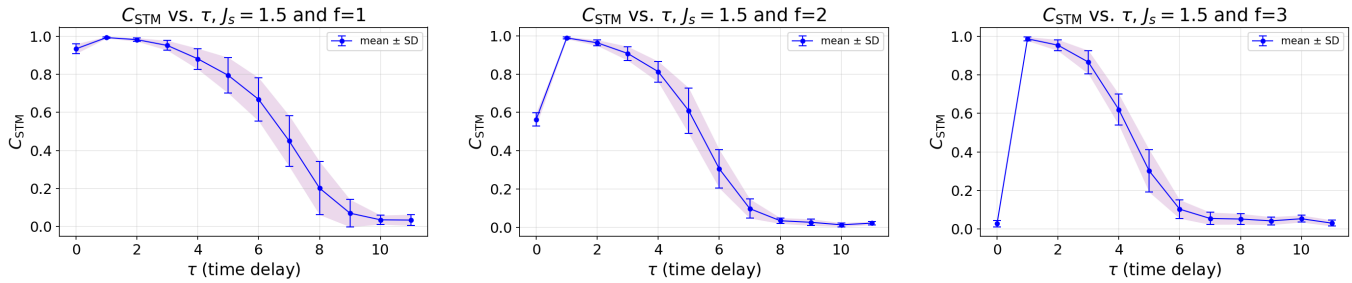
B. Memory tail in different coupling regimes

Examining the memory capacity as a function of time delay τ across different coupling strengths, we find that for very small or nearly vanishing couplings the memory decays around $\tau \approx 8$. (Fig. 3a) For both intermediate couplings ($J_s \approx 6$) and strong couplings ($J_s \approx 75$), the memory also decays rapidly and vanishes near $\tau \approx 8$. (Fig. 3c and Fig. 3d)

In contrast, for a lower but finite coupling of $J_s \approx 0.3$, a pronounced long memory tail emerges, extending until approximately $\tau \approx 18$. (Fig. 3b) As discussed in Sec. 2.1, in this regime the input qubits are more strongly entangled with each other than with the rest of the network. The appearance of a long memory tail therefore suggests that information tends to remain localized between the input qubits, slowing the overall decay of memory.

C. Dip in Memory profile

A pronounced dip is consistently observed in the memory profile as a function of time delay τ. The memory capacity is strongly suppressed at zero delay, corresponding to the moment when the input is injected (Fig. 4). It subsequently rises, reaching nearly $C_{\mathrm{STM}}(1) \approx 1$. For longer delays, the behavior depends on the input frequency: at $f = 1$, the capacity remains relatively high at $C_{\mathrm{STM}} \approx 0.9$; for $f = 2$, it decreases to about $C_{\mathrm{STM}} \approx 0.6$; and for $f = 3$, it vanishes entirely ($C_{\mathrm{STM}} = 0$). Thus,



(a) $f = 1$ (b) $f = 2$ (c) $f = 3$

Figure 4: Short-term memory capacity $C_{\mathrm{STM}}(\tau)$ versus delay τ at fixed coupling $J_s = 1.5$ for three driving frequencies ($f \in \{1, 2, 3\}$). All curves exhibit a dip at $\tau = 0$, a near-term peak, and a subsequent decay. The depth of the zero-delay dip increases with f, consistent with an increasingly challenging task at higher driving frequencies. This initial dip reflects the fact that, in our distributed setting, information requires additional time to propagate throughout the reservoir before it can be effectively recalled.

the dip at zero delay becomes more pronounced as the task difficulty increases (Fig. 4).

This behavior reflects the fact that the system requires a finite amount of time to distribute information throughout the reservoir before it can be effectively recalled.

IV. DISCUSSION

Our results indicate that in our distributed-input setting, the best performance does not occur at the strongest coupling or at the highest level of average entanglement. Instead, it emerges in a regime characterized by moderate coupling and localized entanglement. In this optimal regime, the entanglement between the two input qubits is stronger than that involving other parts of the reservoir. This concentration of entanglement on the input qubits appears to be particularly beneficial for computation. It may enable the reservoir to retain and process correlations from the input time series efficiently, without excessive mixing with the rest of the network. The moderate coupling ensures communication between the inputs while maintaining a degree of isolation that preserves useful correlations. Overall, these observations suggest that it is not only the amount of entanglement but also its structure—specifically, its localization on the input nodes—that matters for short-term memory performance in distributed quantum reservoirs.

A further notable feature is the pronounced dip in memory performance at zero time delay. This dip, which becomes sharper at higher input frequencies, indicates that information cannot be recalled instantaneously at the time of injection. Instead, the reservoir requires a finite time to propagate and redistribute the input across the network before it can be effectively utilized.

These findings open several directions for future research. It would be interesting to investigate whether similar relationships between entanglement structure and memory capacity appear in larger spin networks, especially on experimentally accessible platforms. Another natural extension would be to examine how the performance changes when different inputs are injected into more than two qubits. Another promising direction would be to explore the effect of varying the coupling strength between the input qubits—making it stronger or weaker than the other couplings in the network—and examining how such asymmetries influence different forms of entanglement and the reservoir's overall performance. Moreover, applying the same analysis to other benchmark tasks, such as nonlinear autoregressive moving average (NARMA) sequences, could help assess the generality of the observed trends and clarify the broader computational capabilities of spin-network reservoirs. Such tasks were not explored here due to computational constraints but represent an important direction for future work.

Overall, examining how the amount and structure of entanglement within such network systems affect their memory capacity may help bridge quantum dynamics and information processing, paving the way toward more advanced machine-learning and neural-network algorithms that harness quantum properties for distributed computation.

CODE AVAILABILITY

The simulation and analysis codes used in this study are available in `GitHub`. Additional scripts and data are available from the corresponding author upon reasonable request.

ACKNOWLEDGEMENTS

This work was supported by the National Research Council of Canada through its Applied Quantum Computing challenge program, the Natural Sciences and Engineering Research Council through its Discovery Grant

program, by an Alberta Innovates Discovery Grant supplement, and by Quantum City.

[1] O. Mutlu, S. Ghose, J. Gómez-Luna, and R. Ausavarungnirun, Processing data where it makes sense: Enabling in-memory computation (2019), arXiv:1903.03988 [cs.AR].

[2] M. Li, X. Wen, X. Liu, Y. Tan, L. Deng, J. Pei, and L. Shi, Nature **628**, 974 (2024).

[3] C. Chen, Y. Zhou, L. Tong, Y. Pang, and J. Xu, Advanced Materials **37**, 2400332 (2025).

[4] S. Park, Y. Lee, H. Kim, T. Yoon, Y. Park, J.-U. Park, Y. M. Song, J. Jeong, H. Yoo, and H. Yoo, Nature Communications **16**, 57352 (2025).

[5] P. Blouw and C. Eliasmith, in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020) pp. 8534–8538.

[6] V. Bhatnagar and A. Kumar, Energy Storage **7**, e70272 (2025).

[7] H. Cui, Y. Xiao, Y. Yang, M. Pei, S. Ke, X. Fang, L. Qiao, K. Shi, H. Long, W. Xu, *et al.*, Nature Communications **16**, 2263 (2025).

[8] H. Jaeger, *The "echo state" approach to analysing and training recurrent neural networks—with an erratum note*, Technical Report 148 (German National Research Center for Information Technology GMD, Bonn, Germany, 2001).

[9] W. Maass, T. Natschläger, and H. Markram, Neural computation **14**, 2531 (2002).

[10] K. Nakajima and I. Fischer, Nature Communications **15**, 45187 (2024).

[11] Verstraeten, David and Schrauwen, Benjamin and D'Haene, Michiel and Stroobandt, Dirk, in *Proceedings of the 2006 EPFL LATSIS Symposium* (2006) pp. 139–140
.

[12] M. te Vrugt, An introduction to reservoir computing (2024), arXiv:2412.13212 [cs.ET].

[13] Y. Du, H. Luo, J. Guo, J. Xiao, Y. Yu, and X. Wang, Phys. Rev. E **111**, 035303 (2025).

[14] C. Wringe, M. Trefzer, and S. Stepney, International Journal of Parallel, Emergent and Distributed Systems , 1–39 (2025).

[15] K. Fujii and K. Nakajima, Physical Review Applied **8**, 10.1103/physrevapplied.8.024030 (2017).

[16] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and R. Fazio, npj Quantum Information **5**, 35 (2019).

[17] J. Chen, H. I. Nurdin, and N. Yamamoto, Phys. Rev. Appl. **14**, 024065 (2020).

[18] C. Gyurik, F. Wudarski, E. Philip, A. Sannia, H. Sadeghi, O. Kyriienko, D. Venturelli, and A. A. Gentile, From quantum feature maps to quantum reservoir computing: perspectives and applications (2025), arXiv:2510.01797 [quant-ph].

[19] P. Pfeffer, F. Heyder, and J. Schumacher, Phys. Rev. Res. **4**, 033176 (2022).

[20] Y. Kora and C. Simon, arXiv preprint arXiv:2504.17837 (2025).

[21] Y. Kora, H. Zadeh-Haghighi, T. C. Stewart, K. Heshami, and C. Simon, Frequency- and dissipation-dependent entanglement advantage in spin-network quantum reservoir computing (2024), arXiv:2403.08998 [quant-ph].

[22] N. Götting, F. Lohof, and C. Gies, Phys. Rev. A **108**, 052427 (2023).

[23] J. Settino, L. Salatino, L. Mariani, M. Channab, L. Bozzolo, S. Vallisa, P. Barillà, A. Policicchio, N. L. Gullo, A. Giordano, *et al.*, arXiv e-prints , arXiv (2024).

[24] M. Kornjača, H.-Y. Hu, C. Zhao, J. Wurtz, P. Weinberg, M. Hamdan, A. Zhdanov, S. H. Cantu, H. Zhou, R. A. Bravo, K. Bagnall, J. I. Basham, J. Campo, A. Choukri, R. DeAngelo, P. Frederick, D. Haines, J. Hammett, N. Hsu, M.-G. Hu, F. Huber, P. N. Jepsen, N. Jia, T. Karolyshyn, M. Kwon, J. Long, J. Lopatin, A. Lukin, T. Macrì, O. Marković, L. A. Martínez-Martínez, X. Meng, E. Ostroumov, D. Paquette, J. Robinson, P. S. Rodriguez, A. Singh, N. Sinha, H. Thoreen, N. Wan, D. Waxman-Lenz, T. Wong, K.-H. Wu, P. L. S. Lopes, Y. Boger, N. Gemelke, T. Kitagawa, A. Keesling, X. Gao, A. Bylinskii, S. F. Yelin, F. Liu, and S.-T. Wang, Large-scale quantum reservoir learning with an analog quantum computer (2024), arXiv:2407.02553 [quant-ph].

[25] R. A. Bravo, K. Najafi, X. Gao, and S. F. Yelin, PRX Quantum **3**, 030325 (2022).

[26] D. Beaulieu, M. Kornjača, Z. Krunic, M. Stivaktakis, J. Chen, T. Ehmer, S.-T. Wang, and A. Pham, Journal of Chemical Information and Modeling **65**, 8475 (2025).

[27] G. Llodrà, P. Mujal, R. Zambrini, and G. L. Giorgi, Chaos, Solitons & Fractals **195**, 116289 (2025).

[28] L. C. G. Govia, G. J. Ribeill, G. E. Rowlands, H. K. Krovi, and T. A. Ohki, Phys. Rev. Res. **3**, 013077 (2021).

[29] A. Karimi, H. Zadeh-Haghighi, Y. Kora, and C. Simon, arXiv preprint arXiv:2508.11175 (2025).

[30] W. Hamhoum, S. Cherkaoui, J.-F. Laprade, O. Ahmed, and S. Wang, Multivariate time series forecasting with gate-based quantum reservoir computing on nisq hardware (2025), arXiv:2510.13634 [cs.LG].

[31] Q. Li, C. Mukhopadhyay, A. Bayat, and A. Habibnia, Quantum reservoir computing for realized volatility forecasting (2025), arXiv:2505.13933 [quant-ph].

[32] D. Gilboa, H. Michaeli, D. Soudry, and J. R. McClean, Exponential quantum communication advantage in distributed inference and learning (2024), arXiv:2310.07136 [quant-ph].

[33] J. Ho, G. Moreno, S. Brito, F. Graffitti, C. L. Morrison, R. Nery, A. Pickston, M. Proietti, R. Rabelo, A. Fedrizzi, *et al.*, npj Quantum Information **8**, 13 (2022).

[34] A. Hasegawa, F. L. Gall, and A. Modanese, Maximum separation of quantum communication complexity with and without shared entanglement (2025), arXiv:2505.16457 [quant-ph].

[35] H. Jaeger, GMD Report 152, German National Research Center for Information Technology (2002).

[36] C.-S. Chen and E.-J. Kuo, Unraveling quantum environments: Transformer-assisted learning in lindblad dynamics (2025), arXiv:2505.06928 [quant-ph].

[37] R. Martínez-Peña, J. Nokkala, G. L. Giorgi, R. Zambrini, and M. C. Soriano, Cognitive Computation **15**, 1440 (2023).

[38] Y. Wang, E. Mulvihill, Z. Hu, N. Lyu, S. Shivpuje, Y. Liu, M. B. Soley, E. Geva, V. S. Batista, and S. Kais, Journal of Chemical Theory and Computation **19**, 4851 (2023).
[39] J. M. Miramont, R. Bardenet, P. Chainais, and F. Auger, Benchmarking multi-component signal processing methods in the time-frequency plane (2024), arXiv:2402.08521 [eess.SP].
[40] A. Griffith, *Essential Reservoir Computing*, Ph.D. thesis, ProQuest Dissertations & Theses Global (2021), order No. 29310824.
[41] H. Jaeger, (2002).
[42] T. L. Carroll, Chaos **32 2**, 023123 (2022).
[43] G. Vidal and R. F. Werner, Physical Review A **65**, 10.1103/physreva.65.032314 (2002).