Applications of Artificial Intelligence for Quantum Systems

A thesis submitted to the
Indian Institute of Technology Gandhinagar
for the award of the degree of

Doctor of Philosophy

by

Sreekantham Kumar Rithvik
(Roll No. 19330011)

Under the guidance of

Prof. R.P. Singh
Professor
Atomic Molecular and Optical Physics Division
Physical Research Laboratory, Ahmedabad, India

Department of Physics
Indian Institute of Technology Gandhinagar, India
December 2025

Dedicated to,

Amma and Nanna

CERTIFICATE

This is to certify that the thesis entitled "**Applications of Artificial Intelligence for Quantum Systems**", submitted by **Sreekantham Kumar Rithvik** to **Indian Institute of Technology Gandhinagar**, is a record of bona fide research work under my supervision and guidance, and I consider it worthy of consideration for the award of the degree of *Doctor of Philosophy* of the Institute.

Date: 04-06-2026
Place: AHMEDABAD

Prof. R. P. Singh
Professor
Atomic, Molecular and Optical Physics Division
Physical Research Laboratory Ahmedabad
Gujarat, India

DECLARATION

I certify that

a. the work contained in the thesis is original and has been done by myself under the general supervision of my supervisor.

b. the work has not been submitted to any other institute for any degree or diploma.

c. I have followed the guidelines provided by the institute in writing the thesis.

d. I have conformed to the norms and guidelines given in the ethical code of conduct of the institute.

e. Whenever I have used materials (data, theoretical analysis, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.

f. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Date: 04-06-2026
Place: AHMEDABAD

Sreekantham Kumar Rithvik
Roll No.: 19330011

Acknowledgments

I would like to express my sincere gratitude to my PhD supervisor, Prof. R. P. Singh, for helping in bringing my ideas to fruition and for providing valuable guidance throughout the course of my doctoral research.

I am grateful to the members of my Doctoral Student Committee (DSC) for their time and support. I would like to thank Dr. Shashi Prabhakar for valuable discussions and suggestions, and Dr. Satyajith Seth and Prof. Partha Konar for their support.

I would like to thank my parents, Prof. S. Sreenivasa Murthy and S. Nagalakshmi, for their unwavering support and constant encouragement.

Finally, I would like to thank D. Bharathiganesh and Saurabh Kumar Shukla for their companionship.

Abstract

As quantum technologies evolve from laboratory demonstrations into deployable systems and Artificial Intelligence has matured into a set of tools that can turn high-level intent into computational simulations, data-analysis pipelines, and design workflows, this thesis asks the question *"What can Artificial Intelligence do for Quantum Systems?"* and answers it in three parts.

Nearly all quantum technologies ultimately depend on quantum state characterization, a problem that carries both experimental and computational burdens. In Part I, we begin by leveraging the universal function-approximation capability of *Artificial Neural Networks (ANNs)* to learn the mapping from limited measurement data to the entanglement negativity of higher-dimensional quantum systems, thereby bypassing the need for a large number of measurements and the expensive iterative reconstruction of the density matrix; neural inference then offers a speedup of three orders of magnitude. We then push ANNs to their limit in the next chapter by applying them to the problem of predicting the outputs of *Random Number Generators (RNGs)*. Because this challenge admits no obvious inductive bias in favor of a single architecture, we approach it using 15 different ANN architectures to predict sequences from different types of RNGs at varying sequence sizes. While unprocessed sequences from Pseudo Random Number Generators (PRNGs) are almost completely predictable due to the ANNs' ability to learn the generating algorithm, Quantum Random Number Generators (QRNGs) retain their well-deserved designation as sources of *true* physical randomness. When Toeplitz hashing is applied to process the data, however, all RNG types become immune to neural predictability, thereby aligning with the guarantees implied by the No-Go theorem. Statistical analysis suggests that our multi-architecture neural-network framework can serve as a complementary assessment to the NIST SP 800-22 suite by offering a different perspective on randomness and predictability.

Having applied ANNs to quantum systems, in Part II we turn to their latest incarnation as *Large Language Models (LLMs)*, whose emergence has reshaped the scope and practice of modern artificial intelligence. We begin by assessing the capabilities of 15 LLMs from 5 different providers on 4 categories of quantum-mechanics problem solving,

and we find that while LLMs excel at symbolic reasoning tasks such as derivations and constrained-optimization-based creative tasks, they struggle with numerical problems. By enabling tool usage at roughly three times the token budget, we observe only a modest improvement. A detailed analysis of the responses further shows that LLMs struggle to choose the appropriate formalism for numerical problems. These insights were then applied to develop a multi-agent AI system, Aṇubuddhi, which, by means of a three-layered cognitive architecture, is capable of designing and simulating quantum optics experiments by choosing and arranging the right configuration from a toolbox of optical elements. While this marks a significant improvement over previous automated experiment-design approaches by overcoming the dependence on non-intuitive intermediate representations and by offering more detailed designs, the selected parameters can sometimes be off by orders of magnitude, reflecting LLMs' lack of grounding in actual physical laboratories.

The effectiveness of AI methods for quantum systems, demonstrated in Parts I and II, raises a natural question *"What are the fundamental limits of mechanized reasoning?"*, and we answer this question in the final part of the thesis by revisiting some classical results from earlier centuries. We begin by examining the *Diagonal Argument*, used by Cantor to prove the *uncountability* of real numbers, and note the method's dependence on constructing the diagonal object: a procedure that requires the storage and manipulation of an infinite sequence of infinite-precision numbers. Inspired by the Copenhagen interpretation of quantum mechanics, which insists that one cannot speak meaningfully about physical properties independent of the measurement procedure, we posit that one cannot speak meaningfully about mathematical objects independent of a finite construction procedure and specified up to a finite precision. Under this perspective, we show that numbers with arbitrarily large but specified precision are *countable* by means of a novel canonical bijection with constant-time forward and inverse formulas. Following this, we demonstrate how *classical undecidability* results transform into *bounded decidability* under our *Quantum Inspired Constructive* perspective, indicating that *resource constraints* constitute the primary limitation of mechanized reasoning methods.

Table of Contents

List of Figures

1

Introduction

The previous century bears witness to two profound revolutions in Science and Technology: the Quantum revolution, initiated by Max Planck [1, 2], Albert Einstein[3], and later developed by Werner Heisenberg[4], Niels Bohr[5], Erwin Schrodinger[6], Max Born[7], and other scientific luminaries, altered our perception of the nature of physical reality at the atomic and subatomic level. It shattered the classical conception of an observer-independent reality and replaced it with a fundamentally statistical one, where one cannot speak of the *state* of a system without making repeated measurements on identical copies of it. What was even more remarkable was the fact that the outcomes of these repeated observations also depended on the choice of the measurement apparatus![8–11]. The other profound revolution of the 20th century was the Computer revolution. The theoretical foundations laid by Alan Turing[12] and the stored-program architecture of John von Neumann[13] together laid the practical foundations for mechanized reasoning as envisioned by Gottfried Leibniz[14], George Boole[15], Gottlob Frege[16], Bertrand Russell[17], David Hilbert[18], and other luminaries.

In present times, we see their mature phase, after a century of being nurtured by countless researchers, as both retain their positions among the most important Scientific and Technological fields of our time. Quantum systems have matured from research labs and entered the domain of technology[19, 20], while computing saw exponential growth[21] and is ubiquitous today, with each person having more than 3

computing devices on average globally[22, 23]. The most significant aspect of this is *Artificial Intelligence (AI)*[24], which marks, in some ways, the realization of the dream of mechanized reasoning. The road from the *Universal Computer*[25] to *Large Language Models (LLMs)*[26], a frontier milestone beyond which lie *Agentic Systems*[27] and *Autonomous Machine Intelligence*[28], saw several approaches such as *Automated Theorem Provers*[29], *Rule-Based Expert Systems*[30], *Production Systems*[31], and *Knowledge Graphs*[32], which are commonly referred to as *symbolic approaches* to AI. The other approach, which deals with pattern recognition, is called the *connectionist approach*; it consists of networks of simple units whose connections are governed by statistical learning and optimization. Examples include *Artificial Neural Networks*[33, 34], *Boltzmann Machines*[35], and *Neural/Dynamic fields*[36], where knowledge from training data is encoded in the tunable parameters called *weights*, which are updated based on a *Loss function* that measures the difference between the predicted and true values.

In this *Fourth Industrial Revolution*, marked by the fusion of digital, physical and biological systems, it is natural to ask the question ***What can Artificial Intelligence do for Quantum Systems?***, which exposes a wealth of research opportunities and this is the question that we answer in this thesis in three parts.

Part I deals with two problems, the first being the applications of *Artificial Neural Networks(ANN)/Machine Learning(ML)* for entanglement characterization of bi-partite ququart systems, which are of practical significance in quantum technologies[37, 38]. However, the number of measurements required to accurately characterize these systems with full Quantum State Tomography(QST) typically scales as D^4, D being the dimension of the sub-system, which in practice translates to a large number of measurements as multiple identical measurements have to be performed for each Positive Operator Valued Measure(POVM). Following this, yet another bottleneck has to be scaled in the form of Maximum Likelihood Estimation[39–41] or Bayesian Estimation[42–44] to reconstruct the density matrix from the raw measurement data, which are computationally expensive. Given these challenges, we ask whether Machine Learning techniques can be applied to predict the entanglement of these systems with limited number of measurements and faster inference times. Specifically, we apply three different neural network architectures, namely Multi-Layer Perceptron, Convolutional Neural Network and Transformers to achieve an order of magnitude lower error at just 25% of the to-

tal number of measurements required for full QST and also more remarkably, at 10^3 lower time. ***These sub-second inference times for predicting the entanglement of the higher dimensional quantum system with lower measurements imply that neural networks trained completely with simulation data are capable of rapidly predicting properties of quantum systems without going through a full state reconstruction. The code pertaining to the simulation data, parallelized MLE and Bayesian estimators and the three custom Neural Networks have been released publicly***[45]

The other problem that we address in this part concerns the predictability assessment of Random Number Generators (RNGs), where neural networks, as powerful pattern-recognition systems, can be applied to exploit underlying patterns and thereby predict subsequent bits. Since the hypothesis is that there are no underlying patterns in random sequences, there is no inductive bias that allows us to choose a particular type of neural network, although earlier attempts include Long Short Term Memory (LSTM)[46], Recurrent Neural Networks (RNN)[47, 48], and Convolutional Neural Networks (CNN)[46]. We chose to approach this problem using 15 different types of neural network architectures, including multiple variants of each type as well as novel architectures such as Transformers[49]. We applied these 15 networks to sequences of Pseudo Random Number Generators (PRNGs), Cryptographically Secure Pseudo-Random Number Generators (CS-PRNGs), and Quantum Random Number Generators (QRNGs) with varying sizes ranging from 10,000 to 1,000,000 sequence lengths. Results show that QRNGs are the most resistant to neural-network predictability, followed by CSPRNGs and PRNGs, where prior to post-processing, PRNGs are almost completely predictable, implying that the neural networks completely learn the algorithm (LCRNG) that generates these pseudo-random numbers. Post-processing levels the resistance of all sequences to neural-network predictability. This framework generates sufficient data to analyze predictability behaviours, which indicate that the Multi-Architecture Neural Network framework offers complementary insights to the traditional NIST 800-22[50] statistical tests and identifies Conv1D as the most computationally efficient model and the CNN-LSTM hybrid network as the best model for predictability. ***This positions the multi-architecture framework as a complementary test to evaluate the randomness quality of RNGs***.

Part II analyzes the latest and perhaps one of the most significant developments in the field of AI, namely *Large Language Models*, and assesses their performance with regards to quantum systems in two chapters. To begin with, we systematically analyze the performance of diverse LLMs (fast, mid, and flagship tier models from OpenAI[51–53], Anthropic[54–56], Google[57–59], Alibaba[60–62], and Deepseek[63, 64]) on 4 different task categories, namely Derivations(D), Creative(C), Non-standard(N), and Numerical problems(T). We also compare the token usage and cost per query. Observations include a clear tier stratification based on performance, where flagship models outperform mid tier models at a higher cost, which in turn outperform the fast models, which are the most cost, token, and time efficient. Models, while performing well on derivations and creative tasks, struggle with numerical ones, where even enabling tool usage (enabling python code execution) leads to modest performance improvements (which masks dramatic task-specific variations) at almost 3x token usage. This shows that while models perform generally well on standard quantum mechanics concepts, they struggle with numerical problems as these often involve quantitative analysis, the choice of appropriate numerical methods, and code execution. We also check the reproducibility of these results by running the whole experiment three times while setting the *Temperature* to 0, which makes the models more deterministic, and the results show that fast models vary more than flagship models in terms of their responses to queries. ***This work serves as a compact benchmarking study of the performance of diverse LLMs on different task types in quantum mechanics.***

The experience gained from the previous chapter was then used to develop a multi-agent AI system "Aṇubuddhi " that can design and simulate quantum optics experiments based on natural language conversation. This represents an advancement in the field of *Computer assisted design* of quantum experiments, which previously saw works like MELVIN, AdaQuantum, and PyTheus, all of which depended on counter-intuitive *intermediate representations* that made them harder to adopt. All of these algorithms searched a combinatorially large search space of possible optical configurations from a *Toolbox* of provided optical components. While MELVIN *randomly searched the space of configurations*, evaluating each of them and retaining useful sub-configurations and using them as building blocks for future experiments, AdaQuantum used *Genetic Algorithms* to evolve configurations that would have the highest *fitness function* (configurations that produced desired output states) by means of *mutation and crossover* of configurations

represented as *chromosomes*, wherein the next generation of candidates is produced from a mixing of two chromosomes (including half of each parent). Pytheus, too, relied on representing the modes (path, polarization, or frequency) as nodes and edges as the possible pair-creation processes, with the weights representing the amplitudes of said processes. It then uses a gradient-based optimization to find the *weights* of the edges of the graph, with some additional heuristic search methods if the edges are allowed to be turned on or off. Although these can be mapped to optical physics, the process runs counter to intuition.

It is in this setting that we explore the applicability of LLMs for finding the optimal configuration of optical elements for a quantum experiment. As illustrated by the previous chapters, LLMs have the physics intuition baked into them, having been trained on large quantities of physics text. However, by themselves, LLMs are passive intelligent resources and will not generate anything until prompted. However, when embedded in a *cognitive architecture* consisting of an agentic framework with memory, they can become very powerful tools. We explore this by building a three layered cognitive architecture where the first layer routes the users' natural language query into either *design/chat* mode. Following this, the *designer agent* generates a configuration for the experiment as queried by the user and chooses optical elements from a toolbox of elements whose descriptions alongside the range of parameters are provided to the LLM as context. This makes the process a *Retrieval Augmented Generation*, however, the designs are mostly flawed in the first attempt and are therefore sent to a *reviewer* for validation, which is an LLM (could either be the same one or a different model, in our experiments, we used Claude Sonnet 4.5 as that was one of the latest models at the time). The reviewer's role is to look for problems in the design generated by the designer agent and to suggest improvements. These recommendations become part of the context for the next iteration of the design process. This is repeated for 3 runs, followed by presenting the output to the user along with details about the optical elements chosen for the design. The UI also contains a window that provides a paragraph or two about the experiment and how it is realized by justifying the choice of the components. However, internal validation by itself might not be sufficient and we therefore simulate the various elements of the chosen design and see how the input state, when acted upon by various elements finally evolves into the output state. This is followed by a scoring process where we check for the alignment between the design and the simulation of the experiment. Designs can

be modified by "talking to the agent" through natural language conversation. ***This contributes a multi-agent AI system for designing and simulating quantum optics experiments based on natural-language prompts.***

Part III asks the question *"What are the fundamental limitations of AI methods?"* and to answer this question, we must revisit the *Classical Undecidability* results of the previous century where Kurt Gödel[65] and Alan Turing[66] came up with results in logic, where certain propositions can neither be proved nor disproved in a formal system. These results become important for us to re-examine because, as we saw in Part II, LLMs, which are at the pinnacle of mechanized reasoning, seem to get better as the number of tunable parameters and training data increase[67–69]. In this setting, the classical undecidability results position themselves as an *insurmountable barrier* that cannot be overcome no matter the size of the computational resources that can be invested. Surprisingly, it is Quantum Mechanics that provides us with the inspiration to address this problem, and in this thesis, we argue that: ***Just as we cannot speak of physical quantities prior to measurement[70], we cannot speak of mathematical objects prior to construction[71] up to a finite precision***. The explicit need to specify *finite precision* becomes apparent when one closely examines the classical undecidability results, all of which have the *Diagonal Argument*[72] as their basis. The first instance of this argument was used to prove the *Uncountability of real numbers*, where the proof proceeds by enumerating *all possible combinations of digits* in a table that is assumed to be complete. This is followed by the construction of a diagonal candidate whose elements are constructed in such a way that they differ from the diagonal element of the completed enumeration matrix, thereby constructing a candidate that lies *outside* this *completed enumeration*, which leads to a contradiction. The conclusion that Cantor draws from this result is that the set of *all possible reals is never complete or that the reals can never be enumerated*. However, to come to this conclusion, Cantor implicitly assumes the construction of a diagonal candidate whose elements differ from every element of a *completed enumeration* by construction. In order to actually construct such an element would require a computer with infinite memory and time to store and then manipulate the full sequence of possible numbers to create the diagonal element. In reality, though, one always encounters finite-precision numbers in any computation, whether involving physical quantities, measurements, or financial transactions, and it is only possible to perform meaningful calculations when the numbers

can be faithfully stored and manipulated by a finite-precision machine. This also stands in contrast with Gödel's and Turing's formalisms, which involve infinite enumerations of formulas by means of their Gödel numbers and the infinite-resource Turing Machine, not bounded by Space, Precision, Time, etc. This represents the *Platonic ideal* view of mathematics[73], which posits that abstract mathematical objects exist independent of the language, thought, and practices of intelligent agents[74]. ***In this part of the thesis, we explore how classical undecidability results change when we restrict ourselves to finite resources.***

Part I

Machine Learning Approaches to Quantum Systems

"Entanglement is perhaps the most non-classical feature of quantum mechanics."

Erwin Schrödinger

2

Machine Learning Approaches for Quantum Entanglement Characterization[1]

Quantum Entanglement, a term first coined by Erwin Schrödinger[75] in response to the Einstein-Podolsky-Rosen (EPR) paradox, marks a fundamental departure from the classical-physics view of reality in terms of *local realism*, by positing physical systems that have fundamentally *non-local correlations* even when the subsystems are separated by long distances. What started as a theoretical paradox in the previous century has now matured into a *critical resource* for quantum technologies[76–83]. Of particular interest are *high-dimensional quantum states* due to increased information capacity and enhanced robustness against certain types of noise[84–86]. These states can be created and manipulated through the Orbital Angular Momentum (OAM) states of light[87–91] and have been applied in quantum technologies[37, 38, 92–94].

The inherently statistical nature of quantum systems means that ascertaining the state requires repeated measurements on identical copies of the system. Combine this with the number of measurements required to perform full *Quantum State Tomography*

[1]The contents of this chapter have been presented in: S. K. Rithvik, R. P. Singh, Shashi Prabhakar "Machine Learning-Enhanced Entanglement Characterization in Bi-partite Ququart Systems," Research Square, https://doi.org/10.21203/rs.3.rs-6486345/v1 (2025).

and the computational bottleneck of trying to reconstruct the density matrix of the state from the measurement frequencies[39, 40, 42–44], and we are looking at a formidable challenge in quantum state characterization. The total number of measurements required to characterize the system goes like $N_{mub}^2 D^2 N_{cop}$, where N_{mub} is the number of mutually unbiased bases, D is the subsystem dimensionality, and N_{cop} is the number of identical copies needed for measurement statistics. We are particularly interested in bipartite ququart states, and this brings the number to 400 distinct measurement settings, each requiring multiple copies N_{cop}.

In order to address these challenges in state characterization, we turn to Artificial Neural Networks (ANNs), given their ability to learn multivariate function mappings from high-dimensional inputs to outputs, given a sufficient amount of data and network expressibility in terms of the number of tunable parameters (weights)[95]. Indeed, recent years have seen the application of ANNs to the problem of quantum state characterization[96–101]. However, for higher-dimensional states like bipartite ququarts, it remains an open question whether sufficiently expressible neural networks can indeed be reliably used for quantum state characterization with reduced measurements and faster inference, given that, once trained, a neural network's forward pass is relatively computationally inexpensive. We address this question by customizing three popular architectures, namely the Multi-Layer Perceptron (MLP)[102], Convolutional Neural Network (CNN)[103–105], and Transformer[49], adapting them for the problem of entanglement characterization of bipartite ququart systems. To assess their performance, we compare them with parallelized versions of the Maximum Likelihood Estimator (MLE)[39–41] and Bayesian estimator[42–44].

2.1 Theoretical Framework

The inability to express a quantum state as a tensor product of its subsystems is referred to as *quantum entanglement* ($\rho \neq \sum_i p_i \rho_A^i \otimes \rho_B^i$). The Peres-Horodecki criterion[106, 107] posits that for a state to be separable, its partial transpose should not contain negative eigenvalues. The monotone that readily captures this property is called *entanglement negativity*:

$$\mathcal{N}(\rho) = \frac{||\rho^{\Gamma_A}||_1 - 1}{2} \tag{2.1}$$

where ρ^{Γ_A} represents the partial transpose of the density matrix with respect to subsystem A, and $||X||_1 = \mathrm{Tr}\sqrt{X^\dagger X}$ is the trace norm.

The partial transpose operation acts on the density matrix elements as:

$$(\rho^{\Gamma_A})_{ij,kl} = \rho_{kj,il} \tag{2.2}$$

where i, k index the basis states of subsystem A, and j, l index the basis states of subsystem B.

$\mathcal{N}(\rho)$ provides a continuous measure of entanglement that ranges from 0 for separable states to 0.5 for maximally entangled qubit pairs, and can reach higher values (up to 1.5) for maximally entangled ququart systems. It turns out that $\mathcal{N}(\rho) = \sum_i |\lambda_i^-|$, where λ_i^- are the negative eigenvalues.

2.1.1 Measurement Framework and POVMs

The density matrix ρ of the quantum system is a theoretical representation that contains all relevant information about the system. Experimentally, a set of Positive Operator Valued Measures (POVMs) is employed, with repetitions on identical copies of the physical system, to reconstruct the density matrix of the system. We begin with the Mutually Unbiased Bases (MUBs) and then construct the POVMs as tensor products of the basis vectors of the subsystems.

The first basis is the standard computational basis:

$$M_0 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

The second basis introduces equal superpositions with varying signs:

$$M_1 = \frac{1}{2}\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix} \tag{2.4}$$

The remaining three bases incorporate complex phases to ensure mutual unbiasedness:

$$M_2 = \frac{1}{2}\begin{pmatrix} 1 & -1 & -i & -i \\ 1 & -1 & i & i \\ 1 & 1 & i & -i \\ 1 & 1 & -i & i \end{pmatrix} \tag{2.5}$$

$$M_3 = \frac{1}{2}\begin{pmatrix} 1 & -i & -i & -1 \\ 1 & -i & i & 1 \\ 1 & i & i & -1 \\ 1 & i & -i & 1 \end{pmatrix} \tag{2.6}$$

$$M_4 = \frac{1}{2}\begin{pmatrix} 1 & -i & -1 & -i \\ 1 & -i & 1 & i \\ 1 & i & -1 & i \\ 1 & i & 1 & -i \end{pmatrix} \tag{2.7}$$

These bases satisfy the mutual unbiasedness condition:

$$|\langle\psi_i^{(k)}|\psi_j^{(l)}\rangle|^2 = \frac{1}{d} \tag{2.8}$$

for all bases $k \neq l$ and all basis vectors i, j, where $d = 4$ is the dimension. This property ensures that a state that is definite in one basis is maximally uncertain in all other bases, providing complementary information with each measurement.

To transform these measurement bases into experimentally relevant observables for our bipartite system, we construct positive operator-valued measures (POVMs) through tensor products. The full set of potential measurements consists of all combinations of basis vectors from the two subsystems, resulting in a total of $5 \times 5 \times 4 \times 4 = 400$

distinct POVM elements.

Each POVM element takes the form of a projection operator:

$$\Pi_{ij} = (|v_i\rangle \otimes |w_j\rangle)(\langle v_i| \otimes \langle w_j|) \tag{2.9}$$

where $|v_i\rangle$ is a basis vector from subsystem A and $|w_j\rangle$ is a basis vector from subsystem B.

The probability of observing a particular measurement outcome is given by the Born rule:

$$p_{ij} = \mathrm{Tr}(\rho \Pi_{ij}) \tag{2.10}$$

For resource-efficient tomography, it becomes crucial to note that not all measurements provide equal information about the entanglement properties of the state. We can rank the POVMs by their information content, quantified by the spread of their eigenvalues.

$$\text{Information Rank} \propto \lambda_{\max}(\Pi_{ij}) - \lambda_{\min}(\Pi_{ij}) \tag{2.11}$$

Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) represents the gold standard in quantum state tomography[39–41]. The central principle of MLE is to find the physical density matrix that maximizes the likelihood of the observed measurement outcomes. For a set of POVM elements $\{\Pi_i\}$ and corresponding measurement frequencies $\{f_i\}$, the likelihood function takes the form:

$$L(\rho) = \prod_i p_i^{n_i} = \prod_i [\mathrm{Tr}(\rho \Pi_i)]^{n_i} \tag{2.12}$$

where n_i is the number of times outcome i was observed. In practice, it is often more convenient to work with the log-likelihood:

$$\log L(\rho) = \sum_i n_i \log[\mathrm{Tr}(\rho \Pi_i)] \tag{2.13}$$

The MLE problem can then be stated as:

$$\begin{aligned} \text{maximize} \quad & \log L(\rho) \\ \text{subject to} \quad & \rho \geq 0 \\ & \text{Tr}(\rho) = 1 \end{aligned} \tag{2.14}$$

where the constraints ensure that the resulting density matrix is physically valid.

Several numerical approaches exist for solving this constrained optimization problem. A particularly effective iterative algorithm for quantum state tomography is the diluted iterative algorithm[39, 108], which updates the estimate of ρ according to:

$$\rho^{(k+1)} = \frac{R(\rho^{(k)})\rho^{(k)}R(\rho^{(k)})}{\text{Tr}[R(\rho^{(k)})\rho^{(k)}R(\rho^{(k)})]} \tag{2.15}$$

where the operator $R(\rho)$ is defined as:

$$R(\rho) = \sum_i \frac{f_i}{\text{Tr}(\rho\Pi_i)}\Pi_i \tag{2.16}$$

This iterative process continues until convergence, typically assessed by monitoring changes in the log-likelihood or in the estimated density matrix itself.

To improve the stability of the optimization process, we run the optimizer from several reasonable starting points, including the maximally mixed state and random physical states, keeping the solution with the highest final log-likelihood. We also smooth the iterative update using a momentum-like average of successive R operators. We apply a cautious step-size rule that only increases the update when the likelihood improves. To avoid numerical drift and overfitting to noise, we weakly mix each iterate with the maximally mixed state. This means that we apply $\rho \leftarrow (1-\lambda)\rho + \lambda I/d$ with a small λ. We also use early stopping when the likelihood plateaus, with a limit of 9000 iterations overall.

Bayesian Estimation

As a second baseline, we consider Bayesian state estimation. We treat the unknown density matrix as a random variable with a prior $P(\rho)$, and update it using the observed data D[42–44]. The target is the posterior $P(\rho \mid D) \propto P(D \mid \rho)P(\rho)$. The likelihood follows from the Born probabilities $p_i(\rho) = \mathrm{Tr}(\rho\Pi_i)$, for counts n_i we use $P(D \mid \rho) = \prod_i p_i(\rho)^{n_i}$ (equivalently $\log P(D \mid \rho) = \sum_i n_i \log p_i$). Exact posterior inference for ququart tomography is costly, so we use a practical maximum-entropy style iterative reconstruction. It starts from $\rho_0 = I/d$ and repeatedly reweights the estimate using the measured frequencies: $\rho_{t+1} \propto R(\rho_t)\,\rho_t\,R(\rho_t)^\dagger$. Each step is explicitly renormalized, and we add a small mixing regularizer $\rho \leftarrow (1-\lambda)\rho + \lambda I/d$ to suppress numerical drift[41, 109, 110].

2.2 Machine Learning Approaches

Machine learning provides a different route to entanglement characterization. Instead of reconstructing the full density matrix, we train models to map measurement outcomes directly to an entanglement metric. This can reduce both measurement overhead and post-processing time. In this section we describe the architectures we use, the physics-motivated enhancements, and the data generation and training pipeline.

2.2.1 Neural Network Architectures

Neural networks let us learn this mapping from data. The input is a vector of measurement outcomes, and the output is the estimated negativity. Once trained, inference is just a fast forward pass, and it avoids iterative reconstruction.

We use three neural architectures: a Multi-Layer Perceptron (MLP), a Convolutional Neural Network (CNN), and a Transformer. Each model trades off capacity, inductive bias, and data efficiency.

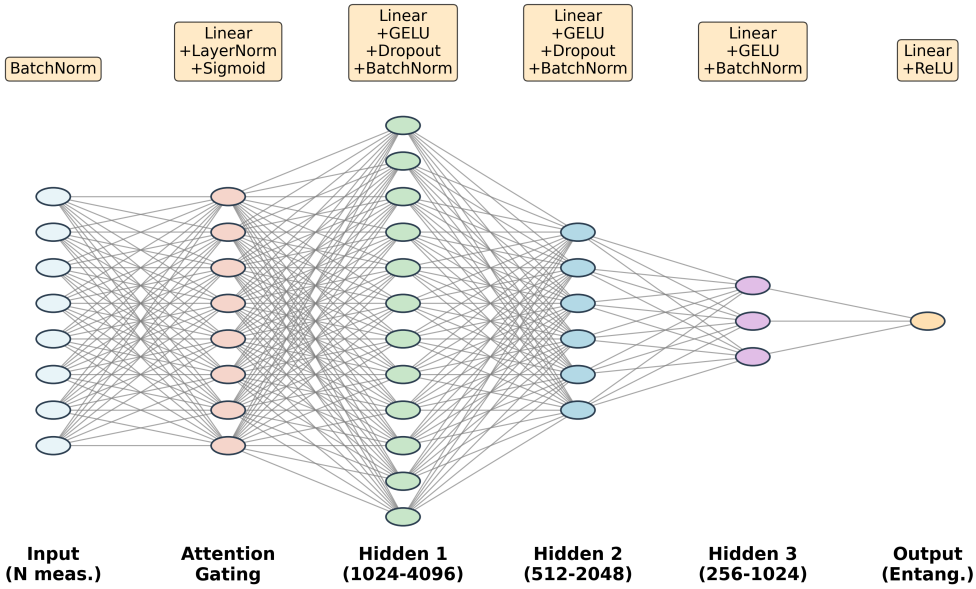


Figure 2.1: Architectural diagram of the enhanced Multi-Layer Perceptron (MLP) for entanglement estimation. The network combines input normalization, measurement-aware attention, and a progressively narrowing hidden layer structure to map from raw measurement data to entanglement negativity.

Multi-Layer Perceptron

The Multi-Layer Perceptron (MLP)[102] serves as a simple baseline. The input is a vector of measurement outcomes, and the output is a single scalar (negativity). We keep the model stable across measurement regimes by normalizing inputs, adapting layer widths to the number of measurements, and using light regularization.

The key components of our enhanced MLP architecture include: We first normalize measurements using a simple scaling that accounts for measurement count,

$$x_{\text{norm}} = \frac{x - \mu}{\sigma\sqrt{n_{\text{measurements}}}}, \tag{2.17}$$

followed by batch normalization. To avoid treating all measurements as equally informative, we include a lightweight attention-style gating: $w = \sigma(\text{LayerNorm}(W_a x_{\text{norm}} + b_a))$ and $x \leftarrow x_{\text{norm}} \odot w$. The hidden width scales with the input size,

$$h_1 = \max(1024, \min(4096, 32\, n_{\text{measurements}})), \tag{2.18}$$

and we use GELU activations throughout. Regularization is modest and measurement-

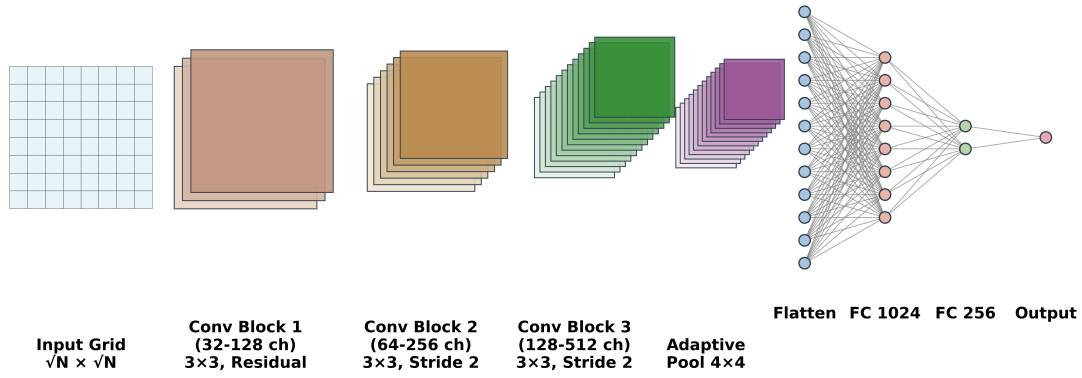


Figure 2.2: Architectural diagram of the Convolutional Neural Network (CNN) for entanglement estimation. The network adaptively reshapes measurements into 2D grids, then processes through three residual blocks with progressive downsampling and channel expansion before regression.

dependent ($p_{\text{dropout}} = \max(0.05, p_0 - 0.001\, n_{\text{measurements}})$), and the final layer uses a ReLU so the predicted negativity is non-negative.

For training, we use a simple composite regression loss,

$$\mathcal{L} = \text{MSE} + 0.01\,\text{RelError} + 0.05\,\text{L1Loss}, \tag{2.19}$$

and scale the learning rate roughly as $n_{\text{measurements}}^{-0.2}$ with cosine decay and warmup. Early stopping is used when validation performance plateaus, and gradient accumulation is applied when needed to keep the effective batch size stable.

Convolutional Neural Network

While an MLP treats the input as an unstructured vector, a Convolutional Neural Network (CNN)[103–105] can exploit local structure after we reshape the measurements into a 2D grid. This is a pragmatic choice (not a claim about the underlying tensor structure), but it provides a useful inductive bias when the number of measurements is moderate.

We map an input vector of length input_size to a square grid of side grid_dim =

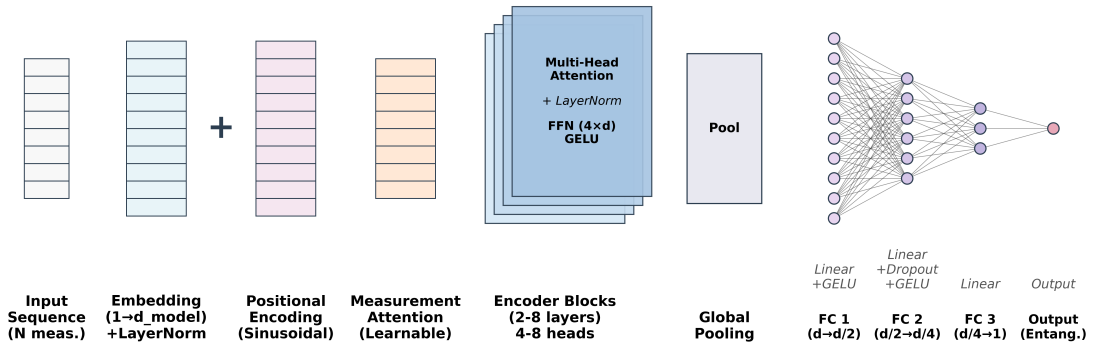


Figure 2.3: Architectural diagram of the Measurement-Adaptive Transformer for entanglement estimation. The network employs dynamic scaling, learnable measurement attention, and multi-head self-attention to identify correlational patterns in quantum measurement data.

$\lceil\sqrt{\text{input\_size}}\rceil$ by zero-padding and reshaping. The number of channels is adapted to the regime via $\text{base\_filters} = \min(128, \max(32, \text{input\_size}/2))$. The network then applies a small stack of residual blocks with downsampling (roughly $\text{base\_filters} \to 2\,\text{base\_filters} \to 4\,\text{base\_filters}$), followed by adaptive average pooling to a fixed 4×4 feature map. A compact fully-connected head outputs the negativity, again with a ReLU at the end. As with the MLP, dropout is decreased as more measurements are available, $p = \max(0.05, 0.2 - 0.001\,\text{input\_size})$. In practice, this convolutional bias tends to work well in intermediate measurement regimes, where local patterns in the reshaped grid are informative.

Transformer Architecture

For our most sophisticated approach, we employ a dynamically-scaling Transformer architecture[49] that leverages self-attention mechanisms to capture complex dependencies in quantum measurement data. Transformers have shown remarkable success in quantum state reconstruction tasks[111], making them particularly well-suited for entanglement estimation. Our implementation adapts key architectural parameters based on measurement count to maintain optimal performance across diverse measurement regimes.

We use a measurement-adaptive Transformer[49] to capture long-range dependencies between measurement outcomes. The embedding dimension grows with measurement count,

$$d_{\text{embed}} = \min(512, \max(128, 4N_{\text{meas}})), \tag{2.20}$$

and the model depth and number of heads are scaled accordingly ($n_{\text{heads}} = \min(8, \max(4, d_{\text{embed}}/64))$ and $n_{\text{layers}} = \min(8, \max(2, N_{\text{meas}}/32))$). We also include simple per-measurement gates $x \leftarrow x \odot \sigma(W_{\text{meas}})$ so the model can down-weight consistently uninformative inputs.

The multi-head self-attention mechanism enables the Transformer to discover complex measurement correlations through parallel attention patterns. Each head captures distinct relationships:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.21}$$

where Q, K, and V are query, key, and value projections, allowing the network to learn which measurement combinations provide complementary entanglement information.

We pool the transformer outputs into a global representation and pass it through a small regression head with GELU and LayerNorm. Dropout is again reduced as measurements increase, $p_{\text{out}} = \max(0.05, 0.1 - 0.0002\, N_{\text{meas}})$, and the final activation is a ReLU to keep the predicted negativity non-negative. Empirically, attention-based models are most useful when measurements are sparse and the relevant signal is distributed across many weak features.

2.2.2 Data Generation and Training

We train the models on simulated data so that the ground-truth negativity is known exactly and we can probe different measurement budgets in a controlled way. We sample pure states by drawing random complex amplitudes c_{ij} and forming $|\psi\rangle = \sum_{i,j} c_{ij}|i\rangle_A \otimes |j\rangle_B$, with $\langle\psi|\psi\rangle = 1$. We also generate mixed states by convex mixing, $\rho = p\,\rho_{\text{pure}} + (1-p)\,I/d^2$. Physicality is enforced throughout ($\text{Tr}(\rho) = 1$, $\rho \geq 0$, $\rho = \rho^\dagger$) and stratification of the samples is done to cover both weakly entangled and highly entangled regimes. For each state we simulate measurements using the Born rule, $p_{ijkl} = \text{Tr}(\rho\Pi_{ijkl})$ and add finite-shot noise by drawing $n_{ijkl} \sim \text{Binomial}(N_{\text{shots}}, p_{ijkl})$,

with $N_{\text{shots}} = 200$. We then evaluate acquisition regimes from 10 to 400 POVMs using the same information-rank based selection. In total we generate 15,000 states and split them 80%/10%/10%, yielding 12,000 training states, 1,500 validation and 1,500 test states. We use a composite loss $\mathcal{L} = \text{MSE} + 0.01\,|(\hat{\mathcal{N}} - \mathcal{N})/(\mathcal{N} + \epsilon)| + 0.05\,\text{L1}$ with $\epsilon = 10^{-6}$. We use AdamW optimizer with cosine decay and warmup. To keep optimization stable across measurement counts, we use scaling rules for learning rate $\propto N_{\text{meas}}^{-\alpha}$ and batch size $\min(256, \max(16, 8192/N_{\text{meas}}))$. Early stopping with patience is implemented to stop the training when validation performance plateaus. All models are implemented in PyTorch and trained on a single NVIDIA RTX 4090 (24GB). We set a maximum budget of 2000 epochs, but early stopping typically ends training earlier depending on the measurement regime. The traditional methods (MLE, Bayesian estimation) are parallelized on a CPU with 32 cores and 32GB RAM

2.3 Computational Results and Analysis

In this section, we present the results comparing the performance of our neural-network approaches (MLP, CNN, Transformer) against traditional Maximum Likelihood Estimation (MLE) and Bayesian methods for entanglement characterization in bipartite ququart systems. We evaluate the models' performance by comparing the *Mean Squared Error (MSE) and estimation time* at varying numbers of measurements to assess the reliability of the methods at different levels of incomplete characterization.

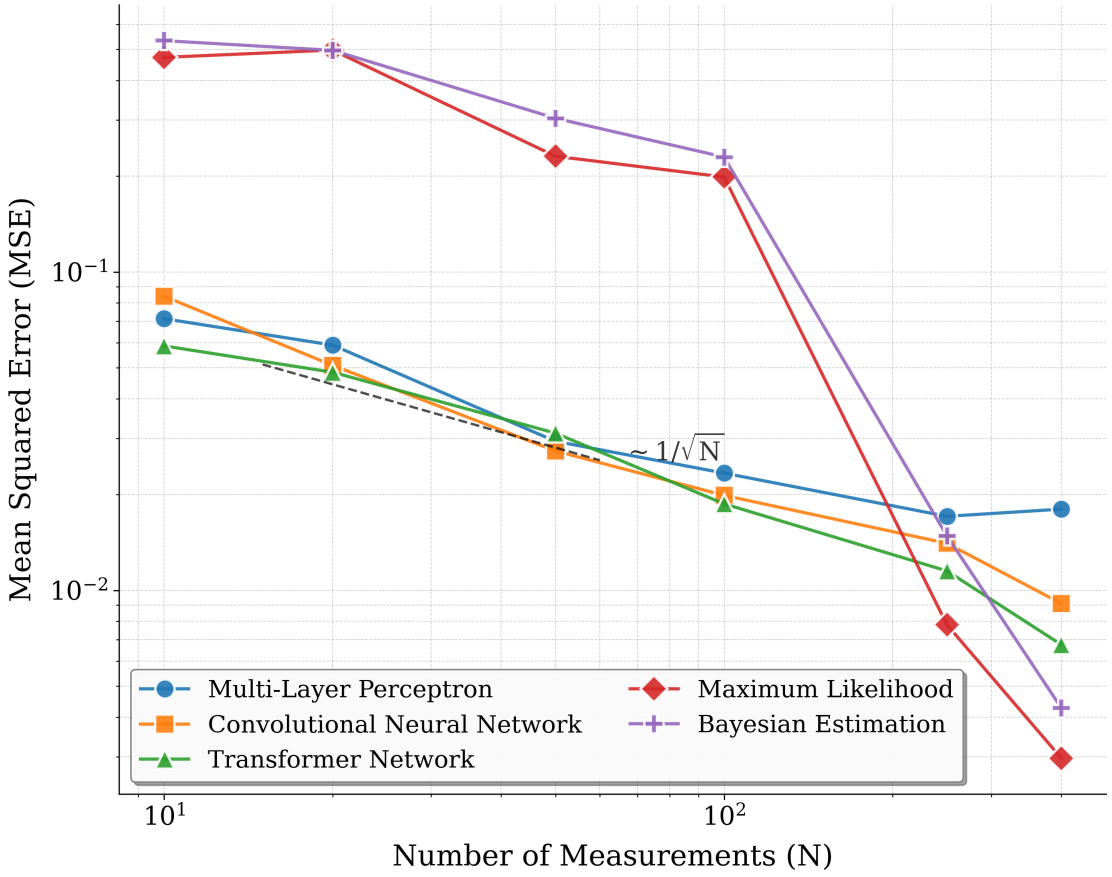


Figure 2.4: MSE in entanglement negativity estimation as a function of number of measurements. Results show Neural Networks outperform traditional methods in the low and intermediate regimes

In the regime of low measurement counts (10 or 20 out of the full 400), we notice neural networks giving estimates that are an order of magnitude better than the traditional methods. However, for a more meaningful comparison, we require more measurements. In the intermediate regime (50 or 100 out of the full 400), we notice that the Transformer achieves an MSE of 1.87×10^{-2} at 100 measurements, compared to 1.99×10^{-2} for CNN and 2.34×10^{-2} for MLP. Traditional methods perform poorly in comparison: 1.99×10^{-1} for MLE and 2.30×10^{-1} for Bayesian methods. In the high-measurement regime, we notice a crossover in Figure 2.4, indicating that traditional methods provide better estimates when sufficiently large numbers of measurements (frequencies for each outcome associated with a POVM when sampled) are available. However, computational efficiency offers a different perspective.

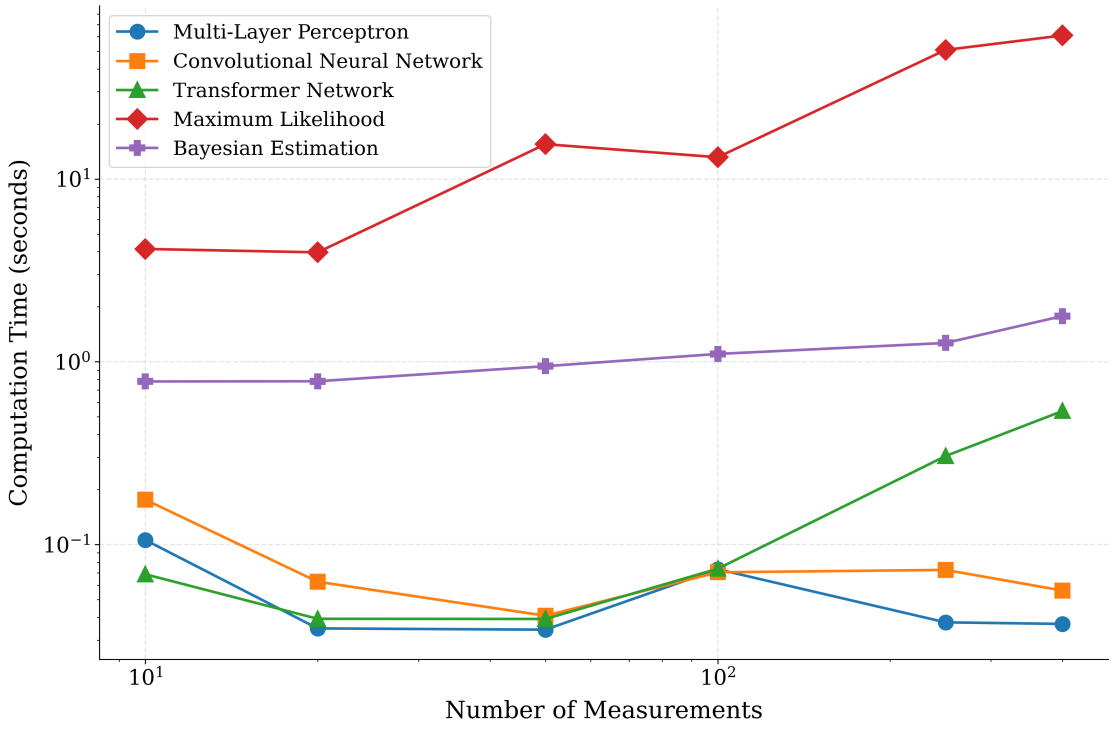


Figure 2.5: Estimation time vs number of measurements for various methods. Neural methods are fast irrespective of the measurement count while the traditional methods show steep scaling

The computational efficiency comparison (Figure 2.5) reveals dramatic differences. For a test set consisting of entanglement-negativity estimates for 1500 states, MLP delivers the fastest predictions (0.03-0.1 seconds) with near-constant time scaling, CNN shows consistent performance (0.04-0.18 seconds) with minimal measurement dependency, and the Transformer scales gracefully (0.04-0.54 seconds) while providing superior accuracy. Maximum Likelihood Estimation shows aggressive time growth from 3.97 seconds at low measurement counts to 60.95 seconds at 400 measurements, becoming prohibitive for real-time applications, particularly when the number of states to be characterized grows (in this work, the test set is 1500 states, but real applications could require the characterization of millions of quantum states). The Bayesian approach, while faster than MLE, still scales poorly from 0.78 seconds to 1.78 seconds across measurement regimes. At 400 measurements, neural networks provide dramatic speedups: the Transformer completes in 0.54 seconds (113× faster than MLE), while MLP and CNN finish in 0.037 and 0.056 seconds, respectively (1657× and 1088× faster than MLE).

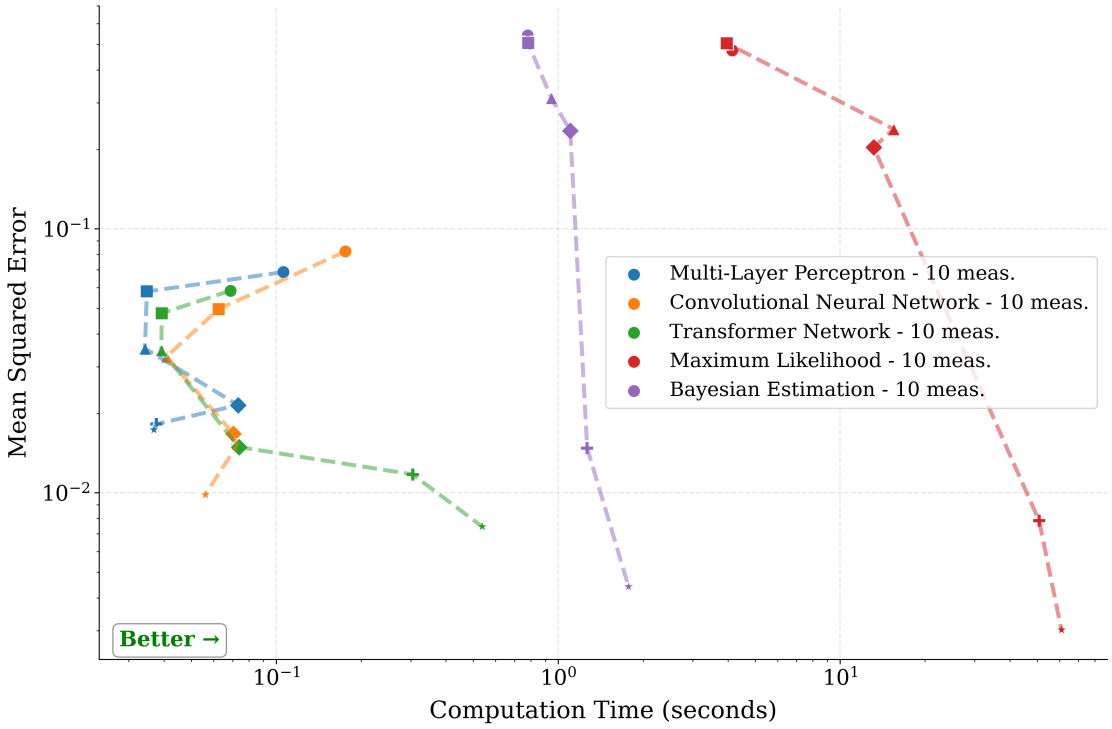


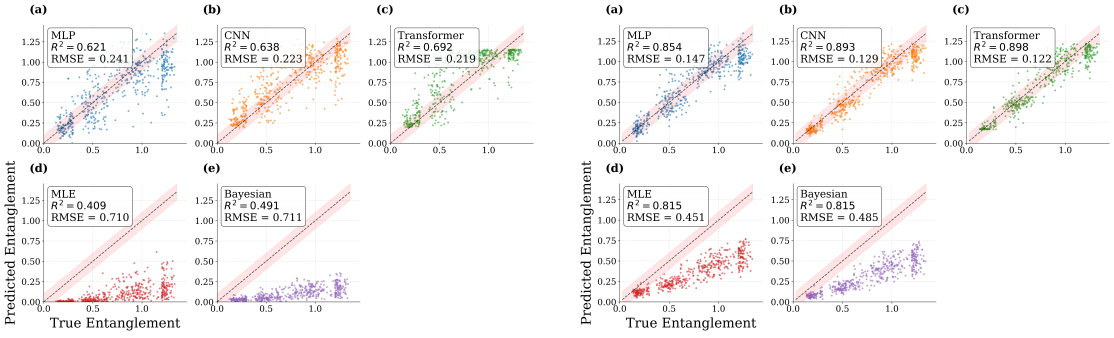
Figure 2.6: Efficiency frontier comparing mean squared error (MSE) versus computation time for different methods. Points show results for measurement counts from 10 to 400. Lower-left region represents better performance (lower error, faster computation). Neural methods consistently occupy the optimal region of this frontier.

The efficiency frontier in Figure 2.6 reveals neural networks consistently occupying the optimal (lower-left) region in the plot, signifying low errors and faster calculation. Neural methods offer calculations that are $10^3 x$ faster than those of traditional methods and consistently provide better estimates in the low- and intermediate-measurement regimes. In the high-measurement regime, the traditional methods offer marginal improvements compared to the neural methods at a substantially higher computational estimation-time cost, which makes them prohibitively expensive if rapid characterization is desired with high data throughput.

2.3.1 Prediction Accuracy Across Entanglement Ranges

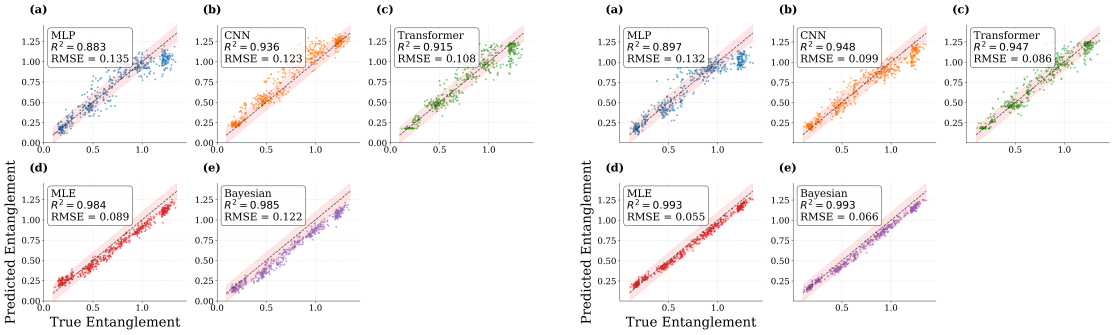
Given that we have segregated the entanglement ranges into 6 separate bins ranging from 0.0 to 1.5 in negativity, it is interesting to observe the prediction performance of the various methods across different measurement ranges. It was observed that the traditional methods, MLE and Bayesian Estimation, *systematically underestimate* the

entanglement negativity, owing to the fact that full state reconstruction is an iterative process and can only succeed with sufficient information (in terms of frequencies) made available to the estimators. The neural estimators do not seem to suffer from this, as they have been trained with simulation data that was carefully chosen to represent the full range of entanglement bins. If we were to re-segregate the available bins into three distinct regimes, the neural methods provide essentially unbiased predictions (-0.009 bias) in the weakly entangled (0.0-0.3) and moderately entangled (0.3-0.9) regimes (-0.001 bias), while having a slight negative bias (-0.065) in the highly entangled regime at 100 measurements. By contrast, the traditional methods show a negative bias that is one to two orders of magnitude larger at 100 measurements.



(a) 20 measurements

(b) 100 measurements



(c) 250 measurements

(d) 400 measurements

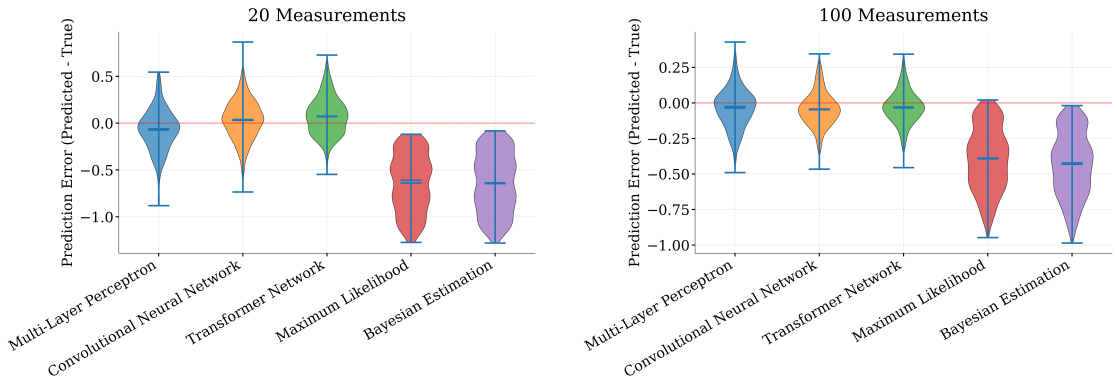
Figure 2.7: Comparison between real vs predicted entanglement negativity : each subplot shows results for (a) 20, (b) 100, (c) 250, and (d) 400 measurements. Neural methods offer better prediction than MLE and Bayesian estimators while the latter suffer from systematic biases

To quantify prediction accuracy, we employ the coefficient of determination (R^2), defined as:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}} \tag{2.22}$$

where y_i are the true negativity values, $\hat{y}_i$ are the predicted values, $\bar{y}$ is the mean of true

values, SS_{res} is the sum of squares of residuals, and SS_{tot} is the total sum of squares. R^2 measures the proportion of variance in the target variable explained by the model, with values near 1 indicating excellent predictive performance and negative values indicating performance worse than simply predicting the mean value for all samples.



(a) 20 measurements

(b) 100 measurements

Figure 2.8: Distribution of the prediction errors as illustrated by violin plots at (a) 20 and (b) 100 measurements. The central line of the violin indicates the median error while the width corresponds to the frequency of errors at that magnitude. Traditional methods show a clear systematic bias while neural methods show more symmetric error distributions with the Transformer demonstrating the most concentrated error distribution with 67.6% of errors below $|0.2|$ and 16.4% within ± 0.05 of true values.

When combined, the scatter plots and the violin plots of the error distributions tell a common story: neural methods provide better estimates than traditional methods, with significantly lower systematic bias across all entanglement ranges and measurement regimes.

2.4 Conclusion

This chapter investigates whether neural networks can be employed for the reliable prediction of the entanglement negativity of higher-dimensional quantum states without a full state reconstruction from limited measurement data, and the analysis of the results indicates that neural estimators provide very high-accuracy estimates (low MSE) with incomplete measurements. The prediction and error analysis reveal a systematic bias in the predictions of the traditional estimators. The reason why the bias always seems to be negative has to do with how the traditional methods usually choose a maximally mixed (low-entanglement) state as the starting point of the iteration and cannot provide an

accurate (high enough) estimate when they are starved of the number of measurement frequencies made available to them, whereas neural estimators seem to benefit from the vast quantities of well-spread-out simulation data (by design, across the entanglement-bin ranges), thereby providing better estimates at all ranges.

Neural methods offer sub-second inference for the prediction of the entanglement negativity of $\approx 10^3$ states, which translates to a $10^3\times$ speedup when compared to traditional iterative density-matrix reconstruction methods. These results posit neural estimators as the preferred choice when rapid characterization of quantum states is desired in high-data-throughput scenarios, an emerging trend in modern quantum technologies. Future research can test the efficacy of neural predictors as applied to multipartite systems and other quantum properties of these systems.

"Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin."

John von Neumann

3

Neural Network-Based Assessment of Random Number Generator Predictability[1]

Having established the ability of neural networks to *learn* complex mappings from measurement frequencies to higher-dimensional quantum states, we now turn to the topic of *randomness*. Several scientific and technological applications, such as Monte Carlo simulations, cryptographic security, blockchain protocols, machine learning, and fairness-based systems like lotteries, rely on random number generation[112–114]. Given the role of neural networks as powerful pattern recognizers, the most challenging problem for them should be to predict the next sequences of bits given a large set of previous sequences, and it is this problem that we explore in this chapter.

There are broadly two ways of generating random-number sequences: one based on physical processes and the other based on algorithms. Physical randomness can be divided into classical and quantum sources. Classical physical sources, such as thermal noise, electronic shot noise, or chaotic systems, are governed by deterministic laws. However, their extreme sensitivity to initial conditions and unavoidable measurement noise (due to the finite least count of detectors) cause tiny uncertainties to grow rapidly,

[1]The contents of this chapter have been presented in a manuscript submitted for publication.

making their outcomes practically unpredictable (as computers have finite precision). Quantum randomness, on the other hand, arises from fundamentally indeterminate processes such as single-photon detection at a beam splitter, quantum vacuum fluctuations measured in a homodyne detection, and radioactive decay processes, where quantum theory predicts only probabilities for each outcome and no additional information can determine the result in advance. For this reason, Quantum Random Number Generators (QRNGs) are called *True Random Number Generators (TRNGs)*. Algorithmic approaches generate pseudorandom numbers (PRNGs) using deterministic rules, such as linear congruential generators (LC-RNGs) or cryptographically secure pseudorandom number generators (CSPRNGs), which rely on an initial seed and produce sequences that appear random but are fully reproducible if the seed is known.

While traditional statistical tests like the NIST Statistical Test Suite and Diehard tests provide valuable insights into the statistical properties of random sequences [115, 116], it is interesting to see whether neural networks might be able to uncover complex hidden patterns that are inaccessible to these tests. Unlike tasks like image classification or natural language processing, which suggest specific architectures like CNNs or Transformers because of the structural regularities in the data, the problem of predicting random sequences does not offer any inductive bias that privileges a particular kind of neural network. Therefore, it is not known what the *best candidate* might be for this problem, although previous attempts have seen the application of Long Short-Term Memory (LSTM)[46], Recurrent Neural Networks (RNNs)[47, 48], and Convolutional Neural Networks (CNNs)[46]. We chose to approach this problem using 15 customized neural network architectures spanning recurrent, convolutional, attention-based, and hybrid models, across varying input-sequence sizes made available during training (we compare neural networks trained with 1000, 10,000, 100,000, and 1,000,000 sequences of random bits).

3.1 Methodology

The task is defined as follows: given *Span* bytes of input sequences (constructed via $sequence_i = bitstream[i : i + (span + 1) \times 8]$), predict the next byte, which makes it a classification problem for integers $[0, 255]$. To learn patterns from the data, we

train neural networks with varying sequence sizes (1K to 1M sequences) to observe how predictability scales with the available input sequences. We also have a Dynamic Span Adaptation where the span varies according to the training-dataset size as $\text{span}_{\text{dynamic}} = \left\lfloor \frac{\text{span}_{\text{base}} \times \sqrt{N_{\text{train}}}}{\sqrt{1000}} \right\rfloor_8$ where $N_{\text{train}} = 0.6 \times N_{\text{total}}$ denotes training-dataset magnitude, incorporating constraints $\text{span}_{\text{min}} = 24$ bits and $\text{span}_{\text{max}} = 96$ bits, thereby maintaining computational tractability while preserving statistical validity. This is done to ensure computational efficiency and statistical power optimization.

Our datasets span three different kinds of RNGs, namely Quantum Random Number Generators (QRNGs) based on quantum-optics measurements, Pseudo Random Number Generators (PRNGs), which are Linear Congruential RNGs $X_{n+1} = (a \cdot X_n + c) \bmod m$, where X_n denotes the n-th sequence value, a constitutes the multiplication coefficient, c represents the additive increment, and m defines the modular constraint. The choice of parameters (a, c, m) directly determines the generator's period length, statistical properties, and vulnerability to pattern detection. Our evaluation employs two carefully selected parameter sets that represent different computational-complexity levels: Parameter Set 1 utilizes the multiplier $a_1 = 25214903917$ (derived from established LCRNG literature [117]) with minimal increment $c_1 = 1$, while Parameter Set 2 employs the Numerical Recipes constants with $a_2 = 1103515245$ and $c_2 = 12345$ [118]. The systematic variation across modulus magnitudes spanning powers of 2 ($m = 2^{24}$ through 2^{32}) facilitates comprehensive period-dependent neural-network detection analysis, given theoretical period-duration scaling directly with modulus magnitude via period $\leq m$. We also have CSPRNGs: ChaCha20 stream-cipher implementations designed for cryptographic-security applications. The details of the datasets can be found in Table 3.1.

We also have two categories: Unprocessed (UP), which is data that comes out of the physical or algorithmic process, and Post-Processed (PP), where Toeplitz hashing [119–121] is used to perform randomness extraction by mapping raw, biased, or correlated data into an almost uniform random sequence.

3.1.1 Neural Network Architectures

To avoid baking in an architectural "inductive bias" for a task that should be close to unpredictable, we evaluated a deliberately broad set of fifteen PyTorch models spanning

Table 3.1: Complete Dataset Summary: Random Number Generator Types and Processing States

Dataset Name	**RNG Type**	**Processing State**	**Description**
Quantum Random Number Generators (QRNG)			
output-0989_1.2.txt	QRNG	PP	Quantum optical source, post-processed
output-0996_1.2.txt	QRNG	PP	Quantum optical source, post-processed
fm_89_5.txt	QRNG	UP	Quantum optical source, unprocessed
fm_96_5.txt	QRNG	UP	Quantum optical source, unprocessed
Cryptographically Secure Pseudo-Random Generators (CS-PRNG)			
output-1.0-ChaCha20_V1_5M.txt	CS-PRNG	PP	ChaCha20 cipher, post-processed
output-1.2-ChaCha20_V1_5M.txt	CS-PRNG	PP	ChaCha20 cipher, post-processed
output-1.5-ChaCha20_V1_5M.txt	CS-PRNG	PP	ChaCha20 cipher, post-processed
output-2.0-ChaCha20_V1_5M.txt	CS-PRNG	PP	ChaCha20 cipher, post-processed
ChaCha20_V1_5M_one.txt	CS-PRNG	UP	ChaCha20 cipher, unprocessed
ChaCha20_V1_5M_two.txt	CS-PRNG	UP	ChaCha20 cipher, unprocessed
Pseudo-Random Number Generators (PRNG) - Post-Processed			
LCRNG_24_1_2M_PP.txt	PRNG	PP	LCRNG: a_1=25214903917, c_1=1, m=2^{24}
LCRNG_26_1_2M_PP.txt	PRNG	PP	LCRNG: a_1=25214903917, c_1=1, m=2^{26}
LCRNG_28_1_2M_PP.txt	PRNG	PP	LCRNG: a_1=25214903917, c_1=1, m=2^{28}
LCRNG_30_1_2M_PP.txt	PRNG	PP	LCRNG: a_1=25214903917, c_1=1, m=2^{30}
LCRNG_32_1_2M_PP.txt	PRNG	PP	LCRNG: a_1=25214903917, c_1=1, m=2^{32}
output-1.2-m_24_5M_A_C.txt	PRNG	PP	LCRNG: a_2=1103515245, c_2=12345, m=2^{24}
output-1.2-m_26_5M_A_C.txt	PRNG	PP	LCRNG: a_2=1103515245, c_2=12345, m=2^{26}
output-1.2-m_28_5M_A_C.txt	PRNG	PP	LCRNG: a_2=1103515245, c_2=12345, m=2^{28}
output-1.2-m_30_5M_A_C.txt	PRNG	PP	LCRNG: a_2=1103515245, c_2=12345, m=2^{30}
output-1.2-m_32_5M_A_C.txt	PRNG	PP	LCRNG: a_2=1103515245, c_2=12345, m=2^{32}
Pseudo-Random Number Generators (PRNG) - Unprocessed			
fm_24_5.txt	PRNG	UP	LCRNG: a_1=25214903917, c_1=1, m=2^{24}
fm_26_5.txt	PRNG	UP	LCRNG: a_1=25214903917, c_1=1, m=2^{26}
fm_28_5.txt	PRNG	UP	LCRNG: a_1=25214903917, c_1=1, m=2^{28}
fm_30_5.txt	PRNG	UP	LCRNG: a_1=25214903917, c_1=1, m=2^{30}
fm_32_5.txt	PRNG	UP	LCRNG: a_1=25214903917, c_1=1, m=2^{32}
m_24_5M_A_C.txt	PRNG	UP	LCRNG: a_2=1103515245, c_2=12345, m=2^{24}
m_26_5M_A_C.txt	PRNG	UP	LCRNG: a_2=1103515245, c_2=12345, m=2^{26}
m_28_5M_A_C.txt	PRNG	UP	LCRNG: a_2=1103515245, c_2=12345, m=2^{28}
m_30_5M_A_C.txt	PRNG	UP	LCRNG: a_2=1103515245, c_2=12345, m=2^{30}
m_32_5M_A_C.txt	PRNG	UP	LCRNG: a_2=1103515245, c_2=12345, m=2^{32}

Total: 30 datasets (4 QRNG, 6 CS-PRNG, 20 LCRNG)

Processing states: 19 post-processed (PP), 11 unprocessed (UP)

recurrent, convolutional, attention-based and hybrid families. On the recurrent side we used LSTM/biLSTM and GRU baselines for long-range temporal structure [122, 123], and an attention-augmented RNN that forms a weighted summary over timesteps when periodic or sparse cues exist [124, 125]. For convolutional processing we used multi-kernel 1D CNNs (local motifs), dilated convolutions (large receptive fields at fixed parameter cost), and residual 1D blocks for stable optimization [104, 126]. We also included a temporal convolutional network with causal dilations [127]. For global-context modeling we used encoder-only Transformer variants, including a standard sinusoidal-positional version and an enhanced pre-norm/GELU variant [128–130]. Hybrid models combine these ideas (e.g., CNN-LSTM and CNN-Transformer) to capture both local and long-range dependencies. Across architectures we keep the interface consistent: a lightweight input embedding, regularization via dropout (typically 0.1–0.2), and standard initialization choices (Xavier/orthogonal for recurrent/linear layers and Kaiming for convolutions) [126, 131], followed by a 256-way classifier for next-byte prediction. Full implementation details and exact hyperparameters are provided in the accompanying code release at https://github.com/rithvik1122/NNRNG/blob/main/code/alternative_nn_architectures.py.

The data is partitioned into 60/20/20 for training, validation, and testing, and overfitting is prevented by employing 10-epoch patience. The Adam optimizer (learning rate 0.001) [132] is fixed to maintain uniform training dynamics across all architectural variants. The core evaluation metrics are *Improvement Factor (IF)*, defined as the ratio of the model's predicting ability to random guess ($1/256 \approx 0.39\%$); *Training Duration*, which is the total time required for complete model-training cycles (measured in seconds); *Memory Consumption*, which is the maximum memory utilization during the training processes (reported in MB); and *Computational Efficiency*, which is the performance-to-time ratio computed as improvement factor divided by training duration (factor/second). Execution occurs on uniform hardware configurations utilizing CUDA acceleration where supported. Resources are monitored through the psutil library, which maintains consistent tracking of computational demands across all experimental setups. Statistical validation using Analysis of Variance (ANOVA) F-statistics and effect-size analysis (Cohen's d) is performed on the 1,269 configurations encompassing 30 datasets, 15 neural architectures, and four sequence magnitudes (1K-1M samples) (some datasets do not have the full 1M sequences, giving us the total number of 1,269).

3.2 Results

We begin by providing a view of neural predictability aggregated over all RNG types, processing states, and architectures (see Figure 3.1). As can be seen, the improvement factors monotonically increase as the sequence length goes from 1K to 1M. This shows that as the training corpus increases, models are able to capitalize more reliably on hidden or weak dependencies and are therefore able to predict better.

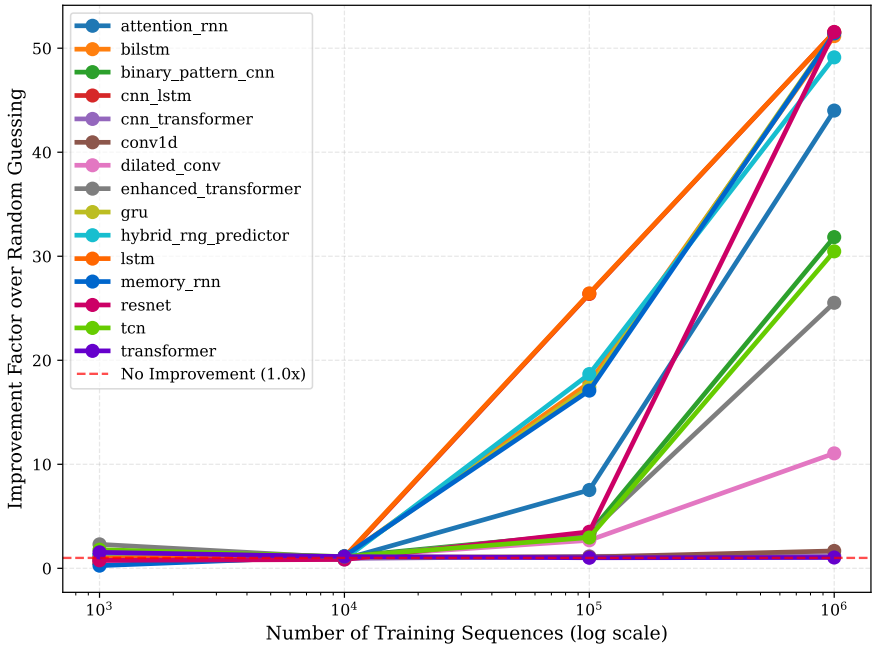
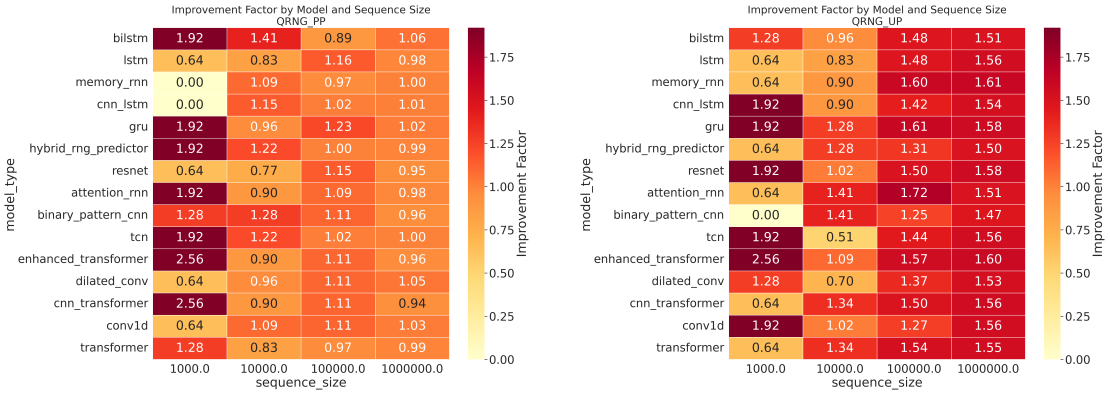


Figure 3.1: Predictability (Improvement Factor) increases monotonically with sequence lengths (1K to 1M) when aggregated over all RNG types, processing states, and architectures

3.2.1 Processing State Analysis: PP vs UP Performance Comparison

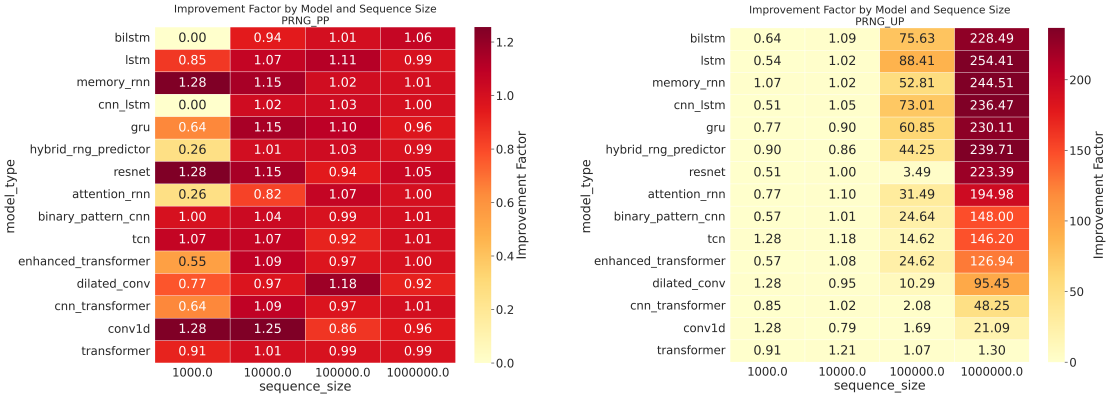
This segment contains the predictability analysis of Unprocessed (UP) versus Post-Processed (PP) data, where the latter is obtained after Toeplitz hashing of the unprocessed data.

For QRNGs, both PP and UP remain close to the random-guess baseline, with only modest and bounded predictability across architectures (Figure 3.2). Post-processing slightly tightens the distribution (PP mean $1.11 \pm 0.79\times$ vs UP mean $1.33 \pm 0.63\times$; UP-/PP $\approx 1.20\times$), consistent with the intuition that quantum sources are already difficult to learn and conditioning mainly removes small residual biases.



(a) (b)

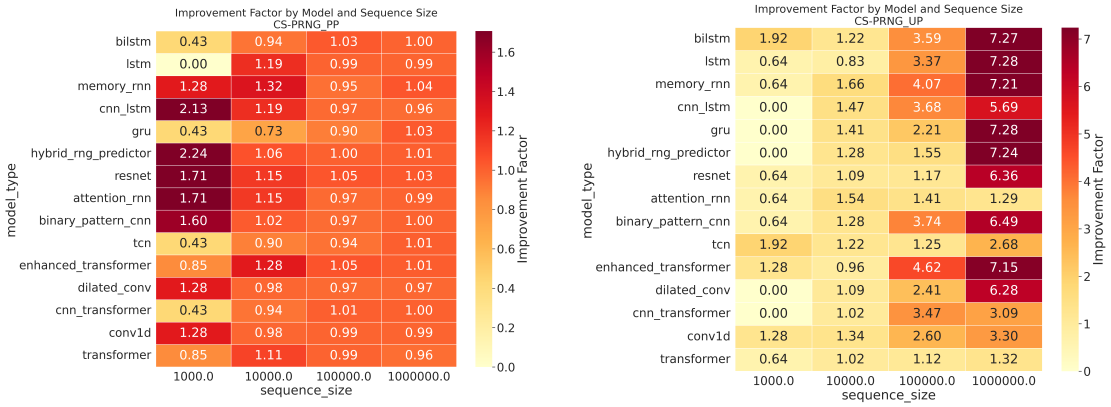
Figure 3.2: QRNG performance heatmaps comparing post-processed (PP) vs unprocessed (UP) data across architectures and sequence sizes. Both processing states show excellent randomness quality with modest improvement factors, demonstrating the inherent strength of quantum sources. Post-processed data shows slightly better consistency (mean=1.11×) while unprocessed data maintains good performance (mean=1.33×), with both states showing low predictability.



(a) (b)

Figure 3.3: PRNG performance comparison showing significant processing state sensitivity. Unprocessed data enables dramatically higher improvement factors (up to 255.1×) compared to post-processed, indicating that processing removes exploitable algorithmic patterns that neural networks can detect.

PRNGs show the strongest processing dependence (Figure 3.3). In the UP condition, improvement factors can become extreme (mean 49.3×, maxima ≈ 255×), indicating that the neural networks can learn the deterministic algorithm that generates the sequences. After post-processing, performance collapses back near baseline (mean ≈ 0.95×), suggesting that Toeplitz hashing randomizes the PRNG sequences enough to make them unpredictable with respect to neural networks.



(a) (b)

Figure 3.4: CS-PRNG analysis revealing minimal predictability across both processing states. (a) Post-processed and (b) unprocessed data both show improvement factors near random baseline (0.0×-13.6×), demonstrating cryptographic strength. Enhanced Transformer slightly outperforms others but remains with low predictability.

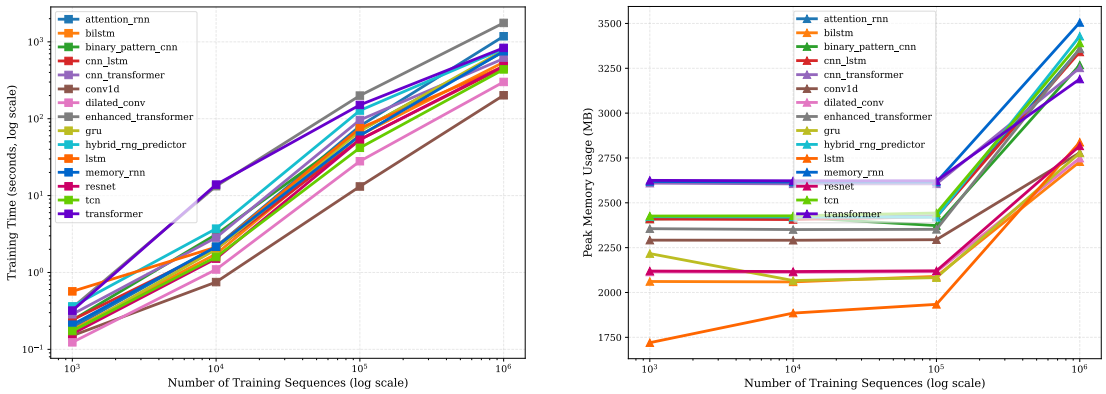
CS-PRNGs, while more predictable than QRNGs in the UP state, offer significantly better quality than UP PRNGs. However, post-processing removes nearly any chance of prediction and brings them back to the random-chance baseline.

3.2.2 Computational Scaling Analysis

Training cost grows predictably with sequence length (Figure 3.5). Across families, training time follows an approximate power law with similar exponents: recurrent models $O(N^{1.16})$ ($R^2 = 0.991$), convolutional models $O(N^{1.14})$ ($R^2 = 0.996$), and transformers $O(N^{1.16})$ ($R^2 = 0.985$). The practical difference is in constant factors: at matched N, transformer variants typically require roughly 1.5–2× longer wall-clock time than recurrent alternatives.

Within each family, scaling is stable but not identical. Recurrent exponents range from 1.01 (LSTM) to 1.27 (Attention-RNN), with GRU exhibiting an excellent fit ($R^2 =$

0.995). CNN variants cluster tightly (1.03 for Conv1D to 1.18 for ResNet). Among transformers, the Enhanced Transformer shows the steepest scaling ($O(N^{1.21})$) while



(a) (b)

Figure 3.5: Computational resource scaling across sequence lengths. (a) Training time follows power-law relationships with different exponents for each architecture class. (b) Memory consumption scales similarly, with transformer models requiring 2-3× more resources than recurrent alternatives.

the standard Transformer scales more favorably ($O(N^{1.13})$) [128].

Across processing conditions, efficiency trade-offs are well-captured by the Pareto frontiers (Figure 3.6). For PP data, Conv1D is consistently the most efficient low-budget option (2.79 imp/s for PRNG, 2.22 imp/s for QRNG, 3.25 imp/s for CS-PRNG), while mid-budget choices vary by generator (TCN for QRNG at 3.05 imp/s). High-budget models can maximize raw improvement factors (Enhanced Transformer reaching 5.12× for QRNG) but pay a clear efficiency penalty. For UP data, Conv1D remains an efficiency leader for QRNG and CS-PRNG (4.36 and 4.03 imp/s respectively), whereas PRNG discrimination benefits more from specialized convolutions and recurrent memory (Dilated Conv at 3.14 imp/s; LSTM reaching the maximum UP PRNG improvement).

Improvement factors increase sharply when moving from 100K to 1M sequences across nearly all architectures, reinforcing that predictability assessment is data-hungry. In practice, training efficiency can plateau, but discrimination capability continues to rise with additional data—most noticeably for higher-capacity models that can exploit subtle long-range structure.

The same pattern holds when summarized at the architecture level: simple convolutional models often provide the best improvement-per-second, and they do so reliably across RNG types. In PP conditions, Conv1D dominates efficiency for PRNG and CS-

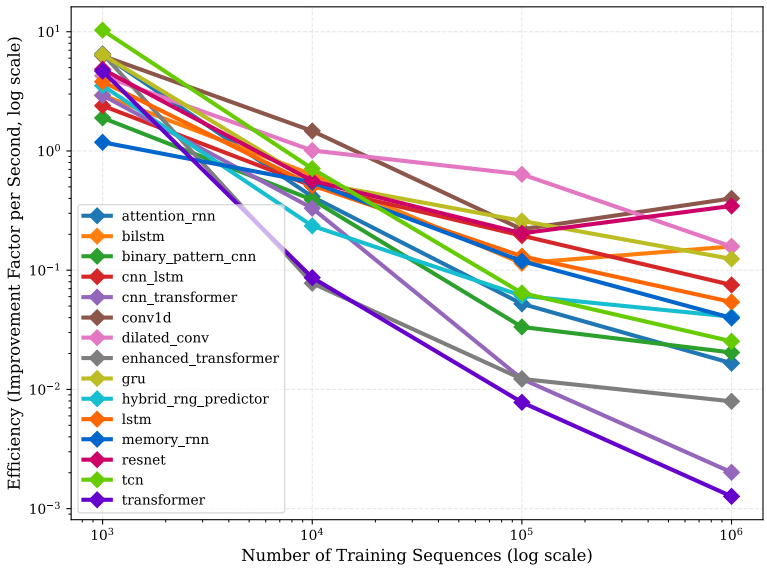
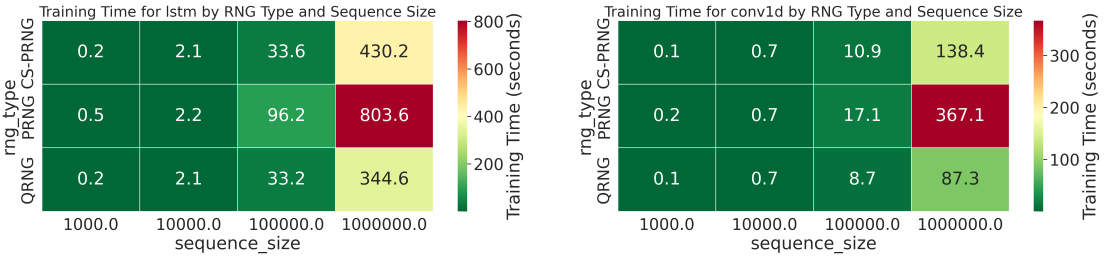


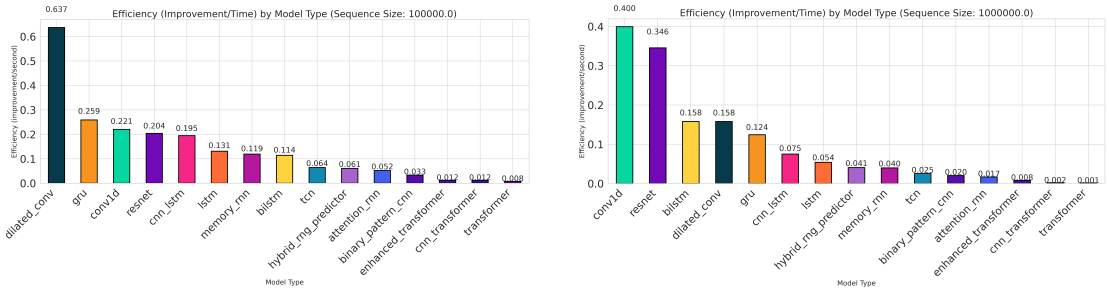
Figure 3.6: Computational efficiency analysis showing improvement factor per unit training time across different architectures and sequence scales. GRU and simple CNN models provide optimal efficiency for resource-constrained applications, while Enhanced Transformer maximizes discrimination capability at higher computational cost.



(a) (b)

Figure 3.7: Training time analysis for top-performing architectures by category. (a) LSTM demonstrates exceptional discrimination capability, achieving 255.052× improvement on UP PRNG data while maintaining efficiency. (b) Conv1D shows optimal computational efficiency across multiple RNG types and processing states, representing the best efficiency-performance balance.

PRNG while TCN is strongest for QRNG. In UP conditions, Conv1D remains best for QRNG and CS-PRNG and Dilated Conv is most efficient for PRNG. When stability is prioritized, the standard Transformer is the most consistent across processing states (low CV in both PP and UP), whereas several strong PP performers (Conv1D, TCN, LSTM) become substantially more variable under UP data.



(a) (b)

Figure 3.8: Computational efficiency analysis across sequence scales demonstrating architecture-dependent performance patterns. (a) 100K sequences and (b) 1M sequences show that simple architectures (Conv1D, Dilated Conv, TCN) consistently achieve superior efficiency ratios (improvement factor per training time) compared to complex models, validating the efficiency leadership identified in our analysis, while complex architectures may achieve higher raw discrimination but at significantly increased computational cost.

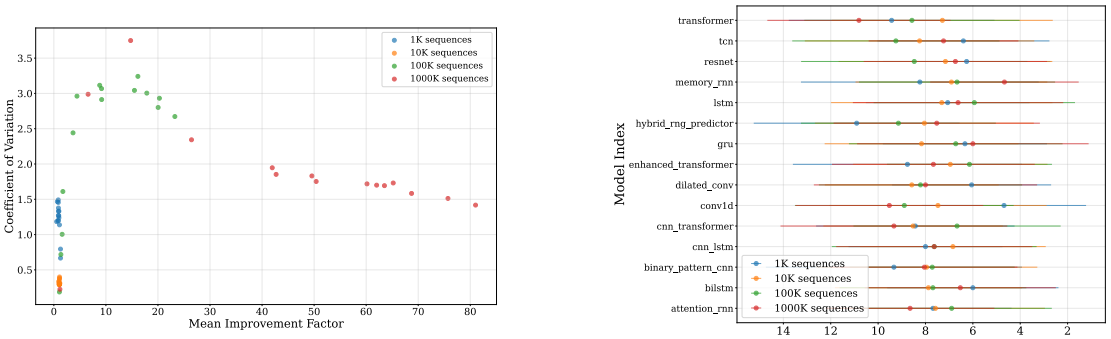
3.2.3 Model Consistency and Reliability Analysis

Because true randomness is essentially the "kryptonite" of neural pattern recognizers, we treat next-symbol prediction as a benchmark-style stress test and report multiple, complementary views (performance, efficiency, and reliability) so the conclusions do not hinge on any single metric.

Figure 3.9 summarizes two practical questions: which models perform well on average, and which do so reliably. Mean performance and consistency are not tightly coupled: some high-performing models are stable, but others achieve their peaks only on particular datasets. Sequential architectures such as Conv1D, TCN, and LSTM tend to be among the most reliable, while more complex variants (like the Enhanced Transformer and Memory-RNN) show higher variability across conditions. As expected from the scaling analysis, the average improvement factor increases with sequence length, and this rise is accompanied by increased dispersion across experimental settings.

Ranking stability across datasets is moderate rather than absolute: architectures can

swap positions depending on RNG type, processing, and sequence length. The ranking-consistency view in Figure 3.9 (panel b) shows that some models maintain relatively stable ranks (e.g., CNN-LSTM with ranking std 0.509), while others fluctuate more (e.g., Conv1D with ranking std 1.882). Overall, most models fall in the 0.5–1.5 range, which is sufficient to support evidence-based architecture selection but also motivates reporting variability alongside mean performance.



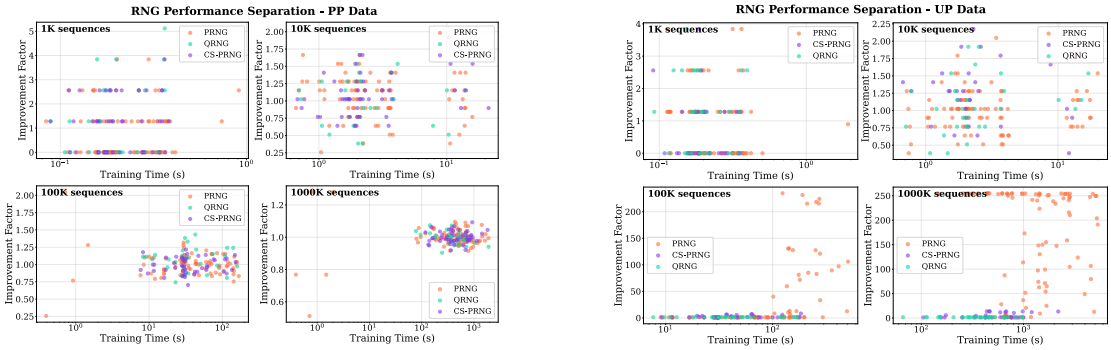
(a) (b)

Figure 3.9: Model consistency analysis revealing stability of architectural rankings across experimental conditions. (a) The consistency scatter plot visualizes the relationship between mean performance and coefficient of variation, while (b) ranking consistency shows how architectural performance varies across different experimental configurations.

3.2.4 RNG Type Discrimination

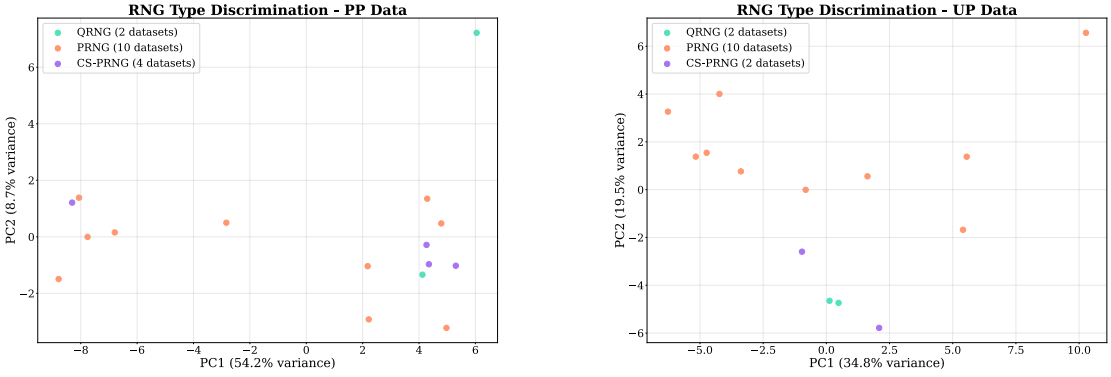
Neural predictability separates generator types most clearly in the UP condition (Figure 3.10). With post-processing, the distributions collapse toward the baseline and medians differ only slightly (QRNG 1.105×, PRNG 0.924×, CS-PRNG 1.047×), consistent with effective conditioning. Without post-processing, the picture changes: PRNG medians rise sharply (45.087× with maxima up to 255.05×), QRNG remains near 1.331×, and CS-PRNG stays relatively low (median 2.482×). Even then, overlap persists, so discrimination is informative at the population level but not a perfect per-dataset classifier.

The PCA view (Figure 3.11) reinforces this limitation. Although the first two PCs explain substantial variance (PP: 62.9%; UP: 54.3%), silhouette scores remain slightly negative (PP: -0.059; UP: -0.020), indicating that clusters by RNG type are not cleanly separable in this reduced feature space.



(a) (b)

Figure 3.10: RNG type performance separation analysis across processing states. (a) Post-processed (PP) data shows convergence of all RNG types toward random baseline performance, while (b) unprocessed (UP) data demonstrates clear discrimination between generator types based on neural network improvement factors.



(a) (b)

Figure 3.11: Principal Component Analysis of RNG type discrimination capability. (a) PCA analysis of post-processed (PP) data shows limited clustering with overlapping distributions, while (b) unprocessed (UP) data reveals moderate separation with lower explained variance and poor silhouette scores indicating limited discrimination capability.

Taken together, discrimination is strongest when processing is removed, but that strength largely reflects learnable structure that conditioning is designed to eliminate. In deployed settings where PP data is the relevant object, neural predictability differences between RNG types are narrow and the main signal becomes whether conditioning has successfully suppressed algorithmic artifacts.

This is consistent with the leftover hash lemma viewpoint: a strong extractor (such as Toeplitz-hashing-based post-processing) is designed to produce outputs that are statistically close to uniform, so any downstream discriminator, including neural predictors, should have little stable signal left to separate generator classes once conditioning is effective [119–121].

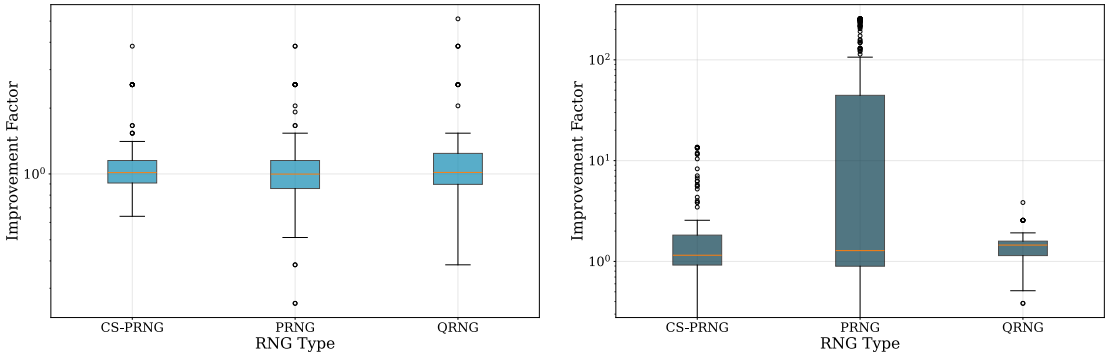


Figure 3.12: Dataset variability analysis by RNG type and processing state. (a) Post-processed (PP) data shows consistent low variability across all RNG types, clustering near baseline performance. (b) Unprocessed (UP) data reveals dramatic differences: PRNG exhibits highest predictability, QRNG shows moderate vulnerability, and CS-PRNG maintains cryptographic resistance.

3.2.5 Statistical Significance and Effect Size Analysis

We perform significance testing to confirm that the differences observed between RNG types are not due to sampling noise alone. We present results stratified by processing (PP/UP) as well as in the aggregate. Each configuration contributes an improvement factor, which is the metric used in testing.

For significance, we apply a one-way ANOVA to test the null hypothesis H_0 : $\mu_{QRNG} = \mu_{PRNG} = \mu_{CS-PRNG}$:

$$F = \frac{MS_{between}}{MS_{within}} = \frac{SSB/(k-1)}{SSW/(N-k)} \tag{3.1}$$

In this context, $MS_{between}$ captures systematic differences in predictability (Improvement Factor) between generator classes, while MS_{within} captures variability within a class arising from dataset idiosyncrasies, processing (PP/UP), architecture choice, and experimental noise.

$$MS_{between} = \frac{SSB}{k-1} = \frac{\sum_{i=1}^{k} n_i(\bar{X}_i - \bar{X})^2}{k-1} \tag{3.2}$$

$$MS_{within} = \frac{SSW}{N-k} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i}(X_{ij} - \bar{X}_i)^2}{N-k} \tag{3.3}$$

Statistical parameters are $k = 3$ RNG categories, n_i the sample size for group i, $\bar{X}_i$ the mean improvement factor for group i, $\bar{X}$ the grand mean, N the total number of observations, and X_{ij} an individual observation.

The p-values for the observed F values are:

$$p = P(F \geq F_{observed} | H_0 \text{ is true}) \tag{3.4}$$

with F distributed with $(k-1, N-k)$ degrees of freedom under H_0.

$$p = 1 - CDF_F(F_{observed}; k-1, N-k) \tag{3.5}$$

Processing has a strong effect on predictability (Table 3.2). PRNGs exhibit extreme dependence on conditioning (UP/PP improvement ratio $\approx 52\times$), while QRNGs are comparatively stable (ratio $\approx 1.2\times$) and CS-PRNGs show a moderate reduction after processing (ratio $\approx 2.4\times$). Consistent with this, Table 3.3 shows that post-processing largely suppresses generator-type discrimination at larger sequence lengths, whereas UP data retains strong and increasing type differences (1M UP: $F = 64.38$, $p = 5.2 \times 10^{-21}$).

Table 3.2: Processing State Impact on Neural Network Predictability

RNG Type	**PP Mean±Std**	**PP Max**	**UP Mean±Std**	**UP Max**	**Improvement Ratio**
QRNG	$1.10 \pm 0.79\times$	$5.12\times$	$1.33 \pm 0.63\times$	$3.84\times$	$1.20\times$
PRNG	$0.95 \pm 0.55\times$	$3.84\times$	$49.32 \pm 89.82\times$	$255.05\times$	$52.03\times$
CS-PRNG	$1.05 \pm 0.57\times$	$3.84\times$	$2.48 \pm 3.48\times$	$13.64\times$	$2.37\times$

To quantify practical importance beyond p-values, we also compute Cohen's d. Cohen's d measures standardized mean differences between groups using pooled standard

Table 3.3: Statistical Significance of RNG Type Differences by Processing State

Sequence Size	PP F-statistic	PP p-value	UP F-statistic	UP p-value
1K	3.84	0.023*	2.46	0.089
10K	0.084	0.920	4.07	0.019*
100K	2.54	0.082	6.52	0.002**
1M	0.078	0.925	64.38	5.2×10^{-21} ***

*$p < 0.05$, **$p < 0.01$, ***$p < 0.001$

deviation. The effect size is calculated as:

$$d = \frac{\bar{X}_1 - \bar{X}_2}{s_{pooled}} \tag{3.6}$$

where $\bar{X}_1$ and $\bar{X}_2$ are the sample means for groups 1 and 2, respectively, and s_{pooled} is the pooled standard deviation computed as:

$$s_{pooled} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \tag{3.7}$$

Cohen's interpretation guidelines [133] classify effect sizes as small (d = 0.2), medium (d = 0.5), or large (d = 0.8), indicating the magnitude of practical significance.

Table 3.4: Cohen's d Effect Sizes for RNG Type Comparisons by Processing State

Comparison	1K	10K	100K	1M
Post-Processed (PP) Data				
QRNG vs PRNG	0.51^{M}	-0.07^{S}	0.36^{S}	-0.01^{S}
QRNG vs CS-PRNG	0.14^{S}	-0.09^{S}	0.61^{M}	-0.14^{S}
PRNG vs CS-PRNG	-0.40^{S}	-0.02^{S}	0.12^{S}	-0.06^{S}
Unprocessed (UP) Data				
QRNG vs PRNG	0.40^{S}	0.15^{S}	-0.55^{M}	-1.70^{L}
QRNG vs CS-PRNG	0.52^{M}	-0.40^{S}	-0.72^{M}	-0.99^{L}
PRNG vs CS-PRNG	0.14^{S}	-0.62^{M}	0.53^{M}	1.66^{L}
Processing State Comparisons (PP vs UP)				
QRNG: PP vs UP	0.06^{S}	-0.09^{S}	-2.29^{L}	-9.65^{L}
PRNG: PP vs UP	-0.09^{S}	0.13^{S}	-0.68^{M}	-2.09^{L}
CS-PRNG: PP vs UP	0.43^{S}	-0.50^{M}	-1.15^{L}	-1.29^{L}

^{S}Small ($|d| < 0.5$), ^{M}Medium ($0.5 \leq |d| < 0.8$), ^{L}Large ($|d| \geq 0.8$)

Table 3.4 shows that, after post-processing, most between-type differences are small in magnitude (typically $|d| < 0.5$), with a modest peak at 100K for QRNG vs CS-PRNG ($d = 0.61$, medium). In contrast, UP data produces large effects at longer sequences

(e.g., at 1M: QRNG vs PRNG $d = -1.70$ and PRNG vs CS-PRNG $d = 1.66$), indicating substantial practical separation when conditioning is absent. The PP vs UP contrasts are also large at high N, confirming that conditioning fundamentally changes what neural models can exploit. Given the breadth of configurations (1,269 total), these conclusions are statistically well-powered and robust across architectures and datasets.

3.2.6 Architecture Performance and Consistency

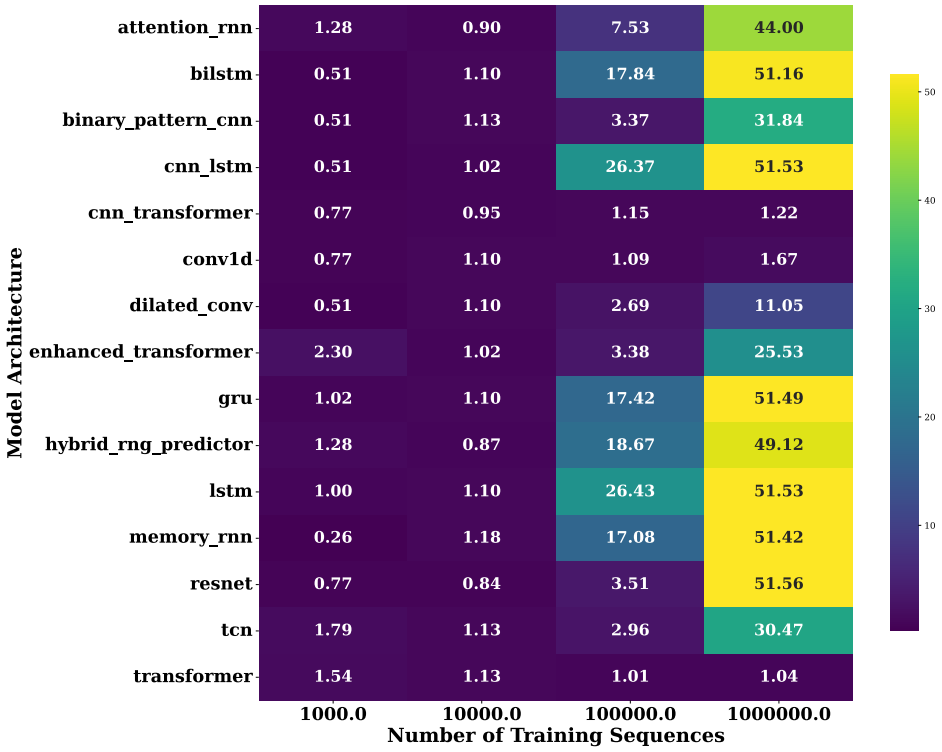


Figure 3.13: Performance heatmap showing improvement factors across neural network architectures and sequence sizes. Enhanced Transformer and Hybrid RNG Predictor models demonstrate superior consistency, while recurrent networks show variable performance dependent on generator type.

Table 3.5 and Figure 3.13 summarize how architecture choice affects both average predictability and run-to-run stability across datasets. We quantify stability via the coefficient of variation (CV), computed separately for PP and UP data to avoid conflating conditioning effects with model variability:

$$CV = \frac{\sigma_{\text{improvement}}}{\mu_{\text{improvement}}} \tag{3.8}$$

where $\sigma_{\text{improvement}}$ and $\mu_{\text{improvement}}$ denote the standard deviation and mean of improve-

Table 3.5: Neural Network Architecture Performance Consistency by Processing State

Architecture	**PP CV**	**UP CV**	**Consistency Pattern**
Highest Consistency (Low CV)			
Transformer	0.634	0.574	Consistently low across states
Conv1D	0.468	3.376	PP-favored, UP variable
LSTM	0.491	2.091	PP-favored, UP variable
ResNet	0.499	2.452	PP-favored, UP variable
TCN	0.478	2.343	PP-favored, UP variable
Dilated Conv	0.581	2.931	PP-favored, UP variable
Moderate Consistency			
BiLSTM	0.580	1.981	PP-favored, UP moderate
CNN-LSTM	0.619	2.049	PP-favored, UP moderate
Binary Pattern CNN	0.580	2.173	PP-favored, UP moderate
GRU	0.719	2.094	PP-favored, UP moderate
Attention-RNN	0.660	2.246	PP-favored, UP moderate
Enhanced Transformer	0.726	2.358	PP-favored, UP moderate
Hybrid RNG Predictor	0.681	1.854	PP-favored, UP moderate
Memory-RNN	0.647	1.911	PP-favored, UP moderate
Variable Consistency (High CV)			
CNN-Transformer	0.575	4.749	PP consistent, UP highly variable

CV = $\sigma_{\text{improvement}}/\mu_{\text{improvement}}$, separated by processing state

ment factors computed separately for post-processed (PP) and unprocessed (UP) data across all 30 datasets, 4 sequence lengths, and 3 RNG types. Overall, most architectures are more consistent on PP data (lower CV) than on UP data, with the standard Transformer being a notable exception that remains consistently low-CV across both states.

The strongest stability is achieved by the standard Transformer (PP CV 0.634, UP CV 0.574). Several simple architectures (e.g., Conv1D, TCN, ResNet) are also stable on PP data (CV < 0.5), but their UP variability increases substantially; this highlights that processing state can dominate perceived architecture robustness.

3.2.7 NIST SP 800-22 Comparison

Finally, we compare the neural predictability results with those of the standard NIST SP 800-22 outcomes. We observe that PRNG-UP can achieve a moderate NIST pass rate (41.2%) while remaining highly predictable by neural networks (116.8×), whereas QRNG-UP can fail many NIST tests (23.5% pass) yet still remain unpredictable by neu-

ral networks, which would conclude they are random-like (1.53×). In well-conditioned PP outputs, the two assessments generally converge (pass rates around 58.8–64.7% with low improvement factors near 1×).

Table 3.6: Complete NIST SP 800-22 vs Neural Network Assessment Comparison

RNG Type	NIST Pass Rate	NN Improv. Factor	NIST Assess.	NN Assess.
QRNG-PP	58.8%	1.18×	Moderate	Random-like
QRNG-UP	23.5%	1.53×	Poor	Random-like
PRNG-PP	64.7%	0.99×	Good	Random-like
PRNG-UP	41.2%	116.8×	Mixed	Highly Predictable
CS-PRNG-PP	64.7%	0.95×	Good	Random-like
CS-PRNG-UP	35.3%	4.38×	Poor	Moderately Predict.

Table 3.7: Detailed NIST SP 800-22 Test Results by Category (17 Individual Test Results from 15 Core Tests)

Test Category	Q-PP	Q-UP	P-PP	P-UP	CS-PP	CS-UP
Basic Frequency Tests	4/4	0/4	4/4	3/4	4/4	2/4
Spectral & Matrix Tests	2/2	2/2	2/2	1/2	2/2	0/2
Template Tests	0/2	0/2	1/2	1/2	1/2	1/2
Complexity Tests	2/3	2/3	2/3	2/3	2/3	2/3
Serial Tests (2 variants)	2/2	0/2	2/2	0/2	2/2	1/2
Cumulative Sum Tests (2 variants)	0/2	0/2	0/2	0/2	0/2	0/2
Random Walk Tests (2 tests)	0/2	0/2	0/2	0/2	0/2	0/2
Total (17 tests)	**10/17**	**4/17**	**11/17**	**7/17**	**11/17**	**6/17**
Pass Rate	**58.8%**	**23.5%**	**64.7%**	**41.2%**	**64.7%**	**35.3%**

Column abbreviations: Q-PP/UP = QRNG Post-processed/Unprocessed, P-PP/UP = PRNG Post-processed/Unprocessed, CS-PP/UP = CS-PRNG Post-processed/Unprocessed

These results suggest that our Multi-Architecture Neural Network Predictors act as a complementary test for randomness alongside the standard NIST SP 800-22 test.

3.3 Discussion and Conclusion

This extensive investigation, spanning 1,269 experimental configurations across 30 datasets, several RNG types, conditioning types, and multiple statistical analyses, offers strong evidence that the Multi-Architecture Neural Network Predictors are a novel and strong candidate for randomness-predictability assessment, showing instances where machine-learning techniques can detect subtle and exploitable patterns and provide predictions that traditional statistical tests might not be able to.

3.3.1 Neural Network Architecture Recommendations

Based on our systematic multi-objective optimization analysis of 15 architectures across 1,269 experimental configurations, we establish evidence-based recommendations using weighted performance criteria (Improvement Factor: 40%, Efficiency: 30%, Training Speed: 20%, Memory Usage: 10%). While pure performance ranking favors CNN-LSTM (19.86× mean improvement), LSTM (19.11× mean improvement), and GRU (17.76× mean improvement), practical deployment requires balancing accuracy with computational efficiency and training speed. Table 3.8 presents the comprehensive ranking and use-case specific guidance.

Table 3.8: Neural Network Architecture Recommendations Based on Multi-Objective Analysis

Rank	Architecture	Overall Score	Improvement Factor	Efficiency	Training Time (s)
Tier 1 - Optimal Balance (Recommended for Most Applications)					
1	Conv1D	0.583	1.16×	2.093	54.0
2	Dilated Conv	0.522	3.84×	1.519	82.5
3	TCN	0.506	9.09×	2.777	120.4
Tier 2 - High Performance (For Maximum Detection)					
4	LSTM	0.482	19.11×	1.256	132.5
5	ResNet	0.470	14.17×	1.489	134.4
6	BiLSTM	0.455	17.65×	0.960	151.5
7	GRU	0.431	17.76×	1.835	211.7
Tier 3 - Specialized Applications					
8	CNN-LSTM	0.416	19.86×	0.794	130.3
9	Attention RNN	0.377	13.43×	1.738	316.0
10	Memory RNN	0.377	17.48×	0.472	204.2

These results in Table 3.8 suggest that Conv1D is a strong default candidate, owing to its predictability and efficiency. If one desires maximum predictability at higher training-time and memory costs, models like CNN-LSTM or LSTM are better suited. When it comes to different sequence-length regimes, TCN performs well on small sequence sizes, Conv1D is good when medium-sized sequences are available, and Dilated Convolutions become attractive at the largest scale, where longer receptive fields can be effective without incurring prohibitive training-time costs.

This work contributes a novel method to assess the randomness quality of an RNG using a Multi-Architecture Neural Network Framework, which provides complementary insights to traditional NIST SP 800-22 tests. Comprehensive analysis using 1,269 configurations spanning multiple RNG types, conditioning states (PP, UP), varying sequence

lengths, statistical analyses (ANOVA, Cohen's d), and multiple neural network architectures establishes that the conclusions are not an artifact of a single neural-network family.

Part II

AI Systems for Quantum Experiment Design and Analysis

"In God we trust. All others must bring data."

W. Edwards Deming

4

Evaluating Large Language Models for Quantum Mechanics Problem Solving[1]

Having seen the applications of Artificial Neural Networks (ANNs) for quantum systems, we now turn our attention to larger versions of these models, trained on large quantities of text data drawn from web crawls (Common Crawl and filtered subsets), books, Wikipedia, and curated datasets such as The Pile ($\sim$ 886 GB of text), experimental scaling regimes like Chinchilla ($\sim$ 1.4 trillion training tokens), while surveys report total pre-training corpora exceeding hundreds of terabytes across many component datasets [134–137]. These large neural networks, typically based on the Transformer architecture[49], are referred to as *Large Language Models (LLMs)*, and it would be an understatement to say that they have taken the world by storm. Large language models (LLMs) have demonstrated remarkable capabilities across multiple domains, ranging from natural language understanding[138] to mathematical reasoning[139] and code generation[140]. As LLMs increasingly serve as research assistants and educational tools in scientific contexts, systematic evaluation of their capabilities and limitations in specialized domains becomes essential.

While prior work has assessed LLMs on mathematical reasoning (GSM8K[139],

[1]The contents of this chapter have been presented in: S. K. Rithvik, "Evaluating Large Language Models on Quantum Mechanics: A Comparative Study Across Diverse Models and Tasks," arXiv preprint, https://arxiv.org/abs/2602.19006 (2025).

MATH[141]), coding (HumanEval[140], MBPP[142]), and broad scientific knowledge (MMLU[143]), quantum mechanics poses a unique challenge as it demands integration of multiple cognitive modes: understanding concepts that defy classical intuition, executing multi-step symbolic derivations with operator algebra, applying design principles under physical constraints, and implementing numerical algorithms using the right formalisms for computational predictions. Recent work has introduced specialized benchmarks for quantum computing (QCircuitBench[144] for circuit design), quantum science broadly (QuantumBench[145]), and condensed matter physics (CMPhysBench[146] for graduate-level calculations). However, assessing the capabilities of LLMs on diverse task categories such as Symbolic, Creative, Numerical, and Non-standard quantum mechanics problems has not yet been explored explicitly, and this is what we present in this chapter.

We answer the questions: **RQ1:** How do state-of-the-art LLMs perform across diverse quantum mechanics task categories? **RQ2:** Does tool augmentation with code execution improve performance on numerical quantum problems, and at what cost? **RQ3:** How reproducible are LLM responses on quantum mechanics tasks across multiple runs?

4.1 Methods

While the technical foundations of large language models were established earlier, the field gained widespread public and scientific attention after GPT-3 (2020), and especially following the release of ChatGPT and GPT-4 around 2023. Since then, owing to the discovery of scaling laws[67–69], bigger models (with higher numbers of parameters), higher compute (more GPUs for training), and larger training datasets have been released frequently. We categorize these LLMs into three tiers (see Table 4.5), namely fast, mid-tier, and flagship models. While initial research into LLMs provided many details about the training process, data, and model architectures, later research became closed-source, with AI companies not disclosing many details about the model architectures, training data, and process. We therefore chose models from 5 different companies: OpenAI[51–53], Anthropic[54–56], Google[57–59], Alibaba[60–62], and DeepSeek[63, 64].

Our evaluation set consists of 20 quantum mechanics tasks spanning four categories,

each testing distinct cognitive capabilities (a more detailed account can be found at [147]). These tasks include *Derivations (D)* (Table 4.1), which test the models' ability in symbolic reasoning; *Creative (C)* tasks (Table 4.2), which test the models' ability to solve constrained optimization problems; *Non-standard (N)* tasks (Table 4.3), which test models' conceptual understanding of quantum phenomena that are not readily found in the standard textbooks of various graduate and undergraduate curricula; and **Numerical (T)** tasks (Table 4.4), which test models' ability to choose the right formalisms and perform quantitative reasoning. For T tasks, code execution is allowed, and the results are compared for tool-use versus no-tool-use scenarios to assess LLMs' quantitative aptitude. Our evaluation protocol includes 900 baseline assessments (15 models, 20 tasks, 3 full runs; sampling uses `temperature=0` / deterministic decoding[148]), plus 75 tool-augmented evaluations (15 models across the 5 numerical tasks, run at `temperature=0`) enabling Python code execution for numerical tasks. All evaluations were performed using the OpenRouter API for unified access and no token limits. We tracked accuracy, cost, tokens, time, and tool calls for each evaluation.

Note: Tool-augmented evaluation was conducted with an earlier model lineup before the final baseline evaluation. Three models in the tool evaluation were subsequently replaced in the baseline: Qwen 2.5 7B (replaced by Qwen 2.5 Coder 32B), Qwen 2.5 72B (replaced by Qwen3 235B), and DeepSeek R1 Qwen 8B (replaced by DeepSeek R1 Distill 32B). The remaining 12 models were retained as-is across both evaluations. Since the three replacement models do not support tool use (function calling) on OpenRouter, we retained the tool-augmented evaluation data from their predecessors. The tool evaluation includes all 15 models from the earlier lineup (75 evaluations: 15 models $\times$ 5 T tasks).

4.2 Results

The overall performance of the models is presented in Table 4.5: models achieve an average accuracy of 75.1% across 900 evaluations (15 models $\times$ 20 tasks $\times$ 3 runs), with individual model performance ranging from 56.7% to 85.0%. Claude Sonnet 4 and Qwen3-Max tie for best performance at 85.0%, followed by Claude Sonnet 4.5 (83.3%), and DeepSeek V3, DeepSeek R1, and GPT-5 (all three at 80.0%). A clear performance

Table 4.1: Task Catalog (Derivations D1–D5): task statements, options, and ground-truth labels

Task	**Task statement and options**	**Correct**
D1	Compute $[\sigma_z, B(\theta)]$ where $B(\theta) = \cos(\theta)\sigma_x + \sin(\theta)\sigma_y$. **A** $2[\cos(\theta)\sigma_x + \sin(\theta)\sigma_y]$. **B** $2i[\cos(\theta)\sigma_y - \sin(\theta)\sigma_x]$. **C** 0. **D** $2i[\sin(\theta)\sigma_x + \cos(\theta)\sigma_y]$.	**B**
D2	For $\|\psi\rangle = \cos(\alpha/2)\|\uparrow\rangle + \sin(\alpha/2)\|\downarrow\rangle$ (eigenstates of σ_z), find $\Delta\sigma_y$. **A** $\|\sin(\alpha)\|$; **B** $\|\cos(\alpha)\|$; **C** $\sqrt{1-\cos^2(\alpha)}$; **D** 1.	**D**
D3	Find unitary U such that $U^\dagger\sigma_z U = \sigma_x$. **A** $\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$. **B** $\begin{bmatrix}1 & 0\\ 0 & i\end{bmatrix}$. **C** $\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$. **D** $\frac{1}{\sqrt{2}}\begin{bmatrix}1 & -i\\ -i & 1\end{bmatrix}$.	**A**
D4	For $H = \sigma_z + \lambda\sigma_x$ with $\lambda = 0.1$, compare exact $E_\pm = \pm\sqrt{1+\lambda^2}$ to second-order PT $E_\pm \approx \pm(1+\lambda^2/2)$; estimate relative error. **A** $\sim 1\%$; **B** $\sim 5\%$; **C** $< 0.01\%$; **D** $> 10\%$.	**C**
D5	For $\rho = p\|0\rangle\langle 0\| + (1-p)\|1\rangle\langle 1\|$, maximize $S(\rho) = -p\log(p) - (1-p)\log(1-p)$. **A** $p = 0$; **B** $p = 1/2$; **C** $p = 1$; **D** $p = 1/\sqrt{2}$.	**B**

Table 4.2: Task Catalog (Creative C1–C5): task statements, options, and ground-truth labels

Task	**Task statement and options**	**Correct**
C1	3-outcome POVM optimizing discrimination between $\|0\rangle$ and $\|+\rangle = (\|0\rangle + \|1\rangle)/\sqrt{2}$. Maximize $d = p(1\|0) - p(1\|+)$. **A** 0.5; **B** 0.15; **C** 0.4; **D** 0.25.	**A**
C2	Entanglement witness W for $\|\Phi^+\rangle = (\|00\rangle + \|11\rangle)/\sqrt{2}$: what is $\mathrm{Tr}(W\|\Phi^+\rangle\langle\Phi^+\|)$ for an ideal witness? **A** -0.5; **B** 0; **C** -1; **D** -0.25.	**A**
C3	3-qubit bit-flip code: how many distinct syndromes are needed to distinguish $\{\text{no error}, X_1, X_2, X_3\}$? **A** 2; **B** 3; **C** 4; **D** 8.	**C**
C4	Parameterized circuit reaching any single-qubit pure state from $\|0\rangle$; minimum number of parameters? **A** 2; **B** 3; **C** 4; **D** 5.	**A** (minimum), **B** (also accepted in chapter)
C5	CHSH game: maximum quantum winning probability. **A** 0.85; **B** 0.75; **C** 0.50; **D** 0.875.	**A**

Table 4.3: Task Catalog (Non-standard N1–N5): task statements, options, and ground-truth labels

Task	Task statement and options	Correct
N1	PT-symmetric Hamiltonian $H = p^2/(2m) + iV_0x$: which statement about the spectrum is most accurate? **A** PT symmetry ⇒ real spectrum. **B** spectrum can be real if PT unbroken, but for iV_0x PT breaks and eigenvalues are complex. **C** reduces to shifted harmonic oscillator. **D** PT symmetry only for bounded Hamiltonians.	**B**
N2	Jarzynski equality with initial coherence: effect of coherence on extractable work vs dephased state? **A** coherence always reduces work. **B** coherence has no effect on average work. **C** coherence can increase extractable work (resource). **D** Jarzynski equality breaks down with coherence.	**C**
N3	Resource theory of coherence: for σ with off-diagonals 1/4, find $C_{l_1}(\sigma)$ and IO reachability from incoherent ρ. **A** 1/4, impossible. **B** 1/2, impossible. **C** 1/2, possible. **D** 1, requires majorization check.	**B**
N4	Majorana braiding: exchange $\gamma_1 \leftrightarrow \gamma_3$; which unitary on encoded qubit? **A** σ_x; **B** σ_z. **C** $e^{\pm i\pi/4}\, e^{i\pi\sigma_z/4}$ (non-Abelian phase gate). **D** identity.	**C**
N5	Quantum metrology for $\lvert+\rangle^{\otimes n}$ under $H = \sum_i \sigma_i^z$: precision scaling $\Delta\theta$? **A** $1/\sqrt{n}$; **B** $1/n$ (achievable here). **C** $1/n$ in principle, but product state only achieves SQL. **D** $1/n^2$.	**C**

Table 4.4: Task Catalog (Numerical T1–T5): task statements, options, and ground-truth labels

Task	Task statement and options	Correct
T1	Harmonic oscillator ($m = \omega = \hbar = 1$): displaced Gaussian $x_0 = 2.5$, $\sigma = 1.5$; find excited-state probability $P_{\text{excited}} = \sum_{n\geq 1} \lvert c_n\rvert^2$. **A** 0.58 ± 0.03; **B** 0.68 ± 0.03; **C** 0.71 ± 0.03; **D** 0.83 ± 0.03.	**C**
T2	Barrier tunneling (split-operator): $V_0 = 48$, $L = 2$, $k = 10$, $x_0 = -5$, $\sigma = 0.5$; transmission probability after collision. **A** 0.23 ± 0.02; **B** 0.54 ± 0.02; **C** 0.73 ± 0.02; **D** 0.91 ± 0.02.	**B**
T3	Two-qubit concurrence under Heisenberg $H = \sum_{j=x,y,z} \sigma_j \otimes \sigma_j$ with $J = 1$; $\lvert\psi_0\rangle = \alpha\lvert 00\rangle + \beta\lvert 11\rangle$, $\alpha = 0.5$, $t = \pi/(4J)$. **A** 0.00 ± 0.02; **B** 0.29 ± 0.02; **C** 0.71 ± 0.02; **D** 1.00 ± 0.02.	**C**
T4	Quantum rotor VQE: $H = -\frac{1}{2} d^2/d\theta^2 + 4\cos(2\theta)$ on $[0, 2\pi]$; trial $\psi(\theta; \alpha, \beta) = Ne^{-\alpha(\theta-\pi)^2}(1 + \beta\cos(2\theta))$; find optimized E_0. **A** -2.00 ± 0.05; **B** -1.56 ± 0.05; **C** -0.42 ± 0.05; **D** $+0.73 \pm 0.05$.	**A**
T5	Lindblad dynamics with spontaneous emission $\gamma = 0.5$ and dephasing $\gamma_\phi = 0.1$, no drive; starting in $\lvert e\rangle$, find steady-state $\rho_{ee}(\infty)$. **A** 0.00 ± 0.01; **B** 0.25 ± 0.01; **C** 0.50 ± 0.01; **D** 1.00 ± 0.01.	**A**

hierarchy can be seen among the tiers: flagship models achieve 81.3% average accuracy, while mid-tier models reach 77.0% and fast models attain 67.0%, representing a 14.3 % spread between flagship and fast tiers.

While the temperature is set to 0, we still observe an average variance of 6.3 %, which may arise from residual sources of non-determinism such as floating-point precision effects, GPU parallelism, or implementation-level decoding. We observe that flagship models usually have the least variance, with GPT-5 exhibiting perfect consistency ($80.0\% \pm 0.0\%$), meaning that the model always chooses the same responses in the MCQ test. Qwen 2.5 Coder, a fast model, exhibits the highest variance ($73.3\% \pm 16.1\%$).

Table 4.5: Overall Model Performance Summary (Average over 3 runs, 20 tasks each)

Tier	**Model**	**Accuracy (%)**	**Cost/Task ($)**	**Tokens/Task**	**Time/Task (s)**
Fast	Claude 3.5 Haiku	56.7	$0.0016	671	7.7
Fast	GPT-3.5 Turbo	63.3	$7.62e-04	698	4.4
Fast	Gemini 2.0 Flash	71.7	$4.79e-04	1,293	7.8
Fast	Qwen 2.5 Coder 32B	73.3	$7.44e-04	5,085	80.1
Fast	DeepSeek R1 Distill 32B	70.0	$0.0028	3,505	136.4
Mid	Claude Sonnet 4	85.0	$0.014	1,214	15.8
Mid	GPT-4o	78.3	$0.0072	942	11.4
Mid	Gemini 2.5 Flash	66.7	$0.020	8,251	36.4
Mid	Qwen3 235B	75.0	$0.0022	4,061	108.3
Mid	DeepSeek V3	80.0	$7.73e-04	914	20.4
Flagship	Claude Sonnet 4.5	83.3	$0.015	1,247	17.1
Flagship	GPT-5	80.0	$0.031	3,306	55.7
Flagship	Gemini 2.5 Pro	78.3	$0.130	13,198	108.0
Flagship	Qwen3 Max	85.0	$0.019	3,384	79.8
Flagship	DeepSeek R1	80.0	$0.018	7,310	115.7

Figure 4.1 depicts the overall performance of the models on the tasks, with (a) showing a clear tier stratification, (b) depicting difficulty by task type, (c) identifying the best performers, and (d) showing the wide variation in individual task difficulty levels.

4.2.1 Individual Task Analysis

Individual task performance metrics offer the best insight into LLMs' performance on quantum mechanics tasks (see Figure 4.2): there is a broad range of difficulty, from D1 (commutator algebra, 97.8% mean accuracy), a near-universal success, to T2 (quantum

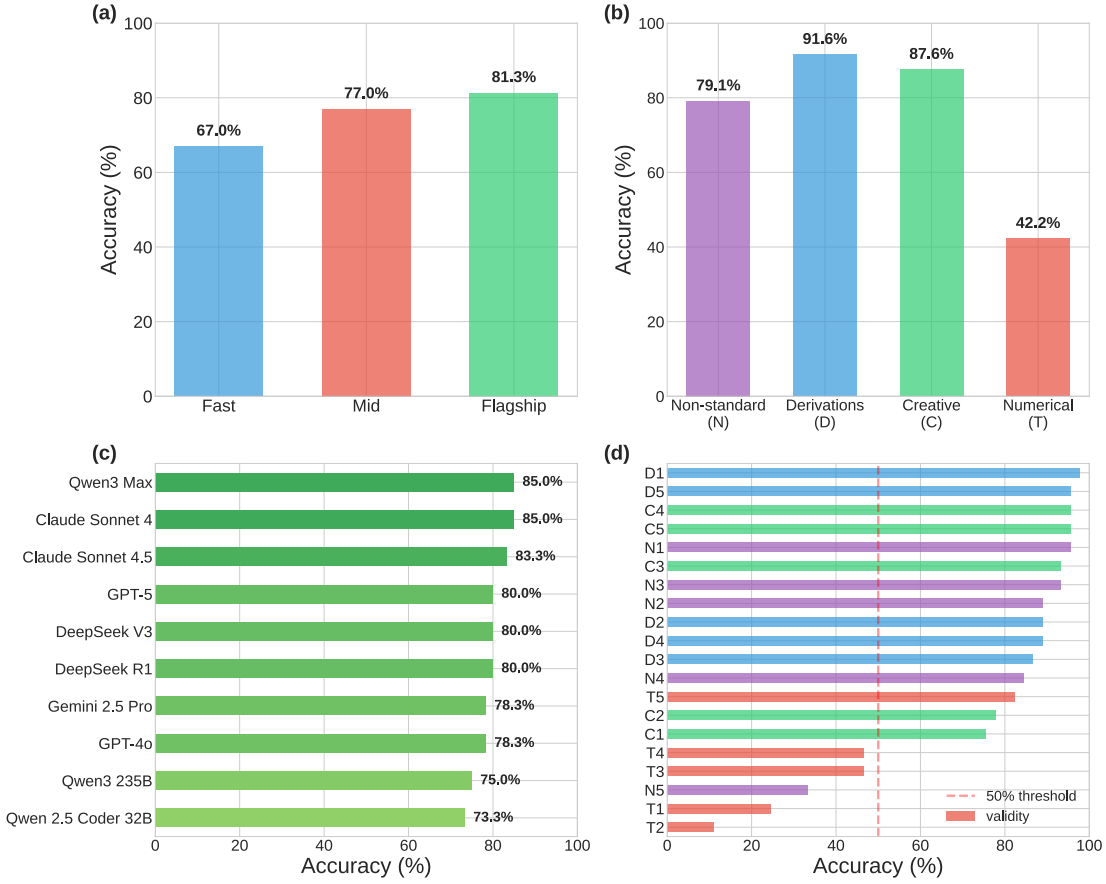


Figure 4.1: **Comprehensive Accuracy Analysis.** (a) Flagship models (81.3% avg) outperform mid-tier (77.0%) and fast models (67.0%) by 4.3 and 14.3 % respectively. (b) Each task type has a different difficulty level for the models.(c) Claude Sonnet 4 and Qwen3 Max are tied at the highest performance of 85.0%, immediately followed by Claude Sonnet 4.5 at 83.3 %.(d) Difficulty for Individual tasks ranges from 11.1% (T2: quantum tunneling) to 97.8% (D1: commutator algebra), exhibiting a significant variation among tasks.

tunneling, 11.1%), a near-universal failure. This is followed immediately by D5 (entropy maximization), N1 (weak measurement), and the design-optimization problems C4 and C5, all of which have a clear algebraic structure and well-circulated concepts. At the other end, we see that T1 (harmonic oscillator) has an average success rate of 24.4 % with a high standard deviation of 43.5%, which indicates conflicting computational formalisms employed by the models to solve it. We continue to notice this high-variability trend among numerical problems: T4 (variational eigensolver) and T3 (entanglement concurrence) each have means of 46.7% with $\sigma = 50.4\%$, while N5 (quantum metrology) has $\sigma = 47.7\%$ at 33.3% accuracy.

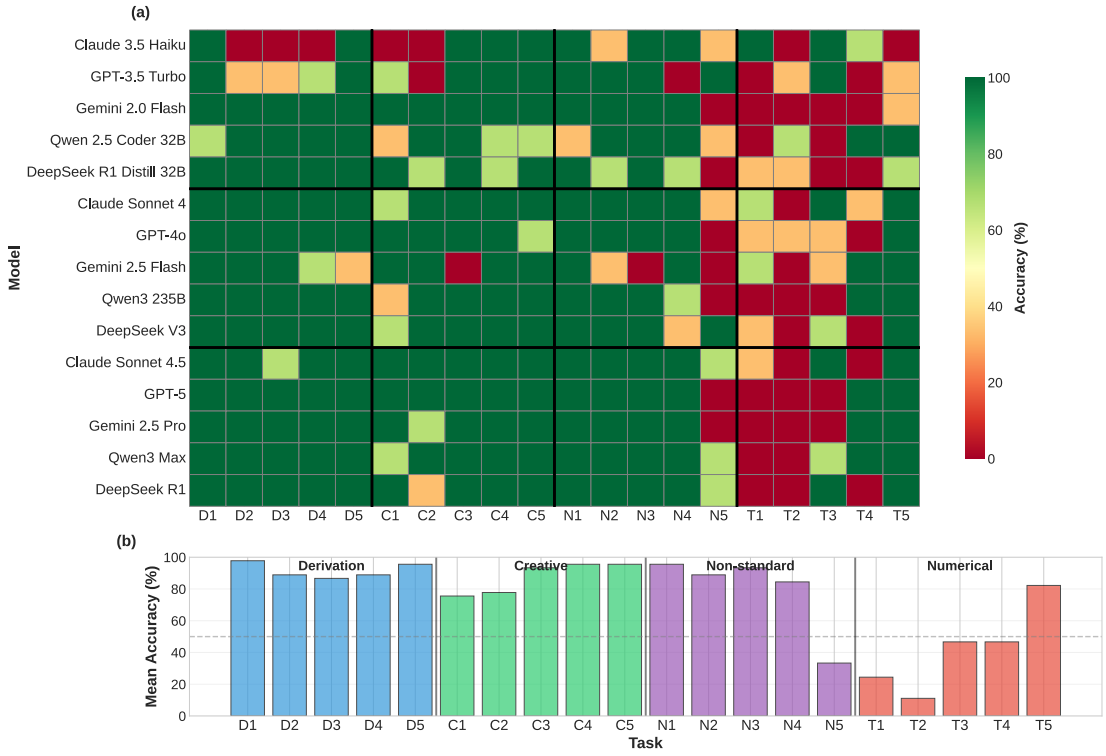
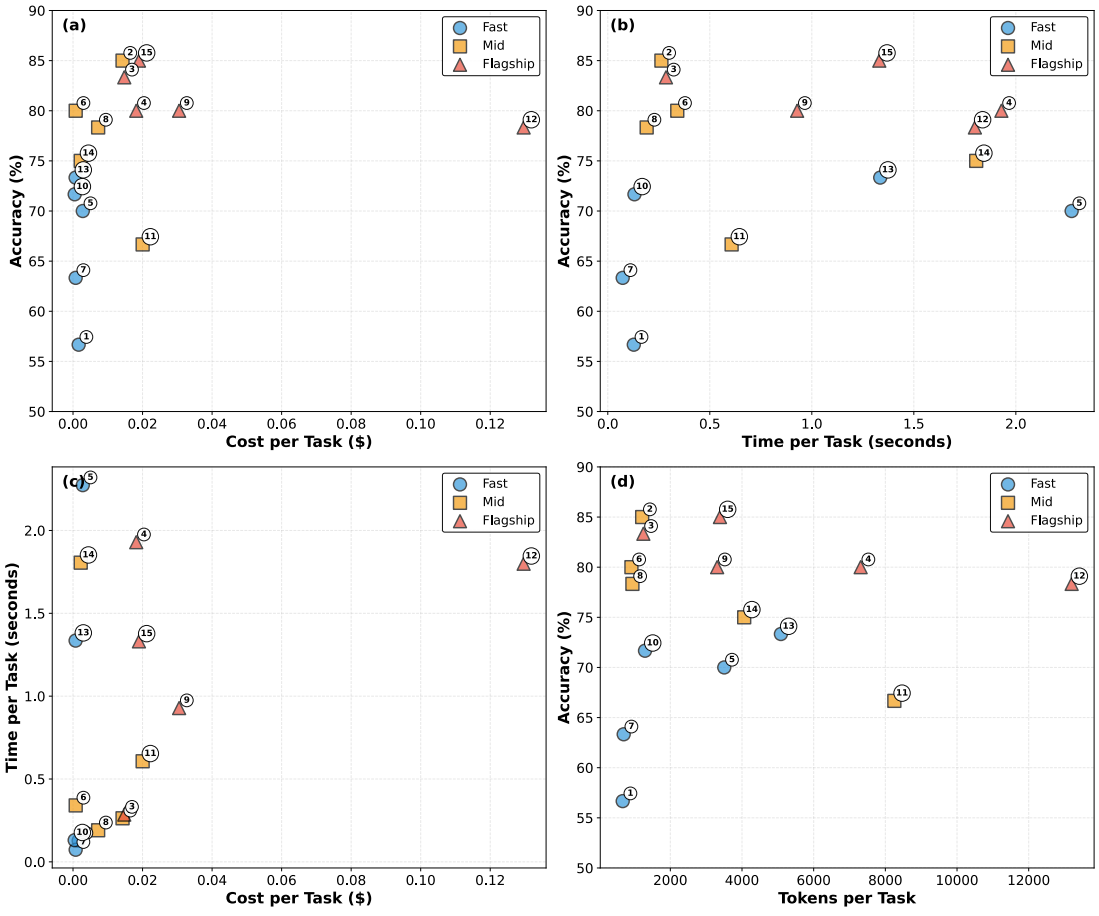


Figure 4.2: Individual task performance metrics : (a) Depicts accuracy for all model-task pairs . Black horizontal lines separate fast, mid-tier, and flagship models, and vertical lines separate task categories (D/C/N/T). (b) Mean accuracy per task across models, highlighting the wide spread in difficulty (11.1% to 97.8%).

Creative tasks such as C1 (POVM design), with mean 75.6% and $\sigma = 43.5\%$, and C2 (entanglement witness), with mean 77.8% and $\sigma = 42.0\%$, also show wide distributions in model performance, indicating that these problems are sensitive to training and architectural differences in LLMs. In most of these high-variance cases, flagship models show a clear lead (for example, T4 at 60% versus 33.3% for fast models, and C1 at 93.3% versus 60%).

Only two tasks show a true tier inversion, and both are the hardest numerical tasks. On T2 (quantum tunneling), fast models reach 26.7% while flagship models collapse to 0.0% (+26.7pp), and on T1 (Harmonic Oscillator) the fast tier again leads (26.7% vs 6.7%, +20.0pp). These outliers suggest that elaborate reasoning can sometimes be counterproductive when a direct calculation is the right strategy.

4.2.2 Cost-Accuracy Trade-offs



Model Key: 1: Claude 3.5 Haiku | 2: Claude Sonnet 4 | 3: Claude Sonnet 4.5 | 4: DeepSeek R1 | 5: DeepSeek R1 Distill 32B | 6: DeepSeek V3 | 7: GPT-3.5 Turbo | 8: GPT-4o | 9: GPT-5 | 10: Gemini 2.0 Flash | 11: Gemini 2.5 Flash | 12: Gemini 2.5 Pro | 13: Qwen 2.5 Coder 32B | 14: Qwen3 235B | 15: Qwen3 Max

Figure 4.3: Resource efficiency and cost-accuracy trade-offs: (a) Cost per task vs accuracy, showing that flagship models are about 33 × more expensive than fast models, for roughly a 14.3 percentage-point gain in accuracy. (b) Inference time per task by tier (flagship is about 1.6 × slower on average). (c) Cost vs time, highlighting tier separation spread within tiers. (d) Accuracy vs token usage, illustrating diminishing returns from longer responses.

Figure 4.3 provides a resource-efficiency analysis by comparing cost per task, tokens per task, time per task, and accuracy for different LLMs. While fast models cluster

around 67.0% average accuracy at roughly $0.0013, 47s, and 2,251 tokens per task on average, mid-tier models reach 77.0% average accuracy at about $0.0089, 39s, and 3,077 tokens per task, whereas flagship models average 81.3% at $0.0424, 75s, and 5,690 tokens per task. This implies that moving from the fast tier to the flagship tier provides a 14.3 pp increase in accuracy at about 33 × the cost and a relatively modest time penalty of 1.6 ×. There is also wide price variation within the flagship tier, with costs varying between $0.015 and $0.130 per task for relatively small accuracy differences, roughly between 78 and 85%, implying that price alone is not a reliable proxy for accuracy. Token usage rises steadily with tier, but accuracy does not scale proportionally, with several mid-tier models reaching flagship-level accuracy at far fewer tokens per task, indicating diminishing returns in the realm of longer, verbose reasoning.

4.2.3 Tool-Augmented Evaluation

To test the effect of enabling tool usage, we ran the models in both baseline and tool-enabled settings for three complete runs (see Figure 4.4 and Table 4.6). We observe that enabling tool usage improves average accuracy from 42.2% to 46.7%, a 4.4-percentage-point improvement, while increasing token usage by about 3×, going from 5,995 to 18,319 average tokens per task. But the averages hide strong task dependence: T1 (harmonic oscillator) improves substantially with tools, from 24.4% to 53.3%, a 28.9pp increase, whereas T3 and T4 improve modestly by 6.7pp each, while T2 (tunneling) drops slightly by 4.4pp and T5 (Lindblad steady state) degrades markedly from 82.2% to 66.7%, a 15.6pp reduction. This pattern suggests that enabling tool access alone does not guarantee better LLM performance on numerical problems, but it benefits them when there is a straightforward mapping to a calculation procedure or numerical recipe. In the absence of this, models struggle to choose the right formalism and approach to solve numerical problems.

4.2.4 Reproducibility Analysis

The summary of reproducibility of model responses across three deterministic ($T = 0$) runs can be seen in Figure 4.5. Flagship models are seen to be the most stable, while fast (lighter) models exhibit higher variance. GPT-5, for instance, is perfectly consistent in

Table 4.6: Tool-Augmented vs Baseline Performance on Numerical Tasks

Task	Description	Baseline (%)	Tool-Aug (%)	Δ Acc (pp)	Avg Tokens Baseline	Avg Tokens Tool
T1	Harmonic Oscillator	24.4	53.3	+28.9	6,077	15,697
T2	Quantum Tunneling	11.1	6.7	-4.4	4,583	15,048
T3	Entanglement	46.7	53.3	+6.7	7,493	11,583
T4	VQE Ground State	46.7	53.3	+6.7	7,865	37,875
T5	Lindblad Steady State	82.2	66.7	-15.6	3,959	11,394
Overall	All T tasks	**42.2**	**46.7**	**+4.4**	**5,995**	**18,319**

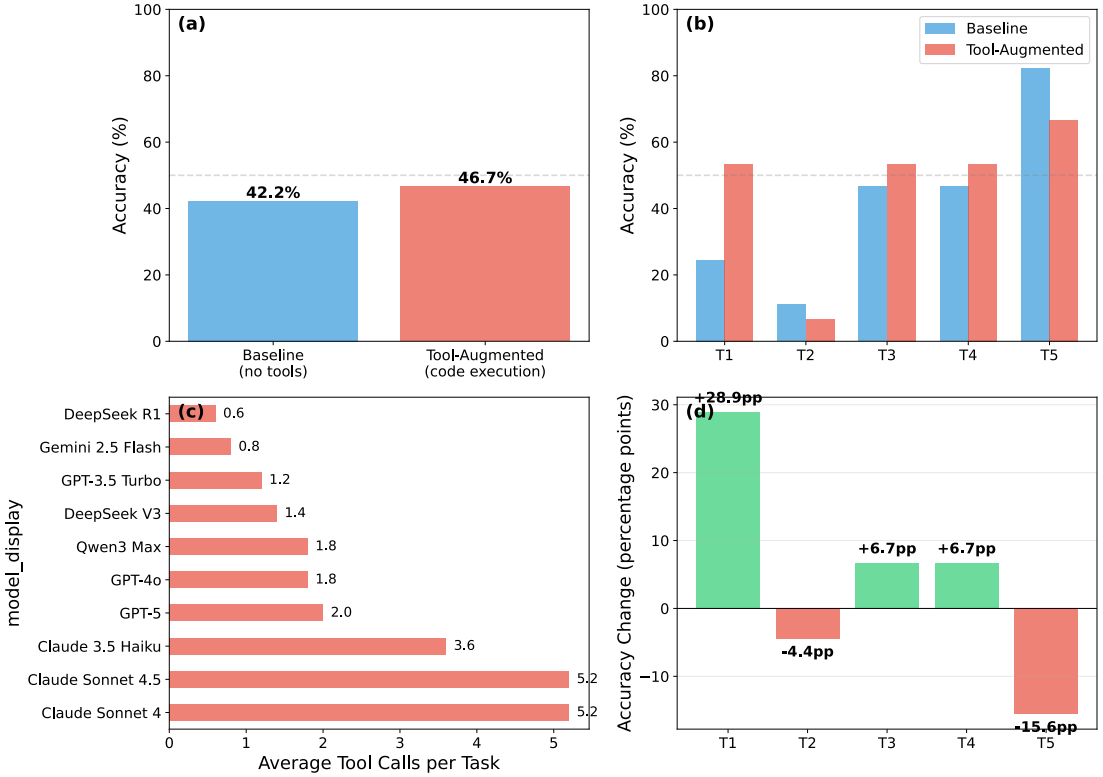


Figure 4.4: Tool augmentation on numerical tasks. (a) Overall accuracy with and without code execution. (b) Per-task comparison for T1–T5, showing that gains are highly task-dependent. (c) Tool-call frequency by model (top 10 shown; mean 1.8 calls per task). (d) Accuracy change by task: T1 +28.9pp, T3/T4 +6.7pp each, T2 -4.4pp, and T5 -15.6pp.

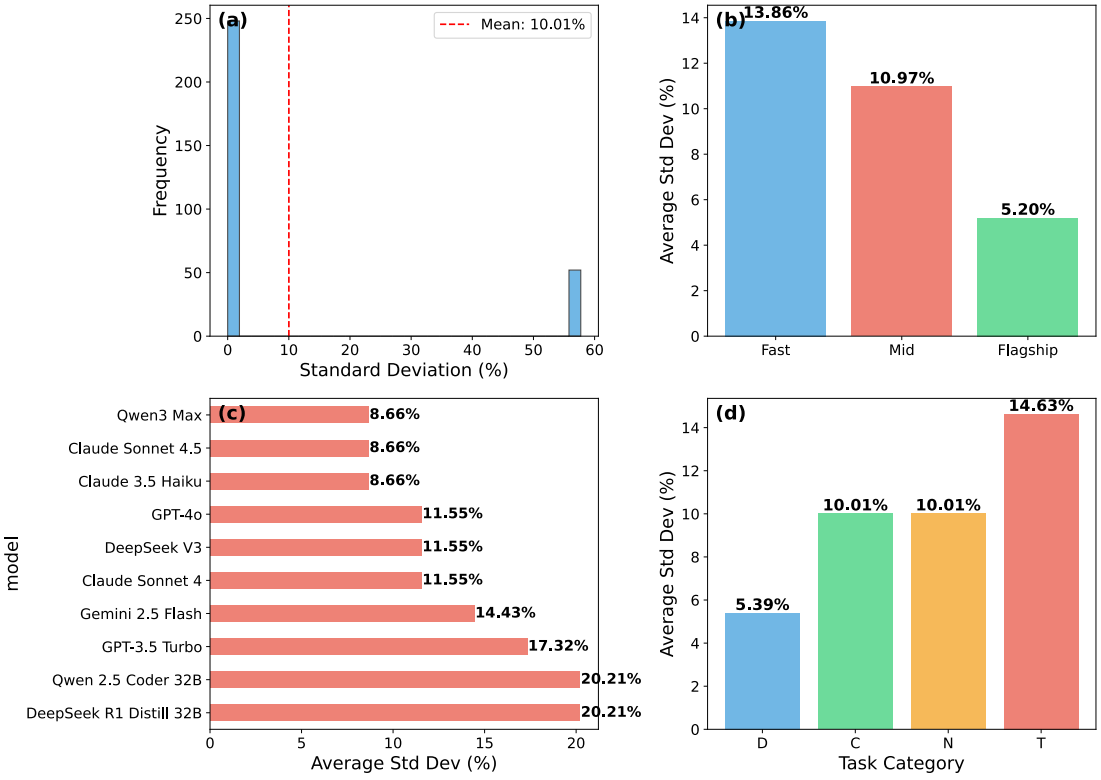


Figure 4.5: Reproducibility across three runs at temperature $T = 0$ (deterministic decoding). Panel (a) shows the distribution of per-pair standard deviations. Panel (b) aggregates variance by tier (fast 7.4pp, mid-tier 6.3pp, flagship 5.3pp). Panel (c) shows model-wise variance (GPT-5 at 0pp; Qwen 2.5 Coder highest at 16.1pp). Panel (d) aggregates variance by task category (Derivations (D) lowest at 5.4pp; Numerical (T) highest at 14.6pp).

the responses chosen for the quantum-mechanics MCQ tasks, with 80.0% ± 0.0%, while Qwen 2.5 Coder exhibits the largest spread (16.1pp), given that it is a lighter model. By task type, derivations are the most reproducible (5.4pp), while numerical tasks are the least reproducible (14.6pp), which also mirrors the difficulty levels (see Figure 4.1).

4.3 Conclusion

We evaluated 15 LLMs on quantum mechanics tasks requiring four kinds of cognitive abilities, and the results indicate a clear tier stratification, with the flagship models outperforming fast models by 14.3 %. Models' performance varies significantly with task type, where Derivations are the easiest (91.6%), followed by Creative (87.6%) and Non-standard (79.1%) tasks, while Numerical problems pose the biggest challenge (42.2 %). Within each category, individual tasks span a wide range of difficulty (11% to 98%), and enabling tool usage leads to task-dependent results: while the overall accuracy increases modestly (from 42.2% to 46.7%), the effect is highly task-dependent (large gains on T1 [harmonic oscillator], modest gains on T3 [entanglement concurrence]/T4 [variational eigensolver], and degradations on T2 [tunneling] and especially T5 [Lindblad steady state]), indicating that tools help most when the calculation is direct rather than requiring a careful choice of formalism. When it comes to reproducibility of results, we note that flagship models are the most stable, while fast models show the most variance.

Our benchmark provides a foundation for assessing AI capabilities in quantum physics. The findings highlight that progress requires not just more powerful models or more tools, but intelligent integration of reasoning strategies with task characteristics. Future work could expand coverage to additional quantum domains (field theory, many-body systems, quantum chemistry) and increase task density per category. This foundation supports the development of agentic AI systems that leverage LLMs for quantum-physics applications, which we shall explore in the next chapter.

Data and Code Availability

All tasks, verifiers, evaluation scripts, and results are publicly available at https://github.com/rithvik1122/llm_qm_benchmark.

"If a machine is expected to be infallible, it cannot also be intelligent."

Alan M. Turing

5

Aṇubuddhi: Multi-Agent AI System for Quantum Optics Experiment Design and Simulation[1]

The problem of the automated design of quantum experiments has seen at least a decade of contributions, beginning with the early pioneering work by Mario Krenn, Anton Zeilinger, and colleagues when they found that the inverse problem of trying to come up with the right *optical configuration* for designing specific quantum states was too challenging for human intuition alone. Therefore, they came up with a computer algorithm, MELVIN [149], which generated quantum optical experiments by randomly assembling discrete components from a predefined toolbox, symbolically evaluating the resulting states against strict target criteria, and accelerating the search by reusing successful sub-configurations, rather than by optimizing a cost function or performing gradient-based learning. This was followed by THESEUS[150], which reformulated experiment design as a weighted graph problem and employed continuous optimization of edge weights (using BFGS with L_1 regularization) combined with an iterative, greedy topological pruning step that removes edges while preserving fidelity, effectively turning

[1]The contents of this chapter have been presented in: S. K. Rithvik, "Aṇubuddhi: A Multi-Agent AI System for Designing and Simulating Quantum Optics Experiments," arXiv preprint, https://arxiv.org/abs/2512.15736 (2025).

the search into a backward simplification of an initially complete graph. Building on the graph-based approach of THESEUS, PyTheus [151] generalizes the problem of inverse design from state creation to a much broader set of quantum optics tasks, including measurement, communication protocols, and multi-photon gates. It begins with an initial graph configuration based on the allocated resources and target state and then performs continuous gradient-based optimization of the complex edge weights using BFGS or L-BFGS-B with loss functions that combine fidelity and count-rate objectives. This is then followed by a discrete pruning of the edges to simplify the graph, while imposing topological and physical constraints specific to different experiment types. This makes it faster and more broadly applicable compared to THESEUS.

Other works like AdaQuantum demonstrated that hybrid genetic-algorithm and deep-neural-network approaches could optimize experimental parameters for quantum state engineering [152]. AdaQuantum's key innovation was the use of evolutionary algorithms to explore the vast design space while employing neural networks to predict experimental outcomes, significantly reducing the computational cost of optimization. One way to characterize these approaches is that all of them attempt to find the optimal configuration of optical elements, like arranging pieces in a puzzle or trying to construct an object out of building blocks, employing optimization techniques that do not have a baked-in physics intuition other than those rules that are explicitly enforced for pruning so as to find the optimal configuration from a combinatorially large search space, sometimes called *heuristic search*[153]. However, we saw in the previous chapter how neural networks trained with large quantities of text data, called Large Language Models (LLMs), exhibit physics intuition and can indeed solve complex quantum mechanics problems[147]. Therefore, in this chapter, we explore LLMs as the *Intuitive Optimizer* for exploring the space of quantum optical elements and arriving at the appropriate configurations for different experiments. This approach has yet another major advantage in that it removes the need for non-user-friendly *Intermediate representations* like graphs in the case of THESEUS and PyTheus, and chromosomes in the case of AdaQuantum, which required further interpretation to express the results in intuitive terms and thereby made them harder to adopt.

Other researchers have explored this aspect of LLMs. Boiko et al. developed Coscientist, which showed that GPT-4 could independently design, plan, and carry out

complex chemistry experiments [154]. In quantum physics, the k-agents framework introduced LLM-based agents to automate quantum-computing lab experiments [155], where K-agents organize lab knowledge and manage multiple specialized agents to carry out calibrations and characterizations of quantum processors. This system has successfully completed single-qubit and two-qubit gate calibrations from natural-language instructions. The AI-Mandel system, created by Arlt, Gu, and Krenn, marks a significant step forward. They developed an LLM agent that generates and implements original research ideas in quantum physics [156]. AI-Mandel formulates hypotheses in natural language and automatically implements them using specific tools like PyTheus. It has also contributed to published research, including the discovery of new quantum teleportation variants.

In this chapter, we present Aṇubuddhi, a Multi-Agent AI system that designs and simulates quantum optics experiments based on natural language conversation.

5.1 Cognitive Architecture

LLMs by themselves are passive blocks of intelligence and need to be appropriately *prompted* in order to elicit responses. However, when they are embedded inside a *Cognitive Architecture* consisting of memory, a smart way to manage and route context, and a compartmentalized structure with each *agent* performing a certain role, as defined by the *system prompt* and supported by control logic and tool usage, their practical effectiveness increases substantially.

Our system Aṇubuddhi, which translates into "Atomic Intelligence" from Sanskrit, consists of a three-layered cognitive architecture (see Figure 5.1). Being a conversational system, the first layer assesses the intent of the user query and routes it either into DESIGN or CHAT mode. In CHAT mode, the system does not attempt to create or modify the quantum-optics designs that might be the result of previous queries, but instead tries to answer questions about the existing designs (in the panel that houses the design in the GUI, all this is made available as context to the LLM so that it can answer questions appropriately).

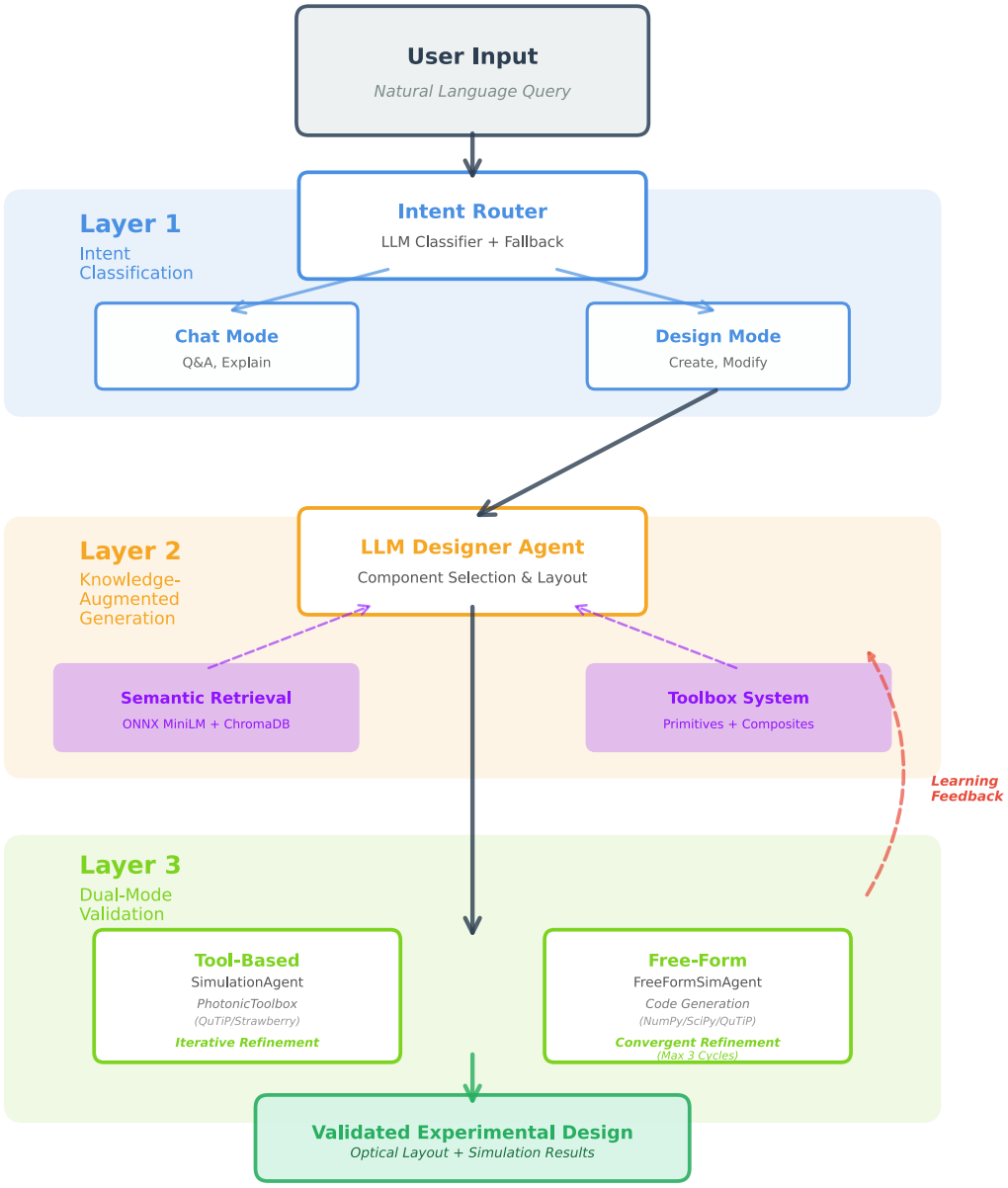


Figure 5.1: The cognitive architecture of Aṇubuddhi consists of three layers. The first layer routes the intent into CHAT/DESIGN mode. The second layer generates a design based on the toolbox of available optical elements and the third layer performs a simulation and assesses the quality of the design-simulation alignment

The second layer designs experiments by arranging a set of optical components on an *optical table* by presenting the coordinates of various optical elements, along with their details such as type and parameters (800 nm wavelength for a laser source, for example, and focal length for a lens), in a JSON file that then gets rendered into an optical table by importing the Python module pyopticaltable [157]. This is achieved using a *Retrieval Augmented Generation (RAG)* strategy [158] where a *Toolbox* of primitives and learned

composites is made available to the LLM as context. The primitives are the optical elements and their descriptions, such as beam splitter, mirror, lens, laser source, etc., and the learned composites are assemblies of these elements like a Mach-Zehnder interferometer setup, Hong-Ou-Mandel setup, Michelson interferometer setup, path-delay stage, etc., which are learned by the system based on designs that have been approved by previous users of the software who generated them and thought they were good enough to be stored for future use. These learned composites are stored and retrieved using ChromaDB[159] with BGE embeddings[160]. Storing learned composites helps build a knowledge database of useful optical setups that can be re-used by other researchers without designing from scratch each time, and this approach not only saves resources but also enables compositionality, where progress is accelerated through compositional reuse, whereby complex capabilities are constructed from previously learned intermediate building blocks rather than primitive elements. In the case that an experiment requires custom components not present in the toolbox, but known to the LLM through the physics knowledge and intuition stored in its weights in the form of memory, we give the agent the ability to include custom elements in the optical-table JSON dictionary with an appropriate description.

The first iteration of output from the *Designer Agent* is usually not optimal and includes errors like incorrect placement of optical elements, an insufficient number of elements chosen for the task, incorrect parameter values, etc., which get flagged by an internal *validation phase* where the output of the designer agent is *reviewed* and this assessment is used as feedback to improve the design. This process is repeated three times, and if a design passes the review, it is reported before the completion of the full three rounds; otherwise, the final design after three rounds is reported.

While internal validation in layer 2 improves the quality of the generated setup, a better assessment can be obtained by performing a quantitative simulation of how the input state evolves as it passes through the various optical elements in the setup, finally producing the output state. One can then check whether the output state aligns with the desired output of the user who is attempting to design the setup. This is performed in layer 3 by means of two modes, namely QuTiP, where the Quantum Toolbox in Python[161] module is used for the simulation, which offers a set of methods used to simulate the action of beam splitters, Fock states, etc., but lacks the ability to model

temporal dynamics, continuous-variable systems, or complex atomic systems, which leads us to the second mode, which we call FreeSim mode, where the LLM is given a free hand to use any Python modules it chooses, including NumPy, SciPy, QuTiP, etc., to adequately model the design coming out of the designer agent.

FreeSim mode relies on a six-stage *Convergent self-refinement* strategy[162, 163] to ensure reliability. The first stage classifies the problem into four physics domains (Continuous Variable systems, Atomic Systems, Temporal Dynamics, Discrete Photonic Systems). This enables the next stage, which provides specific guidance based on the category chosen and consists of prompts of around 5000 characters advising the model on what it should specify in the design, such as parameter ranges (for example, frequency ranges, focal-length ranges, etc.) and specific guidance to avoid mistakes like incorrect phase-matching conditions in SPDC or using Fock states to model temporal dynamics instead of Gaussian wavepackets, etc. This is followed by a pre-execution review stage where four things are explicitly checked: 1) whether the formalism was correctly chosen, 2) whether all optical components necessary are present in the design, 3) whether physical parameters were chosen appropriately, and 4) whether the mathematical approach used to model the simulation was chosen correctly. FreeSim code failing this review is rerouted to refinement, which has 3 attempts by default (and can be edited as per the user's preference). This is then followed by code execution in an isolated environment with error capture, which then passes through a stage that checks for the alignment between the intended design and the FreeSim simulation and provides a rating between 0 and 10. If the scores are below 6, the feedback is used to refine the code using specific instructions, and the refined code is then executed. While this six-stage process is not foolproof, it offers a path to test the quality of the FreeSim simulations.

5.2 Results

Aṇubuddhi was evaluated on 13 quantum optics experiments spread across three tiers that form a rigorous experimental benchmark. The first tier consists of standard quantum-optics experiments that are found in textbooks, namely Bell-state generation based on SPDC [164, 165], Hong-Ou-Mandel interference [166], delayed-choice quantum erasure[11], Mach-Zehnder interferometry [167, 168], and Michelson interferome-

try [169]. The second tier covered quantum-information protocols like Quantum Key Distribution (QKD) using BB84[170], hyperentanglement [171], quantum teleportation (discrete) [172], generation of the GHZ state [173], and Franson interferometry for time-bin entanglement [174]. The third tier consisted of advanced quantum technologies like quantum computational advantage using 4-photon Boson sampling [175, 176], frequency conversion from telecom to visible frequency without losing the quantum information [177, 178], and Electromagnetically Induced Transparency (EIT) in atomic vapor [179, 180], which require multi-domain physics modeling. For purposes of brevity, we present 3 (one per tier) out of the 13 results here, while the full set of experiments and a detailed account can be found in [181, 182].

A note regarding the Optical Table Diagrams: The optical layouts shown in this section are generated from Aṇubuddhi-specified component positions and beam paths. While component selection and beam connectivity are correct, geometric angles and the orientation of element icons may not be optically precise. These diagrams are schematic representations: what matters is the component order and the beam-path connectivity. Beam colors represent separate paths for readability (not frequencies, polarizations, or other degrees of freedom).

5.2.1 Hong-Ou-Mandel Interference

Hong-Ou-Mandel (HOM) interference is a test of two-photon indistinguishability: when two identical photons reach the two inputs of a 50:50 beam splitter at the same time, the two coincidence pathways interfere destructively and the photons preferentially bunch into the same output port. Figure 5.2 shows the optical-table layout generated by Aṇubuddhi which arranges the optical components from the toolbox as follows: The setup begins with a 405 nm pump laser (1), which is focused by a lens (2) into a Type-II BBO crystal (3) to generate photon pairs via SPDC. A PBS (4) then cleanly separates the orthogonally polarized photons into two spatial arms, while pump-blocking filters (5-6) remove residual pump light. Next, half-wave plates (7-8) are used to rotate and align the polarizations so that polarization does not trivially label the two arms, and mirrors (9-10) and (12-13) route the beams toward a common interference point. A delay stage (11) in one arm provides the controlled path-length change needed to scan

the relative arrival time and the two arms are combined at a 50:50 non-polarizing beam splitter (14) where the HOM interference occurs. Finally, 810 nm interference filters (15-16) enforce spectral matching at the outputs, coupling lenses (17-18) collect the light into the detectors, two SPADs (19-20) register single-photon clicks, and a coincidence counter (21) records joint events within a fixed timing window as the delay is scanned.

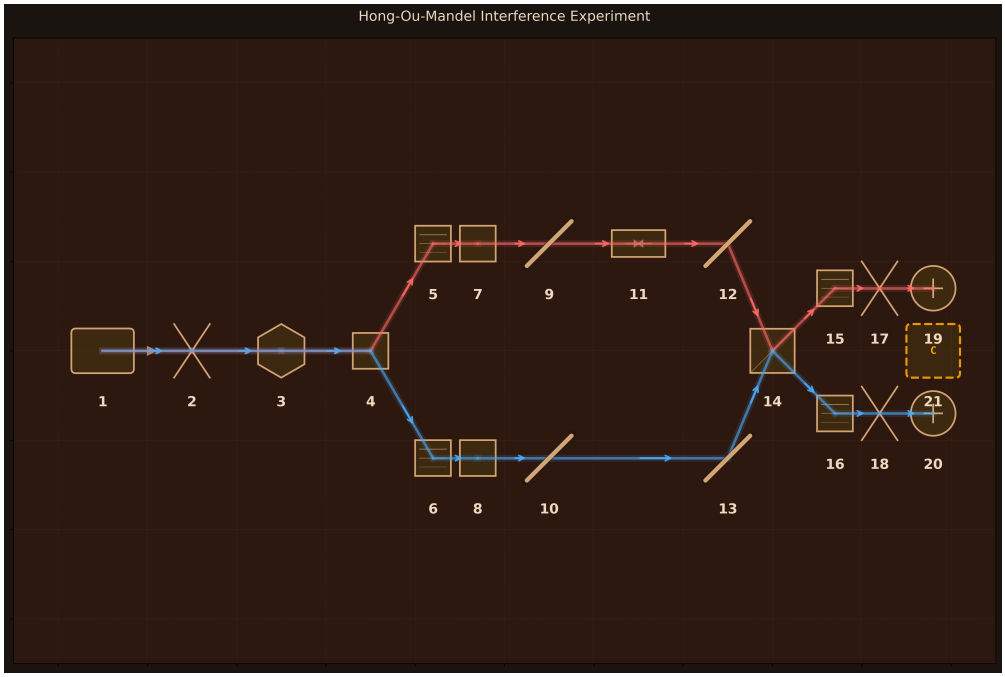


Figure 5.2: Optical table layout for a Hong–Ou–Mandel interference measurement. Type-II SPDC in BBO generates 810 nm photon pairs from a 405 nm pump; a PBS separates the arms; half-wave plates align polarization; a delay stage controls temporal overlap; and interference occurs at a 50:50 beam splitter before coincidence detection.

The design is a standard way to realize a HOM measurement; however, one can see that there are implicit assumptions, such as perfect indistinguishability at the final 50:50 BS, while in practice residual spatial misalignment, polarization mismatch, and Type-II birefringent walk-off (temporal/spectral mismatch) can dampen the dip. An experimentalist would restore visibility by careful mode matching (using single-mode fiber coupling as a spatial filter), tuning polarization with wave plates, and adding birefringent compensation or extra delay to overlap the wavepackets (to overcome the spatial and temporal walk-off), etc. The agentic system relies on Claude Sonnet 4.5 to come up with this design, and we saw in the previous chapter (see Table 4.5) that it had an average accuracy of 83.3% on quantum-mechanics tasks. Added to this is the fact that HOM is a well-circulated concept. Further details about

this design, including the analysis generated by Aṇubuddhi as presented to the user can be found at https://github.com/rithvik1122/Anubuddhi/tree/main/Results_FreeSim/hong-ou-mandel_interference_experiment_freeform_2025-11-28_13-48-50

5.2.2 Quantum Key Distribution(QKD) - BB84 Protocol

BB84 quantum key distribution is a protocol in which the act of measurement is a security test: Alice encodes random bits in one of two mutually unbiased polarization bases, and Bob measures in a randomly chosen basis so that any intercept-resend attack shows up as an increased quantum bit error rate (QBER). Figure 5.3 shows the optical-table layout produced by Aṇubuddhi. A single-photon source (1) is attenuated (2) and fed into an active polarization encoder at Alice (3) that prepares H/V or D/AD states. The photons propagate through a 10 km polarization-maintaining fiber channel (4) to Bob, where a 50:50 beam splitter (5) passively selects the measurement basis by routing each photon into a rectilinear arm (6, 8) or a diagonal arm (7, 9-10). Mirrors (11-14) direct the four measurement outcomes onto SPAD detectors (15-18), and timing/sifting electronics (19) correlate clicks with Alice's basis announcements over an authenticated classical channel (20) to perform basis sifting, QBER estimation, and key distillation.

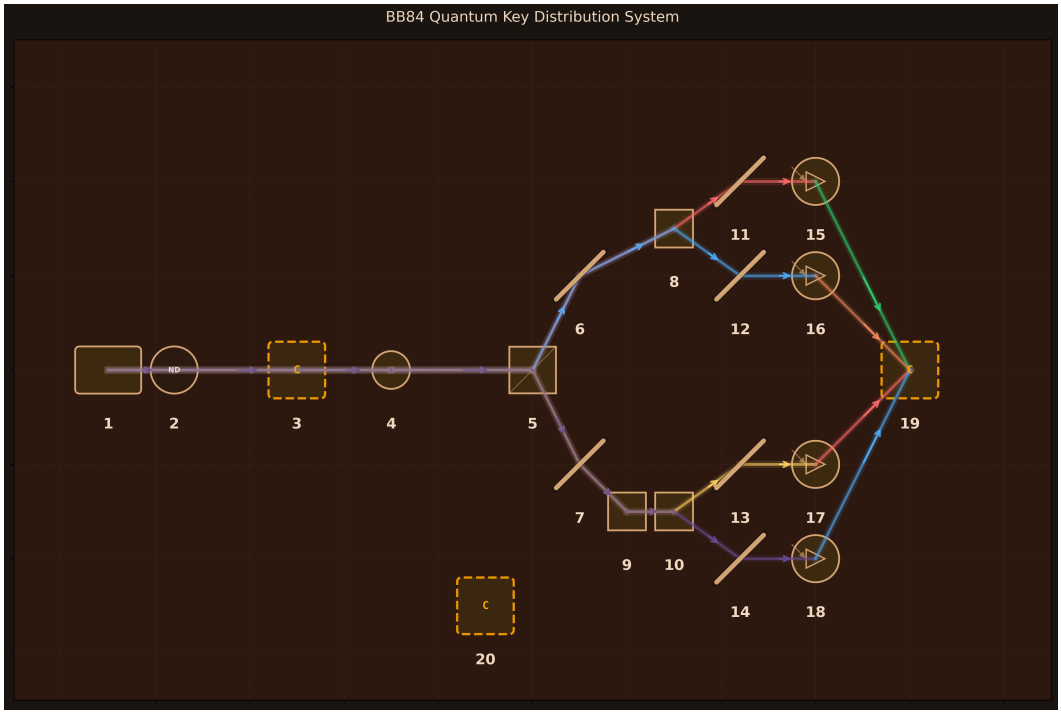


Figure 5.3: Optical table layout for BB84 quantum key distribution as generated by Aṇubuddhi. Alice prepares polarization-encoded single photons, the fiber channel transmits them to Bob and a passive 50:50 basis selector routes each photon to rectilinear or diagonal analysis before detection and classical sifting.

As a design, this captures the standard BB84 experimental logic: active state preparation at Alice's end, passive basis selection at Bob's end, and four single-photon detectors to enable classical post-processing, so the security signature is directly readable as QBER. The main limitation is that the layout contains idealizations and ignores the following: polarization drift and calibration in fiber, source non-idealities (multi-photon leakage motivating decoy states), and detector side effects (efficiency mismatch, afterpulsing, and timing jitter). In practice, experimentalists might add polarization tracking or compensation, decoy-state modulation, and tighter timing synchronization so that the measured QBER remains meaningful and the extracted key rate remains secure. Further details can be found at https://github.com/rithvik1122/Anubuddhi/tree/main/Results_FreeSim/bb84_quantum_key_distribution_system_freeform_2025-11-28_18-56-03.

5.2.3 Electromagnetically Induced Transparency (EIT) in Warm Rb-87 Vapor

Electromagnetically Induced Transparency (EIT) is a phenomenon seen in atomic systems when a strong coupling field opens a narrow transparency window for a weak probe by preparing a dark superposition state in a three-level Λ system. We generated two EIT packages (QuTiP and FreeSim) and present the QuTiP version here because it uses the standard density-matrix (Lindblad) formalism, even though its parameterization is imperfect. Figure 5.4 shows the Aṇubuddhi-designed optical-table layout. A weak probe laser (1) is frequency shifted and scanned with an AOM (2), which is then prepared with polarization optics (3-4) and collimated (5). In parallel, a strong coupling laser (6) is polarization-conditioned (7-8) and collimated (9). A dichroic element combines the two beams into a single spatial mode (10), and an iris (11) enforces overlap through a temperature-controlled Rb-87 vapor cell (12). After the cell, a collection lens (13) sends the transmitted light through a narrowband probe filter (14) to block the coupling field before reaching the photodiode (15). A lock-in amplifier (16) and the RF drive electronics (18) enable reading of the narrow EIT feature, while the cell temperature controller (17) controls the atomic density.

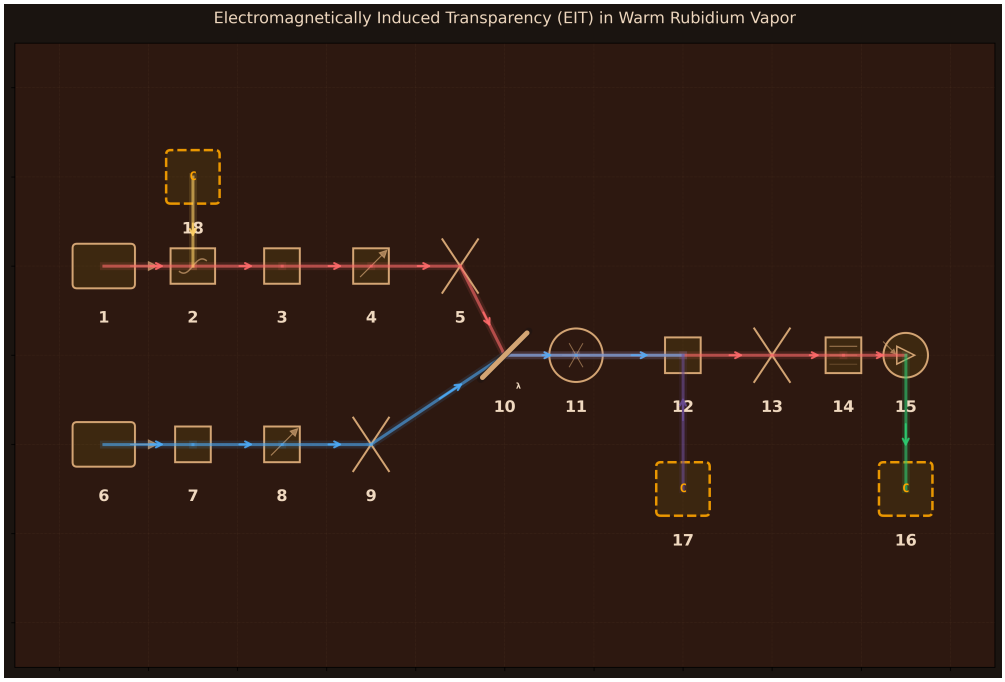


Figure 5.4: Optical table layout for an EIT measurement in warm Rb-87 vapor as generated by Aṇubuddhi. A weak probe beam and a strong coupling beam are independently conditioned, combined collinearly, and sent through a heated vapor cell. Probe transmission is isolated by spectral filtering and measured with phase-sensitive detection.

This design appears to be a reasonable, first pass schematic of an EIT experiment: it includes independent preparation of probe and coupling fields, collinear overlap through the cell and filtered sensitive detection at the output. The main limitation is that the accompanying QuTiP simulation does not reliably validate the design because the key physical parameters are off by orders of magnitude (in particular, the atomic density/optical depth is far too small, so the medium is already essentially transparent and so no EIT contrast can appear). It also violates the weak-probe regime and scans detunings in a way that does not enforce a two-photon resonance condition realistically. It also omits warm-vapor effects (velocity averaging for Doppler broadening, transit-time broadening and propagation/slow-light physics that requires Maxwell-Bloch modeling). Further details about the design can be found at https://github.com/rithvik1122/Anubuddhi/tree/main/Results_QuTiP/Electromagnetically_Induced_Transparency_EIT_in_Warm_Rubidium_Vapor_20251126_124851.

5.3 Conclusion

In this chapter, we use LLMs as an *Intuitive Optimizer* to find configurations of optical elements that realize a broad range of experiments found in quantum optics. Compared to previous works in this field of automated design of quantum experiments, Aṇubuddhi offers a significant advantage: it eliminates the need for non-user-friendly intermediate representations to encode physics ideas and instead uses natural-language representation. It also offers more details about, for example, the range of parameters of the chosen components: wavelength, laser power, beam diameter, lens focal length, filter center wavelength and bandwidth, detector type, detector efficiency, detector dark count rate, timing jitter, coincidence window, etc., which earlier algorithms did not specify. However, this is also where Aṇubuddhi can sometimes choose parameters that are off by orders of magnitude, as in the case of the atomic-vapor density in EIT, and assume ideal conditions, ignoring effects like birefringent walkoff, polarization mismatch, efficiency mismatch among coincidence detectors, spatial misalignment, etc., that occur in practical lab settings. This can potentially be remedied by adding an additional validation layer that looks specifically for these types of issues in the designs and provides feedback to the designer agent so that it can improve the design further, at the cost of increased token usage.

Part III

Quantum-Inspired Constructive Foundations for Mechanized Reasoning

"We must know. We will know."

David Hilbert

6

Fundamental Limits of Mechanized Reasoning: A Quantum-Inspired Perspective

[1]

In the chapters of the previous two parts, we explored the applications of Artificial Intelligence to quantum systems, and having seen their effectiveness, a natural question arises: ***Are there any fundamental limits on what AI can do?*** Contemporary scaling laws[67–69] seem to indicate that bigger models trained on larger corpora of data and more extensive computational resources could lead to increased problem-solving ability across diverse domains, as evidenced by consistent improvements on a wide range of benchmarks. But are there problems that can never be solved by mechanized reasoning? To answer this question, we must inevitably revisit the previous century, when Kurt Gödel [183] and Alan Turing [12] came up with results that are collectively referred to as *Undecidability* results, which posit the existence of well-defined statements within a formal system that can be neither proven nor disproven by any

[1] Detailed proofs and extended treatment can be found in (i) S. K. Rithvik, "A Canonical Bijection Between Finite-Decimal Real Numbers and Natural Numbers with Constant-Time Enumeration Formulas," https://arxiv.org/abs/2508.10750 (2025); and (ii) S. K. Rithvik, "Diagonal Arguments and Infinite Dependencies: Analyzing Classical Undecidability and Universality Under Finite Resource Constraints," Preprints, https://doi.org/10.20944/preprints202510.2040.v1 (2025).

algorithmic procedure.

It is interesting to note that both of these results rely on *Diagonal Arguments* (diagonalization ideas), in the now-standard form introduced explicitly by Georg Cantor in 1891 [72]. Cantor had already proved the uncountability of the real numbers in 1874 [184], but the diagonal method itself dates to 1891:

Theorem 6.1 (Cantor's Classical Diagonal Argument)**.** *The set of infinite binary sequences* $\{0,1\}^{\mathbb{N}}$ *is uncountable.*

Proof (classical, brief). Begin with the assumption that $\{0,1\}^{\mathbb{N}}$ is countable, which implies the existence of the following *complete enumeration*:

$$n_1 = (b_{11}, b_{12}, b_{13}, b_{14}, \ldots) \tag{6.1}$$

$$n_2 = (b_{21}, b_{22}, b_{23}, b_{24}, \ldots) \tag{6.2}$$

$$n_3 = (b_{31}, b_{32}, b_{33}, b_{34}, \ldots) \tag{6.3}$$

$$\vdots \tag{6.4}$$

Now, construct the diagonal object: $m = (m_1, m_2, m_3, \ldots)$ where:

$$m_i = \begin{cases} 0 & \text{if } b_{ii} = 1 \\ 1 & \text{if } b_{ii} = 0 \end{cases}$$

Since the constructed diagonal object m varies at the j^{th} entry from n_j for every $j \in \mathbb{N}$, it cannot be a part of the enumeration, thereby yielding a contradiction.

□

Since Cantor's original aim was to establish the inequality $|\mathcal{P}(\mathbb{X})| > |\mathbb{X}|$, where $\mathcal{P}(\mathbb{X})$ denotes the collection of all subsets of $\mathbb{X}$, the binary-sequence formulation was a convenient representation. However, if the elements b_{ij} can have more values than 0 and 1, say 0 to 9, then we get the decimal representation of the proof, which is more popular. Irrespective of the radix, the important thing to note is the critical dependence of this method on *completed infinities*: **To construct the diagonal element, one needs to store an *infinite* number of elements and, out of that, construct the diagonal**

object, which requires *infinite time* in terms of computational steps.

6.1 Diagonal Arguments in Undecidability Results

Cantor's construction is more than a set-theory curiosity: it is a reusable template for producing a "self-escaping" object from the assumption that some domain has been completely listed. In classical logic and computation, the same diagonal move reappears in two famous forms. In Gödel's setting, the diagonal object is a sentence G that ends up talking about its own provability inside a formal system $\mathcal{F}$, and in Turing's setting, the diagonal object is a machine whose behavior is designed to disagree with the "self-input" entries of an imagined halting table. In both cases, the force of the argument comes from self-reference, created by diagonalization.

6.1.1 Gödel: A Sentence That Talks About Its Own Provability

Theorem 6.2 (Gödel's First Incompleteness Theorem (Classical Form))**.** *For any consistent formal system $\mathcal{F}$ that is expressive enough to formalize basic arithmetic, there exists a sentence G such that (within $\mathcal{F}$) neither G nor its negation $\neg G$ is provable.*

The diagonal structure becomes easiest to see when we picture an idealized table, in direct analogy with Cantor's array. Enumerate all formulas of the system as $F_1, F_2, F_3, \ldots$ (for instance via Gödel numbering). The entry in row i and column j represents the yes/no question "Does F_i prove F_j?"[183, 185, 186].

	F_1	F_2	F_3	$\cdots$
F_1	$F_1 \vdash F_1?$	$F_1 \vdash F_2?$	$F_1 \vdash F_3?$	$\cdots$
F_2	$F_2 \vdash F_1?$	$F_2 \vdash F_2?$	$F_2 \vdash F_3?$	$\cdots$
F_3	$F_3 \vdash F_1?$	$F_3 \vdash F_2?$	$F_3 \vdash F_3?$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Here, the diagonal entries (i, i) are the "self-indexed" ones where the diagonal represents the question: "Does F_i prove itself?" Gödel's key step is to use arithmetization (Gödel numbering) together with a diagonal/self-reference lemma to build a specific

sentence G that, when decoded, asserts its own unprovability in $\mathcal{F}$[187, 188]. If $\mathcal{F}$ could prove G, it would be proving a sentence that asserts its own unprovability, thereby collapsing consistency. So, under the consistency assumption, $\mathcal{F}$ cannot prove G. But then G is a *true-but-unprovable* statement.

6.1.2 Turing: The Halting Problem as a Diagonal Contradiction

While Gödel's diagonal object lives inside arithmetic, Turing's diagonal object lives inside computation itself. The halting problem asks for a single mechanical procedure that, given a program and an input, always answers correctly whether the program eventually halts. Turing's 1936 result shows that this kind of universal "halts/loops" decider cannot exist[12, 189].

Theorem 6.3 (Turing's Halting Problem (Classical Form))**.** *There is no Turing machine that, for every machine M and input x, correctly decides whether M halts when run on input x.*

To see the diagonal shape, imagine that all possible Turing machines are listed as $M_0, M_1, M_2, \ldots$ and that we record their halting behavior on inputs $0, 1, 2, \ldots$ in an idealized infinite table T, where $T[i, j] = 1$ means "M_i halts on input j" and $T[i, j] = 0$ means "M_i does not halt on input j".

	0	1	2	3	$\cdots$
M_0	$T[0,0]$	$T[0,1]$	$T[0,2]$	$T[0,3]$	$\cdots$
M_1	$T[1,0]$	$T[1,1]$	$T[1,2]$	$T[1,3]$	$\cdots$
M_2	$T[2,0]$	$T[2,1]$	$T[2,2]$	$T[2,3]$	$\cdots$
M_3	$T[3,0]$	$T[3,1]$	$T[3,2]$	$T[3,3]$	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

The diagonal entries $T[i, i]$ correspond to "what M_i does on its own index." Now suppose (for contradiction) that there exists a halting-decider H that can compute every diagonal entry $T[i, i]$ correctly. Using H as a subroutine, we can define a new "diagonal machine" D that, on input i, intentionally does the opposite of what the table says on the diagonal: if H reports that M_i halts on i (so $T[i, i] = 1$), then D loops forever and

if H reports that M_i does not halt on i (so $T[i,i] = 0$), then D halts immediately. In symbols, D is constructed as the diagonal object:

$$D(i) = 1 - T[i,i].$$

Because D is itself a Turing machine, it must appear somewhere in the list, say $D = M_k$. But then we can ask what happens on input k. If $T[k,k] = 1$, the construction forces $D(k) = 0$, meaning D does not halt on k and if $T[k,k] = 0$, the construction forces $D(k) = 1$, meaning D halts on k. Either way, D both halts and does not halt on the same input, which is impossible. This contradiction shows that the original assumption (that a universal halting-decider H exists) cannot be correct.

6.2 Quantum Inspired Constructive Perspective

In Sections 6.1.1 and 6.1.2, we observed the undecidability results, which posited the existence of statements that no algorithmic procedures could ever decide (prove or disprove), irrespective of how much time/memory was made available. However, we also showed how these results relied fundamentally on the diagonal argument, which requires storing and manipulating infinite enumerations for constructing the diagonal object.

In real physical situations and computations, one always deals with finite inputs and algorithmic manipulations that take a finite number of steps. This dilemma is deeper: take, for instance, the case of a *real number* π. It has been proven to be a transcendental number[190], which means it has a non-terminating and non-repeating decimal expansion, and therefore π is just a symbol that could mean different rational approximations coming out of a calculation using any one of the Leibniz series, Machin's formula, Chudnovsky formula, etc., all of which have different rates of convergence but will produce a unique number when a *finite precision* is specified. These rational approximations are the ones used in any computation involving π. The symbol π is a *Platonic Ideal* that is assumed to exist independent of any human calculation [74, 191], whereas the view that only objects that can be constructed through explicit finite procedures is called *Constructivism*[71, 192–195]. However, Constructivism does not impose the requirement to specify a *finite precision*.

According to the Copenhagen interpretation of Quantum Mechanics, one cannot speak of physical properties as having observer-independent values prior to measurement. Taking inspiration from this, we adopt the view that *One cannot meaningfully talk about a mathematical object until it is specified up to a finite precision.* This goes beyond the *Constructive Mathematical* view, which requires the specification of a finite procedure, such as the Leibniz series or Chudnovsky formula for calculating the quantity π, by requiring the additional specification of a *finite precision*, as **any practical calculation requires storing and manipulating finite-precision numbers**. As per this view, therefore, the set of real numbers $\mathbb{R}$, which have infinite precision, cannot be accommodated and only their rational approximations can be. And when we restrict ourselves to this practical regime, Diagonal Arguments break down because the construction of the diagonal object necessarily requires the storing and manipulation of infinite-precision numbers (last sentence in paragraph 6). For example, when restricted to a finite precision of one decimal place, the interval $[1, 5]$ yields a finite, exhaustive set of 41 elements, for which Cantor's diagonal construction cannot generate an additional element. And we prove that these numbers with a terminating decimal expansion, no matter how arbitrarily high the precision might be, are *countable*:

Theorem 6.4 (Finite-Decimal Reals are Enumerable (with Explicit Indexing))**.** *Let* $\mathbb{R}_{\textit{finite-decimal}} = \{r \in \mathbb{R} : r \textit{ has a terminating decimal representation}\}$. *There exists an explicit bijection* $f : \mathbb{R}_{\textit{finite-decimal}} \to \mathbb{N}$*, together with an explicit inverse* f^{-1}*, obtained by (i) a canonical 4-tuple representation and (ii) a closed-form indexing rule[196, 197].*

Proof sketch (Key closed-form formulas): Any $r \in \mathbb{R}_{\text{finite-decimal}}$ is first represented as a *canonical* 4-tuple $(\text{sign}, N_1, N_2, N_3) \in \{-1, +1\} \times \mathbb{N}_0^3$ and trailing zeros in the fractional part are removed, so pure integers satisfy $N_3 = 0 \Rightarrow N_2 = 0$, and 0 is represented uniquely as $(+1, 0, 0, 0)$[196, 197]. We then define the information complexity as:

$$K = N_1 + N_2 + N_3.$$

We enumerate tuples by increasing K. Within a fixed K, we order (N_1, N_2, N_3) lexicographically (with N_3 determined by $N_3 = K - N_1 - N_2$ and the constraint $N_3 > 0$ for fractional numbers) and we place the positive sign before the negative sign.

Counting (closed form): The number of canonical tuples at level K is

$$C(K) = \begin{cases} 1, & K = 0, \\ K(K+1) + 2, & K > 0. \end{cases}$$

Hence the cumulative count of tuples with complexity $< K$ is

$$S(K) := \sum_{j=0}^{K-1} C(j) = \begin{cases} 0, & K = 0, \\ 1, & K = 1, \\ \dfrac{(K-1)K(K+1)}{3} + 2(K-1) + 1, & K > 1. \end{cases}$$

Position within a level (closed form): Let $r \neq 0$ have canonical tuple $(\text{sign}, N_1, N_2, N_3)$ of complexity K.We define the sign offset as $\sigma = 0$ for $+1$, $\sigma = 1$ for -1

The 0-based base position among (N_1, N_2, N_3) combinations at complexity K is given by:

$$b(K; N_1, N_2, N_3) = \begin{cases} \dfrac{K(K+1)}{2}, & N_3 = 0 \ \text{ (integer case } (N_1, N_2, N \\ \displaystyle\sum_{a=0}^{N_1-1} (K-a) + N_2 = N_1 K - \frac{(N_1-1)N_1}{2} + N_2, & N_3 > 0 \ \text{ (fractional case).} \end{cases}$$

And now, the 0-based position *within* the K-level, including sign ordering, is

$$\text{pos}(K, \text{sign}, N_1, N_2, N_3) = 2\, b(K; N_1, N_2, N_3) + \sigma(\text{sign}).$$

Forward map: With $S(K)$ and pos as above,

$$f(r) = \begin{cases} 1, & r = 0, \\ S(K) + \text{pos}(K, \text{sign}, N_1, N_2, N_3) + 1, & r \neq 0. \end{cases}$$

Inverse map (explicit reconstruction recipe): Given $n \in \mathbb{N}$, set $q = n - 1$. If $n = 1$ return 0. Otherwise, choose the unique K such that $S(K) \leq q < S(K+1)$, and set

$p = q - S(K)$ (the 0-based in-level position). Then

$$\text{sign} = \begin{cases} +1, & p \text{ even}, \\ -1, & p \text{ odd}, \end{cases} \qquad b = \left\lfloor \frac{p}{2} \right\rfloor.$$

Recover N_1 by

$$N_1 = \left\lfloor \frac{(2K+1) - \sqrt{(2K+1)^2 - 8b}}{2} \right\rfloor,$$

and then set

$$N_2 = b - \left(N_1 K - \frac{(N_1 - 1)N_1}{2} \right), \qquad N_3 = K - N_1 - N_2.$$

Finally reconstruct the finite-decimal real number by

$$\text{construct}(\text{sign}, N_1, N_2, N_3) = \begin{cases} 0, & (\text{sign}, N_1, N_2, N_3) = (+1, 0, 0, 0), \\ \text{sign} \cdot N_1, & N_2 = 0 \text{ and } N_3 = 0, \\ \text{sign} \cdot \text{Decimal}(N_1.\underbrace{00\ldots0}_{N_2 \text{ zeros}} N_3), & \text{otherwise.} \end{cases}$$

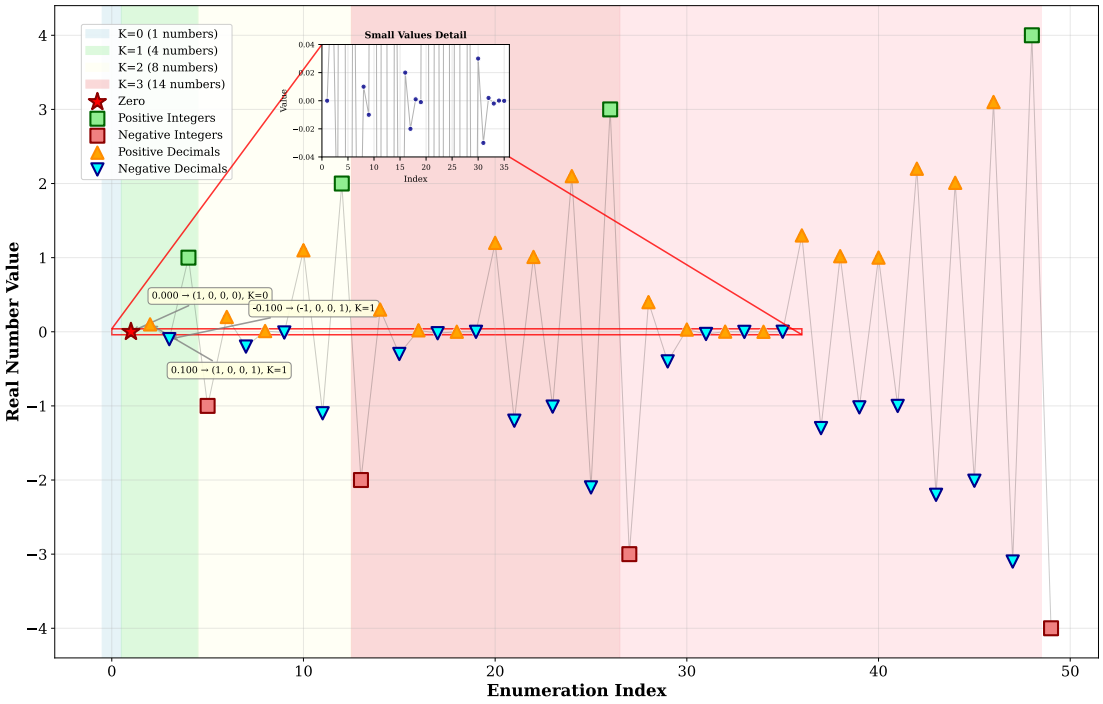


Figure 6.1: Canonical bijection mapping showing the first 49 enumerated finite-decimal real numbers (indices 1–49). Numbers are grouped by increasing "information complexity" and ordered lexicographically within each group[196, 197].

All steps use only a fixed number of arithmetic operations (no enumeration loops), hence $O(1)$ under the usual "unit-cost RAM" model, where each arithmetic operation is assumed to take constant time. In a bit-complexity model the runtime scales with the digit-length/precision of the operands (i.e., with the sizes of K, N_1, N_2, N_3). Full derivations and implementation details are given in [196, 197].

And so we see that $\mathbb{R}_{\text{finite-decimal}}$ is *countable*, as opposed to $\mathbb{R}$, which is *uncountable*, as was shown by Georg Cantor in 1874 [184]. Encouraged by this, we now define a *Finite Resource System* to investigate how the classical undecidability results change when we restrict ourselves to explicit finite resources.

Definition 6.5 (Finite Resource System)**.** A finite resource system $\mathcal{S}(T_{\max}, S_{\max}, P_{\max}, L_{\max})$ is specified by four natural-number bounds: $T_{\max}$ (maximum computational steps), $S_{\max}$ (maximum memory), $P_{\max}$ (maximum numerical precision / total digit positions), and $L_{\max}$ (maximum description length for symbolic objects such as formulas or programs). Any physically realizable mechanized reasoning system must adhere to these bounds[198–200].

Now, under such bounds, the first thing that comes to mind is the loss of *Universality*, which can be formally proven as follows:

Corollary 6.6 (No Universality Inside Fixed Bounds)**.** *There is no universal simulator* $U \in \mathcal{S}(B)$ *such that*

$$\forall (M, x) \in \mathcal{S}(B) \quad U(\langle M \rangle, x) \text{ halts within } T_{\max} \text{ and matches the output of } M(x).$$

under fixed bounds $B = (T_{\max}, S_{\max}, P_{\max}, L_{\max})$.

Proof: Suppose for contradiction that such a U exists but obeys the bound $T_{\max}$. Define a machine M_T (chosen so that $|\langle M_T \rangle| \leq L_{\max}$) that, on any input, performs exactly $T_{\max} + 1$ steps of a simple counter, for example, and then halts. Then for every input x in the bounded domain,

$$\text{time}(M_T(x)) = T_{\max} + 1.$$

If U correctly simulates $M_T(x)$ and matches its output, it must itself execute at least $T_{\max} + 1$ simulated steps or otherwise fail to reproduce the computation, contradicting

the requirement that U halts within $T_{\max}$. Therefore, *no such bounded universal simulator exists.* A similar argument can be made for other bounds like $S_{\max}$ and $L_{\max}$. □

Having lost universality in the finite-resource regime, we now turn to the classical undecidability results and see how they change when constrained to finite resources:

Theorem 6.7 (Adequately Bounded Provability is Decidable)**.** *Let $\mathcal{F}$ denote a formal system with bounds $(L_{\max}, P_{\max})$ and let* $\mathrm{Sent}_{L_{\max},P_{\max}}(\mathcal{F})$ *be the set of all* sentences *in $\mathcal{F}$ whose encodings have length $\leq L_{\max}$ and whose numerical constants lie within the finite precision $\leq P_{\max}$. We can then define:*

$$\mathrm{Prov}_{\leq L_{\max}}(\varphi) \;:\Longleftrightarrow\; \exists \pi \in \Sigma^{\leq L_{\max}} \textit{ such that } \mathrm{Verify}_{\mathcal{F}}(\pi, \varphi) \textit{ accepts.}$$

Then $\mathrm{Prov}_{\leq L_{\max}}$ *is decidable on* $\mathrm{Sent}_{L_{\max},P_{\max}}(\mathcal{F})$ *with an adequate choice of* $(T_{\max}, S_{\max})$.

Proof: Consider $\varphi \in \mathrm{Sent}_{L_{\max},P_{\max}}(\mathcal{F})$ and let Σ be the finite alphabet used to encode formulas and proofs. The candidate *string*-space is

$$\Sigma^{\leq L_{\max}} := \bigcup_{\ell=0}^{L_{\max}} \Sigma^{\ell} \qquad \text{so} \qquad \pi \in \Sigma^{\leq L_{\max}} \iff \pi \in \Sigma^{*} \;\wedge\; |\pi| \leq L_{\max}.$$

So, the quantifier $\exists \pi \in \Sigma^{\leq L_{\max}}$ ranges over *single encoded strings* π of length $\leq L_{\max}$ (not over sequences directly). When π is parsed/decoded it yields either an invalid encoding (immediate reject) or a finite proof-line sequence $\Pi = (\Pi_1, \ldots, \Pi_n)$ of formulas, and $\mathrm{Verify}_{\mathcal{F}}(\pi, \varphi)$ checks Π line-by-line. Define a decider $A(\varphi)$: enumerate all $\pi \in \Sigma^{\leq L_{\max}}$, run $\mathrm{Verify}_{\mathcal{F}}(\pi, \varphi)$, output YES as soon as some π is accepted, and output NO if none are accepted. By construction, $A(\varphi) = \mathrm{YES} \iff \mathrm{Prov}_{\leq L_{\max}}(\varphi)$. Adequate bounds exist since

$$T^{*}(\varphi) := \max_{\pi \in \Sigma^{\leq L_{\max}}} \mathrm{time}(\mathrm{Verify}_{\mathcal{F}}(\pi, \varphi)), \qquad S^{*}(\varphi) := \max_{\pi \in \Sigma^{\leq L_{\max}}} \mathrm{space}(\mathrm{Verify}_{\mathcal{F}}(\pi, \varphi)).$$

and we can set $T_{\max} \geq T^{*}(\varphi)$ and $S_{\max} \geq S^{*}(\varphi)$ to ensure that the verification procedure runs to completion for every candidate proof. □

And so, within a fixed length limit, one can always decide whether a valid proof

exists.

Theorem 6.8 (Adequately Bounded Halting Classification)**.** *For a Machine-input pair* (M, x) *and bounds* $(S_{\max}, L_{\max})$ *with* $|\langle M \rangle| \leq L_{\max}$*, let* $\mathrm{Conf}_{S_{\max}}(M, x)$ *be the set of instantaneous configurations reachable without exceeding* $S_{\max}$*, and let* $N_{S_{\max}}(M, x) := |\mathrm{Conf}_{S_{\max}}(M, x)| < \infty$*. If one chooses an* adequate *time bound* $T_{\max} \geq N_{S_{\max}}(M, x)$*, then running* $M(x)$ *for* $T_{\max}$ *steps decides* HALTS *vs.* LOOPS *(with no* TIMEOUT *outcome).*

Proof: $\mathrm{Conf}_{S_{\max}}(M, x)$ is finite, since an instantaneous configuration is encoded by finite data (state, head position(s), and the contents of the finitely many tape cells that can be visited) and $N_{S_{\max}}(M, x) < \infty$. Now, consider the configurations:

$$\mathrm{cfg}_0, \mathrm{cfg}_1, \ldots, \mathrm{cfg}_{T_{\max}}.$$

If halting occurs within $T_{\max}$ steps, output *HALTS*. Otherwise, since $T_{\max} \geq N_{S_{\max}}(M, x)$, the pigeonhole principle yields $0 \leq i < j \leq T_{\max}$ with $\mathrm{cfg}_i = \mathrm{cfg}_j$. Deterministic evolution implies the system loops forever, therefore output *LOOPS*. □

Therefore, with a fixed memory limit and an appropriately chosen time bound, one can decide whether the run halts or loops.

6.3 Conclusion

We began this chapter with Cantor's proof of the *uncountability* of the set of real numbers ($|\mathbb{R}| > |\mathbb{N}|$). We then explored how the method used to prove this result, namely the Diagonal Argument, was used in other instances by Gödel and Turing to prove the classical *undecidability* results. However, we noticed the critical dependence of the Diagonal Argument on *completed infinities*: the construction of the diagonal object requires the storage of an infinite set and the manipulation of the diagonal element in that enumeration, which takes an infinite number of steps. Motivated by Quantum Mechanics, we restricted ourselves to mathematical objects with explicitly specified finite precision and construction procedures that can be completed in a finite number of steps. We then saw that this is the regime of all practical computations, which require the storage and manipulation of numbers with finite precision, and so we defined

$\mathbb{R}_{\text{finite-decimal}} = \{r \in \mathbb{R} : r \text{ has a terminating decimal representation}\}$ and, by using a novel four-tuple canonical representation, proved that this set is *countable* and that our bijection's forward and inverse formulas are $O(1)$. Inspired by this, we defined a finite resource system with bounds $\mathcal{S}(T_{\max}, S_{\max}, P_{\max}, L_{\max})$ to investigate how the undecidability results would change when constrained to finite resources, and we observed that under finite bounds, it is always possible to decide whether a proof exists for a given statement in the bounded formal system. We also saw that, under adequately chosen finite resource bounds, the halting problem leads to a definite output of halts or loops. Therefore, we can conclude that when restricted to a quantum-inspired finite-precision system, which accommodates only mathematical objects with finite precision and a specified finite construction procedure, diagonal arguments collapse, since any closed interval of numbers with a specified finite precision will be complete and the diagonal method, which inherently requires infinite-precision numbers to manipulate and create a diagonal object, fails to create a number outside of the completed finite enumeration. The set of all numbers with a finite (however arbitrarily large) expansion becomes *countable* ($|\mathbb{R}_{\text{finite-decimal}}| = |\mathbb{N}|$), and the *Classical Undecidability* results are replaced by *Bounded Decidability*. Therefore, in the Quantum-Inspired Constructive perspective, mechanized reasoning methods can decide any proposition within explicit resource bounds and are therefore only fundamentally limited by Resource Constraints.

7

Conclusion

Quantum Mechanics and Computation, the two most consequential developments of the twentieth century, have co-evolved and together ushered in the Fourth Industrial Revolution, whose defining characteristic is the fusion of digital, physical and biological systems. In this setting, Artificial Intelligence has emerged as a practical form of mechanized reasoning, while quantum systems have transitioned from foundational science to deployable technologies. Therefore, this thesis asks the natural yet pertinent question: "What can Artificial Intelligence do for Quantum Systems?" and we answer it in three parts.

Part I applies Machine Learning techniques to quantum systems while leveraging their role as powerful pattern recognizers and universal function approximators to learn properties of higher-dimensional bipartite ququart states from measurement data, which forms the content of Chapter 2. Traditionally, this problem has had two bottlenecks that prevent rapid characterization: the need for large numbers of measurements and the computational burden of iteratively reconstructing the density matrix of the state from experimental data. To address these issues, three variants of Artificial Neural Networks (ANNs), namely the Multi Layer Perceptron, Convolutional Neural Network, and Transformer, have been customized. These neural networks predict the entanglement negativity of the system from incomplete data, achieving, at 100 POVM measurements, a prediction error comparable to what traditional methods require 250 POVM measure-

ments to achieve, while operating 1657× faster, a speedup of three orders of magnitude. This is possible because ANNs learn the structure of these higher-dimensional quantum systems from large quantities of simulation data and approximate the function that maps raw measurement data to entanglement negativity, whereas traditional methods such as the Maximum Likelihood Estimator and Bayesian Estimator iteratively reconstruct the full density matrix, typically starting from an initial state such as the maximally mixed state, without learning from previous experience. Once trained, the forward pass used for prediction is relatively computationally inexpensive, while the iterative nature of the traditional methods keeps them slow, leading to the three-order-of-magnitude difference in speed. This makes neural networks the preferred choice when rapid characterization of high-throughput quantum data is desired.

Having witnessed the effectiveness of ANNs in predicting properties of higher-dimensional quantum systems, we now put them to a stress test in Chapter 3: predicting random binary sequences produced by Random Number Generators (RNGs). While there are algorithmically generated Pseudo Random Number Generators (PRNGs) such as LCRNG, Cryptographically Secure Random Number Generators (CSPRNGs), and Quantum Random Number Generators (QRNGs), true randomness is expected to emerge from quantum processes whose values prior to measurement cannot be predicted. Given the hypothesis that these sequences contain no a priori structure, this task poses a formidable challenge to neural networks. In fact, no obvious inductive bias points to a specific architecture as the optimal solution. We therefore created a framework of 15 different neural network architectures, including networks from the recurrent (Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated Recurrent Unit (GRU), Attention-augmented RNN, Memory-augmented RNN), convolutional (1D CNN, Dilated CNN, ResNet-1D, Binary Pattern CNN, TCN (Temporal Convolutional Network)), and transformer (standard and enhanced) families, along with their hybrids (CNN-LSTM, CNN-Transformer, Hybrid RNG Predictor). When applied to unprocessed PRNGs, CSPRNGs, and QRNGs, the neural networks learn the LCRNG structure of PRNGs almost completely and are able to predict the next byte with 98–99% accuracy when trained on sequence counts ranging from 1K to 1M, with the 1M case providing the best results. CSPRNGs are less predictable than PRNGs, while QRNGs remain the most resistant to neural-network prediction, confirming that they are indeed True Random Number Generators (TRNGs). However, when Toeplitz hashing is

applied to the unprocessed data, all RNG types become strongly and equally resistant to neural-network predictability, aligning with the No-Go theorem. Leveraging the difficulty of the problem and the data from the 15 neural network architectures, several computational studies (efficiency analysis, ANOVA, F statistics, Cohen's d, PCA, consistency analysis, etc.) were performed to assess the performance of the various neural networks on this challenging task. When a weighted overall score was assigned to the models (Improvement Factor: 40%, Efficiency: 30%, Training Speed: 20%, Memory Usage: 10%), Conv1D emerged as the most preferable candidate given its prediction ability and efficiency. However, for maximum predictability, CNN-LSTM and LSTM are better suited, though they come at a higher computational cost. When compared with the results of the traditional NIST SP 800-22 tests, our Multi Architecture Neural Network framework provides complementary insights and can therefore serve as a complementary test for evaluating randomness quality.

Having explored the applications of artificial neural networks (ANNs) to quantum systems in Part I, we now turn to their latest and most consequential embodiment, namely, Large Language Models (LLMs), which have fundamentally reshaped contemporary artificial intelligence. In Part II, we explore the applications of LLMs to quantum systems. In Chapter 4, we evaluate 15 LLMs from 5 major providers on 4 types of problem-solving tasks often encountered in Quantum Mechanics, namely Derivations (D), Creative tasks (C), Non-standard Quantum concepts (N), and Numerical problems (T). The results show a clear tier stratification, with the flagship models outperforming mid-tier and fast models. Derivations (D) remain the easiest tasks for LLMs, which seem to excel at symbolic reasoning, while Numerical problems (T) are the hardest. It was observed that LLMs struggle to choose the right formalism when dealing with numerical problems, as evidenced by the fact that even when code execution is enabled, they show only a modest overall improvement of around 4.4 % at 3 $\times$ the token cost. Three complete runs were performed with deterministic settings (temperature was set to zero, which makes the outputs of the LLMs maximally likely completions under the model distribution), bringing the total to 900 evaluations (15 models $\times$ 20 tasks $\times$ 3 runs) to evaluate reproducibility. We observed that the flagship models are the most consistent in their responses, while fast models vary the most. The insights gained from this chapter were then used as a foundation for the next one. LLMs by themselves are passive neural networks, but when embedded in a cognitive architecture that includes

memory, control, and tool access, they give rise to agentic systems whose capabilities go far beyond a single forward pass of neural inference to include goal-oriented planning, action, and iterative refinement through feedback. In Chapter 5, we present Aṇubuddhi, a Multi-Agent AI system that can design and simulate quantum optics experiments. Although we evaluated 13 experiments spanning three difficulty tiers, we presented one representative result from each tier while providing the full details in [181, 182]. From the three representative results (Hong-Ou-Mandel interference, BB84 Quantum Key Distribution (QKD), and Electromagnetically Induced Transparency in warm Rb vapour), we conclude that while Aṇubuddhi represents a significant leap forward in automated discovery of quantum experiments by eliminating the need for intermediate representations and allowing conversational refinement through natural-language interaction with the user, it can still produce physical parameters that are off by orders of magnitude. This reflects the fact that LLMs do not possess grounding in the physical reality of the laboratory, but instead infer properties from text-based training. However, we also observed that a multi-agent framework consisting of a designer agent, validation loops, a simulation agent, and a cognitive architecture that directs the exchange of information between them significantly improves the quality of the designs compared with using only the designer agent, whose first designs usually contain many errors. An additional layer can be included in future designs to explicitly reference standard data from the web, including atomic transition data from the NIST Atomic Spectra Database and wavelength-dependent refractive indices, dispersion relations, and nonlinear coefficients of optical materials from repositories such as RefractiveIndex.INFO, at an increased token budget to further improve the designs.

In Parts I and II, we saw what AI methods can do for quantum systems, and in the final part of the thesis, we ask "What are the fundamental limitations of AI or mechanized reasoning methods?" We begin this exploration by revisiting the classical undecidability results of the previous century. We examine how both Turing's and Gödel's proofs depend on Cantor's Diagonal Argument and how the diagonal argument itself depends on completed infinities for the construction of the diagonal object. Inspired by Quantum Mechanics, we offer a new perspective in which Mathematical Objects can only be meaningfully discussed when defined up to an explicitly specified finite precision and finite construction procedure, going beyond the regime of Constructive Mathematics, which demands only a finite procedure without requiring a finite preci-

sion to be specified. In this regime, we show that all numbers with a finite decimal expansion, no matter how large the precision may be, are countable by presenting a novel canonical bijection between finite decimal real numbers ($R_{\text{finite-decimal}} = \{r \in R : r$ has a terminating decimal representation$\}$) and natural numbers (N) with $O(1)$ forward and inverse formulas, thereby proving $|R_{\text{finite-decimal}}| = |\text{N}|$. Encouraged by this, we define a Finite Resource Framework for formal systems and computing with explicitly specified bounds on $S(T_{\max}, S_{\max}, P_{\max}, L_{\max})$. Under this Quantum Inspired Constructive framework, we show how the classical undecidability results, which assume unbounded resources and admit infinite construction procedures (since the construction of the diagonal object requires storing and manipulating an infinite sequence of infinite precision numbers), transform into Bounded Decidability. Hence, the Quantum Inspired Constructive perspective posits that mechanized reasoning methods are fundamentally limited only by Resource Constraints and can decide any proposition within explicit resource bounds.

As quantum technologies transition toward broader accessibility and practical deployment, rapid characterization of quantum properties in high-throughput scenarios becomes extremely important, and the Machine Learning methods presented in Part I could be extended to multipartite and other complex scenarios to enable reliable prediction from limited experimental data. The reliability-curves approach (accuracy vs. size of the network), which we presented in Chapter 2 (where the size of the network grew with the available measurement data), can be extended to LLMs to investigate how scaling affects performance on quantum benchmarks specifically.

The next stage in the evolution of agentic systems is autonomous systems[28], an internally motivated, goal-generating system that acts and adapts with minimal external control, in contrast to agentic systems whose objectives are primarily specified and directed externally. This presents a natural future direction that builds on the agentic frameworks developed in Part II. Finally, extending the Quantum Inspired Constructive framework developed in Part III to practical verification workflows in scientific computing could offer a scalable foundation for bounded, reliable decision-making in the era of agentic systems and LLM-centric operating environments.

Publications

1. S. K. Rithvik, R. P. Singh, Shashi Prabhakar, "Machine Learning-Enhanced Entanglement Characterization in Bi-partite Ququart Systems". In: *Research Square* (2025). DOI: https://doi.org/10.21203/rs.3.rs-6486345/v1.

2. S. K. Rithvik, Vardaan Mongia, R. P. Singh, Shashi Prabhakar, "Multi-Architecture Neural Network Evaluation of Quantum and Classical Random Sequence Predictability". *Manuscript submitted for publication* (2025).

3. S. K. Rithvik, "Evaluating Large Language Models on Quantum Mechanics: A Comparative Study Across Diverse Models and Tasks". In: *arXiv preprint* (2025). arXiv: 2602.19006.

4. S. K. Rithvik, "Aṇubuddhi: A Multi-Agent AI System for Designing and Simulating Quantum Optics Experiments". In: *arXiv preprint* (2025). arXiv: 2512.15736.

5. S. K. Rithvik, "A Canonical Bijection Between Finite-Decimal Real Numbers and Natural Numbers with Constant-Time Enumeration Formulas". In: *arXiv preprint* (2025). arXiv: 2508.10750.

6. S. K. Rithvik, "Diagonal Arguments and Infinite Dependencies: Analyzing Classical Undecidability and Universality Under Finite Resource Constraints". In: *Preprints* (2025).
DOI: https://doi.org/10.20944/preprints202510.2040.v1.

References

[1] Max Planck. “Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum”. In: *Verhandlungen der Deutschen Physikalischen Gesellschaft* 2 (1900), pp. 237–245.

[2] Max Planck. “Über das Gesetz der Energieverteilung im Normalspektrum”. In: *Annalen der Physik* 4 (1901), pp. 553–563.

[3] A. Einstein. “Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt”. In: *Annalen der Physik* 17 (1905), pp. 132–148.

[4] W. Heisenberg. “Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen”. In: *Zeitschrift für Physik* 33 (1925), pp. 879–893.

[5] N. Bohr. “On the Constitution of Atoms and Molecules. Part I”. In: *Philosophical Magazine* 26 (1913), pp. 1–25.

[6] E. Schrödinger. “Quantisierung als Eigenwertproblem. Erste Mitteilung”. In: *Annalen der Physik* 79 (1926), pp. 361–376.

[7] M. Born. “Zur Quantenmechanik der Stoßvorgänge”. In: *Zeitschrift für Physik* 37 (1926), pp. 863–867.

[8] Louis de Broglie. “Recherches sur la théorie des quanta (Research on the quantum theory)”. English translation in Louis de Broglie, *Collected Papers on Wave Mechanics*, Blackie and Son, London (1928). PhD thesis. Université de Paris (Sorbonne), 1924.

[9] C. J. Davisson and L. H. Germer. “Diffraction of Electrons by a Crystal of Nickel”. In: *Physical Review* 30.6 (1927), pp. 705–740. DOI: 10.1103/PhysRev.30.705.

[10] M. O. Scully and K. Drühl. “Quantum Eraser: A Proposed Photon Correlation Experiment Concerning Observation and “Delayed Choice” in Quantum Mechanics”. In: *Physical Review A* 25.4 (1982), pp. 2208–2213. DOI: 10.1103/PhysRevA.25.2208.

[11] Yoon-Ho Kim et al. “Delayed ”Choice” Quantum Eraser”. In: *Physical Review Letters* 84.1 (2000), pp. 1–5. DOI: 10.1103/PhysRevLett.84.1.

[12] A. M. Turing. “On Computable Numbers, with an Application to the Entscheidungsproblem”. In: *Proceedings of the London Mathematical Society* s2-42 (1936), pp. 230–265.

[13] John von Neumann. *First Draft of a Report on the EDVAC*. Tech. rep. Contract No. W-670-ORD-4926. Moore School of Electrical Engineering, University of Pennsylvania, 1945.

[14] Gottfried Wilhelm Leibniz. *Dissertatio de arte combinatoria*. Earliest on *characteristica universalis* and *calculus ratiocinator*. Lipsiae: Impressum Johannis Friderici Gleditsch, 1666.

[15] George Boole. *An Investigation of the Laws of Thought, on Which are Founded the Mathematical Theories of Logic and Probabilities*. London: Walton and Maberly, 1854.

[16] Gottlob Frege. *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle: L. Nebert, 1879.

[17] Alfred North Whitehead and Bertrand Russell. *Principia Mathematica*. Vol. 1. Cambridge: Cambridge University Press, 1910.

[18] David Hilbert. "Die Grundlagen der Mathematik". In: *Abhandlungen aus dem Mathematischen Seminar der Hamburgischen Universität* 6 (1928), pp. 65–85.

[19] IBM Quantum. *IBM Quantum Roadmap 2025 Update: Flamingo Processor*. 462-qubit Flamingo; path to fault-tolerant by 2029. 2025.

[20] Jian-Wei Pan et al. "Entanglement-based secure quantum cryptography over 1 120 kilometres". In: *Science* 360.6386 (2018), pp. 285–288. DOI: 10.1126/science.aam6664.

[21] Gordon E. Moore. "Cramming More Components onto Integrated Circuits". In: *Electronics* 38.8 (Apr. 1965), pp. 114–117.

[22] Statista Research Department. *Global number of connected devices by device 2025*. 2025.

[23] United Nations Conference on Trade and Development (UNCTAD). *Digital Economy Report 2024*. 3.6 devices per capita globally; 13 in North America. 2024.

[24] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 4th. Pearson, 2020. ISBN: 978-0-13-461099-3. URL: https://aima.cs.berkeley.edu/.

[25] Martin Davis. *The Universal Computer: The Road from Leibniz to Turing*. 3rd. CRC Press, 2018. ISBN: 9781138502086. DOI: 10.1201/9781315144726. URL: https://www.taylorfrancis.com/books/mono/10.1201/9781315144726/universal-computer-martin-davis.

[26] Wayne Xin Zhao et al. "A Survey of Large Language Models". In: *arXiv preprint arXiv:2303.18223* (2023). URL: https://arxiv.org/abs/2303.18223.

[27] John Wang, Emily Smith, et al. "Agentic AI: A Comprehensive Survey of Architectures, Applications, and Future Directions". In: *arXiv preprint arXiv:2510.25445* (Oct. 2025). PRISMA review of 90 studies (2018–2025). URL: https://arxiv.org/abs/2510.25445.

[28] Yann LeCun. *A Path Towards Autonomous Machine Intelligence*. Version 0.9.2, OpenReview. June 2022. URL: https://openreview.net/pdf?id=BZ5a1r-kVsf.

[29] Monty Newborn. *Automated Theorem Proving: Theory and Practice*. Berlin: Springer-Verlag, 2001. ISBN: 0-387-95075-3.

[30] Bruce G. Buchanan and Edward H. Shortliffe. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project*. The Addison-Wesley Series in Artificial Intelligence. Addison-Wesley, 1984.

[31] Charles L. Forgy. *OPS5 User's Manual*. Tech. rep. CMU-CS-81-135. Carnegie Mellon University, 1981.

[32] Shaoxiong Ji et al. "A Survey on Knowledge Graphs: Representation, Acquisition and Applications". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.2 (2022), pp. 494–514. DOI: 10.1109/TNNLS.2021.3070843. URL: https://arxiv.org/abs/2002.00388.

[33] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[34] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[35] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. "A Learning Algorithm for Boltzmann Machines". In: *Cognitive Science* 9.1 (1985), pp. 147–169. DOI: 10.1207/s15516709cog0901_7. URL: https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog0901_7.

[36] Yulia Sandamirskaya. "Dynamic Neural Fields as a Step Toward Cognitive Neuromorphic Architectures". In: *Frontiers in Neuroscience* 7 (2013), p. 220. DOI: 10.3389/fnins.2013.00220. URL: https://www.frontiersin.org/articles/10.3389/fnins.2013.00220/full.

[37] Frédéric Bouchard et al. "Experimental investigation of high-dimensional quantum key distribution protocols with twisted photons". In: *Quantum* 2 (2018), p. 111. DOI: 10.22331/q-2018-12-04-111. URL: https://doi.org/10.22331/q-2018-12-04-111.

[38] Amin Babazadeh et al. "High-Dimensional Single-Photon Quantum Gates: Concepts and Experiments". In: *Physical Review Letters* 119 (2017), p. 180510. DOI: 10.1103/PhysRevLett.119.180510. URL: https://doi.org/10.1103/PhysRevLett.119.180510.

[39] Zdeněk Hradil. "Quantum-state estimation". In: *Physical Review A* 55.3 (1997), R1561.

[40] Daniel FV James et al. "Measurement of qubits". In: *Physical Review A* 64.5 (2001), p. 052312.

[41] Matteo G. A. Paris and Jaroslav Řeháček, eds. *Quantum State Estimation*. Vol. 649. Lecture Notes in Physics. Springer, 2004. DOI: 10.1007/b98673.

[42] Robin Blume-Kohout. "Optimal, reliable estimation of quantum states". In: *New Journal of Physics* 12.4 (Apr. 2010), p. 043034. ISSN: 1367-2630. DOI: 10.1088/1367-2630/12/4/043034. URL: http://dx.doi.org/10.1088/1367-2630/12/4/043034.

[43] Christopher Granade, Christopher Ferrie, and David G Cory. "Practical Bayesian tomography". In: *New Journal of Physics* 18.3 (2016), p. 033024.

[44] Ferenc Huszár and Neil MT Houlsby. "Adaptive Bayesian quantum tomography". In: *Physical Review A* 85.5 (2012), p. 052120.

[45] SKRithvik. *MLEntChar*. https://github.com/rithvik1122/MLEntChar. ML Tools for entanglement characterization. Apr. 2025.

[46] José Luis Crespo et al. "Assessing the quality of random number generators through neural networks". In: *Machine Learning: Science and Technology* 5.2 (June 2024), p. 025072. DOI: 10.1088/2632-2153/ad56fb. URL: https://doi.org/10.1088/2632-2153/ad56fb.

[47] G. Amigo and B. Dong. "Forecasting Pseudo Random Numbers Using Deep Learning". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 687–692. DOI: 10.1109/QCE52367.2021.00096. URL: https://ieeexplore.ieee.org/document/9660301.

[48] Dmytro Proskurin et al. "Predicting pseudo-random number generator output with sequential analysis". In: *CEUR Workshop Proceedings*. Vol. 3800. 2024. URL: https://ceur-ws.org/Vol-3800/paper5.pdf.

[49] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 5998–6008.

[50] Andrew Rukhin et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Apr. 2010. DOI: 10.6028/NIST.SP.800-22r1a. URL: https://doi.org/10.6028/NIST.SP.800-22r1a.

[51] OpenAI. *GPT-3.5 Turbo Model Card*. OpenAI API Documentation. Reference for GPT-3.5 Turbo (widely available; model persists in docs). 2024.

[52] OpenAI. *GPT-4o Announcement*. OpenAI Blog / Documentation. Launch of GPT-4o multimodal flagship model (mid-2024). 2024.

[53] OpenAI. *GPT-5 Announcement*. OpenAI Blog / Documentation. GPT-5 released August 7, 2025 per model version tracking. 2025.

[54] Anthropic. *Claude 3.5 Haiku System Card*. https://www.anthropic.com/system-cards. Claude 3.5 Haiku model system card (released October 22, 2024). 2024.

[55] Anthropic. *Claude Sonnet 4 Model Card*. Anthropic System Cards. Claude Sonnet 4 (refreshed May 2025). 2025.

[56] Anthropic. *Claude Sonnet 4.5 Model Card*. Anthropic System Cards. Released September/October 2025. 2025.

[57] Google DeepMind. *Gemini 2.0 Flash Model Card*. Google API Changelog. Public API release of Gemini 2.0 Flash (Feb 2025). 2025.

[58] Google DeepMind. *Gemini 2.5 Family Model Card*. Google Developer Blog / API. Gemini 2.5 Flash and Pro described as default in mid-2025. 2025.

[59] Google DeepMind. *Gemini 2.5 Pro Model Card*. Google Developer Blog / API. Part of Gemini 2.5 family release mid-2025. 2025.

[60] Binyuan Hui et al. "Qwen2.5-Coder Technical Report". In: *arXiv preprint arXiv:2409.12186* (2024). Coding-oriented Qwen2.5 series technical report.

[61] Alibaba. *Qwen3 235B Model Card*. Official Qwen Model Release. Qwen3 family including 235B released April 28, 2025. 2025.

[62] *Alibaba Cloud Model Studio: Qwen Large Language Models*. Alibaba Cloud Documentation. Contains official listing and details of Qwen3-Max as a flagship large language model. 2026. URL: https://www.alibabacloud.com/help/en/model-studio/models.

[63] DeepSeek AI. *DeepSeek-V3 Technical Report*. arXiv preprint. Report on DeepSeek V3 series. 2024.

[64] DeepSeek AI. *DeepSeek-R1 Model Technical Report*. arXiv preprint. DeepSeek R1 reasoning model report (inferred release early 2025). 2025.

[65] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte für Mathematik und Physik* 38 (1931), pp. 173–198.

[66] Alan M. Turing. "On computable numbers, with an application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society*. 2nd ser. 42.1 (1937). Received 28 May 1936, pp. 230–265.

[67] Jared Kaplan et al. "Scaling Laws for Neural Language Models". In: *arXiv preprint arXiv:2001.08361* (2020). Demonstrates empirical power-law scaling of language model loss with respect to model parameters, dataset size, and compute, establishing predictable performance improvements from scaling.

[68] Jordan Hoffmann et al. “Training Compute-Optimal Large Language Models”. In: *arXiv preprint arXiv:2203.15556* (2022). Introduces compute-optimal scaling laws (Chinchilla), showing that many large models were undertrained and that optimal performance requires scaling training tokens roughly linearly with model size.

[69] Felipe Maia Polo et al. “Sloth: Scaling Laws for LLM Skills to Predict Multi-Benchmark Performance Across Families”. In: *arXiv preprint arXiv:2412.06540* (2024). Proposes latent skill-based scaling laws that model benchmark performance as emerging skills, enabling cross-family capability prediction beyond parameter-count-based scaling.

[70] Niels Bohr. “Discussion with Einstein”. In: *Albert Einstein: Philosopher–Scientist*. Ed. by Paul Arthur Schilpp. Reprinted in *Atomic Physics and Human Knowledge* (1958), p. 39: “the account of the experimental arrangement and the results of the observations must be expressed in unambiguous language with suitable application of the terminology of classical physics.” New York: Tudor Publishing Company, 1949, pp. 201–241.

[71] Douglas S. Bridges et al. “Constructive Mathematics”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta and Uri Nodelman. Introduces modern constructive mathematics via the BHK interpretation and surveys major schools (Bishop, Martin-Löf, etc.). Metaphysics Research Lab, Stanford University, 2024.

[72] Georg Cantor. “Über eine elementare Frage der Mannigfaltigkeitslehre”. In: *Jahresbericht der Deutschen Mathematiker-Vereinigung* 1 (1891). Introduces Cantor’s diagonal argument for uncountability; English translation in W. B. Ewald (ed.), *From Immanuel Kant to David Hilbert: A Source Book in the Foundations of Mathematics, Vol. 2*, Oxford Univ. Press, 1996, pp. 75–78. URL: https://gdz.sub.uni-goettingen.de/id/PPN241267073_0001?tify=xmlopen%5C%26lang=en%5C%26context=L.

[73] Kurt Gödel. “What is Cantor’s Continuum Problem?” In: *The American Mathematical Monthly* 54.9 (1947), pp. 515–525.

[74] Øystein Linnebo. *Platonism in the Philosophy of Mathematics*. Ed. by Edward N. Zalta and Uri Nodelman. The Stanford Encyclopedia of Philosophy (Fall 2023 Edition). Defines Platonism via three theses: **Existence** (math objects exist), **Abstractness** (non-spatiotemporal), **Independence** (“independent of intelligent agents and their language, thought, and practices”). 2023. URL: https://plato.stanford.edu/archives/fall2023/entries/platonism-mathematics/.

[75] Erwin Schrödinger. “Die gegenwärtige situation in der quantenmechanik”. In: *Naturwissenschaften* 23 (1935), pp. 807–812.

[76] Ryszard Horodecki et al. “Quantum entanglement”. In: *Reviews of Modern Physics* 81.2 (2009), p. 865.

[77] Eric Chitambar and Gilad Gour. “Quantum resource theories”. In: *Reviews of Modern Physics* 91 (2019). Framework of resource theories for entanglement and other quantum features, formalizing “entanglement as a resource.”, p. 025001. DOI: 10.1103/RevModPhys.91.025001.

[78] C. H. Bennett et al. “Teleporting an Unknown Quantum State via Dual Classical and Einstein–Podolsky–Rosen Channels”. In: *Physical Review Letters* 70 (1993). Introduced quantum teleportation using entanglement as the essential resource., pp. 1895–1899. DOI: 10.1103/PhysRevLett.70.1895.

[79] Robert Prevedel et al. “Photonic entanglement as a resource in quantum computation and quantum communication”. In: *Journal of the Optical Society of America B* 24 (2007). Review of experiments using photonic entanglement as a resource for communication, one-way computing, and more., pp. 241–248.

[80] Richard Jozsa and Noah Linden. “On the role of entanglement in quantum-computational speed-up”. In: *Proceedings of the Royal Society A* 459.2036 (2003). Shows that entanglement across many qubits is necessary to achieve quantum speed-ups beyond classical algorithms., pp. 2011–2032. DOI: 10.1098/rspa.2003.1199.

[81] Zixin Huang, Chiara Macchiavello, and Lorenzo Maccone. “Usefulness of entanglement-assisted quantum metrology”. In: *Physical Review A* 94 (2016). Introduces entanglement assistance in quantum metrology protocols, showing how pre-shared entanglement can help measurement precision under noise., p. 012101. DOI: 10.1103/PhysRevA.94.012101.

[82] Jiahao Huang, Min Zhuang, and Chaohong Lee. “Entanglement-enhanced quantum metrology: from standard quantum limit to Heisenberg limit”. In: *Applied Physics Reviews* 11.3 (2024). A recent review on using multi-particle entanglement to boost measurement precision beyond classical limits., p. 031302. DOI: 10.1063/5.0204102.

[83] Rafał Demkowicz-Dobrzański, Jan Kołodyński, and Mădălin Guţă. “Elusive Heisenberg limit in quantum-enhanced metrology”. In: *Nature Communications* 3 (2012). Discusses precision limits in quantum metrology and the role of entanglement in approaching Heisenberg scaling., p. 1063. DOI: 10.1038/ncomms2067.

[84] Daniel Collins et al. “Bell Inequalities for Arbitrarily High-Dimensional Systems”. In: *Phys. Rev. Lett.* 88 (4 Jan. 2002), p. 040404. DOI: 10.1103/PhysRevLett.88.040404. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.040404.

[85] Jonathan Leach et al. “Measuring the Orbital Angular Momentum of a Single Photon”. In: *Phys. Rev. Lett.* 88 (25 June 2002), p. 257901. DOI: 10.1103/PhysRevLett.88.257901. URL: https://link.aps.org/doi/10.1103/PhysRevLett.88.257901.

[86] Richard Bernecker, Baghdasar Baghdasaryan, and Stephan Fritzsche. “High-dimensional maximally entangled photon pairs in parametric down-conversion”. In: *Physical Review A* 110.3 (Sept. 2024). ISSN: 2469-9934. DOI: 10.1103/physreva.110.033718. URL: http://dx.doi.org/10.1103/PhysRevA.110.033718.

[87] Ali Anwar et al. “Selective tuning of Hilbert spaces in states encoded with spatial modes of light”. In: *New Journal of Physics* 22.11 (2020), p. 113020. DOI: 10.1088/1367-2630/abc783. URL: https://doi.org/10.1088/1367-2630/abc783.

[88] Ali Anwar, Shashi Prabhakar, and R. P. Singh. “Size-invariant twisted optical modes for the efficient generation of higher-dimensional quantum states”. In: *Journal of the Optical Society of America B* 38.10 (2021), pp. 2976–2983. DOI: 10.1364/JOSAB.436088. URL: https://doi.org/10.1364/JOSAB.436088.

[89] Alois Mair et al. "Entanglement of the orbital angular momentum states of photons". In: *Nature* 412.6844 (July 2001), pp. 313–316. ISSN: 1476-4687. DOI: 10.1038/35085529. URL: https://doi.org/10.1038/35085529.

[90] Girish S. Agarwal. "Polarization and orbital angularmomentum of quantum fields". In: *Quantum Optics*. Cambridge University Press, 2012, pp. 138–157.

[91] Nijil Lal et al. *Polarization-orbital angular momentum duality assisted entanglement observation for indistinguishable photons*. 2021. arXiv: 2104.11784 [quant-ph]. URL: https://arxiv.org/abs/2104.11784.

[92] "Four-dimensional entanglement distribution over 100 km". In: *Scientific Reports* 8 (2018). High-dimensional time-bin entangled photons transmitted over optical fiber with ¿1 bit secure information capacity, p. 3908. DOI: 10.1038/s41598-017-19078-z.

[93] H. Yu and et al. "Quantum key distribution implemented with d-level time-bin entangled photons". In: *Nature Communications* (2025). Integrated photonic generation of high-dimensional entangled qudits and demonstration of BBM92 protocol over long fiber link. DOI: 10.1038/s41467-024-55345-0.

[94] Juan Yin et al. "Satellite-Based Entanglement Distribution Over 1200 kilometers". In: *arXiv preprint* (2017). Demonstration of entangled photon pair distribution between ground stations thousands of kilometers apart. eprint: 1707.01339.

[95] George Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals, and Systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274.

[96] Giuseppe Carleo et al. "Machine learning and the physical sciences". In: *Reviews of Modern Physics* 91.4 (2019), p. 045002.

[97] Hsin-Yuan Huang et al. "Provably efficient machine learning for quantum many-body problems". In: *Science* 377.6604 (2022), eabk3333.

[98] Giacomo Torlai et al. "Neural-network quantum state tomography". In: *Nature Physics* 14.5 (2018), pp. 447–450.

[99] Tao Xin et al. "Local-measurement-based quantum state tomography via neural networks". In: *npj Quantum Information* 5.1 (2019), pp. 1–8.

[100] Vaneet Rishi, Xiao Wu, and Ish Dhand. "Machine learning reconstruction of quantum entanglement". In: *Science Advances* 8.48 (2022), eadd7131.

[101] Dominik Koutný et al. "Deep learning of quantum entanglement from incomplete measurements". In: *Science Advances* 9.29 (July 2023). ISSN: 2375-2548. DOI: 10.1126/sciadv.add7131. URL: http://dx.doi.org/10.1126/sciadv.add7131.

[102] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[103] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems*. Vol. 25. 2012.

[104] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[105] Jacob Biamonte et al. “Quantum machine learning”. In: *Nature* 549.7671 (2017), pp. 195–202.

[106] Asher Peres. “Separability Criterion for Density Matrices”. In: *Phys. Rev. Lett.* 77 (1996), pp. 1413–1415. DOI: 10.1103/PhysRevLett.77.1413.

[107] Michał Horodecki, Paweł Horodecki, and Ryszard Horodecki. “Separability of Mixed States: Necessary and Sufficient Conditions”. In: *Phys. Lett. A* 223 (1996), pp. 1–8. DOI: 10.1016/S0375-9601(96)00706-2.

[108] Zdeněk Hradil et al. “Quantum-state reconstruction by maximum-likelihood estimation”. In: *Quantum State Estimation*. Ed. by Matteo G. A. Paris and Jaroslav Řeháček. Vol. 649. Lecture Notes in Physics. Springer, 2004, pp. 59–112.

[109] Alexander I Lvovsky. “Iterative maximum-likelihood reconstruction in quantum homodyne tomography”. In: *Journal of Optics B: Quantum and Semiclassical Optics* 6.6 (2004), S556.

[110] Yong Siah Teo et al. “Incomplete quantum state estimation: A comprehensive study”. In: *Physical Review A* 85.4 (2012), p. 042317.

[111] Shuoming Chen et al. “Universal quantum state reconstruction with attention-based quantum transformers”. In: *Communications Physics* 5.1 (2022), p. 257.

[112] Pierre LEcuyer. “Random Number Generation”. In: *Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice*. Ed. by Jerry Banks. John Wiley & Sons, 1998, pp. 55–82. DOI: 10.1002/9780470172436.ch2. URL: https://www.iro.umontreal.ca/~lecuyer/myftp/papers/handstat.pdf.

[113] Haozhe Chai, Qianqian Pan, and Jun Wu. “A survey of random number generator: Approaches, tests, novel applications in block-chain and AI driven industrial networks”. In: *Security and Safety* 4 (Oct. 2025). DOI: 10.1051/sands/2025015.

[114] Mario Stipčević and Bruno J. Shull. “Quantum random number generators and their use in cryptography”. In: *ACM Journal on Emerging Technologies in Computing Systems* 7.2 (2011), Article 11. DOI: 10.1145/2003685.2003692. arXiv: 1103.4381. URL: https://arxiv.org/abs/1103.4381.

[115] Andrew Rukhin et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Tech. rep. NIST Special Publication 800-22 Revision 1a. National Institute of Standards and Technology, 2010.

[116] George Marsaglia. *The Marsaglia Random Number CDROM including the Diehard Battery of Tests of Randomness*. Department of Statistics, Florida State University, 1995.

[117] Chen Li, Wei Zhang, et al. “Deep learning for cryptanalysis of classical ciphers”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 2433–2446.

[118] Minh Truong, Raj Patel, et al. “Machine learning techniques for random number generator analysis”. In: *Computer Security* 88 (2019), pp. 101–115.

[119] Xiongfeng Ma et al. “Postprocessing for quantum random-number generators: entropy evaluation and randomness extraction”. In: *arXiv preprint arXiv:1207.1473* (2012).

[120] Q. Li et al. “Toeplitz-hashing extractor for randomness extraction”. In: *arXiv preprint arXiv:2301.08621* (2023).

[121] Marco Tomamichel et al. “Leftover hashing against quantum side information”. In: *arXiv preprint arXiv:1002.2436* (2010).

[122] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (1997), pp. 1735–1780.

[123] Kyunghyun Cho et al. “Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation”. In: (2014), pp. 1724–1734.

[124] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. “Neural machine translation by jointly learning to align and translate”. In: *arXiv preprint arXiv:1409.0473* (2014).

[125] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. “Effective approaches to attention-based neural machine translation”. In: *arXiv preprint arXiv:1508.04025* (2015).

[126] Kaiming He et al. “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 1026–1034.

[127] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. “An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling”. In: *arXiv preprint arXiv:1803.01271* (2018).

[128] Ashish Vaswani et al. “Attention is All You Need”. In: *Advances in Neural Information Processing Systems* 30 (2017).

[129] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. “Layer Normalization”. In: *arXiv preprint arXiv:1607.06450* (2016).

[130] Dan Hendrycks and Kevin Gimpel. “Gaussian Error Linear Units (GELUs)”. In: *arXiv preprint arXiv:1606.08415* (2016).

[131] Xavier Glorot and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (2010), pp. 249–256.

[132] Diederik P Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[133] Jacob Cohen. *Statistical Power Analysis for the Behavioral Sciences*. 2nd. Lawrence Erlbaum Associates, 1988.

[134] *GPT-3 Training Data Estimates*. Wikipedia: GPT-3. Common Crawl 410B tokens; Books, WebText2, Wikipedia etc.; total hundreds of billions of tokens. 2024. URL: https://en.wikipedia.org/wiki/GPT-3.

[135] *The Pile Dataset for Large Language Models*. Wikipedia: The Pile (dataset). Approx. 886 GB diverse English text corpus for LLM training. 2024. URL: https://en.wikipedia.org/wiki/The_Pile_(dataset).

[136] Jordan Hoffmann et al. “Training Compute-Optimal Large Language Models”. In: *arXiv preprint arXiv:2203.15556* (2022). Describes Chinchilla with 1.4 trillion training tokens for scaling laws. URL: https://arxiv.org/abs/2203.15556.

[137] Yang Liu et al. “Datasets for Large Language Models: A Comprehensive Survey”. In: *arXiv preprint arXiv:2402.18041* (2024). Survey reports total pre-training corpora exceeding 774 TB across many datasets. URL: https://arxiv.org/abs/2402.18041.

[138] Tom Brown et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901.

[139] Karl Cobbe et al. "Training verifiers to solve math word problems". In: *arXiv preprint arXiv:2110.14168* (2021).

[140] Mark Chen et al. "Evaluating large language models trained on code". In: *arXiv preprint arXiv:2107.03374* (2021).

[141] Dan Hendrycks et al. "Measuring mathematical problem solving with the MATH dataset". In: *arXiv preprint arXiv:2103.03874* (2021).

[142] Jacob Austin et al. "Program synthesis with large language models". In: *arXiv preprint arXiv:2108.07732* (2021).

[143] Dan Hendrycks et al. "Measuring massive multitask language understanding". In: *International Conference on Learning Representations*. 2021.

[144] Minzhao Li et al. "QCircuitBench: A benchmark for quantum circuit design". In: *arXiv preprint arXiv:2410.xxxxx* (2024). Preprint.

[145] Zhehui Wang et al. "QuantumBench: Benchmarking LLMs on quantum science". In: *arXiv preprint arXiv:2511.00092* (2024).

[146] Yuxuan Zhao et al. *CMPhysBench: A benchmark for condensed matter physics calculations*. arXiv preprint. 2025.

[147] S. K. Rithvik. *Evaluating Large Language Models on Quantum Mechanics: A Comparative Study Across Diverse Models and Tasks*. 2025. arXiv: 2602.19006 [cs.AI]. URL: https://arxiv.org/abs/2602.19006.

[148] Matthew Renze and Erhan Guven. "The Effect of Sampling Temperature on Problem Solving in Large Language Models". In: *arXiv preprint arXiv:2402.05201* (2024). Systematic analysis of sampling temperature effects on LLM problem solving across models and prompt techniques. URL: https://arxiv.org/abs/2402.05201.

[149] Mario Krenn et al. "Automated Search for New Quantum Experiments". In: *Physical Review Letters* 116.090405 (2016). The "Melvin" algorithm for automated quantum experiment design, pp. 1–6. DOI: 10.1103/PhysRevLett.116.090405. URL: https://doi.org/10.1103/PhysRevLett.116.090405.

[150] Mario Krenn, Xuemei Gu, and Anton Zeilinger. "Quantum Experiments and Graphs: Multiparty States as Coherent Superpositions of Perfect Matchings". In: *Phys. Rev. Lett.* 119 (24 Dec. 2017), p. 240403. DOI: 10.1103/PhysRevLett.119.240403. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.240403.

[151] Carlos Ruiz-Gonzalez et al. "Digital discovery of 100 diverse quantum experiments with PyTheus". In: *Quantum* 7 (2023), p. 1204. DOI: 10.22331/q-2023-12-12-1204.

[152] Rosanna Nichols et al. "Designing Quantum Experiments with a Genetic Algorithm". In: *Quantum Science and Technology* 4.4 (2019). AdaQuantum: Hybrid GA+DNN approach, p. 045012. DOI: 10.1088/2058-9565/ab4d89.

[153] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Standard reference for heuristic search algorithms including A*, admissible and consistent heuristics, and informed search strategies. Prentice Hall, 2010.

[154] Daniil A. Boiko et al. "Autonomous chemical research with large language models". In: *Nature* 624 (2023). Coscientist system, pp. 570–578. DOI: 10.1038/s41586-023-06792-0.

[155] Shuxiang Cao et al. "Agents for self-driving laboratories applied to quantum computing". In: *arXiv preprint arXiv:2412.07978* (2024). k-agents framework.

[156] Sören Arlt, Xuemei Gu, and Mario Krenn. "Towards Autonomous Quantum Physics Research Using LLM Agents with Access to Intelligent Tools". In: *arXiv* (2025). AI-Mandel system, github.com/artificial-scientist-lab/ai-mandel. DOI: 10.48550/arXiv.2511.11752. arXiv: 2511.11752 [quant-ph].

[157] James D. Pickering. *PyOpticalTable: Pain-Free Drawing of Optical Setups*. Online documentation and source code. Python library for creating matplotlib-based optical layout diagrams. Source code available at https://github.com/james-d-pickering/pyopticaltable. 2021. URL: https://jamesdpickering.com/pyopticaltable/.

[158] Patrick Lewis et al. "Retrieval-augmented generation for knowledge-intensive NLP tasks". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 9459–9474.

[159] Chroma. *Chroma: The AI-native open-source embedding database*. https://www.trychroma.com. 2023.

[160] BAAI. *BAAI/bge-base-en-v1.5: English embedding model with 109M parameters*. BGE model documentation. BGE-base-en-v1.5 is an English embedding model in the BGE v1.5 family with approximately 109 million parameters for semantic text embedding tasks. 2023. URL: https://bge-model.com/bge/bge_v1_v1.5.html.

[161] J Robert Johansson, Paul D Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772.

[162] Aman Madaan et al. "Self-Refine: Iterative refinement with self-feedback". In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 46534–46594. URL: https://selfrefine.info.

[163] Xinyun Chen et al. "Teaching Large Language Models to Self-Debug". In: *arXiv preprint arXiv:2304.05128* (2023).

[164] D. C. Burnham and D. L. Weinberg. "Observation of Simultaneity in Parametric Production of Optical Photon Pairs". In: *Physical Review Letters* 25.2 (1970), pp. 84–87. DOI: 10.1103/PhysRevLett.25.84.

[165] P. G. Kwiat et al. "New High-Intensity Source of Polarization-Entangled Photon Pairs". In: *Physical Review Letters* 75.24 (1995), pp. 4337–4341. DOI: 10.1103/PhysRevLett.75.4337.

[166] C. K. Hong, Z. Y. Ou, and L. Mandel. "Measurement of subpicosecond time intervals between two photons by interference". In: *Physical Review Letters* 59.18 (1987), pp. 2044–2046. DOI: 10.1103/PhysRevLett.59.2044.

[167] Ludwig Zehnder. "Ein neuer Interferenzrefraktor". In: *Zeitschrift für Instrumentenkunde* 11 (1891), pp. 275–285.

[168] Ludwig Mach. "Über einen Interferenzrefraktor". In: *Zeitschrift für Instrumentenkunde* 12 (1892), pp. 89–93.

[169] A. A. Michelson and E. W. Morley. "On the Relative Motion of the Earth and the Luminiferous Ether". In: *American Journal of Science* 34 (1887), pp. 333–345.

[170] Charles H. Bennett and Gilles Brassard. "Quantum cryptography: Public key distribution and coin tossing". In: *Proceedings of IEEE International Conference on Computers, Systems and Signal Processing*. Bangalore, India, 1984, pp. 175–179.

[171] Paul G. Kwiat and Harald Weinfurter. "Embedded Bell-state analysis". In: *Physical Review A* 58.4 (1998), R2623–R2626. DOI: 10.1103/PhysRevA.58.R2623.

[172] Dik Bouwmeester et al. "Experimental quantum teleportation". In: *Nature* 390 (1997), pp. 575–579. DOI: 10.1038/37539.

[173] Daniel M. Greenberger et al. "Bell's theorem without inequalities". In: *American Journal of Physics* 58.12 (1990), pp. 1131–1143. DOI: 10.1119/1.16243.

[174] J. D. Franson. "Bell inequality for position and time". In: *Physical Review Letters* 62.19 (1989), pp. 2205–2208. DOI: 10.1103/PhysRevLett.62.2205.

[175] Scott Aaronson and Alex Arkhipov. "The computational complexity of linear optics". In: *Theory of Computing* 9 (2013), pp. 143–252. DOI: 10.4086/toc.2013.v009a004.

[176] Justin B. Spring et al. "Boson Sampling on a Photonic Chip". In: *Science* 339.6121 (2013), pp. 798–801. DOI: 10.1126/science.1231692.

[177] J. Huang and P. Kumar. "Observation of quantum frequency conversion". In: *Physical Review Letters* 68.14 (1992), pp. 2153–2156. DOI: 10.1103/PhysRevLett.68.2153.

[178] S. Zaske et al. "Visible-to-Telecom Quantum Frequency Conversion of Light from a Single Quantum Emitter". In: *Physical Review Letters* 109.14 (2012), p. 147404. DOI: 10.1103/PhysRevLett.109.147404.

[179] S. E. Harris, J. E. Field, and A. Imamoğlu. "Nonlinear optical processes using electromagnetically induced transparency". In: *Physical Review Letters* 64.10 (1990), pp. 1107–1110. DOI: 10.1103/PhysRevLett.64.1107.

[180] M. Fleischhauer, A. Imamoglu, and J. P. Marangos. "Electromagnetically induced transparency: Optics in coherent media". In: *Reviews of Modern Physics* 77.2 (2005), pp. 633–673. DOI: 10.1103/RevModPhys.77.633.

[181] S. K. Rithvik. *Aṇubuddhi: Agentic AI for Quantum Experiment Design*. https://github.com/rithvik1122/Anubuddhi. AI-driven system for conversational quantum experiment design. 2025.

[182] S. K. Rithvik. *Anubuddhi: A Multi-Agent AI System for Designing and Simulating Quantum Optics Experiments*. 2025. arXiv: 2512.15736 [cs.AI]. URL: https://arxiv.org/abs/2512.15736.

[183] Kurt Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte für Mathematik und Physik* 38 (1931), pp. 173–198.

[184] Georg Cantor. "Über eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen". In: *Journal für die reine und angewandte Mathematik* 77 (1874), pp. 258–262.

[185] Raymond M Smullyan. *Gödel's Incompleteness Theorems*. New York: Oxford University Press, 1992.

[186] Torkel Franzén. *Gödel's Theorem: An Incomplete Guide to Its Use and Abuse*. Wellesley, MA: A K Peters, 2005.

[187] George S Boolos. *The Logic of Provability*. Cambridge: Cambridge University Press, 1993.

[188] Jean van Heijenoort. *From Frege to Gödel: A Source Book in Mathematical Logic, 1879-1931*. Editor. Cambridge, MA: Harvard University Press, 1967.

[189] Martin Davis. *Computability and Unsolvability*. New York: McGraw-Hill, 1958.

[190] Ferdinand von Lindemann. "Über die Zahl π". In: *Mathematische Annalen* 20 (1882). Original proof that π is transcendental., pp. 213–225. URL: http://eudml.org/doc/157031.

[191] Kurt Gödel. "What is Cantor's Continuum Problem?" In: *The American Mathematical Monthly* 54.9 (1964). Orig. 1944. Defends mathematical objects as "real" and perceivable independently of construction., pp. 515–525.

[192] Errett Bishop and Douglas S Bridges. *Constructive Analysis*. Berlin: Springer-Verlag, 1985.

[193] Luitzen Egbertus Jan Brouwer. "Intuitionism and Formalism". In: *Bulletin of the American Mathematical Society* 20.2 (1913), pp. 81–96.

[194] Errett Bishop. *Foundations of Constructive Analysis*. New York: McGraw-Hill, 1967.

[195] Douglas Bridges and Fred Richman. *Varieties of Constructive Mathematics*. Cambridge: Cambridge University Press, 1987.

[196] S K Rithvik. "A Canonical Bijection Between Finite-Decimal Real Numbers and Natural Numbers with Constant-Time Enumeration Formulas". In: *arXiv preprint arXiv:2508.10750* (2025). URL: https://arxiv.org/abs/2508.10750.

[197] S K Rithvik. *Canonical Bijection between Finite-Decimal Real Numbers and Natural Numbers: Implementation*. GitHub Repository. 2025. URL: https://github.com/rithvik1122/canonical-bijection-finite-decimals.

[198] Rolf Landauer. "Irreversibility and heat generation in the computing process". In: *IBM Journal of Research and Development* 5.3 (1961), pp. 183–191.

[199] Jacob D Bekenstein. "Universal upper bound on the entropy-to-energy ratio for bounded systems". In: *Physical Review D* 23.2 (1981), pp. 287–298.

[200] M Rey. *Physical Information Theory and Resource-Bounded Computation: Recasting Classical Undecidability Under Physical Constraints*. Preprints. Sept. 2025. DOI: 10.20944/preprints202509.0241.v1. URL: https://doi.org/10.20944/preprints202509.0241.v1.