



**FACULTY
OF MATHEMATICS
AND PHYSICS**
Charles University

MASTER THESIS

Bc. Marko Čechovič

Machine learning models for quantum state tomography

Department of Condensed Matter Physics

Supervisor of the master thesis: RNDr. Pavel Baláž, Ph.D.

Consultant of the master thesis: Kassem Anis Bouali, M.Sc.

Study programme: Computer Science

Study branch: Visual Computing and Game
Development

Prague 2026

I declare that I carried out this master thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In date

Author's signature

I am deeply grateful to RNDr. Pavel Baláž, Ph.D., for his patience, thoughtful guidance, flexibility and support throughout this thesis, and for welcoming me into his grant project. I am also grateful to my consultant and friend Kassem Anis Bouali for his always cheerful vibes and valuable feedback from a computer-graphics perspective. I also want to thank my friend RNDr. Milan Straka, Ph.D. from UFAL, for discussions of the adversarial loss during the early training experiments; my colleagues at Gen for providing access to the company GPU cluster for the initial distributed training experiments; and my family and friends for their support and understanding whenever work on this project cut into our time together. I promise that I will not bring my laptop with me so often now!

Title: Machine learning models for quantum state tomography

Author: Bc. Marko Čechovič

Department: Department of Condensed Matter Physics

Supervisor: RNDr. Pavel Baláž, Ph.D., Department of Condensed Matter Physics

Consultant: Kassem Anis Bouali, M.Sc., Computer Graphics Group, Department of Software and Computer Science Education

Abstract: This thesis asks whether modern machine-learning model families and restoration ideas that have driven progress in computer graphics can also improve quantum state tomography, the reconstruction of an unknown quantum state from repeated measurements. We study four broad families—feed-forward neural networks, conditional generative adversarial networks, iterative generative models from diffusion to flow matching, and transformers—across three practical regimes: incomplete near-clean measurements, complete finite-shot measurements, and measurements that are both incomplete and noisy. Five selected point-estimation pipelines are developed through validation-guided experiments and evaluated on the same synthetic two-qubit state partitions and paired test records. The common formulation treats missing settings as masked data, represents shot-dependent reliability through confidence, and uses a normalised Cholesky map to ensure a physical output. Combined with set-based aggregation and Bures-based training, this formulation reduces the complete high-shot mean Bures distance of a naive multilayer perceptron from 0.079 to 0.022. On this common in-distribution benchmark, RhoTransformer has the lowest mean Bures distance at all four headline operating points, reaching 0.015 with all sixteen settings at 10^5 shots per setting and 0.103 with twelve settings at 10^4 shots per observed setting. A controlled decoder ablation further finds that query self-attention improves complete high-shot reconstruction relative to a larger attention-free control. Together, the selected-pipeline ranking and decoder ablation raise an intriguing possibility: the “attention is all you need” idea may extend to quantum state tomography as well.

Keywords: machine learning, quantum entanglement, quantum state tomography, incomplete noisy measurements, qubits, neural networks, transformer, generative adversarial network, flow matching

Disclosures: Grammar and spelling were checked with Grammarly [1]. GitHub Copilot [2] was used throughout the development of this thesis and research to iterate over some paragraphs for improved clarity, generate initial thesis-outline sections, and assist with plotting figures and debugging experiment implementations. Every output of the AI assistant was thoroughly verified and/or changed by me for its final form, and I am fully responsible for the correctness and final form of this thesis.

Název práce: Modely strojového učení pro kvantovou stavovou tomografii

Autor: Bc. Marko Čechovič

Katedra: Katedra fyziky kondenzovaných látek

Vedoucí diplomové práce: RNDr. Pavel Baláž, Ph.D., Katedra fyziky kondenzovaných látek

Konzultant diplomové práce: Kassem Anis Bouali, M.Sc., Skupina počítačové grafiky, Katedra softwaru a výuky informatiky, Matematicko-fyzikální fakulta Univerzity Karlovy

Abstrakt: Tato práce zkoumá, zda mohou moderní rodiny modelů strojového učení a metody obnovy obrazu, které přispěly k rozvoji počítačové grafiky, zlepšit také tomografii kvantových stavů, tedy rekonstrukci neznámého kvantového stavu ze statistik opakovaných měření. Studujeme čtyři široké rodiny: dopředné neuronové sítě, podmíněné generativní adversariální sítě, iterativní generativní modely od difuzních modelů po metody párování toků (flow matching) a transformery. Porovnání zahrnuje tři praktické režimy: neúplná téměř bezšumová měření, úplná měření s konečným počtem opakování a měření, která jsou současně neúplná a zašuměná. Pět vybraných postupů bodového odhadu vyvíjíme pomocí experimentů řízených výsledky na validační množině. Všechny používají stejné rozdělení syntetických dvouqubitových stavů do trénovací, validační a testovací množiny a jsou hodnoceny na párovaných testovacích záznamech. Společná formulace chápe chybějící nastavení jako maskovaná data, kanálem spolehlivosti kóduje důvěryhodnost měření podle počtu opakování a fyzikálně platný výstup zajišťuje normalizovanou parametrizaci pomocí Choleského rozkladu. Její spojení s množinovou agregací a tréninkem založeným na Buresově vzdálenosti snižuje při úplných měřeních s vysokým počtem opakování střední Buresovu vzdálenost naivního vícevrstvého perceptronu z 0,079 na 0,022. Na tomto společném benchmarku s daty ze stejného rozdělení dosahuje RhoTransformer nejnižší střední Buresovy vzdálenosti ve všech čtyřech hlavních hodnocených bodech: 0,015 při všech šestnácti nastaveních a 10^5 opakováních na nastavení a 0,103 při dvanácti nastaveních a 10^4 opakováních na každé pozorované nastavení. Řízená ablace dekodéru dále ukazuje, že self-attention mezi výstupními dotazy snižuje chybu při úplných měřeních s vysokým počtem opakování oproti většímu dekodéru bez self-attention. Pořadí vybraných postupů a ablace dekodéru tak otevírají zajímavou možnost: myšlenka “attention is all you need” by se mohla uplatnit i v tomografii kvantových stavů.

Klíčová slova: strojové učení, kvantové provázání, kvantová tomografie, neúplná zašuměná měření, qubity, neuronové sítě, transformer, generativní adverzní síť, flow matching

Contents

1	Introduction	4
1.1	A structured restoration perspective	4
1.2	Observation regimes and resource cost	5
1.3	Questions, contributions, and scope	5
1.4	Minimal quantum-state primer	6
1.5	Classical and learning-based reconstruction	7
2	Theory	8
2.1	Quantum state tomography	8
2.2	Reconstruction metrics	21
2.3	Learning-based reconstruction approaches	24
2.4	Summary	43
3	Methodology	44
3.1	Data and the three regimes	44
3.2	Common reconstruction interface	47
3.3	What is held fixed	52
3.4	Compared pipelines and baselines	53
3.5	Model development procedure	54
3.6	Selected reconstruction pipelines	54
3.7	Sequential measurement-setting selection	58
3.8	Evaluation procedure	62
4	Results	65
4.1	Central comparison across reconstruction pipelines	65
4.2	Principal reconstructor: RhoTransformer	68
4.3	Reconstruction from incomplete measurements	69
4.4	Reconstruction under finite-shot noise	74
4.5	Physical properties, computational cost and robustness	75
4.6	Sequential selection within a fixed setting budget	79
4.7	Ablation of the measurement consistency loss	83
4.8	Test-time augmentation by qubit exchange	85
5	Conclusion	87
5.1	Problem and approach	87
5.2	Main findings	87
5.3	Secondary findings and practical meaning	88
5.4	Scope of the conclusions	89
5.5	Future work	89
	Bibliography	91
	List of Figures	97
	Tables and Abbreviations	98

A	Configuration of the reported models and development history	99
A.1	Configuration of the reported models	99
A.2	Development history of the reconstruction models	110

1. Introduction

This thesis studies quantum state tomography as a constrained inverse problem. The hidden object is a quantum state, represented here by a complex 4×4 matrix ρ . Tomography accesses this matrix indirectly: we repeatedly prepare fresh two-qubit systems under the same procedure and model them as independent, nominally identical copies of one unknown state ρ . Sixteen measurement probabilities are estimated from their outcomes. The reconstruction problem is to recover ρ from those probabilities when some estimates are noisy, some are missing, or both.

The complete observation process can be summarised as

$$\rho \xrightarrow{\text{Born-rule measurement}} \mathbf{m} \xrightarrow{\text{finite sampling and masking}} \widetilde{\mathbf{m}} \xrightarrow{\text{reconstruction}} \hat{\rho}. \quad (1.1)$$

The first arrow is the known forward model. The middle arrow describes the available data: finite sampling perturbs an observed probability, while an unrecorded setting removes it. The last arrow is the inverse problem addressed by the methods in this thesis. Its output must be a valid density matrix: Hermitian, positive semidefinite, and of unit trace. Reconstruction therefore respects strict physical constraints. These properties are developed in Section 2.1.1.

1.1 A structured restoration perspective

For the two-qubit James–Kwiat measurement protocol described in Section 2.1.5, the sixteen probabilities form a natural 4×4 tensor-product arrangement. This gives the problem a useful connection to visual computing. Missing settings resemble masked values in an inpainting task, finite-shot fluctuations resemble observation noise, and reconstruction resembles restoration under an output constraint. The analogy follows the tensor-product structure, with grid adjacency serving as a visualisation convention tied to the chosen setting order.

Figure 1.1 shows this correspondence before any detailed quantum formalism is required.

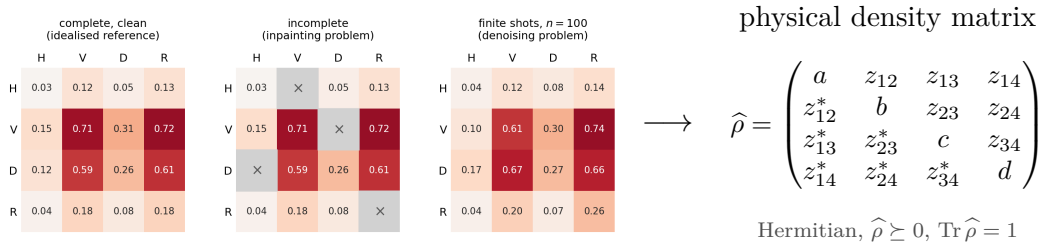


Figure 1.1: The sixteen James–Kwiat probabilities form a 4×4 grid; grey cells mark unobserved settings. Reconstruction maps the incomplete, noisy input to a physical density matrix. The arrangement visualises the tensor-product structure.

The same broad restoration ideas have driven progress in inpainting [3, 4, 5], denoising [6, 7], and masked prediction [8]. Whether their representations, objectives, and model pipelines transfer to a short, globally coupled measurement vector is an empirical question.

1.2 Observation regimes and resource cost

Each probability is estimated by repeating its binary measurement setting on freshly prepared copies of the quantum system and counting the outcomes. The repetitions assigned to that setting are its *shots*; finite sampling produces binomial noise. A setting may also be omitted altogether. The Born-rule and finite-shot measurement model is developed in Section 2.1.4. Separating these effects gives three operating regimes, whose abbreviations are used throughout the experimental chapters:

- *incomplete near-clean (INC) measurements*, which isolate uncertainty caused by missing settings;
- *complete noisy (CN) measurements*, which isolate finite-shot noise; and
- *incomplete noisy (IN) measurements*, where missingness and finite-shot noise occur together.

The first two regimes are controlled diagnostic cases. The third is the most practically relevant one.

Measurements are costly because every shot requires another preparation of the state and another detection event. Long acquisition runs consume apparatus time, use many copies of the state, and are more exposed to equipment calibration drift. A reconstructor that remains accurate with fewer settings or fewer shots could therefore shorten acquisition and improve results. A direct model could also return a density-matrix estimate quickly enough for monitoring, calibration, or adaptive measurement selection. Section 2.3.3 introduces the corresponding closed loop. In that use, a frozen adaptive selector scores the remaining settings after each acquisition and recommends one next setting. A human experimentalist can measure it and return its counts, or a laboratory controller can automate the same loop. Repeating the loop produces a longer schedule; neither the selector nor the reconstructor is retrained online. The synthetic experiments here quantify reconstruction accuracy under controlled setting and shot budgets, providing a basis for subsequent hardware studies of end-to-end acquisition time.

1.3 Questions, contributions, and scope

The benchmark is organised around three questions:

1. What improvement is observed when a naive feed-forward baseline is redesigned around the shared confidence-aware formulation and physical output structure?
2. Which of the five selected implementations achieves the lowest mean Bures distance at the four headline operating points spanning the three observation regimes?
3. What do the reconstructions preserve beyond the primary error metric, and can the same model support measurement selection or inexpensive test-time refinement?

The five headline learned pipelines use the same base state partitions and paired evaluation protocol. Where compatible, they also share the confidence-aware input interface and physical Cholesky output parameterisation (Section 2.1.8); their training samplers, defining objectives, backbones and inference procedures remain pipeline-specific. The study therefore compares complete reconstruction pipelines on one common data benchmark.

Beyond establishing this matched benchmark, the thesis develops RhoTransformer, a confidence-aware physical-output transformer trained through a normalised Cholesky physical map with a regularised density-matrix Bures training loss (Section 2.2.1). It provides controlled benchmark-specific evidence for the contribution of query self-attention (Section 2.3.7); systematically evaluates alternative direct and generative formulations under jointly incomplete and finite-shot measurements; builds on the James–Kwiat normalisation relation to derive the complete single-setting identifiability classification of Claim 1 and experimentally validates the finite-shot implications of the trace-completion identity in Equation (2.21); and develops a closed-loop finite-shot acquisition policy coupled to a frozen secondary-study RhoTransformer.

The resulting hierarchy is stated here only at a high level. The joint redesign of the input representation, set-based aggregation, physical output parameterisation and training objective produces the largest improvement, and the selected direct transformer has the lowest error among the centrally benchmarked point-estimation pipelines. A controlled decoder comparison then finds an additional high-shot benefit from query self-attention. Together, the selected-pipeline ranking and this ablation give the bounded, benchmark-specific sense in which the “attention is all you need” principle [9] may extend to two-qubit QST. Chapter 4 gives the operating-point values, computational costs and complete comparison.

The development path also included a denoising diffusion probabilistic model (DDPM) and a diffusion-transformer denoiser. They are development predecessors, not the selected generative representatives: the final comparison uses a conditional GAN and straight-path flow matching; the selected flow-matching transformer acts as a repeatedly evaluated velocity field rather than as a DDPM denoiser. This terminology is maintained in the following chapters. Sections 2.3.8 and 2.3.9 introduce the corresponding generative-model concepts.

These claims concern synthetic two-qubit states whose training and test splits come from the same generator. The restoration transfer consists of structured masking, confidence-aware denoising and physical output constraints; it is not evidence of two-dimensional image locality in the measurement grid. The grid is encoded through learned setting identities rather than an assumption of natural-image locality. In the matched encoding ablation of Table 3.2, one fixed two-dimensional encoding increased validation error in the incomplete-setting regime and therefore did not support a benefit from that grid geometry.

1.4 Minimal quantum-state primer

Only a small amount of quantum mechanics is needed to follow the reconstruction problem. A qubit is a two-level system whose pure state can be written

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \quad \alpha, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1. \quad (1.2)$$

The squared magnitudes of the complex amplitudes give measurement probabilities in the computational basis. We use *Dirac notation* throughout: a column vector is a ket $|\psi\rangle$, its conjugate transpose is a bra $\langle\psi|$, and $|\psi\rangle\langle\psi|$ is the projector onto the state.

A mixed state is represented by a density matrix

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|, \quad (1.3)$$

where the p_i are classical probabilities. For two qubits this matrix is 4×4 . A projective measurement with projector Π_p has outcome probability

$$m_p = \text{Tr}(\Pi_p \rho), \quad (1.4)$$

the Born rule used in the first arrow of Equation (1.1). The important practical point is that m_p depends linearly on ρ , but only a noisy empirical estimate of m_p is observed. Sections 2.1.1 and 2.1.4 develop this notation, the measurement model, and the physical constraints in detail.

1.5 Classical and learning-based reconstruction

Classical approaches include linear inversion followed by a physicality correction and maximum-likelihood estimation (MLE) [10, 11]. Linear inversion is exact for complete, noise-free probabilities. MLE adds a statistical model for finite-shot noise and is statistically principled when its model is identifiable; a non-identifying incomplete subset admits multiple compatible states, and every MLE estimate requires a new iterative optimisation. Learned reconstructors can instead amortise this inverse map over a state distribution, at the cost of inheriting a prior from their training data. Sections 2.1.7 and 2.1.9 define the classical estimators and their operating domains in detail.

Prior work has studied several parts of this problem. Krawczyk et al. [12] focus on incomplete measurements, while Macarone Palmieri et al. [13] use an attention-based corrector for complete noisy measurements. Danaci et al. [14] combine imputation, denoising, and estimation in a stacked pipeline, and Ma et al. [15] train across limited-copy, incomplete, and noisy settings. Li et al. [16] learn an entanglement-quasiprobability representation from incomplete noisy measurements and validate it on a photonic experiment.

Transformer tomography has also been explored directly. Cha et al. [17] introduce attention-based tomography; Ma et al. [18] query measurement operators in a quantum-aware transformer and validate it on IBM hardware; their later transformer autoencoder learns latent representations of imperfect data [19]. Zhong et al. [20] cast state reconstruction as language modelling over measurement outcomes. These works provide the closest reconstruction and architectural precedents. The present study adds the matched comparison, controlled ablations, single-setting identifiability analysis and closed-loop finite-shot acquisition described in Section 1.3.

2. Theory

This chapter provides the technical background used in the experiments. It first introduces two-qubit quantum state tomography (QST), classical reconstruction and the evaluation metrics, then develops the machine-learning concepts behind four broad reconstruction families: direct feed-forward and transformer regression, adversarial reconstruction, and iterative generative reconstruction through diffusion or flow matching. These labels are not four mutually exclusive backbones: MLPs and transformers describe network architectures, whereas adversarial training and flow matching describe objectives and inference procedures that may themselves use either backbone. Throughout, $n = 2$ qubits gives joint Hilbert-space dimension $d = 2^n = 4$. We use the Dirac notation introduced in Section 1.4.

Reader guidance. Readers familiar with quantum tomography may skim the derivations in Sections 2.1.1–2.1.9; readers familiar with standard deep-learning components may skim Sections 2.3.2 and 2.3.4. The QST-specific architectures in the remaining learning sections connect directly to the experimental chapters.

Essential facts. A two-qubit state is a 4×4 density matrix constrained to be Hermitian, positive semidefinite and of unit trace. The Born rule maps this hidden state linearly to measurement probabilities, while estimating each observed probability from finitely many shots makes it noisy. Whether a missing setting destroys uniqueness depends on which setting is absent. The sections below derive that distinction and show how classical and learned estimators respect the physical state constraints.

2.1 Quantum state tomography

2.1.1 Physical density matrices and degrees of freedom

A two-qubit quantum state is described by a 4×4 density matrix $\rho \in \mathbb{C}^{4 \times 4}$ subject to three physical constraints [21]:

$$\rho = \rho^\dagger, \quad \text{Tr}(\rho) = 1, \quad \rho \succeq 0, \quad (2.1)$$

i.e. ρ is Hermitian ($\rho = \rho^\dagger$, where the *Hermitian conjugate* $\rho^\dagger$ is obtained by transposing ρ and taking the complex conjugate of each entry), has unit trace and is positive semidefinite (PSD). We write I_d for the $d \times d$ identity and retain the subscript whenever the matrix dimension might otherwise be unclear.

These constraints reduce the apparent number of variables step by step. A general complex 4×4 matrix contains 16 complex, or 32 real, values. Hermiticity leaves 4 real diagonal entries and 6 independent complex off-diagonal entries, for $4 + 2 \cdot 6 = 16$ real parameters. Unit trace removes one further parameter, leaving 15. Positivity is an inequality that restricts the allowed points to a fifteen-dimensional convex subset of $\mathbb{R}^{15}$: the set of physically valid two-qubit states.

The extreme points of this set are the *pure states*, the rank-one density matrices $\rho = |\psi\rangle\langle\psi|$ with one eigenvalue equal to 1 and the rest 0. Every other valid state is a *mixed state*, a convex combination $\rho = \sum_k p_k |\psi_k\rangle\langle\psi_k|$ of pure states.

The scalar *purity*

$$\mathcal{P}(\rho) = \text{Tr}(\rho^2) = \sum_{i=1}^4 \lambda_i^2, \quad \frac{1}{4} \leq \mathcal{P}(\rho) \leq 1, \quad (2.2)$$

summarises how concentrated the eigenvalue spectrum $\{\lambda_i\}$ is. The range follows because the four eigenvalues are non-negative and sum to one. A pure state has spectrum $(1, 0, 0, 0)$ and $\mathcal{P} = 1$; the maximally mixed state has four equal eigenvalues and the minimum purity $\mathcal{P} = 1/4$. Intermediate values describe mixed states, although purity alone does not specify the full spectrum.

The words *interior*, *boundary* and *centre* describe the geometry of the set of valid density matrices; they do not refer to a position in the laboratory. All trace-one Hermitian matrices form the surrounding fifteen-dimensional coordinate space, and positivity carves out the bounded convex region of physical states inside it. A full-rank density matrix has four strictly positive eigenvalues. It is an interior point because sufficiently small Hermitian, trace-preserving perturbations leave all four eigenvalues non-negative and therefore remain physical. A rank-deficient state has at least one zero eigenvalue. It lies on the boundary because every neighbourhood around it also contains matrices for which that zero eigenvalue has been perturbed below zero, making them non-physical. The *maximally mixed state* $\rho = I_4/4$, whose four eigenvalues are all $1/4$, is the natural centre: it weights all four basis directions equally, is unchanged by every unitary change of basis, and has the largest possible minimum eigenvalue, $1/4$. For comparison, the one-qubit state space is the familiar three-dimensional Bloch ball: $I_2/2$ is its centre, full-rank states are inside the ball and pure states form its surface. The two-qubit state space has a more complicated fifteen-dimensional shape, but the same geometric terminology applies. This convex-geometric view of density-operator state spaces is developed in detail by Bengtsson and Życzkowski [22].

2.1.2 Two-qubit structure and entanglement

Each qubit has a two-dimensional Hilbert space, spanned for example by $|0\rangle$ and $|1\rangle$. The joint basis contains every ordered pair, $|00\rangle, |01\rangle, |10\rangle, |11\rangle$, so the dimensions multiply:

$$\dim(\mathcal{H}_{AB}) = \dim(\mathcal{H}_A) \dim(\mathcal{H}_B) = 2 \cdot 2 = 4.$$

Thus the two factors of 2 are the dimensions of the two individual qubit spaces; their product is why a two-qubit state vector has four complex components and its density matrix is 4×4 . Formally, the two-qubit Hilbert space is the *tensor product* $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$. We use the symbol $\otimes$ throughout for the tensor (Kronecker) product, both of vectors and of operators. For two-component column vectors $\mathbf{u}, \mathbf{v} \in \mathbb{C}^2$ it is the four-component vector

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \otimes \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} u_1 v_1 \\ u_1 v_2 \\ u_2 v_1 \\ u_2 v_2 \end{pmatrix}, \quad (2.3)$$

and for 2×2 operators A and B acting on the respective single-qubit factors, $A \otimes B$ is the 4×4 block matrix

$$A \otimes B = \begin{pmatrix} a_{11}B & a_{12}B \\ a_{21}B & a_{22}B \end{pmatrix} \quad (2.4)$$

that acts on $\mathcal{H}_{AB}$ by $(A \otimes B)(\mathbf{u} \otimes \mathbf{v}) = (A\mathbf{u}) \otimes (B\mathbf{v})$. The same notation extends to n qubits: $\mathcal{H}^{\otimes n}$ has dimension $d = 2^n$, and operators on it are $d \times d$ matrices.

Two-qubit product states $|\psi\rangle_A \otimes |\phi\rangle_B$ are abbreviated $|\psi\rangle|\phi\rangle$ or, in the computational basis, $|00\rangle, |01\rangle, |10\rangle, |11\rangle$. Pure states that cannot be written in this product form are *entangled*: the two qubits then have correlations that cannot be described one qubit at a time. For mixed states the definition is slightly wider. A mixed state is *separable* when it can be written as a convex combination of product states, $\rho = \sum_k p_k \rho_k^A \otimes \rho_k^B$, and *entangled* otherwise. Thus a mixed state may fail to be a product state and yet remain unentangled. The canonical maximally entangled examples are the four *Bell states*

$$|\Phi^\pm\rangle = \frac{1}{\sqrt{2}}(|00\rangle \pm |11\rangle), \quad |\Psi^\pm\rangle = \frac{1}{\sqrt{2}}(|01\rangle \pm |10\rangle), \quad (2.5)$$

which form an orthonormal basis of the two-qubit space. Entangled states are an important use case for quantum state tomography; Nielsen and Chuang give the standard tensor-product and separability background used here [21].

2.1.3 The Pauli representation

To reconstruct a state, we need real coordinates for its unknown entries. Single-qubit operators have a natural basis in the Pauli matrices, and the same structure carries over to multiple qubits through the tensor product [21]. Let

$$\begin{aligned} \sigma_0 = I_2 &= \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \sigma_1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \\ \sigma_2 = Y &= \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \end{aligned} \quad (2.6)$$

be the three standard Pauli matrices X, Y, Z together with the 2×2 identity I_2 , which we include as a fourth element σ_0 so that the set can expand *any* 2×2 Hermitian matrix and not only the traceless ones (the three Pauli matrices are themselves traceless and so, on their own, span only the traceless Hermitian matrices). The four operators $\{\sigma_0, \sigma_1, \sigma_2, \sigma_3\}$ are Hermitian and orthogonal under the Hilbert–Schmidt inner product $\langle A, B \rangle = \text{Tr}(A^\dagger B)$, with $\text{Tr}(\sigma_i \sigma_j) = 2\delta_{ij}$, where δ_{ij} is one for $i = j$ and zero otherwise. Thus each basis element has squared norm 2 and distinct elements have zero overlap.

To see explicitly why this real vector space is four-dimensional, write a general 2×2 Hermitian matrix as

$$A = \begin{pmatrix} a & c + id \\ c - id & b \end{pmatrix}, \quad a, b, c, d \in \mathbb{R}.$$

The two diagonal entries contribute the two real parameters a and b . The upper off-diagonal entry contributes the two real parameters c and d , while Hermiticity fixes the lower entry to its complex conjugate and introduces no additional

parameter. The count $2 + 2 = 4$ therefore refers to real matrix parameters, not to the number of qubits. Four nonzero mutually orthogonal vectors in this four-dimensional space are necessarily linearly independent; the $\{\sigma_i\}$ therefore form an orthogonal basis of that space.

For two qubits, taking tensor products of single-qubit Pauli operators yields the sixteen two-qubit Pauli operators

$$\gamma_\mu = \sigma_{i_1} \otimes \sigma_{i_2}, \quad \mu = (i_1, i_2), \quad i_1, i_2 \in \{0, 1, 2, 3\}. \quad (2.7)$$

Here μ is a collective index for the pair (i_1, i_2) . Integer subscripts denote a fixed enumeration of the sixteen pairs, with $\mu = 0$ corresponding to $(0, 0)$. The trace of a tensor product factorises, $\text{Tr}(A \otimes B) = \text{Tr}(A) \text{Tr}(B)$, and $(\sigma_i \otimes \sigma_j)(\sigma_k \otimes \sigma_l) = (\sigma_i \sigma_k) \otimes (\sigma_j \sigma_l)$, so the single-qubit orthogonality relation lifts to

$$\text{Tr}(\gamma_\mu \gamma_\nu) = \text{Tr}(\sigma_{i_1} \sigma_{j_1}) \cdot \text{Tr}(\sigma_{i_2} \sigma_{j_2}) = (2 \delta_{i_1 j_1})(2 \delta_{i_2 j_2}) = 4 \delta_{\mu\nu}, \quad (2.8)$$

with $4 = d = 2^n$. The sixteen γ_μ are thus an orthogonal (but *not* orthonormal) basis of the sixteen-dimensional real vector space of 4×4 Hermitian matrices.

Because the basis is complete, any Hermitian ρ has a unique expansion $\rho = \sum_\mu c_\mu \gamma_\mu$. Taking the Hilbert–Schmidt inner product with γ_ν and using Equation (2.8) gives $c_\nu = \frac{1}{4} \text{Tr}(\gamma_\nu \rho)$.

Pauli coefficients. Quantum tomography conventionally moves the factor $1/4$ outside the expansion and defines the dimensionless quantities

$$r_\mu = \text{Tr}(\gamma_\mu \rho), \quad \mu = 0, \dots, 15. \quad (2.9)$$

Every σ_i is Hermitian and satisfies $\sigma_i^2 = I_2$; hence γ_μ is Hermitian and $\gamma_\mu^2 = I_4$, so its eigenvalues lie in $\{-1, +1\}$. Equation (2.9) is therefore the expectation value of that Pauli product in state ρ . An expectation value is a probability-weighted average of the observable’s eigenvalues, so r_μ is real and satisfies $-1 \leq r_\mu \leq 1$. For example,

$$r_{(1,0)} = \text{Tr}[(X \otimes I_2)\rho] = \langle X \otimes I_2 \rangle_\rho, \quad r_{(3,3)} = \langle Z \otimes Z \rangle_\rho.$$

Using the fixed enumeration introduced after Equation (2.7), we collect them into the ordered vector $\mathbf{r} = (r_0, \dots, r_{15})^\top \in \mathbb{R}^{16}$. The density matrix is recovered from this vector as

$$\rho = \frac{1}{4} \sum_{\mu=0}^{15} r_\mu \gamma_\mu. \quad (2.10)$$

Thus the literal multiplier of γ_μ in ρ is $r_\mu/4$; by convention, however, this thesis calls the expectation value r_μ itself a *Pauli (expansion) coefficient*. The prefactor $1/4 = 1/d$ is the inverse of the squared Pauli norm in Equation (2.8). For an n -qubit system the same derivation gives $\rho = 2^{-n} \sum_\mu r_\mu \gamma_\mu$.

Unit trace fixes the identity coefficient: $r_{(0,0)} = \text{Tr}(I_4 \rho) = 1$, denoted $r_0 = 1$ in the integer enumeration. The other fifteen entries $\{r_\mu\}_{\mu \neq 0}$ are the free Pauli coefficients.¹ They obey no further linear equality, although the PSD condition

¹For one qubit, the analogous coefficients (r_1, r_2, r_3) form the Bloch vector in the unit ball, with pure states on the surface and the maximally mixed state at the centre. The fifteen free two-qubit coefficients are therefore also called generalised Bloch coordinates, although their physical region is a more complicated convex subset of $\mathbb{R}^{15}$.

still couples them through nontrivial joint inequalities. This makes them convenient learning targets, but an arbitrary vector in $[-1, 1]^{15}$ need not represent a physical state.

For numerical work it is also useful to flatten ρ directly into a vector of sixteen real numbers exploiting Hermiticity: the four diagonal entries are real, and the six upper-triangle complex entries contribute six real and six imaginary parts. We refer to this representation as $\rho_{\text{flat}} \in \mathbb{R}^{16}$ in the experimental chapters. It is a sixteen-component storage representation, not a sixteen-degree-of-freedom parameterisation: its diagonal components still obey the unit-trace constraint.

2.1.4 Projective measurements and the Born rule

The Pauli coefficients defined in Equation (2.9) describe the hidden state; measurements provide linear observations of those coefficients. In this thesis, one measurement *setting* is a binary yes–no test represented by a rank-one projector $\Pi_p = |\psi_p\rangle\langle\psi_p|$ and its complement $I_4 - \Pi_p$. The Born rule gives the probability of the projected (“yes”) outcome [21],

$$m_p = \text{Tr}(\Pi_p \rho), \quad (2.11)$$

and the complementary outcome has probability $1 - m_p$.

An orthonormal basis can be measured as one multi-outcome experiment, whose mutually exclusive probabilities sum to one. The sixteen inputs used here have a different organisation: each projector is tested separately on a fresh ensemble of nominally identical state preparations. Consequently, the sixteen values m_p are probabilities from sixteen binary experiments, and their sum is generally unconstrained.

Acquiring setting p with n_p *shots* (one shot on each freshly prepared copy; we reserve the bare symbol n for the number of qubits) and counting k_p projected outcomes gives the empirical frequency $\widehat{m}_p = k_p/n_p$. Because this is a binary experiment,

$$k_p \sim \text{Binomial}(n_p, m_p), \quad \widehat{m}_p = \frac{k_p}{n_p} \xrightarrow{n_p \rightarrow \infty} m_p, \quad (2.12)$$

with standard deviation $\sigma_p = \sqrt{m_p(1 - m_p)/n_p}$. Its $\mathcal{O}(n_p^{-1/2})$ decay is slow: reducing the standard deviation by a factor of ten requires one hundred times as many state preparations. This finite-sampling error, traditionally called *shot noise*, is the principal noise source studied in the experimental chapters.

2.1.5 The James–Kwiat projector set

A single projector probability does not determine ρ . A scheme is *informationally complete* when its recorded probabilities uniquely determine every state in the class under consideration. A $d \times d$ density matrix has $d^2 - 1$ real degrees of freedom, so at least that many independent real values are needed. Each separately implemented binary setting can supply one such value; in principle, therefore, fifteen well-chosen settings can identify a two-qubit state.

We use the Cartesian tensor-product convention adopted by Krawczyk et al. [12], based on the polarisation tomography scheme of James, Kwiat, Munro and White [11]. Its single-qubit states are

$$|H\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |V\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |D\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad |R\rangle = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ -i \end{pmatrix}. \quad (2.13)$$

Here H , V , D and R denote horizontal, vertical, $+45^\circ$ diagonal and right-circular polarisation. All four vectors are normalised, but they are not four mutually orthogonal outcomes. Substituting the vectors from Equation (2.13) gives

$$\begin{aligned} \langle H|V\rangle &= 0, \\ \langle H|D\rangle &= \langle V|D\rangle = \langle H|R\rangle = \frac{1}{\sqrt{2}}, \\ \langle V|R\rangle &= -\frac{i}{\sqrt{2}}, \quad \langle D|R\rangle = \frac{1-i}{2}. \end{aligned} \quad (2.14)$$

Every distinct pair except H, V therefore has overlap magnitude $1/\sqrt{2}$. Because the single-qubit Hilbert space is two-dimensional, two normalised orthogonal states form a complete basis; H, V is the only such pair contained in this four-state set. Non-orthogonal pairs such as H, D also span the two-dimensional space, but their projectors are not mutually exclusive outcomes and do not sum to the identity. The orthogonal partners of D and R would instead be

$$|A\rangle = \frac{|H\rangle - |V\rangle}{\sqrt{2}}, \quad |L\rangle = \frac{|H\rangle + i|V\rangle}{\sqrt{2}},$$

the anti-diagonal and left-circular states. They satisfy $\langle D|A\rangle = \langle R|L\rangle = 0$, but neither A nor L belongs to the four-state set used here. The rank-one projectors of the selected states are

$$\begin{aligned} \mu_H &= |H\rangle\langle H| = \frac{1}{2}(I_2 + Z), & \mu_V &= |V\rangle\langle V| = \frac{1}{2}(I_2 - Z), \\ \mu_D &= |D\rangle\langle D| = \frac{1}{2}(I_2 + X), & \mu_R &= |R\rangle\langle R| = \frac{1}{2}(I_2 - Y). \end{aligned} \quad (2.15)$$

Thus the set uses both Z eigenstates but only one eigenstate for each of X and Y . Conversely,

$$\begin{aligned} I_2 &= \mu_H + \mu_V, & Z &= \mu_H - \mu_V, \\ X &= 2\mu_D - \mu_H - \mu_V, & Y &= \mu_H + \mu_V - 2\mu_R. \end{aligned} \quad (2.16)$$

The four μ_a therefore span the same four-dimensional real operator space as I, X, Y, Z and form a basis of the Hermitian 2×2 matrices. Their sixteen tensor products

$$\Pi_{ab} = \mu_a \otimes \mu_b, \quad a, b \in \{H, V, D, R\}, \quad (2.17)$$

consequently form a basis of the Hermitian 4×4 matrices. They constitute a tomographic basis implemented as sixteen separate binary settings. Strictly, Π_{ab} denotes the projector, m_{ab} its exact Born probability and $\widehat{m}_{ab}$ an observed frequency. Where no confusion results, the experimental chapters use *measurement*, *setting*, *slot* and *projector* interchangeably for the corresponding grid element.

The two-letter label ab places polarisation a on the first qubit and b on the second. Thus VH means vertical on the first qubit and horizontal on the second,

while DD means diagonal on both. Ordering both axes as H, V, D, R , with the first qubit selecting the row and the second the column, gives

	H	V	D	R
H	HH	HV	HD	HR
V	VH	VV	VD	VR
D	DH	DV	DD	DR
R	RH	RV	RD	RR

This 4×4 table is a layout of measurement settings, not the row-column layout of the density matrix. A polarisation qubit has a two-dimensional Hilbert space: H and V form an orthonormal basis, while D and R are superposition states used as additional probes. When the H/V basis is used as the computational basis, the rows and columns of the two-qubit density matrix are labelled HH , HV , VH and VV ; D and R do not add further Hilbert-space dimensions.

From a state vector to its density matrix. The following construction explains what the matrix entries mean; tomography works in the reverse direction, using measured probabilities to infer an otherwise unknown ρ . Identifying $|H\rangle \equiv |0\rangle$ and $|V\rangle \equiv |1\rangle$, order the two-qubit H/V basis as $|HH\rangle, |HV\rangle, |VH\rangle, |VV\rangle$. A pure state then has four complex amplitudes,

$$|\psi\rangle = \alpha |HH\rangle + \beta |HV\rangle + \gamma |VH\rangle + \delta |VV\rangle.$$

Its coordinate vector and normalisation condition are

$$\begin{pmatrix} \alpha & \beta & \gamma & \delta \end{pmatrix}^T, \quad |\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1.$$

Its density matrix is the outer product $\rho = |\psi\rangle \langle\psi|$:

$$\rho = \begin{pmatrix} |\alpha|^2 & \alpha\beta^* & \alpha\gamma^* & \alpha\delta^* \\ \beta\alpha^* & |\beta|^2 & \beta\gamma^* & \beta\delta^* \\ \gamma\alpha^* & \gamma\beta^* & |\gamma|^2 & \gamma\delta^* \\ \delta\alpha^* & \delta\beta^* & \delta\gamma^* & |\delta|^2 \end{pmatrix}. \quad (2.18)$$

A mixed state is a weighted sum of such matrices, $\rho = \sum_j w_j |\psi_j\rangle \langle\psi_j|$, where $w_j \geq 0$ and $\sum_j w_j = 1$, with the rows and columns kept in the same four-state order. The Born rule for an H/V projector simply selects the corresponding diagonal entry:

$$\begin{aligned} m_{HH} &= \langle HH | \rho | HH \rangle = \rho_{HH,HH}, & m_{HV} &= \langle HV | \rho | HV \rangle = \rho_{HV,HV}, \\ m_{VH} &= \langle VH | \rho | VH \rangle = \rho_{VH,VH}, & m_{VV} &= \langle VV | \rho | VV \rangle = \rho_{VV,VV}. \end{aligned} \quad (2.19)$$

These four diagonal entries sum to $\text{Tr}(\rho) = 1$. They do not, however, reveal the complex off-diagonal entries, called coherences. A D setting illustrates how the other probes reach them:

$$\begin{aligned} |DD\rangle &= \frac{1}{2} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T, \\ m_{DD} &= \langle DD | \rho | DD \rangle = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix} \rho \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T. \end{aligned}$$

Thus m_{DD} combines diagonal and off-diagonal entries rather than selecting one matrix coordinate. Settings containing R similarly introduce complex phases and

expose imaginary coherences. This is why D and R appear in the measurement grid even though they are not additional rows or columns of ρ . The general reconstruction from all measured probabilities is derived in Section 2.1.7.

In the row-major measurement vector, HH , HV , VH and VV occupy the upper-left 2×2 block: one-based positions 1, 2, 5, 6, or zero-based array indices 0, 1, 4, 5. The phrase “upper-left” is ordering-dependent; the physically invariant description is the four tensor products of H and V .

2.1.6 Single-setting omissions and unit trace

Sections 2.1.1 and 2.1.3 showed that a trace-one two-qubit density matrix has fifteen independent real parameters. This count is only a lower bound on the required data: fifteen probabilities suffice only when their fifteen equations determine all fifteen free directions. The full grid contains sixteen probabilities, so the precise question is whether the known trace can replace whichever one is omitted.

The trace relation. Equation (2.19) already gives the simplest view. The probabilities $m_{HH}, m_{HV}, m_{VH}, m_{VV}$ are the four diagonal entries of ρ in the H/V basis, and diagonal entries sum to the trace. The operator form explains why this relation involves these four settings. Among H, V, D, R , only H, V form a complete orthonormal pair; the partners A of D and L of R are absent from the set. Therefore $\mu_H + \mu_V = I_2$ is the only single-qubit resolution of the identity fully available here. Applying it to both qubits gives

$$\begin{aligned} I_4 &= I_2 \otimes I_2 = (\mu_H + \mu_V) \otimes (\mu_H + \mu_V) \\ &= \Pi_{HH} + \Pi_{HV} + \Pi_{VH} + \Pi_{VV}. \end{aligned} \quad (2.20)$$

Although the dataset stores these as four separately estimated binary settings, their projectors are the four outcomes of the two-qubit H/V basis. Applying the Born rule $m_{ab} = \text{Tr}(\Pi_{ab}\rho)$ from Equation (2.11) to Equation (2.20) gives

$$m_{HH} + m_{HV} + m_{VH} + m_{VV} = \text{Tr}(I_4\rho) = \text{Tr}(\rho) = 1.$$

Hence any one of these four probabilities can be recovered exactly from the other three:

$$m_k = 1 - \sum_{p \in \{HH, HV, VH, VV\} \setminus \{k\}} m_p. \quad (2.21)$$

For example, if HV is missing, then $m_{HV} = 1 - m_{HH} - m_{VH} - m_{VV}$. The same equation cannot recover DD : when DD is missing, all four terms in the trace relation are already observed, so the relation supplies no equation containing m_{DD} .

In count form, this four-term normalisation relation appears as Equation (3.19) in the reconstruction of James et al. [11]. Baláz et al. likewise identify these four measurements with the diagonal density-matrix entries and note that unit trace determines one diagonal component from the others [23]. Building on this established normalisation identity, the self-contained argument below classifies all sixteen possible single deletions from the fixed Cartesian projector basis.

There are only two cases to remember. If one of HH, HV, VH, VV is omitted, Equation (2.21) restores it. If any other setting is omitted, all four H/V probabilities remain, so the trace relation merely repeats information already present in

the retained data. The positions 1, 2, 5, 6 (or 0, 1, 4, 5) are only storage positions; the physical statement concerns HH, HV, VH, VV .

Claim 1 (Single-setting completeness of the Cartesian projector set). *After deleting one setting from (2.17), the exact retained probabilities identify every trace-one density matrix if and only if the omitted setting is HH, HV, VH or VV . For any other omission, the retained measurements leave one trace-preserving direction invisible; equivalently, their map has rank fourteen on the fifteen-dimensional space of traceless Hermitian matrices.*

Proof. The sixteen projectors form a basis of the sixteen-dimensional real space of Hermitian 4×4 matrices, so under the Born rule they give sixteen independent linear equations. After any one equation is deleted, the remaining fifteen leave exactly one matrix direction undetermined: by the rank-nullity theorem, the null-space dimension is $16 - 15 = 1$. In other words, there is a nonzero Hermitian matrix Q , unique up to scale, such that

$$\text{Tr}(\Pi_p Q) = 0 \quad \text{for every retained setting } p.$$

By linearity of the Born rule, replacing a candidate A by $A + tQ$ changes probability p by $t \text{Tr}(\Pi_p Q) = 0$. We therefore call Q the *invisible direction*. The question is whether unit trace removes this direction.

First suppose that the omitted setting is one of HH, HV, VH, VV . For any trace-one candidate, Equation (2.21) recovers the missing probability. All sixteen probabilities are then known, and the full projector basis determines the matrix uniquely. Thus no nonzero invisible direction can preserve trace in this case.

Now suppose that one of the other twelve settings is omitted. All four H/V settings are retained. Applying Equation (2.20) to the invisible direction gives

$$\begin{aligned} \text{Tr}(Q) &= \text{Tr}(I_4 Q) \\ &= \text{Tr}(\Pi_{HH} Q) + \text{Tr}(\Pi_{HV} Q) + \text{Tr}(\Pi_{VH} Q) + \text{Tr}(\Pi_{VV} Q) = 0. \end{aligned}$$

Every term is zero because all four settings are retained. Hence Q is traceless: moving along this invisible direction preserves both the retained probabilities and unit trace. The kernel on the fifteen-dimensional traceless space is therefore one-dimensional, so the retained map has rank $15 - 1 = 14$.

It remains only to check that this ambiguity can occur between physical density matrices, not merely between arbitrary Hermitian matrices. Start from the maximally mixed state $I_4/4$, whose four eigenvalues are all $1/4$ and therefore leave room for a small perturbation in either direction. Rescale Q so that its eigenvalues q_i lie in $[-1, 1]$, choose $0 < \varepsilon < 1/4$, and define

$$\rho_{\pm} = \frac{I_4}{4} \pm \varepsilon Q.$$

Because Q is traceless, both matrices have trace one. Their eigenvalues are $1/4 \pm \varepsilon q_i > 0$, so both are full-rank density matrices, and they are distinct because $Q \neq 0$. For every retained setting p ,

$$\text{Tr}(\Pi_p \rho_+) - \text{Tr}(\Pi_p \rho_-) = 2\varepsilon \text{Tr}(\Pi_p Q) = 0.$$

Thus two different physical states produce exactly the same retained probabilities. This proves the converse and completes the claim. $\square$

Concrete example: omitting DD. The proof identifies Q abstractly, but for a missing DD setting it can be chosen as $X \otimes X$. Within the selected single-qubit probes, $\text{Tr}(\mu_a X)$ equals one for $a = D$ and zero for $a \in \{H, V, R\}$. Therefore $X \otimes X$ changes the DD probability and none of the other fifteen. For any $0 < \alpha < 1$, the two full-rank states

$$\rho_{\pm} = \frac{I_4 \pm \alpha X \otimes X}{4} \quad (2.22)$$

have eigenvalues $(1 + \alpha)/4$ and $(1 - \alpha)/4$, each twice, and are therefore valid. They are an explicit pair that the other fifteen settings cannot distinguish.

Finite shots. Exact algebraic redundancy is not statistical redundancy. With finite shots, the three measured frequencies in Equation (2.21) are independently noisy, and their errors accumulate in the derived fourth value. A separately measured fourth H/V setting therefore supplies additional statistical evidence even though it adds no new exact-probability direction. Section 4.3 tests both the exact identifiability distinction and this finite-shot trace-completion effect.

2.1.7 Reconstruction by linear inversion and pseudoinversion

Linear inversion (LI) solves directly for a coordinate representation of the state. Substituting the Pauli expansion (2.10) into the Born rule (2.11) for each of the sixteen James–Kwiat projectors gives a system of sixteen linear equations for the Pauli-coefficient vector $\mathbf{r}$ defined in Equation (2.9):

$$m_p = \sum_{\mu} M_{p\mu} r_{\mu}, \quad M_{p\mu} = \frac{1}{4} \text{Tr}(\Pi_p \gamma_{\mu}), \quad (2.23)$$

where M is the fixed 16×16 matrix mapping $\mathbf{r}$ to the James–Kwiat projector probabilities. The matrix M depends only on the choice of projector set and projector ordering; in our implementation we compute it once and reuse it.

Given a complete and noise-free measurement vector $\mathbf{m} = (m_1, \dots, m_{16})^{\top}$, the coefficient vector $\mathbf{r}$ can be recovered by inverting Equation (2.23),

$$\mathbf{r} = M^{-1} \mathbf{m}, \quad (2.24)$$

followed by reassembly of ρ via (2.10). In the idealised case, this procedure is exact: $\rho = \frac{1}{4} \sum_{\mu} (M^{-1} \mathbf{m})_{\mu} \gamma_{\mu}$ is the unique density matrix consistent with the observed probabilities. We refer to this reconstructor as *linear inversion* and to its noisy-input variant

$$\hat{\rho}_{\text{LI}} = \frac{1}{4} \sum_{\mu} (M^{-1} \hat{\mathbf{m}})_{\mu} \gamma_{\mu} \quad (2.25)$$

as the *linear-inversion estimate* of ρ .

When the measurement set is incomplete, so that only $K < 16$ of the sixteen Kwiat-projector probabilities are observed, the corresponding rows of the measurement matrix M in (2.23) are missing as well, and the resulting matrix (which we will denote by $M_{\text{obs}} \in \mathbb{R}^{K \times 16}$) is no longer square. The literal inverse M^{-1} then does not exist, and the natural generalisation is the *Moore–Penrose*

pseudoinverse: for the underdetermined (“wide”) case $K < 16$ with linearly independent rows,

$$M_{\text{obs}}^+ = M_{\text{obs}}^\top (M_{\text{obs}} M_{\text{obs}}^\top)^{-1}, \quad (2.26)$$

which is the unique matrix returning the minimum- ℓ_2 -norm least-squares solution to the system $M_{\text{obs}} \mathbf{r} = \widehat{\mathbf{m}}_{\text{obs}}$ [24]. The corresponding reconstructor is

$$\widehat{\rho}_{\text{LI}^+} = \frac{1}{4} \sum_{\mu} \left(M_{\text{obs}}^+ \widehat{\mathbf{m}}_{\text{obs}} \right)_{\mu} \gamma_{\mu}. \quad (2.27)$$

In the unrestricted sixteen-coordinate system, fewer than sixteen retained rows always leave a null space, and the pseudoinverse selects the solution with minimum ℓ_2 norm by setting its null-space component to zero, but it does not impose the physical trace condition $r_0 = 1$. With fifteen retained settings, Claim 1 gives two different cases. For an omitted HH , HV , VH or VV setting, the trace constraint removes the sole unrestricted ambiguity, so there is a unique trace-one solution. For any other omission, one ambiguity remains even after imposing trace. Solving the linear system with $r_0 = 1$ and renormalising an unconstrained Moore–Penrose solution afterward are distinct operations. Because each Cartesian projector mixes several Pauli coordinates, the null directions are generally combinations of coefficients rather than individual missing entries.

The resulting bias is therefore subset-dependent. A compact limiting case illustrates the point:

$$\begin{aligned} \text{only } HH \text{ observed, with } \widehat{m}_{HH} > 0 &\implies \widehat{\rho}_{\text{LI}^+} \propto |HH\rangle \langle HH|, \\ &\xrightarrow{\text{trace normalisation}} \widehat{\rho}_{\text{LI}^+} = |HH\rangle \langle HH|. \end{aligned}$$

The reconstructed state is then pure, and hence far from $I_4/4$. Krawczyk et al. [12] use the same pseudoinverse as the classical baseline for incomplete-measurement tomography and as the linear backbone of their learned corrector. We likewise use it as our classical baseline for the missing-measurement task.

Regardless of whether the literal inverse M^{-1} (complete-data case) or the pseudoinverse M_{obs}^+ (incomplete-data case) was used, the resulting $\widehat{\rho}_{\text{LI}}$ is no longer guaranteed to be a valid density matrix. Statistical fluctuations in $\widehat{\mathbf{m}}$ (and, in the pseudoinverse case, the minimum-norm choice in the underdetermined null space) can push some eigenvalues of $\widehat{\rho}_{\text{LI}}$ below zero, violating the PSD constraint of (2.1). Physicality is typically restored by mapping $\widehat{\rho}_{\text{LI}}$ back to the set of valid density matrices. The implementation used here is the following eigenvalue-clipping-and-renormalisation map, denoted by $\mathcal{P}_{\text{clip}}$.

Algorithm 2.1 (Eigenvalue clipping with trace renormalisation ($\mathcal{P}_{\text{clip}}$)). Given a raw matrix estimate $\widehat{\rho}$, compute:

1. symmetrise: $\widehat{\rho} \leftarrow (\widehat{\rho} + \widehat{\rho}^\dagger)/2$;
2. eigendecompose: $\widehat{\rho} = U \text{diag}(\lambda_1, \dots, \lambda_4) U^\dagger$;
3. clamp negatives: $\lambda_i \leftarrow \max(\lambda_i, 0)$;
4. let $s = \sum_j \lambda_j$ and renormalise: $\lambda_i \leftarrow \lambda_i/s$;
5. reassemble: $\widehat{\rho} \leftarrow U \text{diag}(\lambda_1, \dots, \lambda_4) U^\dagger$.

The returned matrix is $\mathcal{P}_{\text{clip}}(\hat{\rho})$. The map is defined for $s > 0$, as holds for every reported input. The classical implementation also evaluates the divisor as $\max(s, 10^{-12})$ to protect against numerical zero.

Algorithm 2.1 is the clipping-based physicality correction used here. It is not the exact Frobenius metric projection. That projection also keeps the eigenvectors, but projects the eigenvalue vector onto the probability simplex, subtracting a common shift τ before clipping, $\lambda'_i = \max(\lambda_i - \tau, 0)$, with τ chosen so that $\sum_i \lambda'_i = 1$, as used in the projected-least-squares estimator of Guță et al. [25]. The map $\mathcal{P}_{\text{clip}}$ can introduce bias when the linear-inversion estimate has negative eigenvalues. Measurement uncertainty enters this correction only through the initial linear estimate.

2.1.8 Cholesky parameterisation

Directly predicting a general complex matrix would require the output entries to obey the conjugate symmetry of Hermiticity. Even the sixteen-real Hermitian packing described in Section 2.1.3 resolves only that bookkeeping: an arbitrary packed output need not have unit trace or be positive semidefinite. Unit trace is an equality constraint, while positive semidefiniteness is a coupled spectral constraint that cannot be enforced by bounding each matrix entry separately. A gradient optimiser is an iterative method that uses loss derivatives to update adjustable parameters (Section 2.3.2). Asking an otherwise unconstrained optimiser to remain inside this curved set is therefore awkward. The *Cholesky parameterisation* [11] instead lets the network emit ordinary real coordinates and maps them deterministically to a physical state:

$$\rho = \frac{TT^\dagger}{\text{Tr}(TT^\dagger)}, \quad (2.28)$$

where $T \in \mathbb{C}^{4 \times 4}$ is lower triangular with a real diagonal. In the principal learned head, the network predicts exactly sixteen unconstrained real numbers per state: four diagonal entries, six real off-diagonal parts and six imaginary off-diagonal parts. The runtime order is $\mathbf{t} = (\mathbf{d}, \mathbf{a}, \mathbf{b})$, where $\mathbf{d} = (d_1, d_2, d_3, d_4)$ and $\mathbf{a}$ and $\mathbf{b}$ list the real and imaginary parts, respectively, in strict-lower order (21, 31, 32, 41, 42, 43). These values are assembled as

$$T(\mathbf{t}) = \begin{pmatrix} d_1 & 0 & 0 & 0 \\ a_{21} + ib_{21} & d_2 & 0 & 0 \\ a_{31} + ib_{31} & a_{32} + ib_{32} & d_3 & 0 \\ a_{41} + ib_{41} & a_{42} + ib_{42} & a_{43} + ib_{43} & d_4 \end{pmatrix}, \quad \mathbf{t} \in \mathbb{R}^{16}. \quad (2.29)$$

Thus the count is $4 + 2\binom{4}{2} = 4 + 12 = 16$ real network outputs.

The constraints now follow from the map rather than from learned behaviour. The product $TT^\dagger$ is Hermitian, and for every vector v ,

$$v^\dagger TT^\dagger v = \|T^\dagger v\|_2^2 \geq 0,$$

which proves positive semidefiniteness. Moreover, $\text{Tr}(TT^\dagger) = \|T\|_F^2 > 0$ for $T \neq 0$, so the division in Equation (2.28) gives unit trace. Conversely, every PSD

matrix admits such a factor. Matrix multiplication and scalar normalisation are differentiable for $T \neq 0$, so gradients can pass through the map without constrained optimisation or a substantive projection after each forward pass.

The advantage is therefore not a smaller output: a physical two-qubit state has fifteen free real parameters, whereas the network predicts sixteen. The extra coordinate is a continuous redundancy because T and cT , for any real $c \neq 0$, give the same normalised density matrix. Fixing the scale and requiring $d_j \geq 0$ gives the conventional unique Cholesky factor for a positive-definite state. The principal learned head leaves the four d_j unconstrained; this preserves physicality but also retains discrete sign equivalences, since T and TS give the same $TT^\dagger$ for a diagonal S with entries in $\{-1, 1\}$. Additional coordinate ambiguity appears near rank-deficient boundary states. These redundancies are usually easier for ordinary gradient training than the coupled PSD and trace constraints on ρ itself.

Equation (2.28) is defined on the domain $T \neq 0$, which is assumed throughout. During principal-model training and checkpoint selection, the implementation optionally applies the near-identity PSD-and-trace safeguard quantified in Attachment A.1.5: it adds a small perturbation before the clipping-and-renormalisation core of Algorithm 2.1. This stochastic extension of $\mathcal{P}_{\text{clip}}$ corrects rare numerical drift; the Cholesky map supplies the structural guarantee in exact arithmetic. The final common-noise accuracy evaluation disables this optional safeguard, while the conservative timing path retains its 4×4 eigendecomposition. Macarone Palmieri et al. [13] also use the Cholesky parameterisation, which is the output representation of our final learned reconstructor.

2.1.9 Maximum-likelihood estimation

Linear inversion asks for a state that solves linear equations. Maximum-likelihood estimation (MLE) instead asks which physical state most plausibly produced the observed counts. Given k_p projected outcomes in n_p shots of setting p , the binomial model of (2.12) gives

$$\mathcal{L}(\rho) = \prod_p \binom{n_p}{k_p} m_p(\rho)^{k_p} (1 - m_p(\rho))^{n_p - k_p}, \quad (2.30)$$

and

$$\hat{\rho}_{\text{MLE}} = \arg \max_{\rho \in \mathcal{S}} \log \mathcal{L}(\rho), \quad (2.31)$$

where $\mathcal{S}$ is the set of valid density matrices [10]. Unobserved settings simply contribute no factor to this likelihood.

Our implementation optimises the Cholesky factor in (2.28), so positivity and unit trace hold at every iterate. Restricting the search to trace-one matrices allows MLE to exploit the redundancy established in Claim 1. With exact probabilities, omitting one of HH , HV , VH or VV still leaves one unique compatible state. Omitting any of the other twelve settings leaves a traceless likelihood-flat direction; the full-rank construction in the proof of Claim 1 shows that positivity does not restore global identifiability. With finite shots, even the four special fifteen-setting and all-sixteen estimates can differ because the sixteenth setting supplies an independent noisy likelihood term.

For identifiable full-rank states under the usual regularity conditions, MLE has the familiar asymptotic-efficiency property as the total shot count grows [26]. Pure and rank-deficient states lie on the boundary, so the usual regularity assumptions need not hold. The price of the statistical model is a new iterative optimisation for every reconstructed state. The negative log-likelihood is convex as a function of ρ over the convex set $\mathcal{S}$ and admits convex formulations such as projected-gradient optimisation over the density matrix [27]; the Cholesky coordinates used here make the same objective non-convex in the factor entries.

On a genuinely non-identifying subset, the pseudoinverse applies its minimum-norm convention without shot-dependent reliability. MLE uses the noise model and physical constraints. Positivity restricts the compatible family; initialisation and optimisation conventions determine which member a finite implementation returns when the likelihood is flat. This explains why likelihood reconstruction is most favourable with complete, high-shot data, while a finite optimiser can depend more strongly on its initialisation and parameterisation as non-identifying settings are removed. In either case, the state representation still contains $d^2 = 4^n$ real coordinates, so the representation and measurement costs scale exponentially with the number of qubits.

2.2 Reconstruction metrics

Having defined how a state is reconstructed, we next need quantities that compare an estimate $\hat{\rho}$ with its target ρ . We use one primary full-state distance, one inexpensive matrix-space check and one task-specific entanglement quantity.

2.2.1 Fidelity and Bures distance

A standard quantum-state similarity measure is the *squared Uhlmann fidelity*,

$$F(\rho, \sigma) = \left(\text{Tr} \sqrt{\sqrt{\rho} \sigma \sqrt{\rho}} \right)^2 \in [0, 1], \quad (2.32)$$

which satisfies $F(\rho, \sigma) = 1$ if and only if $\rho = \sigma$, and reduces to $|\langle \psi | \phi \rangle|^2$ when both states are pure. Fidelity itself is not a metric, but its corresponding *Bures distance*

$$d_B(\rho, \sigma) = \sqrt{2 - 2\sqrt{F(\rho, \sigma)}} \in [0, \sqrt{2}] \quad (2.33)$$

is a proper metric on density matrices [28, 29]. It respects the curved geometry of quantum states and is closely related to their statistical distinguishability. Smaller values are better, with zero only for identical states. We use it as the primary reconstruction metric and report its mean over the test set.

A useful conversion is

$$F(\rho, \sigma) = \left(1 - \frac{d_B^2(\rho, \sigma)}{2} \right)^2, \quad (2.34)$$

so that, for instance, $d_B = 0.10$ corresponds to $F \approx 0.990$ (a one-percent infidelity), $d_B = 0.05$ to $F \approx 0.9975$, and $d_B \rightarrow 0$ to perfect reconstruction. Because the map is nonlinear, applying it to a *mean* Bures distance gives an illustrative single-state-equivalent fidelity. We use converted values only in this illustrative sense.

Why differentiation is numerically delicate. For the positive-semidefinite matrix $A = \sqrt{\rho} \sigma \sqrt{\rho}$, let μ_i denote its eigenvalues and write $f = \sqrt{F}$ for the root fidelity. For positive μ_i ,

$$f = \text{Tr} \sqrt{A} = \sum_i \sqrt{\mu_i}, \quad \frac{\partial f}{\partial \mu_i} = \frac{1}{2\sqrt{\mu_i}}. \quad (2.35)$$

The value of the fidelity remains well defined when an eigenvalue reaches zero, but this derivative becomes unbounded. The same square-root boundary appears when forming $\sqrt{\rho}$, so pure and other rank-deficient states are the difficult cases for gradient-based optimisation.

An explicit eigendecomposition introduces a second, implementation-level difficulty. A matrix square root is assembled as $V \text{diag}(\sqrt{\lambda_i}) V^\dagger$, but the eigenvectors in V are not unique within a repeated eigenspace. Applying backpropagation (the chain-rule gradient computation introduced in Section 2.3.2) through a complex Hermitian eigendecomposition therefore becomes ill-conditioned as eigenvalue gaps close, even though the assembled matrix function is invariant to the choice of eigenvector phases and basis. Such clustered spectra occur near both the maximally mixed state and low-rank states. Double precision reduces round-off, small negative eigenvalues and cancellation, but it does not remove an exact spectral degeneracy.

The implemented regularised training loss uses ordinary PyTorch automatic differentiation [30], the software evaluation of the chain-rule calculation described in Section 2.3.2, rather than a custom matrix-square-root derivative. Its spectral subcomputation is promoted to complex double precision, matrices are re-Hermitised, eigenvalues are floored or clamped before square roots, and a small random real-symmetric perturbation separates repeated eigenvalues before the first eigendecomposition. The reported evaluation metric omits this training-only perturbation. This mechanism is distinct from the post-Cholesky safeguard, which corrects small complex-single-precision violations of positive semidefiniteness. Attachment A.1.5 specifies both paths and quantifies their effects.

There is also a scalar choice between optimising d_B and d_B^2 . The final square root in $d_B = \sqrt{d_B^2}$ has an unbounded derivative as $d_B \rightarrow 0$. Using d_B^2 removes only that outer square root; it does not remove the spectral boundary in Equation (2.35), and it changes the weighting of examples because

$$\nabla_\theta d_B^2 = 2d_B \nabla_\theta d_B. \quad (2.36)$$

The squared form therefore downweights already accurate examples and changes its scale relative to any auxiliary loss. Both forms were tested; the unsquared distance was retained as described in Section 3.2.

2.2.2 Hilbert–Schmidt distance

The Bures distance is the primary physically motivated metric in this thesis. Each evaluation requires two matrix square roots and their eigendecompositions. A much cheaper alternative is the *Hilbert–Schmidt (HS) distance*,

$$d_{\text{HS}}(\rho, \sigma) = \|\rho - \sigma\|_F = \sqrt{\text{Tr}((\rho - \sigma)^\dagger (\rho - \sigma))} = \sqrt{\text{Tr}((\rho - \sigma)^2)}, \quad (2.37)$$

where the final equality holds because $\rho - \sigma$ is Hermitian. This is simply the Frobenius distance between the two matrices. The Hilbert–Schmidt distance is a true metric on Hermitian matrices and is bounded above by $\sqrt{2}$ for valid density matrices. Its flat matrix-space geometry makes it an efficient diagnostic, while Bures distance follows the quantum-state geometry used for architectural selection. We use d_{HS} as a training sanity check and report it alongside d_B in the final model comparisons.

Macarone Palmieri et al. [13, Eqs. (C5) and (C11)] justify a related Euclidean objective in Cholesky-factor coordinates. If $\rho = C_\rho C_\rho^\dagger$ and $\sigma = C_\sigma C_\sigma^\dagger$ are normalised states represented by chosen Cholesky factors, their result is, in the convention of this thesis,

$$d_B^2(\rho, \sigma) \leq \|C_\rho - C_\sigma\|_F^2 = \|\mathbf{c}_\rho - \mathbf{c}_\sigma\|_2^2. \quad (2.38)$$

The final expression is the sum of squared errors between the real factor coordinates, or an elementwise MSE up to its fixed averaging constant. This is a bound on d_B^2 by *factor-space* squared error; it is not the density-matrix distance $d_{\text{HS}}^2(\rho, \sigma)$ of Equation (2.37). The inequality makes factor-coordinate MSE a principled proxy, but it is not an equality and does not give the proxy the same gradients as direct Bures optimisation. Section 3.2 compares these choices empirically.

2.2.3 Concurrence and entanglement classification

The two metrics above measure full-state reconstruction quality. We complement them with direct agreement in a physical scalar, Wootters concurrence, and with a secondary downstream classifier. Two-qubit entanglement is quantified by the *concurrence* of Wootters [31],

$$\mathcal{C}(\rho) = \max(0, \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4), \quad (2.39)$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$ denote the ordered eigenvalues of the matrix

$$\mathcal{R}_\rho = \sqrt{\sqrt{\rho} \tilde{\rho} \sqrt{\rho}}, \quad \tilde{\rho} = (Y \otimes Y) \rho^* (Y \otimes Y), \quad (2.40)$$

and ρ^* denotes complex conjugation in the computational basis. Concurrence ranges from 0 (separable states) to 1 (maximally entangled states). The labels supplied with the dataset use a nominal threshold near 10^{-6} , below the approximately 3×10^{-4} maximum discrepancy observed in our numerical cross-check. For the binary analysis we therefore apply each displayed threshold identically to the true and reconstructed concurrence and report results at 10^{-3} , 10^{-2} and 0.05.

For the downstream evaluation we train an independent entanglement classifier on clean James–Kwiat projector-probability vectors, with concurrence-derived labels as targets. Its weights are then frozen. Accuracy on probabilities implied by reconstructed states provides a learned binary proxy whose ceiling is set by the classifier’s reference error. Direct concurrence in Section 4.5.2 is therefore the more physically interpretable analysis.

2.3 Learning-based reconstruction approaches

The preceding sections describe tomography as a constrained inverse problem: the observation is a noisy, possibly incomplete measurement record, and the desired output is a physical density matrix. A neural network (NN) reconstructor retains that mathematical structure and uses examples to learn the remaining part of the inverse map, including how to combine noisy measurements and how to select a point estimate when the observations admit more than one state.

This section is self-contained for readers without a machine-learning background. It first defines the shared measurement representation, explains what is learned and how gradient-based training works, then develops the building blocks behind four broad reconstruction families: direct feed-forward and transformer regression, adversarial reconstruction, and iterative generative reconstruction through diffusion or flow matching. MLP and transformer name network backbones; adversarial training and flow matching name objectives and inference procedures whose learned components may use either backbone. The figures in this section show the mechanisms conceptually. The exact layer counts and checkpoint configurations are given in Chapter 3 and Attachment A.1.

2.3.1 Representing noisy and incomplete measurements

The reconstruction approaches compared in this thesis need to distinguish a precise observation from a noisy one and both from a setting that was not recorded. We therefore accompany the sixteen empirical probabilities $\widehat{m}_p$ by a sixteen-dimensional *confidence* channel

$$c_p = \frac{\log(1 + n_p)}{\log(1 + n_{\max})} \in [0, 1], \quad p = 1, \dots, 16, \quad (2.41)$$

where n_p is the number of shots for setting p and $n_{\max} = 100\,000$ is a fixed upper anchor matching the largest per-setting count in the evaluation sweeps. This is deliberately a coarse, monotone *reliability coordinate*. We reserve $c_p = 0$ for a missing setting ($n_p = 0$); one shot gives $c_p \approx 0.06$, while 100, 1 000, 10 000 and 100 000 shots give approximately 0.40, 0.60, 0.80 and 1. Larger counts are clipped to one. The network learns from examples how strongly each confidence should affect its reconstruction.

The selected conditional pipelines also lift each scalar probability through a fixed Fourier-feature map [32]. Two frequency grids are used by the final implementations:

$$\begin{aligned} \mathbf{f}_8^{\text{harm}}(m) &= \left[m, \{\sin(2\pi km), \cos(2\pi km)\}_{k=1}^8 \right], \\ \mathbf{f}_8^{\text{dyad}}(m) &= \left[m, \{\sin(\pi 2^k m), \cos(\pi 2^k m)\}_{k=0}^7 \right]. \end{aligned} \quad (2.42)$$

Both maps have seventeen components and retain the original scalar m , so neither adds measurement information. They supply fixed nonlinear coordinates that a learned affine layer can combine more easily. RhoTransformer and DeepSets use the harmonic grid $\mathbf{f}_8^{\text{harm}}$; the cGAN and both flow-matching models use the dyadic grid $\mathbf{f}_8^{\text{dyad}}$ retained in their family-specific implementations. In either case, the vector presented for slot p is multiplied by c_p , so a missing setting contributes the zero vector. Thus the selected models share the same input shape and confidence

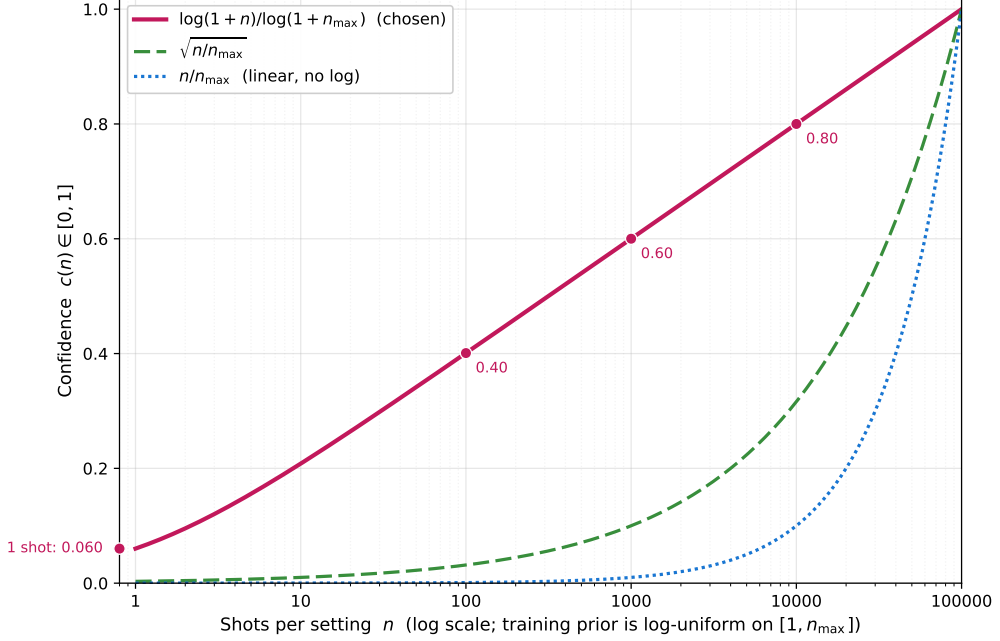


Figure 2.1: The logarithmic confidence map used in the experiments (magenta), compared with linear (blue, dotted) and square-root (green, dashed) alternatives. On the logarithmic shot-count axis, the selected map spreads the log-uniform training range approximately evenly. The missing-setting value $n = 0$, $c = 0$ lies outside the logarithmic horizontal axis; the first plotted point is one shot, $c(1) \approx 0.06$. The network learns how to use this reliability coordinate.

gating principle while retaining their documented fixed frequency grids. The naive baselines instead use the raw measurement/confidence pair. Figure 2.1 shows why the logarithmic coordinate spreads the log-uniform training range more evenly than simpler alternatives. Confidence also weights the optional measurement-consistency term defined in Equation (3.1); Section 4.7 measures its effect.

2.3.2 Learning tomography from examples

We use the standard supervised-learning setup and empirical-risk terminology of Goodfellow et al. [33, Ch. 5]. One supervised example is a pair $(\mathbf{y}_i, \rho_i)$. The observation record

$$\mathbf{y}_i = (\widehat{\mathbf{m}}_i, \mathbf{c}_i) \quad (2.43)$$

contains the sixteen empirical projector probabilities and their sixteen confidence values; $c_{i,p} = 0$ marks a missing setting. The target ρ_i is the density matrix from which that synthetic record was generated. A model F_θ is a function whose adjustable real parameters θ comprise all weights, biases and learned embeddings. The architecture specifies how the function is assembled; *learning* or *training* chooses the parameter values from examples. A *forward pass* evaluates

$$\widehat{\rho}_i = F_\theta(\mathbf{y}_i), \quad (2.44)$$

and a scalar loss $\ell(\widehat{\rho}_i, \rho_i; \mathbf{y}_i)$ measures the error, with smaller values preferred. Training seeks parameters that give a small average loss over many examples; at inference, the fitted parameters are frozen.

For a differentiable scalar objective $L(\theta)$, its *gradient* $\nabla_{\theta}L$ collects one partial derivative with respect to each parameter. It points in the direction of fastest local increase, so basic *gradient descent* moves a small distance in the opposite direction, $\theta \leftarrow \theta - \eta \nabla_{\theta}L$, where the positive *learning rate* η controls the step size. A neural network chains many differentiable operations. Starting at the scalar loss, *backpropagation* traverses that computation in reverse and reuses local chain-rule derivatives to obtain the derivative with respect to every parameter; an *optimiser* then uses those gradients to update the parameters. Automatic-differentiation software performs this reverse-mode calculation [33, Chs. 6 and 8].

In the selected direct DeepSets and RhoTransformer models, the network first predicts the sixteen real Cholesky coordinates of Equation (2.29); the deterministic map in Equation (2.28) then converts them into $\hat{\rho}_i$. In exact arithmetic, this final map enforces positive semidefiniteness and unit trace. For $T \neq 0$, the map is differentiable, so backpropagation carries a density-matrix loss through it to the coordinate predictor and the preceding layers. The finite-precision safeguards are described in Attachment A.1.5.

The data are divided by state into training, validation and test splits. The training split changes θ ; the validation split selects designs and indicates when further epochs no longer improve held-out performance (a validation plateau). The fixed test split is excluded from fitting, early stopping and model selection. It is used to report performance after the relevant checkpoint has been fixed by validation. A low training loss alone can result from memorisation (over-fitting). Agreement between training and held-out behaviour is the relevant evidence that the model has learned a rule that generalises to new states from the evaluated distribution. Attachment A.1.2 records the selection rule and split role for the reported checkpoints.

Visualising examples with t-SNE. High-dimensional example clouds are difficult to inspect directly. The t-distributed stochastic neighbour embedding (t-SNE) method of van der Maaten and Hinton [34] maps local neighbourhood similarities to two dimensions; its perplexity sets the effective neighbourhood scale. It is useful for displaying nearby groups but does not preserve global distances, cluster areas or point density, and its layout depends on the chosen hyperparameters. Figure 3.1 therefore uses t-SNE only as a qualitative view of the sixteen-dimensional measurement vectors and checks its main observation directly in the original space.

The state targets in the training split are fixed, but during training the experiments draw fresh shot noise, shot counts and missingness patterns online. Validation and test observations are instead fixed records, as described in Section 3.1.2. Let $q(\mathbf{y} \mid \rho_i)$ denote this observation-generation distribution for state ρ_i , and let N_{train} be the number of states in the training split. The expected empirical objective targeted by online training is therefore

$$\hat{R}(\theta) = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \mathbb{E}_{\mathbf{Y} \sim q(\cdot \mid \rho_i)} [\ell(F_{\theta}(\mathbf{Y}), \rho_i; \mathbf{Y})]. \quad (2.45)$$

This equation defines an average over all training states and all observations that could be generated for each one. Computing that complete average before every

update would be prohibitively expensive. A *batch* is a group of examples processed together; during training, a *mini-batch* B_s is a small, randomly selected subset of training-state indices used for one parameter update, and $|B_s|$ is its *mini-batch size* (often shortened to *batch size*). At training step s , a fresh observation $\mathbf{Y}_{i,s}$ is drawn for every state in the mini-batch, the losses are averaged and the optimiser takes one step:

$$\begin{aligned}\mathbf{Y}_{i,s} &\sim q(\cdot \mid \rho_i), & i \in B_s, \\ \hat{L}_s(\theta) &= \frac{1}{|B_s|} \sum_{i \in B_s} \ell(F_\theta(\mathbf{Y}_{i,s}), \rho_i; \mathbf{Y}_{i,s}), \\ \hat{\mathbf{g}}_s &= \nabla_\theta \hat{L}_s(\theta) \Big|_{\theta=\theta_s}, \\ \theta_{s+1} &= \theta_s - \eta_s \hat{\mathbf{g}}_s.\end{aligned}\tag{2.46}$$

Here $\hat{L}_s$ is the sampled mini-batch loss, $\hat{\mathbf{g}}_s$ is its gradient at the current parameters, and η_s is the learning rate. Backpropagation computes $\hat{\mathbf{g}}_s$; the final line is the simplest gradient-descent update. The reported neural models instead use Adam [35] or AdamW [36]: both track exponential moving averages of the gradient and squared gradient to choose parameter-wise step sizes; AdamW additionally applies weight decay separately from the loss gradient. If $\mathbf{u}_s$ denotes Adam’s adaptive update direction and $\lambda \geq 0$ is the weight-decay coefficient, the decoupled AdamW step can be written schematically as

$$\theta_{s+1} = (1 - \eta_s \lambda) \theta_s - \eta_s \mathbf{u}_s.\tag{2.47}$$

The factor $1 - \eta_s \lambda$ gently contracts the parameters at every training step. This should be distinguished from adding an L2 penalty $\lambda \|\theta\|_2^2/2$ to the loss: the latter adds $\lambda \theta$ to the gradient before Adam’s coordinate-wise rescaling, whereas AdamW applies the contraction outside that adaptive transformation. They coincide under ordinary gradient descent after matching conventions, but not generally under Adam. Thus the reported RhoTransformer setting is most precisely described as *decoupled weight decay*, not as an L2 loss term.

One pass in which every clean training state is presented once is an *epoch*. The states are shuffled into successive mini-batches, and online observation generation can pair the same ρ_i with a different noisy or incomplete record in the next epoch. Each mini-batch therefore gives a Monte-Carlo estimate of both the state average and the observation expectation in Equation (2.45). With one fixed stored record per state, as used for validation and testing, that equation reduces to the usual finite average of example losses. The loss also need not equal the final reporting metric alone. In particular, the selected direct DeepSets and RhoTransformer models combine a regularised Bures term with a measurement-consistency term; Chapter 3 defines the exact objective.

Some of the reported training loops additionally use *global gradient-norm clipping* [37]. Let $\hat{\mathbf{g}}$ collect the gradients of all trainable parameters after backpropagation, and let $c > 0$ be a threshold. The clipped gradient is

$$\tilde{\mathbf{g}} = \frac{\hat{\mathbf{g}}}{\max(1, \|\hat{\mathbf{g}}\|_2/c)}.\tag{2.48}$$

Gradients whose global Euclidean norm is at most c are unchanged; larger gradients are rescaled to norm c without changing their direction. This is an optimiser-

side safeguard against unusually large mini-batch gradients. It is unrelated to the eigenvalue-clipping physicality map of Algorithm 2.1.

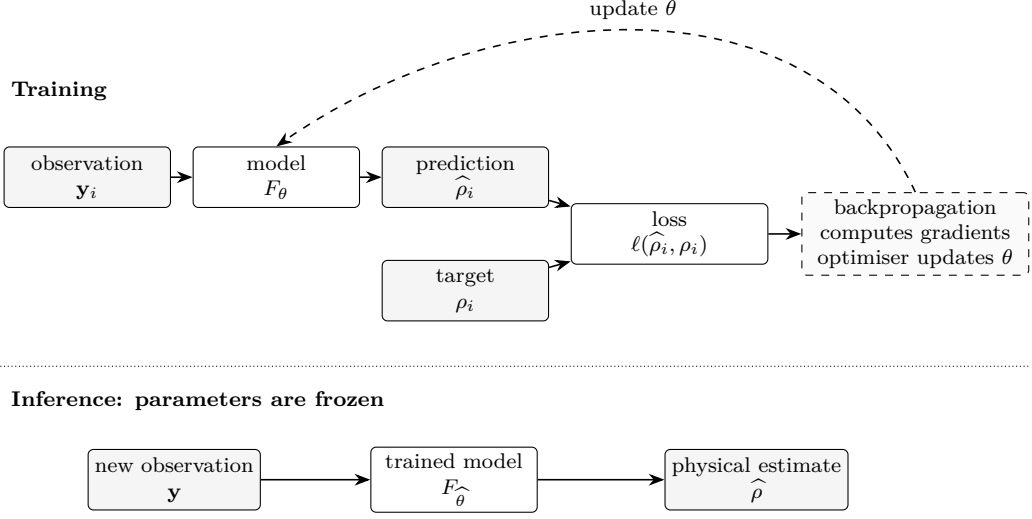


Figure 2.2: During training, a labelled target defines a loss, backpropagation computes its gradients and the optimiser updates the parameters. At inference, a new record passes through the frozen model without a target or parameter update; held-out evaluation compares the estimate with a known target only after the prediction. Generative pipelines retain the same separation.

The population-level interpretation is useful in incomplete tomography. If the training examples are draws from a joint distribution of random variables $(\mathbf{Y}, \rho)$, an ideal deterministic estimator satisfies the conditional-risk rule [33, Ch. 5]

$$F^*(\mathbf{y}) \in \arg \min_{\sigma \succeq 0, \text{Tr}(\sigma)=1} \mathbb{E}[\ell(\sigma, \rho; \mathbf{y}) \mid \mathbf{Y} = \mathbf{y}]. \quad (2.49)$$

Thus the learned output is a loss-dependent representative of the conditional state distribution induced by the data generator. When the measurements identify the state and noise vanishes, that conditional distribution can concentrate at the true state. When a retained projector set has a null direction, several states may remain compatible with the same observation; then a deterministic network also reflects the state distribution seen during training. This is the precise sense in which it uses a learned prior rather than computing a universal algebraic inverse.

2.3.3 Sequential selection of measurement settings

Adaptive QST has previously been formulated with learned sequential policies [38, 12]; the protocol below is the closed-loop acquisition rule evaluated in this thesis. Incomplete tomography raises two related but distinct questions. *Reconstruction* estimates a state from the record already available; *experimental design* chooses which additional setting to acquire when the budget permits another measurement. Active selection addresses the second question. It improves the evidence supplied to a reconstructor rather than changing the reconstructor’s fitted parameters.

Let S_t be the set of settings observed after t acquisition rounds and let $\mathbf{y}_t$ be the resulting partial record. With both learned models frozen, one round computes

$$\begin{aligned}\hat{\rho}_t &= F_{\hat{\theta}}(\mathbf{y}_t), & \widehat{m}_{t,p}^{\text{pred}} &= \text{Tr}(\Pi_p \hat{\rho}_t), \\ s_{t+1} &= \arg \max_{s \notin S_t} [\mathbf{a}_{\hat{\phi}}(\mathbf{y}_t, \widehat{\mathbf{m}}_t^{\text{pred}})]_s, & S_{t+1} &= S_t \cup \{s_{t+1}\}.\end{aligned}\quad (2.50)$$

The selected projector is then applied to N freshly prepared copies of the quantum system,

$$k_{s_{t+1}} \sim \text{Binomial}(N, \text{Tr}(\Pi_{s_{t+1}} \rho)), \quad \widehat{m}_{s_{t+1}} = \frac{k_{s_{t+1}}}{N}, \quad (2.51)$$

and its frequency and confidence are appended to $\mathbf{y}_t$. The loop stops after an externally chosen budget b of distinct settings, and $\hat{\rho}_b = F_{\hat{\theta}}(\mathbf{y}_b)$ is the final reconstruction. The first decision, made from an empty record, is necessarily common across states; later choices can depend on the measured history.

The ideal one-step objective makes clear what the selector is approximating. Let $\mathbf{y}_t \oplus (s, z, c_N)$ denote the record obtained by appending outcome frequency z and its confidence c_N at an unobserved setting s . Under the joint state-and-observation distribution, define the conditional expected Bures reduction

$$u(s \mid \mathbf{y}_t) = \mathbb{E}[d_B(F_{\hat{\theta}}(\mathbf{y}_t), \rho) - d_B(F_{\hat{\theta}}(\mathbf{y}_t \oplus (s, C_s/N, c_N)), \rho) \mid \mathbf{Y}_t = \mathbf{y}_t], \quad (2.52)$$

where $C_s \mid \rho \sim \text{Binomial}(N, \text{Tr}(\Pi_s \rho))$. The expectation covers both uncertainty about the true state conditional on the current record and the as-yet unseen binomial outcome. A Bayes-greedy policy would maximise $u(s \mid \mathbf{y}_t)$ over $s \notin S_t$. The trained selector does not construct this conditional distribution explicitly. Instead, known synthetic training states and sampled future outcomes provide Monte Carlo targets for a direct approximation. It is therefore a myopic policy: it learns immediate one-step reconstruction gain, not the optimal expected loss after all remaining rounds. Recomputing its scores after every acquired outcome still makes the resulting trajectory adaptive.

For a concrete reading of the budget, suppose $b = 5$. After four rounds the record contains four empirical probabilities, each estimated from 10^4 binary shots. The selector scores the twelve settings that remain unobserved, chooses the fifth, and the apparatus estimates that probability with another 10^4 fresh copies. The final reconstruction then uses five observed probabilities. If the budget were $b = 4$, the procedure would stop before this fifth choice. Thus b counts distinct settings; it is not a number of individual shots. The unselected settings remain missing rather than being measured at a lower shot count.

The same closed acquisition loop supports both human-operated and automated use. In a human-in-the-loop laboratory workflow, an experimentalist configures the projector recommended by the selector, collects the counts and returns them to the model before requesting the next recommendation. An instrument controller can perform these operations automatically. In either case the selector, rather than human judgement, supplies the next-setting decision. It is also different from adaptive *remeasurement*. The protocol evaluated in this thesis samples settings without replacement: each newly selected setting receives $N = 10^4$ shots once, so budget b uses bN state preparations. It neither revisits an observed slot nor reallocates shots among slots.

Why can the ordering matter? The current observations define a noise-broadened compatible region

$$\mathcal{R}_t = \{\sigma \succeq 0, \text{Tr}(\sigma) = 1 : \text{Tr}(\Pi_p \sigma) \approx \widehat{m}_p \text{ for } p \in S_t\}.$$

A useful next projector cuts across a direction in which plausible states still differ, whereas a nearly redundant projector narrows this region very little. For example, suppose two states are still compatible with the acquired record. If they predict almost the same probability for candidate setting p but markedly different probabilities for candidate setting q , measuring p will do little to distinguish them, whereas measuring q can decide which part of the compatible region remains plausible. The selector tries to favour the latter kind of setting. Because the observed values change the remaining ambiguity, the best next setting can depend on the state through its measurement history. The selector learns a direct proxy for downstream Bures-error reduction from the current record, rather than representing $\mathcal{R}_t$ as a calibrated posterior. Synthetic ground truth is used only to construct this offline training signal. During evaluation or deployment, the policy sees the acquired record and current reconstruction, never the unknown true state. Figure 2.3 separates these two stages.

The trace identity of Equation (2.21) constrains this utility only after a relevant history has been acquired. Once three of HH, HV, VH, VV are known exactly, the fourth adds no new algebraic direction; at an empty record, however, the four settings are not collectively redundant. With finite shots, even the fourth supplies an independent noisy estimate. The theorem can therefore motivate deferring one H/V setting late in a trajectory, but it does not imply that all four should receive low scores at the first decision.

Simulation also permits counterfactual comparisons that no deployable policy can make. Conditional on one state, pre-draw a vector of potential frequencies

$$\Xi_s = C_s/N, \quad C_s \sim \text{Binomial}(N, \text{Tr}(\Pi_s \rho)), \quad s = 1, \dots, 16, \quad (2.53)$$

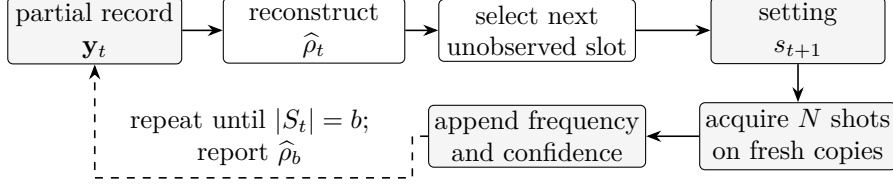
independently across settings. A causal policy reveals Ξ_s only after it selects s . Drawing all sixteen values in advance does not change that policy's distribution, because unselected values remain hidden; it does allow several counterfactual subsets to be evaluated with the same potential outcomes. For distinct size- b candidate subsets $A_1, \dots, A_K$, define the per-state common-noise hindsight value

$$h_b^{(K)}(\rho, \Xi) = \min_{1 \leq i \leq \min\{K, \binom{16}{b}\}} d_B(F_{\hat{\theta}}(\mathbf{y}(A_i, \Xi)), \rho), \quad (2.54)$$

where $\mathbf{y}(A_i, \Xi)$ reveals precisely the entries indexed by A_i . This quantity uses the true state to choose among counterfactuals and is therefore a noncausal headroom reference, not a policy or an attainable ceiling. At $b = 0$ and $b = 16$ the subset is unique, so subset-choice headroom is exactly zero. Moreover, minimising separately at each b does not require the selected subsets to form a nested acquisition path. Consequently, $h_b^{(K)}$ need not decrease with b : a fixed learned reconstructor is not mathematically constrained to improve when another noisy value is supplied, and the hindsight choice disappears at the complete set. After any fixed first action, only fifteen distinct second actions remain; evaluating all of them gives an exhaustive one-step hindsight reference, and a larger distinct-action pool would

require remeasurement or a new noise draw. The exact candidate pools used in the experiments are specified in Section 3.7.3.

(a) Deployment: automated closed acquisition loop



(b) Offline selector training

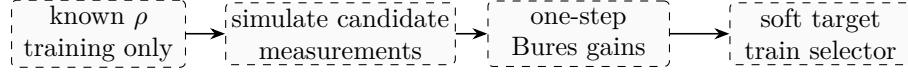


Figure 2.3: At deployment, each round adds one previously unobserved setting and recomputes the state estimate. A human operator may execute each recommendation, or the loop may be automated; the model weights remain fixed and settings are not remeasured. During offline training only, known synthetic states allow candidate settings to be ranked by one-draw Monte Carlo estimates of their immediate Bures improvement.

2.3.4 Neural-network building blocks

At the level used here, a neural network chains parameterised affine maps with differentiable nonlinear operations. The preceding subsection described backpropagation at whole-model level; here the notation follows the standard layer-level treatment of deep feed-forward networks [33, Ch. 6]. Its basic learned operation is an affine map

$$\mathbf{a}^{(\ell)} = W^{(\ell)} \mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}, \quad W^{(\ell)} \in \mathbb{R}^{d_\ell \times d_{\ell-1}}, \quad \mathbf{b}^{(\ell)} \in \mathbb{R}^{d_\ell}. \quad (2.55)$$

The vector $\mathbf{h}^{(\ell)}$ is a *hidden representation*: an internal coordinate system fitted for the task. A coordinate-wise nonlinear activation σ gives $\mathbf{h}^{(\ell)} = \sigma(\mathbf{a}^{(\ell)})$. The nonlinearity is essential. Without it, two successive affine maps collapse into one, $W_2(W_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2 = (W_2 W_1) \mathbf{x} + (W_2 \mathbf{b}_1 + \mathbf{b}_2)$, so additional layers would not create a more expressive input–output relation. The rectified linear unit (ReLU) uses $\text{ReLU}(u) = \max(0, u)$.

Other common choices are the logistic sigmoid $s(u) = (1 + e^{-u})^{-1}$; its zero-centred rescaling $\tanh(u) = 2s(2u) - 1$; the Gaussian error linear unit $\text{GELU}(u) = u\Phi(u)$, where Φ is the standard-normal cumulative distribution function; and the sigmoid linear unit $\text{SiLU}(u) = u s(u)$. The last function is also called *Swish* when its sigmoid slope is fixed to one [39, 40, 41]. Their scalar responses are compared in Figure 2.4.

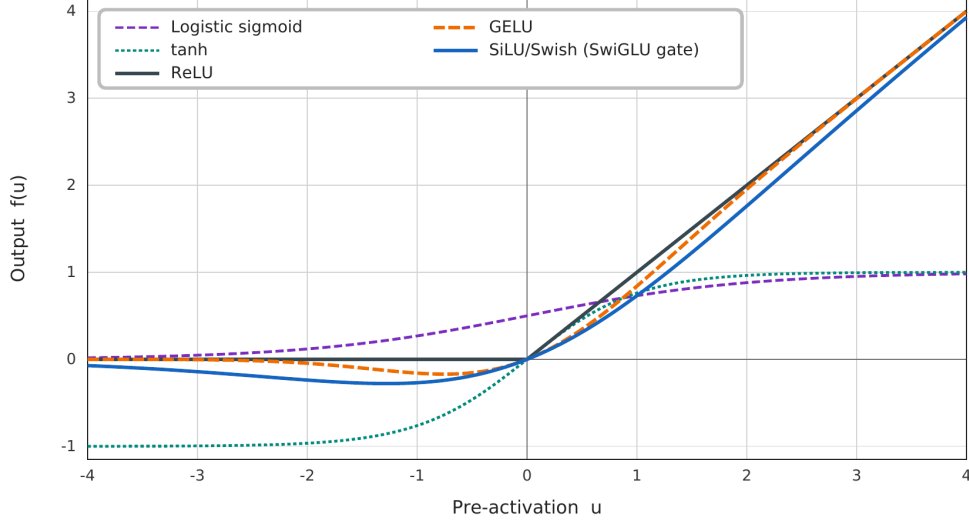


Figure 2.4: Representative scalar nonlinearities. The plotted SiLU/Swish curve is the gate used by SwiGLU; SwiGLU itself is the two-branch vector block in Equation (2.57) and therefore has no single scalar response curve.

Activation shapes affect optimisation through the derivatives multiplied during backpropagation. Sigmoid and tanh saturate at large magnitudes, where their derivatives approach zero; rectifiers retain a nonzero positive-side gradient. ReLU, GELU and SiLU/Swish consequently improved optimisation or accuracy over saturating or earlier activations in the settings studied in Refs. [42, 43, 39, 40, 41]. These results motivate the modern choices used here but do not establish a universal ranking across tasks.

Three refinements recur in the reported architectures. A *residual connection* returns $\mathbf{h} + G_\theta(\mathbf{h})$, giving information and gradients a direct path around a learned transformation [44]. *Layer normalisation* (LN) rescales the coordinates of one representation using their own mean and variance [45],

$$\text{LN}(\mathbf{h}) = \gamma \odot \frac{\mathbf{h} - \mu(\mathbf{h})}{\sqrt{\nu(\mathbf{h}) + \epsilon}} + \beta, \quad (2.56)$$

with learned scale γ , learned shift β and numerical stabiliser $\epsilon > 0$. A Swish-gated linear unit (SwiGLU) is a gated feed-forward block rather than another scalar activation. The original gated linear unit (GLU) forms two learned projections and uses a logistic-sigmoid branch to modulate a value branch by coordinate-wise multiplication [46]. Replacing that scalar gate by ReLU, GELU or SiLU gives ReGLU, GEGLU or SwiGLU, respectively; Shazeer reported improved Transformer feed-forward results for some of these variants in the settings evaluated [47]. For an input $\mathbf{h} \in \mathbb{R}^{d_{\text{model}}}$ and hidden width d_{ff} , it computes

$$\begin{aligned} \mathbf{a} &= W_g \mathbf{h} + \mathbf{b}_g, & \mathbf{v} &= W_v \mathbf{h} + \mathbf{b}_v, \\ \text{SwiGLU}(\mathbf{h}) &= W_o [\text{SiLU}(\mathbf{a}) \odot \mathbf{v}] + \mathbf{b}_o, & \text{SiLU}(u) &= \frac{u}{1 + e^{-u}}. \end{aligned} \quad (2.57)$$

Here $W_g, W_v \in \mathbb{R}^{d_{\text{ff}} \times d_{\text{model}}}$ and $W_o \in \mathbb{R}^{d_{\text{model}} \times d_{\text{ff}}}$; SiLU and $\odot$ act coordinate-wise. The SiLU branch is an input-dependent gate on the value branch $\mathbf{v}$, and W_o

returns the result to the model width. In a pre-normalised residual block this update appears as $\mathbf{h} + \text{SwiGLU}(\text{LN}(\mathbf{h}))$. Residual connections, normalisation and gating preserve the input and output width while making a deep parameterisation easier to optimise. Dropout randomly suppresses hidden coordinates during training [48], whereas weight decay discourages unnecessarily large parameter values through the optimisation update in Equation (2.47). Dropout is disabled and weight decay is no longer applied when the fitted model is used for inference.

Development-only alternatives. A mixture of experts (MoE) routes each input through a weighted subset of jointly trained expert transformations [49]. Monte Carlo (MC) dropout instead keeps dropout active at inference and averages predictions from multiple random masks [50]; an ensemble averages separately trained models [51]. For density-matrix outputs these arithmetic averages remain physical by convexity (Section 2.1.1), but they require multiple forward passes and do not guarantee better accuracy or calibrated uncertainty. Their development results are recorded in Table A.5.

2.3.5 Symmetry-aware test-time augmentation

Test-time augmentation (TTA) is an inference-time way to reuse a known symmetry of the problem. The familiar image analogy is to predict from an image and from its horizontal reflection, undo the reflection on the second prediction, and then average the two aligned results. No new training example is created at this stage, and the model weights do not change. The same recorded input is presented twice in two equivalent coordinate systems.

For the measurement grid, let $M = [\widehat{m}_{ab}]_{a,b \in \{H,V,D,R\}}$ contain the observed frequencies and let $C = [c_{ab}]$ contain their confidences. Exchanging the first and second qubits transposes both grids: the setting ab becomes ba , so HV and VH are exchanged, for example, while HH , VV , DD and RR remain fixed. Write $\mathcal{P}M = M^\top$ for this permutation and let S be the two-qubit SWAP unitary,

$$S(|a\rangle \otimes |b\rangle) = |b\rangle \otimes |a\rangle, \quad S = S^\dagger = S^{-1}. \quad (2.58)$$

This is a genuine symmetry of the observations, not merely a convenient array operation. For $\Pi_{ab} = \mu_a \otimes \mu_b$ from Equation (2.17), cyclicity of the trace and the Born rule give

$$\begin{aligned} m_{ab}(S\rho S^\dagger) &= \text{Tr}[(\mu_a \otimes \mu_b)S\rho S^\dagger] \\ &= \text{Tr}[(\mu_b \otimes \mu_a)\rho] = m_{ba}(\rho). \end{aligned} \quad (2.59)$$

Thus the transposed grids $(\mathcal{P}M, \mathcal{P}C)$ describe the swapped state $S\rho S^\dagger$ using exactly the same measured values, shot counts and missing settings, only relabelled. The two passes are therefore correlated; this operation does not average away shot noise as a second independent experiment would.

Let $F_\theta(M, C)$ be a trained reconstructor. One pass gives the direct estimate; a second pass uses the transposed input and is then mapped back to the original qubit order:

$$\widehat{\rho}_{\text{dir}} = F_\theta(M, C), \quad \widehat{\rho}_{\text{back}} = S^\dagger F_\theta(\mathcal{P}M, \mathcal{P}C) S, \quad \widehat{\rho}_{\text{aug}} = \frac{1}{2}(\widehat{\rho}_{\text{dir}} + \widehat{\rho}_{\text{back}}). \quad (2.60)$$

If the network were exactly equivariant to qubit exchange, then $F_\theta(\mathcal{P}M, \mathcal{P}C) = SF_\theta(M, C)S^\dagger$ and the two aligned estimates in Equation (2.60) would be identical. Augmentation would then change nothing. A general neural network need not obey that identity exactly: it can make slightly different errors when the same record is presented in the two slot orderings. Averaging can cancel the part of this error that reverses under the exchange. It cannot create missing information, and improvement is not guaranteed because the two errors may be correlated or the second presentation may be less accurate.

When both passes return density matrices, their average is also a density matrix by convexity of the state space (Section 2.1.1). The implementation nevertheless applies the clipping-and-renormalisation map $\mathcal{P}_{\text{clip}}$ of Algorithm 2.1 after averaging as a floating-point safeguard. The symmetry itself holds for every number of observed settings; the development-time gate used in the reported experiment is an empirical inference choice, specified in Section 3.8.

2.3.6 Feed-forward networks and DeepSets

A multilayer perceptron (MLP) of depth L maps a fixed-size vector $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ to an output vector by alternating the operations above:

$$\begin{aligned} \mathbf{h}^{(0)} &= \mathbf{x}, \\ \mathbf{h}^{(\ell)} &= \sigma\left(W^{(\ell)}\mathbf{h}^{(\ell-1)} + \mathbf{b}^{(\ell)}\right), \quad \ell = 1, \dots, L-1, \\ f_\theta(\mathbf{x}) &= W^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}. \end{aligned} \tag{2.61}$$

Every output can depend on every input through the dense weight matrices. An ordinary MLP nevertheless treats the inputs only as coordinates in one fixed ordering. Its separate input columns allow it to learn position-specific behaviour, so the fixed order can encode projector identity implicitly. What it lacks is an explicit tagged-set structure, a mechanism that excludes absent slots, and invariance to a reordering of tagged measurements. These properties, and the density-matrix output constraint, must be supplied by the representation and output map.

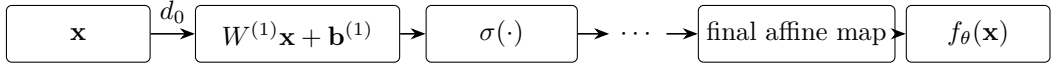
The selected feed-forward reconstructor uses the DeepSets construction [52]. Assume that at least one setting is observed, and let $O = \{p : c_p > 0\}$ be the nonempty set of observed slots. Let $\mathbf{e}_p$ be a learned embedding that identifies projector p and let combine denote the fixed implementation-specific operation that combines the gated measurement feature with this embedding. Schematically, the model computes

$$\begin{aligned} \mathbf{u}_p &= \text{combine}\left(c_p \mathbf{f}_8^{\text{harm}}(\widehat{m}_p), \mathbf{e}_p\right), \\ \mathbf{r} &= \frac{1}{|O|} \sum_{p \in O} \varphi_\theta(\mathbf{u}_p), & \widehat{\mathbf{t}} &= \psi_\theta(\mathbf{r}), \\ \widehat{\rho} &= \frac{T(\widehat{\mathbf{t}})T(\widehat{\mathbf{t}})^\dagger}{\text{Tr}[T(\widehat{\mathbf{t}})T(\widehat{\mathbf{t}})^\dagger]}. \end{aligned} \tag{2.62}$$

The same per-slot MLP φ_θ is reused for every measurement, the mean produces one fixed-size summary, and the output MLP ψ_θ emits the sixteen Cholesky coordinates. A permutation of the *presentation order* merely reorders the terms in

the sum and therefore leaves $\mathbf{r}$ unchanged. The embedding $\mathbf{e}_p$ preserves physical setting identity, including whether the record belongs to HH , HV , DD , or another projector, and changing which tagged records are absent can change the prediction. DeepSets therefore handles a variable observed subset while preserving measurement identity. All reported DeepSets conditions satisfy $|O| \geq 1$, so Equation (2.62) always pools a nonempty observed set. The masked mean makes the retained subset available through the slot embeddings, while its cardinality $|O|$ is represented only indirectly. Adding the observed setting count as an explicit scalar, or concatenating a sum with that count, would provide a direct cardinality channel and is a natural follow-up ablation for the incomplete regime.

(a) Dense feed-forward computation



(b) DeepSets reconstructor used in this thesis

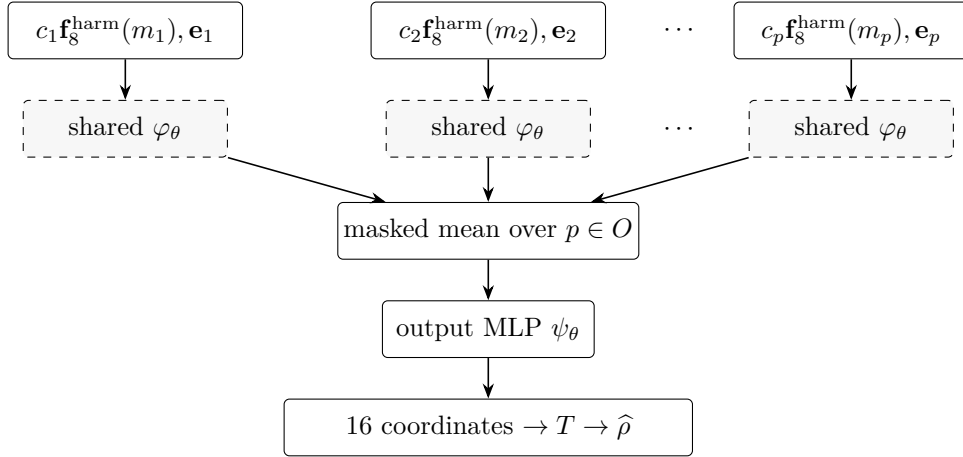


Figure 2.5: Feed-forward computation and the selected DeepSets reconstructor. An MLP alternates learned affine maps with coordinate-wise nonlinearities. DeepSets applies the same learned map to each tagged measurement slot, excludes missing slots from a masked mean, and maps the aggregate to sixteen Cholesky coordinates. Aggregation is invariant to presentation order; projector identity remains available through the slot embedding.

Direct feed-forward inference is a single evaluation of (2.62). The vanilla MLP and the selected DeepSets model are different experimental baselines: the former uses the raw concatenated vector and a flat matrix head. The latter uses the harmonic Fourier encoding shared with RhoTransformer, masked pooling and Cholesky output. Section 3.6.2 explains that comparison, and Figure A.1 records the exact selected configuration.

2.3.7 Attention and transformers

A dense layer uses mixing coefficients that are fixed after training. Scaled dot-product attention instead computes token-to-token mixing weights from the cur-

rent example [9]. Let $X \in \mathbb{R}^{N \times d_{\text{model}}}$ contain N row vectors called *tokens*. One attention head forms

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \quad (2.63)$$

where $W_Q, W_K \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_V \in \mathbb{R}^{d_{\text{model}} \times d_v}$ are learned. More generally, queries may come from N_q tokens and keys and values from N_k tokens. The scaled dot-product rule is

$$A = \text{softmax}_{\text{row}}\left(\frac{QK^\top}{\sqrt{d_k}}\right) \in \mathbb{R}^{N_q \times N_k}, \quad \text{Attention}(Q, K, V) = AV. \quad (2.64)$$

For query i this means

$$A_{ij} = \frac{\exp(\mathbf{q}_i^\top \mathbf{k}_j / \sqrt{d_k})}{\sum_{r=1}^{N_k} \exp(\mathbf{q}_i^\top \mathbf{k}_r / \sqrt{d_k})}, \quad \mathbf{o}_i = \sum_{j=1}^{N_k} A_{ij} \mathbf{v}_j. \quad (2.65)$$

Every row of A is nonnegative and sums to one, so each output is a weighted combination of value vectors. Query–key pairs with larger dot products receive larger weights. Dividing by $\sqrt{d_k}$ keeps dot-product magnitudes from growing with the head width and prevents the softmax from becoming needlessly sharp at initialisation. Informally, a query describes what an output token seeks, a key describes how an input token can be matched, and the corresponding value carries the information mixed into the output.

Multi-head attention (MHA) repeats this calculation with independently learned projections, concatenates the head outputs and applies an output projection:

$$\text{MHA}(X) = \text{Concat}(H_1, \dots, H_h)W_O. \quad (2.66)$$

Different heads can therefore learn different mixing patterns. The pre-normalised transformer form used here combines this operation with a position-wise feed-forward network (FFN) and residual paths,

$$X' = X + \text{MHA}(\text{LN}(X)), \quad X'' = X' + \text{FFN}(\text{LN}(X')). \quad (2.67)$$

Here “position-wise” means that the same FFN is applied independently to every token; interaction between tokens occurs in the attention sublayer. When Q, K, V come from the same token set, this is *self-attention*; when queries read keys and values from a separate representation, it is *cross-attention* [9].

There is an exact special case important to this thesis. For a sequence of one token, $QK^\top / \sqrt{d_k}$ is a 1×1 matrix and

$$\text{softmax}\left([\mathbf{q}^\top \mathbf{k} / \sqrt{d_k}]\right) = [1]. \quad (2.68)$$

The attention weight is independent of the query and key, and the head output is simply the value projection. After the output projection it is a dense affine map (a linear map when projection biases are omitted); no token-to-token interaction is possible. The same reasoning shows that cross-attention to a single key gives every query the same transformed value.

RhoTransformer uses precisely this distinction. Its sixteen confidence-gated measurement feature vectors are concatenated and projected to one

256-dimensional summary token. Each length-one attention sublayer is therefore implemented by its exact dense equivalent; the encoder’s remaining nonlinear computation is supplied by the SwiGLU blocks of Equation (2.57). The decoder begins with sixteen learned query tokens, one for each Cholesky coordinate. Self-attention among these sixteen queries is non-degenerate: each of the eight heads computes a full 16×16 attention-coefficient matrix from its current query and key representations. In the first decoder layer these coefficients depend only on the fixed learned queries, because self-attention occurs before measurement conditioning is added; after training, this first interaction is therefore learned but input-independent. Each layer then receives the transformed encoder summary as an additive conditioning bias. The query representations entering the second and third layers contain measurement information, so those layers can compute input-dependent attention coefficients. Each final query emits one scalar. RhoTransformer is a direct, one-pass reconstructor; the transformer in the flow-matching pipeline serves as a repeatedly evaluated velocity field.

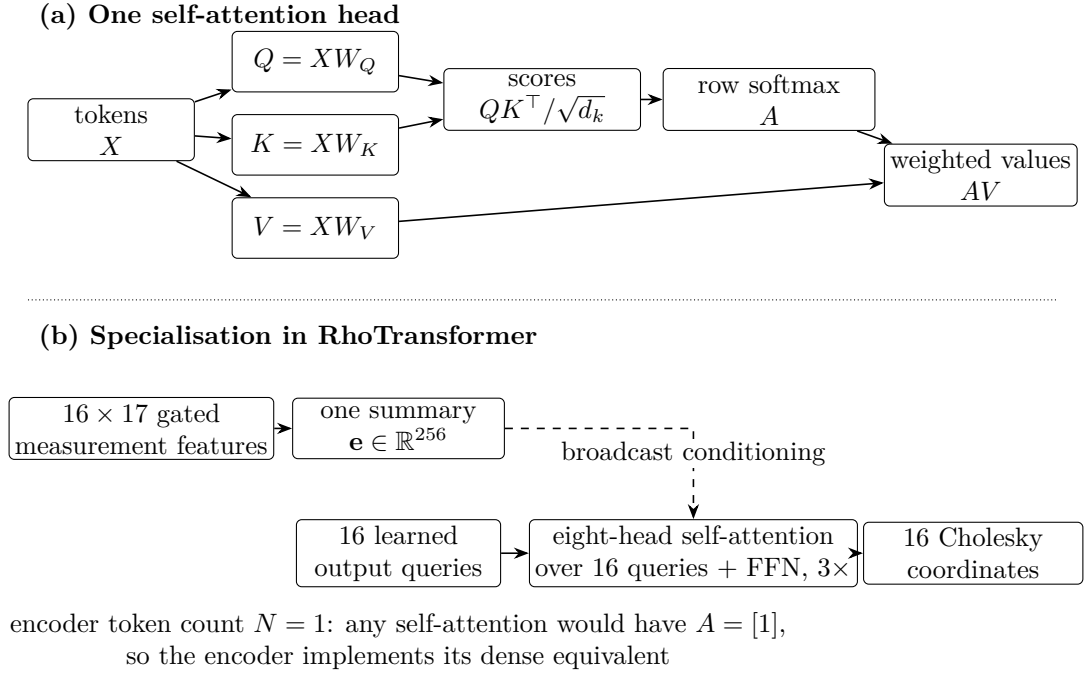


Figure 2.6: A head normalises query–key scores and uses them to mix values. RhoTransformer compresses the measurements to one encoder summary, where attention reduces to a dense map, but retains genuine eight-head self-attention among sixteen decoder queries. Later-layer coefficients can depend on the conditioned queries; the complete architecture is shown in Figure 3.4.

Attention-based QST has also been studied by Cha et al. [17]. In this thesis the transformer name appears in three distinct places: RhoTransformer is the selected direct estimator, the naive transformer is a formulation-free baseline, and a transformer parameterises one flow-matching velocity field that is evaluated repeatedly. These are three distinct training and inference procedures built with related layer types.

2.3.8 Conditional adversarial reconstruction

A generative adversarial network (GAN) trains a generator against a discriminator [53]. A conditional GAN (cGAN) supplies the same condition to both learned functions [54]. The generator $G_\theta(\mathbf{z}, \mathbf{y})$ maps a random latent vector $\mathbf{z} \sim \mathcal{N}(0, I)$ and the measurement condition $\mathbf{y}$ to a candidate state. Define the candidate's Born-probability vector by $\mathbf{b}(\rho)_p = \text{Tr}(\Pi_p \rho)$ for $p = 1, \dots, 16$. The implemented discriminator $D_\phi(\mathbf{b}(\rho), \mathbf{y}) \in (0, 1)$ is a binary classifier trained to assign a high score to the probabilities of a true state paired with its condition and a low score to those of a generated candidate paired with that same condition. With expectations over training pairs and latent samples, its discriminator loss and non-saturating generator loss are

$$\begin{aligned}\mathcal{L}_D(\phi; \theta) &= -\mathbb{E}_{\rho, \mathbf{y}} \log D_\phi(\mathbf{b}(\rho), \mathbf{y}) - \mathbb{E}_{\mathbf{z}, \mathbf{y}} \log(1 - D_\phi(\mathbf{b}(G_\theta(\mathbf{z}, \mathbf{y})), \mathbf{y})), \\ \mathcal{L}_G^{\text{adv}}(\theta; \phi) &= -\mathbb{E}_{\mathbf{z}, \mathbf{y}} \log D_\phi(\mathbf{b}(G_\theta(\mathbf{z}, \mathbf{y})), \mathbf{y}).\end{aligned}\tag{2.69}$$

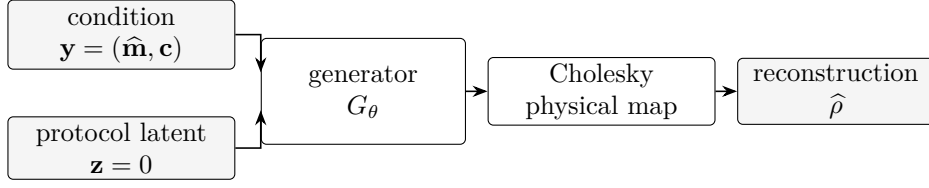
Training alternates a step that decreases $\mathcal{L}_D$ with respect to ϕ and a step that decreases $\mathcal{L}_G^{\text{adv}}$ with respect to θ . Under the idealised assumptions of unlimited capacity and successful equilibrium optimisation, the adversarial game alone matches the generated and data conditional distributions [53]. The implemented combined objective also contains the supervised term below, and its finite-model behaviour is measured empirically.

The QST generator is also supervised directly:

$$\mathcal{L}_G = \mathcal{L}_{\text{rec}}(\hat{\rho}, \rho) + \lambda_{\text{adv}} \mathcal{L}_G^{\text{adv}}.\tag{2.70}$$

The reconstruction term anchors each candidate to its labelled target, while the adversarial term asks whether the candidate has measurement statistics that resemble true states under the same input condition. In the implemented model, the discriminator sees the sixteen Born probabilities implied by a candidate rather than its matrix entries directly. During training, the discriminator supplies the adversarial signal; inference uses the generator alone. The generator accepts a random latent and can therefore define multiple candidates; every reported cGAN evaluation fixes $\mathbf{z} = 0$, selecting one fixed-latent representative for the point-estimation comparison.

(a) Reported inference path



(b) Training objectives

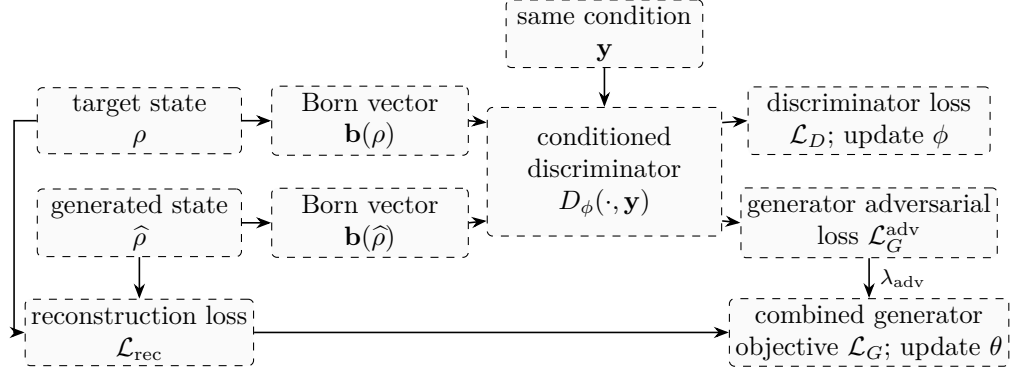


Figure 2.7: Conditional adversarial reconstruction. The solid upper panel is the complete reported inference path. Dashed boxes mark computations used only during training. In the lower panel, the real and generated Born vectors define the discriminator objective, which updates ϕ . With the discriminator fixed, the generated-state adversarial term combines with the supervised reconstruction loss to update θ .

Generative adversarial imputation nets (GAIN). GAIN adapts adversarial training to missing-data completion rather than whole-sample generation [55]. Let $\mathbf{x} \in \mathbb{R}^{16}$ be a complete measurement vector and let $\mathbf{o} \in \{0, 1\}^{16}$ be its observation mask, where $o_p = 1$ marks a known entry. From the zero-filled vector $\tilde{\mathbf{x}}$ and Gaussian noise $\mathbf{z}$, the generator produces

$$\bar{\mathbf{x}} = G_\theta(\tilde{\mathbf{x}}, \mathbf{o}, (1 - \mathbf{o}) \odot \mathbf{z}), \quad \hat{\mathbf{x}} = \mathbf{o} \odot \mathbf{x} + (1 - \mathbf{o}) \odot \bar{\mathbf{x}}. \quad (2.71)$$

Thus the completed vector preserves every observed value and inserts a generated value only where the input was missing.

The GAIN discriminator makes a coordinate-wise decision. Its output $d_p = D_{\phi,p}(\hat{\mathbf{x}}, \mathbf{h})$ estimates whether entry p was observed. The hint

$$\mathbf{h} = \mathbf{b} \odot \mathbf{o} + \frac{1}{2}(1 - \mathbf{b}) \quad (2.72)$$

reveals part of the observation mask. In the implementation used during early development, $\mathbf{b}$ is one at fifteen coordinates and zero at one randomly selected coordinate, so the value $h_p = 1/2$ identifies the single mask entry that the discriminator must infer. Writing $u_p = 1 - b_p$ and $t_p = 0.9o_p + 0.1(1 - o_p)$ for the label-smoothed discriminator target, the implemented losses are

$$\begin{aligned} \mathcal{L}_D^{\text{GAIN}} &= \mathbb{E} \left[\frac{\sum_p u_p \ell_{\text{BCE}}(d_p, t_p)}{\sum_p u_p} \right], \\ \mathcal{L}_G^{\text{GAIN}} &= -\mathbb{E} \left[\frac{\sum_p u_p (1 - o_p) \log d_p}{\sum_p u_p (1 - o_p) + \epsilon} \right] + \alpha \mathbb{E} \left[\frac{\sum_p o_p (\bar{x}_p - x_p)^2}{\sum_p o_p} \right], \end{aligned} \quad (2.73)$$

where the binary cross-entropy (BCE) is $\ell_{\text{BCE}}(d, t) = -t \log d - (1 - t) \log(1 - d)$ and the implementation uses $\epsilon = 10^{-8}$ as a numerical guard. The first generator term encourages a hidden imputation to be classified as observed; the second anchors the generator on known values. GAIN therefore assigns each slot a provenance score (experimentally recorded or generator-filled), whereas the selected cGAN asks whether an entire candidate state’s measurement vector resembles a true one under the same condition. The GAIN output d_p is a mask-origin probability, not the continuous shot-confidence c_p of Equation (2.41).

Conditional GANs have previously been applied directly to QST by Ahmed et al. [56]. Section 3.6.3 instantiates the GAIN objective above for the early measurement-imputation experiment and then describes the selected density-matrix cGAN.

2.3.9 Diffusion and flow-matching reconstruction

Diffusion and flow matching specify a path between a simple reference distribution, usually Gaussian noise, and the distribution of desired outputs. A neural network then learns how to traverse that path while being conditioned on the measurements. The path and training target define the generative paradigm; the network inside it may itself be an MLP or a transformer.

Denoising diffusion. A denoising diffusion probabilistic model (DDPM) [57] fixes cumulative signal-retention coefficients $\bar{\alpha}_t$ and samples a noisy version of a clean coordinate vector $\mathbf{x}_0$ as

$$q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0, (1 - \bar{\alpha}_t)I\right), \quad \mathbf{x}_t = \sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1 - \bar{\alpha}_t} \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I). \quad (2.74)$$

As t increases, $\bar{\alpha}_t$ decreases and the clean signal is gradually replaced by noise. Training can jump directly to any time t using (2.74); a conditional network learns to predict the sampled noise,

$$\mathcal{L}_{\text{DDPM}}(\theta) = \mathbb{E}_{\mathbf{x}_0, \mathbf{y}, t, \boldsymbol{\varepsilon}} \|\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_\theta(\mathbf{x}_t, t, \mathbf{y})\|_2^2. \quad (2.75)$$

At inference, the model starts from $\mathbf{x}_T \sim \mathcal{N}(0, I)$ and uses the predicted noise in each scheduled reverse transition until it reaches a candidate $\mathbf{x}_0$. The early DDPM in this project used standardised flat density-matrix coordinates and a cosine schedule [58]; it serves as a development baseline, while the final iterative generative representatives use flow matching.

The time t is supplied to the network through a vector embedding rather than as an unprocessed scalar. A common fixed embedding uses sine and cosine pairs at several frequencies, $[\sin(t/10000^{2k/D}), \cos(t/10000^{2k/D})]_{k=0}^{D/2-1}$, providing a smooth, multi-scale representation of the step. This is the same mathematical construction used for transformer positions [9], but here it labels noise time rather than token order.

Straight-path flow matching. The final iterative generative pipelines use straight-path flow matching [59, 60] in unstandardised Cholesky-coordinate space. Let $\mathbf{x} \in \mathbb{R}^{16}$ be the clean coordinate vector. Because the Cholesky map is many-to-one (Section 2.1.8), training requires one reproducible coordinate representative for each target state. The implementation uses the numerical factorisation

specified in Attachment A.1.8; near the rank-deficient boundary this coordinate choice can change sharply even when the represented density matrix changes little. Training samples $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, I)$ independently of $\mathbf{x}$ and $t \sim \mathcal{U}[0, 1]$. We use the implementation’s time convention throughout: data lie at $t = 0$, noise at $t = 1$, and the straight training path is

$$\mathbf{z}_t = (1 - t)\mathbf{x} + t\boldsymbol{\varepsilon}, \quad t \in [0, 1], \quad \frac{d\mathbf{z}_t}{dt} = \boldsymbol{\varepsilon} - \mathbf{x}. \quad (2.76)$$

A conditional vector-field network $\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{y})$ is trained against that velocity,

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{\mathbf{x}, \boldsymbol{\varepsilon}, t, \mathbf{y}} \|\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{y}) - (\boldsymbol{\varepsilon} - \mathbf{x})\|_2^2. \quad (2.77)$$

For each sampled training pair, exact prediction of its target displacement would give $\mathbf{x} = \mathbf{z}_t - t(\boldsymbol{\varepsilon} - \mathbf{x})$. Substituting the predicted velocity therefore defines the implied-clean estimate $\tilde{\mathbf{x}} = \mathbf{z}_t - t\mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{y})$. The population MSE optimum is the conditional mean velocity. The implemented objective uses $\tilde{\mathbf{x}}$ for an auxiliary regularised Bures term, with weight 0.25, after the Cholesky map.

Inference samples $\mathbf{z}_1 \sim \mathcal{N}(0, I)$ and solves the ordinary differential equation (ODE)

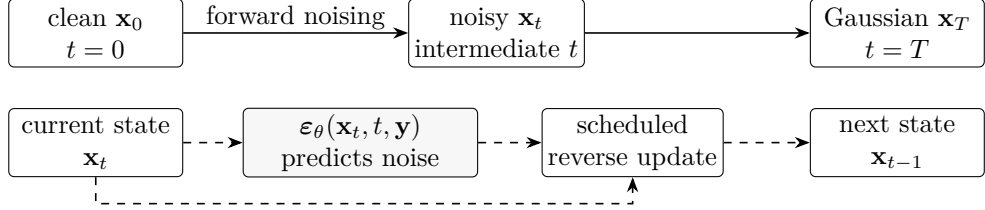
$$\frac{d\mathbf{z}}{dt} = \mathbf{v}_\theta(\mathbf{z}, t, \mathbf{y}) \quad (2.78)$$

backward from $t = 1$ to $t = 0$. The reported solver uses eight Euler steps. For a positive step size Δt , one backward update is

$$\mathbf{z}_{t-\Delta t} = \mathbf{z}_t - \Delta t \mathbf{v}_\theta(\mathbf{z}_t, t, \mathbf{y}).$$

There are therefore eight velocity-network evaluations per sample. Ten independent endpoints are mapped to density matrices and arithmetically averaged. Convexity guarantees that this arithmetic state-space mean is again a valid density matrix. Repeated sampling reduces Monte-Carlo variability and allows the model to represent several plausible endpoints, but it also makes inference much more expensive than a one-pass direct model.

(a) DDPM: discrete noise levels



(b) Flow matching: continuous straight path

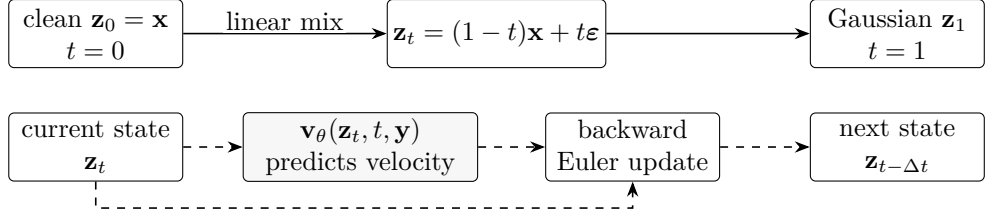


Figure 2.8: DDPM learns discrete reverse transitions through a prescribed noise schedule; straight-path flow matching learns a conditional velocity integrated from noise at $t = 1$ to data at $t = 0$. Either learned component may use an MLP or transformer. The final comparison uses MLP- and transformer-parameterised flow velocities and averages ten eight-step trajectories.

Point-estimation scope of the generative comparisons. A conditional generative model can be evaluated either through a single reconstruction or as a distribution of candidate states. The benchmark in this thesis uses the former scope: the cGAN is evaluated at the fixed latent value $\mathbf{z} = 0$, while each flow-matching reconstruction is the arithmetic mean of ten generated density matrices. The resulting comparisons therefore report the mean Bures distance of two explicitly defined point-estimation rules: a prescribed-latent cGAN output and the Euclidean centroid of ten flow samples. Conditional means or modes and Bures barycentres are distinct summaries of the generated distributions; conditional calibration and posterior coverage require a separate sample-based evaluation.

2.3.10 How the four reconstruction families differ

The terms above describe different levels of a system. MLP and transformer primarily describe how a neural function is parameterised. Adversarial training and flow matching primarily describe objectives and inference procedures, and their generator or vector field can itself use feed-forward or transformer layers. Table 2.1 keeps these levels separate and connects the theory to the evaluated representatives.

Table 2.1: Conceptual comparison of four learned reconstruction families and five selected implementations. One forward pass means one evaluation of the learned component. Direct outputs then pass through the deterministic Cholesky map; the optional 4×4 safeguard is protocol-dependent and is disabled in the final common-noise accuracy comparison. The flow entry counts every velocity-network evaluation. Exact model sizes and objectives are given in Attachment A.1.8.

Family / selected implementation	Learned function	Training signal	Reported inference
DeepSets	Direct map from a set of tagged measurements to Cholesky coordinates	Supervised reconstruction and measurement consistency	Direct point estimate; one neural forward pass
RhoTransformer	Direct map with self-attention among sixteen output queries	Same direct supervised objective	Direct point estimate; one neural forward pass
Conditional GAN	Feed-forward generator plus a training-only discriminator	Supervised reconstruction plus non-saturating adversarial loss	One generator forward pass with $\mathbf{z} = 0$
Flow matching	Conditional velocity field, parameterised by a transformer or an MLP	Velocity matching plus auxiliary Bures loss	Ten samples, eight solver steps each; 80 velocity-network evaluations

2.4 Summary

A two-qubit state has fifteen free real coordinates within a constrained physical state space. The Born rule observes it through sixteen separately sampled binary projectors, whose finite-shot variance depends on probability and shot count. Unit trace preserves identifiability when one of the four H/V settings is omitted; any other single omission leaves a true null direction. Classical and learned reconstructors differ in how they use the noise model, physical constraints and a prior over states. DeepSets and RhoTransformer learn a loss-dependent point estimate with one neural forward pass for a fixed record. Sequential acquisition can invoke that frozen reconstructor between new settings while changing only the evidence in the record; its trained selector approximates immediate conditional error reduction and is distinct from truth-selected hindsight references. The cGAN adds a training-only discriminator, and flow matching repeatedly evaluates a conditional vector field from noise. These concepts (linear observation, sampling uncertainty, identifiability, physical output structure, and the distinction between a network backbone and its training paradigm) organise the methodology and results that follow.

3. Methodology

The central task is to turn sixteen noisy, missing or near-clean measurement probabilities into one physical 4×4 density matrix. From synthetic two-qubit states we generate finite-shot observations online and evaluate four broad reconstruction families through five selected learned implementations: direct feed-forward regression (DeepSets), direct transformer regression (RhoTransformer), adversarial reconstruction (cGAN), and flow matching with either an MLP or a transformer velocity network. All five use the same base state pool and partitions and are evaluated through one paired protocol; their online training samplers remain pipeline-specific. Compatible implementations also share the seventeen-component input format, confidence gating and physical output parameterisation; their backbones, objectives and inference procedures remain family-specific.

Scope of the comparison. Development breadth differed (the direct transformer passed through the longest validation-guided sequence of single-change rounds), but every selected model was retrained to its stated stopping criterion and evaluated on the same paired test records. The resulting ranking compares complete implementations rather than isolated backbones. A separate controlled decoder ablation compares query self-attention with a larger attention-free replacement under one shared recipe (Section 4.1.2), isolating that decoder choice within the tested pair.

The chapter first defines the data, observation regimes and common reconstruction interface; it then describes the compared pipelines, model development, active measurement selection and evaluation procedure.

3.1 Data and the three regimes

3.1.1 The dataset

All headline reconstruction experiments use a fixed processed dataset of about 5.06×10^5 two-qubit states, represented by their sixteen clean James–Kwiat probabilities (Sections 2.1.4 and 2.1.5). The disjoint split contains 413 999 training, 46 000 validation and 45 999 test states. This pool predates the thesis and belongs to the same synthetic ensemble as the state data later released with Krawczyk et al.’s study on Zenodo [61]. Its generator combines random quantum circuits with Haar-measure sampling and deliberately adds maximally entangled states [62]; Baláz et al. [23] characterise its entanglement distribution and use the tomography vectors for entanglement learning. The collection spans ranks one through four and includes concurrence labels (Section 2.2.3). Figure 3.1 characterises the induced distributions in entanglement and measurement space. The public record identifies the state data, while the preserved local split fixes the exact processed benchmark.

The five headline reconstructors draw their training states from this base pool and use the same fixed validation and test populations. Their noisy and incomplete validation and test observations are generated once and stored, as specified in Section 3.1.2. The secondary-study reconstructor and measurement

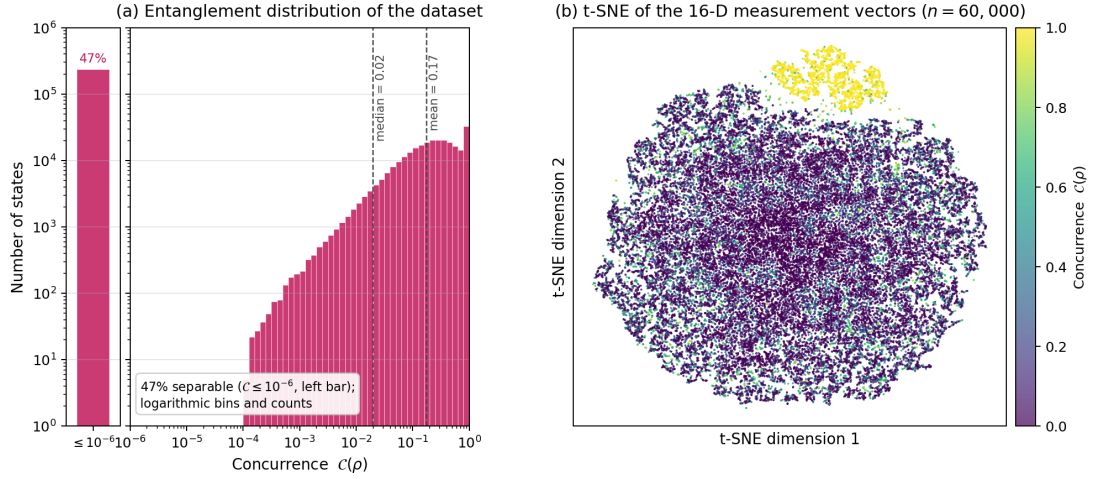


Figure 3.1: The two-qubit dataset. (a) Wootters concurrence $\mathcal{C}$ from Equation (2.39) on logarithmic axes; the separate left bar contains the $\mathcal{C} = 0$ separable states, and dashed lines mark the median and mean. (b) A two-dimensional t-SNE embedding (Section 2.3.2) of 60 000 clean Born-probability vectors, coloured by concurrence.

selector use a fourfold enlarged training pool generated by the same state procedure. Validation states support early stopping and model selection; unless a result is explicitly labelled as development evidence, reconstruction quality is reported on all 45 999 states in the fixed test split, which is excluded from fitting and formal checkpoint selection. Attachment development-history tables retain and label the original partition and protocol of each recorded run. The enlarged Hilbert–Schmidt pool samples rank uniformly from $\{1, \dots, 4\}$; besides the secondary-study platform, it appears in the development-stage data-scaling experiment of Table A.6. It does not alter the common fixed-test evaluation.

Figure 3.1 summarises how the dataset is distributed. The concurrence histogram (panel a) is strongly weighted toward weakly entangled states (about 47% are separable and the median concurrence is only 0.02, compared with a mean of 0.17), yet the entire range up to the maximally entangled states is populated, so a reconstructor must cope with both separable and strongly entangled inputs. The t-SNE embedding (panel b) shows that the two ends of this range play different geometric roles in measurement space: the near-maximally entangled states (about 4.5% of the dataset) form a compact, well-separated cluster of their own, while everything below $\mathcal{C} \approx 0.99$, separable states included, appears as one projected bulk with no clean concurrence-defined boundary. The embedding uses perplexity 40, principal component analysis (PCA) initialisation, Euclidean distance, 1 000 iterations and random seed 0. As discussed in Section 2.3.2, this projection describes local neighbourhoods rather than global metric geometry; apparent cluster spacing and area are therefore not interpreted quantitatively. Its separated island is also visible before projection: in the original measurement vectors, 99.4% of the ten nearest neighbours of a state with $\mathcal{C} > 0.99$ also satisfy $\mathcal{C} > 0.99$. At much smaller t-SNE subsamples the island fragments into strands. The full-sample view therefore serves as a qualitative map of the input distri-

bution on which all of our models are trained and evaluated: a heavy weight of separable and weakly entangled states in one bulk, plus a thin, geometrically distinctive shell of near-maximal ones.

Because the learned models are trained and tested on states from the same generator, two sanity checks on the released splits matter for interpreting every result that follows. The first concerns leakage. No Born-probability vectors are shared between the training and test splits (and only seven duplicates exist inside the 414 000-state training split), and even the single closest training–test pair is 3.3×10^{-3} apart in the sixteen-dimensional measurement space, comparable to the typical distance between neighbouring test states themselves. The test split therefore measures generalisation to new states of the same distribution.

The second check concerns the kinds of states in the benchmark. The three data splits have nearly identical spectral statistics: each has mean purity $\mathcal{P}(\rho) = 0.62$ in the notation of Equation (2.2). About one quarter of the states are numerically pure, and almost half have an eigenvalue below 10^{-6} . These nearly rank-deficient states lie close to the boundary of the physical state space (Section 2.1.1) and, as Chapter 4 shows, are the most difficult to reconstruct. Averaged over the training set, the state is close to the maximally mixed state $I_4/4$.

3.1.2 The noise model and the three operating regimes

Finite-shot noise is generated *online only for training*: every epoch pairs each clean training state with a fresh observation rather than storing one fixed noisy copy. For each of the sixteen settings we draw a shot count and then a binomial count (2.12) from the clean Born probability; a missing setting has zero shots and confidence (Section 2.3.1). Repeated sampling from $p(\text{observed} \mid \rho, N_{\text{shots}})$ exposes the network to independent corruptions, turning the $\sim 4 \times 10^5$ clean training states into an effectively unbounded stream of training pairs at little additional cost.

By contrast, the validation and test observations for each evaluation operating point are generated once, stored and reused by every headline experiment. A given model is therefore evaluated on the same states, observed settings, shot counts and binomial draws as every other model. These fixed records make the benchmark paired and reproducible; evaluation differences cannot arise merely because one pipeline received a more favourable noise or missingness realisation.

The three reconstruction tasks and abbreviations introduced in Section 1.2 are realised as operating regimes of this single noise model, and each maps onto a fixed evaluation sweep:

- **Complete noisy (CN)**: all sixteen measurements observed at a common shot budget N , swept over a logarithmic grid from 65 to 10^5 shots per setting. We call this the *uniform-shot sweep*; its design follows the uniform-shot evaluation of Macarone Palmieri et al. [13]. It isolates the effect of finite-shot noise from that of missing data.
- **Incomplete near-clean (INC)**: only $n_{\text{high}} \in \{0, \dots, 16\}$ randomly chosen measurements are observed, each at a high budget of 10^5 shots, and the rest are missing. Here n_{high} counts settings that receive the stated high per-setting budget; it is not itself a number of shots. We call this the

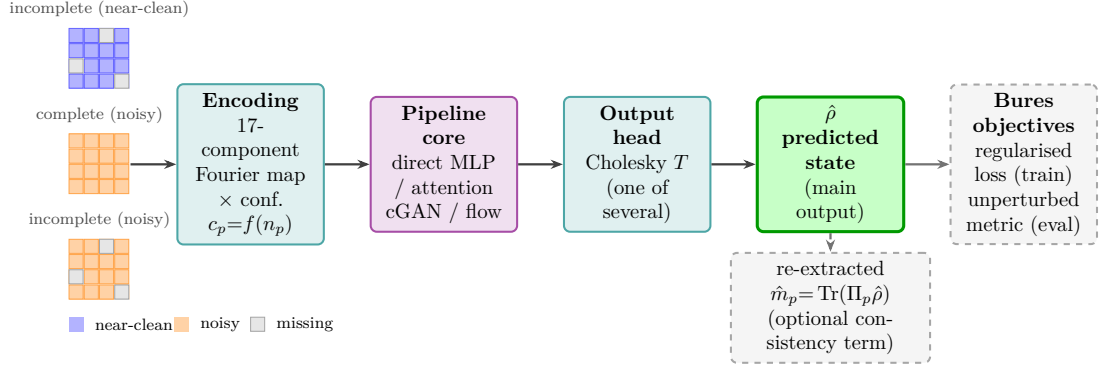


Figure 3.2: Blue, amber and grey slots are near-clean, noisy and missing, respectively. Confidence-gated Fourier features with the fixed frequency grid of each model family feed a family-specific core and physical Cholesky head. Evaluation uses Bures distance; training uses the regularised loss of Section 2.2.1.

missing-measurement sweep; at this high shot budget it isolates the cost of missingness from the cost of shot noise, and it approximates the exact-outcome incomplete setting studied by Krawczyk et al. [12] with high-shot binomial estimates.

- **Incomplete noisy (IN)**: the same missing-measurement sweep is evaluated at 10^4 shots per observed setting for the headline IN condition and at 10^3 shots per observed setting as a lower-shot stress sweep. The observed entries are therefore themselves noisy. With missing measurements *and* finite-shot noise present at once, this is the joint regime under statistical noise that is the central focus of this thesis (Chapter 1).

Each reconstructor we report is a single set of weights evaluated on the complete sweep and all three incomplete shot budgets (CN, INC and the two IN curves); the per-slot confidence channel (2.41) supplies the regime information.

The learned prior. An informationally incomplete projector subset admits multiple compatible two-qubit states, so every point-valued reconstruction needs a rule for choosing among them. The four trace-determined single omissions characterised by Claim 1 remain identifying because unit trace supplies the missing equation. For genuinely non-identifying subsets, the pseudoinverse uses a minimum-norm convention and therefore has a subset-dependent null-space bias with no universal shrinkage target (Section 2.1.7). A learned reconstructor instead uses the empirical state distribution seen during training, as formalised by the conditional-risk estimator in Equation (2.49). Its advantage on incomplete inputs therefore demonstrates the usefulness of that learned prior on the matched test distribution.

3.2 Common reconstruction interface

Figure 3.2 separates shared components from the family-specific core. Direct models return $\hat{\rho}$ in one neural forward pass, the cGAN adds a training-only dis-

criminator, and flow matching integrates a learned velocity field. Every implementation ultimately returns a density matrix, from which Born probabilities can be re-extracted for the optional consistency term of Equation (3.1).

Four common design principles organise the selected formulations, with the family-specific details stated below:

1. **Fourier input features scaled by confidence.** Each measurement m_p is represented by seventeen values: m_p itself and eight sine–cosine pairs. Rho-Transformer and DeepSets use the harmonic grid $\mathbf{f}_g^{\text{harm}}$ of Equation (2.42); the cGAN and flow models retain the dyadic grid $\mathbf{f}_g^{\text{dyad}}$ from their family-specific implementations. Both are instances of the mappings analysed by Tancik et al. [32], have the same input size and preserve the raw probability. Multiplication by the confidence c_p of Equation (2.41) makes a missing slot contribute exactly zero.
2. **Cholesky output parameterisation.** The network emits the sixteen real coordinates of the lower-triangular factor T defined in Equation (2.29); Equation (2.28) maps them to a reconstruction that is Hermitian, PSD and unit-trace *by construction* in exact arithmetic. The principal checkpoint additionally used the documented single-precision safeguard during training; its operation and measured effect are in Attachment A.1.5. An unconstrained matrix output instead required substantive repair by $\mathcal{P}_{\text{clip}}$ (Algorithm 2.1) during development.
3. **Regularised Bures training loss.** Models are trained with the numerically regularised Bures-distance loss of Section 2.2.1. PyTorch automatic differentiation [30] is used, with the spectral subcomputation promoted to complex double precision and the documented eigenvalue regularisation and symmetry-breaking perturbation. The perturbation measurably changes the loss value: on the principal reconstructor’s predictions, its mean absolute per-state shift is $(4\text{--}6) \times 10^{-3}$, with per-state maxima near 3×10^{-2} . Training therefore minimises a stochastic regularised loss, whereas reported metrics omit the perturbation entirely. This training loss must not be confused with the separate post-Cholesky physicality safeguard, whose measured effect on reported mean Bures distances is below 2.4×10^{-6} . The principal direct checkpoint enables that safeguard during training and checkpoint selection but forces it off for the final common-noise evaluation. The cGAN never applies this optional safeguard: during both training and evaluation, its predicted factor T is converted directly by $\hat{\rho} = TT^\dagger / \text{Tr}(TT^\dagger)$. This deterministic Cholesky map is already Hermitian, PSD and unit-trace for $T \neq 0$ in exact arithmetic; omitting the safeguard only removes the additional eigendecomposition, eigenvalue clipping and trace renormalisation intended to correct rare floating-point drift. This distinction concerns the output map, not the separate eigendecompositions used to evaluate the cGAN’s Bures reconstruction loss. Attachment A.1.5 quantifies the distinction.
4. **Missingness-aware online noise.** In an ordinary training row, each of the sixteen settings is independently hidden with probability 0.15. An observed setting receives a shot count drawn log-uniformly from $[1, 10^5]$, followed by

the binomial sampling of Equation (2.12). Thus an ordinary row contains $16(1 - 0.15) = 13.6$ observed settings on average, and rows with only a few observed settings occur rarely.

An *anchor row* deliberately presents a regime that this ordinary sampler would otherwise show infrequently. RhoTransformer and DeepSets replace 25% of their ordinary rows with *complete anchors*: all sixteen settings are observed and share one shot count drawn log-uniformly from $[10, 10^5]$. The cGAN and the two flow-matching models use a 15% complete-anchor switch and an independent 15% *sparse-anchor* switch. A sparse anchor retains only one to four randomly chosen settings, each at 10^4 shots. Consequently, complete inputs are deliberately represented in every pipeline, while the cGAN and flow models also receive explicit examples from the very sparse regime. Attachment A.1.8 records the exact mixtures.

3.2.1 Alternatives considered

The incremental procedure of Section 3.5 retained a candidate only when its validation improvement exceeded run-to-run training noise. The comparisons below are otherwise-matched development evidence; Chapter 4 reports final test-split results.

Input encoding. Raw measurement values were stable but less accurate at large shot counts in the cGAN ablation of Table A.4. A sweep over $K \in \{1, 2, 4, 8, 16, 32, 64\}$ frequencies (Table 3.1) shows that reconstruction quality is insensitive to the number of frequencies: a single frequency already reaches the plateau and additional frequencies leave the validation Bures distance within the run-to-run training noise. Because the sweep resolved no statistically meaningful optimum, $K = 8$ was retained for compatibility with the already-developed direct-model interface.

A matched ablation changed only how the model identifies the sixteen measurement settings: it replaced their learned identity embeddings with fixed two-dimensional sinusoidal coordinates for the rows and columns of the 4×4 measurement grid (Table 3.2). The data, architecture, training budget and validation observations were held fixed. We trained each encoding once with the same random seed, so this is a controlled comparison of those two runs rather than an estimate of variation across repeated training. The fixed encoding remained competitive when all settings were observed (0.030 versus 0.033), but it degraded the intermediate incomplete regime: at twelve observed settings, mean Bures distance increased from 0.115 to 0.514. The learned setting identities were therefore retained for reconstruction from incomplete records. Finally, supplying the confidence as an additive bias on the attention logits of Equation (2.64), rather than as a multiplicative scaling of the input features, was numerically unstable, producing NaN gradients in the eigendecomposition of the Bures loss (Section 2.2.1); the multiplicative scaling we adopted is stable and has the clean limit that a missing slot ($c_p = 0$) contributes exactly the zero vector.

Table 3.1: Matched development sweep over the frequency count K of the harmonic direct-model map in Equation (2.42) for an early pre-Cholesky transformer. Performance is flat within training noise from $K = 1$ onward; $K = 8$ was retained for interface compatibility, not as a resolved optimum. Columns report aggregate validation, CN at 10^5 shots per setting and IN with twelve settings at 10^4 shots per observed setting. Bold identifies the retained configuration.

K	Validation Bures	CN (10^5)	IN ($n_{\text{high}}=12$)
1	0.198	0.049	0.131
2	0.196	0.050	0.128
4	0.198	0.050	0.130
8	0.197	0.049	0.129
16	0.196	0.050	0.129
32	0.197	0.051	0.131
64	0.198	0.051	0.130

Table 3.2: Matched comparison on all 46 000 validation states at 10^4 shots per observed setting. Each encoding was trained once with the same random seed. Data, architecture, the 250-epoch budget, subsets and noise draws were fixed; the second row replaces learned setting identities by fixed 2-D sinusoidal row/column coordinates. Lower is better.

Setting encoding	$n = 4$	$n = 8$	$n = 12$	$n = 14$	$n = 16$
Learned setting identity	0.502	0.238	0.115	0.076	0.033
Fixed 2-D row/column sinusoids	0.516	0.338	0.514	0.470	0.030

Output parameterisation. The obvious alternative to the Cholesky factorisation is to regress the sixteen real entries of ρ directly and to restore physicality afterwards by applying the eigenvalue-clipping-and-renormalisation core $\mathcal{P}_{\text{clip}}$ of Algorithm 2.1. This works (indeed $\mathcal{P}_{\text{clip}}$ was one of the single largest improvements found during development), but it requires an eigendecomposition on every forward pass and leaves the network free to emit unphysical matrices that $\mathcal{P}_{\text{clip}}$ must then repair. Directly predicting the fifteen free Pauli coefficients r_μ , $\mu = 1, \dots, 15$, defined in Equation (2.9) yielded similar aggregate validation error but higher validation error at the high-shot operating point. A *corrector* parameterisation (adding a learned residual to a linear-inversion warm-start, the paradigm of Macarone Palmieri et al. [13] and Krawczyk et al. [12]) reached its strongest relative accuracy with complete measurements and a large shot count. On incomplete inputs, the pseudoinverse backbone instead chooses the minimum-norm coefficient vector and zeros its null-space component, imposing a bias that depends on the observed subset. After trace normalisation and application of the $\mathcal{P}_{\text{clip}}$ core, that bias has no universal direction, such as a pull toward $I/4$ (Section 2.1.7); the corrector had higher error in the incomplete regime. Section 3.4 returns to this development waypoint. The retained Cholesky map is instead Hermitian, positive semidefinite and unit trace *by construction*, requiring neither substantive repair by $\mathcal{P}_{\text{clip}}$ nor a classical backbone. Attachment A.1.5 analyses the optional training-time safeguard. Final common-noise evaluation uses the

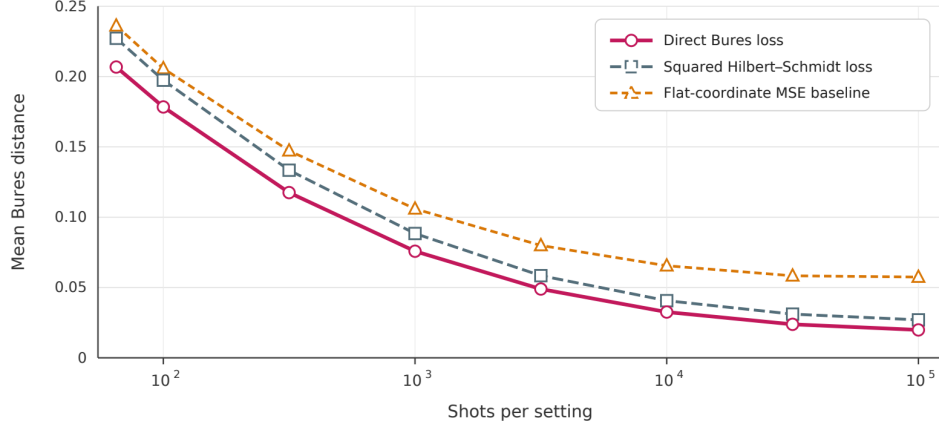


Figure 3.3: Mean Bures distance on complete uniform-shot validation inputs. Direct Bures and state-space Hilbert–Schmidt squared are a matched RhoTransformer ablation; flat-coordinate MSE is the earlier flat-output transformer baseline under the same evaluation protocol.

deterministic Cholesky output for direct models and cGAN, while flow retains its documented endpoint projection (Attachment A.1.5).

Training loss. We compared direct Bures optimisation with two cheaper Euclidean objectives: state-space squared Hilbert–Schmidt distance (Section 2.2.2) and flat-coordinate mean squared error (MSE). Macarone Palmieri et al. [13] use Cholesky-coordinate MSE and show that it upper-bounds squared Bures distance under the conditions stated in Equation (2.38). The bound is not an equality and does not imply identical training behaviour. In the matched RhoTransformer loss ablation of Figure 3.3, direct Bures training achieved lower validation Bures error than state-space Hilbert–Schmidt-squared training at every displayed shot count; the earlier flat-coordinate MSE baseline was higher still on the same uniform-shot evaluation protocol. We therefore retained direct Bures optimisation.

We also tested the *squared* Bures distance d_B^2 . It removes the final scalar square root, but $\nabla_\theta d_B^2 = 2d_B \nabla_\theta d_B$ (2.36), so it reduces the gradient scale near an accurate prediction and changes the balance with the measurement-consistency term. Keeping $w = 50$ was therefore not a matched comparison. A follow-up used $w = 18$ to approximately restore the observed gradient balance; its development-set validation Bures distance was 0.180, compared with 0.177 for the corresponding unsquared run, and it did not improve the complete high-shot points. We therefore retained d_B . The inner spectral regularisation of Section 2.2.1 is required for either form.

Reported metrics omit the training-only Bures perturbation; Attachment A.1.5 quantifies its effect. The direct objective also contains the confidence-weighted measurement-consistency term

$$\mathcal{L}_{\text{meas}}(\theta) = \frac{w}{16} \sum_{p=1}^{16} c_p \left(\text{Tr}(\Pi_p \hat{\rho}) - m_p^{\text{obs}} \right)^2, \quad (3.1)$$

where m_p^{obs} is the noisy observed probability and $c_p = 0$ automatically excludes a missing setting. This is a heuristic reliability-modulated residual rather than a

Table 3.3: Formulation components in the reported models. “Fourier” includes confidence scaling; “Chol” denotes the Cholesky head; “Bures” the regularised training loss; “miss-noise” the use of online missingness and finite-shot noise; and “plateau” validation-plateau training. The detailed sampling mixtures and stopping schedules are pipeline-specific and are reported in Attachment A.1.8.

Model	Fourier	Chol	Bures	miss-noise	plateau	objective
Vanilla MLP	–	–	–	yes	yes ^a	MSE
Naive transformer	–	–	–	yes	yes ^a	MSE
DeepSets MLP	yes	yes	yes	yes	yes	Bures + meas.
RhoTransformer	yes	yes	yes	yes	yes	Bures + meas.
cGAN (adversarial)	yes	yes	yes	yes	yes	Bures + adversarial
Flow matching (MLP)	yes	yes	yes	yes	yes	flow-matching + aux. Bures
Flow matching (transformer)	yes	yes	yes	yes	yes	flow-matching + aux. Bures

^a the formulation-free baselines converge within a few hundred epochs; their plateau is reached well inside the shared budget.

binomial likelihood. The principal direct models use $w = 50$ on every observed setting. The matched secondary study of Section 4.7 compares that schedule with $w = 0$ and with $w = 50$ only for settings having at least 10^4 shots.

Training data and noise. In a matched early-development pair, adding 25% complete uniform-shot anchors reduced the 10^5 -shot validation Bures distance from 0.0488 to 0.0266 (a 45% reduction), while the mixed validation mean remained near 0.20. The adopted mixture is specified above and in Attachment A.1.4; the paired values are retained in Table A.5. A roughly fourfold enlarged clean pool improved one development variant and later supplies the frozen secondary-study reconstructor and selector; the five headline pipelines retain the base pool. Architectural and inference-time alternatives are discussed with the model development sequence below and quantified in Attachment A.2.

3.3 What is held fixed

All headline learned implementations share the base state pool and partitions and are evaluated on paired observations. Compatible models also share the confidence encoding and Cholesky head; training samplers, objectives, stopping schedules and inference remain pipeline-specific. Table 3.3 records which formulation components are present in the two formulation-free baselines and the five selected implementations of the four learned families. Flow matching contributes two implementations because both its MLP and transformer velocity networks enter the final comparison.

3.4 Compared pipelines and baselines

The four learned families differ at more than one level:

- **Direct feed-forward regression** uses a DeepSets backbone, the regularised Bures and measurement-consistency objectives, and one neural forward pass at inference.
- **Direct transformer regression** uses the RhoTransformer backbone with decoder query self-attention, the same direct objectives, and one neural forward pass at inference.
- **Adversarial reconstruction** uses a feed-forward generator and a discriminator, adds an adversarial objective during training, and evaluates one neural generator forward pass with the latent fixed to zero.
- **Flow-matching reconstruction** uses either a feed-forward or transformer velocity network, trains a velocity-matching objective with an auxiliary Bures term, and performs iterative inference. The reported estimate averages ten eight-step trajectories, requiring about eighty velocity-network evaluations.

The corresponding mechanisms are developed in Sections 2.3.6, 2.3.7, 2.3.8 and 2.3.9. These four families yield five selected implementations because flow matching is represented by both MLP and transformer velocity networks. “Transformer” therefore names a backbone in the direct RhoTransformer and in flow matching, whereas “adversarial” and “flow matching” primarily name objectives and inference procedures.

For reporting, models progress from references to the final benchmark:

Classical baselines. A linear/pseudoinverse estimate followed by either $\mathcal{P}_{\text{clip}}$ or eigenvalue-simplex projection (Section 2.1.7), and the 250-step Cholesky optimisation of the binomial likelihood objective (Section 2.1.9); these procedures require no training.

Naive learned baselines. A *vanilla MLP* and a *naive transformer* that receive only the raw measurement/confidence pair as input, emit a flat density matrix followed by the $\mathcal{P}_{\text{clip}}$ core, and are trained with plain mean-squared error. They share the benchmark data and noise and represent the raw-input, flat-output formulation before the shared components are introduced.

Pipeline development. Compatible formulation components are added and each family is developed incrementally (Section 3.5).

Final comparison. The five selected implementations, all trained to a validation plateau, are evaluated in the three regimes with the metrics of Section 3.8.

A note on the corrector paradigm and prior work. The *corrector* models of Macarone Palmieri et al. [13] and Krawczyk et al. [12] learn a residual over a classical MLE or linear estimate. Our development version used a transformer, a fixed M-matrix linear-inversion warm-start and a learned scaled residual. It was strong for complete high-shot inputs, but on incomplete inputs it inherited the pseudoinverse’s subset-dependent minimum-norm/null-space bias and the subsequent subset-dependent effect of $\mathcal{P}_{\text{clip}}$ (Section 2.1.7), yielding higher error. We therefore retain it as related work and a development waypoint; the final benchmark uses direct and generative implementations without a fixed classical reconstruction backbone.

3.5 Model development procedure

Each family was developed incrementally under the same decision rule, although the number of trials differed:

1. **Start minimal.** Begin from the smallest viable implementation of the pipeline: a shallow stack, a single attention layer or a minimal denoiser.
2. **Evaluate one change at a time.** Test a hypothesised improvement under an otherwise identical schedule and retain it only if validation Bures changes by more than the observed 1–3% run-to-run noise. Retained changes include Cholesky output (removing a substantive output-stage physicality repair), confidence scaling (supporting one model across shot regimes), transformer query attention and missingness-aware flow training (restoring incomplete-data performance). Several capacity increases and transformer encoder attention did not meet this rule.
3. **Remove redundant components.** Once additions cease helping, remove components individually while validation Bures remains within the noise floor, retaining the smaller implementation.

For the direct transformer, only decoder query self-attention remains non-degenerate; by Equation (2.68), attention on the single encoder token is exactly a dense map. Section 4.1.2 compares the retained decoder with an attention-free alternative under a shared recipe.

3.6 Selected reconstruction pipelines

This section specifies the five selected implementations across four families. Attachment A.1 gives every final configuration, while Attachment A.2 records the longer development trajectory and alternatives that were not retained.

3.6.1 Direct transformer regression

Selected design. RhoTransformer combines confidence-gated harmonic Fourier features, a single 256-dimensional summary token, sixteen learned output queries with self-attention, SwiGLU feed-forward blocks and a normalised Cholesky output map. The latter enforces positive semidefiniteness and unit

trace by construction (Section 2.1.8). Flat-matrix, corrector, Pauli-coordinate, mixture-of-experts and multi-pass inference alternatives are reported together in Table A.5; the selected pipeline retains one physical output head and one neural forward pass.

Single-token reduction. The encoder output was collapsed to one summary token; cross-attention from sixteen queries to its single key has weight exactly one and is therefore query-independent. Collapsing the encoder input to one token similarly makes self-attention a dense linear map. The active path implements these identities as dense maps and broadcast additive conditioning. Historical checkpoint objects retain bypassed single-key cross-attention tensors; Attachment A.1.1 reconciles stored and active parameter counts.

The resulting *RhoTransformer* (Figure 3.4) has 3 980 545 stored parameters and about 3.69 million in the active forward path, roughly 30% fewer stored parameters than the grown pre-trim model, and is statistically indistinguishable from that model while being marginally better at high shot counts. Replacing its single-token encoder stack by a permutation-invariant DeepSets [52] or a plain MLP gave higher aggregate and high-shot Bures distances, demonstrating the contribution of feed-forward depth in the single-token encoder stack. Its remaining non-degenerate attention is three layers of sixteen-query decoder self-attention. The controlled comparison with an attention-free decoder (Section 4.1.2) shows its high-shot advantage under the shared recipe.

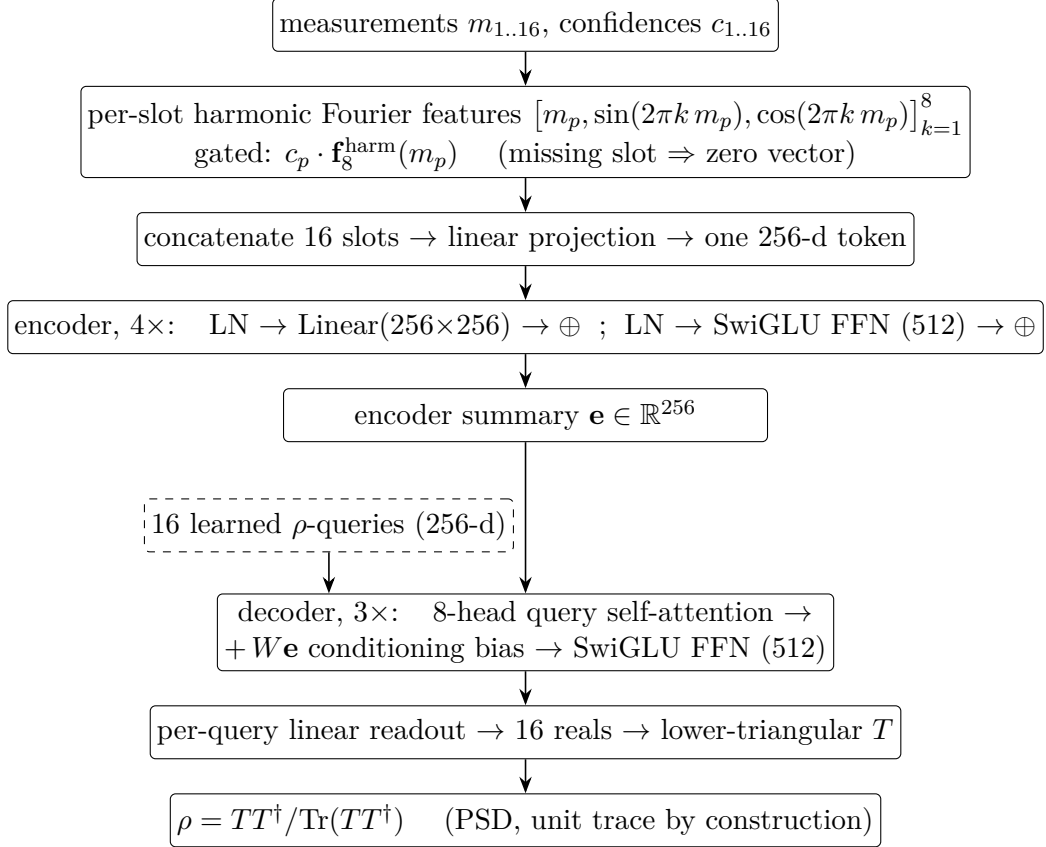


Figure 3.4: RhoTransformer data flow (3.98 million stored parameters; 3.69 million active). Only decoder query self-attention is non-degenerate; single-token encoder attention reduces to a dense map. Attachment A.1.1 gives the layer specification.

3.6.2 Direct feed-forward regression

The feed-forward family contains a permutation-invariant *DeepSets* (per-slot MLP, slot-identity embeddings and masked mean pool as in Equation (2.62); Figure A.1) and a plain concatenated-input MLP. With the full formulation *DeepSets* is stronger, especially at high shots. The formulation-free *vanilla MLP* instead uses raw inputs, a flat head and MSE, so their comparison measures the combined formulation and feed-forward redesign. For this naive model, nonzero Adam weight decay (Equation (2.47)) overwhelmed its very small MSE gradients and caused the model to converge to the constant maximally mixed state; disabling weight decay prevented that behaviour. Attachment A.1.8 gives both configurations.

3.6.3 Adversarial reconstruction

Adversarial development began with a GAIN measurement-imputation precursor using its mask-and-hint mechanism [55]; a second version added entanglement-feature matching. Equation (2.73) defines the objective and Attachment A.2 records both development variants. They completed missing measurement values, whereas the final three-regime task also requires complete-data denoising and a physical density matrix. The selected cGAN therefore shares the common interface (Figure A.2): its generator emits a Cholesky factor and its discriminator reads the reconstructed state’s measurements. The generator instantiates Equation (2.70) with $\mathcal{L}_{\text{rec}} = \mathcal{L}_{\text{Bures}}$ and $\lambda_{\text{adv}} = 0.05$; $\mathcal{L}_G^{\text{adv}}$ is the standard non-saturating objective in Equation (2.69). Binary cross-entropy trains the discriminator, conditioned on the noisy measurements and their confidence, to distinguish measurements implied by true and generated states. Under the same $z = 0$ reporting rule, a short-run reconstruction-only control already had lower validation Bures distance than its corresponding adversarial run. Training samples random latents; reported point estimation sets $z = 0$, so latent-averaged summaries remain a separate protocol. The adversarial validation-plateau model supplies the selected cGAN row. A separately continued reconstruction-only path set $\lambda_{\text{adv}} = 0$ and improved all four reported operating points (Section 4.1.1). Because the two paths have different optimisation histories, this result ranks their implemented $z = 0$ point estimators rather than isolating the adversarial term. Attachment A.1.8 gives the configuration.

3.6.4 Flow-matching reconstruction

Iterative generative development began with a DDPM in flat-density coordinates and later paired it with a transformer denoiser inspired by Peebles and Xie [63]. Its validation plateau and 200 denoiser evaluations per estimate motivated the move to rectified flow; Attachment A.2 reports the sweep. The two selected iterative implementations are instead *flow-matching* models: they predict the velocity along the rectified path of Equation (2.76) with the objective in Equation (2.77), in unstandardised sixteen-dimensional Cholesky space, and map the endpoint to a physical density matrix; missingness-aware training supplied their incomplete-data performance.

One selected flow uses an MLP velocity network and the other a transformer velocity network. The latter is named descriptively because it uses a transformer backbone for the velocity field rather than the image-patch DiT construction. The transformer flow approached the direct transformer’s accuracy. Both integrate eight steps and average ten independently generated density matrices, for about eighty velocity-network evaluations per reconstruction. The physical arithmetic mean remains a density matrix by the convexity of the state space (Section 2.1.1) and is the reported Euclidean-centroid point estimate; a different sample count or Bures barycenter would define another protocol. Thus $S = 10$ fixes the accuracy–cost operating point. Figure A.3 shows the shared flow pipeline, and Attachment A.1.8 specifies both velocity networks.

3.7 Sequential measurement-setting selection

In the incomplete-measurement regime an experimentalist may have a fixed budget of $b \leq 16$ *distinct settings* and be free to choose which projectors to acquire. This differs from allocating a fixed pool of shots among settings. In the present study, every selected setting receives $N = 10^4$ shots exactly once, so a budget of b settings uses bN freshly prepared copies. No policy revisits an observed slot or adds shots to it.

The adaptive policy implements the closed loop introduced in Section 2.3.3. It begins with an empty record and then repeats four operations:

1. run the frozen reconstructor on the partial record;
2. use the selector to score only the settings not yet observed;
3. acquire the highest-scoring setting with N binary shots; and
4. append its empirical frequency and confidence to the record.

The externally specified setting budget stops the loop, and the reconstructor’s estimate after the final acquisition is evaluated. Both networks retain fixed weights throughout. A human experimentalist can carry out the recommended measurements one by one and return their counts, or a laboratory controller can automate the same steps; the synthetic experiment simulates this measurement–reconstruction–selection loop. The fixed-order and random policies use the same setting and shot budgets and the same frozen reconstructor, thereby isolating the effect of which data are acquired.

The selector emits a score for every still-unobserved setting, so its output can be viewed as a current ranking. The evaluated policy nevertheless acquires only the top-ranked setting before scoring again. Its observed frequency may change the reconstruction and therefore the ranking of the settings that remain. Selecting several settings from one ranking without these intermediate outcomes would be a batch-design policy, which is not evaluated here.

This one-setting-at-a-time outer loop is closely related to the recurrent selector–reconstructor design of Krawczyk et al. [12]: their selector proposes an operator, its outcome is acquired, and their long short-term memory (LSTM) reconstructor updates the estimate after each prefix. The reported predefined-basis selector converges to a fixed learned ordering, whereas their adjusted-basis

variant may follow different paths and propose projectors outside the James–Kwiat grid. Our adaptive policy instead remains on the fixed sixteen-setting grid, conditions each decision on the finite-shot record and current reconstruction, and selects for a separately frozen RhoTransformer.

3.7.1 Adaptive selector

The adaptive selector is a small trained network that proposes, given the settings recorded so far, which previously unobserved setting to acquire next. Let $\mathbf{m}_{\text{partial}} \in [0, 1]^{16}$ be the current partial measurement vector, with entries set to 0.5 for unobserved slots, let $\mathbf{c} \in [0, 1]^{16}$ be the per-slot confidence (2.41), which is 0 for unobserved slots, and let $\mathbf{m}_{\text{pred}} \in [0, 1]^{16}$ be the Born probabilities (2.11) implied by the reconstructor’s current estimate $\hat{\rho}$. The selector $g_{\theta} : \mathbb{R}^{4 \times 16} \rightarrow \mathbb{R}^{16}$ consumes the concatenation

$$\left[\mathbf{m}_{\text{partial}}, \mathbf{c}, \mathbf{m}_{\text{pred}}, \mathbf{m}_{\text{pred}} - \mathbf{m}_{\text{partial}} \right] \quad (3.2)$$

and emits a vector of sixteen slot scores; the next slot to acquire is the one with the highest score among the slots not yet observed. All history information supplied to the selector is contained in these input channels: which slots have been observed, their frequencies and their shot-derived confidences. For this without-replacement protocol on independently prepared copies, the order in which the same partial record was acquired is not a separate input. A feed-forward network therefore suffices. The selector is a four-layer MLP with hidden dimension 384, GELU activations and layer normalisation, with approximately 478 000 parameters. The reconstructor’s weights are held fixed throughout selector training.

The training target is constructed offline. For each training state ρ and each random partial subset $S \subset \{0, \dots, 15\}$ of size $|S| \in \{0, \dots, 15\}$, we compute the per-slot Bures gain that would be obtained by additionally measuring slot $s \notin S$,

$$g_s = d_B(\hat{\rho}_S, \rho) - d_B(\hat{\rho}_{S \cup \{s\}}, \rho), \quad (3.3)$$

simulating one binomial-noise outcome for the candidate slot. This is a one-sample Monte Carlo target for the conditional utility in Equation (2.52), with the synthetic training state supplying the otherwise unknown ρ . Several slots typically have near-equal gain, so a target that names only the single best slot is noisy; we therefore train against the full gain vector with a temperature-scaled Kullback–Leibler (KL) objective, in the spirit of policy distillation,

$$\mathcal{L}(\theta) = \text{KL}\left(\text{softmax}(\mathbf{g}/\tau) \parallel \text{softmax}(g_{\theta})\right), \quad \tau = 0.05. \quad (3.4)$$

Each candidate gain uses one finite-shot draw and is therefore a noisy one-sample Monte Carlo target for the expected gain. Averaging multiple draws per candidate would reduce this target variance. Because every candidate is a different setting, drawing one outcome for each candidate is distributionally equivalent to pre-drawing one complete sixteen-slot record and revealing the corresponding entry; the present implementation relies on the full soft target and the large number of training examples rather than repeated outcomes. No multi-step return or policy-gradient objective is used. The policy becomes history-dependent by recomputing this myopic ranking after every acquired outcome. An otherwise

identical cross-entropy variant trained against $\arg \max_s g_s$ converges faster on a top-1-accuracy metric but yields measurably higher downstream Bures distance, and we therefore use (3.4) as the default. The known ρ is needed only to form these synthetic teacher gains; the trained policy never receives it. Training forms 600 000 synthetic teacher examples by drawing states with replacement from the enlarged secondary-study training pool; partial subsets and shot noise are resampled on each of 25 epochs. It applies AdamW with a cosine-annealed learning rate and global ℓ_2 gradient-norm clipping at 1.0 as in Equation (2.48), for a total wall-clock cost of approximately 40 minutes on a single GPU.

3.7.2 Fixed orderings distilled from the selector and from marginal gains

From the adaptive selector we distil a single *state-independent* ordering of the sixteen measurements by taking its modal choice at each step over 2 000 validation states (seed 42). The result, denoted π^* , is a fixed, precomputed measurement schedule: an experimentalist can follow it directly, with no trained network deployed during data taking. The ordering itself, and the physical consistency checks it passes, are presented with the results (Section 4.6.2).

A complementary state-independent ordering, denoted π_{MG} , ranks slots by estimated marginal Bures gain. For each of 2 000 validation states we draw eight random subset sizes, subsets and candidate slots (seed 42), measure the candidate once at 10^4 shots, and average the non-negative Bures reduction for each slot. This Monte Carlo marginal-gain heuristic samples random subsets directly and averages the candidate’s measured contribution. It provides a useful state-independent ranking that can be compared with the selector-distilled ordering (Section 4.6.2). A Shapley estimator would instead weight complete random permutations [64].

3.7.3 Hindsight references and common-noise headroom

To characterise the spread produced jointly by subset choice and finite-shot noise, we also compute a *noisy hindsight reference*, parameterised by the size K of a candidate pool. For each evaluation state ρ and each setting budget b we draw K candidate subsets $A_1, \dots, A_K$ of the sixteen measurements, each consisting of b measurements chosen uniformly at random without replacement, run the same fixed reconstructor on each of the K resulting measurement vectors, each with its own independent binomial noise realisation ξ_i , and keep the single reconstruction closest to the *true* state:

$$\bar{d}_{\text{hind}}(b; K) = \mathbb{E}_{\rho} \left[\min_{1 \leq i \leq K} d_B(\hat{\rho}(A_i, \xi_i), \rho) \right], \quad (3.5)$$

where the expectation is the empirical mean over evaluation states. Because K lies inside a per-state minimum, the reference can only decrease as the pool grows. We report $K = 1000$ and evaluate checkpoints at $K \in \{20, 50, 100, 200, 300, 500, 700, 1000\}$ on the full test split (Figure 3.5). It falls by roughly 5–10% per doubling without saturation; at $b = 12$, for example, it decreases from 0.048 to 0.029. This is a noncausal combined

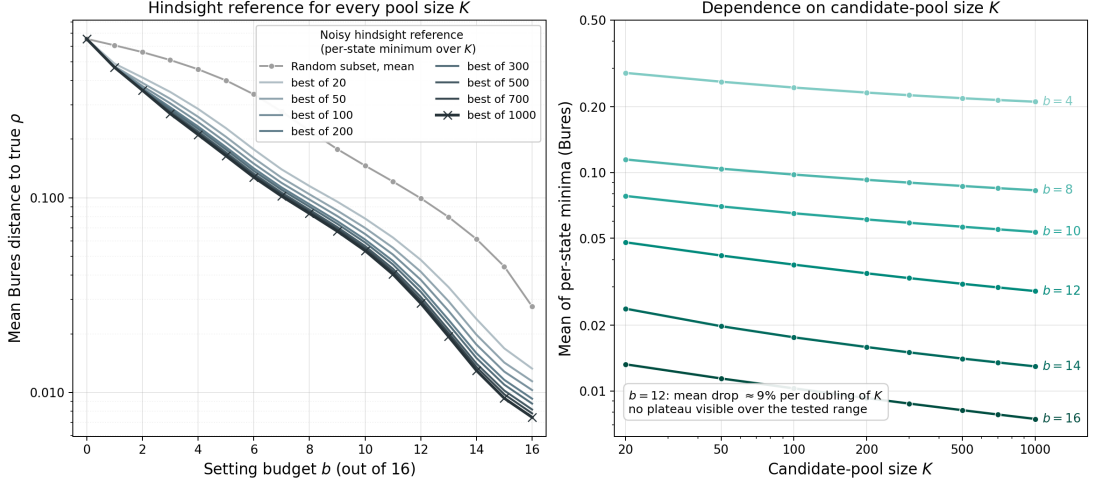


Figure 3.5: Equation (3.5) on the full test split at 10^4 shots per observed setting. Left: Bures distance by setting budget and pool size. Right: the same per-state minima versus K . In the left panel, darker curves use larger pools. At $b = 16$ every candidate contains the same complete setting set, so the remaining decrease with K comes entirely from truth-based selection among independent noise realisations, not from setting selection.

subset-and-noise reference: it uses the true state and gives every candidate an independent noise draw. At $b = 16$ the subset is unique, yet the value still falls from 0.013 to 0.007, isolating retrospective selection among favourable noise realisations.

For setting-selection headroom references, we remove the noise lottery by drawing one complete sixteen-slot binomial record $\xi = (\xi_1, \dots, \xi_{16})$ per test state,

$$\xi_s = C_s/N, \quad C_s \sim \text{Binomial}(N, m_s), \quad s = 1, \dots, 16, \quad (3.6)$$

with the slot outcomes conditionally independent given the state. Every candidate subset reveals entries from that same record. This pre-draw is the potential-outcome construction of Equation (2.53); it does not give a deployable policy access to unselected values. For budget b and a nested pool of K distinct subsets, the *common-noise subset hindsight* is

$$\bar{d}_{\text{sub}}(b; K) = \mathbb{E}_{\rho, \xi, A_{1:K}} \left[\min_{1 \leq i \leq \min\{K, \binom{16}{b}\}} d_B(\hat{\rho}(A_i, \xi), \rho) \right]. \quad (3.7)$$

We report $K \in \{5, 20, 50, 100, 200, 500, 1000\}$. For each state and budget, one random ordering of the subset space makes these pools nested; when K exceeds $\binom{16}{b}$, the pool is exhaustive. The true state still chooses the minimum, so this is noncausal and central-budget values remain candidate-pool references rather than exhaustive optima. It does, however, isolate subset choice under one fixed noisy record. In particular, there is only one subset at $b = 16$, so every K gives the ordinary complete-record result exactly and the setting-selection headroom is zero. Nesting applies across K at a fixed budget, not across budgets: the size- b

and size- $(b+1)$ minimisers are chosen separately and need not satisfy $A_b \subset A_{b+1}$. The resulting hindsight curve is therefore not a sequential trajectory and need not be monotone in b . Common noise also refers to coupling the counterfactual candidates inside this reference. The separately evaluated deployable-policy curves use the same test split and marginal binomial protocol, but their means are not paired state-wise differences from this independently generated reference record.

This global reference is complemented by a local test of the learned policy. Let s_1 be the selector’s first acquisition from the empty record. After revealing the common-record value ξ_{s_1} , we evaluate each of the fifteen distinct possible second actions $s \neq s_1$ and compare the selector’s choice s_2 with nested uniformly permuted pools of size $K \in \{1, 2, 5, 10, 15\}$. The $K = 15$ endpoint is the exhaustive one-step hindsight reference. Here K counts alternative actions considered for the same second decision; it is not the number of sequentially acquired settings. The trained selector and the ordinary random policy each make one second choice, so only $K = 1$ is a direct policy comparison. Values with $K > 1$ quantify increasingly privileged truth-selected hindsight and are retained as numerical diagnostics rather than plotted as policy curves. There can be no valid $K > 15$ one-step curve: it would have to repeat a setting or redraw its noise. Per-state minima are always formed before averaging. For both common-noise experiments, reported mean intervals use $\bar{x} \pm z_{0.975}s/\sqrt{n}$ over the 45 999 test states; comparisons of the selector’s second action with random search use the corresponding paired per-state differences under the shared record. Figure 4.9 uses the $K = 1000$ global endpoint, Figure 4.10 shows the global common-noise diagnostic, and Figure 3.5 separately retains independent-noise hindsight to expose joint subset-and-noise variability. The local second-action audit is reported numerically in Section 4.6.1.

3.8 Evaluation procedure

Metrics. Reconstruction quality is reported primarily as the mean Bures distance (Section 2.2.1) over the evaluation set, complemented by the Hilbert–Schmidt distance (Section 2.2.2) and three physical or task-oriented analyses. First, Wootters concurrence (Section 2.2.3) is computed directly from the reconstructed and true density matrices. Second, a frozen, independently trained entanglement classifier is applied to measurements implied by each reconstruction. Third, we analyse changes in the implied measurement probabilities on complete fixed test records. The LOW band uses 100 shots per setting and the HIGH band uses 10^4 shots per setting. For each model and band we compute the root-mean-square (RMS) value of $\text{Tr}(\Pi_p \hat{\rho}) - m_p^{\text{obs}}$ over states and settings. This quantifies how strongly a model adjusts each confidence band; because it measures magnitude relative to the noisy input, it does not by itself establish whether the adjustment points toward the clean Born probability. The synthetic benchmark also retains that clean probability, so we report the RMS errors of both the noisy input and the reconstructed probability relative to it. Their comparison directly tests whether the model denoises the observed probabilities in each shot band.

Common evaluation conditions. The learned models and linear inversion are evaluated on the same fixed 45 999-state test split. Their paired reruns use the same noise seed, batch boundaries, observed subsets, shot counts and binomial draws so differences reflect the method rather than the input realisation. The 250-step Cholesky likelihood optimisation uses the same paired inputs across all test states. It reports the final iterate after 250 Adam steps from the documented linear-inversion warm start, with a random fallback if factorisation fails. This procedural name identifies it as a fixed-compute implementation-level likelihood reference; tolerance-based convergence and restart diagnostics are identified as follow-up validation. The cGAN reports one generator forward pass at the prescribed latent $\mathbf{z} = 0$, as defined in Section 3.6.3. The direct learned models use one neural forward pass followed by the deterministic Cholesky map in Equation (2.28). Their final common-noise evaluation forces the optional post-Cholesky safeguard off; the separate timing benchmark includes it as the conservative checkpoint-configured path. As defined in Section 3.6.4, each flow estimate averages $S = 10$ physical endpoints from eight-step trajectories (approximately eighty velocity-network evaluations); Equation (3.5) instead takes a per-state minimum. The fixed $S = 10$ is the reported quality–cost choice, with inference cost linear in S .

Uncertainty intervals. Two secondary analyses use a non-parametric percentile bootstrap. A bootstrap replicate constructs a pseudo-test sample by drawing the observed state-level quantities again, with replacement, and recomputes their mean. The 2.5th and 97.5th percentiles of the replicate means form the reported 95% interval. For TTA, we first form the paired per-state difference between direct and augmented Bures error, then resample the 45 999 differences 2 000 times; resampling the differences preserves the fact that both predictions use the same state and noisy observation. The spectral analysis applies the same procedure independently within each bin using 500 resamples. All bootstrap seeds are deterministically derived from evaluation seed 42. These intervals describe finite-test-sample uncertainty for frozen checkpoints and fixed evaluation observations; they do not include variation from retraining. Purity uses 25 equal-width bins on $[1/4, 1]$. Positive minimum eigenvalues use 20 quantile bins, with $\lambda_{\min} \leq 10^{-8}$ separated; bins with fewer than 50 states are omitted.

Test-time augmentation by qubit exchange. Section 2.3.5 derives the qubit-exchange symmetry and the transform–predict–map–back–average mechanism. In the reported TTA experiment, the 4×4 frequency and confidence grids are transposed, the same frozen RhoTransformer makes a second prediction, and the predicted qubits are exchanged back before the two density matrices are averaged. This uses no additional quantum measurements and costs one additional neural forward pass. A development-time gate, fixed at twelve observed settings before the final paired test evaluation, disables augmentation for sparser inputs; this is an empirical operating choice rather than a restriction of the symmetry. Direct and augmented estimates share the state, subset, shot counts and noisy measurements; only the permutation and second pass differ. Section 4.8 reports the paired effect.

Training to a plateau. Every centrally compared pipeline in Table 4.1 is trained to a validation plateau rather than a fixed epoch count. The scheduling metric is mean Bures distance on the common stored validation observations, generated once from the base mixture with a fixed seed: 15% missing, otherwise log-uniform shots over $[1, 10^5]$. A reduce-on-plateau schedule decays the learning rate when this metric stalls and stops after a long patience window. We verified a genuine aggregate plateau for the direct and flow-matching transformers. Because the aggregate emphasises low-shot and partially observed cases, it plateaus before the easiest high-shot reconstructions, which can still improve slightly. Per-model curves accompany the results.

Software, hardware and training. The models and differentiable physics routines are implemented in PyTorch [30]. Numerical processing and analysis use NumPy [65], SciPy [66], pandas [67] and scikit-learn [68]; numerical result figures are produced with Matplotlib [69]. The final campaign ran on one remote Linux compute server with an NVIDIA GeForce RTX 3080 graphics processing unit (GPU; 10 GB), so the reported models share the same hardware. Model widths and the mini-batch size of 2048 were chosen to fit that memory budget. Mixed precision (bf16) was used during development where numerically safe; the final reported networks train in single precision, while the regularised Bures loss promotes its spectral matrix operations to complex double precision before returning the scalar loss in single precision (Section 2.2.1). Under these constraints a 500-epoch run of the principal transformer takes about five hours. Finite-shot noise is generated online on the GPU (Section 3.1), which costs little per mini-batch and avoids storing any noisy copy of the dataset. Exact architectures, optimiser settings, learning-rate schedules and loss weights are recorded in Attachment A.1 from the stored training checkpoints.

Verification of the physics primitives. We tested every physics routine independently of the neural networks. The Bures implementation matches the analytic pure-state formula to 10^{-8} and agrees with Qiskit [70] on random mixed states to 10^{-5} . Linear inversion recovers clean states to single-precision round-off, while the pseudoinverse reproduces the observed probabilities to machine precision. The concurrence implementation agrees with the dataset labels to within 3×10^{-4} .

Each exact simplification in Section 3.5 is also covered by a numerical identity test comparing the rewritten module with the original on the same weights; the disagreement is below 10^{-6} in single precision

4. Results

Table 4.1 gives the central result: among the selected implementations, RhoTransformer has the lowest reported mean Bures distance (Equation (2.33)) at all four headline operating points in Table 4.1, using one neural forward pass followed by the Cholesky map in Equation (2.28). The largest gain, however, comes from replacing the naive feed-forward baseline with the formulation-aware DeepSets model, a joint change of input representation, set-based aggregation, physical output parameterisation and training objective.

The benchmark spans four reconstruction families (direct feed-forward, conditional adversarial, flow-matching and direct attention-based regression) whose selected implementations retain family-specific backbones, objectives and inference procedures. All headline values use the same 45 999-state synthetic test split and evaluation protocol (Section 3.8). A separate matched decoder comparison in Section 4.1.2 isolates query self-attention from one attention-free replacement. The remaining sections analyse incomplete and finite-shot measurements, physical properties and cost, then the secondary selection, consistency-loss and test-time-augmentation studies.

4.1 Central comparison across reconstruction pipelines

Table 4.1 compares the five selected learned implementations, an additional reconstruction-only cGAN comparison, two naive learned baselines and classical reconstructors at three complete-noisy points and one incomplete-noisy point. Every learned entry uses the same data partitions and is evaluated on the same paired test records. Compatible models also share the confidence encoding, Cholesky output and regularised Bures term, while their training samplers, objectives and stopping schedules remain pipeline-specific. Each learned row uses its validation-selected plateau checkpoint; the classical rows are deterministic procedures with their stated compute budgets. The separate seed study quantifies run-to-run variation for the two direct models.

RhoTransformer leads every column, with mean Bures distances of 0.073, 0.029 and 0.015 at 10^3 , 10^4 and 10^5 uniform shots per setting and 0.103 with twelve settings observed at 10^4 shots per observed setting. DeepSets and the flow-matching transformer are closest. Their distinct point-estimation and inference-cost rules are consolidated in Section 3.8. The seed study later uses like-for-like 500-epoch RhoTransformer and DeepSets runs; the headline rows remain their separately selected plateau checkpoints.

4.1.1 Effect of the shared formulation and feed-forward redesign

The combined feed-forward redesign produces a larger gain than the remaining differences among the centrally benchmarked pipelines. At 10^5 uniform shots per setting, the vanilla MLP, which receives raw inputs and is trained with

Table 4.1: Mean Bures distance (lower is better) on the same 45 999 paired test inputs. The first three columns use all settings; the last uses twelve settings at 10^4 shots per observed setting. The two linear rows differ only in their physicality correction; the fixed-step likelihood routine and all model configurations are specified in Attachments A.1.6 and A.1.8. The reconstruction-only cGAN row is an additional within-architecture comparison and is not counted among the five selected implementations. Matched seed variation is reported in Table 4.8.

Model	uniform 10^3	uniform 10^4	uniform 10^5	$n_{\text{high}}=12$
Linear/pseudoinverse estimate + clipping ($\mathcal{P}_{\text{clip}}$)	0.157	0.070	0.031	0.386
Linear/pseudoinverse estimate + simplex projection	0.136	0.057	0.024	0.365
250-step Cholesky likelihood optimisation	0.118	0.049	0.020	0.282
Vanilla MLP	0.119	0.082	0.079	0.167
Naive transformer	0.105	0.065	0.057	0.146
cGAN (adversarial)	0.095	0.069	0.067	0.142
cGAN generator, reconstruction-only ($\lambda_{\text{adv}} = 0$)	0.088	0.058	0.054	0.134
Flow matching (MLP)	0.084	0.046	0.039	0.124
DeepSets MLP	0.075	0.034	0.022	0.116
Flow matching (transformer)	0.077	0.034	0.022	0.110
RhoTransformer	0.073	0.029	0.015	0.103

plain mean-squared error, reaches 0.079. The redesigned DeepSets MLP, with confidence-scaled harmonic Fourier inputs defined by Equations (2.41) and (2.42), the Cholesky coordinates (2.29) and the regularised Bures training loss of Section 2.2.1, reaches 0.022, an improvement by roughly a factor of 3.5 within the broad feed-forward paradigm. This factor quantifies the joint effect of the formulation and the accompanying architectural redesign: DeepSets replaces the flat concatenated input with a per-slot network and permutation-invariant pooling.

The narrower conditional-GAN ablation in Table A.4 provides complementary short-run evidence that Fourier features improve on raw inputs within that family (Section 2.3.8).

The additional cGAN row in Table 4.1 uses the same generator architecture and $z = 0$ reporting rule but a separately continued reconstruction-only training path ($\lambda_{\text{adv}} = 0$). It gives 0.088, 0.058, 0.054 and 0.134 at the three uniform and twelve-setting points, below the adversarial cGAN’s 0.095, 0.069, 0.067 and 0.142. Because the optimisation histories differ, this comparison ranks the two implemented point estimators but does not isolate the causal effect of the adversarial term.

4.1.2 Interpretation of the selected-pipeline ranking

RhoTransformer’s only non-degenerate attention is decoder query self-attention (Section 3.5); by the single-token identity in Equation (2.68), its other attention sublayers reduce to affine maps. We therefore compare two single-encoder-token models with the same encoding, four-layer encoder, Cholesky head, consistency objective, noise mixture and 500-epoch schedule. One retains query self-attention; the other replaces it by an attention-free SwiGLU decoder of Equation (2.57) and has more stored parameters (6.42 versus 5.62 million).

Table 4.2 and Figure 4.1 report the comparison. At the displayed operating points, the two means are equal after rounding at 10^3 uniform shots per setting.

The query-attention decoder is lower at 10^4 (0.030 versus 0.031) and 10^5 shots per setting (0.017 versus 0.019), while the attention-free decoder is slightly lower in the twelve-setting case (0.106 versus 0.108). Thus the smaller attention model has a high-shot advantage over this particular feed-forward replacement. The approximately ten-percent reduction at 10^5 shots is a benchmark-specific query-self-attention benefit [9]: under this recipe, interactions among sixteen output queries help when measurements are precise. Attention-based tomography precedents include [17, 20, 18, 19].

Table 4.2: Mean Bures distance for two reconstructors differing only in the decoder under the same encoding, SwiGLU encoder, Cholesky head, objective, noise mixture and 500-epoch schedule. The query-attention model uses fewer stored parameters. Test inputs and operating points match Table 4.1; lower is better.

Decoder	Params	uniform 10^3	uniform 10^4	uniform 10^5	$n_{\text{high}}=12$
Query self-attention	5.62 M	0.074	0.030	0.017	0.108
Attention-free MLP	6.42 M	0.074	0.031	0.019	0.106

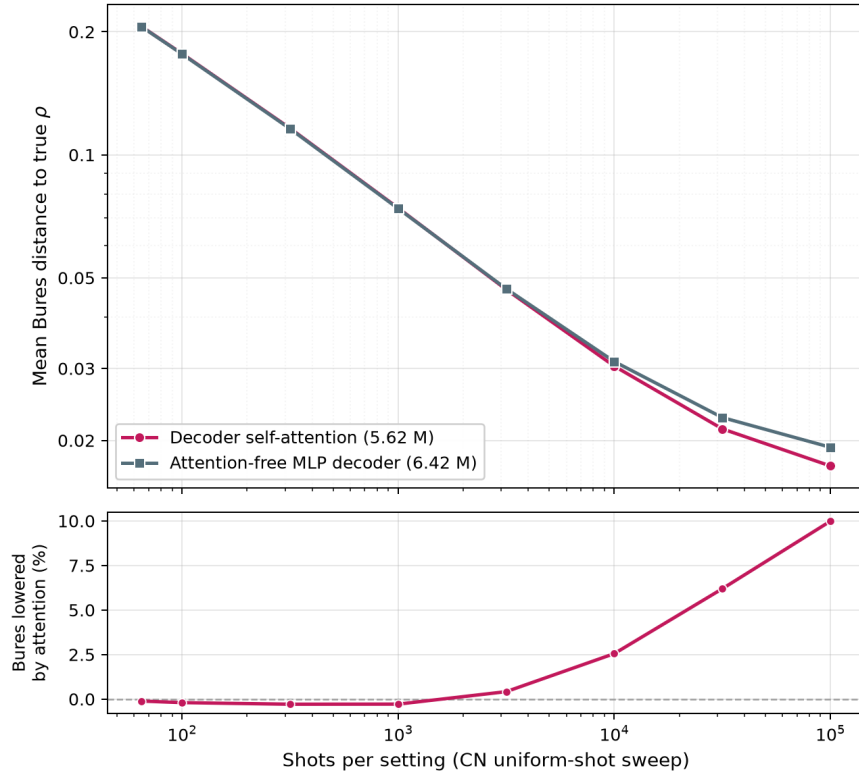


Figure 4.1: Comparison of decoder query self-attention with the larger attention-free feed-forward control of Table 4.2. Upper panel: mean Bures distance on the fixed test split across the complete-noisy (CN) uniform-shot sweep. Lower panel: the percentage by which the query-attention decoder lowers the mean Bures distance. The attention decoder has similar error at low shot counts and lower error at the high-shot end of the complete-measurement sweep.

The flow-matching transformer introduced in Section 2.3.9 is second at twelve settings (0.110), matches DeepSets to displayed precision at 10^4 and 10^5 uniform

shots per setting (0.034 and 0.022), and trails it at 10^3 (0.077 versus 0.075). Its 0.022 also improves on the feed-forward flow model’s 0.039 at 10^5 . Its ten-trajectory averaging rule and the resulting cost are fixed in Section 3.8.

On this point-estimate metric, direct RhoTransformer is lower than all three selected generative implementations and the additional reconstruction-only cGAN comparison. The reconstruction-only implementation is lower than the selected adversarial cGAN. The fixed point-estimation protocols of both generative families are consolidated in Section 3.8 and do not characterise their full conditional distributions. Section 3.6.3 records the earlier GAIN stage.

The 250-step Cholesky likelihood-optimisation baseline, based on the likelihood formulation of Section 2.1.9, reaches 0.020 with complete data at 10^5 shots, but reaches 0.118 at 10^3 and 0.282 with twelve settings. Exact eigenvalue-simplex projection, described in Section 2.1.7, lowers the linear baseline to 0.136, 0.057, 0.024 and 0.365, an approximately 5–23% gain that preserves the ranking (Attachment A.1.6). Relative to these implemented references, learned models gain most when data are missing or noisy, where they can use an in-distribution prior.

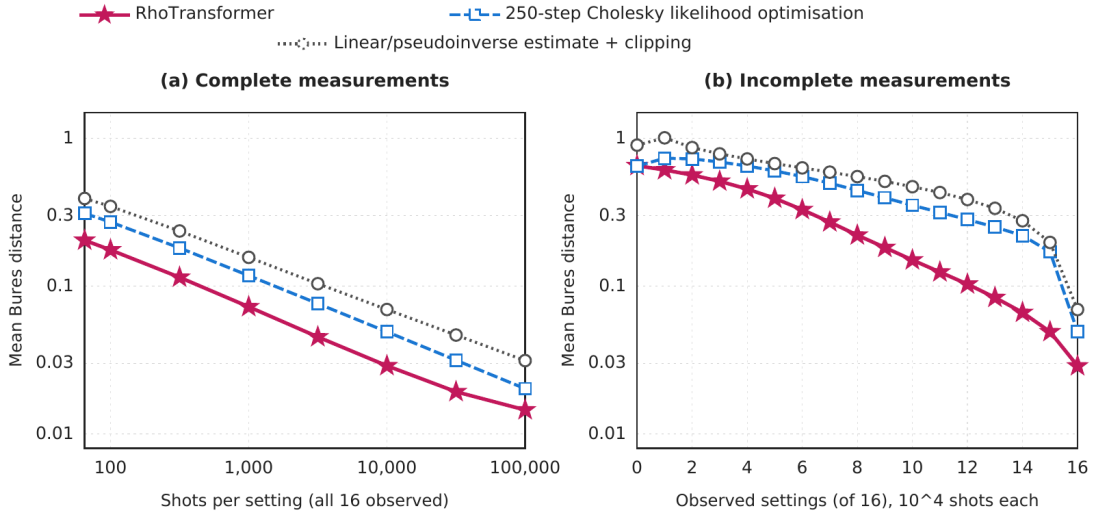


Figure 4.2: Mean Bures distance for the RhoTransformer, 250-step Cholesky likelihood optimisation and the linear/pseudoinverse estimate followed by $\mathcal{P}_{\text{clip}}$ from Algorithm 2.1 on identical test inputs. *Left*: complete measurements as the shot budget varies. *Right*: 10^4 shots per observed setting as measurements are removed. Both panels use a logarithmic y -axis; Attachment A.1.6 specifies the two classical procedures.

4.2 Principal reconstructor: RhoTransformer

The principal reconstructor is a transformer with 3 980 545 stored parameters and about 3.69 million on its active forward path, trained from scratch, which we call the *RhoTransformer*. It is the trimmed endpoint of the design process in Sections 3.5 and 3.6.1. Its data flow is shown in Figure 3.4; the detailed layer configuration, training schedule and loss weights are recorded in Attachment A.1.1.

Each James–Kwiat probability [11] is encoded with the confidence-scaled harmonic Fourier map (2.42), so a missing setting contributes the zero vector. The encoder compresses the sixteen settings into a summary representation. Sixteen learned decoder queries exchange information through query self-attention and produce the sixteen real Cholesky coordinates defined in Equation (2.29). Equation (2.28) then yields a positive-semidefinite, unit-trace density matrix.

One frozen set of weights handles all headline sweeps. During training it sees a randomised mixture of complete and partial inputs at large and small shot counts, and the per-slot confidence (2.41) identifies the operating condition. The selector, measurement-loss and TTA extensions use the explicitly identified secondary-study retrainings described in Sections 4.6, 4.7 and 4.8.

Checkpoint selection and uncertainty. Two frozen RhoTransformer checkpoints appear in this chapter. The *principal* checkpoint is the validation-selected model used in the central comparison of Table 4.1; its plateau retraining followed the pre-specified stopping rule of that benchmark. The separately *secondary-study* checkpoint is a deeper, narrower variant with a per-setting noise-standard-deviation channel (Attachment A.1.1). It remains fixed throughout the selector, consistency-loss and TTA studies.

The checkpoints exchange rank across regimes. On the same paired test records, the principal model gives 0.073, 0.029, 0.015 and 0.103 at the four headline points, whereas the secondary model gives 0.075, 0.028, 0.011 and 0.100. Table 4.1 retains the principal model because it is the preselected checkpoint of the central selected-pipeline benchmark. The other checkpoint remains the fixed platform for the internally matched secondary studies. Attachment A.1.2 records the development order and selection rules.

Unless identified as development evidence, reconstruction values use the fixed 45 999-state test split and direct comparisons use paired inputs (Section 3.1). The sampling standard error of each reported mean Bures distance is below two percent of that mean. A common-input re-evaluation preserves the central ordering for Bures means and medians and for mean and median Hilbert–Schmidt error. Training-run variation is measured separately by the three matched 500-epoch runs in Table 4.8; these are not the headline plateau checkpoints. The TTA study instead uses paired intervals on identical inputs.

4.3 Reconstruction from incomplete measurements

Figure 4.3 follows RhoTransformer as a uniformly random subset of $n_{\text{high}} \in \{0, 1, \dots, 16\}$ James–Kwiat measurements is observed; n_{high} counts settings, not shots. Unselected settings receive zero confidence, while selected probabilities are binomial-sampled according to Equation (2.12) at 10^3 , 10^4 or 10^5 shots per observed setting. The mean Bures distance is 0.103 with twelve settings at 10^4 shots per observed setting and 0.015 with all settings at 10^5 shots per setting. At sixteen settings this is the complete-noisy regime of Section 4.4; active subset choice is studied separately in Section 4.6. Because the training sampler averages 13.6 observed settings (Section 3.2), the sparsest points also extrapolate in mask

severity.

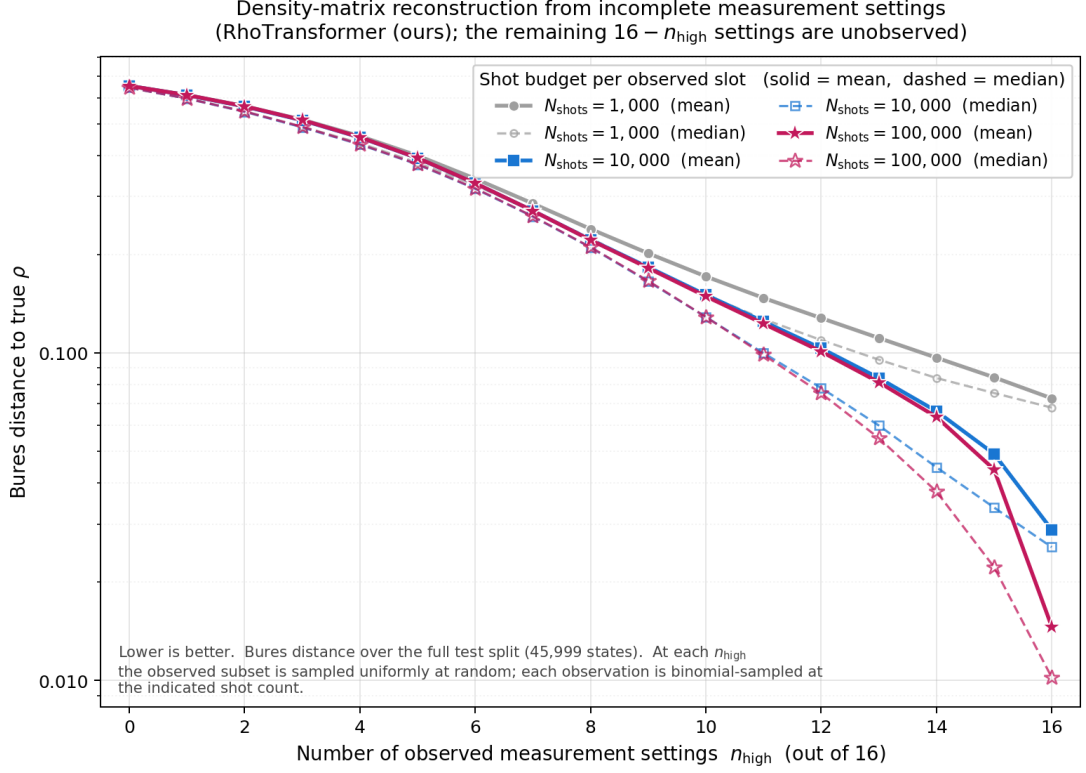


Figure 4.3: Mean (solid) and median (dashed) Bures distance for random observed subsets. Grey, blue and magenta denote 10^3 , 10^4 and 10^5 shots per observed setting; statistics use all 45 999 test states and a logarithmic y -axis. Table 4.4 separates the two single-omission classes at $n_{\text{high}} = 15$.

Table 4.3 complements the three RhoTransformer curves with a cross-pipeline comparison at a representative INC point: twelve settings observed at 10^5 shots per observed setting. RhoTransformer has the lowest mean Bures distance there, followed by the flow-matching transformer. Across the complete $n_{\text{high}} = 0, \dots, 16$ INC sweep, the cGAN is slightly lower when only zero to four settings are observed; the difference at four settings is 0.4553 versus 0.4557. RhoTransformer is lowest among the five selected pipelines from five through sixteen observed settings. This crossover separates the regime with almost no observations, where the input supplies little state-specific information, from the range in which the reconstructor can increasingly use the measured probabilities.

Overall pattern. Mean Bures distance decreases monotonically as settings are added. With no observations, every state produces the same input and the deterministic model returns one learned state with mean $d_B \approx 0.65$, close to the aggregate error of the no-information reference $I_4/4$. At the other endpoint, the same frozen weights reach $d_B \approx 10^{-2}$ with all sixteen settings at 10^5 shots.

The three shot-budget curves remain close while many settings are absent, because uncertainty from the missing settings dominates sampling noise. They begin to separate at about eleven observations; with all sixteen settings, 10^5

Table 4.3: Mean Bures distance (lower is better) for the five selected learned pipelines with twelve randomly chosen settings observed at 10^5 shots per observed setting. All rows use the same 45 999 paired test records. This is a representative point from the incomplete near-clean (INC) sweep.

Model	INC: $n_{\text{high}} = 12, 10^5$
cGAN (adversarial)	0.142
Flow matching (MLP)	0.122
DeepSets MLP	0.114
Flow matching (transformer)	0.107
RhoTransformer	0.101

shots are roughly five times more accurate than 10^3 . At 10^3 shots, the mean improves from about 0.13 with twelve settings to 0.08 with fifteen. Additional shots therefore help most after the observed settings already constrain the state.

Mean and median nearly coincide up to about eight observed settings but separate beyond ten, especially at larger shot counts. The resulting right-skewed error distribution means that a small difficult tail controls the mean once most states are accurate, whereas the median describes the typical state. Figure 4.6 tests whether this tail is associated with purity and the smallest eigenvalue.

Why the sixteenth setting still matters. At $n_{\text{high}} = 15$, Claim 1 shows that the state remains identifiable only when HH , HV , VH or VV is omitted; each other omission leaves one exact null direction (the DD example is Equation (2.22)). A random single omission is therefore non-identifying with probability $12/16$. The sixteenth setting removes that ambiguity and, for the four trace-determined omissions, supplies an independent finite-shot estimate. The final drop grows with shot count because missing-setting uncertainty then dominates binomial noise.

The floating-point audit in Attachment A.1.5 rules out the optional post-Cholesky safeguard as the cause: toggling it changes mean Bures error by at most 2.4×10^{-6} , and the safeguard contains no information about the missing null direction. The final drop instead reflects projector-set identifiability, independent finite-shot information and the reconstructor’s treatment of an absent slot in underdetermined cases.

Empirical tests of identifiability and trace completion. Table 4.4 tests the two omission groups with the frozen checkpoint. Here XY applies X to the first qubit and Y to the second (Section 2.1.5); HH , HV , VH and VV occupy one-based row-major positions 1, 2, 5, 6 (zero-based 0, 1, 4, 5). Omitting one of these four settings is consistently less damaging than omitting one of the other twelve. At 10^5 shots the group means are 0.022 and 0.051; at 10^4 , their 4 : 12 weighted mean is 0.050, consistent with the random-omission point of about 0.049 in Figure 4.3. The widening separation supports the predicted missing-direction mechanism.

Table 4.4: Mean test-set Bures distance for the frozen RhoTransformer on identical states and noise draws. The trace-determined columns average omissions of HH , HV , VH and VV ; “trace-filled” supplies the clamped completion of Equation (2.21) with nominal confidence. The final column averages the other twelve omissions. Shots refer to directly observed settings.

Shots per setting	All observed	Trace-determined omission (4)		Non-identifying omission (12)
	all 16	masked	trace-filled	masked
10^3	0.073	0.077	0.079	0.087
10^4	0.029	0.033	0.031	0.055
10^5	0.015	0.022	0.015	0.051

For the four identifiable omissions, a paired control fills the absent slot by the trace-completion identity (2.21). At 10^5 shots, completion lowers the mean from 0.0222 to 0.0150, removing 94% of the excess over the all-sixteen value 0.0146. It is also close to the 0.0151 descriptive reference obtained by multiplying the all-sixteen error by $\sqrt{16/15}$ under an equal-information $N^{-1/2}$ model. Because the intervention jointly changes the value, mask and confidence, it demonstrates the benefit of presenting the identity as a populated input slot rather than an output-physicality effect.

At 10^3 shots, however, hard completion raises the error from 0.0768 to 0.0791; at 10^4 it lowers 0.0332 to 0.0308, closing 56% of the gap. Before clamping to $[0, 1]$, the completed estimate for an omitted computational-basis outcome j , $\widetilde{m}_j = 1 - \sum_{i \neq j} \widehat{m}_i$, has variance

$$\text{Var}(\widetilde{m}_j) = \frac{1}{n} \sum_{i \neq j} m_i(1 - m_i), \quad \text{Var}(\widehat{m}_j) = \frac{m_j(1 - m_j)}{n}. \quad (4.1)$$

Averaging over the four omissions counts each $m_i(1 - m_i)$ three times for the completed estimate and once for direct sampling, so the mean variance ratio is three whenever the denominator is nonzero. The supplementary choice $n_{\text{eff}} = n/3$ therefore matches the average variance, although the precision still depends on the unknown probabilities and completion is clamped to $[0, 1]$. It recovers most of the low-shot penalty but does not beat masking. Explicit completion therefore helps when the measured frequencies are precise; with few shots, its higher derived variance favours leaving the slot missing.

4.3.1 Context against published incomplete-measurement results

Figure 4.4 places the incomplete sweep beside representative two-qubit results of Krawczyk et al. [12]. Their published coordinates were digitised from the high-resolution Figure 3 in arXiv v4 and retained to two or three significant figures; independent axis calibration and line-centre fits agree within the plot’s resolution. We show the pseudoinverse, both MLE configurations, the value-dependent Corrector NN, the adjusted-basis LSTM and the adapted LI-NN. The original figure contains the remaining ablations. Its unshown predefined-basis

selector reduces to a learned fixed order, whereas the adjusted-basis LSTM can propose projectors outside the fixed grid. The adapted LI-NN follows the paper’s separate Gaussian-corruption protocol for selected outcomes.

The common vertical axis does not make the experiments matched. Our curves use binomially sampled probabilities from random subsets of the fixed James–Kwiat grid; most published curves use exact probabilities, and the adjusted-basis LSTM can choose off-grid projectors. Moreover, when the published pseudoinverse is non-positive, its reported value is an algebraic evaluation of the Bures expression rather than a distance between physical states and can exceed the physical-state maximum $\sqrt{2}$. The overlay therefore provides literature context rather than a controlled ranking.

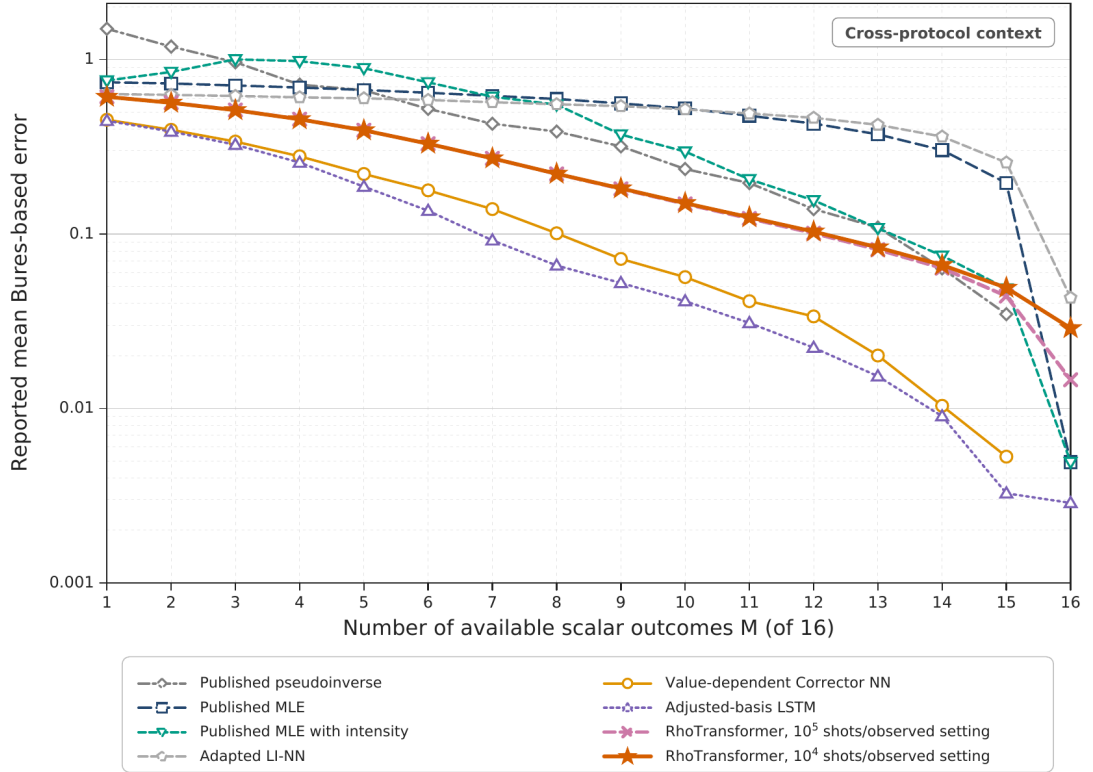


Figure 4.4: Reported mean Bures-based errors versus the number of available scalar outcomes. Published coordinates are digitised from Figure 3 of Krawczyk et al. [12]; the RhoTransformer curves use random fixed-grid subsets under finite-shot noise. The acquisition-policy comparison under one fixed reconstructor appears in Figure 4.9.

The published fixed-grid average at $M = 15$ mixes four trace-determined omissions with twelve non-identifying omissions. MLE with intensity falls from about 0.049 at $M = 15$ to 0.0049 at $M = 16$, while MLE without intensity falls from 0.196 to the same scale; “intensity” is the fit’s present/absent indicator. Because physical MLE enforces unit trace and the sixteenth fixed-grid setting resolves most single-omission null directions, this drop is qualitatively consistent with Claim 1. The adjusted-basis LSTM differs because it selects projectors outside that grid.

The pseudoinverse and Corrector NN stop at $M = 15$ because all sixteen exact James–Kwiat outcomes make the measurement matrix invertible: the pseudoinverse reconstruction is then exact and the corrector has no residual to correct [12]. Their $M = 16$ error is zero and cannot be drawn on the logarithmic axis. With exact probabilities this endpoint also has no binomial-noise floor. Figure 4.9 gives the separate controlled acquisition comparison with one fixed reconstructor.

4.4 Reconstruction under finite-shot noise

With all sixteen settings observed, the mean Bures distance falls to 0.029 at 10^4 shots per setting and 0.015 at 10^5 shots. To isolate this finite-shot effect from missing data, we vary only the uniform shot count N_{shots} over a logarithmic grid from 65 to 10^5 . Figure 4.5 reports Bures distance and squared Hilbert–Schmidt distance. The latter follows the squared-Frobenius convention used by Macarone Palmieri et al. [13], rather than the unsquared metric of Equation (2.37).

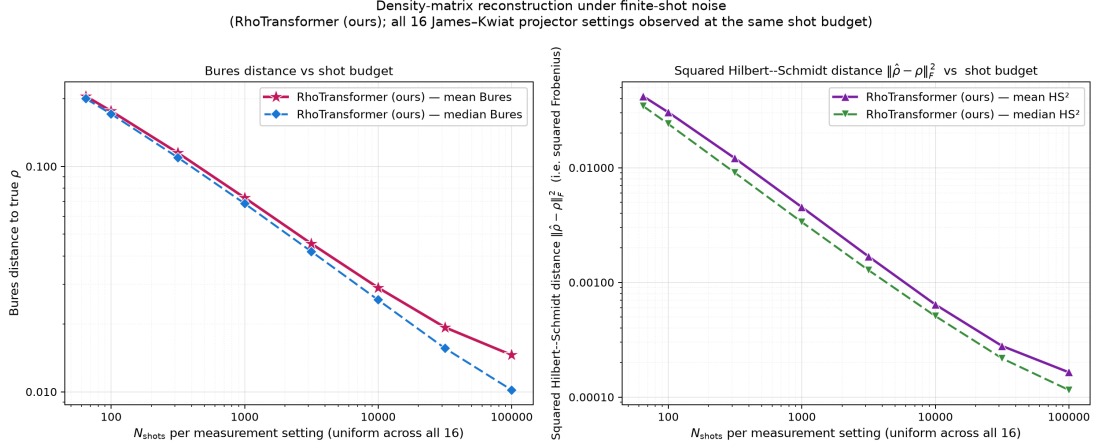


Figure 4.5: All sixteen settings are observed while the shot budget varies. *Left*: mean and median Bures distance. *Right*: mean and median squared Hilbert–Schmidt distance $\|\hat{\rho} - \rho\|_F^2$, following Macarone Palmieri et al. [13]. Both axes are logarithmic; statistics use all 45 999 test states.

Both panels are approximately linear on log–log axes, with no visible model-imposed floor. Improving binomial precision explains the trend, while the squared Hilbert–Schmidt quantity falls more steeply because it squares entrywise error. The mean-to-median ratio grows from about 1.03 at 65 shots to 1.4 at 10^5 , reproducing the right-skewed tail of the incomplete sweep. The means at 10^4 and 10^5 are 0.029 and 0.015, matching Table 4.1; Figure 4.2 supplies the classical comparison.

Dependence on spectral properties. Figure 4.6 bins error by the purity $\mathcal{P}(\rho)$ of Equation (2.2) and the true state’s smallest eigenvalue. At 10^5 shots, error falls as $\lambda_{\min}$ grows and rises with purity across the mixed-state bulk. Exactly pure states are an exception: they are reconstructed an order of magnitude better than adjacent near-rank-deficient mixed states, plausibly because the generator

densely represents the pure family. The hardest group is therefore the nearly rank-deficient mixed states just inside the physical-state boundary described in Section 2.1.1. This association is descriptive because generator density, Cholesky output, loss perturbation and rank frequency vary together; it is flatter under scarce or incomplete data, where observation uncertainty dominates.

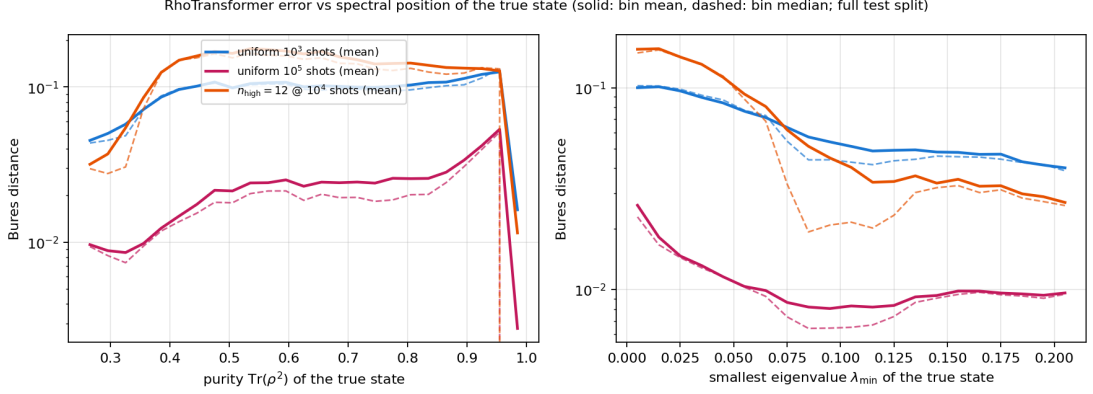


Figure 4.6: Binned mean (solid) and median (dashed) Bures distance of the RhoTransformer against the purity $\mathcal{P}(\rho) = \text{Tr}(\rho^2)$ from Equation (2.2) (left) and the smallest eigenvalue $\lambda_{\min}$ (right) of the true state, on the full test split at three operating points. Shading gives 95% bootstrap intervals for the bin mean under the procedure of Section 3.8; bins with fewer than 50 states are omitted. Rank-deficient states ($\lambda_{\min} \leq 10^{-8}$) form the square-marked bin; vertical axes are logarithmic.

Qualitative examples. Figure 4.7 shows a near-pure entangled state at 10^4 complete shots ($d_B = 0.011$), a near-rank-deficient mixed state (0.060), and the first state with twelve settings (0.007). The last happens to outperform its complete-input noise draw, illustrating per-state variation. Entrywise maxima are 0.006, 0.019 and 0.004 on the tighter difference scale.

4.5 Physical properties, computational cost and robustness

This section adds inference cost, physical diagnostics and training robustness to the central error ranking.

4.5.1 Measured inference cost

Table 4.5 times batch 1 and batch 4096 on the RTX 3080. Here an inference batch is simply a group of states processed simultaneously with fixed parameters; it involves no optimiser update. Its conservative RhoTransformer path includes the optional 4×4 safeguard eigendecomposition, although common-noise accuracy disables it. RhoTransformer and DeepSets share the low-millisecond

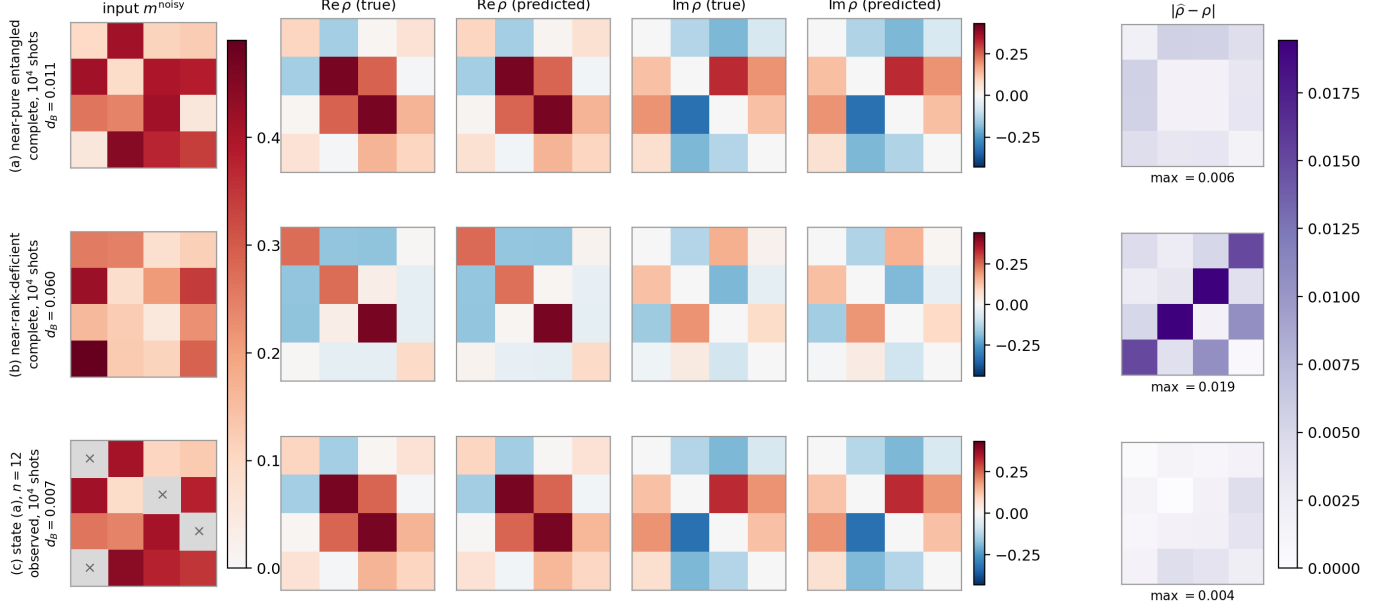


Figure 4.7: Three test-state reconstructions: noisy input grid, real and imaginary parts of the true and predicted matrices, and $|\hat{\rho} - \rho|$ on a tighter shared scale. Grey crosses mark missing settings; rows are described in the text.

batch-1 scale; the GAN generator and linear inversion are cheaper, while RhoTransformer is much faster than the selected eighty-evaluation flow point and 250-step Cholesky likelihood routine. Absolute timings remain hardware-, batch- and implementation-dependent.

Table 4.5: Measured wall-clock inference time on the shared host, reported as the median and interquartile range (IQR) of thirty repetitions after five warm-up iterations. Latency uses an inference batch of 1, and throughput uses an inference batch of 4096. Here an inference batch is a group of states processed simultaneously with fixed parameters. GPU timings use explicit synchronisation and resident inputs; the RhoTransformer row includes its physical output map and safeguard. For linear inversion, “clipping” denotes $\mathcal{P}_{\text{clip}}$ from Algorithm 2.1.

Method	Device	Latency, inference batch 1 (ms)	Throughput, inference batch 4096 (states/s)
RhoTransformer	GPU	2.62 [2.61, 2.62]	149 127 [149 032, 149 422]
DeepSets MLP	GPU	1.54 [1.53, 1.54]	186 132 [185 989, 186 282]
cGAN generator	GPU	0.73 [0.73, 0.73]	1 753 377 [1 750 906, 1 770 211]
Flow matching (transformer; 10×8 evaluations)	GPU	241.34 [240.83, 242.59]	734 [732, 737]
Linear inversion + clipping ($\mathcal{P}_{\text{clip}}$)	GPU	0.41 [0.40, 0.41]	5 690 975 [5 642 691, 5 721 030]
Linear inversion + clipping ($\mathcal{P}_{\text{clip}}$)	CPU	0.13 [0.13, 0.13]	472 549 [470 157, 473 597]
250-step Cholesky likelihood optimisation	GPU	710.59 [707.43, 722.39]	5 058 [5 057, 5 058]

The central processing unit (CPU) linear-inversion row is faster than its GPU counterpart at batch 1, which is consistent with kernel-launch and synchronisation overhead dominating the tiny 4×4 calculation. At batch 4096 the same GPU implementation delivers about twelve times the measured CPU throughput. Cross-device latency should therefore be read together with the batch size and implementation.

Table 4.6: Agreement between Wootters concurrence from Equation (2.39), computed from the RhoTransformer reconstructions and from the true states on the full test split. Mean absolute error (MAE) and root-mean-square error (RMSE) quantify the concurrence-value error; r and ρ_s are Pearson and Spearman correlations. The last three columns give balanced separable-versus-entangled accuracy when the same threshold τ is applied to both the reconstructed and true concurrence. In the last row, $n_{\text{high}} = 12$ of the sixteen measurements are observed at 10^4 shots and the remaining four are missing.

Operating point	MAE	RMSE	r	ρ_s	10^{-3}	10^{-2}	0.05
uniform 10^3	0.019	0.034	0.9925	0.9268	0.821	0.892	0.929
uniform 10^5	0.004	0.008	0.9996	0.9680	0.863	0.969	0.989
$n_{\text{high}}=12$ at 10^4	0.028	0.054	0.9807	0.8780	0.792	0.871	0.896

4.5.2 Physical properties and implied measurements

The RhoTransformer preserves concurrence closely, and all tested pipelines preserve the signal used by the frozen entanglement classifier. These analyses add physical context to the Bures ranking; changes in implied measurements then show how strongly each pipeline adjusts its inputs.

Concurrence of the reconstructed states. The Wootters concurrence (2.39) is computed directly from each reconstructed density matrix and compared directly with the concurrence of the true state. The RhoTransformer preserves this scalar closely: at 10^5 uniform shots per setting the mean absolute concurrence error is 0.004 with a Pearson correlation of 0.9996, and with twelve measurements observed at 10^4 shots the error remains below 0.03 with a correlation near 0.98.

Thresholded entanglement decision. For the separable-versus-entangled decision, each displayed threshold is applied identically to true and reconstructed concurrence. This avoids treating a changing prediction threshold against a fixed true threshold as a sensitivity analysis of the physical boundary. The corresponding prevalence, precision, recall and confusion counts are reported in Attachment A.1.11.

Frozen classifier. The independently trained classifier documented in Attachment A.1.9 is frozen and applied to each reconstruction’s implied measurements. At large shot counts, shared-formulation pipelines lie within about 1.5 accuracy points of its clean-input reference and degrade similarly as data are removed. This is coarser than Bures distance once estimates fall on the correct side of the entanglement boundary, but it confirms preservation of classifier-relevant signal. Under-trained GANs that approach the maximally mixed state instead reach majority-class accuracy.

4.5.3 Training saturation

Most aggregate improvement is complete by about 1000 epochs, while the high-information regime continues to improve modestly. At 500, 1000 and the selected

Table 4.7: Accuracy of the frozen entanglement classifier to measurements implied by each reconstruction. Higher is better; clean inputs give a reference accuracy of 0.869. Columns show complete measurements at 10^5 shots and incomplete measurements at $n_{\text{high}} = 8$ and 12 with 10^4 shots per observed setting. Statistics use all 45 999 test states.

Model	uniform 10^5	$n_{\text{high}}=8$	$n_{\text{high}}=12$
RhoTransformer	0.867	0.723	0.827
Flow matching (transformer)	0.867	0.731	0.827
DeepSets MLP	0.866	0.713	0.816
Flow matching (MLP)	0.863	0.722	0.824
cGAN (adversarial)	0.855	0.701	0.802
Naive transformer	0.867	0.709	0.822
Vanilla MLP	0.868	0.691	0.811

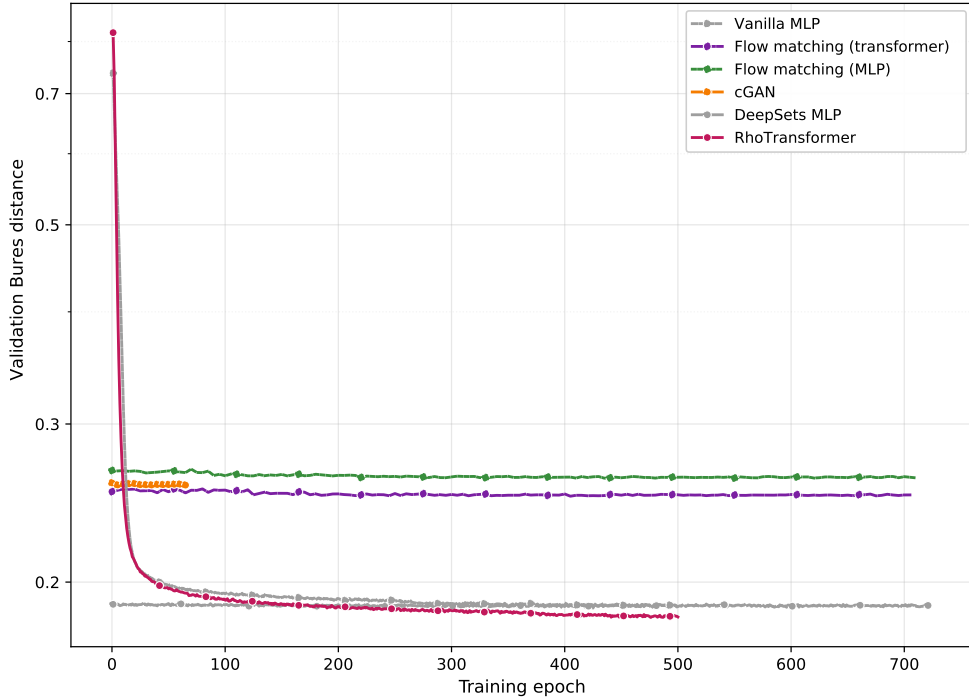


Figure 4.8: Validation Bures distance versus logged training epoch for the selected pipelines. Curves end at their last available logged epoch; the separately evaluated RhoTransformer continuation is reported in the text.

plateau, RhoTransformer’s 10^5 -shot validation Bures distance is 0.017, 0.015 and 0.014, respectively; aggregate validation Bures changes only from 0.183 to 0.181. The corresponding plateau test value in Table 4.1 is 0.015.

Figure 4.8 contains the available training logs, whose lengths differ across pipelines. The displayed RhoTransformer trace ends before the separately evaluated continuation that produced the 1000-epoch and plateau values above. Model ordering is stable after the initial transient, although the smallest differences remain sensitive to optimisation and seed.

Table 4.8 gives three like-for-like 500-epoch seeds for RhoTransformer and

Table 4.8: Mean Bures distance with sample standard deviation in parentheses over matched 500-epoch runs with seeds 42–44 on the same test split. Table 4.1 instead uses the plateau-extended seed-42 checkpoints. The final column has twelve settings observed at 10^4 shots per observed setting.

Model	uniform 10^3	uniform 10^4	uniform 10^5	$n_{\text{high}}=12$
RhoTransformer	0.0736 (0.0002)	0.0301 (0.0002)	0.0169 (0.0003)	0.1065 (0.0003)
DeepSets MLP	0.0754 (0.0002)	0.0343 (0.0004)	0.0230 (0.0006)	0.1167 (0.0008)

DeepSets; other pipeline entries retain one run.

4.6 Sequential selection within a fixed setting budget

This is the closed-loop acquisition protocol introduced in Section 2.3.3. At budget b , a policy acquires b distinct settings, each once at 10^4 shots, and supplies the resulting partial record to the same frozen reconstructor. After each acquisition, the adaptive policy reconstructs the state and chooses the next unobserved setting; neither network is updated and settings are not remeasured. A human experimenter may perform the recommended measurement, or the loop may be automated by the instrument. Thus every policy uses the same total budget of $b \times 10^4$ state preparations.

The evaluation simulates this loop on all 45 999 test states. Deployable policies use only acquired measurements; the true state is used afterwards to calculate Bures distance. Four policies are compared: random selection, the fixed marginal-gain ordering π_{MG} , the distilled ordering π^* , and the history-dependent adaptive selector. All use the same fixed reconstructor, so their differences arise only from the settings they choose. Section 3.7 defines the policies and Attachment A.1.1 documents the reconstructor. Unequal setting informativeness motivates the comparison [38, 23, 12].

4.6.1 Deployable policy comparison

Sequential setting selection and trace-setting preference at 10^4 shots

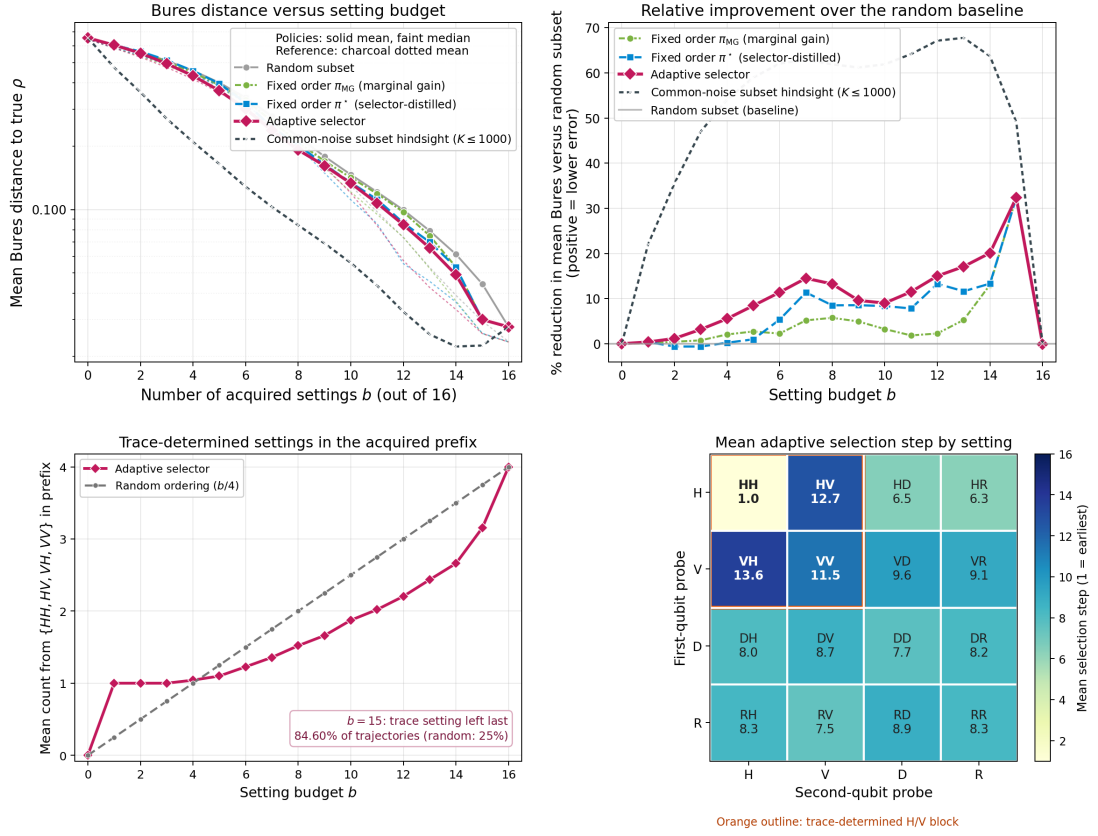


Figure 4.9: Four deployable policies using the same frozen reconstructor and the same number of distinct settings, each acquired once at 10^4 shots. *Top*: mean and median Bures distance and reduction in the mean relative to random subsets. The dotted curve is a noncausal common-noise subset reference. *Bottom*: acquisition of the trace-determined H/V settings and the mean adaptive selection step in the measurement grid. Statistics use all 45 999 test states.

Table 4.9: Mean Bures distance over 45 999 test states at budget b , with settings selected without replacement at 10^4 shots per observed setting. Bold marks the lowest deployable mean. The final column is the truth-selected best of up to 1000 subsets from one shared noisy record; it is exhaustive when $\binom{16}{b} \leq 1000$.

setting budget b	random	π_{MG}	π^*	adaptive	subset hindsight
2	0.560	0.558	0.564	0.554	0.361
4	0.457	0.448	0.456	0.432	0.210
6	0.340	0.332	0.322	0.301	0.128
8	0.221	0.209	0.202	0.192	0.084
10	0.146	0.142	0.134	0.133	0.056
12	0.100	0.097	0.086	0.085	0.033
14	0.061	0.053	0.053	0.049	0.022
15	0.044	0.030	0.030	0.030	0.023

At the reported precision, the adaptive selector is lowest or tied among deployable policies at every displayed budget. At fourteen settings it reaches 0.049, compared with 0.061 for random selection; random selection reaches 0.044 only after a fifteenth setting is acquired. Relative to the better fixed ordering, adaptation improves the mean by at most about seven percent, with smaller gains at most budgets. At fifteen settings, both fixed orderings and the adaptive policy reach 0.030.

The practical gain from adaptation is therefore modest but measurable. Adaptive selection is preferred when the acquisition controller can close the loop, while π^* provides a simple fixed schedule when feedback is impractical. This recommendation applies to distinct settings acquired without replacement at 10^4 shots per selected setting; allocating additional shots to an already measured setting is a different protocol.

4.6.2 Interpretation of the distilled ordering

The selector makes sufficiently consistent choices across validation states to define the fixed ordering below. With $XY = \mu_X \otimes \mu_Y$ for $X, Y \in \{H, V, D, R\}$ (Section 2.1.5),

$$\begin{aligned} \pi^* = \big(& HH, HD, HR, DH, \\ & RH, DD, DR, RR, \\ & RD, DV, RV, HV, \\ & VR, VD, VV, VH \big). \end{aligned} \tag{4.2}$$

The first six settings cover the Z , X and Y Pauli axes on both qubits (Section 2.1.3). The ordering begins with HH but places HV , VV and VH at positions 12, 15 and 16. It therefore uses one computational-basis setting as an initial anchor while deferring the remaining members of the H/V trace-completion group (2.21).

The full adaptive trajectories show the same pattern. The empty initial record is identical for every state, so the selector chooses HH first in every trajectory. At budget fifteen, the one setting left unmeasured belongs to $\{HH, HV, VH, VV\}$ in 84.60% of test trajectories, compared with 25% under random ordering. This is consistent with using trace completion near the end of the sequence. It does not imply that all four settings are redundant: only one becomes algebraically determined after the other three are known.

The marginal-gain ordering instead places HH last because it averages the value of a setting after random subsets have already been observed. Once the other three computational-basis settings are present, HH is nearly determined. Both fixed schedules consequently leave a trace-recoverable setting last, which helps explain their improvement from 0.044 for random omission to 0.030 at budget fifteen.

4.6.3 Noncausal headroom diagnostics

The following studies use the true state to rank counterfactual choices. They measure how much improvement remains possible, but they are not deployable acquisition policies.

Local second-action audit. After the selector’s fixed first choice HH , the mean Bures distance is 0.6048. Its learned second action lowers this to 0.5540, compared with 0.5599 for one uniformly random remaining action. The paired difference, selector minus random, is -0.00586 with a 95% interval from -0.00682 to -0.00491 . The learned action therefore provides a small but resolved one-step improvement. Selecting retrospectively among all fifteen possible second actions using the true state reaches 0.4334, demonstrating substantial but non-deployable local headroom.

Global subset headroom. The best-of- K study extends this comparison to complete size- b subsets. All candidates reveal values from one pre-drawn noisy record, so improvement comes from subset choice rather than from selecting a favourable noise draw. At budget twelve, the adaptive policy reaches 0.0846, whereas truth-selected candidate pools reach 0.0493 for $K = 20$ and 0.0327 for $K = 1000$. At budget sixteen there is only one possible subset, so every common-noise curve returns to 0.02764.

The separate independent-noise diagnostic redraws the measurements for every candidate. Its value can fall to about 0.007 at budget sixteen because it retrospectively selects the most favourable of 1000 noisy records; this is noise-selection hindsight, not setting-selection headroom. Common-noise best-of- K curves also need not decrease monotonically with budget because a new best subset is selected independently at each budget, rather than extending one sequential path.

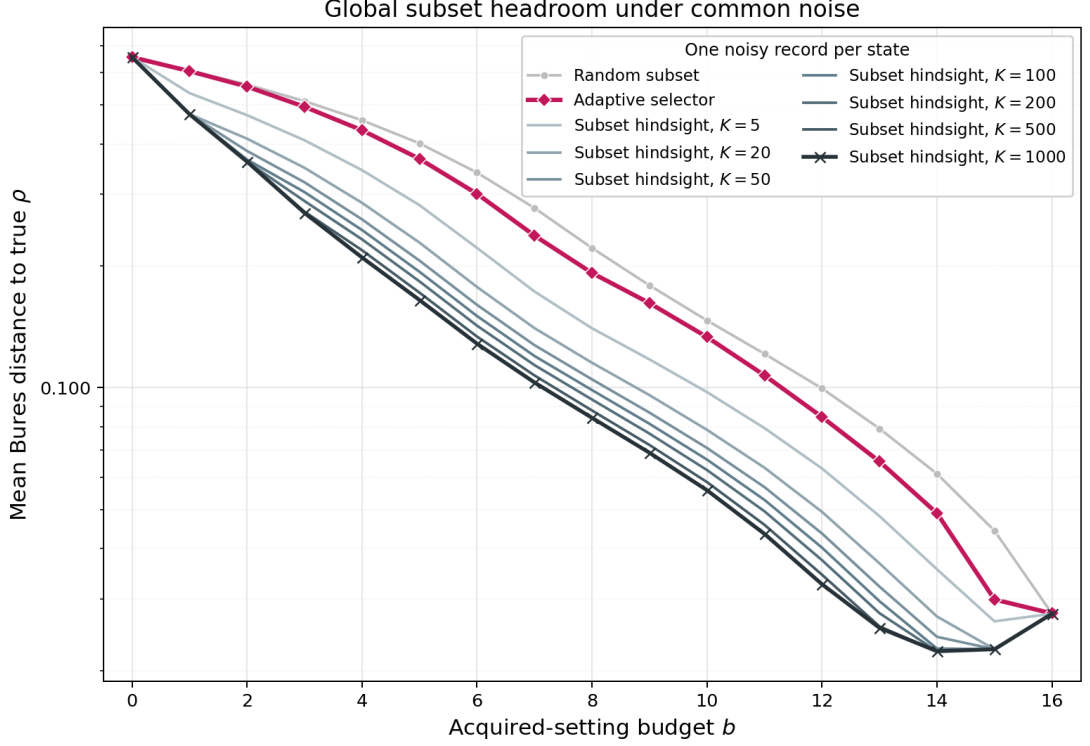


Figure 4.10: Deployable random and adaptive results together with noncausal best-of- K size- b subsets. For each test state, every candidate reveals settings from the same pre-drawn noisy record. Candidate pools are nested across K at each budget, but selected subsets need not form a sequential path across budgets.

4.7 Ablation of the measurement consistency loss

Equation (3.1) defines the confidence-weighted disagreement between reconstructed Born probabilities and observations. To test its effect, we compare three RhoTransformer retrainings that share architecture, noise, training schedule and seed:

- **All observed measurements:** $w = 50$ on every observed slot, as in the headline reconstructor.
- **Zero consistency:** $w = 0$, leaving only the regularised Bures loss. This is the selector and TTA reconstructor.
- **High-shot consistency:** $w = 50$ at 10^4 or more shots and $w = 0$ otherwise.

Only the consistency schedule differs. Figure 4.11 uses direct estimates; Section 4.8 examines augmentation.

Across both sweeps, the three curves remain within about three percent, comparable with the observed 1–3% run-to-run variation, so their single-run ordering is descriptive. Zero consistency is marginally lower in the lower-information region, while all-observed consistency is marginally lower at the information-rich

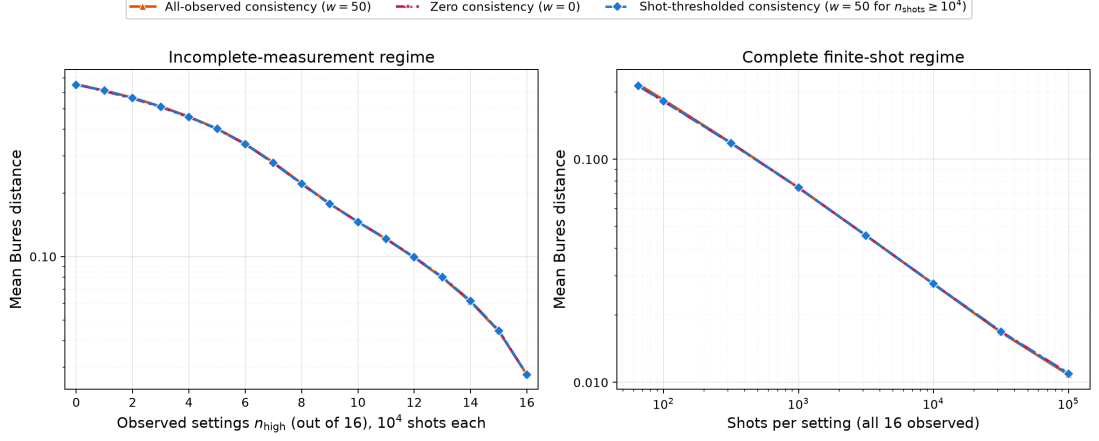


Figure 4.11: Mean Bures distance for three schedules: consistency on all observed settings, the zero-consistency variant, and consistency only at 10^4 or more shots. *Left:* incomplete measurements. *Right:* complete finite-shot measurements. Statistics use direct estimates on all 45 999 test states.

end. The consistency schedule is therefore a secondary design choice; zero consistency is the simplest broad default, while the principal checkpoint retains the schedule selected during its development.

4.7.1 Changes in implied measurement probabilities

For each slot we compare the noisy input m_s^{noisy} with the Born probability $\widehat{m}_s = \text{Tr}(\Pi_s \hat{\rho})$ implied by the reconstruction. Figure 4.12 plots $\widehat{m}_s - m_s^{\text{noisy}}$ for LOW (100 shots) and HIGH (10^4 shots) confidence bands. Because the LOW estimates have much larger finite-shot variance, the intended confidence-aware behaviour is to permit larger corrections there while leaving the more reliable HIGH estimates closer to their observed values.

Because the clean Born probabilities are available in the synthetic benchmark, Table 4.10 reports both the adjustment magnitude and the error before and after reconstruction. At 100 shots, every pipeline reduces the noisy-to-clean RMSE of about 0.041 to between 0.023 and 0.029, directly showing low-shot denoising. At 10^4 shots, the raw sampling error is already about 0.0041, below the reconstructed values' 0.0069–0.0133; the reconstruction error then exceeds the remaining sampling error. RhoTransformer is closest to the clean high-shot probabilities. These per-setting results complement the state-level Bures comparison.

The comparison below holds the RhoTransformer architecture fixed and varies only its consistency schedule.

Table 4.10: Per-slot RMSE on the shared 45 999-state test split. “Noisy-clean” measures $m_s^{\text{noisy}} - m_s^{\text{clean}}$, “prediction-clean” measures $\widehat{m}_s - m_s^{\text{clean}}$, and “adjustment” measures $\widehat{m}_s - m_s^{\text{noisy}}$. LOW and HIGH denote 100 and 10^4 shots per setting, respectively. Each row uses the same test states; the small differences between input RMSEs reflect separately generated binomial records.

Pipeline	LOW: 100 shots			HIGH: 10^4 shots		
	Noisy-clean	Prediction-clean	Adjustment	Noisy-clean	Prediction-clean	Adjustment
RhoTransformer	0.0408	0.0257	0.0263	0.0041	0.0069	0.0060
DeepSets MLP	0.0408	0.0295	0.0226	0.0041	0.0107	0.0103
Flow matching (transformer)	0.0408	0.0233	0.0347	0.0041	0.0084	0.0076
Flow matching (MLP)	0.0408	0.0247	0.0332	0.0041	0.0099	0.0093
cGAN (selected adversarial)	0.0407	0.0265	0.0328	0.0041	0.0133	0.0129

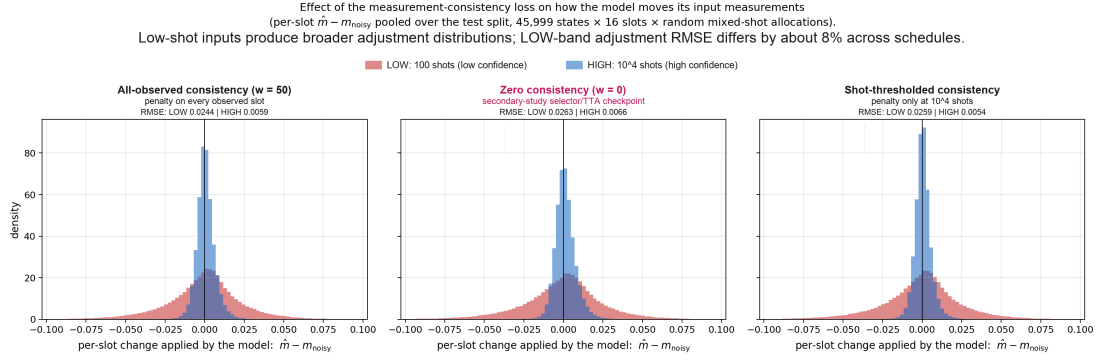


Figure 4.12: Per-slot change $\widehat{m}_s - m_s^{\text{noisy}}$ for the three schedules, split by confidence band (LOW: 100 shots; HIGH: 10^4 shots). Root-mean-square changes are annotated; statistics pool all sixteen slots over 45 999 test states.

LOW-band RMS change is about 0.025, versus 0.006 in HIGH, a four-to-five-fold ratio. All-observed consistency gives (0.024, 0.006), zero consistency (0.026, 0.007) and thresholded consistency (0.026, 0.005) for (LOW, HIGH), only about an 8% LOW-band spread. Thus the Bures loss alone already drives similar adjustment, consistent with the tightly clustered errors.

4.8 Test-time augmentation by qubit exchange

The TTA principle of Section 2.3.5, instantiated by the qubit-exchange procedure of Section 3.8, averages the direct reconstruction with a second prediction made after exchanging the input qubits and mapping the output back. The paired comparison shares states, subsets, shot counts and noise draws. The development-time gate fixed before this final comparison disables augmentation below twelve settings.

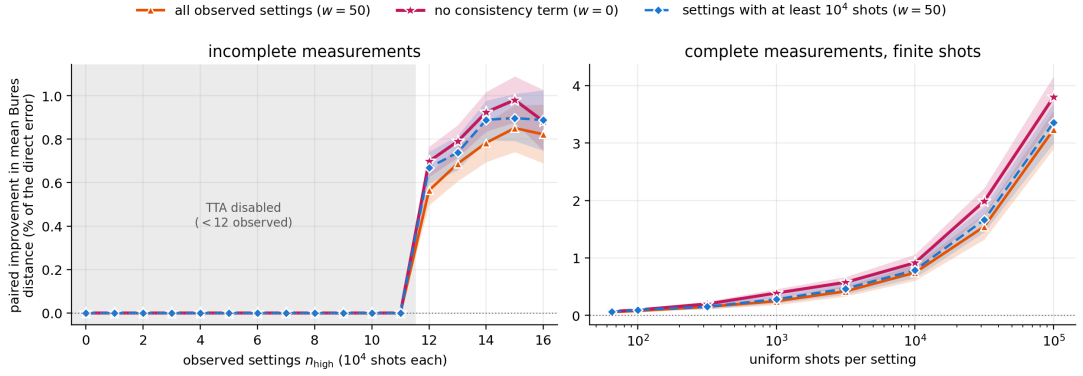


Figure 4.13: Percentage reduction in mean Bures distance from qubit-exchange TTA. *Left*: incomplete measurements, with augmentation disabled below twelve settings. *Right*: complete finite-shot measurements. Bands are paired 95% intervals; positive values indicate improvement. Absolute values are in Table 4.11.

Table 4.11: Qubit-exchange TTA for the zero-consistency fixed reconstructor on all 45 999 test states. Δ is direct minus augmented per-state Bures distance; positive values indicate improvement, brackets are paired 95% confidence intervals, and “Rel.” divides mean Δ by the direct mean.

Operating point	Direct	TTA	Mean Δ [95% CI]	Rel.	Improved / worsened	95th percentile
$n_{\text{high}}=12$, 10^4 shots	0.0995	0.0988	0.00069 [0.00062, 0.00076]	0.7%	47.0% / 36.3%	0.2871 $\rightarrow$ 0.2858
$n_{\text{high}}=14$, 10^4 shots	0.0616	0.0610	0.00057 [0.00051, 0.00063]	0.9%	45.5% / 34.2%	0.1916 $\rightarrow$ 0.1907
uniform 10^3 shots	0.0745	0.0743	0.00029 [0.00024, 0.00034]	0.4%	50.8% / 43.8%	0.1710 $\rightarrow$ 0.1700
uniform 10^5 shots	0.0110	0.0106	0.00042 [0.00038, 0.00046]	3.8%	43.1% / 26.6%	0.0388 $\rightarrow$ 0.0366

At every enabled point, the paired intervals exclude zero, but the mean gain is below one percent except for 3.8% at 10^5 uniform shots per setting. The three consistency variants behave similarly. Table 4.11 reports the absolute and paired diagnostics. Because TTA doubles the number of forward passes and helps mainly where direct estimates are already accurate, it is an optional two-pass refinement rather than part of the headline comparison.

5. Conclusion

5.1 Problem and approach

This thesis investigated whether restoration ideas from image processing and computer graphics can improve two-qubit quantum state tomography. The reconstruction problem contains two distinct sources of uncertainty: a measurement setting may be absent, and an observed setting may contain finite-shot sampling noise. The proposed formulation therefore represents each of the sixteen James–Kwiat probabilities together with its confidence and maps the resulting record to a physical 4×4 density matrix through the normalised Cholesky construction of Section 2.1.8.

The benchmark compares five selected implementations spanning feed-forward, conditional-adversarial, iterative-generative and attention-based reconstruction. All use fixed, paired evaluation records; training observations are generated online, while architectures, objectives, optimisers and inference procedures remain pipeline-specific. The final generative representatives are a conditional GAN and flow-matching models. Earlier iterative development paired DDPM with a transformer denoiser: its complete-setting validation mean flattened from 0.092 at 10^4 shots per setting to 0.088 at 10^5 despite 200 denoiser evaluations per estimate (Figure A.4), motivating the move to flow matching. Together, these experiments address the three questions posed in Section 1.3: the value of the confidence-aware physical formulation, the ranking of the selected pipelines and the usefulness of the reconstructor beyond a single point estimate.

5.2 Main findings

The results support three conclusions, ordered from the largest observed effect to the more specific architectural comparison.

1. **The formulation provides the largest improvement.** Replacing the naive raw-input MLP by confidence-aware features, set-based aggregation, a physical Cholesky output and the regularised Bures training objective reduces the complete high-shot mean Bures distance from 0.079 to 0.022. Because these components change together, the comparison measures the joint redesign rather than the isolated effect of any one component. This result also explains what transfers productively from graphics and image restoration. Missing settings act like masked observations, finite-shot fluctuations act like confidence-dependent noise, and the decoder restores a valid object under known physical constraints. The 4×4 arrangement is a tensor-product visualisation rather than an image with natural spatial locality, consistent with the positional-encoding result in Table 3.2.
2. **RhoTransformer leads the central comparison.** The selected RhoTransformer has the lowest mean Bures distance among the five centrally compared pipelines at every headline operating point. It reaches 0.015 with all sixteen settings at 10^5 shots per setting and 0.103 with twelve observed settings at 10^4 shots per observed setting. Each estimate uses one neural forward pass followed by the physical output map. Across three like-for-like

training runs, the RhoTransformer achieved a lower mean Bures distance than the DeepSets MLP at all four reported operating points.

The matched decoder comparison identifies a smaller but measurable architectural contribution. With the formulation and evaluation fixed, query self-attention lowers complete high-shot error by approximately ten percent relative to a larger attention-free SwiGLU decoder, while the two models remain similar when information is scarce. The resulting hierarchy is therefore clear: the confidence-aware physical formulation supplies the dominant gain, and query self-attention adds a further high-information refinement within the controlled pair. This is the bounded, benchmark-specific sense in which the “attention is all you need” interpretation applies here.

3. **The alternative pipelines clarify the ranking.** The flow-matching transformer is the lowest-error selected generative pipeline at every headline point, but its reported estimate requires approximately eighty velocity-network evaluations. The principal RhoTransformer is therefore both more accurate as a point estimator and less expensive under the present protocol. Within the GAN development path, a separately continued reconstruction-only implementation improves on the selected adversarial cGAN, although their optimisation histories differ. Among the implemented classical references, the gap is smallest when all settings are available and shot counts are high.

5.3 Secondary findings and practical meaning

The secondary studies answer different practical questions and are best read separately from the central architecture ranking.

- **Sequential acquisition.** The frozen RhoTransformer can serve as the state estimator in a closed laboratory loop: after each acquisition it updates the state estimate, and the frozen selector chooses one remaining unobserved setting. Repetition produces a schedule without replacement or retraining; a human experimenter or an instrument controller can execute each recommendation. At fourteen settings and 10^4 shots per observed setting, the learned policy lowers mean Bures distance from 0.061 under random selection to 0.049.

The stricter common-noise second-action audit finds a smaller local effect: after the same first acquisition, the learned second action reaches 0.5540, compared with 0.5599 for a random remaining action. One-step hindsight reaches 0.4334, showing substantial selection headroom, although it is not a deployable policy because it uses the unknown true state.

- **Trace-aware completion.** In one-based row-major order, HH , HV , VH and VV occupy measurement-grid indices 1, 2, 5, 6. Their probabilities sum to one because their projectors resolve the two-qubit identity. If exactly one is absent, its exact probability is therefore determined by the other three and the unit-trace condition.

The finite-shot experiment shows why algebraic completion must be accompanied by uncertainty. At 10^5 shots per observed setting, explicit completion removes 94% of the excess error caused by masking one of these settings. At 10^3 shots per observed setting, masking remains preferable because the

derived value accumulates noise from three measured probabilities. An estimator should therefore mark the value as derived and propagate its larger variance instead of treating it as an independent measurement.

- **Optional refinements.** Qubit-exchange test-time augmentation can improve the estimate at the cost of a second neural forward pass. The three measurement-consistency schedules differ by at most approximately three percent, comparable to run-to-run variation. Neither refinement changes the central ranking.

5.4 Scope of the conclusions

The conclusions concern synthetic two-qubit states drawn from the same generator family for training and evaluation, with independent binomial noise on shared, fixed evaluation records. In incomplete regimes, a learned estimator combines the retained observations with information learned from the training distribution.

Seed variation is measured for RhoTransformer and DeepSets; the other pipeline results use one retained run, and the attention result belongs to the matched decoder pair. The classical likelihood reference is the documented fixed-budget Cholesky optimisation, while the generative results use the point-estimation rules of Section 3.8.

The Bures-based learned objectives use the numerical regularisation documented in Attachment A.1.5. The optional post-Cholesky safeguard is used during principal-model training and checkpoint selection but disabled for the final common-noise sweep, with negligible aggregate effect (Attachment A.1.5). Generalisation to distribution shift and device effects remains to be established experimentally.

5.5 Future work

The next stage should concentrate on strengthening and extending RhoTransformer. The priorities are:

1. **Establish training robustness.** Repeat the final RhoTransformer, DeepSets and matched-decoder experiments over more random seeds, using the same paired test records for every comparison. This would separate persistent architectural effects from variability caused by initialisation and optimisation.
2. **Evaluate on experimental quantum hardware.** The frozen reconstructor should first be tested without fine-tuning on two-qubit data collected from IBM Quantum devices [71]. Device calibration, readout error, drift, circuit layout and acquisition time should be recorded so that the difference from synthetic data can be interpreted. Hardware-specific adaptation should follow only after this zero-shot transfer has been measured.
3. **Simplify and stabilise the loss calculation.** The network currently predicts sixteen Cholesky coordinates, converts them to a density matrix and then evaluates the regularised Bures loss. This objective compares the reconstructed state directly with the target state, but its matrix square roots make the calculation relatively expensive and numerically delicate

when eigenvalues are very small. The bound in Equation (2.38) suggests an alternative: compare the predicted and target Cholesky factors before constructing the density matrices. Such a comparison would avoid the matrix square roots, but a raw coordinate loss is not yet well defined. Multiplying every entry of a factor by a nonzero real constant leaves the normalised density matrix unchanged, and changing the sign of an entire factor column also leaves the represented state unchanged. Two different coordinate vectors can therefore describe exactly the same density matrix. Future work should either choose one unique factor for every state or define a loss that treats equivalent factors as equal. The resulting method should then be compared with the present Bures objective and with likelihood objectives in terms of reconstruction error, gradient stability and runtime.

4. **Develop the input features further.** Matched ablations should test physically meaningful features while holding the decoder, physical output map, loss and schedule fixed. Candidates include Pauli-product summaries, explicit observed-setting counts, binomial variance estimates, raw count information and hardware calibration metadata. This would identify which additions improve the estimator rather than merely changing the complete pipeline.
5. **Make measurement selection trace-aware.** The acquisition policy should recognise explicitly that HH, HV, VH, VV —indices 1, 2, 5, 6—form the unit-trace quadruple. Once three are available, the experimenter or controller could measure the fourth, complete it algebraically with propagated uncertainty, or allocate more shots to reduce the variance of an existing measurement. High-shot completion may save an acquisition; at low shots, direct measurement or remeasurement can be preferable. Testing these choices under a fixed total copy budget would connect the identifiability result directly to a practical acquisition policy. The policy should also be compared with repeated-draw, multi-step and recurrent selectors [12].
6. **Scale beyond two qubits.** For n qubits, the density-matrix representation grows as 4^n together with a rapidly expanding measurement space. Experiments on three and more qubits should measure accuracy, memory, inference cost and acquisition cost, and should test structured, low-rank or symmetry-aware output parameterisations when the full representation becomes impractical.

Secondary work can strengthen the surrounding benchmark. Classical references should include a tolerance-based projected likelihood solver and a trace-constrained, binomially weighted estimator. Generative evaluation should compare latent-averaged cGAN estimates, alternative flow sample counts, state-space Bures barycentres, conditional calibration and posterior coverage.

The central conclusion is that the confidence-aware formulation and physical output map produce the largest improvement. Within that foundation, Rho-Transformer is the lowest-error centrally benchmarked point estimator, and its query self-attention supplies a smaller additional high-shot benefit. The resulting model is a strong basis for real-device validation, larger quantum systems, more stable objectives and trace-aware adaptive tomography.

Bibliography

- [1] Max Lytvyn, Dmytro Lider, and Alex Shevchenko. Grammarly. <https://app.grammarly.com/>, 2026. Online writing assistant, accessed July 2026.
- [2] GitHub. GitHub Copilot documentation. <https://docs.github.com/en/copilot>, 2026. Accessed: 2026-07-15.
- [3] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.
- [4] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.
- [5] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11461–11471, 2022.
- [6] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.
- [7] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2Noise: Learning image restoration without clean data. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 2965–2974, 2018.
- [8] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- [10] Z. Hradil. Quantum-state estimation. *Phys. Rev. A*, 55:R1561–R1564, Mar 1997.
- [11] Daniel F. V. James, Paul G. Kwiat, William J. Munro, and Andrew G. White. Measurement of qubits. *Phys. Rev. A*, 64:052312, Oct 2001.
- [12] Mateusz Krawczyk, Pavel Baláž, Katarzyna Roszak, and Jarosław Pawłowski. Learning quantum tomography from incomplete measurements. arXiv preprint arXiv:2506.19428v4, 2025.

- [13] Adriano Macarone Palmieri, Guillem Müller-Rigat, Anubhav Kumar Srivastava, Maciej Lewenstein, Grzegorz Rajchel-Mieldzioć, and Marcin Płodzień. Enhancing quantum state tomography via resource-efficient attention-based neural networks. *Phys. Rev. Research*, 6:033248, Sep 2024.
- [14] Onur Danaci, Sanjaya Lohani, Brian T. Kirby, and Ryan T. Glasser. Machine learning pipeline for quantum state estimation with incomplete measurements. *Machine Learning: Science and Technology*, 2(3):035014, 2021.
- [15] Hailan Ma, Daoyi Dong, Ian R. Petersen, Chang-Jiang Huang, and Guo-Yong Xiang. Neural networks for quantum state tomography with constrained measurements. *Quantum Information Processing*, 23(9):317, 2024.
- [16] Yu-Zhuo Li, Li-Chao Peng, and Ke-Mi Xu. Learning entanglement quasiprobability from noisy and incomplete data. arXiv preprint arXiv:2603.18414, 2026.
- [17] Peter Cha, Paul Ginsparg, Felix Wu, Juan Carrasquilla, Peter L. McMahon, and Eun-Ah Kim. Attention-based quantum tomography. *Machine Learning: Science and Technology*, 3(1):01LT01, 2022.
- [18] Hailan Ma, Zhenhong Sun, Daoyi Dong, Chunlin Chen, and Herschel Rabitz. Tomography of quantum states from structured measurements via quantum-aware transformer. arXiv preprint arXiv:2305.05433, 2023.
- [19] Hailan Ma, Zhenhong Sun, Daoyi Dong, and Dong Gong. Learning informative latent representation for quantum state tomography. arXiv preprint arXiv:2310.00518, 2023.
- [20] Lu Zhong, Chu Guo, and Xiaoting Wang. Quantum state tomography inspired by language modeling. arXiv preprint arXiv:2212.04940, 2022.
- [21] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [22] Ingemar Bengtsson and Karol Życzkowski. *Geometry of Quantum States: An Introduction to Quantum Entanglement*. Cambridge University Press, 2 edition, 2017.
- [23] Pavel Baláž, Mateusz Krawczyk, Jarosław Pawłowski, and Katarzyna Roszak. Learning entanglement from tomography data: Contradictory measurement importance for neural networks and random forests. *Phys. Rev. A*, 112:022431, Aug 2025.
- [24] Roger Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, 1955.
- [25] Mădălin Guță, Jonas Kahn, Richard Kueng, and Joel A. Tropp. Fast state tomography with optimal error bounds. *Journal of Physics A: Mathematical and Theoretical*, 53(20):204001, 2020.

- [26] A. W. van der Vaart. *Asymptotic Statistics*, volume 3 of *Cambridge Series in Statistical and Probabilistic Mathematics*. Cambridge University Press, 1998.
- [27] Eliot Bolduc, George C. Knee, Erik M. Gauger, and Jonathan Leach. Projected gradient descent algorithms for quantum state tomography. *npj Quantum Information*, 3:44, 2017.
- [28] Donald Bures. An extension of Kakutani’s theorem on infinite product measures to the tensor product of semifinite w^* -algebras. *Transactions of the American Mathematical Society*, 135:199–212, 1969.
- [29] Armin Uhlmann. The “transition probability” in the state space of a $*$ -algebra. *Reports on Mathematical Physics*, 9(2):273–279, 1976.
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, 2019.
- [31] William K. Wootters. Entanglement of formation of an arbitrary state of two qubits. *Phys. Rev. Lett.*, 80:2245–2248, Mar 1998.
- [32] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.
- [33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.
- [34] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [35] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [36] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.
- [37] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318. PMLR, 2013.
- [38] Yihui Quek, Stanislav Fort, and Hui Khoon Ng. Adaptive quantum state tomography with neural networks. *npj Quantum Information*, 7(1):105, 2021.

- [39] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). arXiv preprint arXiv:1606.08415, 2016.
- [40] Stefan Elfving, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.
- [41] Prajit Ramachandran, Barret Zoph, and Quoc V. Le. Searching for activation functions. arXiv preprint arXiv:1710.05941, 2017.
- [42] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256. PMLR, 2010.
- [43] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323. PMLR, 2011.
- [44] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [45] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.
- [46] Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 933–941. PMLR, 2017.
- [47] Noam Shazeer. GLU variants improve transformer. arXiv preprint arXiv:2002.05202, 2020.
- [48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [49] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.
- [50] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059. PMLR, 2016.

- [51] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- [52] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep sets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
- [53] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 27, 2014.
- [54] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014.
- [55] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GAIN: Missing data imputation using generative adversarial nets. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5689–5698. PMLR, 2018.
- [56] Shahnawaz Ahmed, Carlos Sánchez Muñoz, Franco Nori, and Anton Frisk Kockum. Quantum state tomography with conditional generative adversarial networks. *Phys. Rev. Lett.*, 127:140502, Sep 2021.
- [57] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.
- [58] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 8162–8171, 2021.
- [59] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.
- [60] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.
- [61] Mateusz Krawczyk, Pavel Baláž, Katarzyna Roszak, and Jarosław Pawłowski. Learning quantum tomography from incomplete measurements. Zenodo dataset, Oct 2025. Version v1, 6 October 2025. <https://doi.org/10.5281/zenodo.17276583>.
- [62] Jarosław Pawłowski and Mateusz Krawczyk. Identification of quantum entanglement with Siamese convolutional neural networks and semisupervised learning. *Phys. Rev. Applied*, 22:014068, Jul 2024.

- [63] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.
- [64] Lloyd S. Shapley. A value for n -person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, Princeton, 1953.
- [65] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.
- [66] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, 2020.
- [67] Wes McKinney. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, pages 56–61, 2010.
- [68] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake VanderPlas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.
- [69] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.
- [70] Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit. arXiv preprint arXiv:2405.08810, 2024.
- [71] IBM Quantum. IBM Quantum Platform: Compute resources and processor types. <https://quantum.cloud.ibm.com/computers>; <https://quantum.cloud.ibm.com/docs/guides/processor-types>, 2026. Accessed: 2026-07-10.

List of Figures

1.1	The measurement grid and the restoration analogy.	4
2.1	Confidence encoding: logarithmic and no-log alternatives.	25
2.2	Supervised training and inference.	28
2.3	Sequential active measurement selection.	31
2.4	Common activation and gate functions.	32
2.5	MLP and DeepSets computation.	35
2.6	Attention and its RhoTransformer specialisation.	37
2.7	Conditional-GAN training and inference.	39
2.8	DDPM and flow-matching paths.	42
3.1	Entanglement distribution and layout of the dataset.	45
3.2	The common reconstruction interface.	47
3.3	Early development comparison of reconstruction losses.	51
3.4	RhoTransformer architecture.	56
3.5	Convergence of the noisy hindsight reference with the pool size. .	61
4.1	Controlled single-token decoder comparison.	67
4.2	Learned reconstructor versus classical baselines.	68
4.3	Density-matrix reconstruction from incomplete measurements. . .	70
4.4	Context against published incomplete-measurement reconstruction.	73
4.5	Density-matrix reconstruction under finite-shot noise.	74
4.6	Reconstruction error versus spectral position of the true state. . .	75
4.7	Qualitative reconstruction examples.	76
4.8	Training convergence across pipelines.	78
4.9	Sequential selection within a fixed setting budget.	80
4.10	Global common-noise subset headroom.	83
4.11	Ablation of the measurement consistency loss.	84
4.12	Per-slot change of input measurements by loss variant.	85
4.13	Paired effect of qubit-exchange augmentation.	86
A.1	DeepSets MLP architecture.	108
A.2	Conditional-GAN architecture.	108
A.3	Flow-matching architecture.	109
A.4	Development-only DDPM shot sweep.	111

Tables and Abbreviations

List of Tables

2.1	Conceptual comparison of the learned families.	43
3.1	Fourier feature-resolution sweep.	50
3.2	Validation ablation of the setting-identity encoding.	50
3.3	Components used by the reported models.	52
4.1	Comparison across reconstruction pipelines (mean Bures distance). . . .	66
4.2	Controlled single-token decoder comparison.	67
4.3	Selected pipelines with twelve near-clean measurements.	71
4.4	Single-setting omission and explicit trace completion.	72
4.5	Measured inference time of selected methods.	76
4.6	Agreement between reconstructed and true concurrence.	77
4.7	Downstream entanglement-classifier accuracy.	78
4.8	Three-seed variability of the principal direct models.	79
4.9	Setting-selection policies with incomplete measurements.	80
4.10	Cross-pipeline denoising of observed measurements.	85
4.11	Paired diagnostics for qubit-exchange augmentation.	86
A.1	Chronology of the reported checkpoints and their split roles.	101
A.2	Per-pipeline architecture and training configuration.	107
A.3	Binary concurrence-decision diagnostics.	110
A.4	Development-only cGAN input ablation.	111
A.5	Development checks of training, output and inference alternatives. . . .	112
A.6	Development history of the reconstruction models.	113

List of Abbreviations

BCE	Binary cross-entropy		
cGAN	Conditional generative adversarial network		
CI	Confidence interval	LSTM	Long short-term memory
CN	Complete noisy measurement regime	MAE	Mean absolute error
CPU	Central processing unit	MC	Monte Carlo
DDPM	Denoising diffusion probabilistic model	MHA	Multi-head attention
DiT	Diffusion transformer	MLE	Maximum likelihood estimation
EMA	Exponential moving average	MLP	Multilayer perceptron
FFN	Feed-forward network	MoE	Mixture of experts
GAIN	Generative adversarial imputation nets	MSE	Mean squared error
GAN	Generative adversarial network	NN	Neural network
GELU	Gaussian error linear unit	ODE	Ordinary differential equation
GPU	Graphics processing unit	PCA	Principal component analysis
HS	Hilbert–Schmidt	PSD	Positive semidefinite
IN	Incomplete noisy measurement regime	QST	Quantum state tomography
INC	Incomplete near-clean measurement regime	ReLU	Rectified linear unit
IQR	Interquartile range	RMSE	Root-mean-square error
KL	Kullback–Leibler divergence	SiLU	Sigmoid linear unit
LI	Linear inversion	SwiGLU	Swish-gated linear unit
LN	Layer normalisation	t-SNE	t-distributed stochastic neighbour embedding
		TTA	Test-time augmentation

A. Configuration of the reported models and development history

A.1 Configuration of the reported models

This attachment records the available configuration of every learned model reported in Chapter 4. All values were extracted from the stored training checkpoints of the reported models.

A.1.1 The principal reconstructor in detail

Figure 3.4 (on page 56, in the methodology chapter) depicts the complete data flow of the RhoTransformer; its components are described below in the order in which the data traverses them.

- **Input encoding.** Each of the sixteen James–Kwiat projector probabilities m_p is lifted to a seventeen-component Fourier feature vector containing m_p and the pairs $\sin(2\pi km_p)$, $\cos(2\pi km_p)$ for $k = 1, \dots, 8$, and multiplied by its confidence c_p from Equation (2.41), following the feature map in Equation (2.42), so that a missing slot contributes exactly the zero vector. The sixteen gated feature vectors are concatenated and mapped by a single linear projection to one 256-dimensional token; slot identity is carried by the distinct column blocks of this projection.
- **Encoder.** Four pre-norm blocks operating on that single token, each consisting of layer normalisation, a dense 256×256 linear map with a residual connection, then layer normalisation and a SwiGLU feed-forward block from Equation (2.57) (hidden width 512) with a residual connection. The single-token encoder implements a dense linear map, exactly matching the length-one reduction of multi-head self-attention in Equation (2.68); Section 3.5 explains the resulting architectural trim.
- **Decoder.** Sixteen learned 256-dimensional query embeddings, one per output parameter, processed by three layers. Each layer applies (i) an eight-head self-attention over the sixteen queries, the only non-degenerate attention in the network; (ii) an additive conditioning bias, a linear map of the encoder summary added to every query (the exact rewriting of cross-attention over a single key, whose softmax is identically one); and (iii) a SwiGLU feed-forward block (hidden width 512). Dropout 0.1 is applied throughout.
- **Output head.** A per-query linear readout produces sixteen real numbers, assembled into the lower-triangular complex factor T as in Equation (2.29) (four real diagonal entries and six complex off-diagonal entries), and the reconstruction is $\rho = TT^\dagger / \text{Tr}(TT^\dagger)$ (2.28), positive semidefinite and unit-trace by construction in exact arithmetic. The diagonal outputs are unconstrained real numbers rather than a nonnegative Cholesky convention,

so sign-equivalent factors represent the same density matrix while preserving physicality. The final training run used the optional finite-precision safeguard of Attachment A.1.5 after this map.

The active decoder path implements the single-key identity directly: each layer applies one dense broadcast additive-conditioning map to the encoder summary and broadcasts the result over the sixteen queries. An implementation smoke test forms this map as $W_O W_V$ with the corresponding folded bias and matches the single-key attention output to floating-point precision. The evaluated forward pass therefore uses only the folded value and output maps.

For compatibility with historical checkpoint keys, the archived model object still registers the bypassed cross-attention tensors. It consequently contains 3 980 545 stored parameters, while the active forward path contains about 3.69 million. The parameter column of Table A.2 reports the stored count so that the architecture record refers to the exact evaluated checkpoint; the corresponding timing measurements use the active forward path of that checkpoint. Exporting a checkpoint with the dormant tensors removed would preserve its predictions and create a distinct stored artifact.

Controlled decoder pair. The auxiliary comparison of Section 4.1.2 uses two recipe-matched models distinct from the principal checkpoint (3.98 million stored; about 3.69 million active). Both use the confidence-scaled Fourier input, a four-layer single-token SwiGLU encoder, the Cholesky output head, the measurement-consistency objective, the same training-noise mixture and the same 500-epoch schedule. One member uses a query self-attention decoder; the other replaces that decoder with an attention-free SwiGLU feed-forward decoder. The complete stored models contain 5.62 and 6.42 million parameters, respectively. The larger control rules out a simple advantage from having more total parameters in the query-attention model. Its feed-forward replacement necessarily uses a different decoder parameter allocation and functional form. Table 4.2 reports their fixed-test results.

Secondary-study RhoTransformer. The selector, consistency-loss and TTA studies use a separately developed secondary-study RhoTransformer checkpoint that is held fixed within those comparisons. It has model width 160, eight heads, ten no-mix SwiGLU encoder blocks, five query-self-attention decoder layers, feed-forward width 512, eight Fourier frequencies, a per-setting noise-standard-deviation channel and the direct Cholesky head. Its matched variants train for 250 epochs; the selector and TTA use the $w = 0$ zero-consistency checkpoint. On the shared test records, that checkpoint gives 0.075, 0.028 and 0.011 at 10^3 , 10^4 and 10^5 uniform shots per setting and 0.100 with twelve settings at 10^4 shots per observed setting, compared with 0.073, 0.029, 0.015 and 0.103 for the principal checkpoint. The two checkpoints exchange rank across regimes: the principal model is lower at 10^3 shots, while the secondary model is lower at the information-rich operating points. The term *principal* identifies the preselected, saturation-tested checkpoint used for the central cross-family benchmark; the separately developed checkpoint is the fixed platform for the internally matched secondary experiments. Section 4.2 states this distinction in the main text; the chronology below documents it.

A.1.2 Checkpoint chronology and split roles

Table A.1 records the order in which the reported checkpoints were produced, the rule that selected each one and the role of the fixed test split. Fitting, early stopping and model selection use training and validation data. The test split is excluded from these procedures and supplies the reported evaluation after the relevant checkpoint or policy has been fixed.

Table A.1: Chronology of the reported checkpoints. Approximate periods summarise the preserved experiment reports and repository history. Fitting, stopping and selection use training and validation data; the fixed test split supplies the reported evaluations after each relevant checkpoint or policy is frozen.

Period (2026)	Checkpoints and role	Selection rule	Reported test evaluation
to mid-May	single-change development rounds, including the Cholesky-head and corrector variants of Table A.6 and the architecture of the principal reconstructor	validation Bures under each round’s protocol; one change retained per round	reported after validation selection
mid-May	controlled decoder pair (query-attention and attention-free, Table 4.2)	matched 500-epoch schedule; both runs reported	common sweep reported after both runs were frozen
late May	secondary-study platform: three matched consistency-weight retrainings (the w sweep of Section 4.7); the $w = 0$ member later serves the selector and TTA studies	validation Bures; the platform is then frozen	study sweeps reported for the frozen platform
mid-June	central selected-pipeline campaign: the five selected implementations and two naive baselines of Table 4.1, including the principal plateau checkpoint	pre-specified reduce-on-plateau rule; best-validation checkpoint of each run	paired central benchmark reported for the frozen checkpoints
June/July	selector, fixed orderings, noisy-hindsight and TTA studies on the frozen $w = 0$ platform	orderings and the augmentation gate fixed before the final paired test evaluation	paired results reported after the policies were frozen
early July	seed-variability runs (seeds 43 and 44, Table 4.8)	none: all runs reported	common test records used for the dispersion table

A.1.3 Training objective

The principal reconstructor minimises

$$\mathcal{L}(\theta) = d_B(\hat{\rho}, \rho) + w \cdot \frac{1}{16} \sum_{p=1}^{16} c_p \left(\text{Tr}(\Pi_p \hat{\rho}) - m_p^{\text{obs}} \right)^2, \quad w = 50, \quad (\text{A.1})$$

with the Bures spectral subcomputation evaluated in complex double precision (Section 2.2.1) and the consistency term matching the *noisy* observations, weighted by the per-slot confidence. Three design decisions concerning (A.1) merit explicit documentation. First, the consistency target is the noisy observation m_p^{obs} , matching the information available in deployment and the objective of Macarone Palmieri et al. [13]. Second, the weight w was swept over $\{0, 18, 50, 100, 250, 500\}$ (the $w = 0$ endpoint is the zero-consistency variant of Section 4.7); $w = 50$ was optimal, and it keeps the consistency contribution between roughly 0.1 and 2.5 times the Bures contribution over a training run under this fixed weighting. Third, the ablation of Section 4.7 finds a second-order effect across the swept weights. This confidence-weighted squared residual is a heuristic consistency penalty whose c_p acts as a logarithmic regime indicator; the likelihood baseline separately models binomial sampling.

A.1.4 Training configuration and direct-model noise mixture

The principal RhoTransformer was trained in two stages. The first began from random initialisation and ran for 1 000 epochs without early stopping. It used AdamW [36] with target learning rate 10^{-3} , a ten-epoch linear warm-up and cosine annealing towards 10^{-7} . The selected saturation continuation loaded the best exponential-moving-average weights from that stage while starting a fresh optimiser and learning-rate scheduler. Its target learning rate was 1.5×10^{-4} , with a five-epoch warm-up followed by reduction on a validation plateau: the rate was multiplied by 0.3 after 50 epochs without an absolute Bures-distance improvement of at least 5×10^{-4} , down to a floor of 10^{-7} . The continuation was capped at 2 000 epochs and stopped after 300 consecutive validation epochs without a new best checkpoint.

Both stages used mini-batches of 2 048, dropout 0.1, AdamW decoupled weight decay 0.01 as defined by Equation (2.47), exponential-moving-average (EMA) decay 0.999 and random seed 42. After each backward pass, the shared RhoTransformer–DeepSets training loop clips the global ℓ_2 norm of all trainable-parameter gradients to 1.0 before the AdamW step, following Equation (2.48). Weight decay is therefore an optimiser update rather than an L2 term added to the reconstruction loss. The selected cGAN, flow, naive-transformer and vanilla-MLP trainers step without gradient clipping and retain their separately documented optimisation settings.

The network trains in single precision; the Bures routine promotes its spectral matrix operations to complex double precision and returns the scalar loss in single precision. The selected DeepSets checkpoint used the same plateau-continuation settings after its own precursor. The cGAN uses a 1 500-epoch cap, early-stopping patience 60 and scheduler patience 15; both flow models use a 2 000-epoch cap, early-stopping patience 60 with validation every five epochs, and scheduler patience 10. The five headline reconstructors draw training states from the base pool of 413 999 states. The secondary-study RhoTransformer and measurement selector draw training states from the fourfold enlarged pool of Section 3.1; all reported reconstruction values use the fixed validation or test partition identified in their tables.

Each training row draws its noise independently per slot: with probability 0.15 the slot is missing (shot count zero, confidence zero), otherwise its shot count is drawn log-uniformly from $[1, 10^5]$ and the observation is binomial-sampled (2.12). Independently, 25% of the rows are complete uniform-shot anchor rows in which all sixteen slots share a single shot count drawn log-uniformly from $[10, 10^5]$ (Section 3.2); at the low end of that range such a row is complete and strongly shot-noisy. This two-component mixture is the common online observation recipe of the direct RhoTransformer–DeepSets comparison. The cGAN and flow models instead use two independently sampled anchor switches, each with probability 0.15. The complete-anchor switch uses the uniform-shot row just described. The sparse-anchor switch retains one to four randomly selected settings, each at 10^4 shots, and makes all other settings missing; it takes precedence if both switches select the same row. Every training run draws fresh noise realisations, while the final cross-pipeline comparison uses paired test states, observed subsets, shot counts and binomial draws.

A.1.5 Numerical details of the Cholesky and Bures computations

Finite-precision safeguard after the Cholesky map. The principal RhoTransformer run disables automatic mixed precision: its network output is single precision, and $TT^\dagger$ is assembled in complex single precision, re-Hermitised and normalised by its trace. This map is PSD and unit-trace in exact arithmetic; the training run additionally enables a numerical safeguard. It draws a fresh real matrix with independent Gaussian entries at scale 10^{-6} , symmetrises that matrix, adds it to the Cholesky output, and then applies the symmetrisation, eigendecomposition, clipping, trace-renormalisation and reassembly core $\mathcal{P}_{\text{clip}}$ of Algorithm 2.1. The perturbation makes this safeguard a stochastic extension of $\mathcal{P}_{\text{clip}}$. For the principal direct model, the resulting matrix enters the Bures and measurement-consistency losses, validation and checkpoint selection, and the projection remains in the automatic-differentiation graph. The cGAN does not enable this safeguard at any stage: during both training and evaluation, its predicted factor passes only through the deterministic normalised Cholesky map. The final common-noise evaluation of the direct models likewise forces the optional safeguard off and reports their deterministic Cholesky outputs; the flow implementation retains its separate endpoint projection. Thus only the principal direct model’s training path adds this stochastic safeguard and its 4×4 eigendecomposition; cGAN training and the final direct-model and cGAN reporting paths do not.

We checked the distinction directly on 3 000 validation states in four noise regimes for each of two Cholesky checkpoints, giving 24 000 prediction–regime pairs. Depending on the regime, only 0 to 2.4% of raw Cholesky matrices had any computed negative eigenvalue, none had an eigenvalue below -10^{-6} and the worst value was -1.5×10^{-7} ; the mean absolute trace deviation was approximately 3×10^{-8} . Because the safeguard injects the small perturbation even for an already valid matrix, it generally moves the output slightly: the mean Bures distance between the raw and safeguarded matrices was 5.5 to 6.4×10^{-4} and the maximum was 2.7×10^{-3} . Its effect on the quantity of interest was much smaller: the

change in mean Bures distance to the true state ranged from -2.0×10^{-6} to $+2.4 \times 10^{-6}$, and it improved between 48.7% and 51.1% of individual predictions. The aggregate effect of retaining this safeguard is therefore far below the precision of the reported values. The script `analyze_psd_impact.py` in the experimental repository reproduces this diagnostic.

Training-loss eigendecompositions. Separately from the Cholesky-output safeguard, a dedicated differentiable Bures routine uses the standard automatic-differentiation machinery in PyTorch [30]; no custom backward or iterative matrix-square-root derivative is defined. The training routine first casts the prediction and target to complex double precision and re-Hermitises the prediction. A fresh real matrix with independent Gaussian entries is then symmetrised and added only to $\hat{\rho}$, immediately before the first Hermitian eigendecomposition. The eigenvalues of this perturbed matrix are floored at 10^{-14} before taking their square roots. The intermediate matrix $\sqrt{\hat{\rho}}\rho\sqrt{\hat{\rho}}$ is re-Hermitised, its eigenvalues are clamped at zero, the resulting fidelity is clamped to $[0, 1]$ and the radicand of d_B^2 is clamped at zero. The routine then returns d_B^2 in single precision and, for the selected unsquared objective, takes its final scalar square root in single precision.

These operations address different numerical hazards. Complex double precision reduces round-off and cancellation; the eigenvalue floor controls the square-root boundary; and the random perturbation almost surely separates an exact repeated spectrum, stabilising the complex-eigenvector backward pass. Double precision alone does not remove a true degeneracy. Conversely, the post-Cholesky projection restores numerical physicality but does not by itself define a stable derivative of the matrix square root. The training perturbation is therefore a fixed-basis stochastic regularisation of the loss; it is not part of the Bures definition.

The perturbation has one fixed numerical hyperparameter, its Gaussian scale σ . Before symmetrisation, each matrix entry is sampled independently from $\mathcal{N}(0, \sigma^2)$; the final implementation uses $\sigma = 10^{-4}$. This value belongs to the differentiable training-loss routine: it is neither learned by the model nor part of the Bures distance. The diagnostic below applies the same regularisation procedure to stored predictions solely to quantify its numerical effect. The direct models, the cGAN reconstruction term and the flow model’s auxiliary Bures term all use this routine.

The reported learned-model metric is evaluated without gradients and omits this training-loss perturbation. It uses the common complex-single-precision path, a 10^{-12} floor for the first spectrum and a zero clamp for the intermediate spectrum. In the final common-noise campaign, the direct-model and cGAN evaluation also forces the separate post-Cholesky safeguard off; their reported outputs are therefore the raw deterministic Cholesky map followed by this no-gradient metric. The flow endpoint projection is part of its own sampling procedure rather than this optional conversion-stage safeguard.

The effect of this regularisation on loss values is appreciable and is therefore documented here. On a fixed random subset of 512 test states, comparing the loss with a 10^{-4} perturbation scale (five independent draws) against the unperturbed metric for the principal reconstructor’s predictions gives a mean absolute

per-state shift of 4.0×10^{-3} at 10^3 uniform shots per setting, 5.7×10^{-3} at 10^5 uniform shots per setting and 4.1×10^{-3} at twelve observed measurements with 10^4 shots, with per-state maxima near 3×10^{-2} ; on exactly degenerate spectra (identical pure or maximally mixed pairs) the perturbed loss reads up to 1.1×10^{-2} where the true distance is zero. The measured difference confirms that this is a genuinely stochastic training objective rather than an algebraically identical implementation of the reporting metric. The script `quantify_bures_jitter.py` in the experimental repository reproduces this table, and `verify_physics.py` validates the unperturbed implementation against an analytic formula and against Qiskit [70].

A.1.6 Fixed-budget likelihood-optimisation configuration

The reported 250-step Cholesky likelihood optimisation minimises the negative binomial log-likelihood of the observed counts (the parameter-independent binomial coefficients are dropped), with settings whose shot count is zero excluded from the likelihood product. The optimisation runs in double precision over a Cholesky factor of ρ (rendering the parameterised problem non-convex, Section 2.1.9), with Born probabilities clamped to $[10^{-8}, 1 - 10^{-8}]$. The optimiser is Adam [35] with learning rate 0.08 for a fixed budget of 250 steps per state; the final iterate is reported. The factor is warm-started from $\mathcal{P}_{\text{clip}}(\hat{\rho}_{\text{LI}})$ (Algorithm 2.1), with a 10^{-5} diagonal shift before the Cholesky factorisation; a random $\mathcal{N}(0, 0.1)$ initialisation is the fallback if the factorisation fails, which is essential for reaching the reported high-shot values within the step budget. Because the likelihood fit is a per-state optimisation, the evaluation is substantially slower than the direct models. The fixed step count, final-iterate rule and absence of a convergence tolerance are part of the estimator’s procedural definition; its values are therefore a fixed-compute likelihood reference. The reported sweeps use all 45 999 test states, with the same states, observed subsets, shot counts and binomial draws as the other methods.

Eigenvalue-simplex projection of the linear estimate. The linear/pseudoinverse baseline of the comparison restores physicality with $\mathcal{P}_{\text{clip}}$ (Algorithm 2.1). A second variant projects the unconstrained estimate’s eigenvalues exactly onto the probability simplex and therefore returns the Frobenius-nearest valid state to that estimate. The reported procedure is thus a state-space projection of an unconstrained linear estimate; a trace-constrained measurement-space fit is a separate estimator. Evaluated with paired noise draws on the full test split, the exact projection improves the linear-inversion values to 0.136, 0.057 and 0.024 at 10^3 , 10^4 and 10^5 uniform shots and to 0.365 at twelve observed measurements at 10^4 shots, an approximately 5–23% improvement over the $\mathcal{P}_{\text{clip}}$ variant that preserves the ranking in Section 4.1. Both variants are retained in Table 4.1, with $\mathcal{P}_{\text{clip}}$ included for continuity with earlier comparisons.

A.1.7 Code availability

Experiment repository location:

<https://gitlab.mff.cuni.cz/cechovim/qubits>

The repository contains the implementation, tracked evaluation outputs and development history used for this thesis; the state data are available through the Zenodo record cited in Section 3.1.1.

A.1.8 Per-pipeline configurations

Table A.2 gives the stored parameter count, architecture and objective of the five selected learned implementations and two naive baselines in Table 4.1; the RhoTransformer active-path count is reported separately. The reconstruction-only cGAN comparison shares the cGAN generator and $\mathbf{z} = 0$ reporting rule, but sets $\lambda_{\text{adv}} = 0$ and has the separate optimisation history described in Section 4.1.1. All use the same base state partitions and paired test observations; their training samplers are reported above. The selected models share seventeen-component inputs, confidence gating and a Cholesky head, while Equation (2.42) records the harmonic and dyadic frequency grids used by the different families. Figures A.1–A.3 show the selected feed-forward, adversarial and flow-matching data paths in the same convention as Figure 3.4.

Conditional GAN. The generator and discriminator are each updated once per training mini-batch with Adam at learning rate 10^{-4} in the plateau continuation and $(\beta_1, \beta_2) = (0.5, 0.999)$. The discriminator has two hidden layers of width 512, leaky-ReLU activations and dropout 0.1, and receives the sixteen projector probabilities implied by a real or generated state together with the noisy input and confidence. During training the generator receives $\mathbf{z} \sim \mathcal{N}(0, I_{16})$. During every reported evaluation it runs in evaluation mode and sets $\mathbf{z} = 0$, making the reported point estimate deterministic. The non-saturating adversarial term has weight 0.05.

Development-only GAIN precursor. The initial GAIN measurement-imputation run standardises the sixteen exact measurement values and hides four randomly chosen entries per example. Its generator and discriminator are MLPs with two hidden layers of width 64, ReLU activations and dropout. The generator is trained with Equation (2.73) using $\alpha = 20$; the discriminator target is smoothed to 0.9 for observed and 0.1 for imputed entries. Training uses mini-batches of 128, seed 42, one discriminator update per three generator updates, and Adam with learning rates 2×10^{-4} for the generator and 5×10^{-5} for the discriminator. Both gradient norms are clipped to 1.0, and validation missing-entry RMSE selects the checkpoint within a 250-epoch cap.

The later feature-matching branch min-max scales the measurement values, replaces these small MLPs by residual generator and discriminator networks, and adds an MSE between hidden activations of a fixed entanglement classifier on the true and generator-output vectors. The historical classifier takes all sixteen probabilities, has hidden widths 1 024, 1 024, 512 and 128 with ReLU and dropout 0.2, and was trained on binary entanglement tags with binary cross-entropy, AdamW at learning rate 10^{-3} and weight decay 0.01, mini-batches of 256 and validation early stopping. This sixteen-input feature extractor belongs to the GAIN development branch; Attachment A.1.9 documents the later twelve-input classifier used for the reported reconstruction diagnostic.

Flow-matching models. The final flow-matching models use rectified flow in the unstandardised sixteen-dimensional Cholesky-coordinate space (2.29). For a clean target x and $\varepsilon \sim \mathcal{N}(0, I)$, training draws $t \sim \mathcal{U}[0, 1]$ and forms $z_t = (1-t)x + t\varepsilon$ with velocity target $\varepsilon - x$, as in the rectified path (2.76). The objective follows (2.77), with velocity MSE plus 0.25 times the regularised Bures term evaluated on the implied clean estimate $z_t - tv_\theta(z_t, t, c)$. Adam uses weight decay zero; the plateau continuations use learning rate 1.5×10^{-4} . Inference starts from noise, $z_{t=1} \sim \mathcal{N}(0, I)$, and integrates backward in t to the reconstructed Cholesky vector at $t = 0$ with eight Euler steps, then averages ten independent density matrices; the arithmetic average of density matrices is again a valid state and serves as the reported point estimate. Before that average, each sampled endpoint passes through the normalised Cholesky map and the flow sampler’s endpoint correction: an independent real symmetric Gaussian perturbation of scale 10^{-6} is added, after which negative eigenvalues are clipped and the trace is renormalised. This correction is part of flow inference and is separate from the metric used to score the final estimate. Clean Cholesky targets are extracted from ρ in complex double precision with a 10^{-6} diagonal jitter and an eigendecomposition-based fallback for the rank-deficient states on which a strict factorisation fails. This flow-time convention is separate from the DDPM notation of Section 2.3.9, where x_0 denotes clean data.

Table A.2: Stored parameters, architecture and objective for the learned models in the final comparison. Exact training and inference settings are stated immediately above and in the cited sections. RhoTransformer has about 3.69 million active-path parameters.

Model	Stored params	Architecture	Objective
RhoTransformer	3 980 545	Section A.1.1; harmonic Fourier(8), Cholesky head	Bures + $w=50$ consistency (A.1)
DeepSets MLP	4 754 704	per-slot $3 \times 1\,024$, masked mean-pool, output $3 \times 2\,048$; harmonic Fourier(8), Cholesky head	Bures + $w=50$ consistency
cGAN (adversarial)	3 461 136	generator $4 \times 1\,024$, latent $\mathbf{z} \in \mathbb{R}^{16}$; dyadic Fourier(8), Cholesky head, conditional discriminator	Bures + $\lambda_{\text{adv}}=0.05$ non-saturating adversarial
Flow matching (transformer)	3 835 137	transformer velocity network: 4 layers, $d_{\text{model}} = 256$, 8 heads; dyadic Fourier(8), Cholesky head	flow matching + 0.25 auxiliary Bures
Flow matching (MLP)	1 860 368	velocity MLP 4×256 ; dyadic Fourier(8), Cholesky head	flow matching + auxiliary Bures (weight 0.25)
Naive transformer	2 117 648	4-layer encoder, $d_{\text{model}} = 256$, 8 heads, feed-forward 512; raw $[\mathbf{m}; \mathbf{c}]$ input, flat ρ output + $\mathcal{P}_{\text{clip}}$ core	MSE
Vanilla MLP	42 156 304	5 hidden layers $\times 2\,048$; raw $[\mathbf{m}; \mathbf{c}]$ input, flat ρ output + $\mathcal{P}_{\text{clip}}$ core	MSE (weight decay 0; see Section 3.6.2)

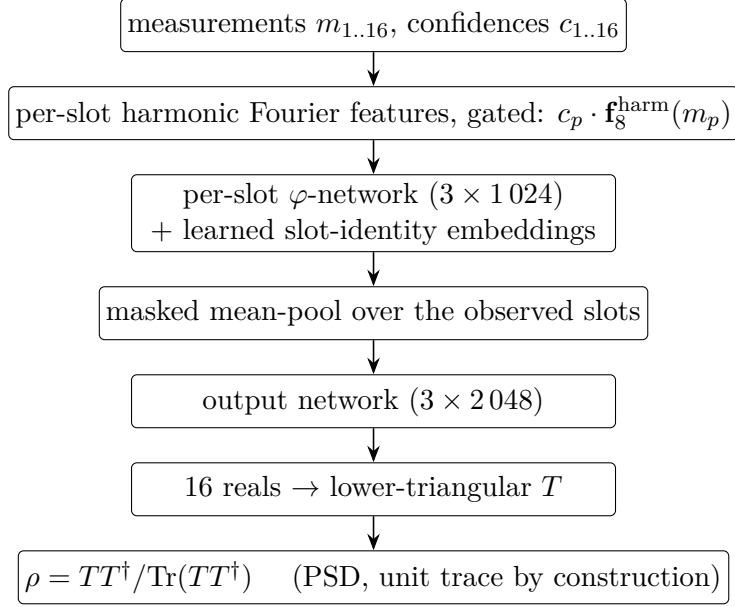


Figure A.1: The selected 4.8-million-parameter feed-forward model applies one shared φ -network to embedded measurement slots, masked-mean-pools the observed slots and predicts Cholesky coordinates. Its input and output conventions match RhoTransformer.

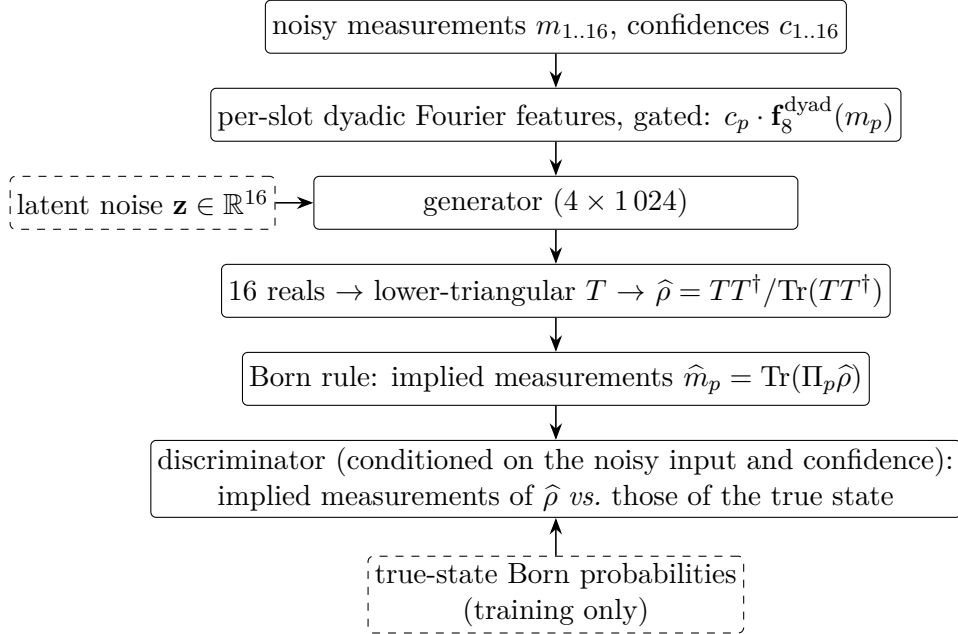


Figure A.2: The 3.5-million-parameter generator uses Bures reconstruction plus a non-saturating adversarial term ($\lambda_{\text{adv}} = 0.05$). The discriminator is training-only; reported inference fixes $\mathbf{z} = 0$ and uses one generator pass.

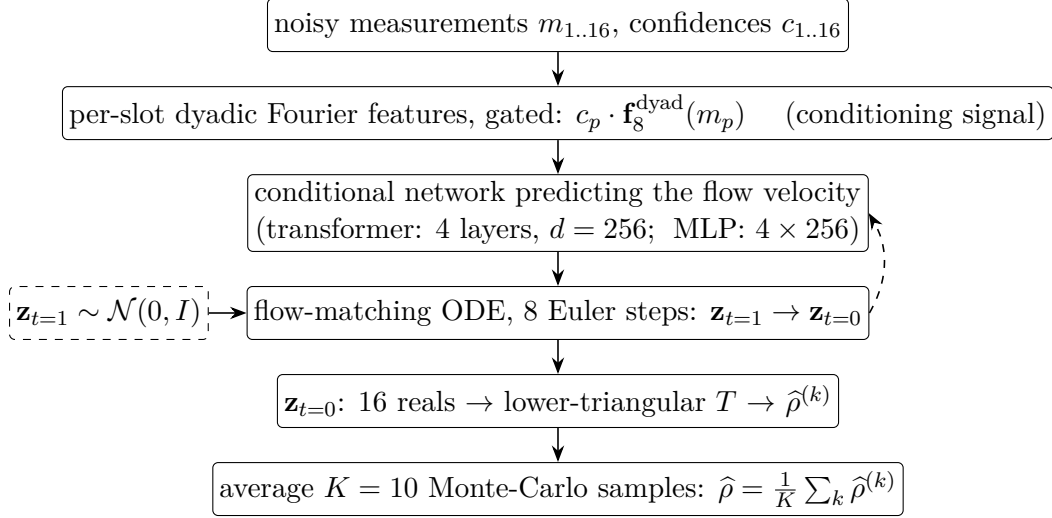


Figure A.3: A conditional transformer (3.8 million parameters) or MLP (1.9 million) velocity network transports Gaussian noise to Cholesky coordinates through eight Euler steps. Averaging ten trajectories requires about eighty velocity-network evaluations.

A.1.9 Frozen entanglement classifier

The classifier in Table 4.7 is a four-hidden-layer MLP with widths 1 024, 1 024, 512 and 128, ReLU activations and dropout 0.2, followed by one binary logit. Its input consists of twelve James–Kwiat projector probabilities: all sixteen probabilities are min–max scaled using statistics fitted only on `data_train.csv`, after which columns 0, 1, 4, 5 are omitted to match the stored classifier. The 459 999 rows in that file are split 80:20 for fitting and internal validation with seed 42; the separate 45 999-row `data_valid.csv` supplies the original fixed classifier evaluation. The filename `data_valid.csv` is historical: in the tomography experiments this file serves as the final fixed test split, while checkpoint selection uses the distinct tomography validation split. Training uses binary cross-entropy, AdamW with learning rate 10^{-3} and weight decay 0.01, mini-batch size 256, a reduce-on-plateau factor of 0.1 after five stagnant epochs, and early stopping after ten stagnant epochs within a 2 000-epoch cap. A sigmoid threshold of 0.5 defines its binary output. For the tomography probe, the stored weights and preprocessing are frozen and applied to projector probabilities implied by each reconstruction.

A.1.10 Adaptive selector

The selector reported in Section 4.6 has four hidden layers of width 384, GELU activations and layer normalisation, for 477 712 stored parameters. It forms 600 000 training examples by drawing states with replacement from the enlarged secondary-study training pool; partial subsets and shot noise are resampled on each of 25 epochs. A fixed validation subset contains 4 096 states. Training uses mini-batch size 512, AdamW at learning rate 3×10^{-4} and weight decay 10^{-4} , cosine annealing, and seed 42. After backpropagation on each mini-batch, the global ℓ_2 norm of the selector gradients is clipped to 1.0 before the AdamW step, following Equation (2.48). Partial subsets range from zero to fifteen observed

settings, each observed setting uses 10^4 shots, and the soft gain target of (3.4) has temperature 0.05. Each candidate target is estimated from one finite-shot outcome, as noted in Section 3.7.1. The two fixed orderings are derived on 2 000 validation states with seed 42: π^* uses modal selector choices, while π_{MG} uses eight random marginal-gain probes per state. The orderings are then frozen before test evaluation.

The conceptual acquisition loop is shown in Figure 2.3; its complete experimental specification is in Section 3.7.

A.1.11 Binary concurrence-decision diagnostics

Table A.3: Binary decision diagnostics corresponding to Table 4.6. The same threshold τ is applied to true and reconstructed concurrence. Prevalence is the fraction labelled entangled at that threshold; precision and recall treat entanglement as the positive class. Counts are over all 45 999 test states.

Operating point	τ	Prevalence	Precision	Recall	TP	FP	TN	FN
uniform 10^3	10^{-3}	0.532	0.781	0.942	23076	6465	15043	1415
uniform 10^3	10^{-2}	0.516	0.868	0.936	22235	3382	18861	1521
uniform 10^3	0.05	0.453	0.912	0.932	19397	1883	23299	1420
uniform 10^5	10^{-3}	0.532	0.812	0.988	24193	5611	15897	298
uniform 10^5	10^{-2}	0.516	0.957	0.985	23407	1039	21204	349
uniform 10^5	0.05	0.453	0.989	0.987	20556	233	24949	261
$n_{\text{high}}=12$ at 10^4	10^{-3}	0.532	0.771	0.881	21586	6407	15101	2905
$n_{\text{high}}=12$ at 10^4	10^{-2}	0.516	0.874	0.877	20823	3012	19231	2933
$n_{\text{high}}=12$ at 10^4	0.05	0.453	0.902	0.869	18089	1959	23223	2728

A.2 Development history of the reconstruction models

The controlled comparison of Chapter 4 (Table 4.1) reports the final common-protocol selected-pipeline result. For completeness, Table A.6 records the longer development trajectory that preceded it: the main model variants explored on the way to the principal reconstructor, including intermediate development variants. Each row retains the metric and protocol of its original experiment. The experiments differ in noise model, in evaluation metric (some rows give a validation Bures distance, others the complete noisy Bures at 10^5 shots or a measurement-imputation MSE), and in training budget. The table therefore documents the search trajectory: adjacent rows may differ in several factors, so numerical comparisons belong within their original protocols. Table 4.1 provides the common-protocol ranking.

The same checkpoint reached mean Bures distance 0.157 with twelve observed settings at 10^4 shots per observed setting. The complete-data curve shows diminishing benefit above 10^4 shots and documents why the selected iterative branch moved to flow matching.

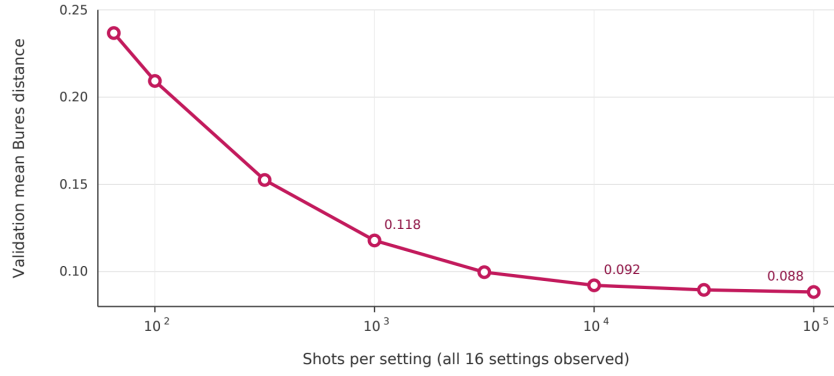


Figure A.4: Complete-setting validation sweep for a preliminary DDPM with a transformer denoiser. Each estimate averages ten independently sampled end-points from 20-step deterministic diffusion trajectories in flat coordinates before physicsity correction. This 4 096-state development subset is separate from the central test ranking.

Table A.4: Conditional-GAN development at 10^5 shots per setting. Short runs use validation data; the plateau row uses the test split. This within-series comparison supports the Fourier input; Table 4.1 gives the final common-protocol result.

cGAN variant	Partition	Mean Bures
raw input, adversarial (short)	validation	0.083
Fourier input, adversarial (short)	validation	0.072
Fourier input, zero adversarial weight (short)	validation	0.069
Fourier input, adversarial (plateau)	test	0.067

Table A.5: Development checks of complete-anchor training, Pauli output, likelihood polishing, a mixture-of-experts (MoE) head, Monte Carlo dropout and equal-weight state averaging. Entries are validation-set mean Bures distances; lower is better. Each study block uses matched evaluation records under its own development protocol; Table 4.1 remains the headline test ranking.

Study	Evaluation rule	CN at 10^5 shots per setting	12 observed at 10^4 shots per observed setting
Uniform anchors	ordinary online sampler	0.04878	–
	25% complete uniform-shot anchors	0.02665	–
Pauli head	Cholesky-coordinate reference	0.01503	0.10816
	Pauli-coefficient output	0.04837	0.11807
Likelihood polish	direct checkpoint output	0.01571	0.10802
	five Adam likelihood steps	0.01517	0.10717
MoE head	single flat-output head	0.01985	0.11497
	four experts, top-2 routing	0.01194	0.11441
MC dropout	deterministic inference	0.01852	0.11431
	50 stochastic passes, averaged	0.02202	0.11663
Ensembling	MoE flat-output checkpoint	0.01058	0.10280
	direct-Cholesky checkpoint	0.01531	0.10023
	equal-weight state average	0.01232	0.10021

The anchor rows form a matched early-development comparison; the complete high-shot reduction is 45.4%, while the mixed validation mean changes from 0.19679 to 0.19888. The Pauli-head reference and variant use the same development evaluation records; their mixed validation values are 0.17989 and 0.18773. Likelihood polishing is a post-processing comparison on one checkpoint. Its five stable steps help the high-information points modestly but add per-state optimisation and worsen lower-shot points by up to 1.5%.

The MoE comparison changes only the head of the same 250-epoch flat-output corrector. It is therefore an intermediate success: routing improved the complete high-shot point substantially and produced a smaller improvement at twelve observed settings. This result belongs to the preceding flat-output lineage; the selected direct-Cholesky design uses a single output head. The Monte Carlo study evaluates the same checkpoint once deterministically and with 50 explicit-dropout passes. The ensemble study uses the same 8 000-state validation subset for both constituent checkpoints and their density-matrix average. It gains only 0.00002 over the better constituent at twelve settings while increasing the complete high-shot error and requiring two neural forward passes.

Table A.6: Chronological development record under each row’s original protocol. “val” is validation Bures, “CN@10⁵” is complete-noisy Bures at 10⁵ shots per setting, “early/main” identifies the development stage, and “meas./cons.” abbreviate the measurement-consistency terms. Cross-row values are not a controlled ranking.

Approach	Stage	Loss	Result	Outcome
GAIN + entanglement-feature matching	early	adversarial + MSE + feature MSE	test MSE 0.0117	four missing min–max-scaled measurement values; imputation precursor
Cholesky feed-forward (deterministic)	early	HS + Bures	val 0.106	first deterministic baseline
Matrix-space diffusion, missing-slot	early	noise MSE	val 0.125	diffusion proof of concept
Matrix-space diffusion (16-D)	early	noise MSE	val 0.098	substantial improvement over the preceding diffusion model
+ meas.-consistency loss	early	noise MSE + cons.	val 0.106	selected early diffusion baseline
+ continuous confidence channel	early	noise MSE + cons.	val 0.117	flexible masking; small cost
DDPM with transformer denoiser	main	noise MSE	Fig. A.4	complete-setting development sweep
Feed-forward baseline	main	MSE	val 0.188	development reference
Transformer + confidence + $\mathcal{P}_{\text{clip}}$ layer	main	MSE	val 0.127	$\mathcal{P}_{\text{clip}}$ map: +11%
Transformer, Bures loss	main	MSE + Bures	val 0.109	reference transformer
Diffusion (ϵ -pred.) + MC-dropout	main	MSE + Bures	complete at 10 ⁴ : 0.029	later development baseline under a separate protocol
Corrector transformer (linear-inversion warm-start)	main	Bures + meas. (clean)	CN@10 ⁵ 0.047	linear-inversion corrector baseline
+ Fourier features, noisy-target loss	main	Bures + meas. (noisy)	CN@10 ³ 0.032	Fourier input features
+ Cholesky output	main	Bures + meas.	CN@10 ⁵ 0.032	PSD by construction
Flat-output corrector + mixture-of-experts head	main	Bures + meas.	CN@10 ⁵ 0.012	high-shot improvement in the preceding flat-output lineage; focused comparison in Table A.5
Direct Cholesky transformer	main	Bures + meas.	CN@10 ⁵ 0.016	selected direct reconstructor
minus axial-FFN mixer	main	Bures + meas.	val 0.177	near-identical ablation
RhoTransformer (single-token)	main	reg. Bures + meas.	val 0.183	principal model (3.98M stored; 3.69M active)
SwiGLU-only encoder	main	Bures + meas.	val 0.182	smallest variant (3.4M)
Data-scaled retrain (4× states)	main	Bures + meas.	CN@10 ⁴ 0.029	side experiment under its original development protocol; the Chapter 4 checkpoint remains trained on the base pool