BuresTomFlow: Bures-Geometric Flow Matching for Posterior Quantum Tomography

Lu Wei
Data Science Department
Stony Brook University
luw744895@gmail.com

Chenfeng Cao
Dahlem Center for Complex Quantum Systems
Freie Universität Berlin
chenfeng.cao@fu-berlin.de

Yufeng Wang
Department of Computer Science
Stony Brook University
yufeng.wang.2@stonybrook.edu

Haibin Ling
Department of Artificial Intelligence
Westlake University
linghaibin@westlake.edu.cn

Abstract

Finite-shot quantum tomography often leaves a posterior over many density matrices compatible with the same measurement record, but fast reconstruction methods collapse this uncertainty to a single state. We frame this setting as amortized posterior sampling and introduce BuresTomFlow, a measurement-conditioned Flow Matching sampler for full-rank density matrices. The method transports base density matrices along Bures-Wasserstein paths on the positive matrix cone, projects the path back to the trace-one state space, and trains the velocity field with a Bures-aligned tangent loss. Thus the sampler is trained in the same fidelity-based geometry used to judge posterior quality. We evaluate BuresTomFlow in dense four-qubit simulations with count, shadow, and mixed measurement records. Against Cholesky, log-density, and generic Riemannian Flow Matching controls at comparable total runtime for training and posterior sampling, BuresTomFlow reduces Bures distance and credible-ball coverage error, and improves held-out observable prediction. In the main comparison across multiple seeds, mean Bures distance decreases from 0.618 for the Cholesky baseline to 0.367 for BuresTomFlow, and mean coverage error decreases from 0.203 to 0.100. These results show that the value of Bures geometry is not merely enforcing physical states, but improving posterior estimates and reducing credible-ball coverage error when finite-record tomography cannot be reduced to one reconstruction.

1 Introduction

Quantum state tomography infers an unknown density matrix from measurement data. In many finite-record regimes, however, the data do not determine a single plausible state. Low-shot Pauli counts, classical-shadow records, and mixed measurements can leave a posterior family of density matrices that agree with the observations while disagreeing on fidelity, held-out observables, or repeated-run calibration. Collapsing this family to one reconstruction removes the uncertainty that the experiment actually leaves.

This posterior objective is inherently geometric. The unknown state is constrained to the positive semidefinite, trace-one state space, and quantum states are naturally compared by fidelity and Bures-type distances rather than entrywise Euclidean error. A tomography posterior sampler must therefore satisfy three requirements at once: produce physical density matrices, represent multiple

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

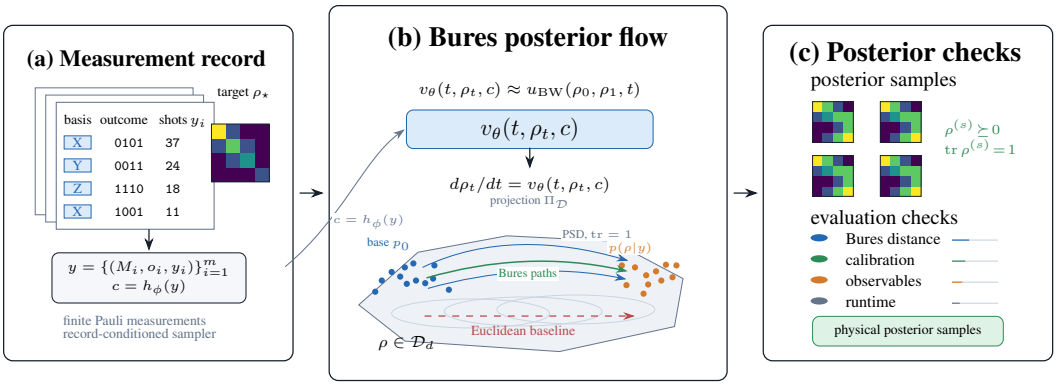

Figure 1: Overview of BuresTomFlow. A finite tomography record conditions a neural velocity field. Base density matrices are transported through Bures-Wasserstein paths, and the trained ODE sampler then returns physical samples from the learned posterior.

plausible states conditioned on the same record, and calibrate uncertainty in a geometry matched to quantum-state distinguishability.

Existing tomography pipelines meet these requirements only partially. Linear inversion and maximum-likelihood estimators provide efficient point estimates, but they require projection or regularization under sparse records and do not represent posterior uncertainty. Bayesian and Markov-chain methods expose uncertainty, but they can be expensive for each new measurement record. Neural tomography amortizes reconstruction, yet most learned approaches optimize reconstruction or observable prediction rather than a density-matrix posterior whose paths, losses, and calibration checks are geometry aligned.

We introduce BuresTomFlow, a measurement-conditioned Flow Matching sampler for full-rank density matrices. Starting from base density matrices, the model learns velocity fields along Bures-Wasserstein positive-matrix paths, projects intermediate states back to the trace-one state space, and trains with Bures-aligned tangent supervision. This construction makes the metric a modeling ingredient rather than a post-processing score: the same fidelity-based geometry shapes the training paths and evaluates posterior samples.

Empirically, this geometry is not interchangeable with generic physical parameterizations. Across multi-seed dense four-qubit experiments, BuresTomFlow reduces Bures distance and credible-ball coverage error, and improves held-out observable prediction over Cholesky, log-density, and vanilla Riemannian Flow Matching controls at comparable total runtime for training and posterior sampling. The gains persist across the tested shot budgets in shadow, count, and mixed measurement regimes, where the data leave meaningful posterior ambiguity.

Figure 1 summarizes the pipeline: finite measurement records condition the velocity field, Bures-Wasserstein paths define the training target, and posterior samples are evaluated by state-space error, calibration, observable prediction, and runtime.

Contributions. The main contributions are:

1. We formulate finite-record quantum tomography as amortized posterior sampling over density matrices, evaluated by Bures distance, credible-region coverage, held-out observable error, and total runtime for training and posterior sampling.

2. We develop a Bures-geometric conditional Flow Matching construction for density matrices, including Bures-Wasserstein paths, tangent supervision, and trace-one projection.

3. We isolate the role of geometry through Cholesky, log-density, generic Riemannian, and point-estimation controls, and organize the results by measurement regime.

2 Related Work and Background

2.1 Finite-Record Quantum Tomography

Tomography under finite records. Quantum state tomography estimates a density matrix from measurement data, with classical estimators, Bayesian methods, compressed-sensing variants, and randomized-measurement protocols covering different regimes [1, 5, 13, 15–17, 25]. A point estimate can be sufficient when the record is highly informative. In the finite Pauli and shadow settings studied here, however, the record leaves posterior ambiguity over compatible physical states. We therefore use the record as a conditioning variable for posterior sampling rather than as input to one estimator. Appendix A gives the fuller tomography background.

Neural tomography. Neural tomography and conditional generative tomography learn fast reconstruction or generation maps from measurement statistics [2, 8, 9, 27, 30, 32, 36]. The distinction here is the target: posterior samples over density matrices, evaluated by Bures distance, held-out observables, and credible-region coverage.

2.2 Flow Matching and Posterior Sampling

Flow matching and posterior inference. Amortized neural posterior estimation learns conditional distributions from simulated parameter–observation pairs rather than running separate inference for each observation [12]. Flow Matching and closely related rectified-flow and stochastic-interpolant formulations train continuous-time generative models by regressing velocity fields along prescribed probability paths [3, 21, 23]. In standard notation, a base sample $x_0 \sim p_0$ is coupled with a target sample $x_1 \sim q$, and the path and regression loss are

$$x_t = \psi_t(x_0, x_1), \qquad \mathcal{L}_{\mathrm{FM}}^{\mathrm{std}}(\theta) \triangleq \mathbb{E}_{t,x_0,x_1} \left\| v_\theta(t, x_t) - u_t(x_t \mid x_0, x_1) \right\|^2. \tag{1}$$

Conditional Flow Matching (CFM) adds a condition c to the same regression, replacing $v_\theta(t, x_t)$ by $v_\theta(t, x_t, c)$ and using endpoint pairs drawn under the condition. Minibatch optimal transport variants make the choice of coupling part of the model [31]. Riemannian Flow Matching extends the same regression principle to non-Euclidean domains [10], and Flow Matching posterior estimation applies it to simulation-based inference [35]. These tools specify how to train conditional flows. The modeling question here is which geometry should define paths, velocities, and losses for posterior density-matrix tomography. We use credible-region coverage as a posterior-calibration metric, related to but not replacing full simulation-based calibration with rank statistics [29].

For the posterior-tomography instance, the condition is the measurement record y. A base state $\rho_0 \sim p_0$ is coupled with a posterior target $\rho_1 \sim p(\rho \mid y)$, a path $\rho_t = \psi_t(\rho_0, \rho_1)$ is chosen, and the network is trained to match the corresponding velocity:

$$\rho_t = \psi_t(\rho_0, \rho_1), \qquad \mathcal{L}_{\mathrm{PFM}}(\theta) \triangleq \mathbb{E}_{t,\rho_0,\rho_1,y} \left\| v_\theta(t, \rho_t, h_\phi(y)) - u_t(\rho_t \mid \rho_0, \rho_1) \right\|^2. \tag{2}$$

The path ψ_t, velocity u_t, and norm in Eq. (2) are where geometric assumptions enter the model.

2.3 Geometric Flow Matching

Riemannian Flow Matching. Riemannian Flow Matching replaces Euclidean straight-line paths with geometry-aware paths on a manifold [10]. For points x_0, x_1 on a Riemannian manifold $\mathcal{M}$, the geodesic path and its velocity are

$$x_t = \exp_{x_0}\left(t \, \log_{x_0}(x_1)\right), \qquad u_t(x_t \mid x_0, x_1) = \frac{dx_t}{dt}. \tag{3}$$

The corresponding velocity-regression loss is

$$\mathcal{L}_{\mathrm{RFM}}(\theta) \triangleq \mathbb{E}_{t,x_0,x_1,c} \left\| v_\theta(t, x_t, c) - u_t(x_t \mid x_0, x_1) \right\|_{g_{x_t}}^2. \tag{4}$$

Thus the geometry enters through the path, tangent velocity, and norm. BuresTomFlow follows this principle, but the geometry is chosen from the posterior-tomography target itself: Bures-Wasserstein positive-matrix transport rather than a generic chart.

Geometry-aware flow models. Recent geometric flow models follow this problem-driven pattern by adapting flow-based generation to structured output spaces such as SPD matrices and SE(3) poses [11, 24]. In posterior tomography, the structured object is a density matrix, the condition is a finite quantum measurement record, and the central evaluation target is calibrated uncertainty under quantum-state metrics. The contribution is therefore not another generic manifold flow, but a posterior-tomography construction whose value is judged by posterior quality.

2.4 Bures Geometry for Density Matrices

Bures geometry and quantum optimal transport. Fidelity and Bures distance compare quantum states through distinguishability rather than entrywise matrix error [7, 18, 33]. Bures-Wasserstein geometry gives closed-form positive-matrix transport operations [4]. Because trace-one density matrices are not a geodesic submanifold of the positive cone [34], BuresTomFlow uses positive-cone interpolation followed by projection back to the physical state space. A generic projection need not preserve geodesics, but on the evaluated paths this projection reduces to trace normalization of optimally aligned factors, which traces the induced-metric geodesic of the trace-one set with a non-constant-speed parameterization [34, Theorem 7]; Appendix B gives the details. This choice is not only a positivity constraint. Cholesky, log-density, and generic Riemannian parameterizations can preserve physical-looking matrices, while the tested claim is that posterior sampling improves when the training geometry is aligned with the fidelity-based geometry used for evaluation.

Positioning. The gap is therefore a posterior-sampling problem, not merely a reconstruction problem. Tomography, amortized posterior inference, Flow Matching, and Bures geometry exist separately, but they have not been combined into a density-matrix posterior sampler whose paths, training loss, and uncertainty checks are aligned with the metric used to judge quantum states. The main experiments isolate this adaptation by comparing BuresTomFlow against Cholesky, log-density, vanilla Riemannian Flow Matching, and point-estimation controls.

3 BuresTomFlow

3.1 Posterior Tomography Target

Finite-record tomography is a conditional generation problem on the density-matrix state space, rather than a deterministic regression problem. For an n-qubit system with $d = 2^n$, the physical domain is

$$\mathcal{D}_d = \left\{ \rho \in \mathbb{C}^{d\times d} \;\middle|\; \rho = \rho^\dagger,\ \rho \succeq 0,\ \mathrm{Tr}(\rho) = 1 \right\}. \tag{5}$$

A measurement record $y = \{(a_i, M_i)\}_{i=1}^m$ is generated from Born probabilities, and the corresponding Bayesian tomography posterior is [25]

$$p(\rho \mid y) \propto p_{\mathrm{prior}}(\rho) \prod_{i=1}^{m} \mathrm{Tr}(M_{i,a_i}\rho), \qquad \rho \in \mathcal{D}_d. \tag{6}$$

The role of BuresTomFlow is to amortize this conditional distribution: for a new finite record y, the sampler returns

$$\rho^{(s)} \sim q_\theta(\rho \mid y), \qquad s = 1, \ldots, S, \tag{7}$$

where q_θ is trained to approximate the posterior family induced by the simulator and measurement model.

The training pairs come from the joint law rather than from hand-labeled posterior clouds. We draw a latent state $\rho_\star \sim p_{\mathrm{prior}}$, simulate a finite record $y \sim p(y \mid \rho_\star)$, and use the pair $(y, \rho_\star)$ for conditional Flow Matching. Under this prior-predictive sampling scheme, the conditional law of $\rho_\star$ given y is exactly the posterior in Eq. (6). The neural sampler therefore learns posterior samples by seeing many simulated inverse problems. This is the same reason the method targets a sample cloud rather than a single reconstruction: when y is sparse, several states remain plausible under the same record.

3.2 Measurement-Conditioned Velocity Field

The measurement record is treated as an unordered set of tokens. Each token encodes the product-Pauli setting, the measured bit pattern or aggregate count summary, the shot budget, and the token

type:

$$
\begin{aligned}
r_j &= \big[\mathrm{onehot}(B_j),\, b_j,\, c_j,\, n_j/10^4,\, \mathrm{onehot}(\tau_j)\big] \in \mathbb{R}^{24}, \\
c_y &= h_\phi(\{r_j\}_{j=1}^m).
\end{aligned}
\tag{8}
$$

Here $B_j \in \{X, Y, Z\}^{\otimes n}$ denotes the local measurement basis, b_j stores bit-level outcomes for shadow snapshots or averaged bits for count records, c_j stores an all-plus/count statistic, n_j is the shot count, and τ_j distinguishes snapshot and count tokens. Shadow records use randomized Pauli settings. Count records aggregate repeated shots for fixed settings. Mixed records concatenate the two. The map h_ϕ is implemented as either a Deep Sets encoder or a small masked self-attention encoder.

The velocity network receives the current matrix, time, and record context. In the dense four-qubit model, the input feature vector is the real-imaginary matrix representation augmented by time and the record embedding:

$$
\chi(t, \rho_t, y) = \big[\mathrm{Re}\,\mathrm{vec}(\rho_t),\, \mathrm{Im}\,\mathrm{vec}(\rho_t),\, t,\, c_y\big]. \tag{9}
$$

Before the velocity is used in either a Bures or Frobenius loss, a matrix proposal is projected onto the Hermitian trace-free tangent space:

$$
\begin{aligned}
\mathcal{P}_T(A) &= \mathrm{Herm}(A) - \frac{\mathrm{Tr}(\mathrm{Herm}(A))}{d} I, \\
\mathrm{Herm}(A) &= \frac{A + A^\dagger}{2}.
\end{aligned}
\tag{10}
$$

Thus the network learns trace-preserving density-matrix directions rather than arbitrary complex matrices.

3.3 Bures-Wasserstein Probability Paths

The geometric choice enters before any evaluation metric is computed. BuresTomFlow builds probability paths using Bures-Wasserstein transport on the positive matrix cone, then returns each intermediate point to the trace-one state space. Full-rank states are used in the main dense experiments because the matrix square roots and inverse square roots in the Bures map become singular at the boundary.

When a density-matrix target may be close to the boundary, it is first moved into the interior by the depolarizing mixture $\rho^\eta = (1-\eta)\rho + \eta I/d$. This smoothing is separate from the projection used for a general matrix proposal. If $\mathrm{Herm}(A) = U\,\mathrm{diag}(\lambda_i)\,U^\dagger$, define $\lambda_i^+ = \max\{\lambda_i, \delta\}$ for a small floor $\delta > 0$. The density-matrix projection is

$$
\Pi_{\mathcal{D}_d}(A) = \frac{U\,\mathrm{diag}(\lambda_i^+)\,U^\dagger}{\sum_i \lambda_i^+}. \tag{11}
$$

Given a base state ρ_0 and a smoothed target ρ_1^η, the positive-cone Bures-Wasserstein transport map defines the positive factor $T_{0\to1}$ [4]:

$$
T_{0\to1} = \rho_0^{-1/2}\big(\rho_0^{1/2}\rho_1^\eta\rho_0^{1/2}\big)^{1/2}\rho_0^{-1/2}. \tag{12}
$$

The interpolation uses the affine factor $H_t = I + t(T_{0\to1} - I)$, transports the base state in the positive cone as $\tilde{\rho}_t = H_t^\dagger \rho_0 H_t$, and then returns the path to the trace-one domain:

$$
\rho_t = \Pi_{\mathcal{D}_d}(\tilde{\rho}_t), \qquad u_t = \dot{\rho}_t = \frac{d}{dt}\Pi_{\mathcal{D}_d}(\tilde{\rho}_t). \tag{13}
$$

All derivatives are computed through the differentiable matrix operations defining Eqs. (12) and (13). The important modeling point is that ρ_t is not an ambient straight line between two matrices. The path first follows the positive-matrix geometry associated with fidelity and only then applies the trace-one constraint required by tomography. At every training path state we evaluated, the eigenvalue floor is inactive, so $\Pi_{\mathcal{D}_d}$ reduces to the trace normalization $\rho_t = \tilde{\rho}_t/\,\mathrm{Tr}(\tilde{\rho}_t)$.

This projection is part of the learned probability path rather than a post-hoc validity repair. The sampler is trained on the differentiated projected path, the velocity loss is evaluated after tangent projection, and posterior quality is measured with the same Bures distance. Thus the method couples three objects that are often separated in reconstruction pipelines: the physical density-matrix domain, the velocity geometry, and the uncertainty metric. Appendix B gives the positive-cone interpolation, projection, and tangent-metric details.

3.4 Bures-Aligned Training Objective

The conditional velocity field is trained by matching the Bures path velocity in the local Bures tangent metric. For a Hermitian trace-free tangent direction ξ, the metric is defined through the Lyapunov operator $L_\rho(\xi)$:

$$L_\rho(\xi)\rho + \rho L_\rho(\xi) = \xi, \qquad \xi = \xi^\dagger, \quad \mathrm{Tr}(\xi) = 0. \tag{14}$$

The resulting tangent norm is

$$\|\xi\|_{B,\rho}^2 \triangleq \frac{1}{2}\mathrm{Re}\ \mathrm{Tr}\big(L_\rho(\xi)^\dagger \xi\big)\,. \tag{15}$$

Write the projected velocity residual as $\Delta_\theta(t, \rho_t, y)$. With the Bures tangent norm, the Flow Matching objective becomes

$$\begin{aligned} \Delta_\theta(t, \rho_t, y) &\triangleq \mathcal{P}_T(v_\theta(t, \rho_t, h_\phi(y)) - u_t)\,, \\ \mathcal{L}_{\mathrm{BFM}}(\theta) &\triangleq \mathbb{E}_{\rho_\star, y, \rho_0, t}\left[\|\Delta_\theta(t, \rho_t, y)\|_{B,\rho_t}^2\right]. \end{aligned} \tag{16}$$

following the Flow Matching and Riemannian Flow Matching regression principle [10, 21]. The Lyapunov equation defines the local Bures tangent norm, so the loss penalizes velocity errors according to the same local geometry used by the Bures distance. A Frobenius loss can still preserve Hermitian trace-free outputs after projection, but it does not weight tangent directions by the local density-matrix geometry.

For finite-record training, we use a tomography-specific variant that keeps the same Bures velocity term and adds a low-shot endpoint consistency penalty. The predicted endpoint obtained by following the velocity for the remaining time is

$$\rho_\theta^+ = \Pi_{\mathcal{D}_d}\big(\mathrm{Exp}_{\rho_t}((1-t)v_\theta)\big)\,. \tag{17}$$

The sparse-record weight $w(n)$ increases the contribution of low-shot records while remaining bounded. The finite-record loss is

$$\begin{aligned} \mathcal{L}_{\mathrm{FR}}(\theta) &\triangleq w(n)\big[\mathcal{L}_{\mathrm{BFM}}(\theta) + \lambda d_{\mathrm{Bures}}^2(\rho_\theta^+, \rho_\star)\big], \\ w(n) &= \min\Big(w_{\max}, \max\Big(1, \Big(\frac{n_{\mathrm{ref}}}{n}\Big)^p\Big)\Big)\,. \end{aligned} \tag{18}$$

The endpoint term asks whether the predicted velocity, if followed for the remaining time, points toward the latent state that generated the record. The shot-dependent weight makes sparse records contribute more strongly without letting any one regime dominate training.

3.5 Training and Posterior Sampling

The training construction can be written compactly as

$$\begin{aligned} \rho_\star \sim p_{\mathrm{prior}}, \quad y \sim p(y \mid \rho_\star), \\ \rho_0 \sim p_0, \quad t \sim \mathcal{U}(0,1), \\ (\rho_t, u_t) = \big(\psi_t^B(\rho_0, \rho_\star), \dot\psi_t^B(\rho_0, \rho_\star)\big). \end{aligned} \tag{19}$$

The optimizer updates θ so that $v_\theta(t, \rho_t, h_\phi(y))$ matches u_t under Eq. (16) or Eq. (18). This makes the learned object a measurement-conditioned vector field on density matrices, not a decoder that emits one matrix from one record.

At test time the record is fixed and uncertainty comes from independent base states. For $s = 1, \ldots, S$, draw $\rho_0^{(s)} \sim p_0$ and integrate

$$\dot\rho_t^{(s)} = v_\theta(t, \rho_t^{(s)}, h_\phi(y)), \qquad \rho^{(s)} = \rho_1^{(s)}. \tag{20}$$

In the dense implementation, the same record context $h_\phi(y)$ is reused across all S trajectories, while the base density matrix is resampled for each posterior draw. The ODE integration applies the density-matrix projection used in Eq. (11) at intermediate and terminal states, so the posterior cloud remains a collection of physical states throughout sampling. Posterior spread is produced by the learned conditional transport, not by rerunning a separate reconstruction algorithm for each sample. The sample set $\{\rho^{(s)}\}_{s=1}^S$ is then a posterior object whose mean gives a reconstruction, whose spread gives credible balls, and whose pushforward through observables gives predictive uncertainty.

4 Experiments

The experiments test whether Bures-aligned Flow Matching improves posterior samples under finite records. We compare against Cholesky, vanilla Riemannian Flow Matching, log-density chart controls, and linear inversion across count, shadow, and mixed measurement records. The metrics are Bures distance, coverage error, held-out observable error, and amortized runtime.

4.1 Experimental Setup

We evaluate synthetic posterior tomography tasks over full-rank 4-qubit density matrices. The measurement regimes include count, classical-shadow, and mixed records, with shot budgets ranging from low-shot to more informative settings. These regimes stress posterior uncertainty: an underdetermined record requires a distribution over compatible states, not only a plausible single estimate.

Targets are full-rank four-qubit thermal states from an open-boundary transverse-field Ising Hamiltonian [26],

$$H(J,h) = -J\sum_{i=1}^{3} Z_i Z_{i+1} - h\sum_{i=1}^{4} X_i,$$
$$\rho_\star = \frac{\exp[-\beta H(J,h)]}{\mathrm{Tr}[\exp[-\beta H(J,h)]]}.$$

We sample $J \sim \mathcal{U}[0.5, 1.5]$, $h \sim \mathcal{U}[0.2, 1.8]$, and $\beta \sim \mathcal{U}[0.5, 5.0]$, then add a small identity floor to keep targets full rank.

Measurement records are sampled from product-Pauli Born probabilities. Shadow records draw a fresh product-Pauli basis for each shot, count records aggregate repeated shots over a fixed prefix of 32 product-Pauli settings, and mixed records concatenate both views for the same latent state. Regime names such as shadow-16 and mixed-32 denote the measurement family and shot budget.

Implementation summary. All dense posterior experiments use $d = 16$ density matrices. The calibrated finite-record comparison uses a Set Transformer record encoder [19], a time-conditioned velocity network, Adam with learning rate 10^{-3}, batch size 8, 420 optimization steps per comparison, 32 posterior samples per record, and Euler integration with step size 0.25, whereas the expanded geometry comparison in Table 2 uses 2,000 updates and batch size 32 across seven measurement regimes and five independent seeds. Appendix C gives the full implementation details.

4.2 Baselines and Metrics

The baseline set separates physical parameterization, generic manifold machinery, convenient positive-matrix charts, and point estimation. Cholesky parameterization tests whether positivity-preserving coordinates are enough. Vanilla Riemannian Flow Matching uses manifold machinery without the Bures-specific path and tangent construction. Log-Euclidean and mirror/Frobenius variants form a log-density chart family. Linear inversion acts as a fast point-estimation control that does not produce posterior samples.

Metrics. We report Bures distance, held-out observable error, posterior coverage error, and runtime. These metrics test state accuracy, downstream prediction, uncertainty calibration, and sampling cost. Appendix D gives the full evaluation protocol, including simulation-based calibration and physical-state diagnostics.

State-space accuracy. Bures distance is the primary geometry-aligned state metric. We use the trace-normalized positive-matrix form

$$\begin{aligned} F_{\mathrm{r}}(\rho,\sigma) &\triangleq \mathrm{Tr}\big[\big(\rho^{1/2}\sigma\rho^{1/2}\big)^{1/2}\big], \\ d^2_{\mathrm{Bures}}(\rho,\sigma) &\triangleq \mathrm{Tr}(\rho) + \mathrm{Tr}(\sigma) - 2F_{\mathrm{r}}(\rho,\sigma). \end{aligned} \tag{21}$$

Table 1: Calibrated finite-record geometry-baseline comparison. Lower is better for Bures distance, coverage error, observable error, and runtime. Values are mean ± standard deviation across eight finite-record regimes with five seeds each. BuresTomFlow has the best Bures distance, coverage error, and observable error among the posterior-sampler baselines. The coverage column reports signed empirical coverage at the nominal level 0.9, from which each run's coverage error is computed.

Method	Runs	Posterior quality ↓				Cost ↓
		Bures distance	Coverage error	Obs. error	Coverage	Runtime (s)
BuresTomFlow	40	0.3667 ± 0.0054	0.1000 ± 0.0000	0.0401 ± 0.0043	1.0000 ± 0.0000	236.2 ± 148.9
Cholesky geometry	40	0.6175 ± 0.0163	0.2030 ± 0.1612	0.2422 ± 0.0063	0.7205 ± 0.1876	236.4 ± 149.2
Vanilla Riemannian FM	40	0.6638 ± 0.0352	0.8347 ± 0.1098	0.1822 ± 0.0209	0.0653 ± 0.1098	234.8 ± 149.0
Log-density chart controls	80	0.8462 ± 0.0273	0.9000 ± 0.0000	0.2470 ± 0.0102	0.0000 ± 0.0000	235.9 ± 148.7
Linear inversion	40	0.4778 ± 0.1048	n/a	0.1001 ± 0.0468	n/a	0.34 ± 0.19

which reduces to $2 - 2\,\mathrm{Tr}[(\rho^{1/2}\sigma\rho^{1/2})^{1/2}]$ for trace-one density matrices [4, 7]. For posterior samplers, we compare the true state with the projected posterior center

$$\bar{\rho}_\theta(y) \triangleq \Pi_{\mathcal{D}_d}\left(\frac{1}{S}\sum_{s=1}^{S}\rho^{(s)}\right). \tag{22}$$

For point-estimation controls, the returned estimate plays the same role as this posterior center.

Held-out observables. Observable error asks whether posterior samples support accurate predictions on observables that are not directly used as the headline state-space distance:

$$\begin{aligned} \mathrm{ObsErr}(y) &\triangleq \frac{1}{|\mathcal{O}|}\sum_{O\in\mathcal{O}}\left|\langle O\rangle_{\bar{\rho}_\theta(y)} - \langle O\rangle_{\rho_\star}\right|, \\ \langle O\rangle_\rho &\triangleq \mathrm{Tr}(O\rho). \end{aligned} \tag{23}$$

Posterior calibration. Coverage error uses a credible-ball check. Given a nominal level α, let $r_\alpha(y)$ be the empirical α-quantile of $\{d_{\mathrm{Bures}}(\rho^{(s)}, \bar{\rho}_\theta(y))\}_{s=1}^{S}$. Across repeated synthetic tomography tasks (ρ_j, y_j), empirical coverage is

$$\begin{aligned} r_\alpha(y_j) &\triangleq \mathrm{Quantile}_\alpha\big(\{d_{\mathrm{Bures}}(\rho_j^{(s)}, \bar{\rho}_\theta(y_j))\}_{s=1}^{S}\big), \\ \mathrm{Cov}_\alpha &\triangleq \frac{1}{N}\sum_{j=1}^{N}\mathbf{1}[d_{\mathrm{Bures}}(\rho_j, \bar{\rho}_\theta(y_j)) \le r_\alpha(y_j)], \\ \mathrm{CovErr}_\alpha &\triangleq |\mathrm{Cov}_\alpha - \alpha|. \end{aligned} \tag{24}$$

A low distance error alone is not sufficient for posterior tomography. The sample cloud must also match empirical coverage. At the nominal level $\alpha = 0.9$, BuresTomFlow has empirical coverage 1.0000 in all 40 runs of the main comparison. Its coverage error of 0.1000 therefore reflects overcoverage. Linear inversion is marked as a point-estimation control.

Runtime. Runtime in Table 1 reports total wall-clock time combining the one-time training loop and posterior sample generation, testing whether the Bures-geometric quality gains are obtained at total computational cost comparable to the learned geometry baselines. For BuresTomFlow, the 236.2 ± 148.9 s total in the 420-step, batch-size-8 comparison comprises an average of 235.75 s for training and 0.490 ± 0.117 s in total to generate 32 posterior samples for each of 160 held-out records (3.06 ms per 32-sample record query).

4.3 Main Posterior Tomography Results

Table 1 gives the compact multi-seed geometry comparison. BuresTomFlow has the best posterior-sampler quality metrics while running at comparable learned-sampler cost.

The baseline pattern isolates the source of the improvement. Cholesky already encodes positive definiteness, so the advantage is not simply physical validity. Vanilla Riemannian Flow Matching

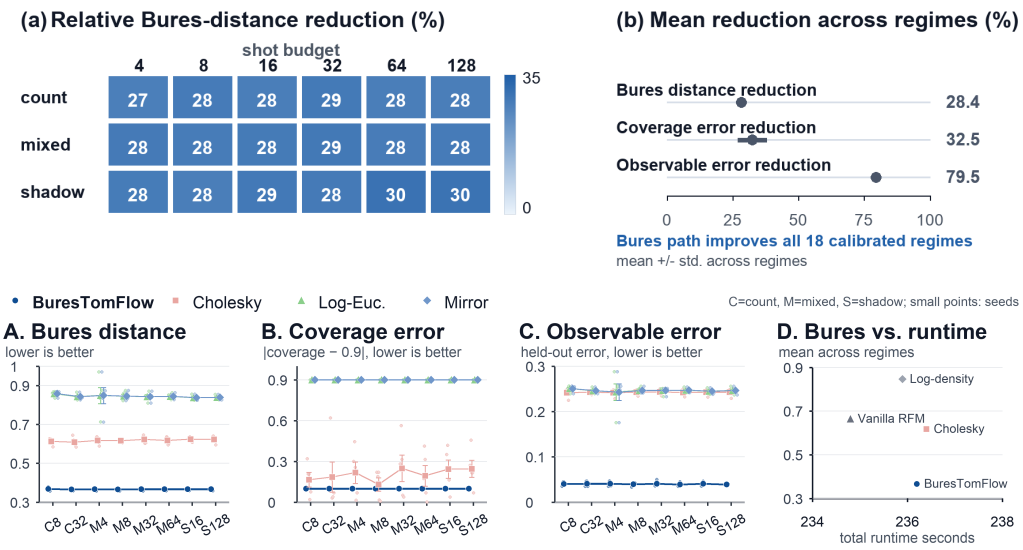

Figure 2: Measurement-regime analysis. Top: the complete three-family, six-budget sweep, reporting relative Bures-distance reduction and mean reductions in Bures distance, coverage error, and held-out observable error. The improvements over the geometry controls persist across the tested shot budgets in all three measurement families. Bottom: the finite-record summary, reporting Bures distance, coverage error, and held-out observable error with seed-level points and regime means, and Bures distance versus total runtime for the learned-sampler rows of Table 1. In panel B, the BuresTomFlow coverage error of 0.1 corresponds to empirical coverage of 1.0.

uses manifold machinery without the Bures-specific path and tangent construction, so the advantage is not simply the use of a Riemannian solver. Log-density chart controls show that convenient positive-matrix coordinates can still be poorly calibrated as posterior samplers.

Point-estimation control. Linear inversion is the fast-reconstruction baseline. Its role clarifies the contribution: the posterior formulation targets calibrated uncertainty over physical density matrices, not only a single-state estimate.

Figure 2 expands Table 1 across measurement families, shot budgets, and seed-level finite-record diagnostics.

4.4 Finite-Record Measurement Regimes

Figure 2 reports the comparison by measurement regime. The seed-level panels show that the ordering is not an artifact of averaging across count, mixed, and shadow records.

The regime view matters because the method is designed for finite-record ambiguity. Mixed records test heterogeneous observations, and low-shot rows test whether the posterior remains informative without collapsing uncertainty. Appendix Tables 8 and 9 give the wider family-budget grid.

Table 2 tests whether this geometry ordering persists under a larger shared training budget and split-conformal region evaluation [20]. Across the expanded comparison, BuresTomFlow achieves the lowest posterior-center Bures distance and held-out observable error among the compared methods. When posterior-center Bures balls are calibrated from center-to-truth distances on a disjoint split, all four methods attain comparable coverage near the nominal level while BuresTomFlow yields the smallest mean Bures radius.

To evaluate the generated sample cloud beyond truth-centered summaries, Table 3 compares each learned sampler directly against a deterministic dense-grid Bayesian reference posterior constructed under the same thermal-state prior and conditioned on the same stored finite measurement record. Discrepancy between the learned and reference posterior clouds is measured by an energy statistic

Table 2: Expanded geometry comparison using 2,000 updates, batch size 32, seven regimes, and five seeds. The split-conformal columns report empirical coverage and mean Bures radius for posterior-center Bures balls calibrated from center-to-truth Bures distances on a disjoint calibration split at nominal level 0.9. Values are mean $\pm$ standard deviation, except held-out observable error, which is reported as a mean. BuresTomFlow achieves the lowest posterior-center Bures distance, observable error, and conformal Bures radius at comparable near-nominal coverage.

Method	Posterior-center accuracy		Split-conformal regions	
	Bures distance ↓	Obs. error ↓	Coverage	Mean radius ↓
BuresTomFlow	0.2716 ± 0.095	0.0487	0.9130 ± 0.0450	0.4992 ± 0.1515
Cholesky	0.4597 ± 0.039	0.1478	0.8955 ± 0.0374	0.8804 ± 0.0335
Log-Euclidean	0.3552 ± 0.081	0.0956	0.8964 ± 0.0329	0.7468 ± 0.0604
Vanilla RFM	0.4710 ± 0.043	0.1087	0.9039 ± 0.0265	0.7194 ± 0.0333

Table 3: Posterior-to-posterior comparison against a same-prior, same-record dense-grid Bayesian reference posterior using an energy statistic with the unsquared Bures-Wasserstein ground distance (↓). Values are mean $\pm$ standard deviation across five seeds. BuresTomFlow achieves the lowest mean discrepancy in all four tested regimes, with reductions of 63.5% on mixed-4 and 37.4% on mixed-16 relative to the strongest alternative.

Regime	Vanilla RFM	Log-Euclidean	Cholesky	BuresTomFlow
mixed-4	0.5253 ± 0.040	0.4521 ± 0.214	0.4436 ± 0.067	0.1620 ± 0.024
mixed-16	0.5289 ± 0.074	0.3505 ± 0.078	0.4815 ± 0.066	0.2193 ± 0.035
shadow-4	0.5983 ± 0.110	0.4945 ± 0.098	0.4292 ± 0.037	0.4108 ± 0.197
shadow-16	0.5968 ± 0.095	0.5212 ± 0.119	0.4156 ± 0.071	0.4046 ± 0.203

[28] using the unsquared Bures-Wasserstein distance as its ground distance. Across the five-seed evaluation, BuresTomFlow attains the lowest mean Bures-ground energy statistic in all four tested regimes, reducing the mean discrepancy relative to the strongest alternative by 63.5% on mixed-4 and 37.4% on mixed-16.

4.5 Control Experiments

Table 1 also functions as the component comparison: the controls remove Bures transport, Bures tangent supervision, or posterior sampling. Their ordering ties the improvement to the Bures-specific path and tangent construction rather than physical validity alone. The full control table is deferred to Appendix E.

We also evaluate BuresTomFlow under standard simulation-based calibration [29] across 200 prior-predictive tasks in each of the count-8 and count-32 regimes using 63 posterior samples per task and six fixed scalar functionals (four Pauli expectations, purity, and the largest eigenvalue). In these tested regimes, the resulting BuresTomFlow rank distributions remain significantly non-uniform across the evaluated functionals, as detailed in Appendix G.5.

5 Conclusions, Limitations, and Future Work

We introduced BuresTomFlow, a Bures-geometric conditional Flow Matching sampler for finite-record posterior quantum tomography. The method treats tomography as posterior sampling over physical density matrices rather than as single-state reconstruction, and the experiments isolate the gains from the Bures-specific path and tangent construction. The current dense implementation is evaluated in four-qubit tomography simulations with known ground-truth density matrices. In the main comparison under the structured four-qubit thermal-state prior, BuresTomFlow reduces posterior-center Bures distance, credible-ball coverage error, and held-out observable error relative to the tested geometry baselines. The Bures-kernel scaling results support the feasibility of the sampler's core geometric operations at the larger matrix sizes tested. End-to-end posterior tomography at larger qubit counts remains future work.

Acknowledgments and Disclosure of Funding

Lu Wei and Yufeng Wang were supported by research assistantships at Stony Brook University. The authors declare no competing interests.

References

[1] Scott Aaronson. Shadow tomography of quantum states. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 325–338, 2018. doi: 10.1145/3188745.3188802.

[2] Shahnawaz Ahmed, Carlos Sánchez Muñoz, Franco Nori, and Anton Frisk Kockum. Quantum state tomography with conditional generative adversarial networks. *Physical Review Letters*, 127:140502, 2021. doi: 10.1103/PhysRevLett.127.140502.

[3] Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209): 1–80, 2025. URL `https://jmlr.org/papers/v26/23-1605.html`.

[4] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures-Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 37(2):165–191, 2019. doi: 10.1016/j.exmath.2018.01.002.

[5] Robin Blume-Kohout. Optimal, reliable estimation of quantum states. *New Journal of Physics*, 12:043034, 2010. doi: 10.1088/1367-2630/12/4/043034.

[6] Samuel L. Braunstein and Carlton M. Caves. Statistical distance and the geometry of quantum states. *Physical Review Letters*, 72(22):3439–3443, 1994. doi: 10.1103/PhysRevLett.72.3439.

[7] 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. doi: 10.1090/S0002-9947-1969-0236719-2.

[8] Juan Carrasquilla, Giacomo Torlai, Roger G. Melko, and Leandro Aolita. Reconstructing quantum states with generative models. *Nature Machine Intelligence*, 1:155–161, 2019. doi: 10.1038/s42256-019-0028-1.

[9] Peter J. 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. doi: 10.1088/2632-2153/ac362b.

[10] Ricky T. Q. Chen and Yaron Lipman. Flow matching on general geometries. In *International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=g7ohDlTITL`.

[11] Antoine Collas, Ce Ju, Nicolas Salvy, and Bertrand Thirion. Riemannian flow matching for brain connectivity matrices via pullback geometry. In *Advances in Neural Information Processing Systems*, 2025. URL `https://openreview.net/forum?id=NY3LzmUXl7`.

[12] Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

[13] Andreas Elben, Steven T. Flammia, Hsin-Yuan Huang, Richard Kueng, John Preskill, Benoît Vermersch, and Peter Zoller. The randomized measurement toolbox. *Nature Reviews Physics*, 5:9–24, 2023. doi: 10.1038/s42254-022-00535-2.

[14] Christopher Granade, Joshua Combes, and D. G. Cory. Practical Bayesian tomography. *New Journal of Physics*, 18:033024, 2016. doi: 10.1088/1367-2630/18/3/033024.

[15] David Gross, Yi-Kai Liu, Steven T. Flammia, Stephen Becker, and Jens Eisert. Quantum state tomography via compressed sensing. *Physical Review Letters*, 105:150401, 2010. doi: 10.1103/PhysRevLett.105.150401.

[16] Zdeněk Hradil. Quantum-state estimation. *Physical Review A*, 55(3):R1561–R1564, 1997. doi: 10.1103/PhysRevA.55.R1561.

[17] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.

[18] Richard Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41(12):2315–2323, 1994. doi: 10.1080/09500349414552171.

[19] Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3744–3753, 2019. URL http://proceedings.mlr.press/v97/lee19d.html.

[20] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018. doi: 10.1080/01621459.2017.1307116.

[21] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

[22] Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow matching guide and code, 2024. URL https://arxiv.org/abs/2412.06264.

[23] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

[24] Wenzhe Ouyang, Qi Ye, Jinghua Wang, Zenglin Xu, and Jiming Chen. RFMPose: Generative category-level object pose estimation via riemannian flow matching. In *Advances in Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6aaixHco6C.

[25] Matteo G. A. Paris and Jaroslav Rehacek, editors. *Quantum State Estimation*, volume 649 of *Lecture Notes in Physics*. Springer, 2004. doi: 10.1007/b98673.

[26] Subir Sachdev. *Quantum Phase Transitions*. Cambridge University Press, 2 edition, 2011. doi: 10.1017/CBO9780511973765.

[27] Tobias Schmale, Moritz Reh, and Martin Gärttner. Efficient quantum state tomography with convolutional neural networks. *npj Quantum Information*, 8:115, 2022. doi: 10.1038/s41534-022-00621-4.

[28] Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and RKHS-based statistics in hypothesis testing. *Annals of Statistics*, 41(5): 2263–2291, 2013. doi: 10.1214/13-AOS1140.

[29] Sean Talts, Michael Betancourt, Daniel Simpson, Aki Vehtari, and Andrew Gelman. Validating Bayesian inference algorithms with simulation-based calibration, 2018.

[30] Yehui Tang, Mabiao Long, and Junchi Yan. QuaDiM: A conditional diffusion model for quantum state property estimation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P7f55HQtV8.

[31] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=CD9Snc73AW.

[32] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14:447–450, 2018. doi: 10.1038/s41567-018-0048-5.

[33] Armin Uhlmann. The transition probability in the state space of a *-algebra. *Reports on Mathematical Physics*, 9(2):273–279, 1976. doi: 10.1016/0034-4877(76)90060-4.

[34] Jesse van Oostrum. Bures–wasserstein geometry for positive-definite hermitian matrices and their trace-one subset. *Information Geometry*, 5(2):405–425, 2022. doi: 10.1007/s41884-022-00069-7.

[35] Jonas Bernhard Wildberger, Maximilian Dax, Simon Buchholz, Stephen R. Green, Jakob H. Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. In *Advances in Neural Information Processing Systems*, volume 36, pages 16837–16864, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3663ae53ec078860bb0b9c6606e092a0-Abstract-Conference.html.

[36] Yuchen Zhu, Tianrong Chen, Evangelos A. Theodorou, Xie Chen, and Molei Tao. Quantum state generation with structure-preserving diffusion model, 2024.

A Measurement Protocol and Estimators

Finite-record quantum tomography. Quantum state tomography is a statistical inverse problem on the convex set of density matrices. Standard approaches include linear inversion, maximum likelihood, and Bayesian estimation [5, 16, 25]. A point estimator can be accurate when the measurement record is highly informative, but finite records typically leave many physical states with comparable likelihood. Practical Bayesian tomography addresses this ambiguity by representing uncertainty over states rather than returning a single reconstruction [14]. BuresTomFlow follows this posterior viewpoint while amortizing inference: after training, the same conditional sampler can be queried for density-matrix posterior samples from a new finite record.

Randomized measurements and shadow records. The conditioning input is a finite measurement record rather than the true state. We use count, mixed, and randomized shadow records to represent increasingly heterogeneous observation settings. Shadow tomography and classical shadows use randomized measurements to predict many observables without reconstructing every matrix entry [1, 17]. Broader randomized-measurement protocols are reviewed in Ref. [13]. In this paper, shadow records are not used as a substitute for posterior uncertainty. They are one family of finite records from which the sampler must infer a distribution over compatible density matrices.

Synthetic state family. The main dense benchmark uses full-rank four-qubit thermal states from an open-boundary transverse-field Ising Hamiltonian [26],

$$\begin{aligned} H(J,h) &= -J\sum_{i=1}^{3} Z_i Z_{i+1} - h\sum_{i=1}^{4} X_i, \\ \rho_\star &= \frac{\exp(-\beta H(J,h))}{\mathrm{Tr}[\exp(-\beta H(J,h))]}. \end{aligned} \tag{25}$$

The implementation samples $J \sim U[0.5, 1.5]$, $h \sim U[0.2, 1.8]$, and $\beta \sim U[0.5, 5.0]$, then mixes in a small identity component to keep the density matrix full rank. This construction gives a controlled target distribution with noncommuting terms, nontrivial eigenstructure, and exact access to the latent density matrix for evaluation. The full-rank floor is a numerical device for stable matrix square roots and logarithmic tangent calculations. It is not additional measurement information.

Record construction. All measurement records are encoded as fixed-width tokens carrying a product-Pauli basis label, outcome statistics, the shot budget, and a token type. Each four-qubit token has dimension 24: sixteen entries encode the per-qubit basis labels, four entries encode bit or mean-bit statistics, one entry stores the all-plus/count statistic, one entry stores the scaled shot budget, and two entries encode whether the token is a snapshot or count token. In shadow mode, each token is one randomized Pauli snapshot: a product-Pauli basis is sampled, a bitstring outcome is drawn from the Born probabilities of the latent state, and the token stores the measured basis and outcome summary. In count mode, the experiment uses aggregate count tokens over a fixed prefix of 32 product-Pauli settings. Each token stores mean bit statistics from repeated shots under the same basis. Mixed mode concatenates the shadow and count records for the same latent state. Thus the three measurement families differ in the information presented to the encoder, while the posterior target remains the same density matrix conditioned on the finite record.

Born sampling for product-Pauli records. For a product-Pauli setting $p = (p_1, \ldots, p_4) \in \{X, Y, Z\}^4$ and bitstring $b \in \{0, 1\}^4$, let $M_{p,b}$ denote the tensor product of the one-qubit projectors onto the measurement eigenvectors specified by p and b. The record simulator samples outcomes from

$$\Pr_{\rho_\star}(b \mid p) = \mathrm{Tr}(M_{p,b}\rho_\star). \tag{26}$$

Shadow records draw a fresh product-Pauli setting for each shot. Count records reuse a fixed prefix of the 3^4 product-Pauli settings and store aggregate bit statistics for each setting. Mixed records concatenate both views. Consequently, all finite-record comparisons use the same Born rule and the same latent-state family, while varying only how the measurement information is packaged for the conditional encoder.

Relation to classical-shadow estimators. A randomized Pauli snapshot can also be converted into the classical-shadow estimator [17]. If U_{p_q} maps the measured Pauli basis on qubit q to the computational basis and the observed bit is b_q, the local inverse estimator is

$$\begin{aligned} \widehat{\rho}_q &= 3U_{p_q}^{\dagger}|b_q\rangle\langle b_q|U_{p_q} - I, \\ \widehat{\rho}_{\mathrm{shadow}} &= \bigotimes_{q=1}^{4} \widehat{\rho}_q. \end{aligned} \tag{27}$$

This estimator explains why randomized Pauli snapshots are informative for observable prediction. BuresTomFlow uses the same physical measurement interface, but it does not average Eq. (27) into a single reconstructed state. The snapshot is kept as a conditioning token, and the learned model outputs posterior samples over density matrices compatible with the finite record.

Posterior estimators. For each measurement record y, the sampler returns a finite posterior cloud $\{\rho^{(s)}(y)\}_{s=1}^{S}$ of positive trace-one matrices. Point summaries use the projected posterior center

$$\hat{\rho}(y) = \Pi_{\mathcal{D}_d}\left(S^{-1}\sum_{s=1}^{S}\rho^{(s)}(y)\right), \tag{28}$$

where $\Pi_{\mathcal{D}_d}$ is the density-matrix projection defined in Appendix B. The full sample cloud is used for credible-ball coverage and simulation-based calibration. Linear inversion is evaluated as a reconstruction baseline but is not assigned posterior coverage metrics, because it produces a single matrix rather than a sample cloud.

Neural and amortized tomography. Neural quantum state tomography uses learned representations or generative models to reconstruct states from measurement statistics [8, 32]. Conditional generative tomography is especially close in spirit, since it learns a measurement-conditioned generator whose output is mapped to a physical density matrix [2]. Later work explores convolutional and attention-based reconstruction [9, 27], while diffusion-style quantum generative models target structure-preserving state generation or property estimation [30, 36]. The present work differs in its evaluation target: it asks whether a learned sampler returns calibrated posterior samples over density matrices, not only whether it reconstructs one accurate state.

Prior components and contributions. Table 4 lists the prior work that BuresTomFlow builds on, the role each component plays in this paper, and the components introduced here.

Table 4: Prior components used by BuresTomFlow and the contributions of this paper.

Component	Role in this paper
Flow Matching and Riemannian Flow Matching [10, 21]	Velocity-regression training principle
Bures-Wasserstein geometry [4, 34]	Positive-cone transport maps and tangent metric
Shadow tomography, classical shadows, randomized measurements [1, 13, 17]	Measurement protocols that generate the conditioning records; classical-shadow estimator as a comparison
Bayesian quantum tomography [5, 14]	Posterior target $p(\rho \mid y)$
Neural tomography [8, 32]	Comparison class of learned reconstruction methods
This paper	Measurement-conditioned posterior sampler over density matrices combining projected Bures-Wasserstein conditional paths, Bures-aligned tangent supervision, density-matrix projection, and posterior evaluation by Bures distance, credible-ball coverage, and held-out observable error

B Bures Geometry and Projected Density-Matrix Flow

Bures geometry for density matrices. Fidelity and Bures distance compare quantum states through distinguishability rather than entrywise matrix error [7, 18, 33]. At the infinitesimal level, the Bures

metric is proportional, up to convention, to the symmetric-logarithmic-derivative Fisher metric underlying the quantum Cramér–Rao bound [6]. On the positive-definite cone, Bures-Wasserstein geometry has closed-form transport maps, geodesics, means, and barycentres [4]. The trace-one density-matrix set is not a geodesic submanifold of this cone [34]. We therefore use positive-cone Bures-Wasserstein interpolation followed by projection back to the trace-one state space, rather than claiming that the trace-one path is itself an exact cone geodesic.

Distance used in the experiments. For positive matrices A and B, the Bures-Wasserstein distance used by the metrics is

$$\begin{aligned} F_{\mathrm{r}}(A,B) &\triangleq \mathrm{Tr}\big[(A^{1/2}BA^{1/2})^{1/2}\big], \\ d^2_{\mathrm{BW}}(A,B) &\triangleq \mathrm{Tr}(A)+\mathrm{Tr}(B)-2F_{\mathrm{r}}(A,B). \end{aligned} \tag{29}$$

For trace-one density matrices, d_{BW} is the paper's d_{Bures}, and $d^2_{\mathrm{Bures}}(A,B) = 2 - 2F_{\mathrm{r}}(A,B)$. The implementation computes Eq. (29) by Hermitian eigendecomposition with a small eigenvalue floor. Thus the reported Bures distance and the credible-ball radius use the same numerical distance.

Positive-cone interpolation. For two positive-definite matrices A and B, the Bures-Wasserstein transport map, positive-cone path, and velocity target are

$$\begin{aligned} T_{A\to B} &= A^{-1/2}\left(A^{1/2}BA^{1/2}\right)^{1/2}A^{-1/2}, \\ \Gamma_t(A,B) &= H_t^\dagger A H_t, \\ H_t &= (1-t)I + tT_{A\to B}, \qquad t\in[0,1], \\ \dot\Gamma_t(A,B) &= K^\dagger A H_t + H_t^\dagger A K, \\ K &= T_{A\to B} - I. \end{aligned} \tag{30}$$

This path has the correct endpoints on the positive cone. Since $T_{A\to B}AT_{A\to B} = B$, Eq. (30) gives $\Gamma_0(A,B) = A$ and $\Gamma_1(A,B) = B$. Equivalently, $\Gamma_t = (I+tK)^\dagger A(I+tK)$, so differentiating gives the last line of Eq. (30). The equation states which geometric path is being regressed before any neural approximation is introduced. The code uses the same operation through the manifold logarithm and exponential map: $\log_A(B) = T_{A\to B} - I$ and $\exp_A(U) = (I+U)^\dagger A(I+U)$, followed by the density projection.

Projection to physical states. The sampler must return a density matrix, so the positive-cone path is projected to $\mathcal{D}_d = \{\rho \succeq 0, \mathrm{Tr}(\rho) = 1\}$. For a Hermitian matrix $X = Q\,\mathrm{diag}(\lambda_1,\dots,\lambda_d)Q^\dagger$ and the small numerical floor $\delta > 0$ used in Eq. (11), the projection used throughout the experiments is

$$\Pi_{\mathcal{D}_d}(X) = \frac{Q\,\mathrm{diag}(\max(\lambda_1,\delta),\dots,\max(\lambda_d,\delta))Q^\dagger}{\sum_{i=1}^d \max(\lambda_i,\delta)}. \tag{31}$$

The learned path is therefore $\rho_t = \Pi_{\mathcal{D}_d}(\Gamma_t(\rho_0,\rho_\star))$, and its velocity is obtained by differentiating this projected path. The projection is part of the mathematical object being trained and evaluated. It is not an after-the-fact validity correction applied only to generated samples.

Trace-normalized aligned-factor path. A generic projection onto the trace-one set need not preserve geodesics. The projected path used here has a more specific form. Let T be the Bures-Wasserstein transport map from ρ_0 to the smoothed target ρ_1^η, and define the factors $G_0 = \rho_0^{1/2}$ and $G_1 = T\rho_0^{1/2}$, so that $G_0G_0^\dagger = \rho_0$ and $G_1G_1^\dagger = \rho_1^\eta$. Since $G_0^\dagger G_1 = (\rho_0^{1/2}\rho_1^\eta\rho_0^{1/2})^{1/2} \succeq 0$, these factors are optimally aligned. Across the 19,200 evaluated training path states, the eigenvalue floor $\delta = 10^{-10}$ is never active, and the minimum pre-projection eigenvalue is 1.0×10^{-6}. At every evaluated training path state the projection therefore reduces to trace normalization:

$$\gamma_t = (1-t)G_0 + tG_1, \qquad \rho_t = \frac{\gamma_t\gamma_t^\dagger}{\|\gamma_t\|_F^2}. \tag{32}$$

This path remains in the trace-one manifold and traces the induced-metric geodesic of van Oostrum [34, Theorem 7] with a non-constant-speed parameterization. The chord parameter $t\in[0,1]$ is not generally constant speed in the Bures-Wasserstein angle; a constant-speed parameterization

is obtained by spherical interpolation of the aligned factors, and Flow Matching permits either differentiable parameterization [10, 21]. Writing $\theta_{\mathrm{BW}}(\rho,\sigma) = \arccos F_{\mathrm{r}}(\rho,\sigma)$ for the induced Riemannian distance on the trace-one set, the reported Bures distance satisfies [34]

$$d_{\mathrm{Bures}}(\rho,\sigma) = 2\sin\left(\frac{\theta_{\mathrm{BW}}(\rho,\sigma)}{2}\right). \tag{33}$$

Tangent projection. The velocity model outputs a Hermitian trace-free tangent before the Bures loss is evaluated. For any matrix-valued velocity proposal U, the tangent projection implemented by the sampler is

$$\begin{aligned} \mathcal{P}_T(U) &= \mathrm{Herm}(U) - \frac{1}{d}\,\mathrm{Tr}(\mathrm{Herm}(U))I, \\ \mathrm{Herm}(U) &= \frac{U+U^\dagger}{2}. \end{aligned} \tag{34}$$

This projection enforces the linearized trace-one constraint and removes anti-Hermitian numerical components before either Bures or Frobenius velocity losses are evaluated. It also makes the geometry comparison sharper. The non-Bures learned controls are not failing because they output non-Hermitian trace-changing velocities. They are trained under a different path or velocity metric.

Bures tangent inner product. The Bures velocity loss uses the local tangent metric induced by a Lyapunov equation. For a density matrix ρ and tangent directions U, V, define $L_\rho(U)$ and the tangent inner product by

$$\begin{aligned} L_\rho(U)\rho + \rho L_\rho(U) &= U, \\ g_\rho(U,V) &\triangleq \frac{1}{2}\Re\,\mathrm{Tr}\big(L_\rho(U)^\dagger V\big). \end{aligned} \tag{35}$$

The implementation solves Eq. (35) in the eigenbasis of ρ, with the same eigenvalue floor used by the Bures distance. This is the step that turns the geometric construction into a concrete training objective: velocity errors are measured in a state-dependent tangent metric, not in a fixed Euclidean norm.

Projected flow on density matrices. The constrained state space is not a cosmetic detail. The sampler must output positive, trace-one matrices. The training target must be expressed in a tangent geometry that respects this domain, and the uncertainty checks must judge posterior samples by physically meaningful state distances. The Bures construction ties these requirements together. Probability paths, velocity regression, and posterior evaluation all use the same fidelity-aligned geometry. Projection back to the density-matrix set is therefore part of the method, not a post-hoc repair step.

C Training Objective and Implementation Details

Conditional Flow Matching objective. BuresTomFlow is trained as a conditional Flow Matching model [21] on a matrix manifold, following the general principle that probability paths and velocity targets should be expressed in the geometry of the sample space [10]. Given a measurement record encoding $c = h_\phi(y)$, the neural velocity field $v_\theta(t,\rho_t,c)$ regresses the Bures-Wasserstein path velocity between a base density matrix and the target state. The loss uses the local Bures tangent inner product rather than a flat Frobenius loss, so that velocity errors are penalized in the same geometry used for state comparison.

More explicitly, each training example draws $(\rho_0,\rho_\star,y,t)$, forms the projected Bures path ρ_t, and differentiates that path to obtain $\dot\rho_t$. The Bures-geometric objective is

$$\begin{aligned} \Delta_\theta(t,\rho_t,y) &\triangleq \mathcal{P}_T(v_\theta(t,\rho_t,h_\phi(y)) - \dot\rho_t), \\ \mathcal{L}_{\mathrm{Bures}} &\triangleq \mathbb{E}\left[g_{\rho_t}(\Delta_\theta(t,\rho_t,y),\Delta_\theta(t,\rho_t,y))\right]. \end{aligned} \tag{36}$$

where g_{ρ_t} is Eq. (35) and $\mathcal{P}_T$ is Eq. (34). The Frobenius controls replace g_{ρ_t} with a flat matrix mean-squared error after the same Hermitian trace-free projection. This is why the control comparison isolates the metric and path choice rather than merely testing output validity.

Shared evaluation budget. All learned posterior samplers are evaluated as amortized samplers after training. The geometry controls use the same posterior-sampling role but replace the path, coordinates, or velocity loss. This design makes the comparison about geometry rather than about whether a method can enforce positivity at all. The point-estimation control is kept separate because it answers a different question: how well a single reconstructed matrix can perform without representing posterior uncertainty.

Experimental draw. Each training draw samples a four-qubit target state $\rho_\star$, samples a finite measurement record y from $\rho_\star$, samples a base density matrix ρ_0, and samples $t \in (0, 1)$. The Bures, Cholesky, log-density, mirror, and projected-density controls receive the same record encoder input and the same posterior-sampling task. For learned samplers, evaluation fixes a held-out record and generates a posterior cloud by drawing independent base density matrices and integrating the learned vector field. The calibration split is used only to choose the credible-ball radius scale. Posterior metrics are then reported on held-out finite-record tasks.

Implementation details. The dense posterior-tomography experiments use four-qubit density matrices, so the matrix dimension is $d = 16$. The calibrated finite-record comparison uses a Set Transformer record encoder [19] with two masked self-attention blocks, four attention heads, encoder hidden dimension 128, and context dimension 128. The velocity network receives real and imaginary parts of the flattened density matrix, the scalar time, and the record context, uses hidden dimension 256, and outputs a Hermitian trace-free velocity on the 16×16 matrix space. Optimization uses Adam with learning rate 10^{-3}, batch size 8, 420 training steps, and 32 posterior samples per held-out record. Credible-ball calibration uses a held-out calibration set before evaluating coverage. Learned samplers are integrated with Euler steps of size 0.25. The eigenvalue floor δ in Eq. (11) is 10^{-10}, and the derivative of the projected path is computed by PyTorch automatic differentiation through `torch.linalg.eigh`, which is valid away from the floor and from repeated eigenvalues. Sampling requires no gradient through the projection. These details fix the computation needed to interpret the reported tables, while the conclusions remain tied to the posterior metrics rather than to the optimizer itself.

Sampling after training. At evaluation time, a held-out measurement record is encoded once and then reused across independent base density matrices. Each base density matrix is sampled from a random full-rank density-matrix distribution, the learned ODE is integrated from $t = 0$ to $t = 1$, and the manifold projection keeps the generated endpoint in $\mathcal{D}_d$. Repeating this process gives the posterior cloud used by Bures distance, observable error, coverage, and simulation-based calibration. This is the amortized setting evaluated in the runtime figures: training is a one-time cost, while posterior queries reuse the learned sampler.

D Evaluation Protocol

We use the following metrics for all supplementary tables and figures.

Posterior quality. Bures distance measures reconstruction quality in the same geometry used by the method. Held-out observable error checks whether the projected posterior center predicts quantities not used to fit the sampler. Coverage error tests whether credible balls formed from posterior samples contain the held-out truth at the intended nominal level. These three quantities are complementary: a sampler may have a good point estimate while still producing overconfident or poorly calibrated samples. For held-out tasks $\{(\rho_j, y_j)\}_{j=1}^N$, posterior samples $\{\rho_j^{(s)}\}_{s=1}^S$, and fixed held-out Pauli

observables $\mathcal{O} = \{ZIII, IZII, ZZII, XXXX\}$, the reported posterior-center metrics are

$$\begin{aligned} \hat{\rho}_j &= \Pi_{\mathcal{D}_d}\left(S^{-1}\sum_{s=1}^{S}\rho_j^{(s)}\right), \\ E_{\mathrm{Bures}} &= \frac{1}{N}\sum_{j=1}^{N} d_{\mathrm{Bures}}(\hat{\rho}_j, \rho_j), \\ E_{\mathrm{obs}} &= \frac{1}{N|\mathcal{O}|}\sum_{j=1}^{N}\sum_{O\in\mathcal{O}} \left|\mathrm{Tr}(O\hat{\rho}_j) - \mathrm{Tr}(O\rho_j)\right|. \end{aligned} \tag{37}$$

The evaluation observables in $\mathcal{O}$ were sampled uniformly without replacement from the non-identity four-qubit Pauli strings before model comparison and were then fixed across all methods, regimes, and seeds. They represent 1.57% of this set.

Calibration protocol. Coverage is measured by credible balls centered at the projected posterior center. A nominal level fixes the empirical posterior radius, and repeated finite-record tasks test how often the true state lies inside that ball. For nominal level α, define the empirical radius, coverage, coverage error, and calibrated radius scale by

$$\begin{aligned} r_\alpha(y_j) &= \mathrm{Quantile}_\alpha\left(\left\{d_{\mathrm{Bures}}(\rho_j^{(s)}, \hat{\rho}_j)\right\}_{s=1}^{S}\right), \\ \widehat{\mathrm{Cov}}_\alpha(s) &= \frac{1}{N}\sum_{j=1}^{N}\mathbf{1}\{d_{\mathrm{Bures}}(\rho_j, \hat{\rho}_j) \le s\, r_\alpha(y_j)\}, \\ E_{\mathrm{cov}} &= \left|\widehat{\mathrm{Cov}}_\alpha(s^\star) - \alpha\right|, \\ s^\star &= \arg\min_{s\in\mathcal{S}}\left|\widehat{\mathrm{Cov}}_\alpha(s) - \alpha\right|. \end{aligned} \tag{38}$$

The scale grid $\mathcal{S}$ is evaluated on a separate calibration split, with ties resolved toward the scale closest to one. The held-out coverage error is then computed with $s^\star r_\alpha(y_j)$. This separates radius calibration from the reported held-out posterior-quality metrics.

Split-conformal posterior-center regions. Whereas the credible-ball check in Eq. (38) rescales the empirical spread of generated posterior samples, the split-conformal evaluation in Table 2 calibrates prediction regions directly from the Bures distance between the true density matrix and each method's projected posterior-center estimate [20]. Evaluating this center-to-truth nonconformity score on a disjoint calibration split and taking its standard finite-sample split-conformal quantile fixes the Bures radius before test evaluation. At test time, the prediction region for each held-out record is the Bures ball of that calibrated radius centered at the method's posterior-center estimate. Under prior-predictive exchangeability of the calibration and test tasks, this procedure provides finite-sample marginal coverage of the true state. Table 2 compares the resulting held-out empirical coverage and mean Bures radius across the learned geometry baselines at nominal coverage 0.9.

Simulation-based calibration. We also evaluate prior-predictive distributional calibration of BuresTomFlow using standard simulation-based calibration (SBC) on fixed scalar functionals [29]. For each prior-predictive task (ρ_j, y_j) and scalar functional $f : \mathcal{D}_d \to \mathbb{R}$, we evaluate $f(\rho_j)$ on the true state and $f(\rho_j^{(\ell)})$ on the posterior samples conditioned on y_j, rank the true functional value among the posterior-sample values, and compare the empirical rank distribution across tasks against the discrete-uniform reference expected under SBC. This diagnostic evaluates BuresTomFlow on 200 prior-predictive tasks in each of the count-8 and count-32 scenarios using 63 posterior samples per task, which produces integer ranks from 0 to 63. We test six fixed scalar functionals consisting of four Pauli expectations (including Pauli $ZIII$), purity, and the largest eigenvalue. Appendix G.5 reports representative rank means and standard deviations, χ^2 p-values, and 20-bin ECE values for Pauli $ZIII$ and purity in the count-8 and count-32 scenarios, along with the 20-bin rank histogram for Pauli $ZIII$ in count-32.

Physical-state diagnostics. Every posterior-center estimate is also checked for trace error, Hermitian error, minimum eigenvalue, root fidelity, and held-out Pauli observable error. These diagnostics are not substitutes for posterior calibration, because a valid density matrix can still represent uncertainty poorly. They ensure that the main comparison is made among physically meaningful density-matrix outputs and that observable prediction is not inferred from Bures distance alone.

D.1 Bayesian Reference Posterior and Sample-Cloud Comparison

Truth-centered metrics evaluate point reconstruction accuracy and marginal credible-ball coverage relative to the latent state $\rho_\star$, but they do not directly compare the generated sample cloud against the full Bayesian posterior for a fixed measurement record y. To support the posterior-to-posterior comparison in Table 3, we construct a deterministic Bayesian reference posterior and measure sample-cloud discrepancy in the Bures geometry.

Conditioning parity and record likelihood. The reference posterior uses the same prior p_{prior}, observation map, and finite measurement record $y = \{r_j\}_{j=1}^m$ provided to the learned samplers. For a shadow token, r_j specifies the observed product-Pauli basis B_j and bitstring outcome b_j, and its likelihood at a candidate state ρ is the single-shot Born probability $\mathrm{Tr}(M_{B_j,b_j}\rho)$. For a count token, r_j stores only the measurement basis B_j, the shot count n_j, the per-qubit mean-bit statistics b_j, and the all-plus count statistic c_j, rather than a full 16-outcome count vector. The reference posterior therefore evaluates the exact induced likelihood of these stored summary statistics by marginalizing the Born distribution over all unobserved full outcome-count vectors consistent with r_j. For a mixed record, the reference posterior conditions jointly on the concatenated shadow and count tokens supplied to the neural encoder.

Dense-grid construction and validation checks. We exploit the three-dimensional parameterization $\vartheta = (J, h, \beta)$ of the four-qubit transverse-field Ising family in Eq. (25) to build a deterministic grid reference posterior, avoiding the finite-chain and mixing errors of MCMC or SMC [14]. For each held-out record y, we evaluate the exact record likelihood on a $48 \times 48 \times 64$ grid containing $G = 147{,}456$ parameter points, multiply by the prior, normalize the resulting weights, and map the weighted parameter points to density matrices $\{\rho(\vartheta_g)\}_{g=1}^G \subset \mathcal{D}_d$. Refining the grid eightfold to $96 \times 96 \times 128$ points changes the reference posterior-mean Bures distance by 8.8×10^{-5} on average and by at most 6.6×10^{-4} across the tested scenarios. As a separate finite-sample check, the Bures-ground energy statistic between two independent sets of $R = 64$ samples drawn from the same reference posterior is at most 2.46×10^{-3}. These checks confirm that neither posterior-mean discretization error nor reference-sampling noise is the limiting factor in Table 3.

Bures-ground energy statistic. For each held-out record y, we compare the learned sample cloud $\mathcal{Q}_y = \{\rho^{(s)}\}_{s=1}^S$ against a reference sample cloud $\mathcal{P}_y = \{\tilde{\rho}^{(r)}\}_{r=1}^R$ using a two-sample energy statistic [28] with the unsquared Bures-Wasserstein distance d_{Bures} as its ground distance. Excluding within-cloud self-pairs, the record-level statistic is

$$\begin{aligned}\widehat{\mathcal{E}}_{\mathrm{Bures}}(\mathcal{Q}_y, \mathcal{P}_y) \triangleq \frac{2}{SR}\sum_{s=1}^{S}\sum_{r=1}^{R} d_{\mathrm{Bures}}\big(\rho^{(s)}, \tilde{\rho}^{(r)}\big) \\ - \frac{1}{S(S-1)}\sum_{s\neq s'} d_{\mathrm{Bures}}\big(\rho^{(s)}, \rho^{(s')}\big) \\ - \frac{1}{R(R-1)}\sum_{r\neq r'} d_{\mathrm{Bures}}\big(\tilde{\rho}^{(r)}, \tilde{\rho}^{(r')}\big).\end{aligned} \tag{39}$$

We average $\widehat{\mathcal{E}}_{\mathrm{Bures}}(\mathcal{Q}_y, \mathcal{P}_y)$ across the held-out evaluation records and report the mean $\pm$ standard deviation over five seeds in Table 3, where lower values indicate closer posterior sample clouds in the Bures geometry. We use $\widehat{\mathcal{E}}_{\mathrm{Bures}}$ as a comparative sample-discrepancy statistic.

E Main Finite-Record Evidence

The main result compares BuresTomFlow against posterior samplers that remove or replace one geometric ingredient at a time. Cholesky geometry keeps positive-definite coordinates. Vanilla

Riemannian Flow Matching keeps the manifold-solver framework without the Bures-specific path and tangent construction. Log-density and mirror/Frobenius controls use convenient positive-matrix coordinates with flat velocity losses. Table 5 presents the control summary. The point of this comparison is not physical validity alone: several controls already enforce valid density matrices. The remaining gap is therefore attributed to the Bures-specific path and tangent construction.

Role of the baselines. The baselines are chosen to isolate different assumptions. Cholesky geometry is a strong physical-coordinate control: it enforces positive definiteness but does not use Bures transport. Vanilla Riemannian Flow Matching tests whether generic manifold machinery is sufficient without the Bures-specific path and tangent loss. Log-density and mirror/Frobenius controls test convenient positive-matrix charts with flatter velocity supervision. Linear inversion is fast and useful as a reconstruction baseline, but it cannot represent posterior uncertainty. The interpretation of the table is therefore structural: each row keeps one part of the modeling problem fixed and removes one part of the Bures-geometric posterior construction.

Baseline assumptions and limitations. The Cholesky control assumes that a positive coordinate chart is enough to learn posterior density matrices. Its validity is strong, but the chart does not encode the fidelity geometry used by the target metric. The vanilla Riemannian Flow Matching control keeps the manifold-learning framework but replaces the Bures-specific construction, so it tests whether the gain comes from the particular density-matrix geometry rather than from using an ODE sampler. The log-density and mirror/Frobenius controls keep convenient positive-matrix coordinates and flat velocity losses. They are computationally natural, but Table 5 shows that, as a group, these controls have higher errors than BuresTomFlow on all three metrics. Linear inversion is intentionally separated from these samplers: it provides a fast reconstruction point, but no credible ball, rank distribution, or posterior sample cloud.

Reading the control table. The table should be read column by column. Bures distance measures posterior-center accuracy, coverage error measures departure from the nominal credible-ball coverage, and observable error measures downstream predictive usefulness on held-out Pauli observables. BuresTomFlow is the row that keeps the Bures path, Bures tangent loss, projected density-matrix output, and posterior sample cloud simultaneously, which is why the table is the central component comparison rather than a generic leaderboard.

Table 5: Geometry and coordinate controls on calibrated finite-record regimes. Lower is better for Bures distance, coverage error, and observable error. Each control removes or replaces one part of the full Bures-geometric posterior sampler while keeping the same multi-seed evaluation set.

Method / control	Component tested	Runs	Posterior quality ↓		
			Bures distance	Coverage error	Obs. error
BuresTomFlow	Bures path + tangent	40	0.3667 ± 0.0054	0.1000 ± 0.0000	0.0401 ± 0.0043
Cholesky geometry	Cholesky coordinates	40	0.6175 ± 0.0163	0.2030 ± 0.1612	0.2422 ± 0.0063
Vanilla Riemannian FM	Generic manifold flow	40	0.6638 ± 0.0352	0.8347 ± 0.1098	0.1822 ± 0.0209
Log-density chart controls	Log-density coordinates	80	0.8462 ± 0.0273	0.9000 ± 0.0000	0.2470 ± 0.0102
Linear inversion	Point estimate	40	0.4778 ± 0.1048	n/a	0.1001 ± 0.0468

E.1 Comparison with Classical Shadows and a Neural Point Estimator

Classical shadows. We compare the posterior mean of BuresTomFlow with the standard local-Pauli classical-shadow estimator of Eq. (27) [17], computed from exactly the same randomized Pauli snapshots that condition the sampler. The comparison uses five seeds, 32 held-out states per snapshot budget, and the four held-out Pauli observables of Appendix D. These results come from a separate shadow-only comparison rather than the eight-regime aggregate of Table 1. Table 6 reports the mean absolute held-out observable error. Under the tested low-shot budgets, the posterior mean has lower observable error than the classical-shadow estimator. Classical shadows are prior-free estimators designed for observable prediction, whereas BuresTomFlow learns the thermal-state prior and returns posterior samples over physical density matrices.

Table 6: Held-out observable error (↓) of the classical-shadow estimator and the BuresTomFlow posterior mean on identical shadow records. Values are means over five seeds with 32 held-out states per budget.

Estimator	4 snapshots	16 snapshots	32 snapshots
Classical shadows	1.060	0.652	0.503
BuresTomFlow posterior mean	**0.164**	**0.122**	**0.073**

Neural point estimator. We also train an amortized neural density-matrix point estimator on count-32, mixed-4, and shadow-16. It shares the state prior, measurement simulator, record tokenization, records, and 32 held-out states per regime with BuresTomFlow. The estimator uses the same two-block, four-head record encoder followed by a Cholesky density-matrix head, and is trained for 2,000 steps with batch size 32 using Adam at learning rate 10^{-3} over three random seeds. Table 7 reports the posterior-center Bures distance. BuresTomFlow is lower in two of the three tested regimes and, unlike the point estimator, also returns posterior samples.

Table 7: Posterior-center Bures distance (↓) of an amortized neural point estimator and BuresTomFlow on identical records. Values are mean ± standard deviation over three seeds.

Method	count-32	mixed-4	shadow-16
Neural point estimator	0.239 ± 0.130	0.252 ± 0.120	0.289 ± 0.085
BuresTomFlow	0.219 ± 0.022	0.241 ± 0.030	0.321 ± 0.002

F Additional Regime Tables

The main figure compresses two views of the measurement-regime evidence: a family-by-budget sweep and a calibrated finite-record diagnostic with seed-level variation. The tables below give the same evidence more room. Table 8 reports detailed finite-record settings in which the Bures sampler has a clear margin against the strongest non-Bures comparator. Table 9 then reports the broader family-budget grid in the same row-wise comparison format. The two tables serve different purposes: the first gives detailed finite-record cases, while the second shows breadth across measurement families and shot budgets.

Row construction. Each row fixes a measurement family, shot budget, and seed set. For that row, the comparison first identifies the strongest non-Bures posterior sampler under the row-wise Bures-distance metric, then reports relative reductions in Bures distance, coverage error, and observable error. For a lower-is-better metric E, the row-wise reduction is computed as

$$\Delta_E = \frac{E_{\text{best non-Bures}} - E_{\text{BuresTomFlow}}}{E_{\text{best non-Bures}}}. \tag{40}$$

Runtime cost is reported as the corresponding learned-sampler runtime ratio. This organization keeps the finite-record evidence tied to measurement regimes rather than averaging away the distinction between count, mixed, and randomized shadow records.

F.1 Detailed Finite-Record Rows

Table 8 identifies regimes where the Bures-specific path and tangent construction produce the clearest posterior-quality gains. These rows come from the calibrated diagnostic used for seed-level comparison. Together with the expanded family-budget grid, they show both the magnitude and breadth of the finite-record advantage.

F.2 Expanded Family-Budget Grid

Table 9 reports the broader measurement grid across count, mixed, and shadow records. This view is deliberately organized by measurement family and shot budget, so that the reader can see whether the Bures-leading comparison is concentrated in a single corner of the design space or persists across

Table 8: Detailed finite-record regimes under the calibrated diagnostic, denoted by `(diag.)` in the regime labels. Best non-Bures is the strongest comparator in the same row that does not use the Bures-specific path construction. Improvements and runtime cost ratios are reported as mean ± standard deviation across seeds.

Regime	Best non-Bures	Bures-leading seeds	Relative improvement ↑			Cost ratio
			Bures gain	Coverage red.	Observable red.	Runtime
shadow-128 (diag.)	Cholesky	5/5	0.4110 ± 0.0119	0.1450 ± 0.1401	0.8384 ± 0.0136	1.0627 ± 0.1175
shadow-16 (diag.)	Cholesky	5/5	0.4120 ± 0.0107	0.1425 ± 0.1509	0.8317 ± 0.0159	0.8806 ± 0.0712
count-32 (diag.)	Cholesky	5/5	0.3978 ± 0.0207	0.0850 ± 0.2487	0.8322 ± 0.0226	1.0490 ± 0.1728
mixed-32 (diag.)	Cholesky	5/5	0.4102 ± 0.0145	0.1487 ± 0.2166	0.8329 ± 0.0228	1.0738 ± 0.0828
mixed-4 (diag.)	Cholesky	5/5	0.4056 ± 0.0136	0.1163 ± 0.1758	0.8323 ± 0.0217	0.9972 ± 0.1239
mixed-64 (diag.)	Cholesky	5/5	0.4049 ± 0.0146	0.0925 ± 0.1701	0.8388 ± 0.0143	1.0340 ± 0.0771
count-8 (diag.)	Cholesky	5/5	0.3994 ± 0.0103	0.0650 ± 0.1190	0.8335 ± 0.0197	0.9932 ± 0.0919

multiple record types. Because this table is aggregated from the broader family-budget sweep, rows carry the `(sweep)` qualifier to distinguish them from the `(diag.)` calibrated-diagnostic slice in Table 8.

Table 9: Expanded measurement-regime grid from the broader family-budget sweep, denoted by `(sweep)` in the regime labels. Bures-leading seeds count seed-level cases where the Bures-specific sampler improves the row-wise comparison against the best non-Bures comparator. Best non-Bures denotes the strongest non-Bures comparator in that row. Improvements and runtime cost ratios are reported as mean ± standard deviation over seeds.

Regime	Best non-Bures	Bures-leading seeds	Relative improvement ↑			Cost ratio
			Bures gain	Coverage red.	Observable red.	Runtime
count-4 (sweep)	Cholesky	5/5	0.274 ± 0.013	0.333 ± 0.212	0.801 ± 0.040	0.88 ± 0.02
count-8 (sweep)	Cholesky	5/5	0.282 ± 0.009	0.357 ± 0.149	0.801 ± 0.032	0.84 ± 0.04
count-16 (sweep)	Cholesky	5/5	0.275 ± 0.028	0.261 ± 0.202	0.760 ± 0.057	1.04 ± 0.24
count-32 (sweep)	Cholesky	5/5	0.286 ± 0.019	0.373 ± 0.178	0.799 ± 0.029	1.01 ± 0.03
count-64 (sweep)	Cholesky	5/5	0.276 ± 0.035	0.298 ± 0.348	0.790 ± 0.022	1.00 ± 0.06
count-128 (sweep)	Cholesky	5/5	0.276 ± 0.035	0.303 ± 0.372	0.787 ± 0.021	1.07 ± 0.07
count-256 (sweep)	Cholesky	5/5	0.260 ± 0.041	0.232 ± 0.257	0.794 ± 0.023	1.00 ± 0.03
mixed-4 (sweep)	Cholesky	5/5	0.285 ± 0.010	0.393 ± 0.135	0.797 ± 0.031	0.89 ± 0.03
mixed-8 (sweep)	Cholesky	5/5	0.282 ± 0.019	0.375 ± 0.102	0.808 ± 0.038	0.88 ± 0.04
mixed-16 (sweep)	Cholesky	5/5	0.284 ± 0.025	0.286 ± 0.142	0.791 ± 0.036	0.98 ± 0.03
mixed-32 (sweep)	Cholesky	5/5	0.288 ± 0.026	0.359 ± 0.129	0.811 ± 0.023	0.99 ± 0.04
mixed-64 (sweep)	Cholesky	5/5	0.280 ± 0.013	0.363 ± 0.140	0.811 ± 0.024	1.04 ± 0.03
mixed-128 (sweep)	Cholesky	5/5	0.282 ± 0.041	0.240 ± 0.352	0.811 ± 0.023	1.00 ± 0.04
shadow-4 (sweep)	Cholesky	5/5	0.282 ± 0.025	0.338 ± 0.139	0.801 ± 0.021	0.82 ± 0.10
shadow-8 (sweep)	Cholesky	5/5	0.276 ± 0.048	0.186 ± 0.287	0.788 ± 0.043	0.91 ± 0.16
shadow-16 (sweep)	Cholesky	5/5	0.287 ± 0.009	0.398 ± 0.121	0.787 ± 0.038	0.83 ± 0.05
shadow-32 (sweep)	Cholesky	5/5	0.284 ± 0.026	0.317 ± 0.164	0.779 ± 0.046	0.99 ± 0.03
shadow-64 (sweep)	Cholesky	5/5	0.303 ± 0.027	0.323 ± 0.162	0.797 ± 0.031	1.01 ± 0.03
shadow-128 (sweep)	Cholesky	5/5	0.302 ± 0.023	0.348 ± 0.145	0.801 ± 0.029	1.00 ± 0.06
shadow-256 (sweep)	Cholesky	5/5	0.279 ± 0.036	0.257 ± 0.270	0.809 ± 0.020	1.12 ± 0.03

G Supplementary Runtime, Calibration, and Scaling Figures

G.1 Composite Regime Score

Figure 3 asks whether Bures-distance gains, coverage-error reductions, and observable-error reductions appear in the same measurement cells. The figure is computed from the calibrated finite-record summary, using the same row-wise Bures-vs-best-non-Bures comparison as the additional regime tables. It gives the visual analogue of Tables 8 and 9: the numerical tables show the individual metrics, while the heatmap shows where the three reductions co-occur.

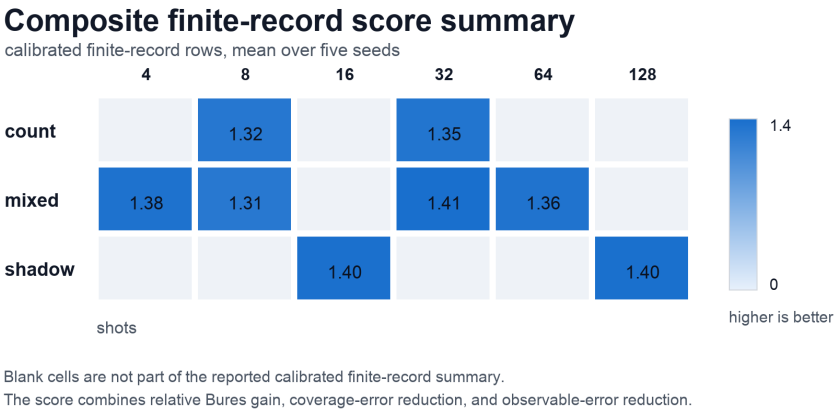

Figure 3: Composite measurement-regime summary. Color encodes a combined finite-record score using relative Bures-distance gain, coverage-error reduction, and held-out observable-error reduction. The displayed cells are the calibrated finite-record rows entering this summary.

G.2 Runtime and Tradeoff Views

Figure 4 combines the two runtime diagnostics. The left panel asks how amortized inference cost changes as more posterior queries are made. The right panel asks how dense posterior quality and total runtime trade off across geometry choices. Together, the panels show that the learned samplers amortize training cost over repeated posterior queries, while the Bures-geometric sampler occupies a lower-distance region among the implemented geometry controls.

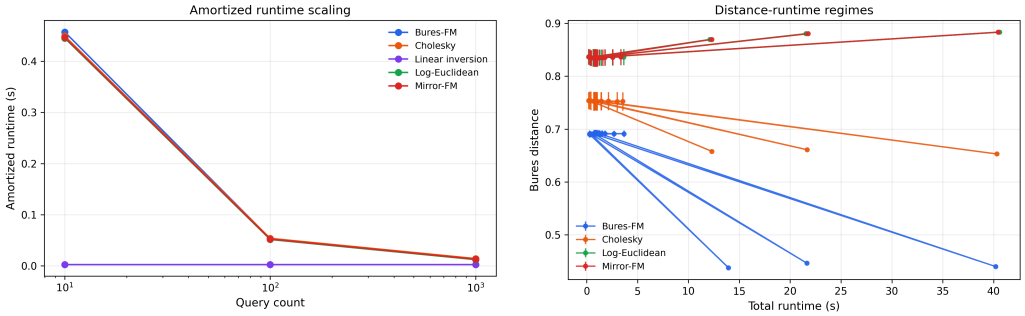

Figure 4: Supplementary runtime diagnostics. Left: amortized runtime per query as posterior-query count increases. Learned posterior samplers amortize the one-time training cost, while linear inversion remains a point-estimation reference. Right: Bures distance versus total runtime for the implemented geometry and coordinate controls.

G.3 Credible-Ball Coverage

Figure 5 reports credible-ball coverage error. The BuresTomFlow value corresponds to empirical coverage of 1.0000 at nominal level 0.9, so it represents systematic overcoverage. Linear inversion remains outside this comparison because it is a point estimator and does not generate a posterior cloud.

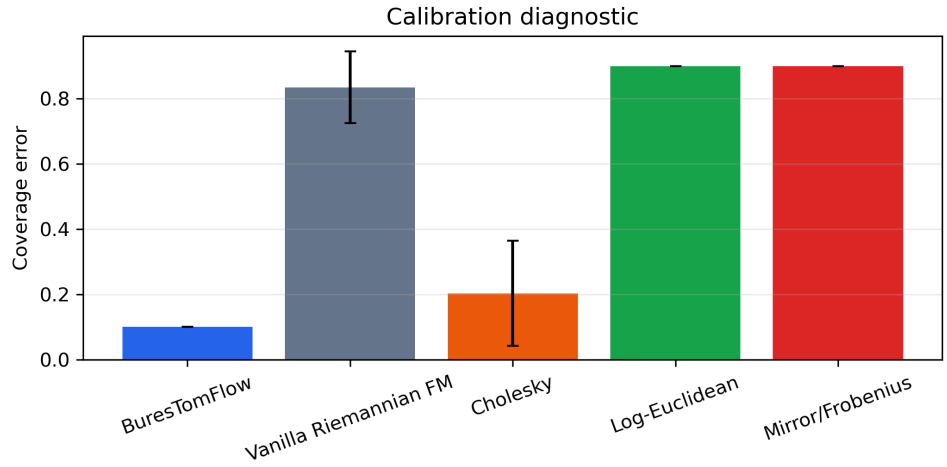

Figure 5: Coverage-error comparison across posterior-sampler baselines. Lower coverage error indicates closer agreement between nominal credible-ball level and empirical coverage.

G.4 Regime-Level Credible-Ball Coverage Under the Expanded Protocol

Table 10 reports regime-level empirical coverage probabilities for BuresTomFlow under the expanded 2,000-update, batch-size-32 protocol across seven measurement regimes and five seeds. Unlike the split-conformal regions in Table 2, which calibrate Bures balls directly from center-to-truth distances, this diagnostic follows the sample-derived credible-ball protocol of Eq. (38). Each credible ball is formed from the spread of generated posterior samples around the posterior center, and a multiplicative radius scale factor between 0.8 and 1.2 is selected on a disjoint calibration split before evaluation on 96 independent held-out test states.

Table 10: Regime-level empirical coverage probabilities for BuresTomFlow sample-derived credible balls at nominal level 0.9 across the seven expanded-protocol regimes. Columns report unadjusted held-out coverage, coverage after applying the calibration-split radius scale factor, and the mean selected scale factor across five seeds. Coverage values are mean $\pm$ standard deviation.

Regime	Raw coverage	Adjusted coverage	Mean radius scale
count-32	0.692 ± 0.089	0.896 ± 0.058	1.14
mixed-32	0.679 ± 0.138	0.888 ± 0.064	1.14
count-8	0.646 ± 0.153	0.848 ± 0.070	1.16
shadow-32	0.675 ± 0.075	0.840 ± 0.050	1.20
mixed-4	0.577 ± 0.096	0.825 ± 0.058	1.18
shadow-16	0.525 ± 0.068	0.698 ± 0.063	1.20
shadow-4	0.454 ± 0.072	0.542 ± 0.063	1.20

Across the seven regimes, calibration-split rescaling increases mean held-out coverage from 0.607 before adjustment to 0.791 after adjustment. Empirical coverage improves in every regime and reaches near-nominal levels in the most informative count-32 and mixed-32 settings, whereas sparser shadow regimes remain below the 0.9 target and the selected scale factor reaches the upper grid boundary of 1.20.

G.5 Simulation-Based Calibration

We evaluate BuresTomFlow under standard prior-predictive simulation-based calibration [29] across 200 tasks in each of the count-8 and count-32 scenarios using 63 posterior samples per task, which produces integer ranks from 0 to 63 across six fixed scalar functionals (four Pauli expectations, purity, and the largest eigenvalue). Across the evaluated functionals in both scenarios, the observed BuresTomFlow rank distributions deviate significantly from the discrete uniformity expected under the tested prior-predictive SBC protocol. Table 11 reports representative rank summary statistics for Pauli $ZIII$ and purity in the count-8 and count-32 scenarios, and Figure 6 displays the 20-bin rank histogram for Pauli $ZIII$ in the count-32 scenario.

Table 11: Representative simulation-based calibration results for BuresTomFlow across 200 prior-predictive tasks in each of the count-8 and count-32 scenarios using 63 posterior samples per task (ranks in $\{0, \ldots, 63\}$). Columns report the empirical rank mean $\pm$ standard deviation, the χ^2 p-value, and the 20-bin expected calibration error (ECE).

Scenario	Functional	Rank mean $\pm$ std.	χ^2 p-value	20-bin ECE
count-8	Pauli $ZIII$	28.21 ± 5.26	7.85×10^{-88}	0.0633
count-8	Purity	44.05 ± 25.01	3.27×10^{-200}	0.0559
count-32	Pauli $ZIII$	28.34 ± 4.25	3.70×10^{-125}	0.0685
count-32	Purity	44.21 ± 24.85	6.88×10^{-228}	0.0561

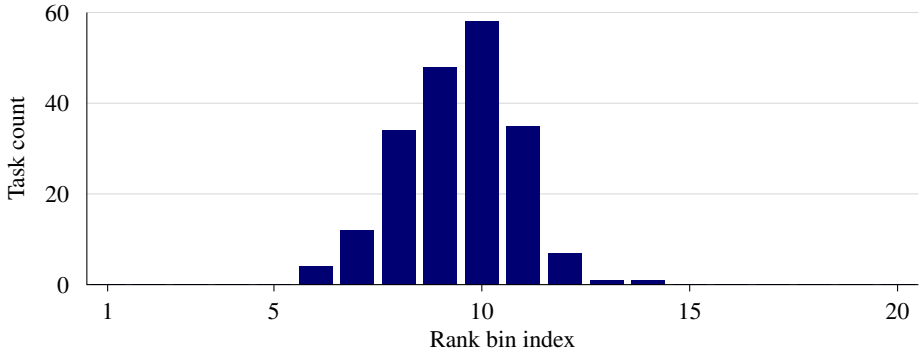

Figure 6: Observed 20-bin simulation-based calibration rank histogram for BuresTomFlow on the Pauli $ZIII$ functional in the count-32 scenario across 200 prior-predictive tasks with 63 posterior samples per task, plotted against bin indices 1 to 20. In this distribution, 175 of the 200 ranks fall into four adjacent bins (bins 8 to 11), illustrating the substantial departure from uniformity reported in Table 11.

G.6 Bures-Kernel Scaling

Figure 7 asks a narrower supporting question: whether Bures kernel operations and a spectral low-rank proxy remain computationally tractable in larger matrix dimensions. The figure comes from the standalone Bures-kernel scaling diagnostic and compares exact dense operations with a low-rank proxy in runtime and memory. It supports the feasibility of the geometric component beyond the dense four-qubit training benchmark.

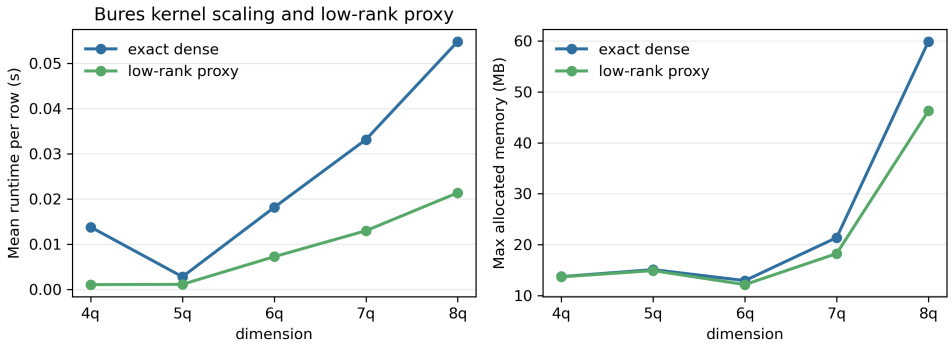

Figure 7: Bures-kernel scaling and spectral low-rank proxy study. Exact dense Bures operations complete up to the 8-qubit kernel-operation diagnostic shown here, while the low-rank proxy reduces runtime and memory in this diagnostic. The study concerns the geometry kernel used by the sampler.

H Scope and Limitations

Bures-geometric posterior sampling improves posterior estimation in the tested dense finite-record tomography regimes and yields lower mean credible-ball coverage error than the corresponding geometry baselines. The experiments use exact dense four-qubit density matrices, finite count/mixed/shadow records, and posterior summaries formed from a finite number of learned samples. Within this scope, the method is evaluated against geometry and coordinate controls that isolate the role of the Bures path and tangent loss.

The present dense implementation keeps the full 16×16 density matrix explicit. This choice makes the geometry and posterior metrics directly measurable: trace, positivity, Bures distance, held-out observables, credible-ball coverage, and simulation-based calibration can all be evaluated without replacing the density matrix by a compressed ansatz. The same choice also fixes the computational scope. Larger Hilbert spaces require either structured state families, low-rank parameterizations, tensor-network structure, or additional approximations to the Bures kernel. Figure 7 studies one such geometric kernel proxy, but the main posterior-tomography claim remains the dense finite-record result reported in the tables.

The measurement model is likewise controlled. The records come from ideal product-Pauli Born sampling, with finite-shot ambiguity but without device noise, drift, or adaptive measurement design. This is the right setting for isolating whether Bures geometry improves posterior sampling under finite records. Extending the sampler to noisy measurement channels, adaptive protocols, and experimentally collected records is a separate evaluation problem. The appendix is organized to make this separation explicit: protocol and geometry first, metrics and calibration second, then evidence tables and computational diagnostics.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the finite-record posterior tomography setting, the Bures-geometric Flow Matching construction, and the dense four-qubit experimental scope. The scope and limitations are further stated in Appendix H.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses the dense four-qubit setting, ideal product-Pauli measurement model, and computational scaling in Appendix H. These statements delimit the empirical claims without weakening the reported Bures-path-flow advantage in the evaluated regimes.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not claim new formal theorems. Appendix B states the known Bures-Wasserstein construction and cites the relevant geometry references.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Appendices A, C, and D specify the simulator, measurement records, model components, optimizer, posterior sampling, metrics, and calibration protocol needed to reproduce the main comparisons.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in

some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The paper does not yet provide an open code or data archive; the authors will release the code publicly. The experimental data are synthetic and can be regenerated from the simulator, measurement protocol, and training details in Appendices A, C, and D.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Appendices A, C, D, and F specify the data generation, training details, model hyperparameters, posterior sampling settings, evaluation splits, and measurement regimes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The main and supplementary result tables report variability across repeated runs, and Appendix D defines the reporting and calibration quantities used to interpret those comparisons.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The paper reports runtime evidence, but it does not fully enumerate the hardware, memory, and total compute used for every experiment. The compute environment was heterogeneous, so a complete per-run resource accounting is not included in the current manuscript.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The study uses synthetic quantum-tomography simulations and does not involve human subjects, private data, deployed decision systems, or other settings that would require a deviation from the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The work is foundational methodology for synthetic quantum-state tomography and is not tied to a deployed application or a direct societal-impact pathway in the paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release a high-risk model, scraped dataset, or dual-use asset requiring release safeguards.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The implementation uses the public `flow_matching` package [22]; its repository states that the code is distributed under the CC BY-NC 4.0 license. The experiments use synthetic data generated by the stated protocol rather than an external dataset.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new dataset, benchmark asset, model checkpoint, or software package as part of the submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects, so IRB approval or an equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not an important, original, or non-standard component of the core method or experiments in this research.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.