

Toward Reliable Hybrid Quantum-Classical Biomedical AI:

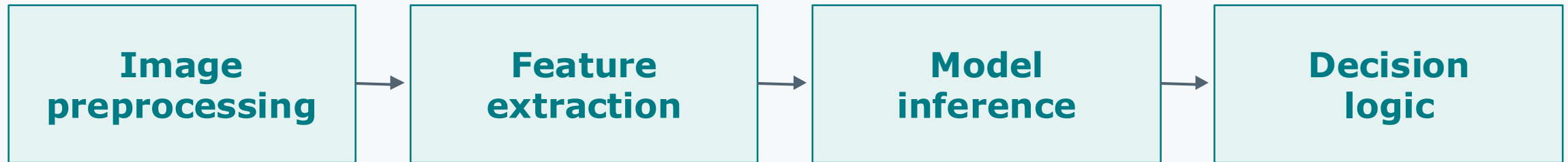
Programming and Verification Challenges
from an Engineering Perspective

Er-Xuan Jian
Electrical Engineering Research Institute
National Taiwan University

Preliminary research direction • 25-minute talk + 5-minute discussion

Engineering motivation

Correct components still depend on compatible interfaces.



CPU / GPU / FPGA

Experience: FPGA convolution for dental imaging
Multimodal chest X-ray reporting and embedded AI systems

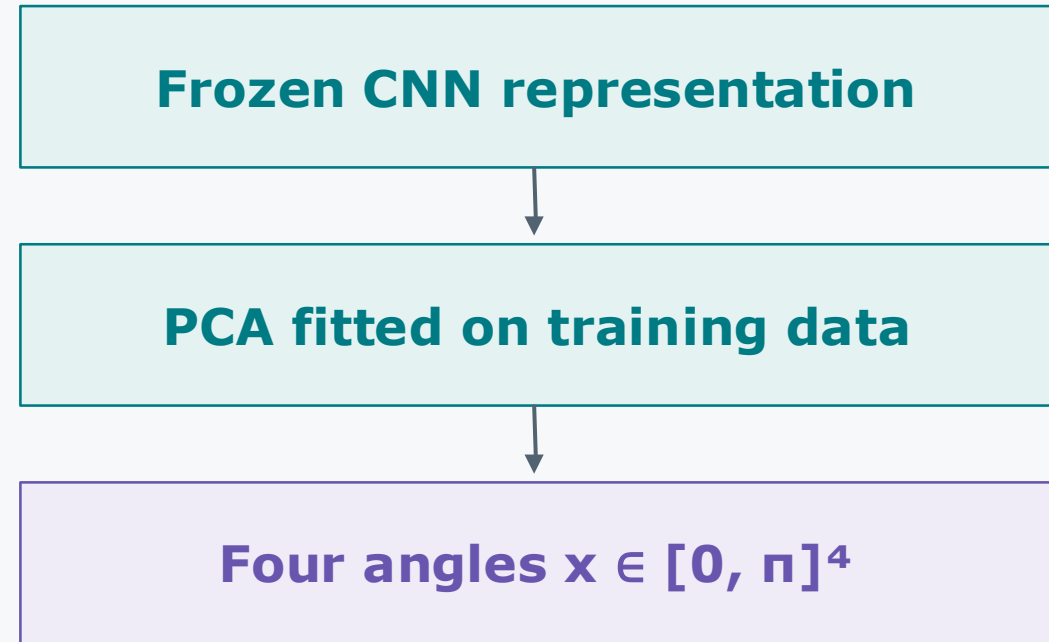
What changes when an interface returns quantum measurements?

A biomedical toy task

Proposed case study: PneumoniaMNIST dataset-label classification

**Chest X-ray
normal / pneumonia
labels**

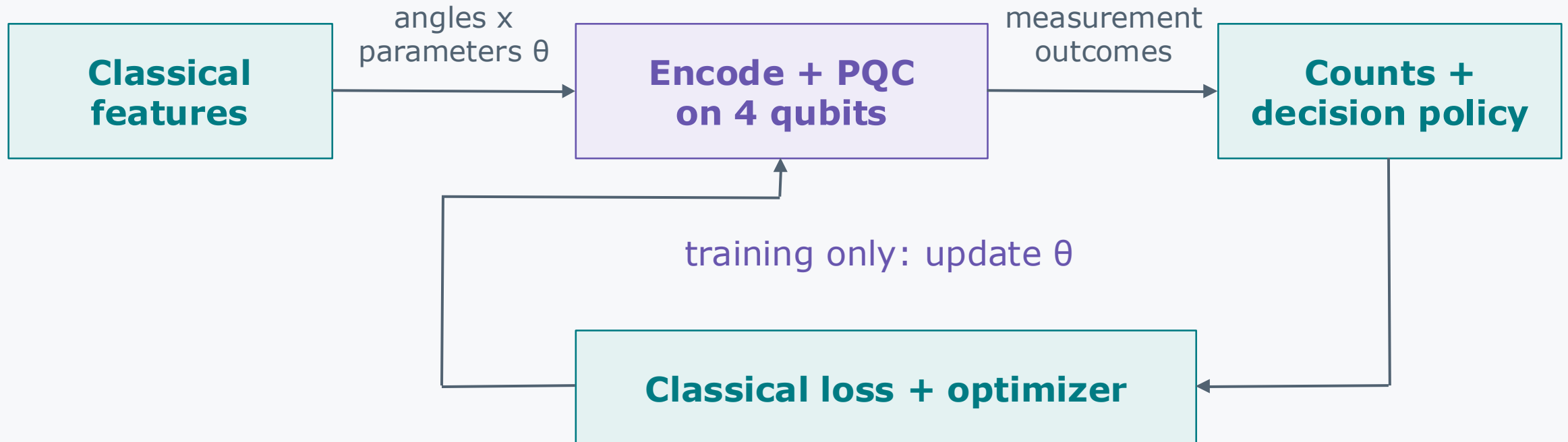
Research benchmark
No clinical-use claim



Four coordinates make analysis tractable; information loss must be measured.

The hybrid architecture

Inference uses fixed parameters. Training adds a classical feedback loop.



Initial backend: simulator. FPGA and QPU integration remain future work.

Why this variational classifier?

VQC: variational quantum classifier. PQC: parameterized quantum circuit.

- 01** Angle encoding gives a direct feature-to-gate contract.
- 02** A shallow PQC exposes the exact operations and readout.
- 03** Four qubits allow a tractable classical reference.

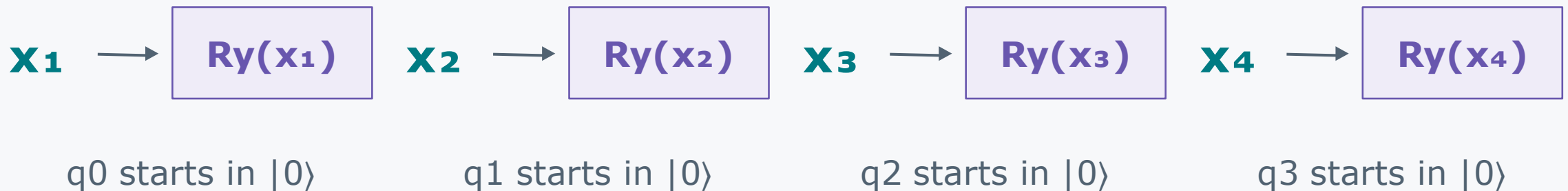
Alternatives: quantum kernels change the learning interface.
Amplitude encoding adds state-preparation obligations.

No assumption of quantum advantage or superior clinical performance.

Encoding makes the interface mathematical

Each classical feature sets a rotation angle.

$$Ry(x_i) |0\rangle = \cos(x_i/2) |0\rangle + \sin(x_i/2) |1\rangle$$



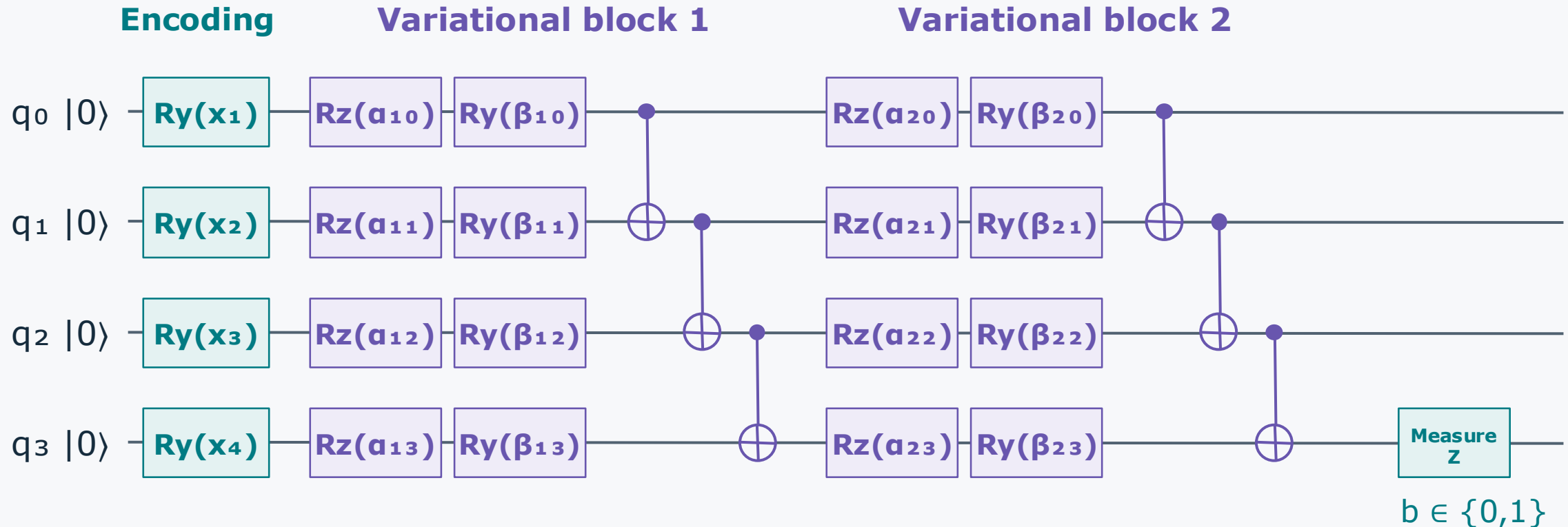
Superposition describes amplitudes.

One shot yields a sampled outcome, not the full state.

Encoding contract: valid angles prepare the specified joint state.

A concrete four-qubit circuit

Proposed logical circuit • 16 trainable angles • 6 CNOTs



Reinitialize and rerun the whole circuit for every shot.

Measurement becomes a classical score

Read q_3 in the computational basis.

$$p_{\theta}(x) = \Pr(b = 1) = (1 - \langle Z_3 \rangle) / 2$$

**N repeated
executions**

count ones / N

Illustration: 52 ones in 100 shots
Estimated score = 0.52

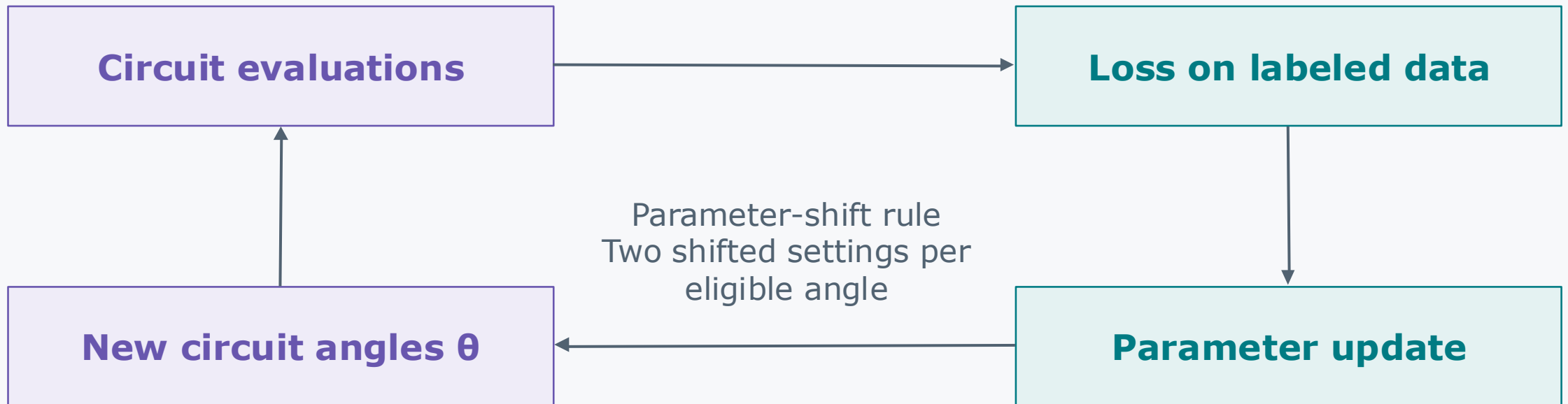
Sampling uncertainty
changes the estimated score.

Clinical calibration
requires separate evidence.

A model score is not automatically a calibrated disease probability.

The classical optimization loop

Training estimates scores and gradients; inference freezes θ .



First verification target: the fixed-parameter inference program.

Three distinct questions

Each claim needs its own evidence.

Claim	Question	Example evidence
Predictive validity	Does it predict labels well?	Held-out AUC and calibration
Program correctness	Does it satisfy its contract?	Proof of encoding and decision rules
System reliability	Do assumptions hold in operation?	Noise, drift and timeout analysis

A correct program can faithfully execute an inaccurate classifier.

The workflow as a program

Fixed x , θ and N define the inference run.

```
validate image and configuration
x := frozen_feature_pipeline(image)
for shot = 1 ... N:
    reset; encode(x); apply_PQC( $\theta$ )
    b[shot] := measure(q3)
score := mean(b)
interval := bounded_estimate(score)
```

**Interval above τ
return 1**

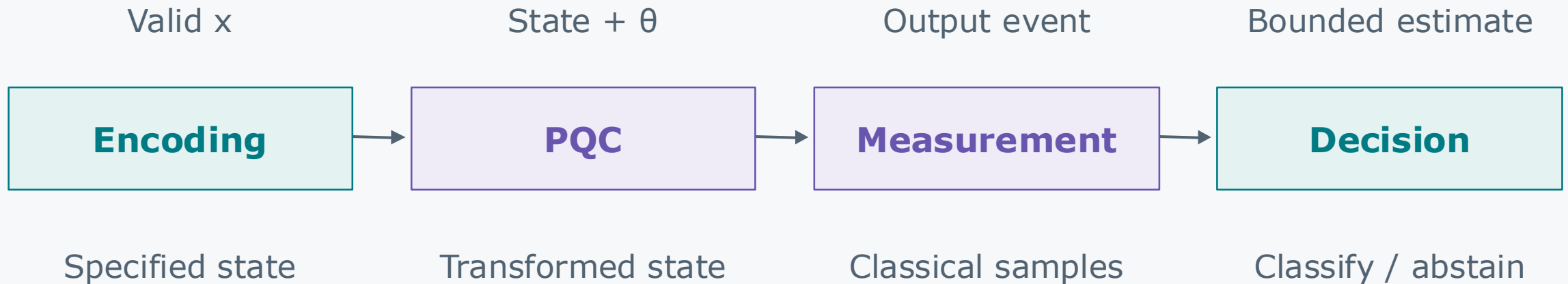
**Interval below τ
return 0**

**Otherwise
abstain**

Semantics must include classical branches and probabilistic outcomes.

Contracts across the boundary

Hoare-style contract $\{P\} C \{Q\}$: assumptions, program, obligations.



$$wp(U, Q) = U^\dagger Q U$$

For an effect Q : $0 \leq Q \leq I$
Its value on ρ is $\text{tr}(Q\rho)$.

Ying 2011 • Feng & Ying 2021 • Ying 2026

A conditional reliability bound

Separate sampling error from implementation bias.

$\hat{p}$: sampled score p_{dev} : device probability p_{ref} : reference probability

$$\Pr(|\hat{p} - p_{\text{ref}}| \leq r) \geq 1 - \delta$$

$$r = \sqrt{[\ln(2/\delta) / (2N)]} + b$$

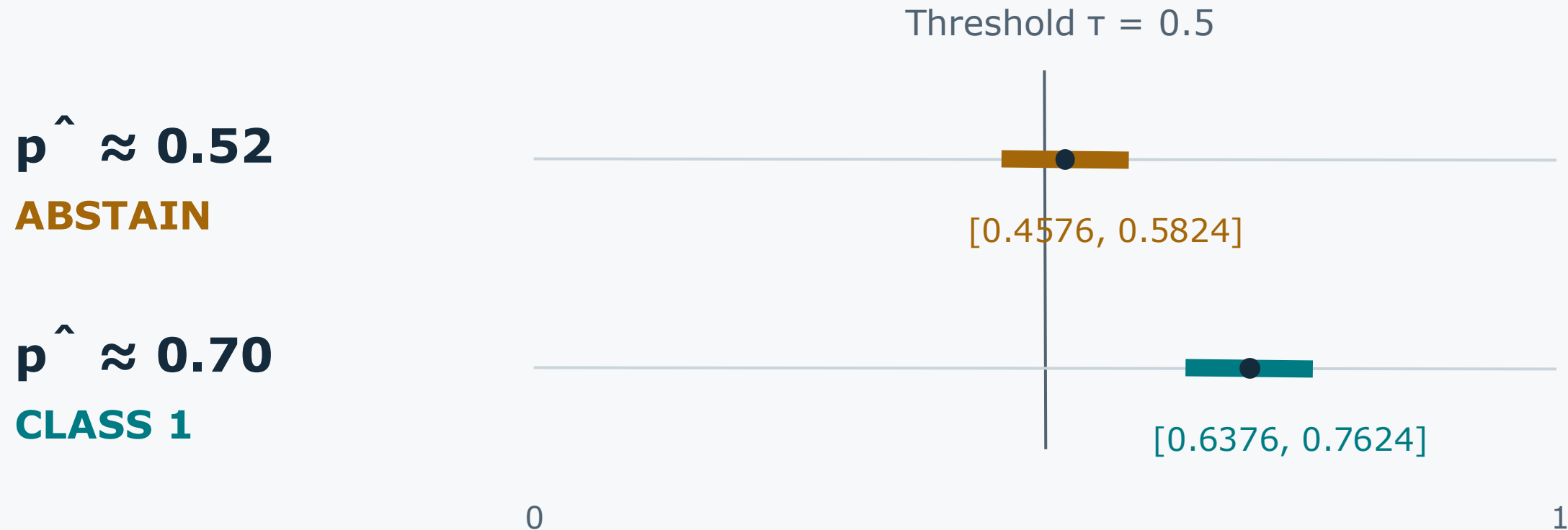
Sampling assumption
Fixed N , independent stationary shots

Implementation assumption
 $|p_{\text{dev}} - p_{\text{ref}}| \leq b$, with justification

More shots reduce sampling error; they do not remove systematic bias.

A decision policy with abstention

Rounded illustration: $N = 1024$, $\delta = 0.05$, assumed $b = 0.02$; $r \approx 0.0624$



The guarantee concerns agreement with the reference decision.

Reliability failure map

Different failure sources require different evidence.

Classical front end

Feature order / scaling
Quantization error

Interface checks
Numeric-error analysis

Quantum execution

Mapping / gate order
Noise and drift

Semantic checks
Noise-model validation

Estimation and use

Finite shots / threshold
Medical distribution shift

Valid intervals + coverage
Independent model
evaluation

A shallow circuit still needs explicit noise, sampling and resource assumptions.

Evaluating the quantum component

Fair comparison includes execution cost and verification effort.

Predictive value

Same four-feature inputs
Logistic regression, SVM, MLP
Fuller-feature classical baseline
No-entanglement ablation

Reliability and cost

Reference-decision disagreement
Abstention coverage
Shots, circuit calls and latency
Ideal, noisy and hardware runs

Tests and simulations inform validation; universal proof needs formal reasoning.

Research questions and roadmap

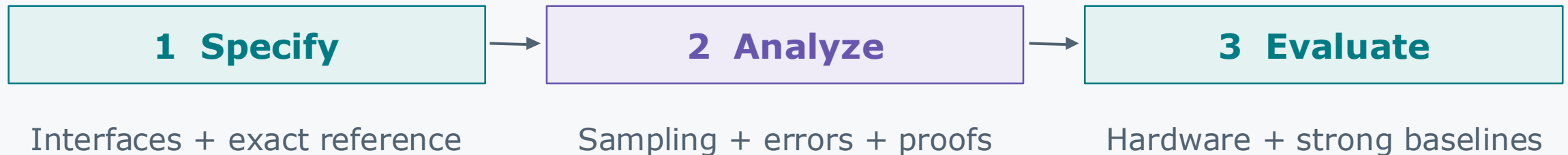
Proposed work, with milestones that produce checkable evidence

RQ1 What end-to-end contracts should the program satisfy?

RQ2 How can sampling and execution errors be composed?

RQ3 How can existing logics support modular reasoning?

RQ4 When does the quantum component add measurable value?



A feasible starting point

A small VQC exposes the interfaces we need to specify.

Specify the whole hybrid program.
State the assumptions behind every guarantee.
Evaluate whether the quantum component adds value.

Open for discussion

Which contract should be formalized first?