# The Binary–Quantum Bridge for AGI: Information-Preserving Differentiable Programs with a Causal Objective*

Peter De Ceuster

October 31, 2025

*Addressed to Suriya Gunasekar, Microsoft.*

## Preface: Scope, Intent, and Archival Use

The *Binary–Quantum Bridge* (BQB) is a *mathematical framework* for analyzing a dual objective in AGI design: maintain differentiable, information-preserving learning dynamics while guaranteeing classical, auditable behavior at decision commit. It is not an engineering blueprint. We recommend archiving this work and deferring any real-world development until (i) interpretability benchmarks, (ii) incident response and containment, and (iii) externally auditable objective governance are established and independently verified.

**Remark 1** (What "quantum" means here)**.** *Hilbert spaces, unitaries, and superpositions are used as a* representational tool *for differentiable exploration over programs. No assumption about quantum hardware is required. The term "binary" refers to classical, reversible behavior at commit time.*

**Abstract**

We formalize a bridge between continuous learning and discrete execution. A unitary *executor* $U$ evolves working registers; a classical *program* $C$ selects behavior; a causal world model $M$ evaluates interventional likelihoods. The bridge objective

$$\min_{U,\,C,\,M} \mathcal{F}_\beta(U,C,M) \quad \text{s.t.} \quad U^\dagger U = \mathbb{1},\ \|\|_\diamond \left(\mathcal{U}_C - \Pi_C\right) \le \varepsilon$$

uses

$$\mathcal{F}_\beta(U,C,M) = \underbrace{L(C)}_{\text{program length}} + \underbrace{L(M)}_{\text{model length}} + \underbrace{\mathbb{E}[-\log p_M(o \,|\, \mathrm{do}(\pi_C), h)]}_{\text{causal fit}} - \beta\, \underbrace{J(\pi_C)}_{\text{return}},$$

balancing compression, causal adequacy, and control. The diamond-norm *binary boundary* enforces classical exactness at commit while allowing differentiable superpositions for credit assignment during learning. We define *sectors* (behaviorally equivalent programs) and *gates* (evidence-triggered transitions), prove basis-level correctness from the boundary, propose a safe self-rewrite rule, and separate ideal theory from practical surrogates. Claims of gradient advantage are explicitly framed as conjectures.

*Theory paper for archival study. Implementation requires approximations and independent governance verification; premature deployment is discouraged.

**Deployment gate.** Any future system inspired by this framework should only be exposed to open environments if independently verified governance readiness $\mathsf{Gov}$ exceeds capability $\mathsf{Cap}$: $\mathsf{Gov} \geq \alpha\,\mathsf{Cap}$ for some $\alpha > 1$.

# 1 Core Concepts: Duality, Sectors, and Gates

## 1.1 Binary–Quantum duality

- **Learning ("quantum-like"):** $U$ supports differentiable exploration over candidate behaviors via superpositions; gradients propagate through interference.
- **Execution ("binary-like"):** At commit, behavior must match a classical reversible program $\Pi_C$ up to boundary error $\varepsilon$ for auditability and safety.

## 1.2 Sectors and gates

**Definition 1** (Behavioral sector)**.** *Fix an observable $O$ on $\mathcal{H}_{\text{bits}} \otimes \mathcal{H}_{\text{env}}$ and tolerance $\delta \geq 0$. The* $\delta$-sector *of program $C$ is*

$$\mathcal{S}_\delta(C) = \left\{ C' : \sup_\rho \; d_{\text{TV}}\big(\text{Diag Tr}_{\neg\mathcal{H}_{\text{bits}}}\, \mathcal{U}_{C'}(\rho),\ \text{Diag Tr}_{\neg\mathcal{H}_{\text{bits}}}\, \mathcal{U}_C(\rho)\big) \leq \delta \right\}.$$

*A* gate *is a transformation $C \to C'$ with $C' \notin \mathcal{S}_\delta(C)$.*

**Remark 2** (Learning as sector search)**.** *Learning comprises (i) finding a high-performing sector (macro strategy) and (ii) selecting the shortest invariant-preserving code within that sector (micro implementation). Superpositions provide smooth exploration across nearby sectors; the boundary snaps to classical correctness at commit.*

**Remark 3** (Total variation (TV) at commit and the $\varepsilon/2$ factor)**.** *For classical distributions $p, q$, the total variation distance is*

$$\text{TV}(p,q) = \tfrac{1}{2}\|p - q\|_1 = \tfrac{1}{2}\sum_x |p(x) - q(x)|.$$

*If the binary boundary holds, $\|\mathcal{U}_C - \Pi_C\|_\diamond \leq \varepsilon$, then after measurement (commit) the induced classical outputs differ by at most $\varepsilon/2$ in TV because measurement contracts trace distance and* $\text{TV} = \frac{1}{2}\|\cdot\|_1$ *on classical laws.*

# 2 Mathematical Setup

## 2.1 Spaces and reduced channels

Let $\mathcal{H}_{\text{bits}} = (\mathbb{C}^2)^{\otimes n}$ (working memory & action), $\mathcal{H}_{\text{code}}$ spanned by orthonormal $\{|C\rangle\}$ (finite, prefix-free programs), $\mathcal{H}_{\text{env}}$ (environment), and optionally $\mathcal{H}_{\text{mem}}$ (scratch). Write $\mathcal{H} = \mathcal{H}_{\text{bits}} \otimes \mathcal{H}_{\text{code}} \otimes \mathcal{H}_{\text{env}} (\otimes \mathcal{H}_{\text{mem}})$. For unitary $U : \mathcal{H} \to \mathcal{H}$ and fixed $|C\rangle \in \mathcal{H}_{\text{code}}$,

$$\mathcal{U}_C(\rho) = \text{Tr}_{\neg(\mathcal{H}_{\text{bits}} \otimes \mathcal{H}_{\text{env}})}\Big[U\,(\rho \otimes |C\rangle\langle C|)\,U^\dagger\Big]. \tag{1}$$

Let $\Pi_C$ denote the classical reversible channel (permutation on basis of $\mathcal{H}_{\text{bits}}$, trivially extended over $\mathcal{H}_{\text{env}}$).

### 2.2 Binary boundary and basis correctness

**Definition 2** (Binary boundary). *For $\varepsilon \geq 0$, pair $(U, C)$ satisfies the $\varepsilon$-boundary if $\|\mathcal{U}_C - \Pi_C\|_\diamond \leq \varepsilon$.*

**Lemma 1** (Basis correctness). *If $\|\mathcal{U}_C - \Pi_C\|_\diamond \leq \varepsilon$, then for any basis input $|s\rangle\langle s| \otimes \rho_{\text{env}}$,*

$$d_{\text{TV}}(\,\text{Diag}\,\text{Tr}_{\neg\mathcal{H}_{\text{bits}}}\,\mathcal{U}_C(|s\rangle\langle s| \otimes \rho_{\text{env}}),\ \Pi_C(|s\rangle\langle s|)) \leq \tfrac{\varepsilon}{2}.$$

*Sketch.* The diamond norm upper-bounds worst-case output trace distance; classical TV distance is at most half the trace distance after dephasing (diagonal) and partial trace. □

**Remark 4** (Why diamond norm). *It certifies program-level worst-case proximity, including adversarially entangled inputs, aligning with deployment guarantees at commit.*

## 3 Causal Model and Bridge Objective

### 3.1 Executable causal world model

Let $\mathcal{E}_M$ be a CPTP map on $\mathcal{H}_{\text{bits}} \otimes \mathcal{H}_{\text{env}}$. Interventions $\text{do}(a)$ replace the action-controlled subchannel. For observation POVM $\{O_o\}$ and policy $\pi_C$,

$$p_M(o \,|\, \text{do}(\pi_C), h) = \text{Tr}\Big[O_o\, \mathcal{E}_M^{\text{do}(\pi_C),h}(\rho_0)\Big].$$

**Assumption 1** (Identifiability and scope). *(1) Interventions only modify action subchannels; (2) unobserved confounding beyond $\mathcal{E}_M$ is negligible for the task; (3) finite-sample estimation errors are controlled by validation. These can be relaxed but then causal terms require robustification.*

### 3.2 Bridge objective

We adopt computable description-length surrogates $L(\cdot)$ (e.g., compressed bit-length, architecture-size):

**Definition 3** (Bridge free energy).

$$\mathcal{F}_\beta(U, C, M) = L(C) + L(M) + \mathbb{E}_h[-\log p_M(o \,|\, \text{do}(\pi_C), h)] - \beta\, J(\pi_C), \quad \beta > 0.$$

**Remark 5** (Scaling). *$L(\cdot)$ and $-\log p_M(\cdot)$ are in bits/nats; $J$ is task-specific. Use $\beta = \beta_0/\sigma_J$ with empirical return scale $\sigma_J$ to normalize trade-offs.*

## 4 Optimization Problem and Practical Parameterization

$$\boxed{\min_{U,C,M}\ \mathcal{F}_\beta(U, C, M)\ \ \text{s.t.}\ \ U^\dagger U = \mathbb{1},\ \ \|\mathcal{U}_C - \Pi_C\|_\diamond \leq \varepsilon.} \tag{2}$$

**Remark 6** (Parameterizing $U$). *Let $U_\theta = \prod_\ell \exp(-i\theta_\ell H_\ell)$ for fixed Hermitian generators $\{H_\ell\}$. Optimize $\theta$ via Riemannian gradients on the unitary group. Classical orthogonal/symplectic analogues are viable when modeling over reals.*

**Remark 7** (Ancilla discipline). *Allow ancilla $A$ with $|0\rangle^{\otimes m}$ during computation, but require $\Pi_C : |s\rangle|0\rangle_A \mapsto |\pi_C(s)\rangle|0\rangle_A$; workspace must be uncomputed before commit.*

**Remark 8** (Ancilla (scratch) register). *An* ancilla *is a temporary helper register $A$ (size $m$) used during computation and* uncomputed *before commit. We require the classical target to act cleanly with ancilla:*

$$\Pi_C :\ |s\rangle|0\rangle_A^{\otimes m} \longmapsto |\pi_C(s)\rangle|0\rangle_A^{\otimes m},$$

*so no garbage remains in $A$ at commit. This makes the classical permutation auditable and reversible.*

## 5 Commit, Readout, and Timing

**Policy readout.** Designate action POVM $\{A_a\}$ on $\mathcal{H}_{\text{bits}}$:

$$\pi_C(a \,|\, \rho) = \mathrm{Tr}\Big[(A_a \otimes \mathbb{1})\, U\, (\rho \otimes |C\rangle\langle C|)\, U^\dagger\Big].$$

**Commit operation.** Produce a classical bitstring by computational-basis readout:

$$\mathsf{Commit}(\rho) = \arg\max_{s\in\{0,1\}^n} \ \langle s|\, \mathrm{Tr}_{\neg\mathcal{H}_{\text{bits}}}(\rho)\, |s\rangle\,.$$

By Lemma 1, boundary error $\varepsilon$ implies classical correctness at commit up to $\varepsilon/2$ in TV.

**Remark 9** (When to commit)**.** *Delay to accumulate gradient signal; commit when marginal value of further deliberation falls below cost. The boundary makes the act auditable even if deliberation used superpositions.*

## 6 Safe Self-Rewrite Under Invariants

Let $W_\Delta$ act on $\mathcal{H}_{\text{code}}$ and set $|C'\rangle = W_\Delta\,|C\rangle$. Let $G$ encode invariants (projection or zero-violation loss $L_G$).

**Definition 4** (Safe acceptance rule)**.** *Accept $C'$ iff*

$$\mathcal{F}_\beta(U,C',M)+\lambda\,\mathbb{E}_{s\sim d^{\text{mix}}} D_{\text{KL}}\big(\pi_{C'}(\cdot|s)\,\|\,\pi_C(\cdot|s)\big) < \mathcal{F}_\beta(U,C,M), \quad L_G(U,C')=0, \quad \|\mathcal{U}_{C'} - \Pi_{C'}\|_\diamond \le \varepsilon.$$

**Remark 10** (Why the KL term)**.** *It bounds behavioral drift under a reference state-distribution $d^{\text{mix}}$ (mixture of recent on-policy rollouts), avoiding optimism from evaluating under only old or only new policy.*

## 7 Existence, Limits, and Conjectured Benefits

**Theorem 1** (Existence and classical limit)**.** *Assume the feasible set of* (2) *is nonempty and compact modulo global phase (e.g., finite-dimensional truncations) and that $L(\cdot)$, $p_M$, and $J$ are lower semicontinuous. Then:*

1. *$\mathcal{F}_\beta$ attains a minimizer $(U^\star, C^\star, M^\star)$.*
2. *As $\varepsilon \to 0$, committed behavior converges to $\Pi_{C^\star}$, while preserving differentiable learning dynamics.*

**Conjecture 1** (Gradient advantage of unitary exploration)**.** *Relative to classical probabilistic mixtures, unitary evolution yields richer gradient signals via interference across near-sectors, improving credit assignment for sector discovery under smoothness assumptions on the task metric.*

## 8 Tractability, Surrogates, and Verification

**Remark 11** (Idealizations and barriers)**.** *(i) $K(\cdot)$ is uncomputable $\Rightarrow$ use computable $L(\cdot)$; (ii) diamond-norm proximity is QMA-hard in general $\Rightarrow$ use certified relaxations and statistical tests.*

**Definition 5** (Practical boundary estimator)**.**

$$\widehat{\Delta}(U,C) = \max_{s\in\mathcal{S}_{\text{test}},\ \rho_E\in\mathcal{R}} \ \frac{1}{2}\Big\|\mathrm{Diag}\ \mathrm{Tr}_{\neg\mathcal{H}_{\text{bits}}}\left(\mathcal{U}_C(|s\rangle\langle s|\otimes\rho_E)\right) - \Pi_C(|s\rangle\langle s|)\Big\|_1.$$

*Here $\mathcal{S}_{\text{test}}$ is adversarially expanded over training, and $\mathcal{R}$ includes representative (and stress-test) environment states.*

**Remark 12** (Certification pipeline)**.** *During training, use* $\widehat{\Delta}$ *as a regularizer and periodically project toward nearest permutation channel. For* deployment*, run a separate certification pipeline (tomographic estimation + SDP bounds + held-out adversarial tests) to bound commit TV error.*

# 9 Algorithmic Skeleton (Practical Approximation)

**Algorithm 1** Binary–Quantum Bridge (Approximate Training Loop)

Initialize $U_\theta$, program $C$, model $M$, invariant tests $G$, test-set $\mathcal{S}_{\text{test}}$, tolerance $\varepsilon$
**repeat**
  Roll out under $\pi_C$; collect trajectories $(h, o, r)$; augment $\mathcal{S}_{\text{test}}$ adversarially
  Estimate NLL $\sum -\log p_M(o \,|\, \text{do}(\pi_C), h)$ and return $\widehat{J}$
  Compute boundary estimate $\widehat{\Delta}(U_\theta, C)$
  $\widehat{\mathcal{F}}_\beta \leftarrow L(C) + L(M) - \beta\,\widehat{J} + \text{NLL} + \mu\,\widehat{\Delta}$
  Riemannian update: $\theta \leftarrow \theta - \eta \nabla_\theta \widehat{\mathcal{F}}_\beta$
  Update $M$ to minimize $L(M) + \text{NLL}$ under Assumption 1
  Propose code patch $C' = W_\Delta C$
  **if** $L_G(C') = 0$ and $\widehat{\mathcal{F}}_\beta(C') + \lambda\,\widehat{D_{\text{KL}}}(\pi_{C'} \| \pi_C) < \widehat{\mathcal{F}}_\beta(C)$ and $\widehat{\Delta}(U_\theta, C') \le \varepsilon$ **then**
    $C \leftarrow C'$
  **if** $\widehat{\Delta}(U_\theta, C) > \varepsilon$ **then**
    Project toward nearest permutation-like channel (architecture-specific)
**until** stopping criteria

**Remark 13** (Stopping criteria)**.** *Boundary tolerance met;* $\widehat{\mathcal{F}}_\beta$ *stabilizes; no accepted self-rewrites for* $T$ *rounds; resource limits reached.*

# 10 Governance and Readiness

We adopt a simple deployment gate: publish audited capability $\mathsf{Cap}$ and governance $\mathsf{Gov}$ indices and require $\mathsf{Gov} \ge \alpha\,\mathsf{Cap}$ with $\alpha > 1$. Absent this inequality, the system remains research-only and non-deployed.

# 11 Discussion and Interpretation

**Definition 6** (Observability covenant)**.** *A system is* observable in motion *if external evaluators can (i) recover its classical actions at commit, (ii) reconstruct which sector it occupied and which gates it traversed between commits, and (iii) detect, bound, or reproduce any self-rewrite that occurred during that interval.*

**What we can and should observe.**

1. **Classical commits (ground truth interface):** log the commit string $s^\star$, inputs, and outputs; this is the verifiable, binary face of the system.
2. **Sector dynamics (how it's moving):** record sector IDs or hashes, gate-crossing events, and *sector occupancy entropy* $H_t$ over a sliding window to show exploration vs. concentration.
3. **Boundary health (are guarantees intact):** track estimated boundary breach rate $\widehat{\Delta} > \varepsilon$, and publish test-set versions/coverage.

4. **Self-rewrite trace (what changed and why):** store $C \to C'$ diffs, objective deltas $\Delta\mathcal{F}_\beta$, policy drift $D_{\mathrm{KL}}(\pi_{C'} \| \pi_C)$, and guardian outcomes $G$.

5. **Causal calibration (is the model $M$ honest under interventions):** log interventional calibration error (NLL under $\mathrm{do}(\cdot)$) and targeted counterfactual checks on high-leverage actions.

**Suggested "motion" metrics (live dashboards).**

| | |
|---|---|
| Gate crossing rate | # of sector transitions per $N$ steps |
| Sector occupancy entropy $H_t$ | $-\sum_i p_t(S_i) \log p_t(S_i)$ |
| Boundary breach rate | $\Pr\left[\widehat{\Delta}(U, C) > \varepsilon\right]$ |
| MDL delta | $\Delta L(C) + \Delta L(M)$ per accepted rewrite |
| Policy drift | $D_{\mathrm{KL}}(\pi_{C'} \| \pi_C)$ on reference states |
| Interventional error | $\mathbb{E}[-\log p_M(o \,|\, \mathrm{do}(\pi_C), h)]$ |

**Remark 14** (What we *don't* need to see)**.** *We need not inspect internal amplitudes directly. The bridge guarantees classical correctness at commit; "motion" is evidenced by sector/gate telemetry, causal calibration, and reproducible self-rewrites*—all auditable at the classical interface.

- **Binary exactness at act time:** When we commit, the system behaves like a simple classical program $\Pi_C$ we can test and audit.
- **Information preservation while learning:** Unitary $U$ keeps information intact so gradients can flow end-to-end (no lossy bottlenecks).
- **One scalar objective to trade off three goods:** short code (*compression*), correct counterfactuals (*causal adequacy*), and good outcomes (*control*).
- **Self-rewrite with rails:** Patches must improve the score, stay close in behavior unless justified, pass invariants, and keep the boundary tight.

Learning is continuous and smooth; deployment is discrete and verifiable; both use the same executor. Ultimate it should be possible to observe an AGI in motion.

## How this differs from classical ML

If you replace $U$ with a standard neural network and drop the boundary, you get ordinary end-to-end training: powerful, but no program-level guarantees at act time. The bridge retains differentiable learning *and* offers a classical face at commit that can be certified and audited and perhaps lead to an AGI.The bridge retains fully differentiable learning while presenting a certifiable, auditable classical face at commit; it does not claim to constitute AGI, but it specifies a verifiable pathway that could enable AGI-like systems under additional assumptions and engineering.

## Limits (what we do not and can not claim)

- **Gradient advantage is a conjecture:** We hypothesize superposition/interference helps credit assignment across sectors; it is not proven here.
- **Boundary verification is hard:** Exact diamond-norm checks are QMA-hard; we rely on carefully chosen surrogates with statistical guarantees.

- **Description length is approximated:** Kolmogorov complexity is uncomputable; we use computable surrogates (compressed byte-lengths, gate counts).

These are explicit, testable, and should be the focus of future benchmarks.

### Failure modes and how the framework addresses them

- *Reward hacking:* Interventional likelihood punishes models that merely fit correlations; guardian $G$ enforces invariants.
- *Boundary drift:* Regular boundary testing + rollback triggers stop deployment if approximate classicality degrades.
- *Unsafe self-rewrite:* Acceptance rule adds a KL drift leash, invariant checks, and boundary re-validation before adopting $C'$.
- *Opaque behavior:* Commit strings, signed logs, and reproducible builds enable external auditing and forensics.

## 12 Conclusion

**Rounding up the prototype.** The BQB frames safe, self-improving agents as solutions to a constrained optimization that cleanly separates *how we learn* (differentiable, sector-aware, information-preserving) from *what we guarantee* (classical, auditable commit).

### A checklist

1. Have we parameterized $U$ to remain unitary and kept a clear policy readout?
2. Are we minimizing a single objective that balances description length, interventional likelihood, and return?
3. Do we enforce a binary boundary, with certified or statistically bounded error at commit?
4. Do self-rewrites pass (i) objective improvement, (ii) KL drift leash, (iii) invariants via $G$, (iv) boundary re-check?
5. Are telemetry, builds, and commits *signed* and *reproducible*?
6. Do independently audited dashboards satisfy $\mathsf{Gov} \geq \alpha\mathsf{Cap}$?

## A Finite-Dimensional Truncation and Compactness

**What we assume.** We restrict $\mathcal{H}_{\text{env}} \otimes \mathcal{H}_{\text{mem}}$ to an energy-bounded subspace of finite dimension $d_{\text{env}}$ and take $\mathcal{H}_{\text{bits}} \simeq (\mathbb{C}^2)^{\otimes n}$. Then $U$ lives in $\mathrm{U}(2^n d_{\text{env}})$, which is compact modulo global phase.

**Why it matters.** With a compact search space, lower semicontinuity of $L(\cdot)$, $p_M$, and $J$ guarantees $\mathcal{F}_\beta$ attains a minimizer. Without compactness, minimizers can "run to infinity," and existence proofs break.

## B Practical Complexity Measures

**We cannot use $K(\cdot)$; we use $L(\cdot)$.** Recommended, reproducible surrogates:

- $L(C)$**:** compressed byte-length of $C$ under a fixed prefix-free codec (report codec + version), plus gate counts or AST node counts if $C$ is a circuit/program.
- $L(M)$**:** compressed size of causal graph encoding + parameter tensors; also report graph sparsity and parameter count.

Publish raw sizes *and* compressed sizes to avoid compressor cherry-picking.

## C Diamond-Norm Surrogates

**Goal.** We want small $\|\mathcal{U}_C - \Pi_C\|_\diamond$, but exact checks are intractable. Use a three-pronged approach:

1. **Basis-state TV tests (fast):** Maintain an adversarially grown test set $\mathcal{S}_{\text{test}}$ of basis and small superposition states; bound

$$\widehat{\Delta}_{\text{TV}} = \max_{s \in \mathcal{S}_{\text{test}}} \text{TV}\big(\text{Diag } \text{Tr}_{\neg \mathcal{H}_{\text{bits}}} \mathcal{U}_C(|s\rangle \langle s|),\, \Pi_C(|s\rangle \langle s|)\big)$$

   with confidence intervals.

2. **Partial process tomography (medium):** Periodically estimate channel slices with confidence bounds to detect systematic deviations.

3. **SDP relaxations (slow but tighter):** On frozen snapshots, solve semidefinite relaxations to upper-bound $\|\cdot\|_\diamond$; publish the upper bound and the relaxation details.

**Certification flow.** Use (1) during training; gate checkpoints with (2); use (3) for the certified build before exposure.

## Notation quick reference (for readers)

- **Ancilla:** Temporary scratch register $A = |0\rangle^{\otimes m}$ used during computation and returned to $|0\rangle^{\otimes m}$ before commit:

$$\Pi_C : \ |s\rangle\, |0\rangle_A^{\otimes m} \mapsto |\pi_C(s)\rangle\, |0\rangle_A^{\otimes m}\,.$$

- **Total variation (TV):** For classical $p, q$, $\text{TV}(p,q) = \frac{1}{2}\sum_x |p(x) - q(x)|$. Under the boundary $\|\mathcal{U}_C - \Pi_C\|_\diamond \le \varepsilon$, commit-time distributions differ by $\le \varepsilon/2$ in TV.
- **Diamond norm:** Worst-case channel distance over all inputs and side information; upper-bounds post-measurement discrepancies used at commit.