Layer 1: Guided read-through of your draft

0. Basic explanation and section-by-section "translation" into plain English
(Based on "Draft v0.4.6 - Manuscript")

Document purpose: To help I quickly read your paper draft like a reviewer with a technical background, without changing your conclusions. I explain each main section in plain language, point out what is evidence vs. interpretation, and highlight typical pitfalls a reader may misunderstand.

Source: Your attached manuscript (Draft v0.4.6).

Version: Layer 1 - v2 (generated 15 January 2026).

Note: Draft v0.4.6 includes (i) APA-style in-text citations and a consolidated References section, (ii) an explicit Run registry/Provenance section (2.3a) and Table 0, which links table/figure results to specific output files. This guide therefore refers to Table 0 as an "audit trail".

How to use this guide
Read the Overview and Key Concepts first (5-10 min).
Then go to the section I are working on (e.g., "3.2 Dephasing-only").
When I revise the draft: use the "Reviewer will ask..." boxes as a checklist for clarifications.
This guide does not change your results; it helps make them clearer to the reader.

1. Overview: What does the paper investigate?

Your article investigates when a particular REAL construction provides a genuine advantage for a logical qubit subspace, when the system is exposed to noise (Lindblad noise). I compare REAL against a carefully matched baseline (NULL), so differences are not simply due to an "easier" control.

The central finding is that what is best depends on the noise type (noise regime): Under amplitude damping I typically see a coherent-but-leaky pattern (better quality inside the subspace, but more leakage out of it). Under pure dephasing, especially at stronger dephasing, REAL instead shows a net advantage on the unconditional metric (unconditional fidelity).

The core idea of the paper in one sentence
If I only look at conditional fidelity I can be misled: a construction can look "better"

because it improves the state given that it is still in the logical space, but at the same time it can increase the probability that the system leaves the logical space (leakage) - and thus be worse in practice.

2. Key concepts: short explanations

Term in the draft	What it means (plain language)	How it is used here
Logical subspace (L)	A two-dimensional "logical space" inside the full 4-qubit Hilbert space.	I measure how well the dynamics stay in this space over time.
Leakage L(t)	The probability that the system is outside the logical space.	If L(t) increases, I lose logical information out of the code/subspace.
Conditional fidelity F_cond(t)	How correct the logical state is, ignoring the cases where the system has leaked out.	Good for measuring quality "given survival", but can hide that many runs fail via leakage.
Unconditional fidelity F_uncond(t)	Quality weighted by the survival probability (survival-weighted).	Your primary net metric when leakage is a real failure mechanism.
Matched NULL	A baseline constructed to match key properties (here: spectrum) with REAL.	Makes the comparison fair: differences are attributed to structure, not trivial parameter differences.
Probe states (ZX vs rand64)	Which input states I test on the logical Bloch sphere.	ZX is a few canonical directions; rand64 tests many directions and reduces direction bias.
AUC	Area under a curve over time; a single number that summarizes the entire time evolution.	I use AUC to summarize F_uncond, F_cond and leakage over t in [0,5].
Threshold times	The time when a metric crosses a threshold, e.g., F_uncond < 0.90 or L > 0.10.	Captures time-to-failure and can be more relevant than AUC for certain

		applications.
Leakage-aware cost proxy	A simple score that penalizes leakage with a weight lambda.	Used to ask: when is REAL still a gain if leakage is costly?

3. Section by section: What are I saying - in plain language?

3.1 Title and Abstract

The title clearly signals that I are looking for crossover points between noise regimes: regions where REAL shifts from being net-bad to net-good, depending on the character and strength of the noise.

The abstract does four important things right: (1) it defines the matched control (REAL vs NULL), (2) it explains why I have two probe regimes (ZX for screening, rand64 for headline evidence), (3) it separates conditional/unconditional and leakage, and (4) it summarizes the main findings regime by regime.

A reviewer will typically ask (Abstract)
Is rand64 defined precisely (count, distribution, incl. basis states)?
Is "spectrum-matched NULL" explained enough so a reader understands what is matched - and what is not?
Are the key numbers/findings mentioned with direction (e.g., "DeltaAUC(F_uncond) > 0 at gamma_phi = 0.10"), without drowning the abstract?

3.2 1. Introduction

The introduction is about methodological risk: if I measure the wrong thing, I can conclude that an encoding is "better" even though it is worse in practice. I make it clear that leakage is a central failure mechanism, and therefore unconditional metrics and leakage must be included.

Next I explain your strict evaluation package: matched NULL, manifold probing, batch stability, basis robustness, and noise-localization sweeps. In practice this is an argument for internal validity: "I tried to close the most important loopholes".

A clear point I already have - which I can emphasize even more
The introduction can benefit from explicitly saying: "This is not a claim of universal error correction; it is a controlled benchmark of retention and leakage in an n = 4 testbed."

3.3 2. Methods: what I do, step by step

The Methods section describes the experiment as a virtual-lab setup. I have a 4-qubit system, but I define a 2D logical subspace (a logical qubit). I take a set of logical input states (probe states), evolve them in time under a Lindblad noise model, and measure how much they remain in the logical space (leakage) and how correct they are (fidelities).

The methodological key trick is that REAL is always compared to NULL under the same conditions (same probe states, time grid, noise parameters, randomization/batches). This makes the difference Delta = REAL - NULL meaningful: it can be attributed to the construction rather than to differences in the evaluation setup.

3.3.1 2.1 Logical subspace and probe states

I define a logical basis |0_L>, |1_L> and describe a general logical superposition on the Bloch sphere. The key point is that I test many directions, because the effect can be anisotropic: large for some logical directions and small or negative for others.

Why both ZX and rand64?
ZX (4 directions) is fast and can reveal obvious effects, but can miss direction dependence.
rand64 (about 66 directions) is manifold probing: better coverage of the Bloch sphere and more robust evidence.
In the draft, rand64 is used as the headline regime, while ZX is kept as diagnostics.

3.3.2 2.2 Open-system dynamics (Lindblad)

Here I state that I simulate open-system dynamics using the Lindblad formalism. For the reader this means: I assume Markovian noise (no memory) and model it via standard channels such as amplitude damping and dephasing.

I localize the noise to one qubit at a time (subset 0-3). This is a strength, because I can test robustness: if an effect is only seen for one specific qubit, it could be an artifact of the embedding or basis choice.

3.3.3 2.3 Matched-control design (REAL vs NULL)

This is the most important fairness component: NULL is constructed to be spectrum-matched to REAL. Intuition: two constructions that look similar on a key property, but where REAL has more structure. If REAL wins, it is plausible that this is due to structure rather than trivial differences.

A reviewer will typically ask (Matched control)
What does "spectrum-matched" mean concretely? (Which spectrum: of an operator, a Hamiltonian, a superoperator?)
What degrees of freedom is NULL allowed to have? (Fully random under a constraint, or optimized?)

Are REAL and NULL evaluated on exactly the same probe set and seed plan? (I say yes - that is important.)

3.3.4 2.3a Run registry and provenance

I include a run table that links your results to concrete output files and settings. This is very strong documentation: a reader can see exactly which noise model, rate, basis and probe regime underpin a figure or table.

When I later create figures, I may want to include the file ID/filename (from Table 0) directly in figure captions or in a supplement, so everything can be traced back to a specific artifact.

How to use Table 0 in practice: (1) Find the row corresponding to a given noise setup (noise_model, gamma, basis, subset, states). (2) Note the associated output files/logs. (3) Use Table 1/2 as the summary, but treat Table 0 as the source of truth for provenance, batch variation and sign (Delta) per batch.

3.3.5 2.3b Evidence criteria (decision rule)

I propose a simple decision criterion: an effect must (a) be matched (REAL-NULL), (b) be consistent across batches, (c) be large enough to be practically relevant, and (d) be interpreted jointly with leakage.

In practice this is a lightweight statistics framework without heavy assumptions: I use batch replication rather than asymptotic tests. This is reasonable in a pilot setup, but can later be strengthened by adding confidence intervals or bootstrap.

Practical tip: make the criterion operational
Consider adding one sentence: "We call an effect robust if all 3 batches have the same sign, and median |DeltaAUC| >= 0.01."
If I want to be stringent: also specify how I handle borderline cases (e.g., 2/3 batches positive but one batch near zero).

3.3.6 2.4 Metrics: leakage, conditional fidelity, unconditional fidelity

Here I define the three central quantities. A reader should especially understand the relation: F_uncond = (1 - leakage) x F_cond (in your notation: Tr[P_L rho(t)] x F_cond). This means unconditional fidelity automatically drops if leakage rises, even if F_cond improves.

This is exactly why the amplitude-damping regime can be coherent-but-leaky: REAL increases F_cond, but simultaneously increases leakage, so the net metric (F_uncond) becomes worse.

Rule of thumb for the reader
F_cond corresponds to "quality when it succeeds".
Leakage corresponds to "how often it fails by leaving the space".
F_uncond is "average quality", i.e., quality x survival.

3.3.7 2.5 Leakage-aware cost proxy

The cost proxy is your way of saying: "what if leakage is costly?" I apply a weight lambda to leakage and see when REAL remains positive. This produces a crossover argument: REAL is only attractive if the application tolerates a certain leakage price.

This is a good communication mechanism because it makes the tradeoff quantitative without claiming a universal best.

A reviewer will typically ask (Cost proxy)
What does lambda represent in practice? (e.g., a system requirement or a penalty in a higher-level control problem).
Is the proxy linear for convenience, or does it reflect a real cost model?
Should lambda be constant in time, or could early leakage be more costly than late leakage?

4. Results: What do your findings mean?

In the Results section I primarily report batch-wise DeltaAUC values (REAL-NULL) as well as threshold times. This guide focuses on: (a) sign and robustness (2/3 or 3/3 batches), (b) which metric wins/loses, and (c) how leakage explains the differences.

4.1 3.1 Amplitude damping: coherent-but-leaky tradeoff

When amplitude damping dominates (gamma_1 > 0), I consistently see the pattern: DeltaAUC(F_cond) > 0 but DeltaAUC(leakage) > 0 and DeltaAUC(F_uncond) < 0. In plain language: REAL makes the state nicer when the system is still in the logical space, but REAL also more often drives the system out of the logical space, so the net outcome is worse.

I test three strengths (gamma_1 = 0.10, 0.05, 0.02) and see that the effect scales down with lower gamma_1, but the sign structure does not change. This is a strong regime argument: not a single data point, but a trend.

How I can explain this to a reader without over-interpreting
State empirically: "REAL increases conditional fidelity but also increases leakage in the amplitude-damping regime."

Avoid claiming causality without analysis: I can write "consistent with a tradeoff between coherence preservation and subspace retention".

4.2 3.2 Dephasing-only at gamma_phi = 0.10: net unconditional advantage

When I turn off amplitude damping (gamma_1 = 0) and run pure dephasing at gamma_phi = 0.10, the picture changes: DeltaAUC(F_uncond) becomes positive and batch-stable. That is, REAL provides a net advantage on the metric that already accounts for leakage.

I also show that the effect is robust with respect to logical basis (noise_diag vs eigen) and robust with respect to which physical qubit receives the noise (subsets 0-3). This is exactly the type of robustness that makes a reviewer more comfortable calling it a real effect.

What the reader should take away here
REAL is not always good. But under dephasing-only at moderate strength, REAL appears net-better (unconditional).
Your basis and subset robustness is a substantial evidence boost.

4.3 3.3 Dephasing sweep: crossover as gamma_phi changes

I test gamma_phi = 0.05, 0.10, 0.20. The point is to find a crossover: a point where sign(DeltaAUC(F_uncond)) flips.

At gamma_phi = 0.05 I see no net unconditional advantage (DeltaAUC(F_uncond) around zero or negative), even though DeltaAUC(F_cond) is still positive and a leakage penalty is present. At gamma_phi = 0.20 the unconditional advantage becomes large, and leakage is even lower for REAL than for NULL (DeltaAUC(leakage) < 0).

In plain language: when dephasing is strong enough, REAL appears both to retain more logical information on average and to leak less.

A reviewer will typically ask (Sweep)
Are three points enough to localize the crossover, or should I add 0.07/0.08/0.12 to show a smoother transition?
Is there a physical explanation for why stronger dephasing helps REAL? (can be left as future work).
Are all sweep runs identical except for gamma_phi? (I should explicitly say yes).

4.4 3.4 Mixed noise (both): AUC advantage, but early threshold penalty

When I combine dephasing (gamma_phi = 0.10) with a small amplitude damping (gamma_1 = 0.02), I again see a positive unconditional AUC gain. However, the threshold times show an early penalty: REAL crosses a given threshold earlier than NULL (negative Delta t), i.e., earlier failure under that definition.

This is an important and mature point: AUC and thresholds measure different things. AUC rewards later improvements over the full interval, while thresholds capture the first time things become unacceptable.

Which conclusion is safe here?
Safe: "REAL can have a net AUC advantage under mixed noise, but can be worse on early-time thresholds."
Not safe without extra analysis: "REAL is better in practice" - it depends on the application's cost function (AUC vs threshold).

5. How to read the tables in the draft

The tables (Table 1 and Supplementary) are your evidence-carrying elements. Here is a way to read them that makes the sign logic completely clear.

5.1 Table 1: Batch-level AUC deltas

Each block shows three numbers per batch: DeltaAUC(F_uncond), DeltaAUC(F_cond), DeltaAUC(leak). Interpretation in one line:

- If DeltaAUC(F_uncond) > 0 and DeltaAUC(leak) <= 0, REAL is clearly better (net and with leakage under control).

- If DeltaAUC(F_uncond) < 0 but DeltaAUC(F_cond) > 0, I have a coherent-but-leaky tradeoff.

- If everything is close to zero, there is no practical effect at those settings (or statistical power is low).

Good practice (which I already do)
Report batches separately instead of only giving an average - it shows stability.
State what I consider practical significance (your |Delta| >= 0.01 rule).

5.2 Endpoint quantiles and thresholds

Endpoint quantiles (q10/q50/q90) at t = 5.0 help the reader see the distribution: is the effect a broad shift (all quantiles move), or only a tail effect (only q90 moves)?

Thresholds tell when something becomes unacceptable under a concrete threshold. This is especially relevant if the system must remain above, e.g., 0.90 fidelity for a period.

If I want to make thresholds even stronger
State precisely how the threshold is computed (interpolation between time steps or first discrete crossing).
Report both median and IQR (or q25/q75) if possible.

6. Language and communication tips that do not change your conclusions

The items below are low-risk improvements: they do not change the results, but make them harder to misunderstand.

6.1 Skærp problemformuleringen i 1–2 sætninger
Forslag: 'Vi undersøger, om en struktureret konstruktion (REAL) giver bedre retention af et logisk underspace end en spektrum-matchet baseline (NULL) under Lindblad-støj, når vi eksplicit medregner leakage.'

6.2 Gentag relationen mellem metrikkerne
Indsæt evt. en kort boks: 'F_uncond = survival × F_cond; derfor kan F_cond stige samtidig med at F_uncond falder, hvis leakage stiger.'

6.3 Vær eksplicit om hvad 'robust' betyder
Du bruger allerede batches og $|\Delta|\geq 0.01$. Skriv det meget tydeligt i Methods og referér til det i Results.

7. Mini-checklist before I move to the next version

Add 1-3 figures that visualize the tradeoff (F_uncond vs leakage) and the sweep crossover (DeltaAUC vs gamma_phi).

Make "spectrum-matched NULL" more concrete with one explanatory sentence or a short technical note.

Ensure that all places with placeholders (missing formulas/definitions) are either filled in or marked as to be filled.

Consider adding one extra sweep point (e.g., gamma_phi = 0.12) if I want to localize the crossover more sharply.

If stable_pool is ever used for primary evidence: separate selection and evaluation (holdout seeds) and document it.

End of the Layer 1 guide. If I want, the next "Layer 2" can be a more technical reviewer commentary, where we propose concrete rewrites line by line without changing the substance.