Lag 1: Guidet gennemlæsning af dit draft

Indhold

0. Basal forklaring og afsnit-for-afsnit 'oversættelse' til almindeligt dansk

(Baseret på 'Draft v0.4.6 — Manuskript')

Dokumentformål: At hjælpe dig med hurtigt at læse dit paper-draft som en 'reviewer med teknisk baggrund', uden at ændre dine konklusioner. Jeg forklarer hvert hovedafsnit i almindeligt sprog, peger på hvad der er evidens, hvad der er fortolkning, og hvilke faldgruber en læser typisk kan misforstå.

Kilde: Dit vedlagte manuskript (Draft v0.4.6).

Version: Lag 1 – v2 (genereret 15. January 2026).

Note: Draft v0.4.6 inkluderer (i) APA-style in-text citations og en samlet References-sektion, (ii) et eksplicit Run registry/Provenance-afsnit (2.3a) samt Table 0, som binder tabel/figur-resultater til konkrete outputfiler. Denne guide refererer derfor til Table 0 som "audit trail".

Sådan bruger jeg denne guide

- Læs først Overblik og Nøglebegreber (5–10 min).
- Gå derefter til den sektion jeg arbejder på (fx '3.2 Dephasing-only').
- Når jeg retter draftet: brug 'Reviewer vil spørge…'-bokse som tjekliste til præciseringer.
- Guiden ændrer ikke dine resultater; den hjælper med at gøre dem tydeligere for læseren.

1. Overblik: Hvad undersøger paperet?

Din artikel undersøger, hvornår en bestemt 'REAL'-konstruktion giver en reel fordel for et logisk qubit-underspace, når systemet udsættes for støj (Lindblad-støj). jeg sammenligner REAL mod en nøje matchet baseline (NULL), så effekter ikke bare skyldes en 'nemmere' kontrol.

Det centrale fund er, at 'hvad der er bedst' afhænger af støjtypen (noise regime): Ved amplitude damping får jeg typisk et 'coherent-but-leaky' mønster (bedre kvalitet inde i underspacet, men mere lækage ud af det). Ved ren dephasing, især ved stærkere dephasing, får REAL derimod en nettofordel på det ubetingede mål (unconditional fidelity).

Paperets kerneidé i én sætning

- Hvis jeg kun kigger på conditional fidelity kan jeg narres: en konstruktion kan se 'bedre' ud, fordi den forbedrer tilstanden givet at man stadig er i logik-rummet, men samtidig kan den øge sandsynligheden for at systemet forlader logik-rummet (leakage) – og dermed blive dårligere 'i praksis'.

2. Nøglebegreber: korte forklaringer

Term i draftet	Hvad det betyder (almindeligt sprog)	Hvordan det bruges her
Logical subspace (L)	Et 2-dimensionelt 'logik-rum' inde i hele 4-qubit Hilbert-rummet.	Du måler hvor godt dynamikken bliver i dette rum over tid.
Leakage L(t)	Sandsynligheden for at systemet er uden for logik-rummet.	Hvis L(t) stiger, 'taber' jeg logisk information ud af koden/underspacet.
Conditional fidelity F_cond(t)	Hvor korrekt logik-tilstanden er, hvis man ser bort fra de tilfælde hvor systemet er lækket ud.	God til at måle kvalitet 'givet overlevelse', men kan skjule at mange runs fejler ved leakage.
Unconditional fidelity F_uncond(t)	Kvalitet vægtet med overlevelsessandsynlighed (survival-weighted).	Dit primære 'netto'-mål når leakage er en reel fejlmekanisme.
Matched NULL	En baseline der er konstrueret til at matche nøgleegenskaber (her: spektrum) med REAL.	Gør sammenligningen fair: forskelle tilskrives struktur, ikke trivielle parametre.
Probe states (ZX vs rand64)	Hvilke inputtilstande jeg tester på den logiske Bloch-sfære.	ZX er få kanoniske retninger; rand64 prøver mange retninger og reducerer 'retning-bias'.
AUC	Arealet under en kurve over tid; én talværdi der summerer hele tidsforløbet.	Du bruger AUC til at opsummere F_uncond, F_cond og leakage over $t \in [0,5]$.
Threshold times	Tidspunkt hvor et mål krydser en tærskel, fx F_uncond<0.90 eller L>0.10.	Fanger 'tid-til-fejl' og kan være mere relevant end AUC for visse applikationer.
Leakage-aware cost proxy	En simpel score der 'straffer' leakage med en vægt λ.	Bruges til at sige: hvornår er REAL stadig en gevinst hvis leakage er dyr?

3. Afsnit-for-afsnit: Hvad siger jeg – i almindeligt sprog?

3.1 Titel og Abstract

Titlen signalerer tydeligt, at jeg leder efter 'crossover'-punkter mellem støjregimer: altså områder hvor REAL skifter fra at være netto-dårlig til netto-god, afhængigt af støjens karakter og styrke.

Abstractet gør fire vigtige ting rigtigt: (1) det definerer den matchede kontrol (REAL vs NULL), (2) det forklarer hvorfor jeg har to probe-regimer (ZX som screening, rand64 som hoved-evidens), (3) det adskiller conditional/unconditional og leakage, og (4) det opsummerer hovedfundene regime-for-regime.

Reviewer vil typisk spørge (Abstract)

- Er rand64 præcist defineret (antal, fordeling, inkl. basisstates)?
- Er 'spectrum-matched NULL' forklaret nok til at en læser forstår, hvad der matches – og hvad der ikke matches?
- Er de vigtigste tal/fund nævnt med retning (fx 'ΔAUC(F_uncond)>0 ved γϕ=0.10'), uden at drukne abstractet?

3.2 1. Introduction

Introduktionen handler om metodisk risiko: hvis man måler forkert, kan man konkludere at en encoding er 'bedre', selvom den i praksis er dårligere. jeg gør det klart, at leakage er en central fejlmekanisme, og derfor skal unconditional mål og leakage med.

Dernæst forklarer jeg din 'strenge' evalueringspakke: matched NULL, manifold probing, batch-stabilitet, basis-robusthed, og noise-localization sweep. Det er i praksis et argument for intern validitet: 'jeg har forsøgt at lukke de vigtigste loopholes'.

Klar pointe jeg allerede har – som jeg kan fremhæve endnu mere

- Introduktionen kan med fordel eksplicit sige: 'Dette er ikke en påstand om universel fejlkorrigering; det er en kontrolleret benchmark af retention og leakage i et n=4 testbed.'

3.3 2. Methods: hvad jeg gør, trin for trin

Methods-delen beskriver eksperimentet som et 'virtuel-lab' setup. jeg har et 4-qubit system, men jeg definerer et 2D logisk underspace (et logisk qubit). jeg tager en række logiske inputtilstande (probe states), kører dem frem i tid under en Lindblad-støjemodel, og måler hvor meget de bliver i logik-rummet (leakage) og hvor korrekte de er (fidelities).

Det metodiske 'nøgletrick' er, at REAL altid sammenlignes mod NULL under samme betingelser (samme probe states, tidsgrid, støjparametre, randomisering/batches). Dermed bliver forskellen Δ = REAL − NULL meningsfuld: den kan tilskrives konstruktionen og ikke en forskel i evalueringsopsætningen.

3.3.1 2.1 Logical subspace and probe states

Du definerer en logisk basis |0_L>, |1_L> og beskriver en generel logisk superposition på Bloch-sfæren. Det afgørende er, at jeg tester mange retninger, fordi 'effekten' kan være anisotrop: den kan være stor for nogle logiske retninger og lille eller negativ for andre.

Hvorfor både ZX og rand64?

- ZX (4 retninger) er hurtigt og kan afsløre åbenlyse effekter, men kan overse retning-afhængighed.
- rand64 (≈66 retninger) er 'manifold probing': bedre dækning af Bloch-sfæren og mere robust evidens.
- I draftet bruges rand64 som 'headline', mens ZX gemmes som diagnostik.

3.3.2 2.2 Open-system dynamics (Lindblad)

Her fortæller du, at jeg simulerer åben-system dynamik med Lindblad-formalisme. For læseren betyder det: jeg antager Markovsk støj (ingen hukommelse) og beskriver støjen via standard kanaler som amplitude damping og dephasing.

Du lokaliserer støjen til én qubit ad gangen (subset 0–3). Det er en styrke, fordi jeg kan teste robusthed: hvis en effekt kun ses for én bestemt qubit, kan det være en artefakt af embedding eller basisvalg.

3.3.3 2.3 Matched-control design (REAL vs NULL)

Dette er den vigtigste 'fairness'-del: NULL er konstrueret til at være spektrum-matchet til REAL. Intuition: to konstruktioner der 'på papiret' ligner hinanden på en nøgleegenskab, men hvor REAL har mere struktur. Hvis REAL vinder, er det derfor plausibelt at det skyldes strukturen og ikke trivielle forskelle.

Reviewer vil typisk spørge (Matched control)

- Hvad betyder 'spectrum-matched' konkret? (Hvilket spektrum: af en operator, et Hamiltonian, en superoperator?)
- Hvilke frihedsgrader må NULL have? (Er den helt tilfældig under en constraint, eller optimeret?)
- Er REAL og NULL evalueret på helt samme probe-sæt og seed-plan? (Du skriver ja – det er vigtigt.)

3.3.4 2.3a Run registry og provenance

Du har en run-tabel, der binder dine resultater til konkrete outputfiler og settings. Det er meget stærkt ift. dokumentation: en læser kan se præcis hvilken støjmodel, rate, basis og probe-regime der ligger bag en figur eller tabel.

Når jeg senere laver figurer, kan jeg med fordel skrive fil-ID/filnavn (fra Table 0) direkte i figur-caption eller i et supplement, så man kan spore alt tilbage til en bestemt artefakt.

Sådan bruger jeg Table 0 i praksis: (1) Find den række, der svarer til en given støjopsætning (noise_model, γ, basis, subset, states). (2) Notér de tilhørende outputfiler/logs. (3) Brug derefter Table 1/2 som "summary", men lad Table 0 være sandhedskilden for provenance, batch-variation og fortegn (Δ) pr. batch.

3.3.5 2.3b Evidence criteria (decision rule)

Du opstiller et simpelt beslutningskriterium: en effekt skal (a) være matched (REAL−NULL), (b) være konsistent på tværs af batches, (c) være stor nok til at være praktisk relevant, og (d) tolkes sammen med leakage.

Det her er i praksis et 'mini-statistik'-framework uden tunge antagelser: jeg bruger batch-replikation i stedet for asymptotiske tests. Det er helt rimeligt i en pilot-opsætning, men kan forbedres ved at tilføje konfidensintervaller eller bootstrap senere.

Praktisk tip: gør kriteriet operationelt

- Overvej at skrive én sætning: 'Vi kalder en effekt robust, hvis alle 3 batches har samme fortegn, og median $|\Delta AUC| \geq 0.01$.'
- Hvis jeg vil være stringent: angiv også, hvordan jeg håndterer borderline cases (fx 2/3 batches positive men én batch tæt på nul).

3.3.6 2.4 Metrics: leakage, conditional fidelity, unconditional fidelity

Her definerer jeg de tre centrale størrelser. En læser skal især forstå relationen: F_uncond = (1 − leakage) × F_cond (i din notation: Tr[P_L ρ(t)] × F_cond). Det betyder, at unconditional fidelity automatisk falder hvis leakage stiger, selv hvis F_cond forbedres.

Det er netop derfor amplitude damping-regimet kan blive 'coherent-but-leaky': REAL øger F_cond, men det sker samtidig med øget leakage, så netto (F_uncond) bliver dårligere.

Huskeregel til læseren

- F_cond svarer til 'kvalitet når det lykkes'.
- Leakage svarer til 'hvor tit det mislykkes ved at falde ud af rummet'.
- F_uncond er 'kvalitet i gennemsnit', dvs. kvalitet × overlevelse.

3.3.7 2.5 Leakage-aware cost proxy

Cost proxy'en er din måde at sige: 'hvad hvis leakage er dyrt?' jeg lægger en vægt λ på leakage, og ser hvornår REAL stadig er positiv. Det giver et crossover-argument: REAL er kun attraktiv hvis applikationen tolererer en vis leakage-pris.

Det er en god kommunikationsmekanisme, fordi den gør tradeoff'en kvantitativ uden at påstå et universelt 'bedst'.

Reviewer vil typisk spørge (Cost proxy)

- Hvad repræsenterer λ i praksis? (fx et systemkrav eller en 'penalty' i et higher-level control problem).
- Er proxy'en lineær af bekvemmelighed, eller afspejler den en egentlig cost-model?
- Skal λ være konstant i tid, eller kunne tidlig leakage være dyrere end sen leakage?

4. Resultater: Hvad betyder dine fund?

I resultatafsnittet rapporterer jeg primært batch-vise ΔAUC'er (REAL−NULL) samt threshold-tider. Guiden her fokuserer på: (a) fortegn og robusthed (2/3 eller 3/3 batches), (b) hvilken metrik der vinder/taber, og (c) hvordan leakage forklarer forskellene.

4.1 3.1 Amplitude damping: 'coherent-but-leaky' tradeoff

Når amplitude damping dominerer (γ1>0), ser jeg konsekvent mønsteret: ΔAUC(F_cond) > 0 men ΔAUC(leakage) > 0 og ΔAUC(F_uncond) < 0. I almindeligt sprog: REAL laver tilstanden pænere, når systemet stadig er i logik-rummet, men REAL får oftere systemet til at forlade logik-rummet, så netto bliver det dårligere.

Du tester tre styrker (γ1=0.10, 0.05, 0.02) og ser at effekten 'skalerer ned' med lavere γ1, men sign-strukturen ændrer sig ikke. Det er et stærkt regime-argument: ikke et enkelt datapunkt, men en trend.

Hvordan jeg kan forklare det til en læser uden at overfortolke

- Konstater empirisk: 'REAL øger conditional fidelity men øger også leakage i amplitude damping-regimet.'
- Undgå at påstå årsag uden analyse: skriv evt. 'konsistent med et tradeoff mellem koherensbevarelse og subspace-retention'.

4.2 3.2 Dephasing-only ved γϕ=0.10: netto unconditional fordel

Når jeg slår amplitude damping fra (γ1=0) og kører ren dephasing ved γϕ=0.10, ændrer billedet sig: ΔAUC(F_uncond) bliver positiv og batch-stabil. Dvs. REAL giver en nettofordel på det mål, der tæller leakage med.

Du viser også, at effekten er robust ift. logisk basis (noise_diag vs eigen) og robust ift. hvilken fysisk qubit der får støjen (subsets 0–3). Det er præcis den type robusthed, der gør en reviewer mere tryg ved at kalde det en 'rigtig' effekt.

Hvad læseren bør tage med her

- REAL er ikke 'altid god'. Men under dephasing-only ved moderat styrke ser REAL ud til at være netto bedre (unconditional).
- At jeg har basis- og subset-robusthed er et væsentligt evidensløft.

4.3 3.3 Dephasing sweep: crossover når γϕ ændres

Du tester γϕ=0.05, 0.10, 0.20. Pointen er at finde et 'crossover': et punkt hvor sign(ΔAUC(F_uncond)) skifter.

Ved γϕ=0.05 får jeg ingen netto unconditional fordel (ΔAUC(F_uncond) omkring nul eller negativ), selvom ΔAUC(F_cond) stadig er positiv og leakage-penalty er til stede. Ved γϕ=0.20 bliver unconditional fordelen stor, og leakage bliver endda lavere for REAL end for NULL (ΔAUC(leakage) < 0).

I almindeligt sprog: når dephasing er stærk nok, ser REAL ud til både at holde mere logisk information i gennemsnit og samtidig lække mindre.

Reviewer vil typisk spørge (Sweep)

- Er de tre punkter nok til at lokalisere crossover, eller bør jeg tilføje 0.07/0.08/0.12 for at vise en glattere overgang?
- Er der en fysisk forklaring på hvorfor stærkere dephasing hjælper REAL? (kan holdes som 'future work').
- Er alle sweep-runs helt identiske bortset fra γϕ? (du bør eksplicit sige ja).

4.4 3.4 Mixed noise ('both'): fordel i AUC, men threshold-penalty tidligt

Når jeg kombinerer dephasing (γϕ=0.10) med en lille amplitude damping (γ1=0.02), ser jeg igen en positiv unconditional AUC-gevinst. Men threshold-tiderne viser en tidlig penalty: REAL krydser en given tærskel tidligere end NULL (negativ Δt), dvs. tidligere 'fejl' efter den definition.

Det er en vigtig, moden pointe: AUC og thresholds måler forskellige ting. AUC belønner senere forbedringer over hele intervallet, mens threshold fanger 'første gang det bliver dårligt'.

Hvilken konklusion er 'sikker' her?

- Sikker: 'REAL kan have netto AUC-fordel under mixed noise, men kan være dårligere på early-time thresholds.'
- Ikke sikker uden ekstra analyse: 'REAL er bedre i praksis' – det afhænger af applikationens cost-funktion (AUC vs threshold).

5. Sådan læser jeg tabellerne i draftet

Tabellerne (Table 1 og Supplementary) er dine 'bevisbærende' elementer. Her er en måde at læse dem på, som gør sign-logikken helt klar.

5.1 Table 1: Batch-level AUC deltas

Hver blok viser tre tal pr. batch: ΔAUC(F_uncond), ΔAUC(F_cond), ΔAUC(leak). Tolkning i én linje:

• Hvis ΔAUC(F_uncond)>0 og ΔAUC(leak)≤0, er REAL klart bedre (netto og med kontrol over leakage).

• Hvis ΔAUC(F_uncond)<0 men ΔAUC(F_cond)>0, har jeg et 'coherent-but-leaky' tradeoff.

• Hvis alt er tæt på nul, er der ingen praktisk effekt ved de settings (eller power er lav).

God praksis (som jeg allerede gør)

- Rapportér batches separat i stedet for kun at give et gennemsnit – det viser stabilitet.
- Angiv hvad jeg betragter som 'praktisk betydning' (din $|\Delta| \geq 0.01$ regel).

5.2 Endpoint quantiles og thresholds

Endpoint quantiles (q10/q50/q90) ved t=5.0 hjælper læseren med at se fordelingen: er effekten en bred 'shift' (alle kvantiler flytter sig), eller er det kun en hale-effekt (kun q90 flytter sig)?

Thresholds fortæller 'hvornår' noget bliver uacceptabelt efter en konkret tærskel. Det er særligt relevant hvis systemet skal holde sig over fx 0.90 fidelity i en periode.

Hvis jeg vil gøre thresholds endnu stærkere

- Angiv præcis hvordan threshold beregnes (interpolation mellem tidssteg eller første diskrete crossing).
- Rapportér både median og IQR (eller q25/q75) hvis muligt.

6. Sprog- og formidlingstips, der ikke ændrer dine konklusioner

Nedenstående er 'lav-risiko' forbedringer: de ændrer ikke resultaterne, men gør dem sværere at misforstå.

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-tjekliste før jeg går videre til næste version

- Tilføj 1–3 figurer, der visualiserer tradeoff'en (F_uncond vs leakage) og sweep-crossover (ΔAUC vs $\gamma\phi$).
- Gør 'spectrum-matched NULL' mere konkret med én forklarende sætning eller en kort teknisk note.
- Sørg for at alle steder med placeholders (manglende formler/definitioner) er udfyldt eller markeret som 'to be filled'.
- Overvej at tilføje et ekstra sweep-punkt (fx $\gamma\phi=0.12$) hvis jeg vil lokalisere crossover skarpere.
- Hvis stable_pool nogensinde bruges til primær evidens: adskil selection og evaluation (holdout seeds) og dokumentér det.