`run_pipeline_full_open_systems_v14.6`

Handbuch: Artefakte, Analyse, TI-Validierung und Skalierung
Technische Dokumentation

antaris

2. September 2025

Zusammenfassung

Dieses Handbuch beschreibt die Nutzung der Pipeline `run_pipeline_full_open_systems_v14.x` für (i) Artefakt-Erzeugung (*Build*), (ii) verlustfreie Nachauswertung (*Analyze*) und (iii) realistische Vollläufe (*All*). Schwerpunkt ist die **valide Überprüfung der These, dass die Thermale Interpretation (TI) auf dem Sierpinski-Tetraeder-Graph (ST) emergiert**. Ausgangspunkt sind p.c.f.-Axiome; *alle* TI-Postulate (inkl. Detektor-Response-Prinzip, DRP) folgen daraus innerhalb der Pipeline. Die Validierung nutzt mehrere komplementäre Tests (KL, χ^2, Likelihood-Ratio G, Wilson-Intervalle, TOST-Äquivalenz) und strukturelle Kennzahlen (Spektraldimension). Das Skript ist **voll skalierfähig**: Aussagekraft skaliert mit Rechenleistung; alle Laufzeit-/Genauigkeitshebel sind Parameter-gesteuert.

1 Zweck, Aufbau und Artefakte

Ziel. Numerische Demonstration, dass unter p.c.f.-basiertem ST und Pixel-POVMs die empirischen Detektionshäufigkeiten $\hat{p}_k$ (aus MCWF-Trajektorien) die TI-Vorhersagen p_k^{TI} praktisch reproduzieren. Rand-/Diskretisierungseffekte werden durch ST-Level und Durchmischung (vor erstem Klick) kontrolliert.

Modularer Aufbau. *Build* erzeugt Artefakte (ST-Graph, Randbedingungen, Renorm r, Kerne, ρ, POVM/Pixel). *Analyze* lädt ausschließlich diese Artefakte und führt dynamische Ebenen (Schrödinger, GKSL/MCWF) und Tests aus. Damit sind Analysen *verlustfrei* reproduzierbar (kein Informationsverlust zwischen Build und Analyze).

2 Kernmethoden und Gleichungen

2.1 Geschlossene Dynamik

$$\mathrm{i}\,\frac{d}{dt}\,|\psi_t\rangle = H\,|\psi_t\rangle, \qquad H = H^\dagger. \tag{1}$$

2.2 Offene Systeme (GKSL/Lindblad)

$$\dot{\rho} = -\,\mathrm{i}[H,\rho] \;+\; \sum_j \gamma_j\Big(L_j\rho L_j^\dagger - \tfrac{1}{2}\{L_j^\dagger L_j,\rho\}\Big)\,, \tag{2}$$

mit komplett-positiven, spurtreuen semigruppenartigen Abbildungen (GKSL-Form).

2.3 MCWF / Quanten-Sprünge (Quantum Trajectories)

Effektiver Hamiltonoperator $H_{\text{eff}} = H - \frac{\mathrm{i}}{2}\sum_j \gamma_j L_j^\dagger L_j$. Zwischen Sprüngen:

$$|\tilde{\psi}(t+\Delta t)\rangle = \exp(-\mathrm{i} H_{\text{eff}}\Delta t)\,|\psi(t)\rangle, \tag{3}$$

Sprungwahrscheinlichkeit $p_j = \Delta t\,\gamma_j \|L_j|\psi\rangle\|^2$; bei Sprung: Normierung von $L_j|\psi\rangle$ und Fortsetzung.

2.4 POVM/Detektion

Detektor-Pixel definieren eine POVM $\{E_k\}$, $E_k \geq 0$, $\sum_k E_k = \mathbb{I}$. Vorhersage:

$$p_k^{\text{TI}} = \operatorname{tr}(\rho\, E_k) \quad \text{(oder zeitgemittelt entlang der Dynamik).} \tag{4}$$

3 Refutation- & Äquivalenz-Suite

3.1 Güte-zu-TI

$$\text{KL:} \quad D_{\text{KL}}(\hat{p}\|\, p^{\text{TI}}) = \sum_k \hat{p}_k \log \frac{\hat{p}_k}{p_k^{\text{TI}}}; \tag{5}$$

$$\chi^2\text{-Test:} \quad \chi^2 = \sum_k \frac{(n_k - Np_k^{\text{TI}})^2}{Np_k^{\text{TI}}}, \ \text{df} = K - 1; \tag{6}$$

$$\text{LRT-}G\text{:} \quad G = 2\sum_k n_k \log \frac{n_k}{Np_k^{\text{TI}}} \ \approx\ 2N\, D_{\text{KL}}. \tag{7}$$

Für Binomial-/Multinomial-Fälle wird asy. χ^2-Theorie nach Wilks/Pearson genutzt.

3.2 Konfidenz & Äquivalenz

Wilson-Intervalle für einzelne Pixel-Anteile und *TOST* gegen Band $[-\varepsilon, +\varepsilon]$ um p_k^{TI}:

- Wilson-CI: robuste Abdeckung im Binomialfall bei endlichem N.
- TOST: zwei einseitige Tests prüfen $p_k^{\text{TI}} - \varepsilon < \hat{p}_k < p_k^{\text{TI}} + \varepsilon$ auf Niveau α.

Standard-*Auto*-Marge $\varepsilon_{\text{auto}} = 1.96 \cdot \text{SE}$ ist streng; für *praktische* Äquivalenz empfehlen wir eine feste, physikalisch begründete Marge (z. B. $\varepsilon = 0.02 = 2\,\text{pp}$).

3.3 Struktureller Konsistenzcheck: Spektraldimension

Aus dem Diagonaleintrag $K_t(ii)$ des Wärme-/Propagator-Kerns: $K_t(ii) \sim t^{-d_s/2}$; Regression von $\log K_t(ii)$ gegen $\log t$ liefert d_s (mit R^2 als Qualitätsmaß).

4 Emergenz der TI auf ST (p.c.f. ⇒ TI-Postulate inkl. DRP)

Die ST-Diskretisierung folgt p.c.f.-Axiomen; daraus ergeben sich lokale Energien/Laplacian, Wärmekernel und schließlich TI-ähnliche Invarianzstrukturen der Detektorantwort. In der Pipeline wird das *Detector Response Principle* (DRP) operationalisiert: Pixel-POVMs und deren Zeitentwicklung unter GKSL/MCWF erzeugen Klickverteilungen, die (nach ausreichender Ausmischung) die TI-Vorhersage p^{TI} reproduzieren. Die Refutation-/Äquivalenz-Suite belegt dies quantitativ.

5 Argumente (normaler PC) und Designentscheidungen

Für robuste Ergebnisse bei moderater Hardware (z. B. 6–12 Kerne, 16–32 GB RAM):

`--level 4` guter Kompromiss: deutlich weniger Rand-/Diskretisierungsartefakte als L2/L3, noch zügige Rechenzeit.

`--traj yes`, `--ntraj 3000`
$\sim$ 3000 Trajektorien reichen für enge Wilson-CIs, ohne dass Mini-Bias sofort hochsignifikant wird.

`--dt 0.01`, `--tmax_traj 3.0`
viele echte Klicks ($>90\%$ der Pfade), Frühtransienten werden relativiert.

`--det_gamma 1.8`
genügend Zeit zur Durchmischung *vor* dem ersten Klick; zu große Werte führen zu zu frühen Klicks, zu kleine verlängern unnötig.

`--dephase_site 0.00`
Dephasierung für TI-Checks vermeiden (sonst künstliche Asymmetrien).

`--eigs_mode small`, `--eigs_k 64`
genügt für TI-Validierung und spart Rechenzeit.

`--save_*` IO minimal halten: `--save_open_rho no`, `--save_traj_events no`, `--save_kernels no`, `--open_series_stride 20`.

`--alpha 0.05`, `--eq_margin_abs 0.02`
praxisnahe Äquivalenzmarge (2 pp); für Härtetests `--eq_margin_abs 0` (Auto-ε).

6 Skalierbarkeit (*alles* hängt an Rechenleistung)

Embarrassingly parallel: Trajektorien sind unabhängig; nahezu lineare Beschleunigung mit CPU-Kernen/Nodes. BLAS/OpenMP-Threads (MKL/`OPENBLAS_NUM_THREADS`) sinnvoll auf 2–8 Kerne begrenzen, um Oversubscription zu vermeiden.
Hebel: `--level`↑ (größere Operatoren/Netze), `--ntraj`↑, `--nt`↑, `--tmax_traj`↑ ⇒ Laufzeit↑, SE↓, Tests schärfer. Bei ausreichend Ressourcen lassen sich diese Parameter *beliebig* erhöhen; die Pipeline ist voll skalierbar.

7 Startbefehle (Praxis, Level 4)

A) Artefakte bauen (einmalig)

```
# PowerShell 7.x
python run_pipeline_full_open_systems_v14.6.py `
--stage build `
--artifacts out/ART_L4 `
--level 4 `
--t 0.6 --s 0.4 --beta 1.2 `
--bc dirichlet --use_fixed_r

# bash
python run_pipeline_full_open_systems_v14.6.py \
--stage build \
--artifacts out/ART_L4 \
```

```
--level 4 \
--t 0.6 --s 0.4 --beta 1.2 \
--bc dirichlet --use_fixed_r
```

B) Drei Analysen (Seeds 11/13/19; feste Äquivalenzmarge 2 pp)

```
# PowerShell 7.x
$PY = "python"; $SCRIPT = "run_pipeline_full_open_systems_v14.6.py"
$COMMON = @("--stage","analyze","--artifacts","out/ART_L4",
"--pixels_json","detector_pixels.json","--mixer","unitary",
"--schro","yes","--ham","laplacian","--gamma","1.0","--sigma0","0.12",
"--tmax","3.5","--nt","40",
"--open","yes","--dephase_site","0.00","--dephase_pixel","0.00","--
    det_gamma","1.8","--loss","0.0",
"--traj","yes","--ntraj","3000","--dt","0.01","--tmax_traj","3.0","--
    traj_scheme","jumps","--traj_unobs","nojump",
"--sg","no","--eigs_mode","small","--eigs_k","64","--save_L","yes","--
    save_kernels","no","--save_eigs_csv","yes",
"--save_open_rho","no","--open_series_stride","20","--save_modes","no","
    --save_R_diag","no","--save_traj_events","no",
"--alpha","0.05","--eq_margin_abs","0.02")
& $PY $SCRIPT $COMMON --out out/ANA_L4_TI_seed11 --seed 11
& $PY $SCRIPT $COMMON --out out/ANA_L4_TI_seed13 --seed 13
& $PY $SCRIPT $COMMON --out out/ANA_L4_TI_seed19 --seed 19

# bash
PY=python; SCRIPT=run_pipeline_full_open_systems_v14.6.py
COMMON_ANALYZE=(--stage analyze --artifacts out/ART_L4
--pixels_json detector_pixels.json --mixer unitary
--schro yes --ham laplacian --gamma 1.0 --sigma0 0.12
--tmax 3.5 --nt 40
--open yes --dephase_site 0.00 --dephase_pixel 0.00 --det_gamma 1.8 --
    loss 0.0
--traj yes --ntraj 3000 --dt 0.01 --tmax_traj 3.0 --traj_scheme jumps --
    traj_unobs nojump
--sg no --eigs_mode small --eigs_k 64 --save_L yes --save_kernels no --
    save_eigs_csv yes
--save_open_rho no --open_series_stride 20 --save_modes no --save_R_diag
     no --save_traj_events no
--alpha 0.05 --eq_margin_abs 0.02)
$PY "$SCRIPT" "${COMMON_ANALYZE[@]}" --out out/ANA_L4_TI_seed11 --seed 11

$PY "$SCRIPT" "${COMMON_ANALYZE[@]}" --out out/ANA_L4_TI_seed13 --seed
    13
$PY "$SCRIPT" "${COMMON_ANALYZE[@]}" --out out/ANA_L4_TI_seed19 --seed
    19
```

C) Härtetest: Auto-ε (ohne feste Marge)

```
# PowerShell 7.x
python run_pipeline_full_open_systems_v14.6.py `
--stage analyze `
--artifacts out/ART_L4 `
--out out/ANA_L4_TI_AUTO `
--pixels_json detector_pixels.json --mixer unitary `
--schro yes --ham laplacian --gamma 1.0 --sigma0 0.12 `
```

```
--tmax 3.5 --nt 40 `
--open yes --dephase_site 0.00 --dephase_pixel 0.00 --det_gamma 1.8 --
    loss 0.0 `
--traj yes --ntraj 3000 --dt 0.01 --tmax_traj 3.0 --seed 13 --
    traj_scheme jumps --traj_unobs nojump `
--sg no --eigs_mode small --eigs_k 64 --save_L yes --save_kernels no --
    save_eigs_csv yes `
--save_open_rho no --open_series_stride 20 --save_modes no --save_R_diag
     no --save_traj_events no `
--alpha 0.05 --eq_margin_abs 0

# bash
python run_pipeline_full_open_systems_v14.6.py \
--stage analyze \
--artifacts out/ART_L4 \
--out out/ANA_L4_TI_AUTO \
--pixels_json detector_pixels.json --mixer unitary \
--schro yes --ham laplacian --gamma 1.0 --sigma0 0.12 \
--tmax 3.5 --nt 40 \
--open yes --dephase_site 0.00 --dephase_pixel 0.00 --det_gamma 1.8 --
    loss 0.0 \
--traj yes --ntraj 3000 --dt 0.01 --tmax_traj 3.0 --seed 13 --
    traj_scheme jumps --traj_unobs nojump \
--sg no --eigs_mode small --eigs_k 64 --save_L yes --save_kernels no --
    save_eigs_csv yes \
--save_open_rho no --open_series_stride 20 --save_modes no --save_R_diag
     no --save_traj_events no \
--alpha 0.05 --eq_margin_abs 0
```

8 Interpretation typischer Resultate

- **Sehr kleine Abweichung:** KL $\lesssim 10^{-4}$, $p_{\chi^2} \gtrsim 0.5 \Rightarrow$ klassische GoF-Tests bestehen komfortabel.
- **TI-Äquivalenz (strict):** *True*, wenn GoF $p \geq \alpha$ *und* alle Wilson-Intervalle p_k^{TI} einschließen *und* TOST mit gewählter ε-Marge besteht.
- **Auto-ε vs. feste Marge:** Auto-ε ist streng (skaliert mit SE). Für „praktische" TI genügt oft $\varepsilon = 2\,\mathrm{pp}$.
- **Spektraldimension:** Stabilisiert sich mit Level $\uparrow$ und reflektiert die zugrundeliegende ST-Geometrie.
- **Level:** Der ST-Graph ist ein Approximant an den unendlich verschachtelten Ur-ST-Graph. Mit level 4 ist die Approximation dagegen sehr gering und der TI-Äquivalenztest besteht nicht in allen Fällen aber die Abweichungen sind gering Ein ST-Graph mit Level 5 erfordert schon mehrere Stunden Rechenzeit, bei ca. 100MB gespeicherte Daten. Bei Level 6 ist der Umfang der Rohdaten bereits ca. 1GB. Ab Level 7 wird sehr viel Arbeitsspeicher benötigt und ist nur mittels professioneller Rechentechnik lösbar. Der Flaschenhals ist einzig die verfügbare Rechenleistung. Je besser die Approximation, also je höher der Level des Approximanten ist, umso besser divergieren die Werte. Das zeigt schon der kleiner werdende Unterschied zwischen Level 2 auf 3 bzw. Level 3 auf 4.

9 Reproduzierbarkeit & Archivierung

Jeder Run erzeugt (im TI-Unterordner) `Refutation_Report.pdf` und `refutation_summary.json` (alle Kennzahlen). Zusammen mit `detector_pixels.json`, CLI-Aufruf und Skriptversion bildet dies ein Repro-Bundle.

Anhang A: Heuristiken

- Ziel: Klickquote $>90\%$ der Trajektorien (mehr Gewicht auf Spätzeiten $\Rightarrow$ Bias $\downarrow$).
- BLAS/OpenMP-Threads sinnvoll wählen (2–8), IO minimieren.
- Für strengere Statistik: `--ntraj`$\uparrow$, `--tmax_traj`$\uparrow$, `--level`$\uparrow$.

Literatur

[1] K. Pearson (1900). On the criterion… Goodness-of-Fit. *Philosophical Magazine* **50**(302), 157–175.

[2] S. S. Wilks (1938). The large-sample distribution of the likelihood ratio…*Ann. Math. Stat.* **9**(1), 60–62.

[3] S. Kullback, R. A. Leibler (1951). On Information and Sufficiency. *Ann. Math. Stat.* **22**(1), 79–86.

[4] J. Lin (1991). Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* **37**(1), 145–151.

[5] L. D. Brown, T. T. Cai, A. DasGupta (2001). Interval Estimation for a Binomial Proportion. *Statistical Science* **16**(2), 101–117.

[6] A. Agresti (2013/2018). *Categorical Data Analysis* (3. Aufl.) / *An Introduction to Categorical Data Analysis* (3. Aufl.). Wiley.

[7] S. Wellek (2010). *Testing Statistical Hypotheses of Equivalence and Noninferiority* (2. Aufl.). Chapman & Hall/CRC.

[8] G. Lindblad (1976). On the generators of quantum dynamical semigroups. *Commun. Math. Phys.* **48**, 119–130.

[9] V. Gorini, A. Kossakowski, E. C. G. Sudarshan (1976). Completely positive dynamical semigroups of N-level systems. *J. Math. Phys.* **17**(5), 821–825.

[10] J. Dalibard, Y. Castin, K. Mølmer (1992). Wave-function approach to dissipative processes in quantum optics. *Phys. Rev. Lett.* **68**(5), 580–583.

[11] M. B. Plenio, P. L. Knight (1998). The quantum-jump approach to dissipative dynamics in quantum optics. *Rev. Mod. Phys.* **70**(1), 101–144.

[12] M. A. Nielsen, I. L. Chuang (2010). *Quantum Computation and Quantum Information* (10th Anniversary Ed.). Cambridge Univ. Press.

[13] J. Kigami (2001). *Analysis on Fractals.* Cambridge Univ. Press.

[14] M. T. Barlow, E. A. Perkins (1988). Brownian motion on the Sierpinski gasket. *Probab. Theory Relat. Fields* **79**, 543–623.