

# TI-on-ST v14.6 (full): Formale Supplementary zur Pipeline, Dynamik, POVMs und MCWF

antaris

3. September 2025

## Zusammenfassung

Dieses Dokument erläutert die *vollständige* formale Grundlage der v14.6 “full” Open-Systems-Pipeline zum ST-Graphen, eng angelehnt an die CLI-Optionen und die Konfigurationsdateien (`run_pipeline_full_open_systems_v14.6.py`, `detector_pixels.json`, `ARGUMENTS_v16.4.md`).<sup>1</sup> Schwerpunkt sind (i) die Operator- und Semigruppen-Struktur (Laplacian, Heat-Kernel, Gibbs-Zustand), (ii) POVM-Detektion inkl. Pixel-Geometrien und Unitär-Mischer, (iii) GKSL-Mastergleichung und Monte-Carlo-Wellenfunktions-Trajektorien (MCWF), sowie (iv) Statistik (Goodness-of-Fit & Äquivalenzttests), *vergleichend* zu Annahmen und Vorhersagen der Thermalen Interpretation (TI) von Neumaier [1, 2, 3, 4, 5]. Nebenbemerkungen zum Render/Output erfolgen nur am Rande.

## 1 ST-Graph, Laplace-Operator und Heat-Kernel

### 1.1 Diskreter ST-Graph und Laplacian

Der Sierpinski-Tetraeder-Graph (ST) auf Level  $L$  ist ein endlich großer, post-critically finite (pcf) selbstähnlicher Graph. Der (ungewogene) Graph-Laplacian lautet

$$L = D - A,$$

mit Adjazenzmatrix  $A$  und Gradmatrix  $D$ . Auf p.c.f. Fraktalen bilden Widerstandsformen (resistance forms) die Grundlage der Analysis; die Effektiv-Widerstandsmetrik  $R(\cdot, \cdot)$  induziert Sobolev- und Dirichlet-Strukturen [13, 14]. Dies trägt *sub-gaussiane* Heat-Kernel-Schätzungen und spektrale Dimensionen  $d_S < 2$  auf typischen Fraktalgraphen [16, 17].<sup>2</sup>

### 1.2 Heat-Semigruppe und Gibbs-Zustand

Die (Markov-)Semigruppe

$$K_t := e^{-tL}, \quad t \geq 0,$$

definiert den Heat-Kernel auf Knotenebene (Matrixelemente  $K_t(x, y)$ ). Für p.c.f. Fraktale gelten sub-gaussian Asymptotiken

$$K_t(x, y) \lesssim \frac{1}{V(x, t^{1/d_w})} \exp\left(-c \left(\frac{R(x, y)^{d_w}}{t}\right)^{\frac{1}{d_w-1}}\right),$$

wobei  $d_w$  die walk-Dimension und  $V(x, r)$  Volumina in der Widerstandsmetrik sind [14, 16, 17].<sup>3</sup>

<sup>1</sup>Zur Codebasis: `run_pipeline_full_open_systems_v14.6.py`. Argumente/Optionen (aktualisierte Doku v16.4, argument-nah zu v14.6): `ARGUMENTS_v16.4.md`. Detektor-Definitionen: `detector_pixels.json`.

<sup>2</sup>Fraktal-Analysis und Widerstandsmetrik: [13, 14]. Sub-Gaussian HK auf (Random/Fractal) Graphs: [16, 17]. Siehe auch Überblick [15].

<sup>3</sup>Semigruppen-Grundlagen sind Standard; für fraktale Geometrien siehe [13, 14].

Ein kanonischer Gibbs-Zustand (Dichteoperator) für inverse Temperatur  $\beta > 0$  bzgl.  $L$  ist

$$\rho_\beta = \frac{e^{-\beta L}}{\text{Tr}(e^{-\beta L})}.$$

Er wird im Skript direkt (bzw. indirekt über spektrale Zerlegung von  $L$ ) gebildet und dient als *a-priori* Zustandsbeschreibung für die TI-Vorhersage von Zählern/Pixeln (s.u.).<sup>4</sup>

**Zeitskalierung.** Die Option `--s` (oder äquivalente Pipeline-Schalter) skaliert effektive Zeiten  $t \mapsto s t$  (und/oder  $L \mapsto s^{-1} L$ ), um Level-abhängige Zeitskalen (Harmonic-Scaling) konsistent zu halten (Hintergrund: Energieskalen und Widerstandsmetriken auf p.c.f. Graphen) [13].<sup>5</sup>

### 1.3 Spektrale Dimension

Aus dem diagonalen Heat-Kernel gilt heuristisch

$$K_t(x, x) \asymp t^{-d_S/2} \quad (t \downarrow 0),$$

und damit  $d_S = -2 \frac{d \log K_t(x, x)}{d \log t}$  (Fit in einem geeigneten Fenster) [15]. In der Pipeline wird  $d_S$  über Slope-Fits der log-log-Kurve bestimmt.

## 2 Detektion als POVM: Pixel, Geometrien, Mischer

### 2.1 Pixel-Definition und Selektoren

Die Datei `detector_pixels.json` beschreibt  $K$  Pixel über Mengen  $S_k \subseteq V$  (Knoten), aufgebaut aus *Selektoren* (`selectors`): `sphere`, `gauss3d`, `cone`, `polyhedron`, `address_prefix`, `union`, `exclude` usw.<sup>6</sup> Formal entsteht für jedes Pixel  $k$  ein Effekt-Operator  $E_k$  durch Projektion auf den (normalisierten) Indikator  $P_{S_k}$  und eine Effizienz  $0 \leq \eta_k \leq 1$ :

$$E_k = \eta_k P_{S_k}, \quad P_{S_k} = \sum_{x \in S_k} |x\rangle\langle x|.$$

Die Menge  $\{E_k\}$  bildet zusammen mit ggf. einem “No-Click”-Effekt  $E_0$  eine POVM, d. h.

$$E_k \geq 0, \quad \sum_{k=0}^K E_k = \mathbb{I},$$

ggf. nach geeigneter Normierung von Effizienzen/Masken. Dies entspricht der Standard-Messungstheorie mit verallgemeinerten Messungen [11, Ch. 2][12].<sup>7</sup>

### 2.2 Unitäre Mischer (`--mixer`) und 50/50-Beamsplitter

Die Pipeline gestattet `--mixer unitary` mit vorgegebenem  $U \in \text{U}(K)$  (z. B. aus JSON) oder `--mixer beamsplitter_50_50` mit

$$U_{\text{BS}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix},$$

<sup>4</sup>Implementationsnah: Konstruktion von  $e^{-\beta L} / e^{-tL}$  über Exponential von  $L$  im Code.

<sup>5</sup>Zur Zeitskalierung siehe die Argument-Doku; die v16.4-ARGUMENTS spiegelt das v14.6-Set weitgehend.

<sup>6</sup>JSON-Schema: `detector_pixels.json`.

<sup>7</sup>Die Bildung von  $E_k$  erfolgt im Skript in `povm_from_pixels` analog zu obigem Schema (gewichtete Projektoren auf Pixelmengen).

oder äquivalenten Konventionen aus der Quantenoptik. Der Mischer definiert eine neue Pixelbasis (“Moden”) durch

$$E'_j = \sum_{k,\ell} U_{jk}^* E_k U_{j\ell},$$

sodass die *Messwahrscheinlichkeit* (TI-Vorhersage) wird

$$p_j^{(\text{TI})} = \text{Tr}(\rho_\beta E'_j).$$

Dies entspricht der Standardbehandlung verallgemeinerter Messungen nach vorgelagerter unitärer Modenmischung [11].<sup>8</sup>

**Formale Sicht:** Definiert man die (spärliche) Matrix  $C \in \mathbb{R}^{K \times |V|}$  mit  $C_{k,x} = 1$  für  $x \in S_k$  (sonst 0), dann sind  $E_k = \eta_k C^\dagger |k\rangle\langle k| C$  und  $E'_j = C^\dagger U^\dagger |j\rangle\langle j| U C$ . Die TI-Wahrscheinlichkeiten  $p^{(\text{TI})}$  werden zu

$$p^{(\text{TI})} = \text{diag}(U C \rho_\beta C^\dagger U^\dagger).$$

### 3 Dynamik: geschlossen/offen, GKSL und MCWF

#### 3.1 Geschlossene Dynamik (--schro yes)

Bei geschlossener Dynamik wird (falls aktiviert) die Schrödinger-Entwicklung mit Hamiltonoperator  $H$  integriert. In der ST-Pipeline wird häufig  $H$  aus dem Laplacian gewonnen (--ham laplacian), etwa  $H = \gamma L$  mit Skalenfaktor  $\gamma > 0$  (CLI: --gamma). Damit

$$\dot{\rho}(t) = -i[H, \rho(t)], \quad \rho(0) = \rho_\beta \text{ oder präpariert.}$$

#### 3.2 Offene Dynamik (GKSL, --open yes)

Für offene Systeme implementiert die Pipeline die GKSL-Mastergleichung [6, 7]:

$$\dot{\rho}(t) = -i[H, \rho(t)] + \sum_{\alpha} \gamma_{\alpha} \left( L_{\alpha} \rho(t) L_{\alpha}^{\dagger} - \frac{1}{2} \{ L_{\alpha}^{\dagger} L_{\alpha}, \rho(t) \} \right), \quad (1)$$

mit Sprung-/Dephasierungsoperatoren  $L_{\alpha}$  und Raten  $\gamma_{\alpha} \geq 0$ . Typisch:

- **Site-Dephasierung** (--dephase\_site):  $L_x = |x\rangle\langle x|$  für Knoten  $x \in V$  (Pure-Dephasing in Ortsbasis).
- **Pixel-Dephasierung** (--dephase\_pixel):  $L_k = \sum_{x \in S_k} |x\rangle\langle x|$  (Dephasing innerhalb eines Pixels).
- **Detektionskanal(e)** (--det\_gamma): beobachtete Sprünge  $J_k$  (s. u.) für MCWF-Klicks; zusätzlich Verlustkanäle --loss mit  $L_{\text{loss}}$ .

Die rechte Seite (1) wird im Skript als `lindblad_rhs_full` ausgewertet, Integration via explizitem RK4 (`integrate_master_rk4_full`) mit Zeitschritt `--dt` und  $t \in [0, t_{\text{max}}]$  (`--tmax`).<sup>9</sup>

<sup>8</sup>Im Code wird dies effizient über einen Pixel-Projektionsoperator  $C$  umgesetzt (“mode projection”); dann ist  $p = \text{diag}(U C \rho C^\dagger U^\dagger)$ .

<sup>9</sup>Vektorisiert/Operatorform im Code: `lindblad_rhs_full`, `integrate_master_rk4_full`. Formale Grundlagen GKSL: [6, 7].

### 3.3 Monte-Carlo-Wellenfunktionsmethode (MCWF, --traj yes)

Die MCWF/“Quantum Jumps”-Methode simuliert einzelne Trajektorien  $|\psi_t\rangle$  [8, 9, 10]. Unbeobachtete Kanäle (z. B. Dephasierung) gehen in den *effektiven* nicht-hermiteschen Hamiltonoperator

$$H_{\text{eff}} = H - \frac{i}{2} \sum_{\alpha \in \text{unobs}} \gamma_{\alpha} L_{\alpha}^{\dagger} L_{\alpha}$$

ein; die stochastischen Sprünge zu beobachteten Kanälen  $J_k$  (Detektoren) treten mit Rate  $r_k(t) = \gamma_k \langle \psi_t | J_k^{\dagger} J_k | \psi_t \rangle$  auf.<sup>10</sup> Im Skript werden  $N$  Trajektorien (--ntraj) bis  $t_{\text{max}}^{(\text{traj})}$  (--tmax\_traj) mit Schritt --dt simuliert, wahlweise mit --traj\_unobs nojump (No-Jump für unbeobachtete Kanäle).<sup>11</sup>

**Detektionsoperatoren.** Für beobachtete Pixelkanäle  $k$  wählt man  $J_k$  proportional zur Projektorstruktur des Pixels (ggf. nach Mischer  $U$ ), z. B.  $J_k \propto C^{\dagger} U^{\dagger} |k\rangle\langle k| U C$ . Dann ist die MCWF-Click-Statistik  $f_k$  (relative Häufigkeiten) mit der TI-Vorhersage  $p_k^{(\text{TI})} = \text{Tr}(\rho_{\beta} E'_k)$  zu vergleichen (s. u.).

## 4 Statistik: TI-Vorhersage $p^{(\text{TI})}$ vs. MCWF-Häufigkeiten $f$

### 4.1 Wahrscheinlichkeitsvektoren und Divergenzen

Aus der POVM ergibt sich die TI-Vorhersage  $p^{(\text{TI})} = (\text{Tr}(\rho_{\beta} E'_1), \dots, \text{Tr}(\rho_{\beta} E'_K))$ , aus MCWF die relativen Häufigkeiten  $f = (f_1, \dots, f_K)$ . Als Distanz-/Divergenzmaße nutzt die Pipeline u. a.: KL-Divergenz  $D_{\text{KL}}(f||p)$  [20], Jensen-Shannon-Divergenz  $\text{JSD}(f, p)$  [21, 22].<sup>12</sup>

### 4.2 Goodness-of-Fit und Äquivalenz

Zur Güteprüfung werden  $\chi^2$ -/G-Tests (Likelihood-Ratio; Wilks) [23] und Konfidenzintervalle (Wilson-Score für Binomialanteile) [24] verwendet. Für *Äquivalenz* (praktische Gleichheit innerhalb Toleranz) kommt die TOST-Prozedur nach Schuirmann [26, 27] zur Anwendung: Man wähle Äquivalenzgrenzen  $\pm\Delta$  (z. B. max. akzeptierte Abweichung der Anteile) und teste  $H_{01} : (f_k - p_k) \leq -\Delta$  und  $H_{02} : (f_k - p_k) \geq +\Delta$ . Werden *beide* Einseiten-Tests verworfen, gilt Äquivalenz.<sup>13</sup>

## 5 Vergleich mit der Thermalen Interpretation (TI)

Die TI (Neumaier) interpretiert die Größen  $\langle A \rangle = \text{Tr}(\rho A)$  als *q-Erwartungswerte* realer, objektiver Eigenschaften physikalischer Systeme [1, 2]. Messgeräte sind *makroskopische* Systeme mit grobgranularen Zeiger-/Pointervariablen (“world tubes”); Messergebnisse sind *näherungsweise* Repräsentationen solcher q-Erwartungswerte, nicht Einzelereignisse [3].<sup>14</sup> Im vorliegenden Rahmen:

- **Zustand:**  $\rho_{\beta}$  ist die effektive Zustandseingabe (ggf. nach Dynamik  $t \mapsto \rho(t)$ ), passend zur TI-Lesart als realer Erwartungswertträger.
- **Detektor als POVM:** Die Pixel-POVM  $\{E'_k\}$  modelliert makroskopische Pointer (Coarse-Graining eines feinen Observablen-Algebras), TI-konform: die *Vorhersagen* sind  $p_k^{(\text{TI})} = \text{Tr}(\rho E'_k)$ .

<sup>10</sup>MCWF-Formalismus: [8, 9, 10].

<sup>11</sup>Trajektorien-Routine im Skript (Klickerfassung pro Pixel/Modus).

<sup>12</sup>Hintergrund: KL [20], JSD [21, 22]. Siehe auch Wilks’ LRT [23] für Likelihood-Vergleiche.

<sup>13</sup>Implementations-Details der Tests liegen in den Statistik-Hilfsroutinen; die Argumentlisten enthalten Schalter für Toleranzen/Signifikanz.

<sup>14</sup>TI-Grundlagen & Messung: [1, 2, 3, 4, 5].

- **Dynamik & Irreversibilität:** GKSL/MCWF liefern eine mikroskopische (effektive) Beschreibung offener Systeme, deren *makroskopische* Zähler die  $p_k^{(\text{TI})}$  approximieren (*Frequenznäherung* bei großen Ensembles).
- **Trajektorien vs. TI:** Einzel-Trajektorien (Klickfolgen) sind *nicht* die Realität der TI; relevant ist, dass ihre *Statistik*  $f$  im Äquivalenztest die  $p^{(\text{TI})}$  innerhalb tolerierter Grenzen reproduziert.

Damit wird die Kernforderung der TI erfüllt: q-Erwartungswerte (hier:  $\text{Tr}(\rho E'_k)$ ) steuern makroskopisch beobachtbare Anteile in realen Geräten; Einzelereignisse sind emergente, stochastische Manifestationen [3].<sup>15</sup>

## 6 Optionen der Pipeline und ihr Formalismus (Auswahl)

(Optionennamen gem. ARGUMENTS\_v16.4.md; das Set spiegelt v14.6; Implementierung in run\_pipeline\_full\_op)

**--level L.** Wählt ST-Level  $L$  (Knotenzahl, Geometrie, Spektrum  $L$ ). Formal beeinflusst dies  $K_t = e^{-tL}$ ,  $\rho_\beta$ ,  $d_S$ , Widerstandsmetriken.

**--t, --tmax, --nt.** Zeiten für Heat-Kernel/Dynamik:  $t \in [0, t_{\max}]$ , ggf. Gitter  $t_j$ . In Fits für  $d_S$  und Sub-Gaussian-Checks wird ein geeignetes Fenster genutzt.

**--s.** Zeitskalierung  $t \mapsto s t$  bzw.  $L \mapsto s^{-1} L$  (Level-Harmonisierung; vgl. fraktale Zeitskalen).

**--beta.** Inverse Temperatur für  $\rho_\beta = e^{-\beta L}/Z$ .

**--bc dirichlet/neumann.** Randbedingungen an Eckknoten/äußeren Schalen; formal wirkt dies als Einschränkung des Operatorraums (Projektions- bzw. Laplace-Modifikation).

**--use\_fixed\_r.** Fixiert ein Harmonisierungsskalierungs- $r$  (z.B. aus Vorläufen), ansonsten Schätzung aus harmonischen Energien/Resistancen (niveauabhängige Zeit-/Energieskalen) [13].

**--pixels\_json.** Lädt  $\{S_k, \eta_k\}$  und Selektorlogik (s. o.). Formal definiert es die POVM  $\{E_k\}$  (bzw. nach Mischer  $\{E'_k\}$ ).

**--mixer unitary / beamsplitter\_50\_50.** Anwendung eines  $U \in \text{U}(K)$  vor der Messung:  $E'_j = C^\dagger U^\dagger |j\rangle\langle j| UC$  und  $p^{(\text{TI})} = \text{Tr}(\rho E'_j)$ .

**--schro yes, --ham, --gamma.** Geschlossene Dynamik. Üblich:  $H = \gamma L$ .

**--open yes, --dephase\_site, --dephase\_pixel, --det\_gamma, --loss.** GKSL nach (1). Site-/Pixel-Dephasing (unbeobachtet), Detektionssprünge (beobachtet), Verlustkanäle.

**--traj yes, --ntraj, --dt, --tmax\_traj, --seed, --traj\_unobs nojump.** MCWF-Simulation von Klicksequenzen; Statistik  $f$  für Tests gegen  $p^{(\text{TI})}$ .

<sup>15</sup>Die Pipeline realisiert diese Abbildung *explizit* (POVM aus Pixeln, Mischer, MCWF-Statistik  $f$  und Tests  $f \approx p^{(\text{TI})}$ ).

<sup>16</sup>Optionen/Erklärung: ARGUMENTS\_v16.4.md. Implementierende Routinen: run\_pipeline\_full\_open\_systems\_v14.6.py.

--sg yes, --dw, --sg\_tmin, --sg\_tmax. Sub-Gaussian-Check mit Exponent  $d_w$  im Zeitfenster; vergleicht  $-\log K_t(x, y)$  gegen  $(\frac{R(x, y)^{d_w}}{t})^{1/(d_w-1)}$  [14, 17].

## 7 Formale Ableitungen und Prüfpunkte

### 7.1 TI-Vorhersage als Erwartungswert einer POVM

Mit  $E'_j \geq 0$  und  $\sum_j E'_j = \mathbb{I}$  ist

$$p_j^{(\text{TI})} = \text{Tr}(\rho E'_j) \in [0, 1], \quad \sum_j p_j^{(\text{TI})} = 1.$$

Dies ist die *einzig* Annahme auf TI-Seite: Messgeräte sind grobgranulare POVMs; ihre Anzeigen sind (bei Repetition) Näherungen der q-Erwartungswerte [3, 2].

### 7.2 MCWF-Klickraten und Erwartungswerte

Wähle beobachtete Sprünge  $J_k$  proportional zu den Mess-Effekten  $E'_k$  (“optisches” Modell). Dann ist die Momentanrate  $r_k(t) = \gamma_k \langle \psi_t | J_k^\dagger J_k | \psi_t \rangle$ . Für ergodische/steady-state-Setups folgt asymptotisch

$$\mathbb{E}[f_k] \approx \text{Tr}(\rho_{\text{ss}} E'_k),$$

wobei  $\rho_{\text{ss}}$  stationär zu (1) ist. Damit erhalten wir die TI-Relation  $f \approx p^{(\text{TI})}$  als emergente, statistische Aussage über Makro-Zähler [9, 10, 3].

### 7.3 Sub-Gaussian Konsistenz und Geometrie

Die Validierung der sub-gaussianen Form mittels Widerstandsmetrik  $R$  (aus  $L^+$ , dem Moore-Penrose-Pseudoinversen) und des Fit-Exponenten  $d_w$  ist ein *geometrischer* Konsistenzcheck: TI benötigt keine spezielle (klassische) Geometrie; sie ist kompatibel mit fraktaler Diffusion, solange die makroskopische POVM wohldefiniert ist.<sup>17</sup>

## 8 Assumptions & Gültigkeitsbereich

- **Markov-Nähe:** Offene Dynamik ist GKSL-Markovian; Nicht-Markoveffekte sind nicht explizit modelliert (TI erlaubt dies, erfordert aber keine).
- **Coarse-Graining:** Pixel-POVMs approximieren makroskopische Pointer. Die konkrete Selektor-Geometrie ist ein *Modell* des Detektors.
- **Einheiten/Skalen:** Zeitskalierung --s (oder  $r$ ) harmonisiert Level;  $H = \gamma L$  setzt Energieskalen relativ zu Graph-Laplacian.
- **Numerik:** RK4-Integration, Exponentialauswertung und MCWF weisen übliche Diskretisierungs-/Samplingfehler auf; Statistik (TOST, Wilson) kontrolliert daraus resultierende Unsicherheiten.

## 9 Kurz zum Output (am Rande)

Die Pipeline erzeugt: Heat-Kernel-Plots,  $d_S$ -Fits, Sub-Gaussian-Relationen, sowie Tabellen/Plots für  $p^{(\text{TI})}$  vs.  $f$  (Divergenzen, GoF, CI/TOST). Diese sind prüfbar gegen obige Formalismen.<sup>18</sup>

<sup>17</sup>Effektiv-Widerstand und  $L^+$ : klassische Graphentheorie/Netzwerke [18, 19]. Sub-Gaussian Estimates [14, 17].

<sup>18</sup>Die zugehörigen Writer/Render-Routinen sind in der v14.6-Datei enthalten; Details siehe Code.

## Fazit

Die v14.6-Pipeline implementiert eine *kanonische* Kette:

$$\text{ST-Geometrie} \Rightarrow L, K_t, \rho_\beta \Rightarrow \text{POVM } \{E'_k\} \Rightarrow p^{(\text{TI})} = \text{Tr}(\rho E'_k),$$

und verifiziert mit GKSL/MCWF, dass die makroskopischen Klickstatistiken  $f$  die TI-Vorhersage innerhalb statistischer Toleranzen reproduzieren (GoF & TOST). Damit ist die *Emergenz der TI* auf dem ST-Graphen formal vollständig operationalisiert: q-Erwartungswerte (TI)  $\leftrightarrow$  Detektor-Anteile (POVM)  $\leftrightarrow$  beobachtete Häufigkeiten (MCWF).

## Literatur

- [1] A. Neumaier, *Foundations of Quantum Physics I. A critique of the tradition*, arXiv:1902.10745 (2019).
- [2] A. Neumaier, *Foundations of Quantum Physics II. The thermal interpretation of quantum physics*, arXiv:1902.10746 (2019).
- [3] A. Neumaier, *Foundations of Quantum Physics III. Measurement*, arXiv:1902.10780 (2019).
- [4] A. Neumaier, *Foundations of Quantum Physics IV. Quantum field theory*, arXiv:1902.10781 (2019).
- [5] A. Neumaier, *Foundations of Quantum Physics V. The thermal interpretation in a nutshell*, arXiv:1902.10812 (2019).
- [6] V. Gorini, A. Kossakowski, E.C.G. Sudarshan, *Completely positive dynamical semigroups of N-level systems*, J. Math. Phys. **17**, 821 (1976).
- [7] G. Lindblad, *On the generators of quantum dynamical semigroups*, Commun. Math. Phys. **48**, 119–130 (1976).
- [8] J. Dalibard, Y. Castin, K. Mølmer, *Wave-function approach to dissipative processes in quantum optics*, Phys. Rev. Lett. **68**, 580 (1992).
- [9] H. J. Carmichael, *An Open Systems Approach to Quantum Optics*, Springer (1993).
- [10] M. B. Plenio, P. L. Knight, *The quantum-jump approach to dissipative dynamics in quantum optics*, Rev. Mod. Phys. **70**, 101 (1998).
- [11] M. A. Nielsen, I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge Univ. Press (2000/2010).
- [12] A. S. Holevo, *Statistical Structure of Quantum Theory*, Springer (2001).
- [13] J. Kigami, *Analysis on Fractals*, Cambridge Univ. Press (2001).
- [14] J. Kigami, *Resistance forms, quasisymmetric maps and heat kernel estimates*, Mem. AMS **216** (2012).
- [15] M. T. Barlow, *Random Walks and Heat Kernels on Graphs*, Cambridge Univ. Press (2017).
- [16] T. Kumagai, J. Misumi, *Heat kernel estimates for strongly recurrent random walk on random media*, arXiv:0806.4507 (2008).
- [17] M. T. Barlow, R. F. Bass, T. Kumagai, *Parabolic Harnack inequality and heat kernel estimates for random walks with long range jumps*, 2008 preprint.

- [18] P. G. Doyle, J. L. Snell, *Random Walks and Electric Networks*, MAA (1984/2000).
- [19] D. J. Klein, M. Randić, *Resistance distance*, J. Math. Chem. **12**, 81–95 (1993).
- [20] S. Kullback, R. A. Leibler, *On Information and Sufficiency*, Ann. Math. Statist. **22**, 79–86 (1951).
- [21] J. Lin, *Divergence measures based on the Shannon entropy*, IEEE Trans. Info. Theory **37**(1), 145–151 (1991).
- [22] J. Briët, P. Harremoës, *Properties of Classical and Quantum Jensen–Shannon Divergence*, arXiv:0911.5073 (2009).
- [23] S. S. Wilks, *The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses*, Ann. Math. Statist. **9**, 60–62 (1938).
- [24] E. B. Wilson, *Probable Inference, the Law of Succession, and Statistical Inference*, JASA **22** (1927). (Allg. Darstellungen: NIST/ITL; vgl. [25]).
- [25] NIST/ITL Handbook, *Confidence intervals (Wilson method)*.
- [26] D. J. Schuirmann, *Two One-Sided Tests Procedure (TOST)* J. Pharmacokinet. Biopharm. **15**, 657–680 (1987).
- [27] D. Lakens, *Equivalence Tests: A Practical Primer*, Soc. Psych. Pers. Sci. **8**, 355–362 (2017).