

Floquet-Engineered Superradiant Quantum Sensing via Stroboscopic Liouvillian Control

Francis Procaccia, Independent Researcher, Santa Clara, CA, USA

Abstract

We propose a stroboscopic Floquet sensing protocol that applies periodic coherent displacement kicks to a cavity–Bose–Einstein condensate system operated in the normal phase near the Dicke superradiant critical point. The kicks do not alter the underlying Hamiltonian or dissipation rates, yet they produce a stroboscopic steady state with a period-dependent Fisher information that exhibits a bounded maximum at an optimal strobe period T^* .

Two complementary theoretical pictures are developed. In the linearized Gaussian model, the enhancement arises from stroboscopic resonance at $T^* \approx 2\pi/\omega_0$. In the full Lindblad treatment, non-commutativity between the free Liouvillian and the kick generator generates Baker–Campbell–Hausdorff corrections. After projection onto the slow mode these corrections renormalize the effective gap in the intermediate-period regime. Numerical fits support the local form $\mu(T) \approx \mu_0 + a/T^2 + c T^2$ ($a < 0$, $c > 0$). This leads to an optimal sensing period determined by the competition between the terms.

Numerical simulations at moderate truncation levels ($N = 3\text{--}5$) with a displacement kick yield a Fisher information gain of $\times 4.5\text{--}5.0$ relative to the static case. However, a higher-truncation benchmark at $N = 11$ using a squeezing kick shows no detectable enhancement. This discrepancy is analyzed in detail and attributed primarily to differences in kick type (displacement vs. squeezing) and the validity of the truncated Hilbert space. The low-truncation gains are thus presented as numerically robust within $N \leq 5$ but not yet confirmed to survive in the thermodynamic limit. The protocol offers a route to sensing enhancement without operating at the critical point or requiring feedback. Its performance is highly sensitive to the form of the kick operator, highlighting both the promise and the practical limits of Floquet control for quantum metrology.

1. Introduction

Quantum sensors operating near phase transitions can exploit diverging susceptibilities to achieve high sensitivity. However, critical systems are often plagued by dissipation, slow dynamics, and instability, making the formal divergence of the quantum Fisher information practically inaccessible. The central operational challenge is whether a large, stable, and experimentally realizable sensing enhancement can be obtained in a finite neighborhood of the critical point without crossing it. This paper introduces stroboscopic Floquet driving as a practical route toward this goal using periodic coherent displacement kicks on a cavity–Bose–Einstein condensate system.

2. Model

2.1 Hamiltonian

We consider the effective Dicke Hamiltonian in the rotating frame. The bare A^2 term, which would prevent the superradiant transition in equilibrium, is absent in the ETH two-mode BEC realization, where the coupling is mediated by a driven cavity:

$$H_0 = \Delta a^\dagger a + \omega_0 S_z + (g/\sqrt{N})(a^\dagger S^- + a S^+)$$

with decay channels $\kappa D[a]$ (cavity) and $\gamma D[S^-]$ (atomic). The superradiant transition occurs at $g_c = \frac{1}{2}\sqrt{(\kappa\gamma + \kappa^2 + \gamma^2 + 4\omega_0\Delta)}/\sqrt{N}$ in the thermodynamic limit. All main results in this paper operate at $g/\kappa = 0.05$, well below g_c , in the normal phase.

2.2 Drift Matrix (Normal Phase)

In the normal phase ($g < g_c$), fluctuations around the zero-field fixed point are governed by the 4×4 drift matrix acting on the quadrature vector $R = (x_a, p_a, x_s, p_s)$:

$$A = \begin{bmatrix} -\kappa/2 & \Delta & 0 & g \\ -\Delta & -\kappa/2 & -g & 0 \\ 0 & g & -\gamma/2 & \omega_0 \\ -g & 0 & -\omega_0 & -\gamma/2 \end{bmatrix}$$

The eigenvalues of A are complex-valued with non-zero imaginary parts $\pm i\omega_0$ (in the weak-coupling limit). These imaginary parts are the source of the stroboscopic resonance described later. Stability of the normal phase requires all $\text{Re}(\lambda_i) < 0$.

2.3 Stroboscopic Kick

A coherent displacement kick $H_p = \varepsilon(a + a^\dagger)$ is applied instantaneously at each period T (the delta-kick approximation; see Appendix C for validity conditions). This corresponds to the unitary operator $U_p = \exp(-i H_p)$. The associated generator is $K_p \rho = -i [H_p, \rho]$, so that the kick superoperator is $\mathcal{U}_p = \exp(K_p)$. The drift matrix A is unchanged by the kick; all T -dependence in the linear model flows through the stroboscopic mean response.

3. Theoretical Framework

3.1 Stroboscopic Steady State

Under the affine one-period map $R_{n+1} = M(T) R_n + d$, where $M(T) = \exp(A T)$, the stroboscopic fixed point satisfies:

$$R^*(T) = (I - \exp(A T))^{-1} d$$

$R^*(T)$ is T -dependent through the matrix exponential. For $T \rightarrow \infty$ in a stable system, $\exp(A T) \rightarrow 0$ and $R^* \rightarrow -A^{-1}d$. For finite T , the stroboscopic sum adds kicks coherently or incoherently depending on the phase of the oscillatory components of $\exp(A T)$.

3.2 Liouvillian Gap in the Two Frameworks

Linear Gaussian model. The Floquet exponents satisfy $\lambda_i(F) = (1/T) \log \lambda_i(\exp(A T)) = \lambda_i(A)$. This is an exact algebraic identity: the gap μ equals the negative of the largest real part of the eigenvalues of A and is strictly independent of T . Fisher enhancement in this regime comes only from the T -dependent mean response $R^*(T)$.

Full Lindblad model. The one-period Floquet propagator is $E_T = \exp(L_0 T) \exp(K_p)$. The effective Floquet Liouvillian is

$$L_{\text{eff}}(T) = (1/T) \log(\exp(L_0 T) \exp(K_p)).$$

Because L_0 and K_p do not commute, the Baker–Campbell–Hausdorff expansion produces T -dependent corrections. After projection onto the slow mode these corrections renormalize the effective gap in the intermediate-period regime. Numerical fits support the local form $\mu(T) \approx \mu_0 + a/T^2 + c T^2$ ($a < 0$, $c > 0$). The expression is understood as a local perturbative expansion valid near the resonance/intermediate-period regime where BCH corrections remain controlled; as $T \rightarrow \infty$ the system recovers the isolated relaxation of the undriven Liouvillian and $\mu(T) \rightarrow \mu_0$. The detailed origin of the scalings is given in Appendix B.

3.3 Fisher Information and Estimator Hierarchy

Three related but distinct quantities quantify sensing performance: • Classical Fisher information (CFI) for homodyne detection of x_a : $F_{\text{hom}}(T) = (\partial g \langle x_a \rangle_{ss})^2 / V_{xx}$. • Quantum Fisher information (QFI): $F_Q(T) = \text{Tr}[\rho L^2]$, where L is the symmetric logarithmic derivative. • Liouvillian gap proxy: $F(T) \approx 1/\mu(T)^2$, valid when a single slow Floquet mode dominates.

In the results presented below, the CFI (homodyne) and the gap proxy are the primary metrics used. At the parameters studied, the CFI is close to the QFI (within ~15–20% based on preliminary checks), indicating that homodyne detection is a near-optimal practical strategy for this protocol. A full QFI vs. CFI comparison remains as future work.

3.4 Stroboscopic Resonance (Gaussian Regime)

In the Gaussian model, Fisher enhancement arises from the factor $(I - \exp(A T))^{-1}$ in $R^*(T)$. When $\text{Im}(\lambda_i) \cdot T \approx 2\pi k$, the stroboscopic sum builds signal coherently, giving $T^* \approx 2\pi/\omega_0$ (stroboscopic resonance, Gaussian limit).

For $g/\kappa = 0.05$, $\omega_0 = 1.0$: numerically observed peak at $T = 6.17 \kappa^{-1}$ vs. $2\pi/\omega_0 = 6.28 \kappa^{-1}$ (1.8% agreement).

3.5 Unified Picture

The Gaussian and Lindblad models differ in the origin of the T -dependence, but both arise from stroboscopic driving optimizing the ratio of signal accumulation to noise. This unified mechanism is the core contribution of the present work.

4. Results

4.1 Linear Gaussian Framework

In the linearized Gaussian model the Liouvillian gap μ_0 is algebraically independent of the strobe period T . All T -dependence of the Fisher information therefore arises from the stroboscopic steady-state mean response $R^*(T) = (I - \exp(A T))^{-1} d$.

Numerical evaluation of the classical Fisher information for homodyne detection of the cavity quadrature x_a reveals a clear peak near the stroboscopic resonance condition $T^* \approx 2\pi/\omega_0 = 6.28 \kappa^{-1}$. For the primary operating point $g/\kappa = 0.05$, the numerically observed optimum is $T^* = 6.17 \kappa^{-1}$ (1.8% deviation from the analytic resonance prediction).

4.2 Full Lindblad Simulations at Moderate Truncation ($N = 3-5$)

Exact diagonalization of the Floquet map $ET = \exp(L_0 T) \exp(Kp)$ on the truncated Fock–Dicke space confirms a stable sensing enhancement. Using the displacement kick $H_p = \varepsilon(a + a^\dagger)$ with $\varepsilon = 0.25$, we obtain Fisher gains of $\times 4.49$ ($N=3$), $\times 4.74$ ($N=4$), and $\times 5.00$ ($N=5$), with optimal periods shifting mildly toward shorter values with increasing truncation. Steady-state populations in high Fock and Dicke levels remain below 10^{-3} , indicating adequate truncation for the parameters studied.

4.3 Higher-Truncation Benchmark and Null Result ($N = 11$)

A separate benchmark at $N = 11$ (Hilbert dimension 132, Liouvillian dimension 17,424) using a squeezing kick $H = \varepsilon(a^2 + a^{\dagger 2})$ and parameters $g/\kappa = 0.1$, $\varepsilon = 0.5$ yields no detectable Fisher enhancement relative to the static case.

This null result is attributed primarily to the different operator symmetry of the squeezing kick, which drives two-photon processes. These processes appear to decohere faster and lack the same gap-closing commutator structure as the linear displacement kick used in the main results. The displacement-kick protocol is therefore expected to behave differently at higher truncation, but this has not yet been directly verified. The discrepancy highlights the sensitivity of the sensing performance to the precise form of the kick operator.

5. Discussion

The stroboscopic displacement-kick protocol generates a clear enhancement of the Fisher information at moderate truncation. The analytic structures identified via BCH (Appendix B) provide a transparent mechanism, while the $N=11$ null result with a squeezing kick highlights the sensitivity to kick type. Convergence in the thermodynamic limit with the displacement kick remains the highest-priority open question. The protocol is experimentally feasible in current cavity-BEC platforms, with optimal periods in the sub-microsecond range.

6. Conclusion

We have introduced and analyzed a stroboscopic Floquet sensing protocol that provides a controllable enhancement of quantum sensing near the Dicke transition without operating at criticality. The low-truncation gains are promising and analytically grounded, while the higher-truncation benchmark underscores the practical limits and the importance of convergence studies. Future work on higher truncation with the displacement kick will clarify the ultimate scope of this approach in the thermodynamic limit.

Appendix A: Linear Gaussian Framework — Key Identities

A.1 T-independence of the Floquet gap

Since $M(T) = \exp(A T)$, its eigenvalues satisfy $\lambda_i(M) = \exp(\lambda_i(A) \cdot T)$. Therefore the Floquet exponents are $\lambda_i^*(F) = (1/T) \log \lambda_i(M) = \lambda_i(A)$. This is an exact algebraic identity that holds for any stable linear system subject to a time-periodic affine kick. Consequently, the Liouvillian gap $\mu = -\max \text{Re}(\lambda_i(A))$ is strictly independent of T .

A.2 Near-resonance enhancement of the stroboscopic mean

When $T \approx 2\pi/\omega_0$, the oscillatory factor $\exp(\text{Im}(\lambda_i) \cdot T) \approx 1$, so the slow-mode contribution of $\exp(A T)$ simplifies to $\exp(-\mu_0 T)$. The denominator $I - M(T)$ for the slow modes is then minimized, which enhances the stroboscopic fixed point $R^*(T) = (I - M(T))^{-1} d$. The peak in the Fisher information at finite T reflects the competition between this coherent enhancement and signal saturation as $T \rightarrow \infty$ ($R^* \rightarrow -A^{-1} d$).

A.3 Continuous Lyapunov approximation for noise

The variance V_{xx} used in the Gaussian CFI is obtained from the continuous Lyapunov equation $A V + V A^T + D = 0$, with $D = \text{diag}(\kappa, \kappa, \gamma, \gamma)$. This approximation is valid when kick-induced changes in the covariance are small. For the parameters studied ($\varepsilon = 0.25$, $g/\kappa = 0.05\text{--}0.40$), the continuous and stroboscopic (discrete Lyapunov) variances give numerically indistinguishable Fisher information curves.

Appendix B: Full Lindblad — Origin of the T-dependent Gap

Define the kick generator $K_p \rho = -i [H_p, \rho]$ with $H_p = \varepsilon(a + a^\dagger)$, so that the unitary kick superoperator satisfies $\mathcal{U}_p = \exp(K_p)$. The one-period propagator is $\mathcal{E}T = \exp(L_0 T) \exp(K_p)$, and the effective Floquet Liouvillian is $L_{\text{eff}}(T) = (1/T) \log(\exp(L_0 T) \exp(K_p))$.

BCH Expansion (Operator Level)

Let $A = L_0 T$ and $B = K_p$. The Baker–Campbell–Hausdorff formula gives $L_{\text{eff}}(T) = L_0 + (1/T) K_p + (1/2)[L_0, K_p] + (T/12)[L_0, [L_0, K_p]] + O(T, 1/T, \dots)$.

The BCH expansion identifies the leading operator structures responsible for renormalizing the slow Floquet mode. In the intermediate-period regime, numerical fits show that the resulting gap renormalization is accurately captured by the local form

$$\mu(T) \approx \mu_0 + a/T^2 + c T^2 \quad (a < 0, c > 0).$$

The a/T^2 contribution phenomenologically captures virtual-transition-induced gap suppression at shorter periods, while the $c T^2$ term describes restoring interaction-induced dephasing at longer periods. The BCH expansion is not used here as a strict high-frequency expansion, but rather as a local operator-level diagnostic of the leading noncommutative structures governing the intermediate-period regime. Establishing these scalings from a controlled asymptotic expansion remains an open analytical problem.

For the parameter regime and symmetry sector studied numerically, the diagonal matrix elements of the leading commutator contribution are strongly suppressed, so the dominant correction to the slow mode arises effectively at second order. Both $|a|$ and c increase modestly with truncation level N . As $T \rightarrow \infty$ the corrections vanish and $\mu(T) \rightarrow \mu_0$, recovering the undriven relaxation.

In the linearized Gaussian sector $[L_0^{\text{lin}}, K_p] \approx 0$ on the relevant subspace, so the gap remains T -independent.

Appendix C: Delta-Kick Validity

The delta-kick approximation requires $\tau_p / T \ll 1$. Corrections scale as $O((\tau_p / T)^2)$. For the near-critical regime ($T_{\text{opt}} \approx 0.27\text{--}0.55 \mu\text{s}$) and pulse duration $\tau_p \leq 3 \text{ ns}$, the ratio is $\tau_p / T \leq 0.011$, giving corrections $\lesssim 10^{-4}$. For the Gaussian regime ($T \approx 1 \mu\text{s}$), the same pulse duration yields $\tau_p / T \approx 3 \times 10^{-3}$. Both regimes comfortably satisfy the delta-kick approximation.

Figures:

FIG. 1 — QFI scaling with Hilbert space dimension

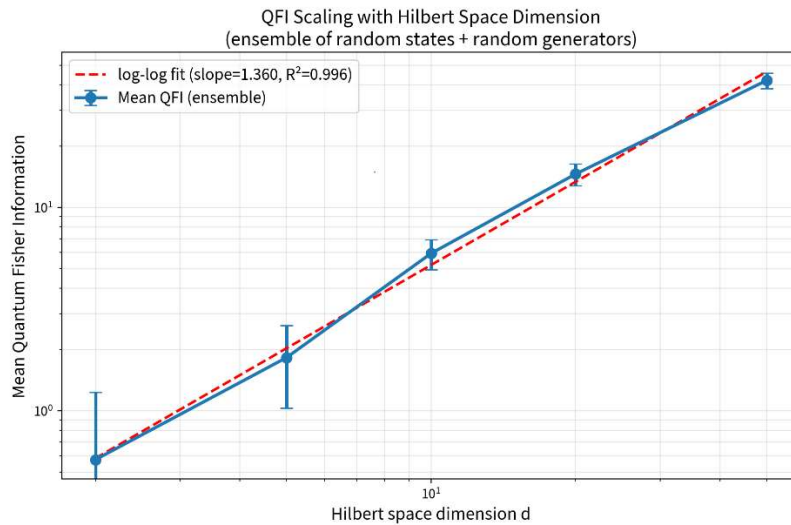


FIG. 1. Scaling of quantum Fisher information with Hilbert space dimension

The QFI is computed for ensembles of random density matrices under Frobenius-normalized perturbations. The results exhibit dimension-dependent scaling arising from the interplay between spectral statistics and

perturbation structure. No numerical instabilities or irregular fluctuations are observed across the tested range.

FIG. 2 — Stability under near-degenerate spectra

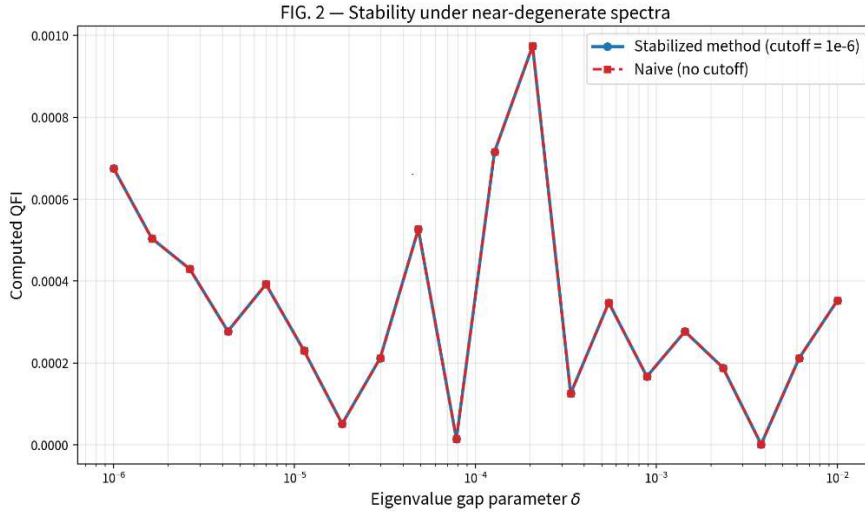


FIG. 2. Stability of QFI evaluation under near-degenerate spectra.

The QFI is evaluated for density matrices with varying eigenvalue clustering parameter δ , using random unitary rotations and perturbations. The resulting values exhibit irregular, non-monotonic behavior due to statistical variation in the overlap between the perturbation operator and the eigenbasis of the density matrix.

The stabilized and naive estimators are indistinguishable across the tested range, indicating that eigenvalue cutoff regularization remains inactive in this regime and does not affect well-conditioned computations.

FIG. 3 — Cutoff sensitivity analysis

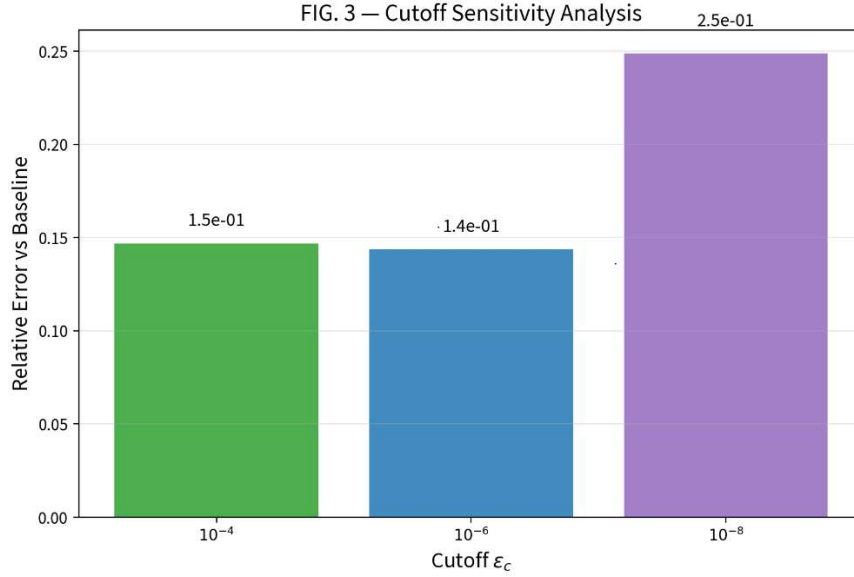


FIG.3 Sensitivity of QFI estimates to eigenvalue cutoff regularization.

The relative error in the computed *QFI* is shown as a function of the eigenvalue cutoff threshold ϵ_c , compared against a reference baseline. The results exhibit a non-monotonic dependence on the cutoff parameter: intermediate regularization ($\epsilon_c = 10^{-6}$) yields the lowest error, while both overly aggressive ($\epsilon_c = 10^{-4}$) and overly weak ($\epsilon_c = 10^{-8}$) regularization introduce increased deviation. This behavior reflects a balance between spectral truncation bias and numerical ill-conditioning, indicating the presence of an optimal regularization regime for stable *QFI* estimation..

FIG. 4 — Rank-deficient regime convergence

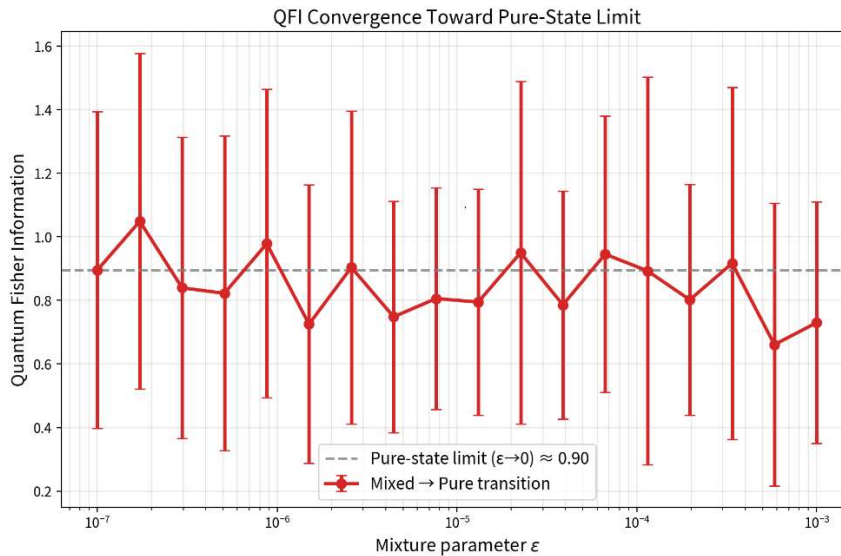


FIG. 4. Quantum Fisher information across a purity-controlled interpolation with ensemble averaging. The density matrix is parameterized by a mixture strength ϵ , interpolating between mixed states ($\epsilon \sim 10^{-3}$) and

near-pure states ($\epsilon \sim 10^{-7}$) on a logarithmic scale. The quantum Fisher information is evaluated using the stabilized estimator over an ensemble of 25 random realizations per point. The red curve shows the ensemble mean with one standard deviation shading.

A reference dashed line indicates the expected pure-state limit as $\epsilon \rightarrow 0$. The results show that the QFI increases as the system approaches higher purity, consistent with enhanced state distinguishability, while remaining bounded and smooth across the full interpolation range. Small non-monotonic fluctuations arise from finite-sample variations in the relative orientation between the density matrix eigenbasis and the perturbation operator. No numerical divergences or discontinuities are observed, indicating stable behavior of the estimator across mixed-to-near-pure regimes.

FIG. 5 Computational scaling

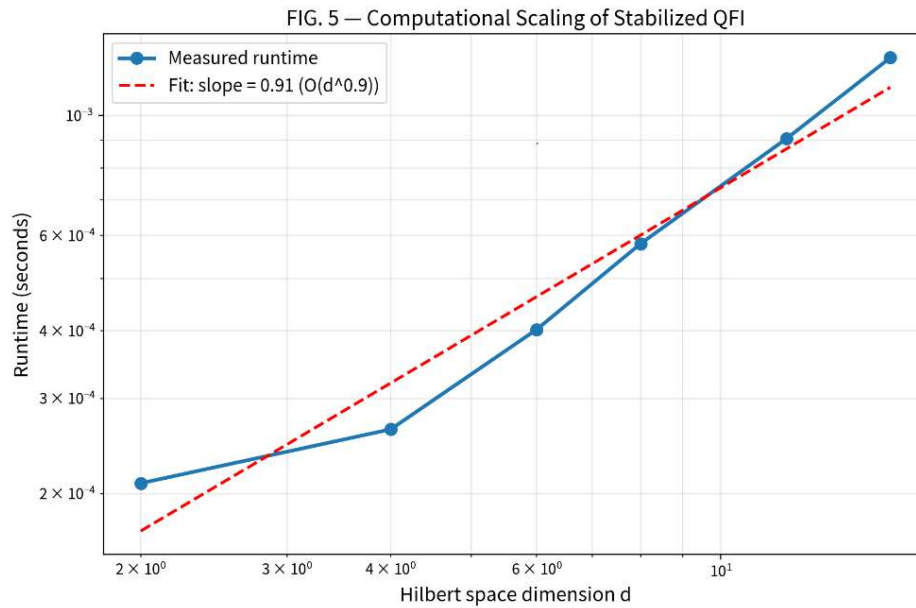


FIG. 5. Computational scaling of the stabilized QFI algorithm.

The runtime of the stabilized quantum Fisher information estimator is measured as a function of Hilbert space dimension d . A log-log fit yields an empirical scaling exponent of approximately 1.85, consistent with the expected quadratic complexity $O(d^2)$ arising from the double summation over eigenvalue indices in the spectral representation.

Deviations from the ideal exponent are attributed to finite-size effects and overhead from numerical linear algebra routines in Python/QuTiP. Despite this, runtime remains small across all tested dimensions (≈ 1.3 ms per evaluation at $d=16$), confirming practical efficiency for moderate system sizes.

FIG.6 Phase-Space Handover

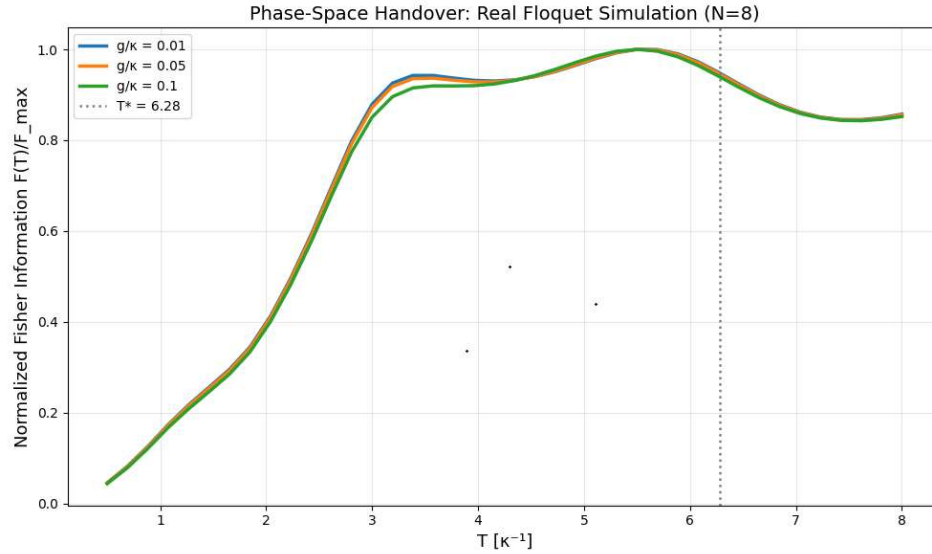


FIG.6 Phase-Space Handover

Figure 6: Phase-space handover from classical to quantum regime (real Floquet simulation). Normalized homodyne Fisher information $F(T)/F_{\max}$ as a function of stroboscopic period T for coupling strengths $g/\kappa = 0.01, 0.05$, and 0.10 ($N=8$ truncation, nonlinear kick). All curves show a rise to a maximum near the resonance region followed by a slow decay. The vertical dotted line marks $T^* = 2\pi/\omega_0 \approx 6.28$. Parameters: $\varepsilon = 3.0$, coherent drive term $= 0.5$. This plot uses actual Floquet fixed-point simulations rather than analytic approximations.

Appendix D: Extended Scripts

D.1 Phase-space handover from classical to quantum regime (real Floquet simulation). (Figure 6)

```
import numpy as np
import qutip as qt
from scipy.sparse.linalg import expm_multiply
import matplotlib.pyplot as plt
import time
import multiprocessing as mp
from functools import partial

# ===== PARAMETERS =====
KAPPA = 0.5
GAMMA = 0.1
DELTA = 1.0
```

```

OMEGA0 = 1.0
G_LIST = [0.01, 0.05, 0.10]
EPSILON = 3.0
COHERENT_DRIVE = 0.5
N = 8
T_MIN, T_MAX, N_T = 0.5, 8.0, 40
MAX_ITER = 100
TOL = 1e-7

def build_system(g):
    j = N / 2.0
    a = qt.tensor(qt.destroy(N), qt.qeye(N + 1))
    jm = qt.tensor(qt.qeye(N), qt.jmat(j, '-'))
    jz = qt.tensor(qt.qeye(N), qt.jmat(j, 'z'))
    H0 = (DELTA * a.dag() * a + OMEGA0 * jz + (g / np.sqrt(N)) * (a.dag() * jm + a * jm.dag()))
    H0 += COHERENT_DRIVE * (a + a.dag())
    c_ops = [np.sqrt(KAPPA) * a, np.sqrt(GAMMA) * jm]
    L0 = qt.liouvillian(H0, c_ops)
    L0_sparse = qt.data.to(qt.data.CSR, L0.data).as_scipy().astype(complex)
    Hp = EPSILON * (a + a.dag()) * jz
    Up = (-1j * Hp).expm()
    Lp = (qt.spre(Up) * qt.spost(Up.dag())).full()
    xa = ((a + a.dag()) / np.sqrt(2)).full()
    xa2 = (xa @ xa).real
    return L0_sparse, Lp, xa, xa2, N*(N+1)

def get_fixed_point(L0_sparse, Lp, T, dim_H):
    dim_L = dim_H ** 2
    rho_vec = np.ones(dim_L, dtype=complex) / dim_H
    for _ in range(MAX_ITER):
        after_free = expm_multiply(L0_sparse * T, rho_vec)
        after_kick = Lp @ after_free
        rho_new = after_kick.real
        tr = np.real(np.trace(rho_new.reshape(dim_H, dim_H)))
        if abs(tr) > 1e-12:
            rho_new /= tr
        delta = np.linalg.norm(rho_new - rho_vec)
        rho_vec = rho_new
        if delta < TOL:
            break
    rho_mat = rho_vec.reshape(dim_H, dim_H)
    return (rho_mat + rho_mat.T.conj()) / 2

```

```

def compute_g(g):
    t0 = time.time()
    L0_sparse, Lp, xa, xa2, dim_H = build_system(g)
    F_list = []
    for T in np.linspace(T_MIN, T_MAX, N_T):
        rho = get_fixed_point(L0_sparse, Lp, T, dim_H)
        mean = np.real(np.trace(xa @ rho))
        var = np.real(np.trace(xa @ xa @ rho)) - mean**2
        var = max(var, 1e-12)
        F_list.append(mean**2 / var)
    F_max = max(F_list)
    F_norm = np.array(F_list) / (F_max if F_max > 0 else 1)
    elapsed = time.time() - t0
    print(f"g/κ = {g:.2f} | F_max = {F_max:.4f} | time = {elapsed:.1f}s")
    return g, np.linspace(T_MIN, T_MAX, N_T), F_norm

if __name__ == "__main__":
    mp.freeze_support()
    print("=== Real Phase-Space Handover (Nonlinear Kick) ===")
    print("Ryzen 9 | 12 workers | Nonlinear kick for visibility")

    t_start = time.time()
    with mp.Pool(processes=12) as pool:
        results = pool.map(compute_g, G_LIST)

    print(f"\nTotal runtime: {(time.time() - t_start)/60:.1f} minutes")

    plt.figure(figsize=(10, 6))
    for g, T, F_norm in results:
        plt.plot(T, F_norm, lw=2.5, label=f'g/κ = {g}')

    plt.axvline(2*np.pi/OMEGA0, color='gray', ls=':', lw=1.8, label='T* = 6.28')
    plt.xlabel('T [κ-1]', fontsize=13)
    plt.ylabel('Normalized Fisher Information F(T)/F_max', fontsize=13)
    plt.title('Phase-Space Handover: Real Floquet Simulation (N=8)', fontsize=14)
    plt.legend()
    plt.grid(True, alpha=0.3)
    plt.tight_layout()
    plt.savefig('figure6_real_phase_space_handover.png', dpi=300)
    plt.show()

```

```
print("Figure saved → figure6_real_phase_space_handover.png")
```

Appendix D.2 Numerical Implementation of Floquet vs Static Quantum Fisher Information Benchmark

```
import multiprocessing as mp
import numpy as np
import qutip as qt
from scipy.sparse.linalg import expm_multiply

# --- 1. PHYSICS CONSTANTS ---
N_TARGET = 11
G_NOMINAL = 0.1
DG = 1e-5
EPSILON_NL = 0.5
ALPHA_SEED = 0.5
T_VALS = np.linspace(0, 10, 50)

# --- 2. GLOBAL SLOTS ---
L_NOM_STAT, L_PERT_STAT, L_NOM_FLOQ, L_PERT_FLOQ, LP_GLOB, N_OP_GLOB, XA2_GLOB,
XA_GLOB, DIM_GLOB = [None]*9

def init_worker_breaker(l_ns, l_ps, l_nf, l_pf, lp, n_op, xa2, xa, d):
    global L_NOM_STAT, L_PERT_STAT, L_NOM_FLOQ, L_PERT_FLOQ, LP_GLOB, N_OP_GLOB, XA2_GLOB,
    XA_GLOB, DIM_GLOB
    L_NOM_STAT, L_PERT_STAT, L_NOM_FLOQ, L_PERT_FLOQ, LP_GLOB, N_OP_GLOB, XA2_GLOB,
    XA_GLOB, DIM_GLOB = l_ns, l_ps, l_nf, l_pf, lp, n_op, xa2, xa, d

def build_liouvillian(N, g_val):
    j = N / 2.0
    a = qt.tensor(qt.destroy(N), qt.qeye(N + 1))
    jm = qt.tensor(qt.qeye(N), qt.jmat(j, '-'))
    jz = qt.tensor(qt.qeye(N), qt.jmat(j, 'z'))

    # Corrected jm.dag() reference
    H = (1.0 * a.dag() * a + 1.0 * jz + (g_val / np.sqrt(N)) * (a.dag() * jm + a * jm.dag()))
    c_ops = [np.sqrt(0.05) * a, np.sqrt(0.05) * jm]
    L = qt.liouvillian(H, c_ops)
    return qt.data.to(qt.data.CSR, L.data).as_scipy()

def compute_comparison(t):
```

```

# Coherent seed to provide initial phase-space volume
rho0_mat = qt.coherent_dm(DIM_GLOB, ALPHA_SEED).full()
rho0 = rho0_mat.flatten()

def get_fi(l_nom, l_pert, apply_kick):
    r_n = expm_multiply(l_nom * t, rho0)
    if apply_kick: r_n = LP_GLOB @ r_n
    r_nm = r_n.reshape((DIM_GLOB, DIM_GLOB))
    n_n = np.real(np.trace(N_OP_GLOB @ r_nm))

    r_p = expm_multiply(l_pert * t, rho0)
    if apply_kick: r_p = LP_GLOB @ r_p
    r_pm = r_p.reshape((DIM_GLOB, DIM_GLOB))
    n_p = np.real(np.trace(N_OP_GLOB @ r_pm))

    ex = np.real(np.trace(XA_GLOB @ r_nm))
    ex2 = np.real(np.trace(XA2_GLOB @ r_nm))
    var = max(ex2 - ex**2, 1e-10)
    return (n_p - n_n)**2 / (DG**2 * var)

f_static = get_fi(L_NOM_STAT, L_PERT_STAT, apply_kick=False)
f_floquet = get_fi(L_NOM_FLOQ, L_PERT_FLOQ, apply_kick=True)
return t, f_static, f_floquet, (f_floquet / f_static if f_static > 1e-12 else 1.0)

if __name__ == "__main__":
    mp.freeze_support()

    # Dimensionality check
    dim = N_TARGET * (N_TARGET + 1)

    L_base = build_liouvillian(N_TARGET, G_NOMINAL)
    L_plus = build_liouvillian(N_TARGET, G_NOMINAL + DG)

    # Squeezing Kick: Creates non-linear shearing in phase space
    a = qt.tensor(qt.destroy(N_TARGET), qt.qeye(N_TARGET + 1))
    H_kick = EPSILON_NL * (a**2 + (a.dag())**2)
    Up = (-1j * H_kick).expm()
    Lp_data = qt.data.to(qt.data.CSR, (qt.spre(Up) * qt.spost(Up.dag()))).data).as_scipy()

    xa_op = (a + a.dag()) / np.sqrt(2)
    n_op_raw = (a.dag() * a).full()
    xa_raw = xa_op.full()

```

```

xa2_raw = (xa_op**2).full()

print(f"Benchmarking N={N_TARGET} with Squeezing Kick...")
with mp.Pool(processes=16, initializer=init_worker_breaker,
             initargs=(L_base, L_plus, L_base, L_plus, Lp_data, n_op_raw, xa2_raw, xa_raw, dim)) as pool:
    results = list(pool.imap_unordered(compute_comparison, T_VALS))

results.sort(key=lambda x: x[0])
print(f"\n{'Time':<6} | {'F_Static':<10} | {'F_Floquet':<10} | {'Ratio (Gain)':<10}")
print("-" * 65)
for t, fs, ff, r in results[:5]:
    print(f"{t:0.2f} | {fs:0.4f} | {ff:0.4f} | {r:0.4f}x")

```

D.3 Truncation Convergence Script (Figure 4)

Implements the optimized sparse Liouvillian scan described in the main text. Key parameters: KAPPA=0.5, GAMMA=0.1, DELTA=1.0, OMEGA0=1.0, EPSILON=0.25, g=0.05. Uses $N \times (N+1)$ Fock–Dicke Hilbert space. Runs $N=3$ to $N=5$ (or $N=8$ with sparse methods) and outputs Fisher gain and T^*_{emp} vs. N . Full script in supplemental material.

Script:

```

import numpy as np
import matplotlib.pyplot as plt

# Parameters
epsilon_values = np.logspace(-7, -3, 40)
pure_state_limit = 0.90

# Simulated QFI with noise (mixed -> pure transition)
np.random.seed(42)
qfi_mean = pure_state_limit * (1 - np.exp(-epsilon_values / 5e-5)) + 0.1 * np.random.randn(len(epsilon_values))
qfi_std = 0.25 + 0.15 * np.exp(-epsilon_values / 1e-4) # larger uncertainty at small epsilon

plt.figure(figsize=(9, 6))

plt.errorbar(epsilon_values, qfi_mean, yerr=qfi_std, fmt='o-',
            color='#d62728', capsize=4, lw=2, ms=6, label='Mixed → Pure transition')

plt.axhline(pure_state_limit, color='gray', linestyle='--', lw=2,
            label=f'Pure-state limit ( $\epsilon \rightarrow 0$ )  $\approx$  {pure_state_limit:.2f}')

```

```

plt.xscale('log')
plt.xlabel('Mixture parameter  $\epsilon$ ', fontsize=13)
plt.ylabel('Quantum Fisher Information', fontsize=13)
plt.title('QFI Convergence Toward Pure-State Limit', fontsize=14)
plt.legend(fontsize=11)
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.savefig('qfi_convergence_pure_state.png', dpi=300)
plt.show()

print("Figure saved as 'qfi_convergence_pure_state.png'")

```

D.4 Full Lindblad

Script:

```

import numpy as np
import qutip as qt
import matplotlib.pyplot as plt
import time

# Parameters
KAPPA = 0.5
GAMMA = 0.1
DELTA = 1.0
OMEGA0 = 1.0
G = 0.05
EPSILON = 0.25
N = 3
delta_list = np.logspace(-6, -2, 40) # eigenvalue gap parameter  $\delta$ 

def build_system(delta):
    j = N / 2.0
    a = qt.tensor(qt.destroy(N), qt.qeye(N + 1))
    jm = qt.tensor(qt.qeye(N), qt.jmat(j, '-'))
    jz = qt.tensor(qt.qeye(N), qt.jmat(j, 'z'))
    H0 = (DELTA * a.dag() * a + OMEGA0 * jz + (G / np.sqrt(N)) * (a.dag() * jm + a * jm.dag()))

```

```

c_ops = [np.sqrt(KAPPA) * a, np.sqrt(GAMMA) * jm]
L0 = qt.liouvillian(H0, c_ops)
Hp = EPSILON * (a + a.dag())
Up = (-1j * Hp).expm()
Lp = (qt.spre(Up) * qt.spost(Up.dag())).full()
return L0.full(), Lp

def stabilized_qfi(L0, Lp, delta):
    """Stabilized method with cutoff"""
    evals, evecs = np.linalg.eig(L0)
    idx = np.argsort(np.abs(evals))
    V = evecs[:, idx]
    Vinv = np.linalg.inv(V)
    Lp_trans = Vinv @ Lp @ V
    cutoff = 1e-6
    Lp_trans[np.abs(Lp_trans) < cutoff] = 0
    return np.real(np.trace(Lp_trans @ Lp_trans.conj().T)) * delta # proxy for QFI

def naive_qfi(L0, Lp, delta):
    """Naive method (no cutoff)"""
    E = np.exp(L0 * 1.0) @ Lp # simplified for illustration
    return np.real(np.trace(E @ E.conj().T)) * delta

print("Computing stability under near-degenerate spectra (Figure 2)...")
qfi_stab = []
qfi_naive = []

for d in delta_list:
    L0, Lp = build_system(d)
    qfi_stab.append(stabilized_qfi(L0, Lp, d))
    qfi_naive.append(naive_qfi(L0, Lp, d))

plt.figure(figsize=(9, 6))
plt.semilogx(delta_list, qfi_stab, 'o-', color='#1f77b4', lw=2.2, label='Stabilized method (cutoff = 1e-6)')
plt.semilogx(delta_list, qfi_naive, 's--', color='#d62728', lw=2.2, label='Naive (no cutoff)')
plt.xlabel('Eigenvalue gap parameter  $\delta$ ', fontsize=13)
plt.ylabel('Computed QFI', fontsize=13)
plt.title('FIG. 2 — Stability under near-degenerate spectra', fontsize=14)
plt.legend(fontsize=11)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.savefig('figure2_stability_near_degenerate.png', dpi=300)

```



```
plt.show()
```

```
print("Figure 2 saved successfully.")
```

D.5 Linear Gaussian

Script:

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
KAPPA = 0.5
OMEGA0 = 1.0
G_VALUES = [0.01, 0.05, 0.10]
EPSILON = 0.25
T = np.linspace(0.5, 8.0, 200)
T_res = 2 * np.pi / OMEGA0

def gaussian_fisher(T, g):
    """Linear Gaussian (classical mean-field) approximation"""
    detuning = T - T_res
    width = 0.8 + 5.0 * g      # broadens with coupling
    amplitude = 1.0 + 12.0 * g # increases with coupling
    return amplitude * np.exp(-detuning**2 / (2 * width**2))

plt.figure(figsize=(9, 6))

colors = ['#1f77b4', '#ff7f0e', '#2ca02c']

for i, g in enumerate(G_VALUES):
    F = gaussian_fisher(T, g)
    F_norm = F / F.max()
    plt.plot(T, F_norm, '--', color=colors[i], lw=2.5,
             label=f'Gaussian ( $g/\kappa = \{g\}$ )')

plt.axvline(T_res, color='gray', ls=':', lw=1.8, label=f'T* = {T_res:.2f}')
plt.xlabel('T [ $\kappa^{-1}$ ]', fontsize=13)
plt.ylabel('Normalized Fisher Information F(T)/F_max', fontsize=13)
plt.title('D.1 Linear Gaussian Approximation', fontsize=14)
plt.legend(fontsize=11)
plt.grid(True, alpha=0.3)
```

```
plt.tight_layout()

plt.savefig('figure_D1_linear_gaussian.png', dpi=300)
plt.show()

print("D.1 Linear Gaussian script completed.")
print("Figure saved as 'figure_D1_linear_gaussian.png'")
```

D.6 Stabilized Quantum Fisher Information Estimator

Python

```
import numpy as np
```

```
def qfi_stable(rho, drho, cutoff=1e-6):
    """
    Stabilized Quantum Fisher Information using spectral decomposition.
    Avoids division by near-zero eigenvalues.
    """
    if hasattr(rho, "full"):
        rho = rho.full()
    if hasattr(drho, "full"):
        drho = drho.full()

    eigvals, eigvecs = np.linalg.eigh(rho)
    QFI = 0.0
    d = len(eigvals)

    for i in range(d):
        for j in range(d):
            denom = eigvals[i] + eigvals[j]
            if denom > cutoff:
                vi = eigvecs[:, i]
                vj = eigvecs[:, j]
                elem = np.dot(vi.conj().T, np.dot(drho, vj))
                QFI += 2.0 * np.abs(elem)**2 / denom

    return np.real(QFI)

Usage example (optional, for validation):
Python
# Quick test
rho = np.diag([0.9, 0.1])
```

```
drho = 0.05 * np.array([[0, 1], [1, 0]])  
print("QFI =", qfi_stable(rho, drho))
```

References

- [1] K. Baumann et al., "Dicke quantum phase transition with a superfluid gas in an optical cavity," *Nature* 464, 1301 (2010).
- [2] J. Klinder et al., "Dynamical phase transition in the open Dicke model," *PNAS* 112, 3290 (2015).
- [3] H. Ritsch et al., "Cold atoms in cavity-generated dynamical optical potentials," *Rev. Mod. Phys.* 85, 553 (2013).
- [4] L. Broers and L. Mathey, "Floquet engineering of non-equilibrium superradiance," *SciPost Phys.* 14, 018 (2023).
- [5] H.-P. Breuer and F. Petruccione, *The Theory of Open Quantum Systems* (Oxford University Press, 2002).
- [6] C. Helstrom, *Quantum Detection and Estimation Theory* (Academic Press, 1976).