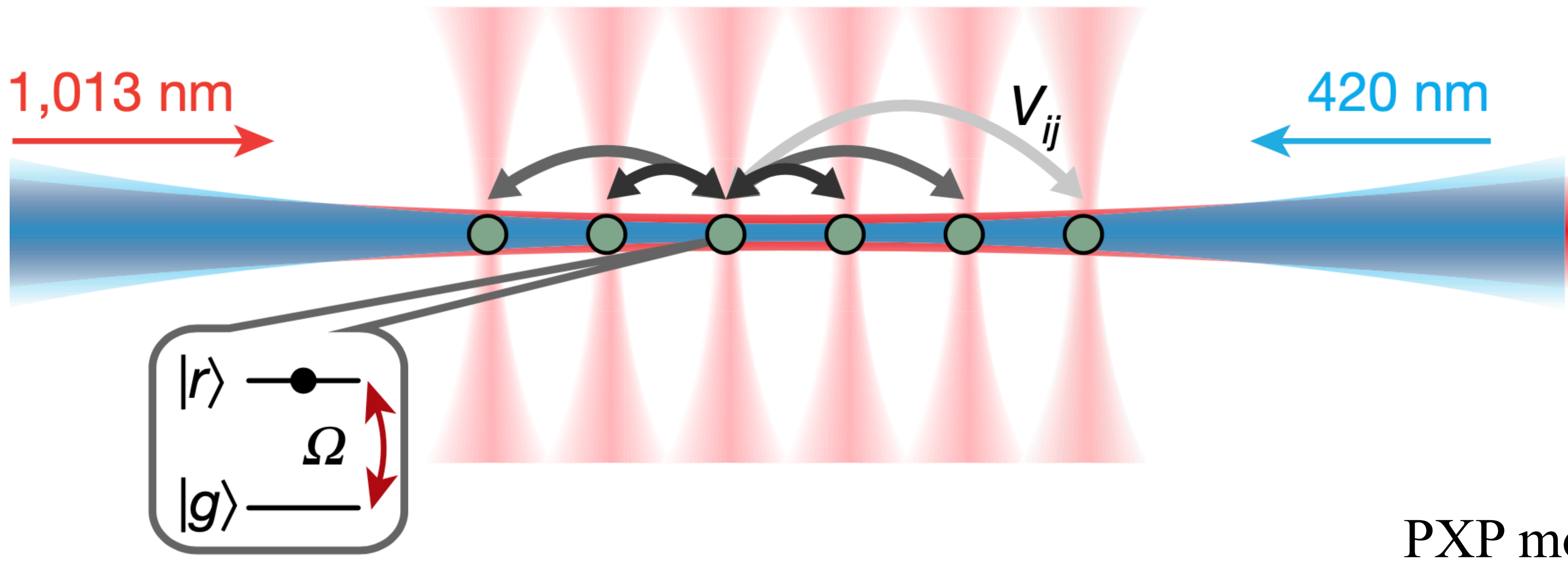
Differential Quantum Control

Optimizing driven many-body scar dynamics

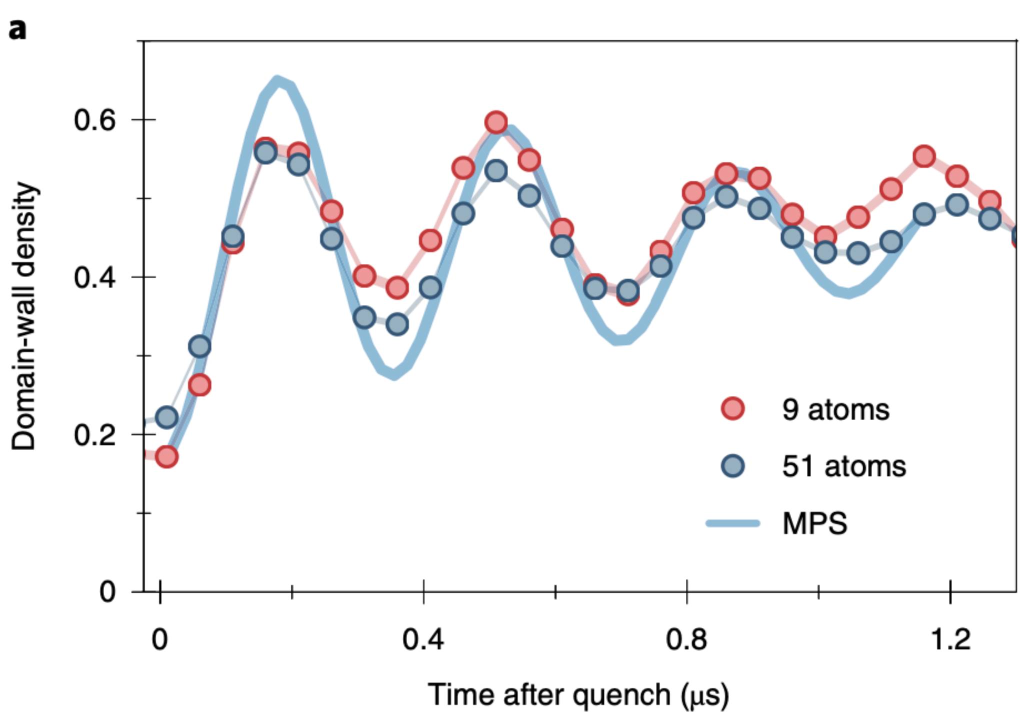
Jie Ren & Yu-Peng Wang

Rydberg Atom Experiment



PXP model

$$\hat{H} = \sum_i \frac{\Omega}{2}\hat{\sigma}_i^x - \sum_i \Delta \hat{n}_i + \sum_{i<j} V_{ij}\hat{n}_i\hat{n}_j \xrightarrow{\Delta = 0,\ V \gg \Omega} \boxed{\hat{H}_c = \sum_i \hat{P}^0_{i-1}\hat{X}_i\hat{P}^0_{i+1}}$$



It is experimentally observed that starting from a product state:

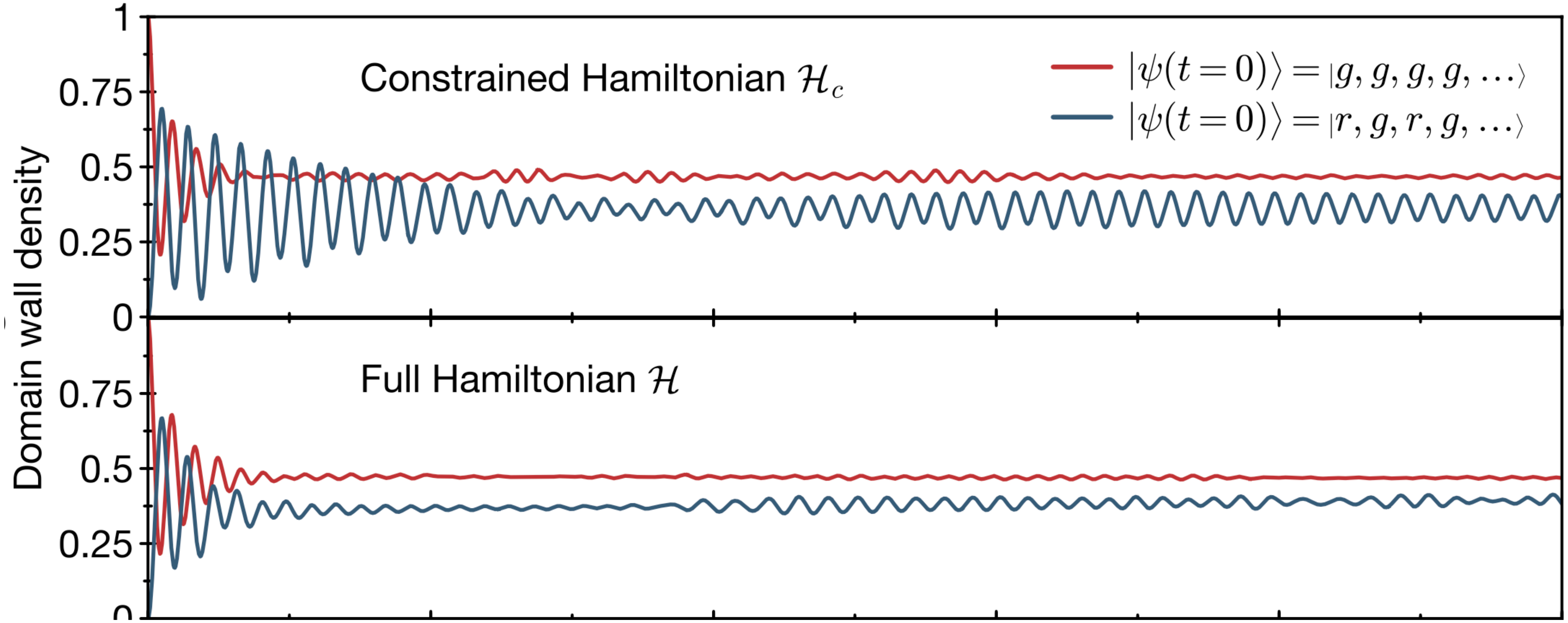
$$|Z_2\rangle = |\uparrow\downarrow\cdots\uparrow\downarrow\rangle,$$

the many-body evolution features an approximate revival period.

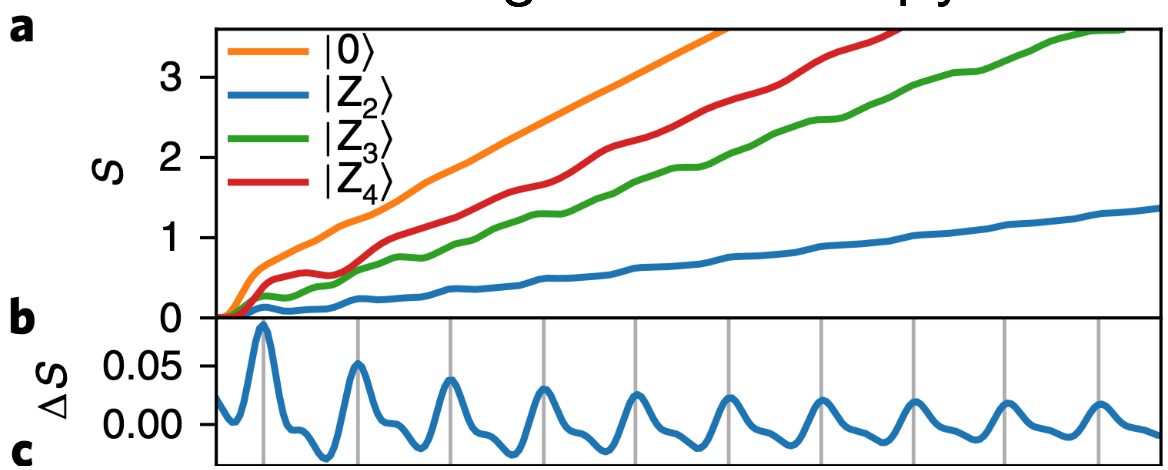
$$|Z_2\rangle = |\uparrow\downarrow\cdots\uparrow\downarrow\rangle,$$

Experiment & Numerics

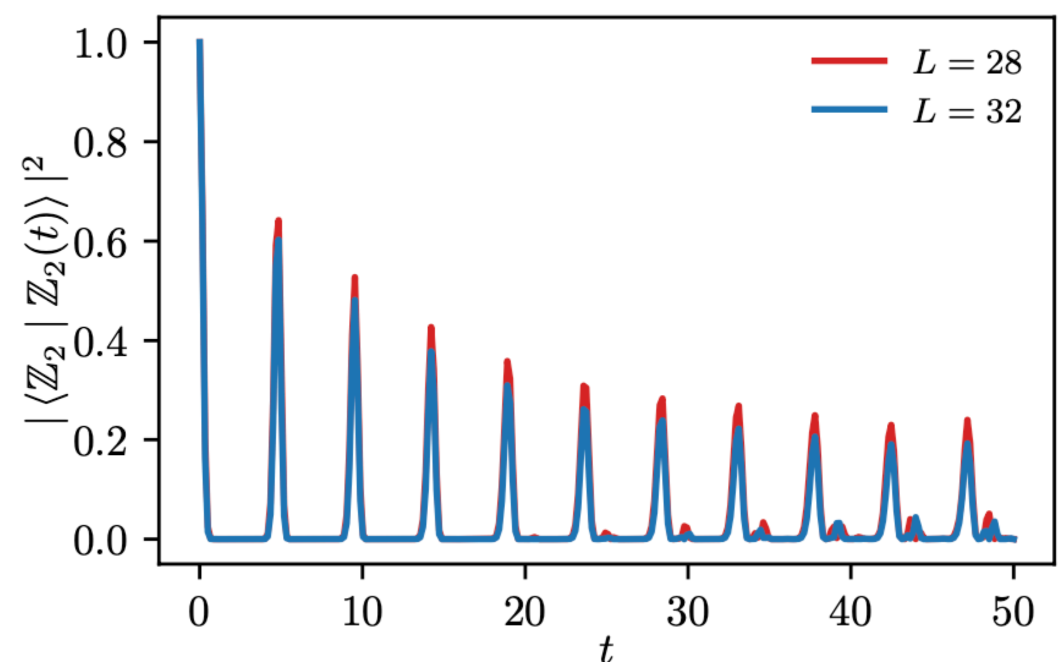
Local observable



Entanglement entropy



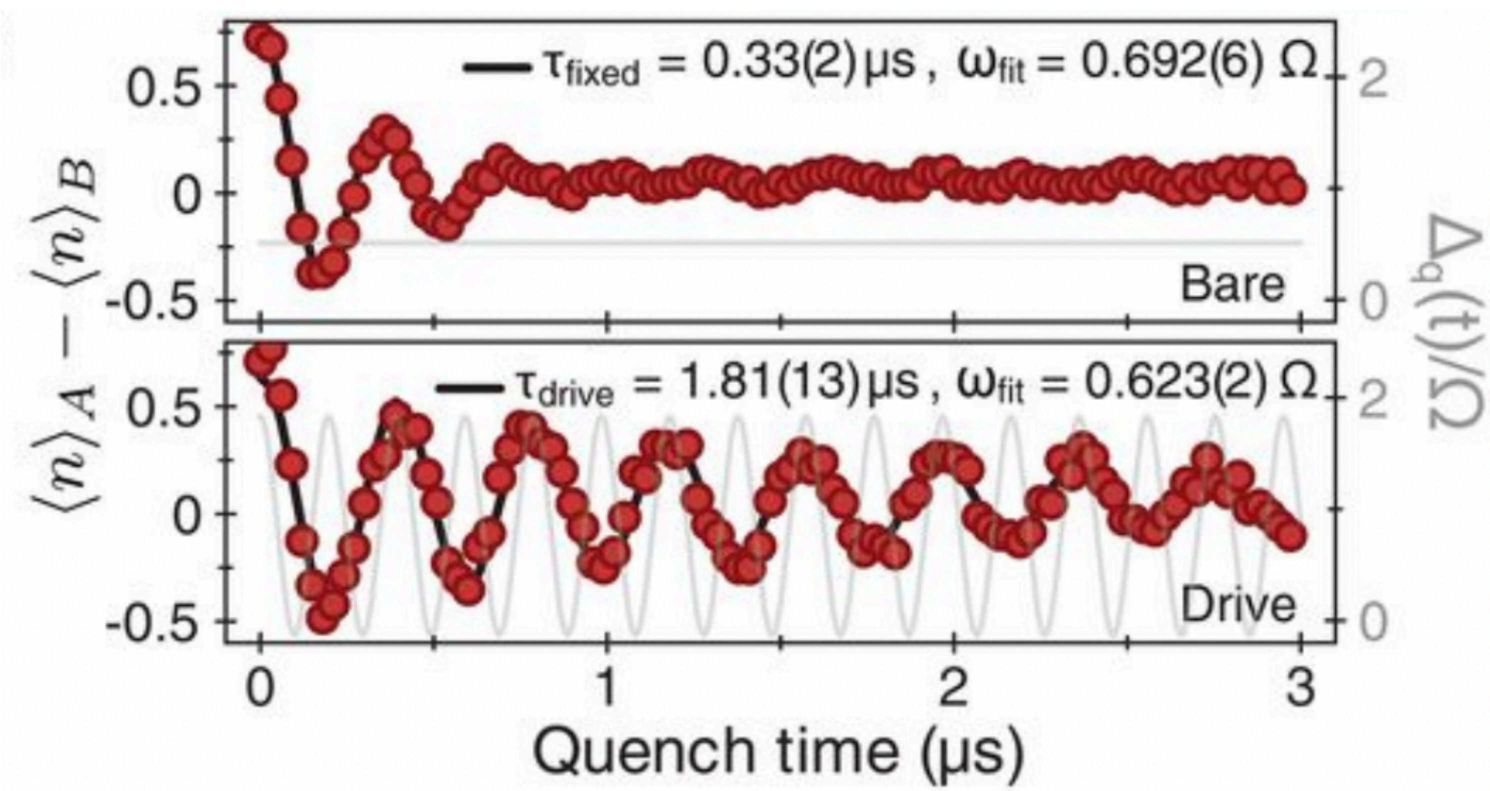
Fidelity



Driven PXP model

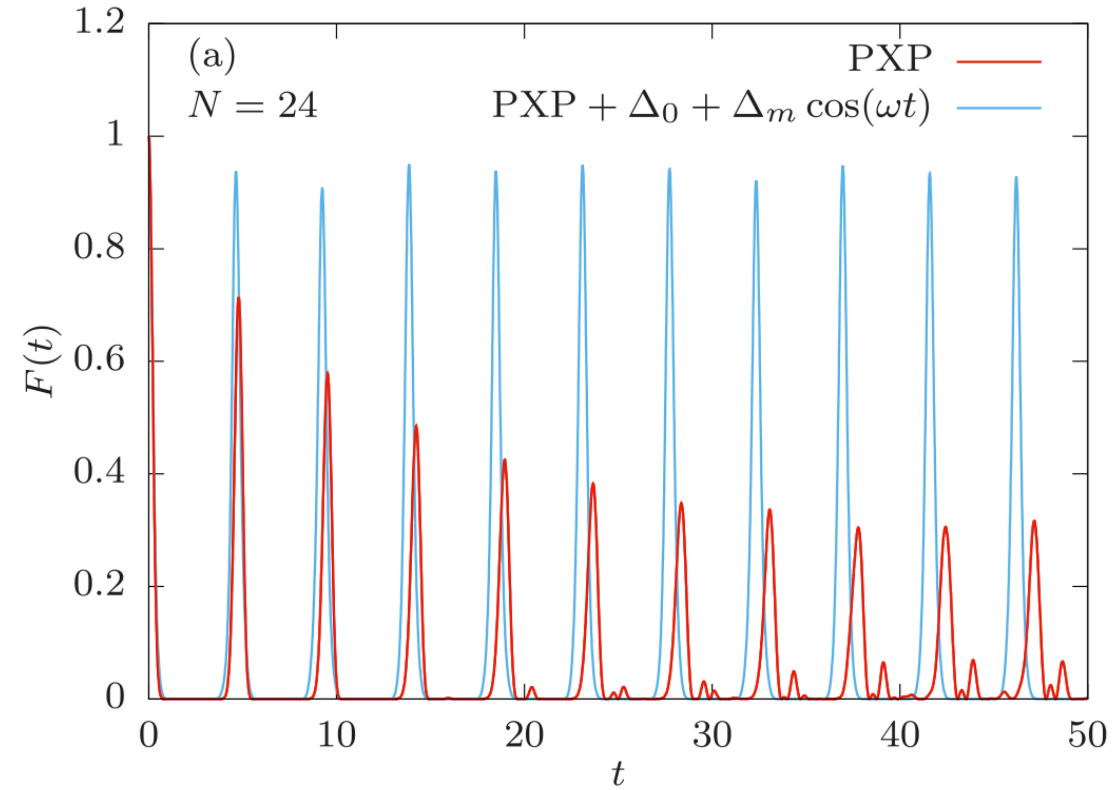
$$H(t) = H_{\text{PXP}} - \Delta(t) \sum_i n_i \qquad \Delta(t) = \Delta_0 + \Delta_m \cos(\omega t)$$

Experiment



$$\omega_{\text{drive}} = 2\omega_{\text{revival}}$$

Numerics



Optimal parameters:

Δ_0	Δ_m	ω
1.15	2.67	2.72

Optimize Driving

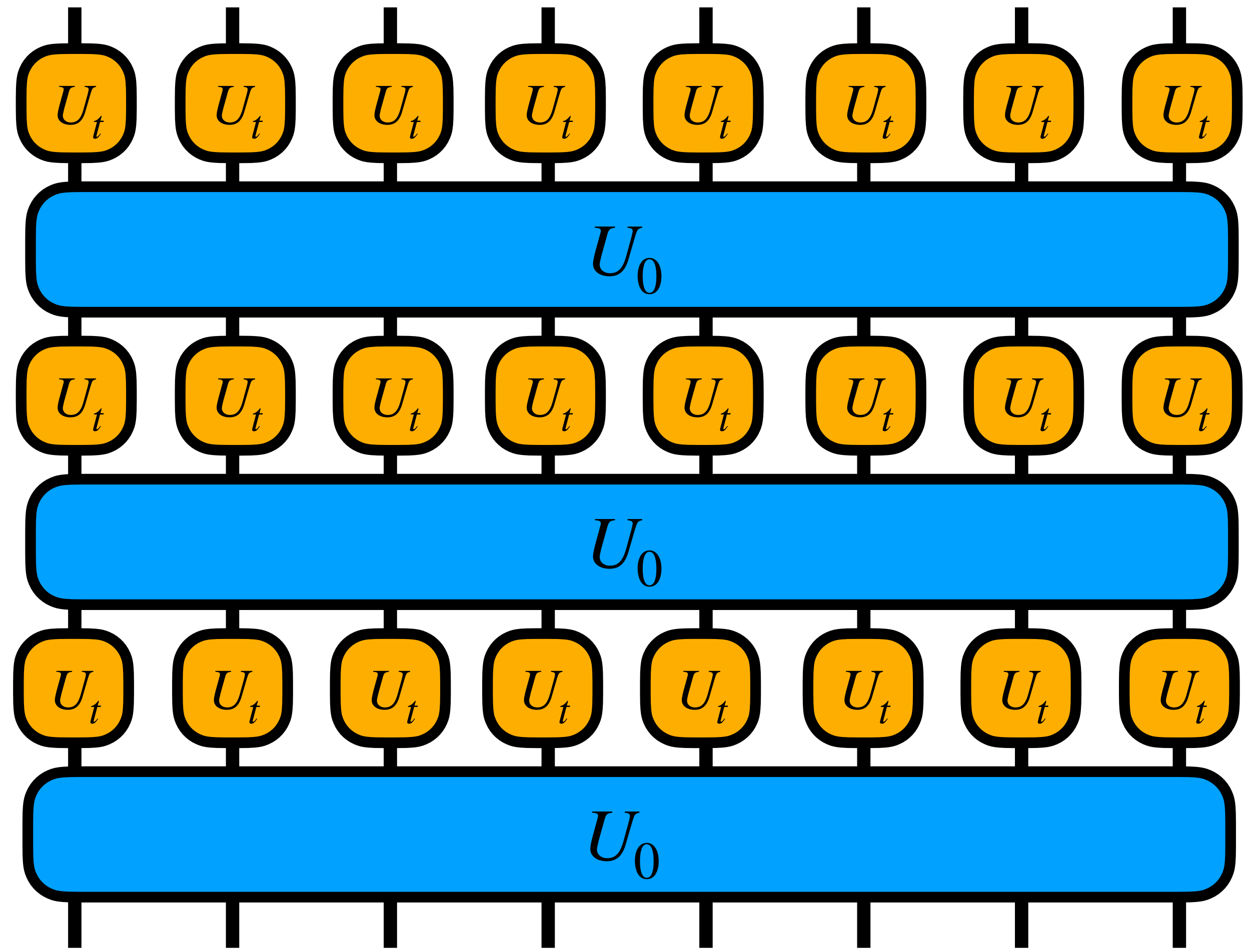
$$H(t) = H_{\mathrm{PXP}} - \Delta(t)\sum_i n_i$$

$$\Delta(t) = \Delta_0 + \sum_n \Delta_n \cos(n\omega t)$$

Goal: optimize the parameters $\{\omega, \Delta_n\}$, so the Floquet dynamics become nearly revival for the $|Z_2\rangle$ initial state.

To simulate the many-body dynamics, we use the Trotter decomposition:

Trotterized Quantum Circuit



$$U(\tau) = e^{-i\Delta(t)\sum_i n_i \Delta t} e^{-iH_{PXP}\Delta t} \cdots e^{-i\Delta(t)\sum_i n_i \Delta t} e^{-iH_{PXP}\Delta t}$$

Also, using the local constraint, plus the translation and reflection symmetry, we are able to reduce the Hilbert space dimension to ~400 for $L = 20$ system.

Implementing Details

```julia
function fidelity(params, T=100)
    v1, v2 = V01, V02
    N = round(Int, T/dt)
    map(1:N) do i
        f = params[2]
        f += params[3] * cos(params[1] * i * dt)
        f += params[4] * cos(params[1] * 2i * dt)
        f += params[5] * cos(params[1] * 3i * dt)
        f += params[6] * cos(params[1] * 4i * dt)
        v1 = exp.(-f*SN1*dt*1im) .* (evo1 * v1)
        v2 = exp.(-f*SN2*dt*1im) .* (evo2 * v2)
        abs(dot(V01, v1)+dot(V02, v2))^2 / 4
    end
end
```

For a set of discrete times, we calculate the fidelity

$$F(t) = \langle Z_2 | U(\tau) | Z_2 \rangle.$$

The cost function is defined as the averaged fidelity difference between even and odd periods:

```julia
T=100
function loss(params)
    period = 2π/params[1]
    F = fidelity(params, T)
    M = floor(Int, T/(2*period))
    out = 1.0
    for i in 1:M
        j1 = round(Int, 2i*period/dt)
        j2 = round(Int, (2i-1)*period/dt)
        out -= (F[j1] - F[j2])/M
    end
    out
end
```

The cost function is defined as the average fidelity difference between even and odd periods:

$$\mathscr{L}[\{\omega, \Delta_n\}] = 1 - \frac{1}{M}\sum_{i=1}^{M}[F(2nT) - F((2n-1)nT)],$$

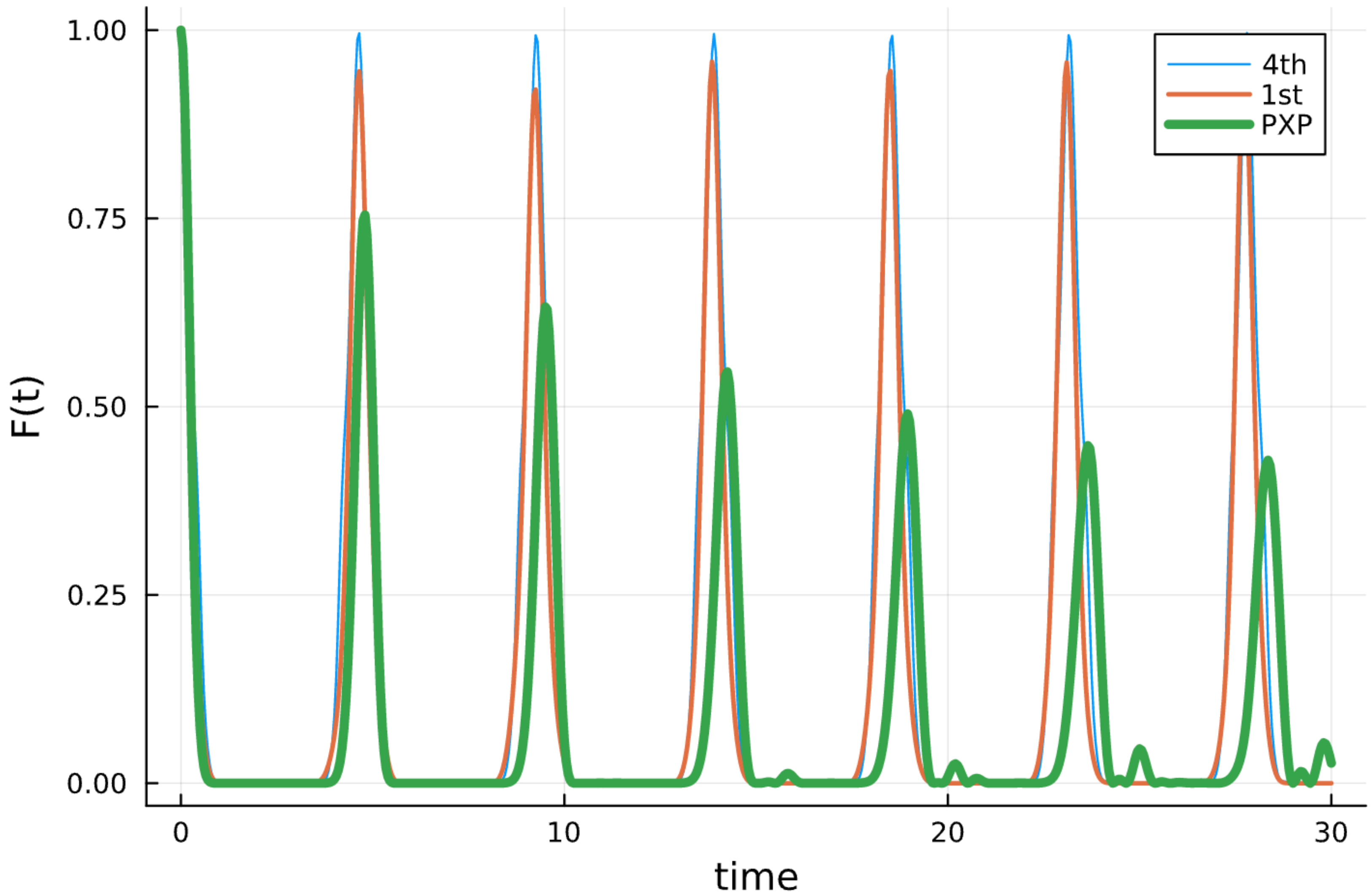
Where $T = 2\pi/\omega$.

```julia
init_p = [2.72, 1.15, 2.67, 0., 0., 0.]
res = optimize(loss, init_p, BFGS(); autodiff = :forward)
```

We use the Julia package Optim.jl to optimize the loss function. The Optim.jl package support forward auto differentiation.

Results

Using the 4th-order driving frequency, we are able to obtain better revival dynamics than that in the literature.



$$\begin{aligned} \omega &= 2.712188250074131, \\ \Delta_0 &= 1.244658202348191, \\ \Delta_1 &= 3.0357165151222536, \\ \Delta_2 &= 1.628636337215439, \\ \Delta_3 &= 5.04015853467762, \\ \Delta_4 &= -6.719927850497436. \end{aligned}$$

$$\mathscr{L}_{1st} = 5.736 \times 10^{-2}$$

$$\mathscr{L}_{4th} = 5.601 \times 10^{-3}$$