Q-Flow: Generative Modeling for Differential Equations of Open Quantum Dynamics with Normalizing Flows

Owen Dugan[1,5], Peter Y. Lu[2,5], Rumen Dangovski[3,5], Di Luo*[4,5,6], Marin Soljačić[1,5]

[1]MIT Physics, [2]Data Science Institute UChicago, [3]MIT EECS, [4]MIT CSAIL, [5]IAIFI, [6]Harvard Physics

**corresponding author:* diluo@mit.edu

Outline

1. Overview of Problem: Open Quantum Systems
2. Q-Flow Overview
3. Q-Function Details
4. Q-Flow Method: Modeling and Evolving Q
5. Results: Dissipative Bose-Hubbard
6. Results: Dissipative Harmonic Oscillator

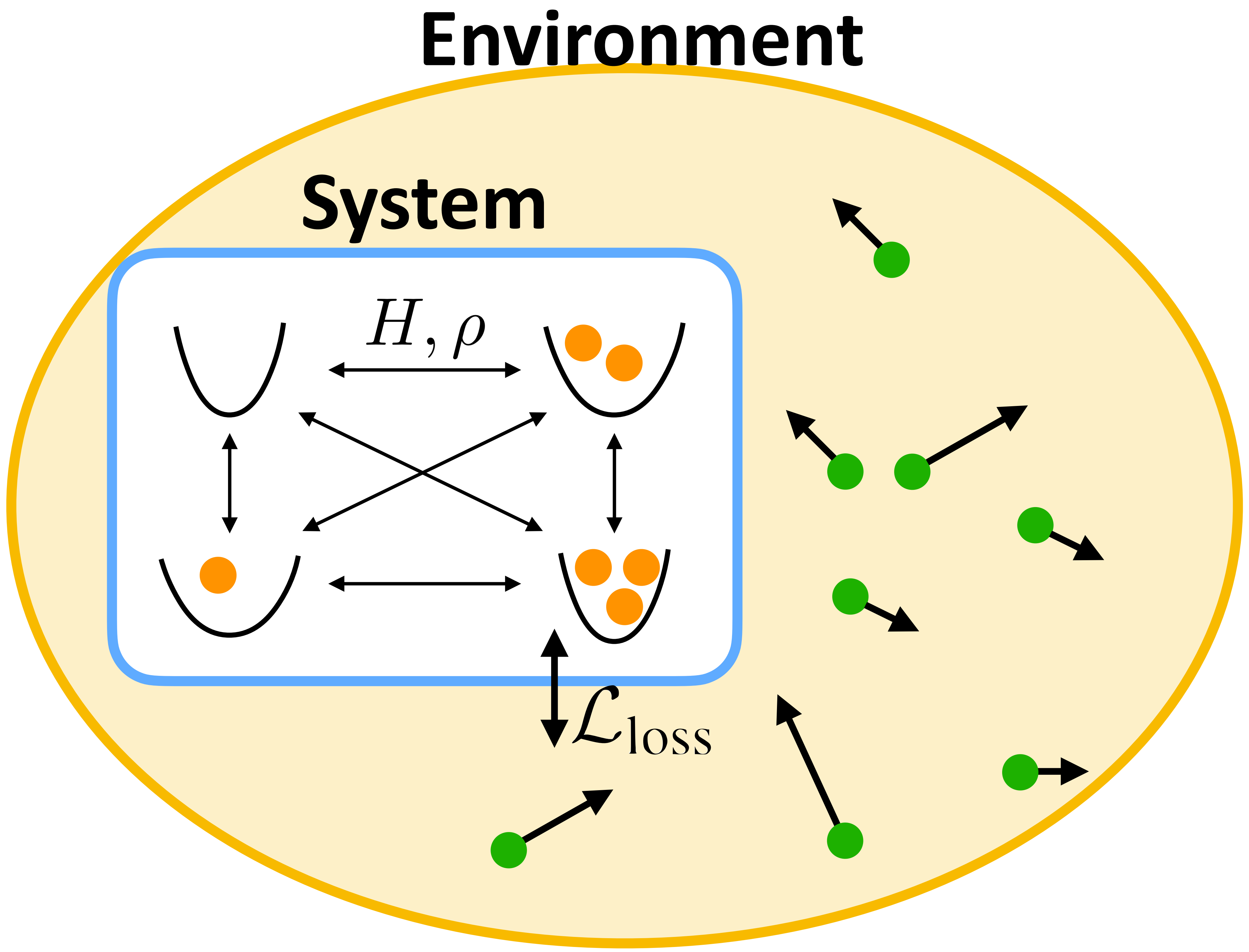
Overview of Problem: Open Quantum Systems

Open Quantum Systems

Definition: any quantum system that interacts with the environment

Applications:

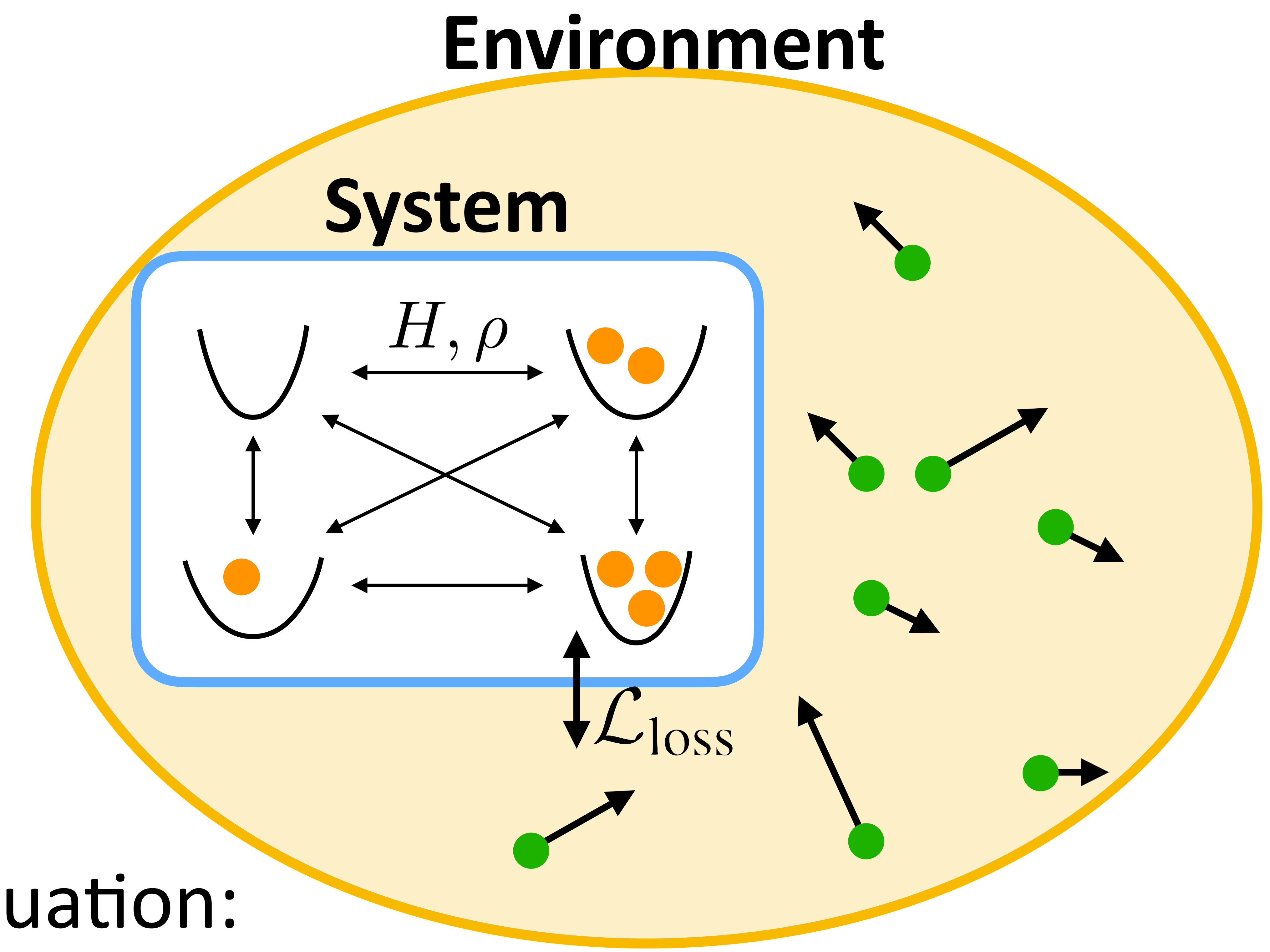
- Fundamental science
  - Phase transitions
- Quantum technology
  - Quantum computers
  - Superconductors



Simulating Open Quantum Systems (OQS)

Density matrix ρ:

- Represents the state of an OQS
- Square complex-valued matrix
- $\mathrm{Tr}(\rho) = 1$



Equation of Motion (EOM):

- ρ evolves according to the differential equation:

$$\dot{\rho} = \mathcal{L}\rho = -i[H, \rho] + \mathcal{L}_{\mathrm{loss}}\rho$$

The "Liouvillian" superoperator

System Hamiltonian

Loss superoperator

Challenges Simulating Open Quantum Systems

Curse of Dimensionality:

- Density matrix exponential in system size

Large systems infeasible for conventional solvers

Challenging Parametrization:

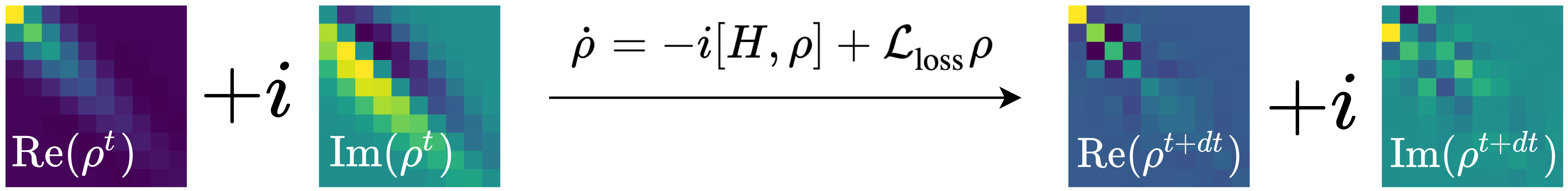
- ρ is complex valued and $\mathrm{Tr}(\rho) = 1$
- Complex dynamics and interactions

Difficult to model with Neural Networks

Past neural approaches limited to spin systems!

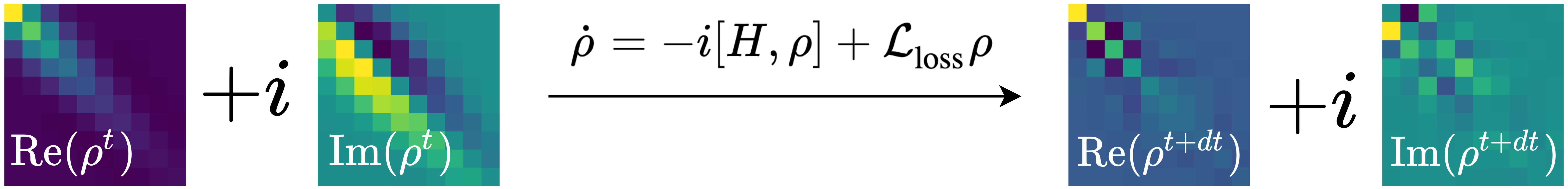
Q-Flow Overview

Q-Flow Procedure



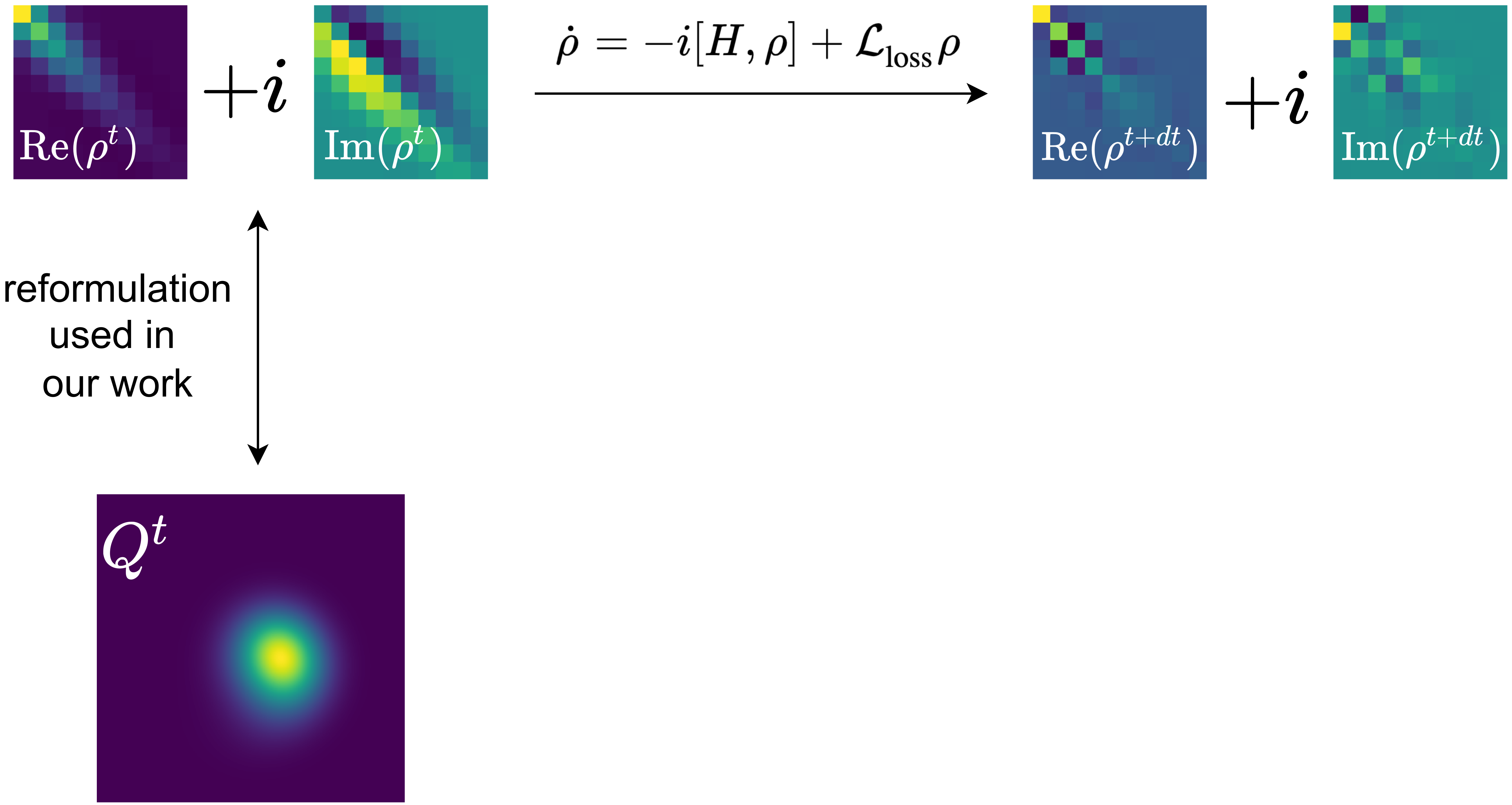
Q-Flow Procedure

1. Convert EOM and ρ to Q Function



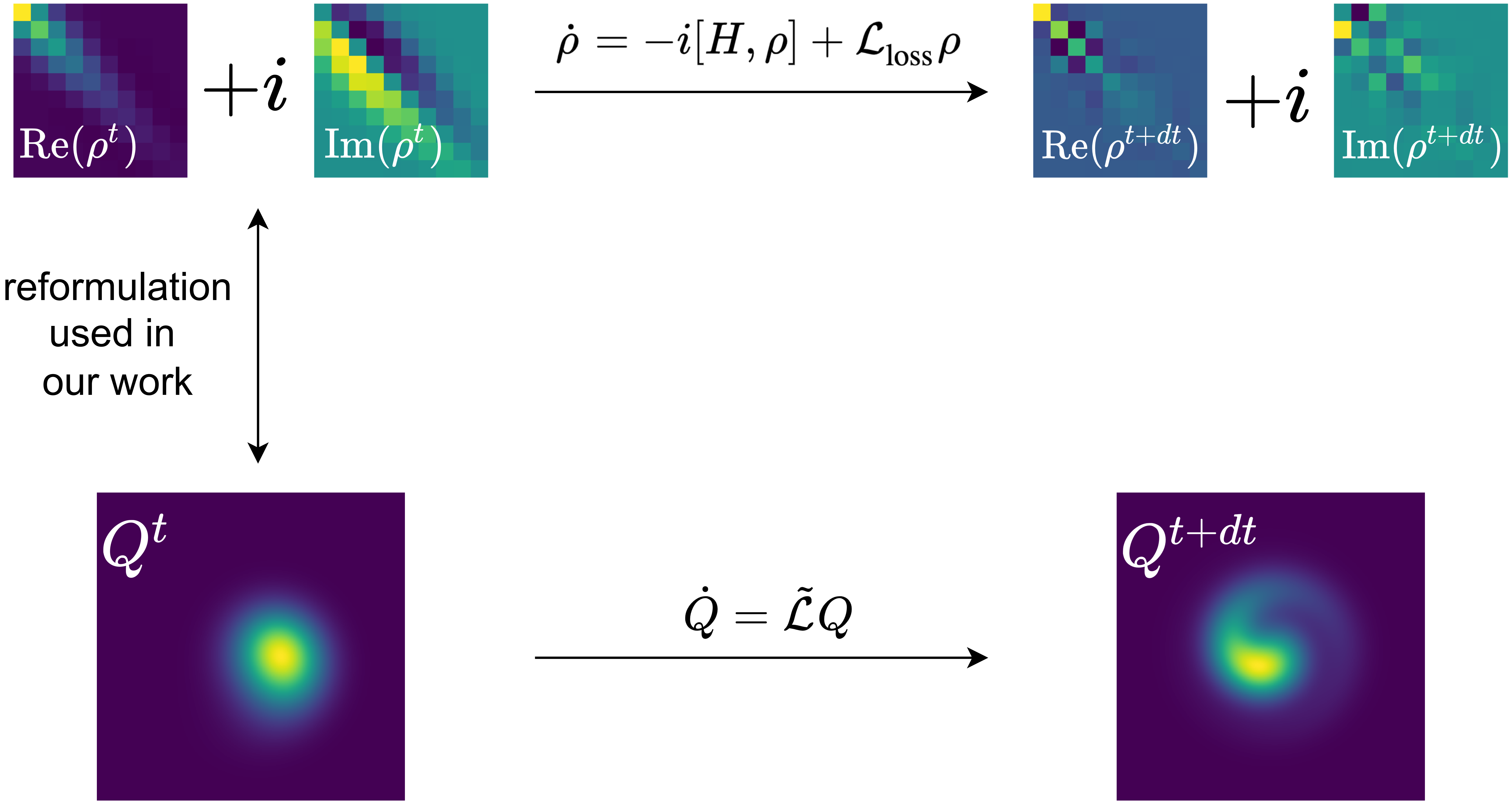
Q-Flow Procedure

1. Convert EOM and ρ to Q Function



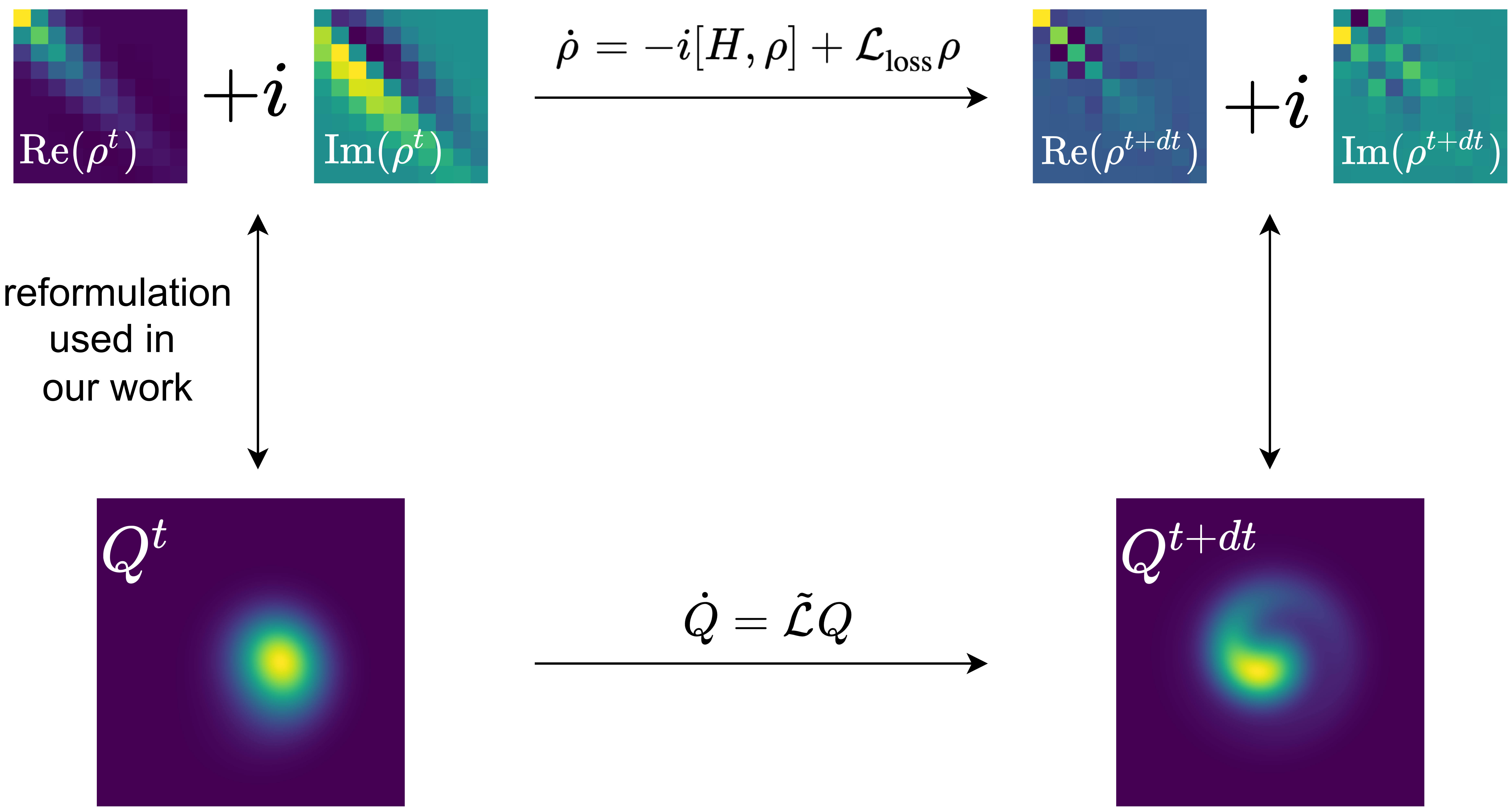
Q-Flow Procedure

1. Convert EOM and ρ to Q Function



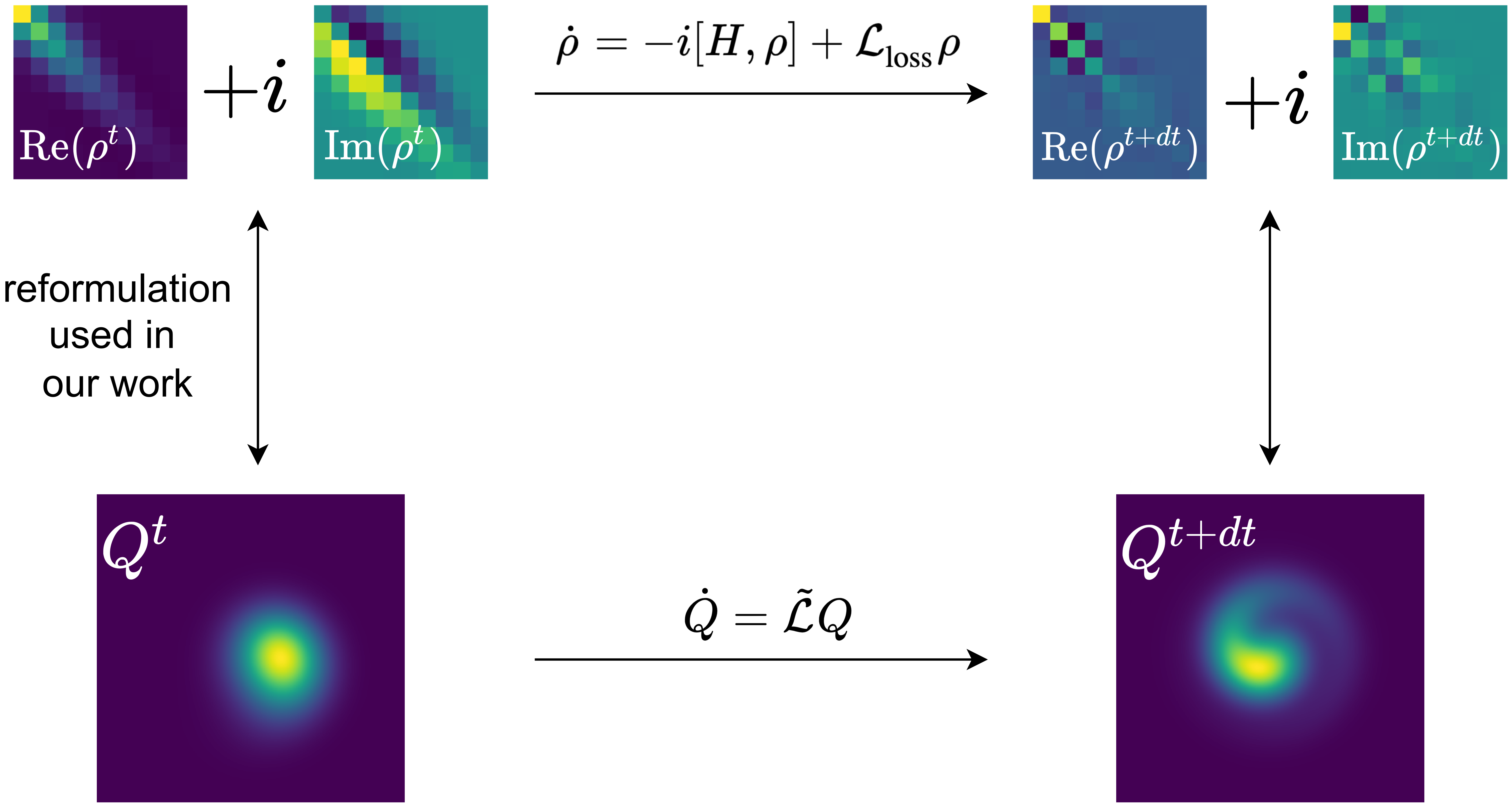
Q-Flow Procedure

1. Convert EOM and ρ to Q Function



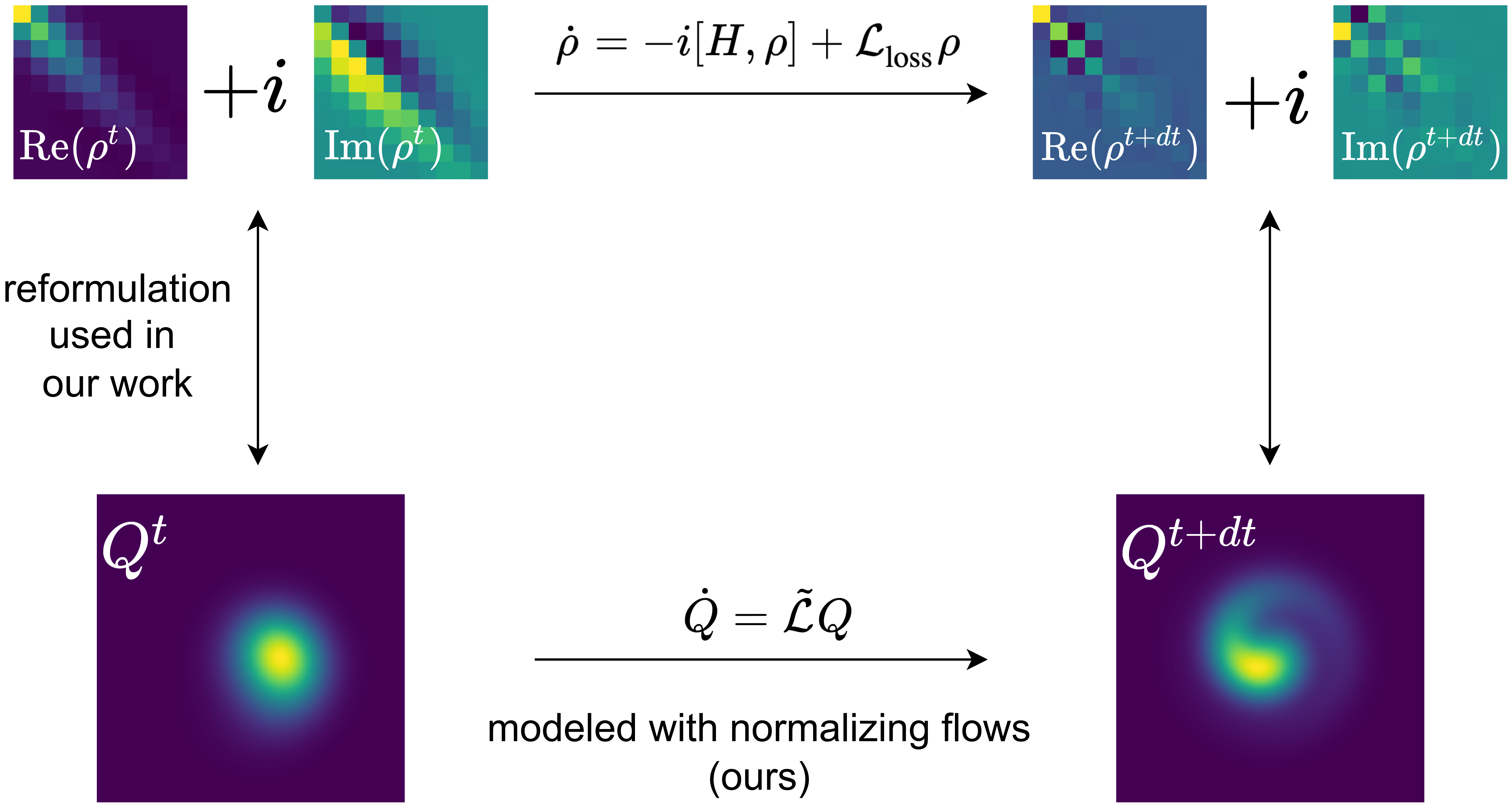
Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow



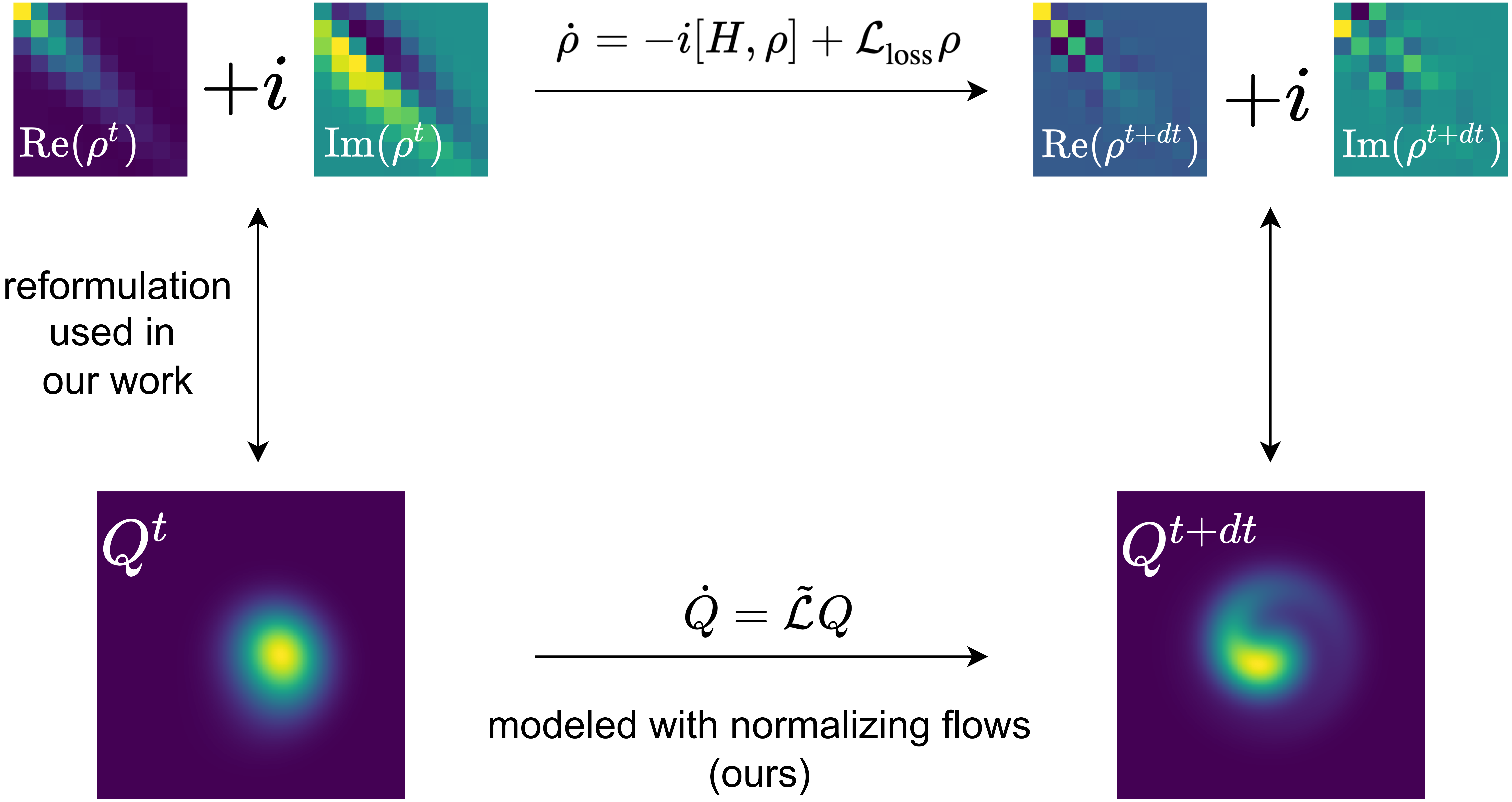
Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow



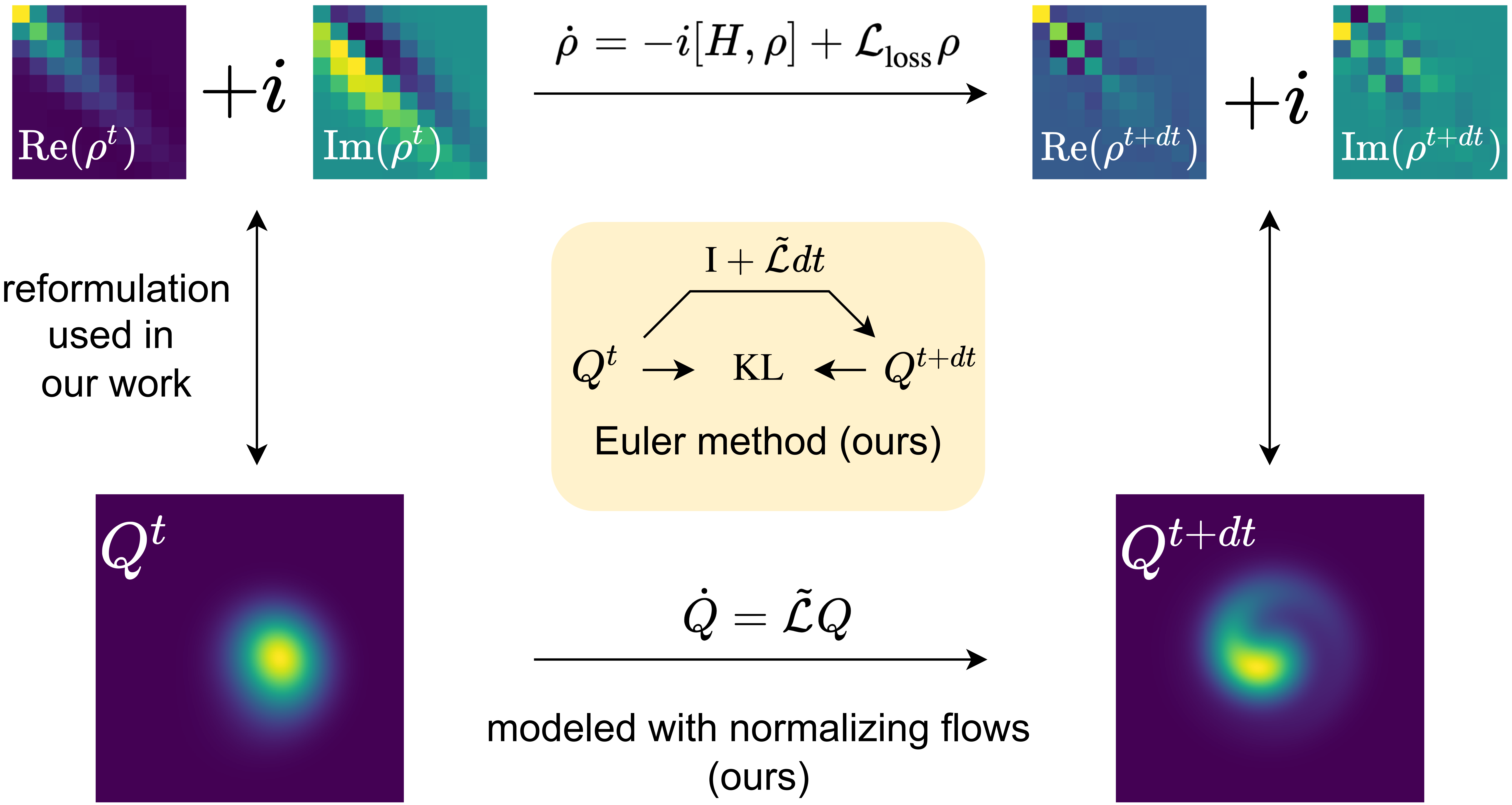
Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow
3. Evolve flow with Euler/TDVP



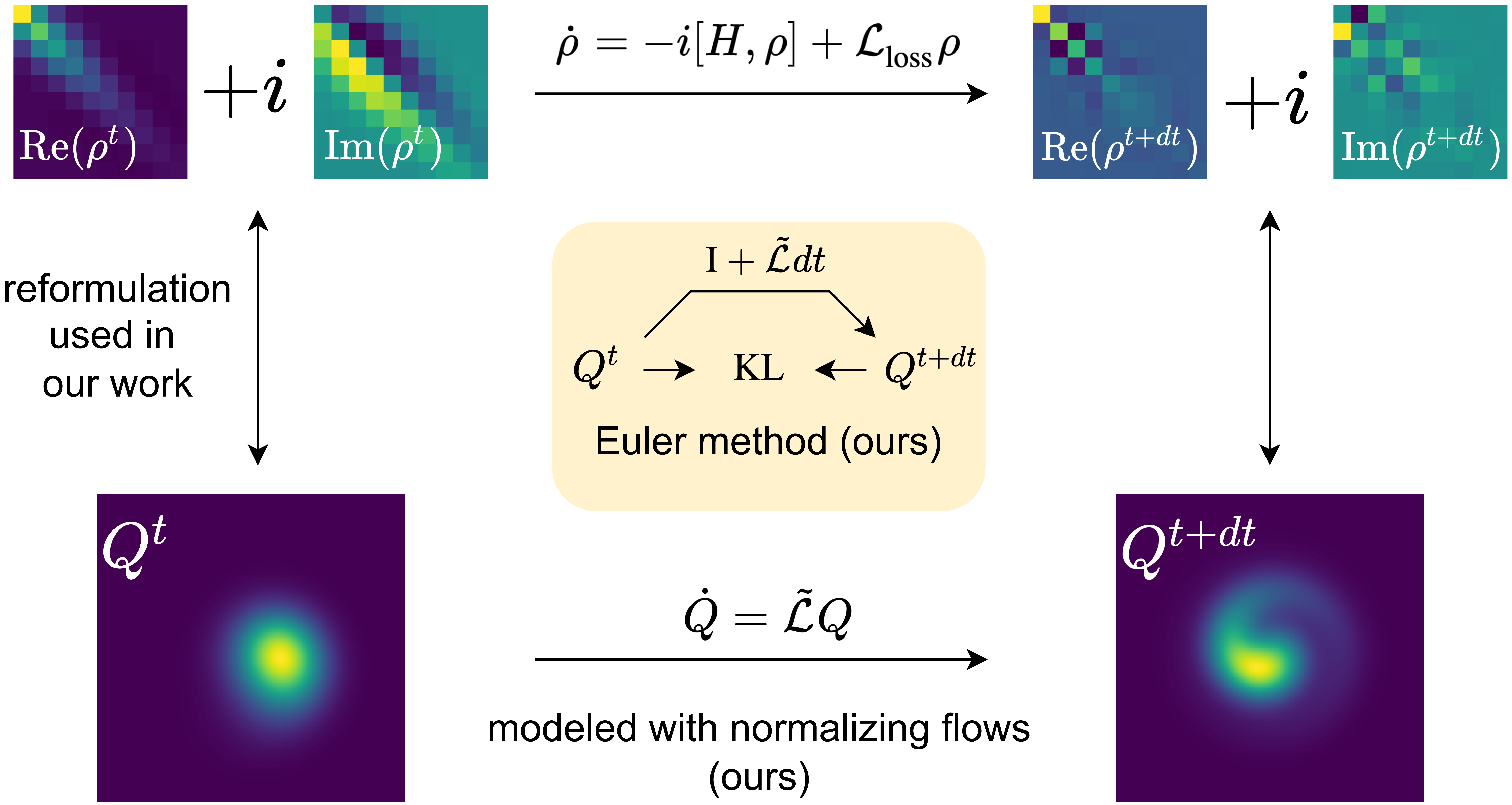
Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow
3. Evolve flow with Euler/TDVP



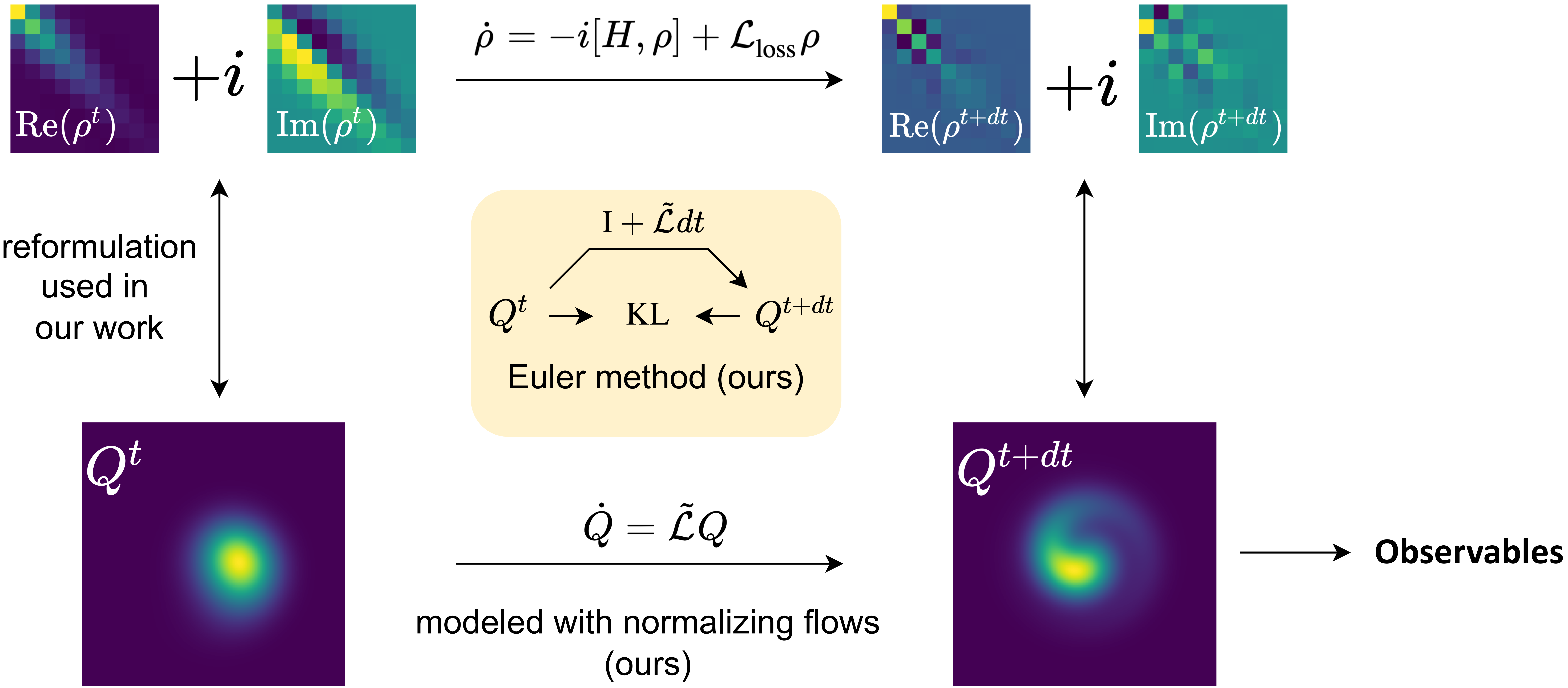
Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow
3. Evolve flow with Euler/TDVP
4. Sample Q to compute observables



Q-Flow Procedure

1. Convert EOM and ρ to Q Function
2. Model Q with a normalizing flow
3. Evolve flow with Euler/TDVP
4. Sample Q to compute observables



Q-Flow Contributions, Part 1

Establish Connection to Generative Modeling:

- Model continuous quantum systems with the Husimi Q function
- Q is a (quasi)probability distribution —> generative modeling

Q-Flow will continue to improve as generative models improve

Novel Methods for Solving Complex PDEs with Normalizing Flows:

- New Stochastic Euler-KL method evolves normalizing flows according to EOM
- Can also use Time Dependent Variational Principle (TDVP)

Can simulate any PDE for probability distributions (not just quantum systems)

Q-Flow Contributions, Part 2

Demonstration of Scalability and Efficiency:

- Superior performance to benchmarks for Bose-Hubbard systems
- Superior performance to benchmarks in high dimensional Dissipative Harmonic Oscillator Systems

With Q-Flow, the challenge of simulating open quantum systems shifts from high dimensionality to Q function complexity

Q-Function Details

Husimi Q Function

Definition:

$$Q(\vec{q},\vec{p}) = \frac{1}{\pi}\langle \overrightarrow{\alpha} \,|\, \rho \,|\, \overrightarrow{\alpha} \rangle, \text{ where } \overrightarrow{\alpha} = \vec{q} + i\vec{p} \text{ and } |\, \overrightarrow{\alpha} \rangle = |\, \alpha_1 \rangle \otimes \cdots \otimes |\, \alpha_n \rangle$$

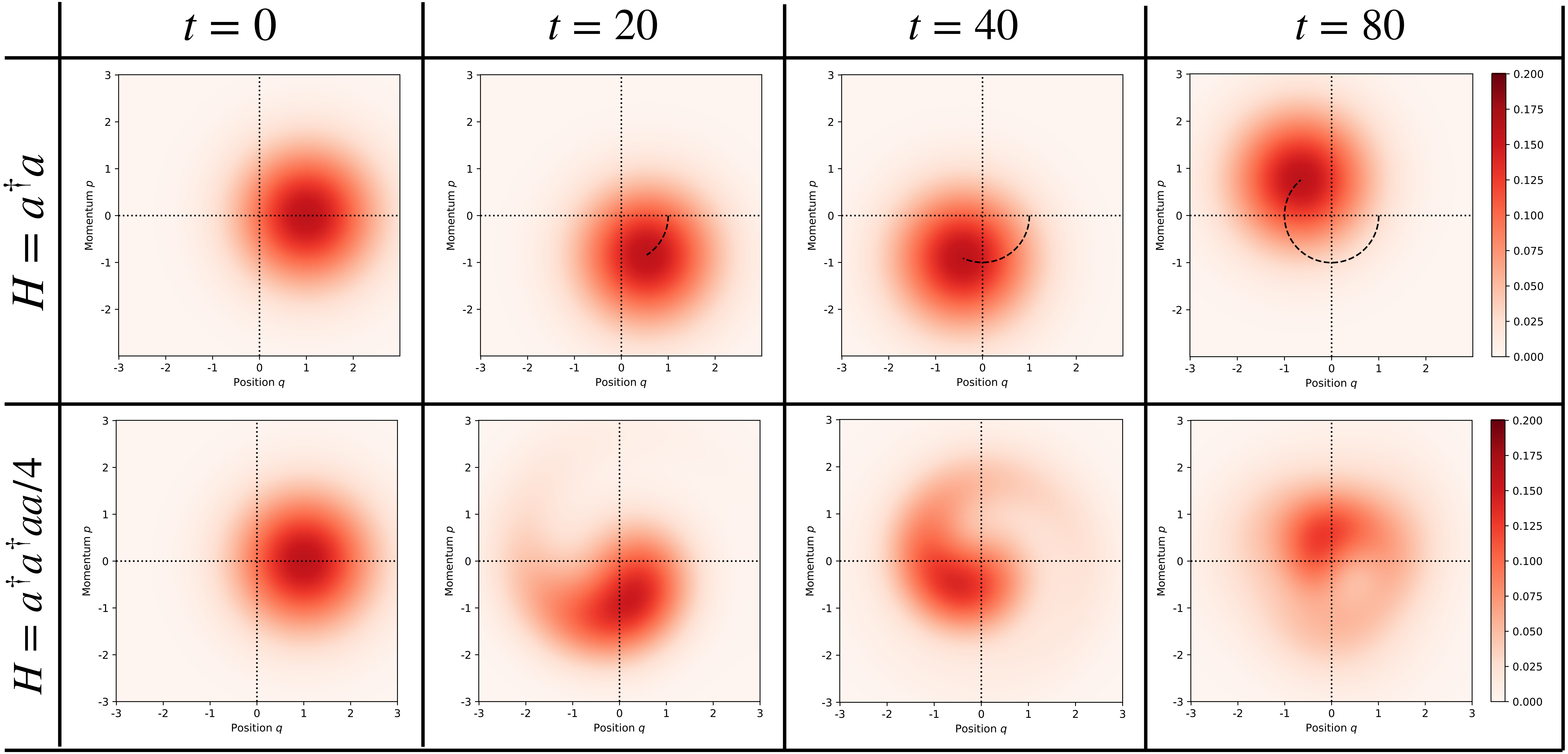
Coherent states

Normalization: $\int Q(\vec{q},\vec{p})\, \mathrm{d}\vec{q}\, \mathrm{d}\vec{p} = 1$

Positivity: $\forall \vec{q},\vec{p}, \quad Q(\vec{q},\vec{p}) \geq 0$

Q-Function is a (quasi)probability distribution!

Husimi Q Function Visualizations



Q Function Properties

Property 1: Q functions and density matrices are 1-1.

Property 2: For every local observable, there exists a Q-Flow representation which can compute the observable efficiently.

Property 3: Any density matrix EOM in composed of raising and lowering operators $a^\dagger$ and a can be converted to a Q-Function EOM.

Q-Flow Method: Modeling and Evolving Q

Normalizing Flows to Model Q

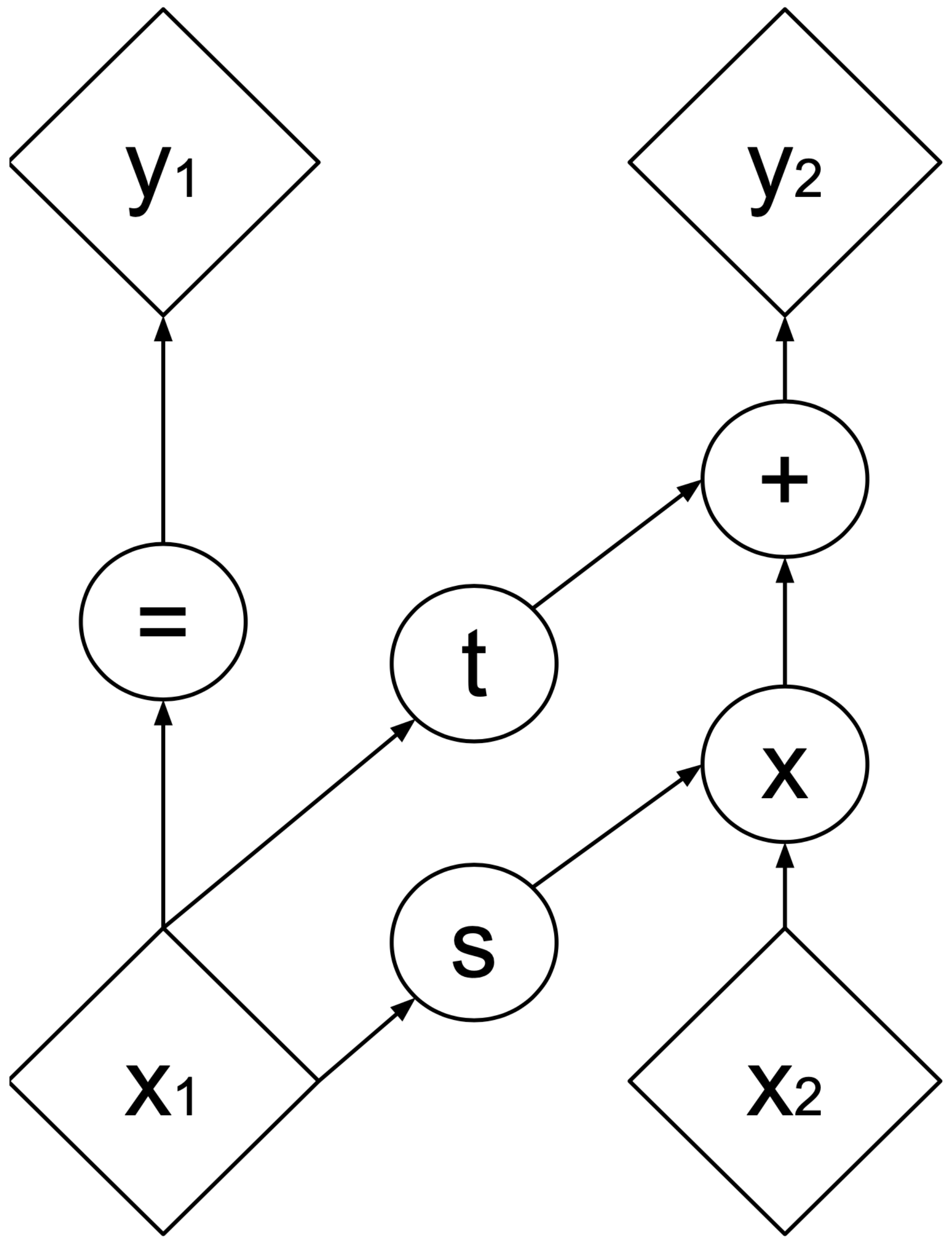
Requirements:

- Model Q, a probability distribution
- Sample from and evaluate Q
- Q must be differentiable

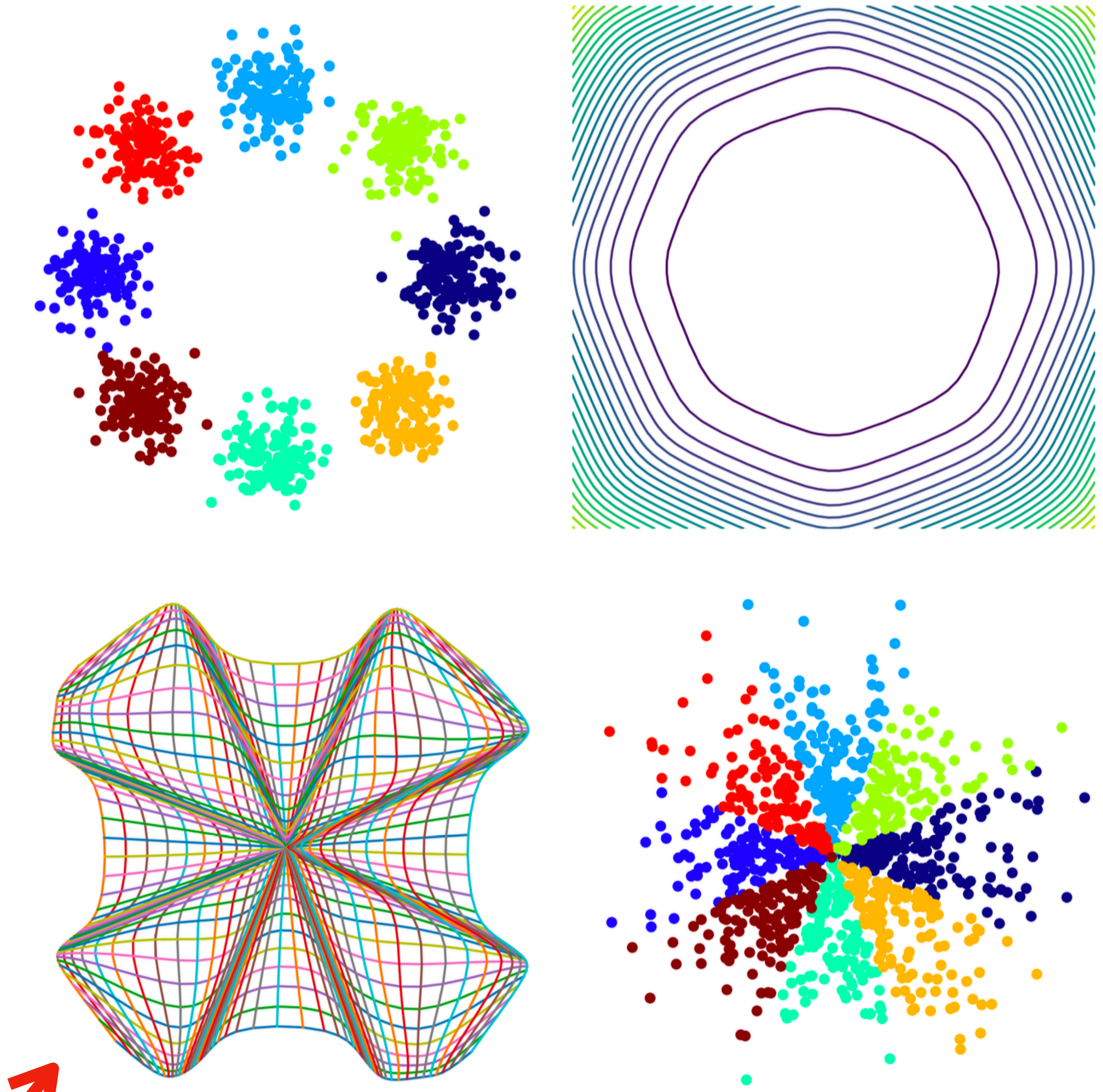
Solution — Normalizing Flows:

- RealNVP
- Convex Potential Flows

$$p_Y(y) = p_X(f_\theta^{-1}(y)) \left| \frac{\partial f_\theta^{-1}(y)}{\partial y} \right|$$



https://arxiv.org/abs/1605.08803



https://arxiv.org/abs/2012.05942

Modeling Initial Q Function — Pretraining

Problem: Fit a known distribution Q_{init} using a normalizing flow Q_θ

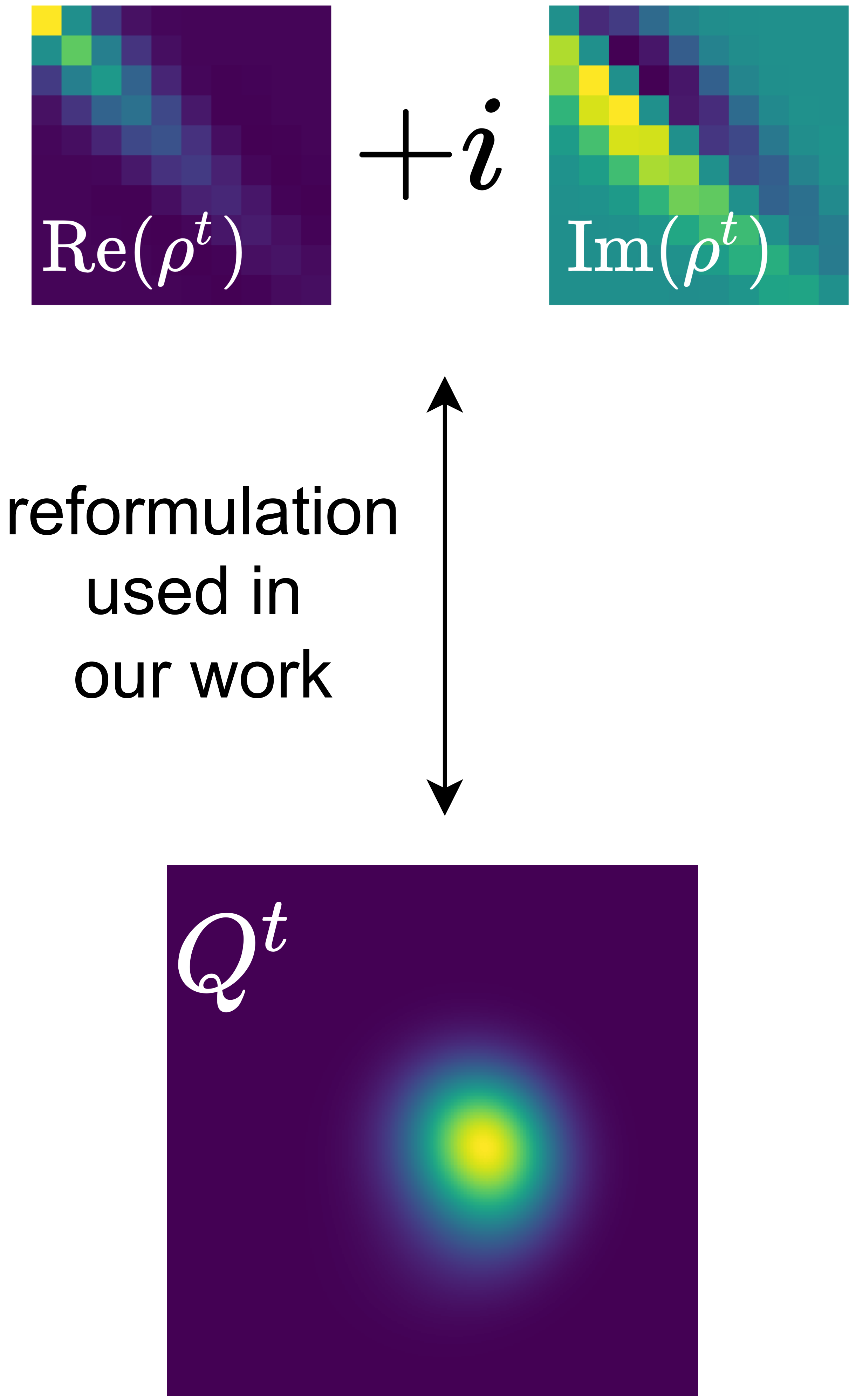
Solution: Train using KL divergence loss

Step 1 — sample from Q_{init} (MCMC):

$$\nabla_\theta KL = -\frac{1}{N}\sum_{x\sim Q_{init}} \nabla_\theta \ln Q_\theta(x)$$

Step 2 — sample from Q_θ:

$$\nabla_\theta KL = -\frac{1}{N}\sum_{x\sim Q_\theta} \frac{Q_{init}(x)}{Q_\theta(x)} \nabla_\theta \ln Q_\theta(x)$$

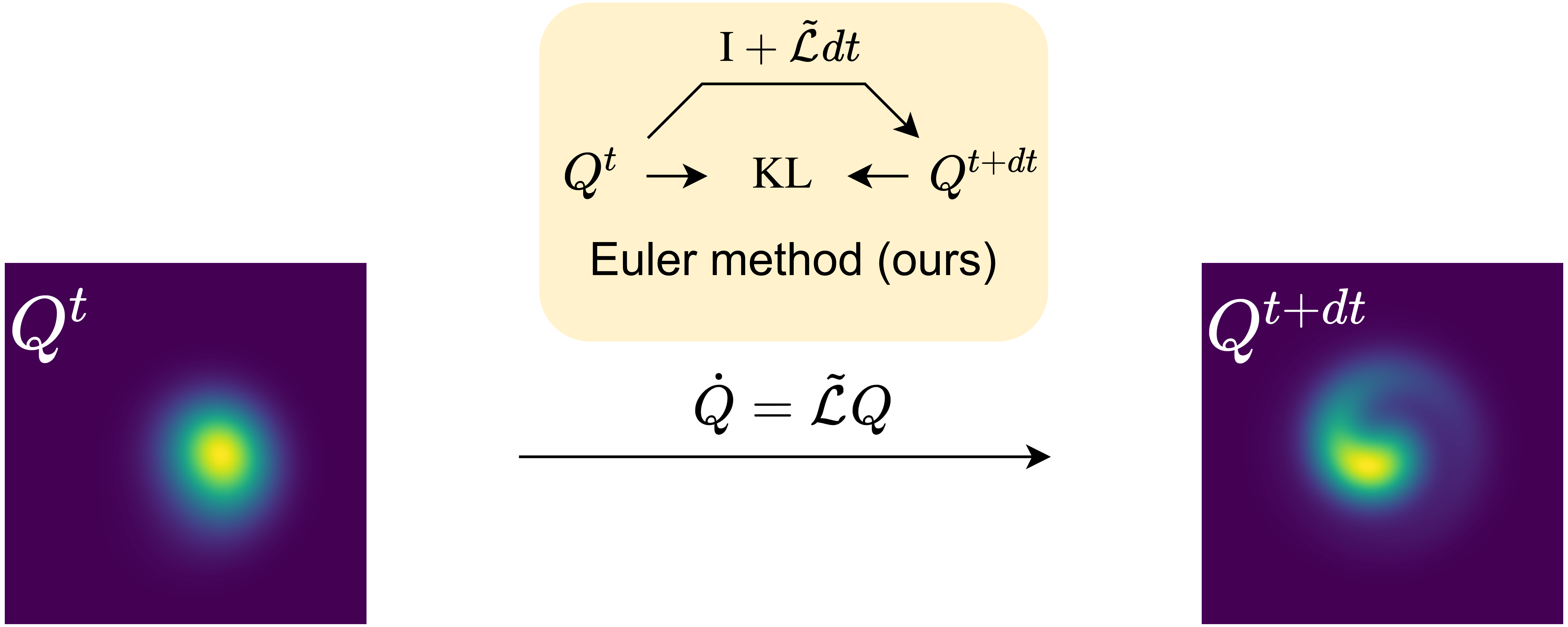


Q Function Time Evolution: Stochastic Euler Method

Euler Method: using $\dot{Q} = \tilde{\mathcal{L}}Q$, repeatedly evolve by dt: $Q^{t+dt} = Q^t + \tilde{\mathcal{L}}Q^t dt$

To model Q_θ^{t+dt}, train a new flow using a KL loss between Q_θ^{t+dt} and $Q^t + \tilde{\mathcal{L}}Q^t dt$:

$$KL(Q_\theta^{t+dt} || Q_\theta^t + \tilde{\mathcal{L}}Q_\theta^t dt) = \int Q_\theta^{t+dt} \ln \frac{Q_\theta^{t+dt}}{Q_\theta^t + \tilde{\mathcal{L}}Q_\theta^t dt}$$

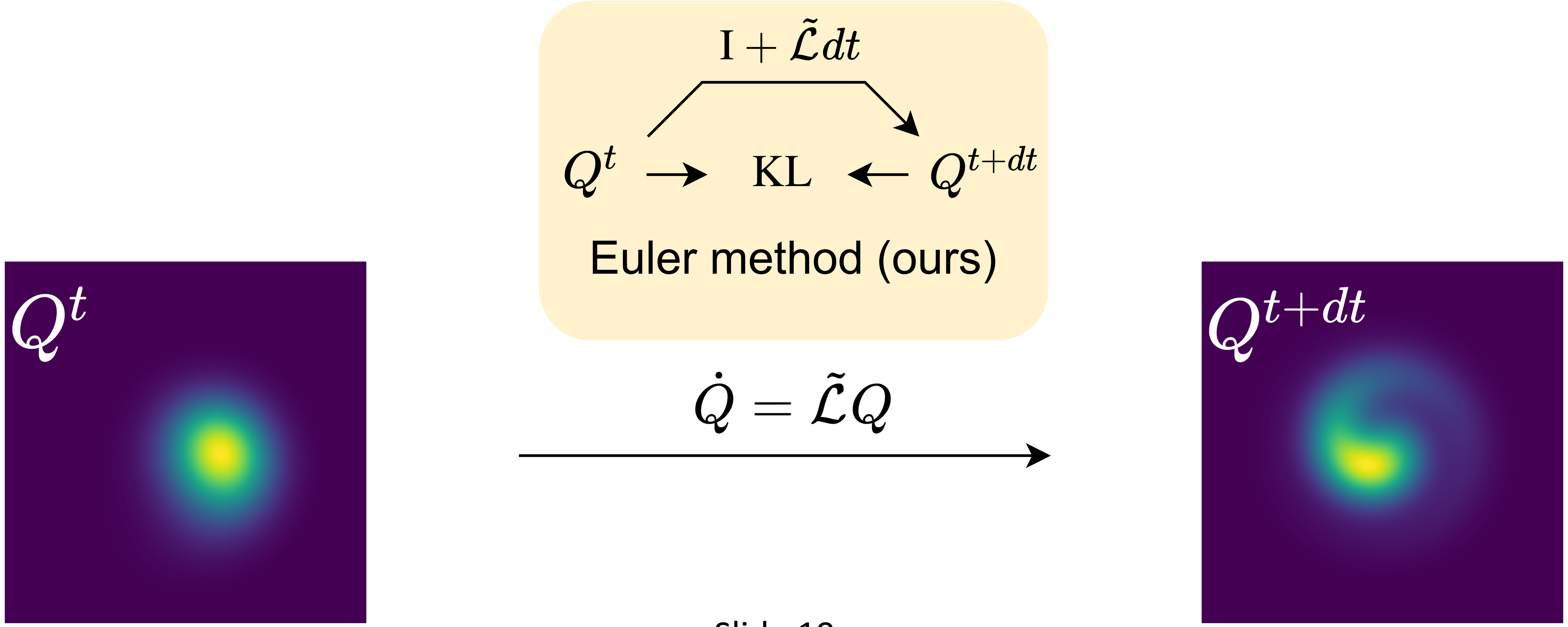


Q Function Time Evolution: TDVP Method

Computes $\Delta\theta$ directly: $\theta + \Delta\theta$ parametrizes the flow closest to $Q^t + \tilde{\mathcal{L}}Q^t dt$ for $\Delta\theta = \dot{\theta} dt$ and $S_{kk'}\dot{\theta}_{k'} = F_k$, where

$$S_{kk'} = \mathbb{E}[(\partial_{\theta_k} \ln Q)(\partial_{\theta'_k} \ln Q)] \qquad F_k = \mathbb{E}[(\partial_{\theta_k} \ln Q)(\partial_t \ln Q)]$$

$$\partial_t \ln Q = (\partial_t Q)/Q = (\tilde{\mathcal{L}}Q)/Q$$



Results: Dissipative Bose-Hubbard

Motivation for Experiment:

Real-world applications: Bosonic analog of Fermi-Hubbard, model for superconductors

Complex inter-site interactions: difficult to model

Bose-Hubbard to Q Function Formalism

	Density matrix	Q function
Equation of Motion	$H = -J \sum_j \left(a_{j+1}^{\dagger} a_j + a_j^{\dagger} a_{j+1} \right)$ $\mathcal{L}_{\text{loss}}\rho = -\frac{1}{2} \sum_j \gamma_j \left(n_j \rho + \rho n_j - 2 a_j \rho a_j^{\dagger} \right)$	$\tilde{\mathcal{L}} = \sum_j \gamma_j \left(\frac{1}{4} \left(\frac{\partial^2}{\partial q_j^2} + \frac{\partial^2}{\partial p_j^2} \right) + \frac{1}{2} \left(q_j \frac{\partial}{\partial q_j} + p_j \frac{\partial}{\partial p_j} + 1 \right) \right)$ $+ J \sum_j \left(p_{j+1} \frac{\partial}{\partial q_j} - q_{j+1} \frac{\partial}{\partial p_j} + p_j \frac{\partial}{\partial q_{j+1}} - q_j \frac{\partial}{\partial p_{j+1}} \right)$
Initial State	$\rho = \frac{1}{100!} (a_1^{\dagger} - a_2^{\dagger})^{100} \|0\rangle \langle 0\| (a_1 - a_2)^{100}$	$Q = \frac{\left[(q_1 - q_2)^2 + (p_1 - p_2)^2 \right]^{100}}{\pi^2 \cdot 2^{100} \cdot 100!} e^{-(q_1^2 + p_1^2 + q_2^2 + p_2^2)}$

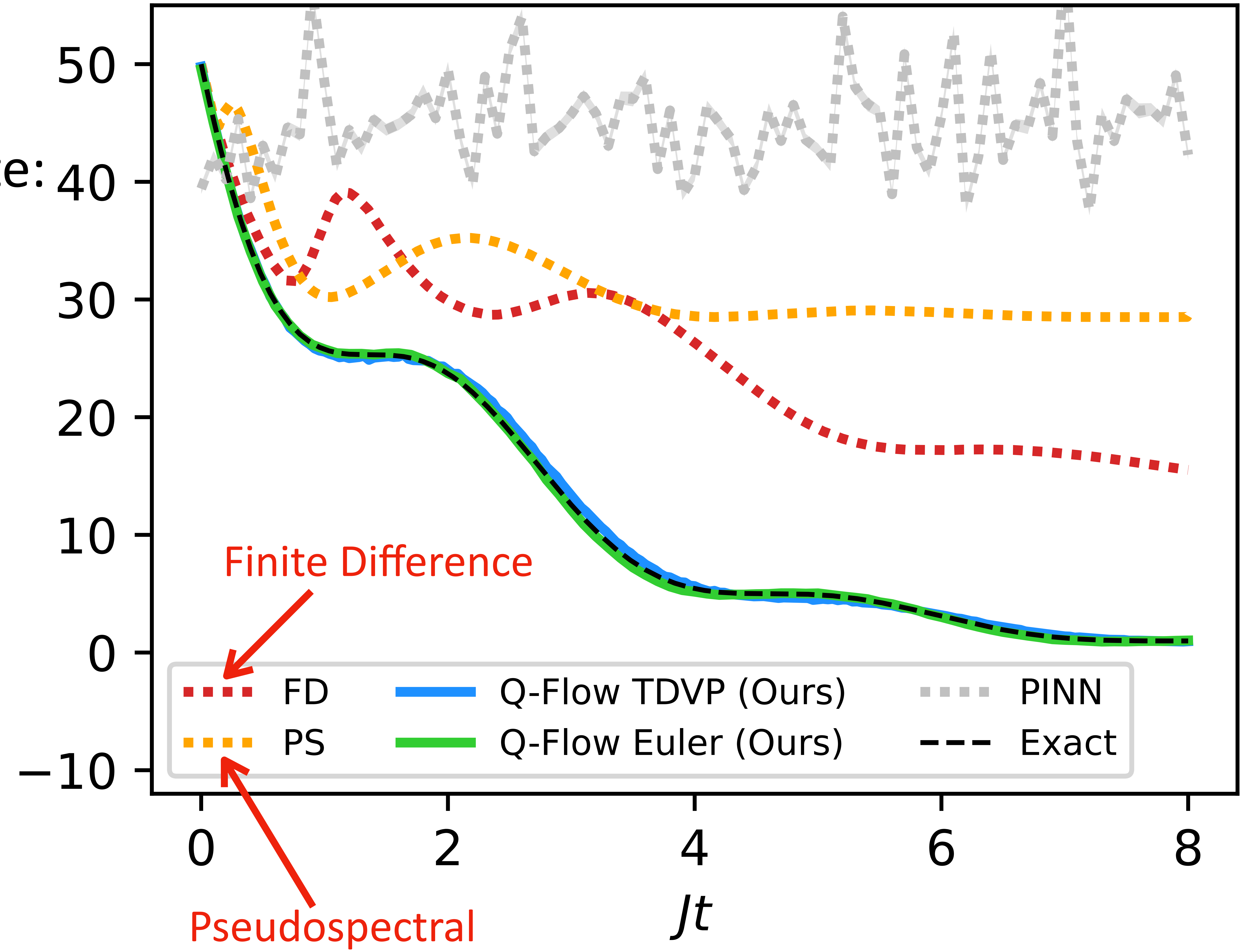
Results: Particle Number Evolution

Metric — particle number in first site:

$$\langle n_1 \rangle \approx \frac{1}{N} \sum_{(\vec{q},\vec{p})\sim Q_{\text{sim}}} (q_1^2 + p_1^2 - 1)$$

Exact evolution known:
arxiv.org/abs/1510.00127

Both Q-Flow methods
outperform all other methods!



Results: Dissipative Harmonic Oscillator

Motivation for Experiment:

Exact solution known: useful for testing high-dimensional systems

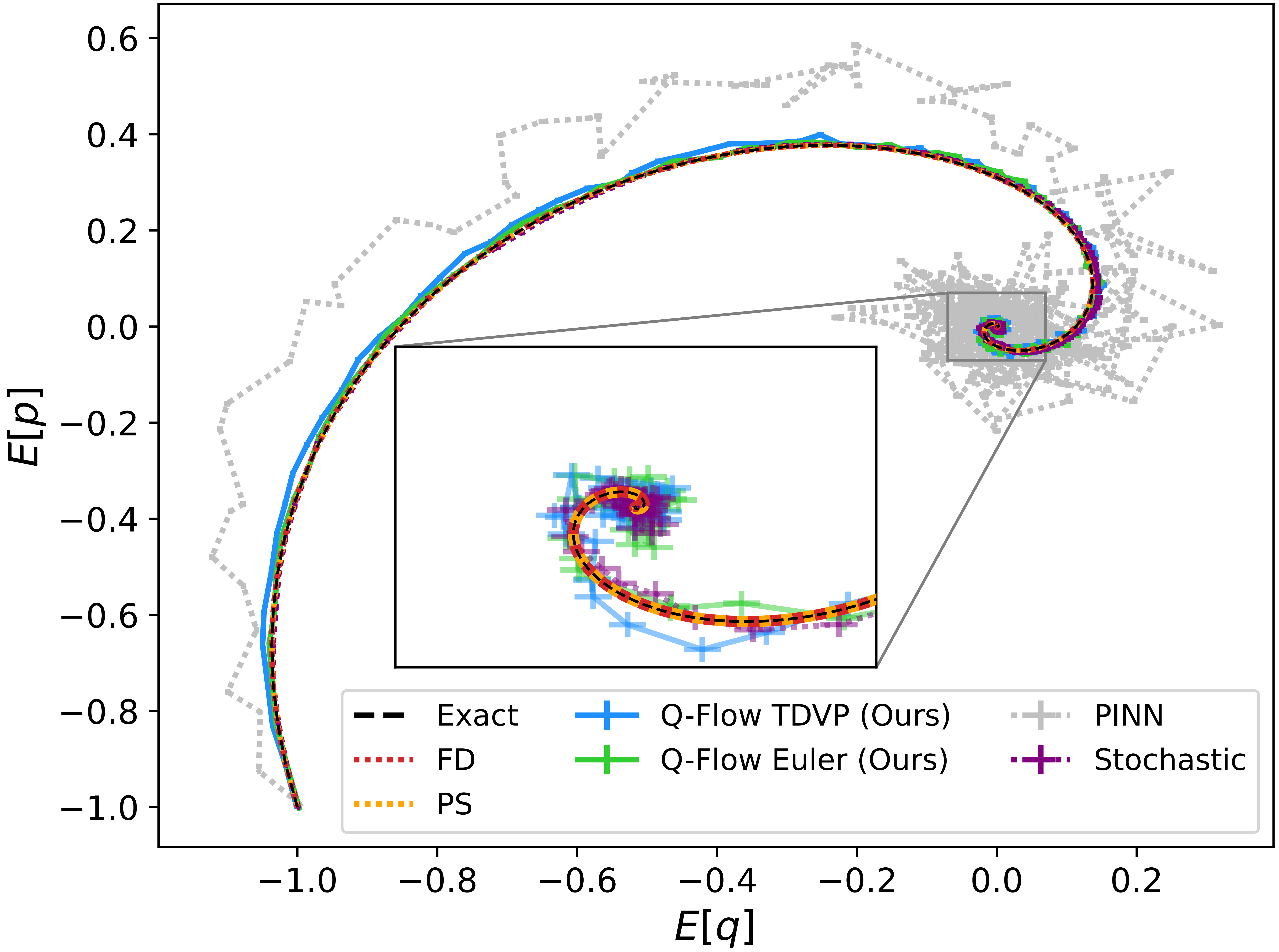
Dissipative Harmonic Oscillator to Q Function Formalism

	Density matrix →	Q function
Equation of Motion	$H = \sum_j \omega_j a_j^\dagger a_j$ $\mathcal{L}_{\text{loss}}\rho = \sum_j \gamma_j \left[\frac{1}{2}(2a_j\rho a_j^\dagger - a_j^\dagger a_j \rho - \rho a_j^\dagger a_j) + \bar{n}_j(a_j\rho a_j^\dagger + a_j^\dagger \rho a_j - a_j^\dagger a_j \rho - \rho a_j a_j^\dagger)\right]$	$\tilde{\mathcal{L}} = \sum_j \left[\gamma_j + \frac{1}{4}\gamma_j(\bar{n}_j + 1)\left(\frac{\partial^2}{\partial q_j^2} + \frac{\partial^2}{\partial p_j^2}\right) + \left(\frac{\gamma_j}{2}q_j - \omega_j p_j\right)\frac{\partial}{\partial q_j} + \left(\frac{\gamma_j}{2}p_j + \omega_j q_j\right)\frac{\partial}{\partial p_j}\right]$
Initial State	$\rho = \lvert\vec{\alpha}\rangle\langle\vec{\alpha}\rvert$ where $\forall j, \vec{\alpha}_j = -1 - i$	Gaussian: $Q(\vec{x}) = \pi^{-N/2} \exp\left(-\lvert\vec{x} - 1\rvert^2\right)$

Results: 1 Well Centroid Evolution

Metric — Q function centroid:

$$\mathbb{E}[\vec{x}] \approx \frac{1}{N}\sum_{x\sim Q_{\text{sim}}} \vec{x}$$



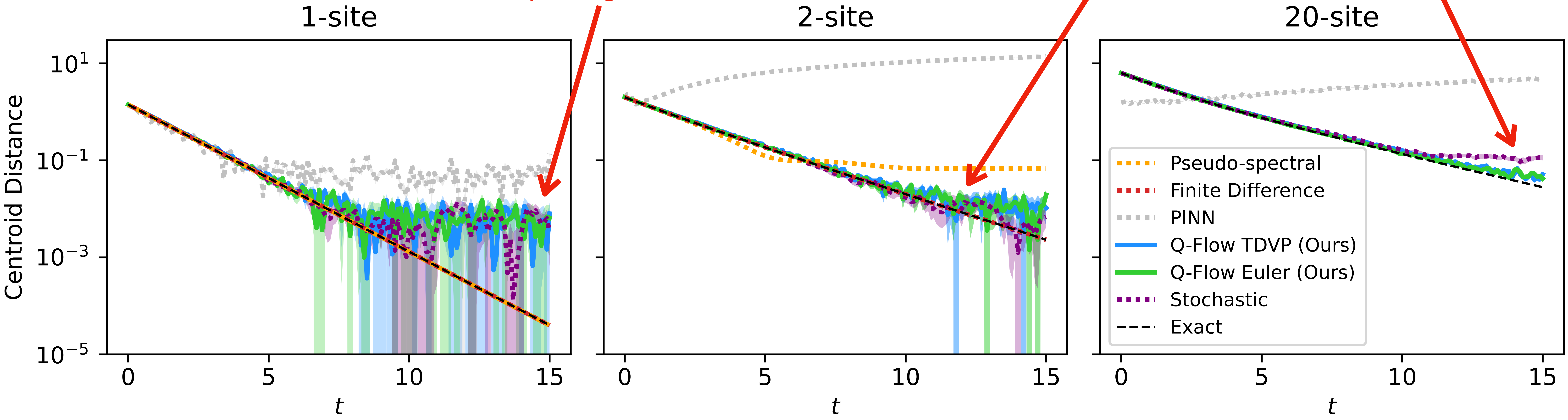
Q-Flow close to classical solver accuracy in a low dimensional system!

Results: Centroid Distance

Metric — distance of Q function centroid to origin

Divergence from exact due to sampling error in computing observable

Q-Flow and Stochastic Method match exact even for high dimensional systems



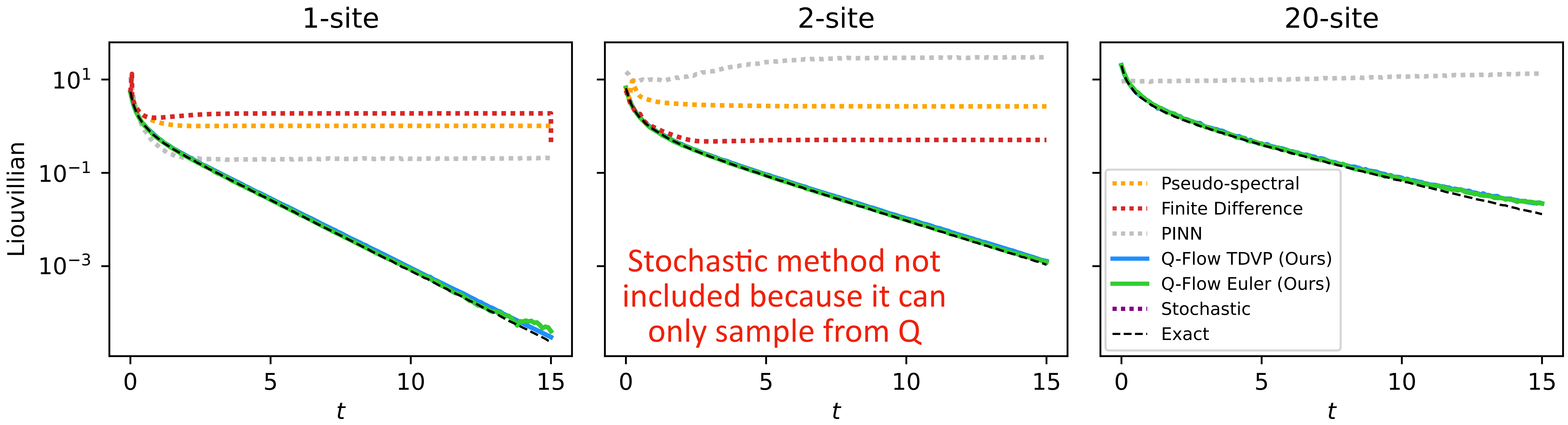
Finite Difference and Pseudospectral methods infeasible for 20-site systems (all metrics)

Results: Liouvillian

Measure of convergence to equilibrium

Metric — integral of the Liouvillian:

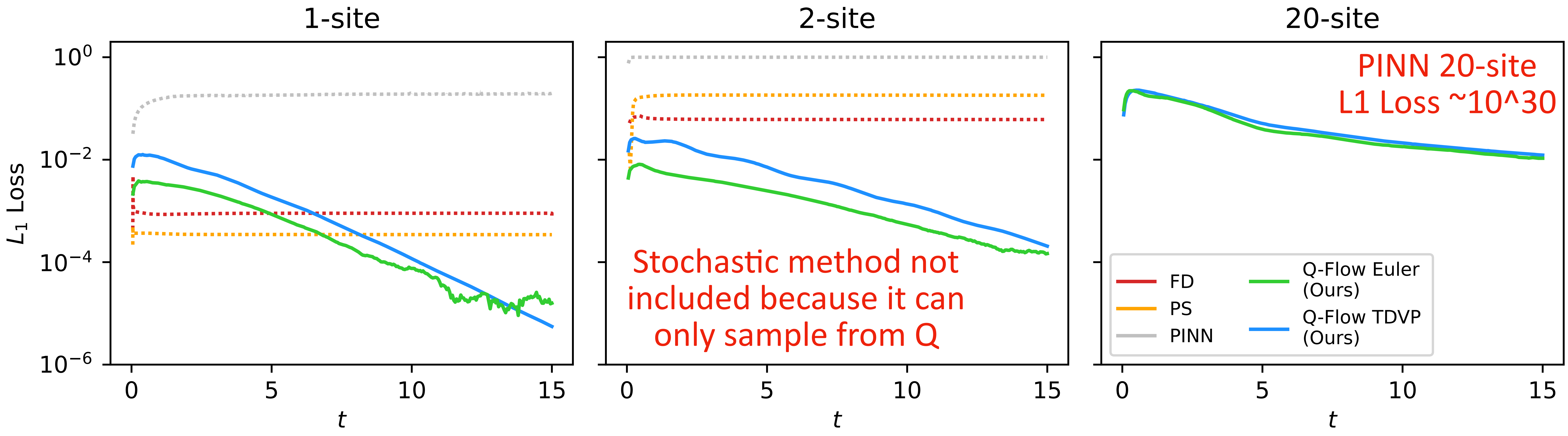
$$\int dx \, \left|[\tilde{\mathcal{L}}Q](x)\right| = \mathbb{E}[|\tilde{\mathcal{L}}Q|/Q]$$



Only Q-Flow correctly estimates Liouvillian Loss for any number of sites

Results: L1 Loss

Metric — L1 loss between simulated and exact Q: $L_1[Q_{\text{sim}}, Q_{\text{exact}}] \equiv \int \mathrm{d}^d x \, |Q_{\text{sim}}(x) - Q_{\text{exact}}(x)|$



Q-Flow L1 Loss consistently less than other methods' for high dimensional systems

Conclusion

- Developed Q-Flow:
  - Connects off-the-shelf generative models to open quantum systems
  - Novel method for solving complex PDEs with Normalizing Flows
  - Demonstrated scalability and efficiency
    - Improved performance relative to standard PDE solvers in high-dimensional systems and systems with complex interactions

With Q-Flow, the challenge of simulating open quantum systems shifts from high dimensionality to Q function complexity

Q-Flow will continue to improve as generative models improve!

Thank you!