LieStoNet:
Learning Lie Symmetries from Spatiotemporal Data for Stochastic Dynamical Systems

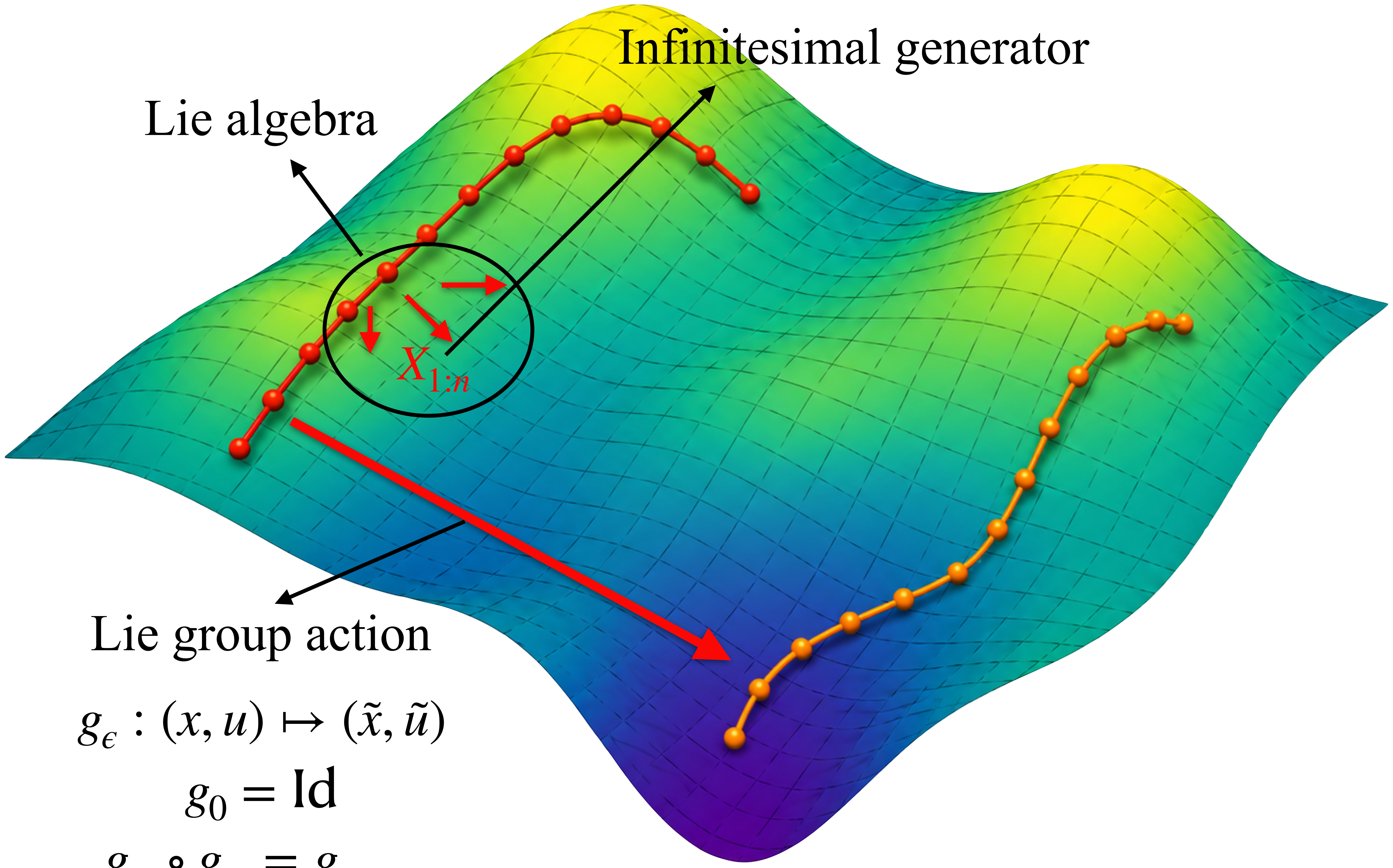
ICML 2026

Authors: S. Liu, A. Gupta, S. Sinha, L. Mahadevan
Speaker: S. Liu

“The Game on the Rug”

What is symmetry and why is it important?

Preliminaries of Lie Theories



$$g_\epsilon : (x, u) \mapsto (\tilde{x}, \tilde{u})$$
$$g_0 = \mathrm{Id}$$
$$g_{\epsilon_1} \circ g_{\epsilon_2} = g_{\epsilon_1 + \epsilon_2}$$

- **Lie group -** a smooth one-parameter family of transformations
- **Infinitesimal generator** / **Lie algebra -** the tangent vector field at the identity is:

$$X = \sum_i \xi^i(x, u)\partial_{x^i} + \sum_\alpha \phi^\alpha(x, u)\partial_{u^\alpha}$$

These generators form a Lie algebra with bracket: $[X, Y] = XY - YX$

- **Lie-point symmetry of an ODE/PDE -** for a system:

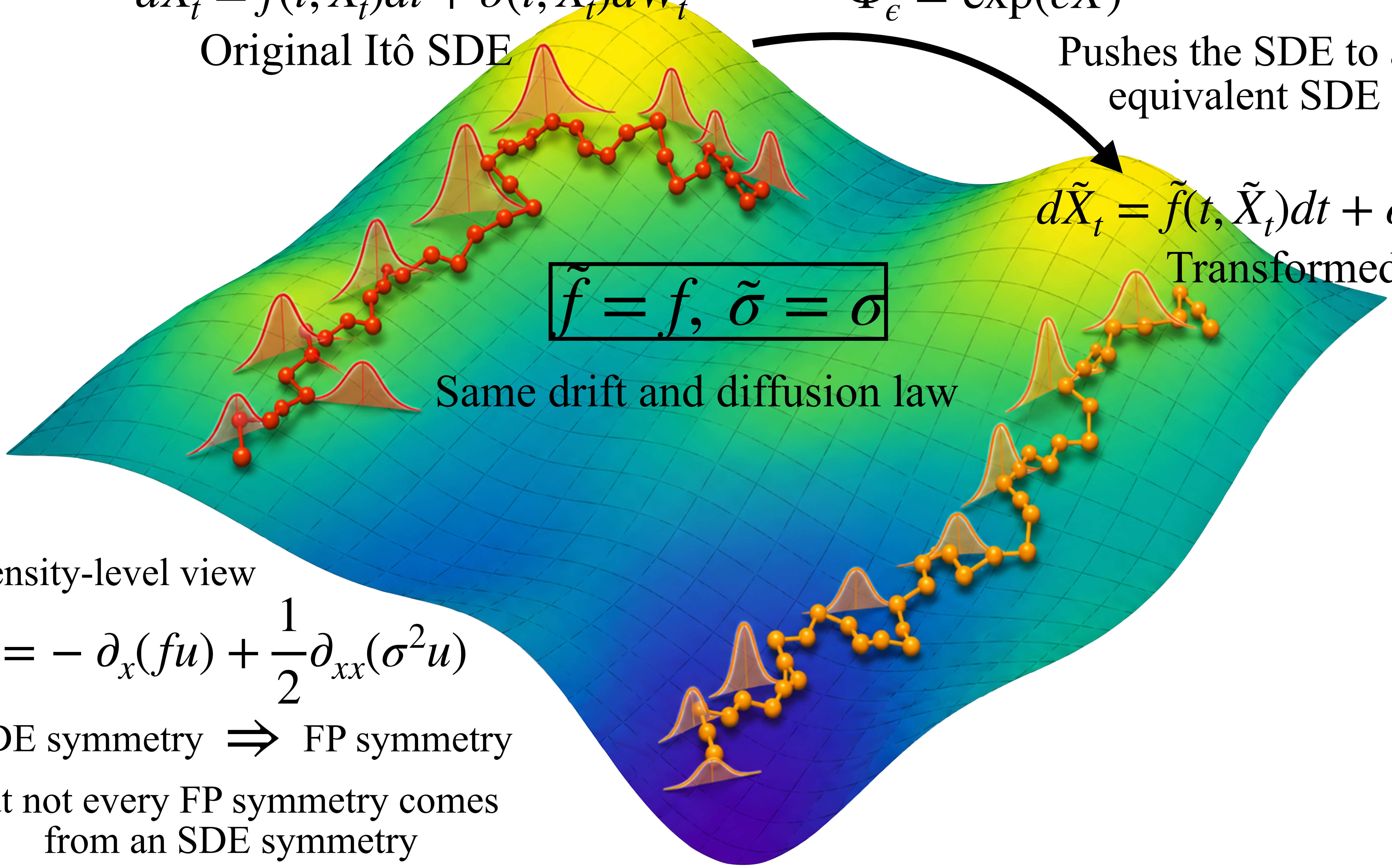
$$\Delta(x, u, u^{(1)}, \ldots, u^{(n)}) = 0$$

A Lie-point symmetry satisfies the *infinitesimal invariance condition*:

$$\mathrm{pr}^{(n)} X[\Delta]\,|_{\Delta=0} = 0$$

i.e. the transformation maps solutions to solutions.

Lie-Point Symmetries of Itô SDEs



$$dX_t = f(t, X_t)dt + \sigma(t, X_t)dW_t$$

Original Itô SDE

$$\Phi_\epsilon = \exp(\epsilon X)$$

Pushes the SDE to an equivalent SDE

$$d\tilde{X}_t = \tilde{f}(t, \tilde{X}_t)dt + \tilde{\sigma}(t, \tilde{X}_t)dW_t$$

Transformed Itô SDE

$$\boxed{\tilde{f} = f,\ \tilde{\sigma} = \sigma}$$

Same drift and diffusion law

Density-level view

$$u_t = -\partial_x(fu) + \frac{1}{2}\partial_{xx}(\sigma^2 u)$$

SDE symmetry $\Rightarrow$ FP symmetry

but not every FP symmetry comes from an SDE symmetry

- **Projectable infinitesimal generator**

$$X = \tau(t)\partial_t + \xi(t,x)\partial_x$$

- **Gaeta-Quintero determining equations**

$$\xi_t + f\xi_x - \xi f_x - \partial_t(f\tau) + \frac{1}{2}\sigma^2\xi_{xx} = 0$$

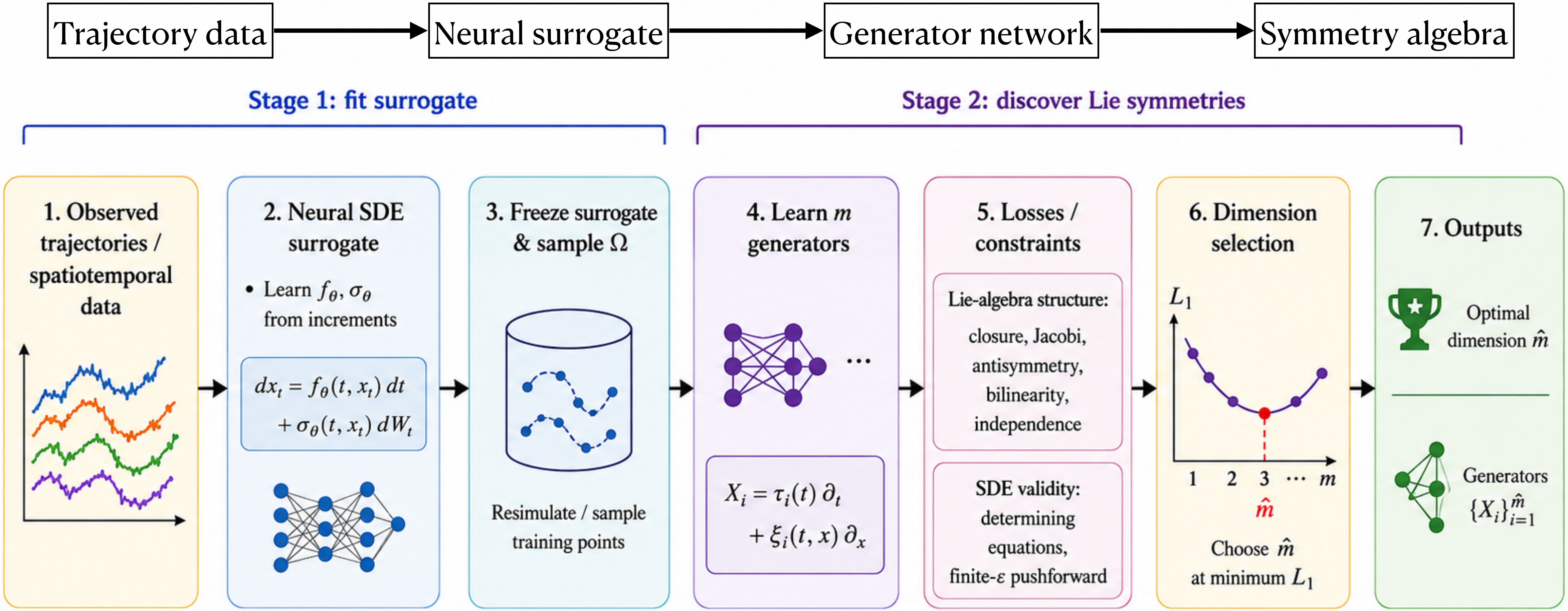
(Drift invariance)

$$\sigma\xi_x - \xi\sigma_x - \tau\sigma_t - \frac{1}{2}\sigma\tau_t = 0$$

(Diffusion invariance)

Takeaway: For SDEs, Lie symmetry must preserve both deterministic drift and stochastic diffusion.

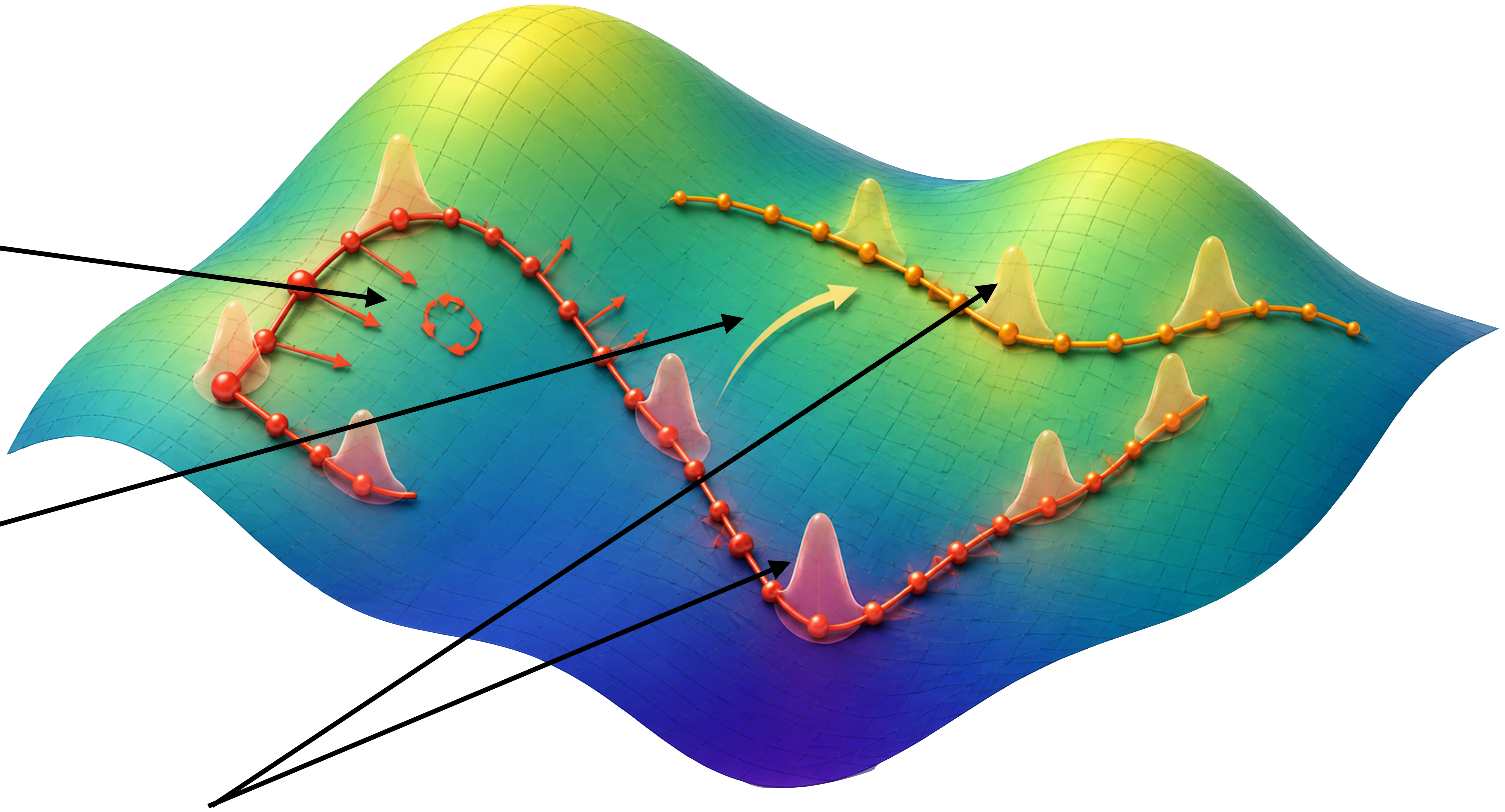
LieStoNet Pipeline



Two-stage approach: first learn a neural SDE surrogate, then learn symmetry generators under algebraic and SDE-validity constraints.

Loss Functions: Enforcing Symmetry and Algebra

$X = \tau(t)\partial_t + \xi(t,x)\partial_x,\ i = 1,\ldots,m$



1. Lie-algebra structure losses

- $\mathscr{L}_1$: closure & constant structure coefficients
- $\mathscr{L}_2$: Jacobi identity
- $\mathscr{L}_3, \mathscr{L}_4$: antisymmetry & bilinearity
- $\mathscr{L}_5$: functional independence

2. SDE symmetry losses

Purpose: each generator must preserve the learned drift and diffusion.

- $\mathscr{L}_6 = \mathbb{E}_\Omega[\,|r_1|^2 + |r_2|^2],$

  $r_1 = \xi_t + f\xi_x - \xi f_x - \partial_t(f\tau) + \dfrac{1}{2}\sigma^2\xi_{xx}\,,$

  $r_2 = \sigma\xi_x - \xi\sigma_x - \tau\sigma_t - \dfrac{1}{2}\sigma\tau_t$

- $\mathscr{L}_7$: finite-ϵ pushforward consistency

3. FP symmetry variant

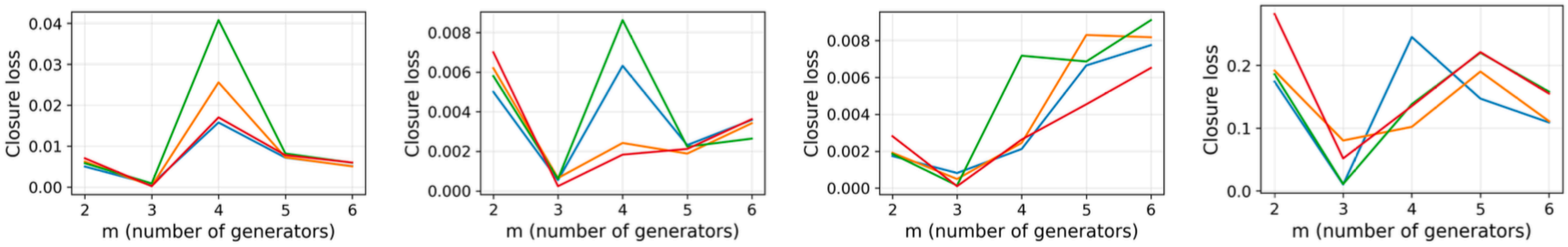
Purpose: replace SDE-validity losses with density-level validity losses.

$Y_i = \tau_i(t)\partial_t + \xi_i(t,x)\partial_x + \beta_i(t,x)u\partial_u$

- $\mathscr{L}_8$: FP determining-equation residual
- $\mathscr{L}_9$: finite-ϵ pushforward on u

Evaluation I: Recovering the Symmetry Algebra

Did LieStoNet recover the correct Lie-algebra dimension and span?



(a) Ex1 *(b)* Ex2 *(c)* Ex3 *(d)* Ex4

Figure 2. **Post-training Lie bracket closure loss L_1 vs. number of generators m across four experiments.** Colors denote independent runs with identical settings (different random seeds), showing the minimum is consistently attained at the ground-truth m rather than a one-off.

Dimension selection:

$$m^\star = \arg\min_m \mathscr{L}_1(m)$$

Sweep number of generators m.
Choose the minimum closure loss.

Basis-invariant span alignment:

$$\cos \theta_l = \theta_l(Q_{learned}^\top Q_{GT})$$

Principal angles compare learned and analytic generator spans; small angles mean correct symmetry algebra up to basis change.

Principal angles at selected dimensions

Ex.	θ_1	θ_2	θ_3
1	6.26°	2.49°	2.31°
2	19.37°	11.05°	0.57°
3	7.64°	1.33°	0.28°
4	14.29°	6.24°	2.51°

Result: $\mathscr{L}_1$ selects the analytic dimension, and principal angles are small.

Evaluation II: Finite-Flow and Empirical Validation

Do learned generators behave like true symmetries beyond infinitesimal residuals?

Finite-ϵ after-push check

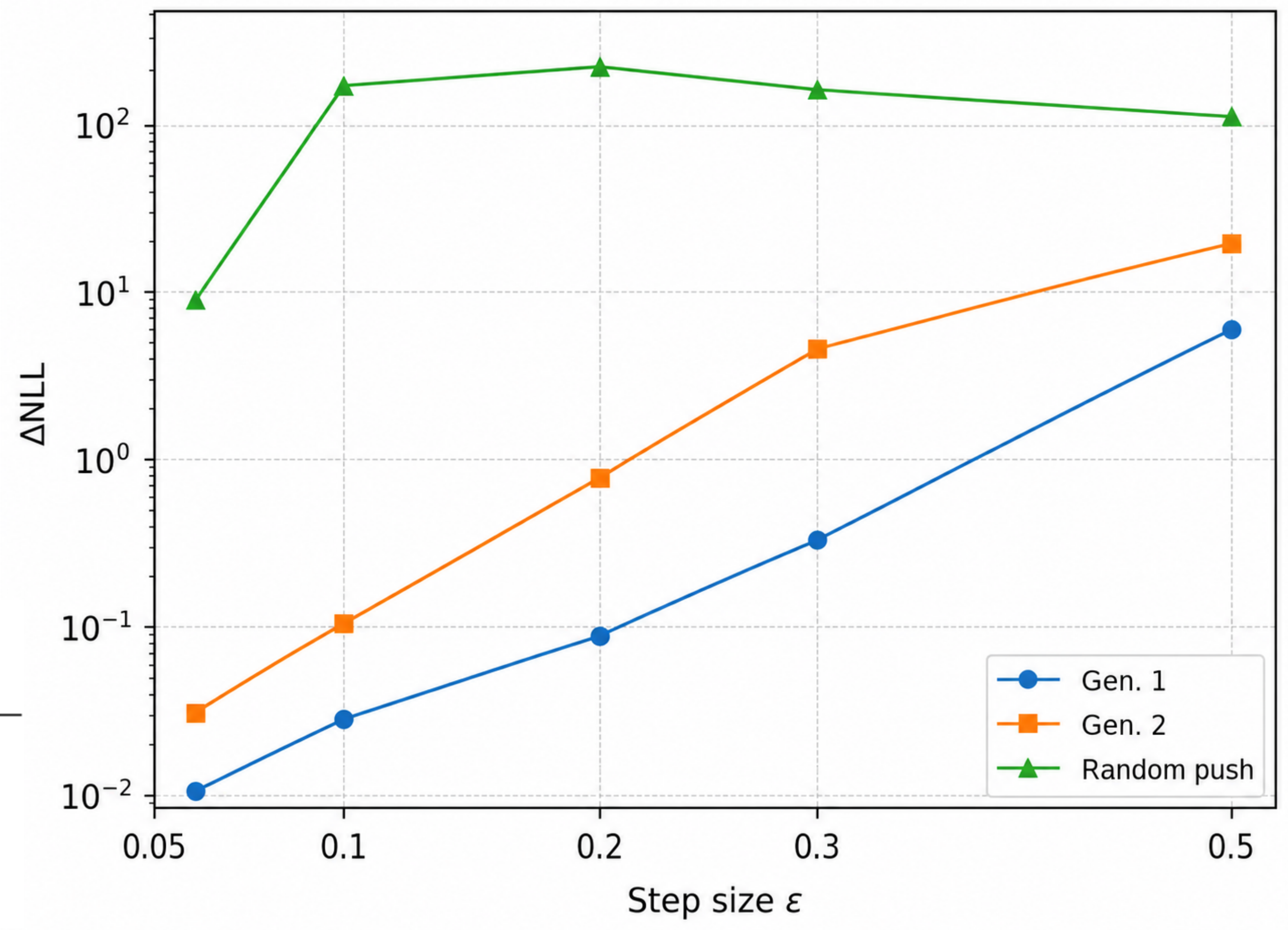
$$\Phi_\epsilon = \exp(\epsilon X_i),\ (t, x) \mapsto (\tilde{t}, \tilde{x})$$

$$(\hat{f}_{pushed}, \hat{\sigma}_{pushed}) \approx (f, \sigma)(\tilde{t}, \tilde{x})$$

Push trajectories along learned generator flow; refit/compare drift and diffusion.

ε	max drift/diffusion residual
$1, 3$	$< 10^{-3}$
5	$< 3 \times 10^{-3}$

Empirical BTC/USDT: learned push vs random push



Generalization checks

- **Analytic SDEs:** small drift/diffusion after-push residuals.
- **BTC/USDT data:** learned pushed preserve stochastic residual distribution better than random pushes.
- **Ablations:** removing any of the loss functions worsens recovery.

Result: learned generators preserve stochastic dynamics, not only local equations.

Contributions, Applications, and Future Directions

Contributions

- SDE symmetry enters ML - first ML framework centered on continuous Lie-point symmetries of stochastic dynamics.
- First SDE symmetry discovery pipeline
- Template-free symmetry discovery - no prescribed group, coordinates, or symbolic templates; works at both SDE and density/FP level.

Downstream applications

- Symmetry-constrained stochastic surrogates
- Data-efficient modeling of noisy systems
- Interpretable stochastic dynamics
- Symmetry-aware simulation/sampling

Future directions

- Build equivariant SDE models
- Scale to high-dimensional real systems
- Go beyond projectable symmetries
- Use symmetries for control and design

Takeaway: LieStoNet turns stochastic symmetry from a theoretical object into a learnable structure for modeling, interpretation, and control.

Thanks!