Feedback Favors the Generalization of Neural ODEs

Oral Presentation

Jindou Jia*, Zihan Yang*, Meng Wang, Kexin Guo,
Jianfei Yang, Xiang Yu, Lei Guo

Beihang University

Nanyang Technological University

1. Motivation

Generalization problem hinders the application of NN-based methods !

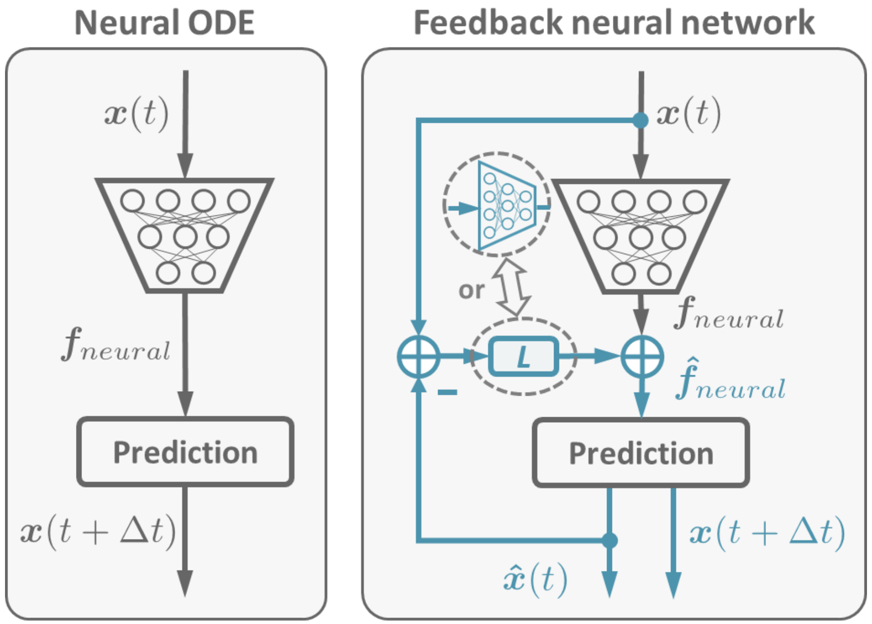
Predominant solution
- Large training dataset
- Large model
- Large training time

Continuous-time tasks, like robots...

- Domain randomization[1]
  Randomize *sim* parameters;
  Trade precise for robust, similar to *robust control*[3].
- Domain adaptation[2]
  Identify uncertainty and adjust decision;
  Maintain precise, similar to *adaptive control*[3].

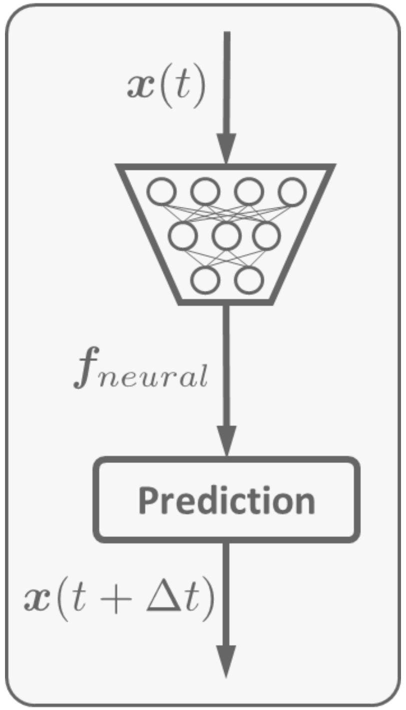
All Need Large Resources !

Our contribution



[1] Tobin et al., *ICRA*, 2018.
[2] Kouw et al., *T-PAMI*,2019.
[3] Ha et al., *IJRR*, 2025.

2. Neural ODE and learning residual



Considering a general ODE:

$$\frac{d\boldsymbol{x}(t)}{dt} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{I}(t), t)$$

Training Neural ODEs with external input $I(t)$

Appendix A.1

Neural ODE[1]:

Reverse-mode automatic differentiation

Given trajectories $\{\boldsymbol{x}(t_1, t_2, \cdots, t_N)\}$ → Latent dynamics $\boldsymbol{f}(t)$

Prediction:

$$x(t+\Delta t) = x(t) + \int_t^{t+\Delta t} \boldsymbol{f}(\boldsymbol{x}(\tau), \boldsymbol{I}(\tau), \tau) d\tau$$

Learning residual:

$$\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{I}(t), t) = \boldsymbol{f}_{\boldsymbol{neural}}(\boldsymbol{x}(t), \boldsymbol{I}(t), t, \boldsymbol{\theta}) + \Delta \boldsymbol{f}(t)$$

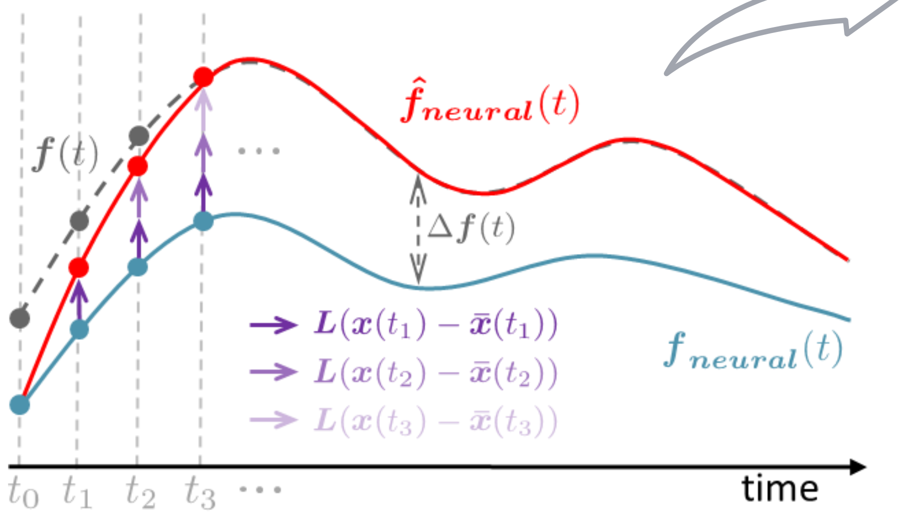
out of distribution

Bounded assumption: $\|\Delta \boldsymbol{f}(t)\| \le \gamma$

[1] Chen et al., Neural ordinary differential equations, *NeurIPS*, 2018.

3. Linear feedback

Correcting latent dynamics:



$$\hat{f}_{neural}(t) = f_{neural}(t) + \sum_{i=1}^{k} L(x(t_i) - \bar{x}(t_i))$$

Accumulated errors

L – positive-definite gain

$\bar{x}(t_i)$ - last prediction → $\bar{x}(t_i) = x(t_{i-1}) + T_s \hat{f}_{neural}(t_{i-1})$

Define $\hat{x}(t) = \bar{x}(t) - \sum_{i=1}^{k-1}(x(t_i) - \bar{x}(t_i))$

Achieve $\begin{cases} \hat{f}_{neural}(t) = f_{neural}(t) + L(x(t) - \hat{x}(t)) \\ \hat{x}(t_k) = \hat{x}(t_{k-1}) + T_s \hat{f}_{neural}(t_{k-1}) \end{cases}$

Convergence analysis:

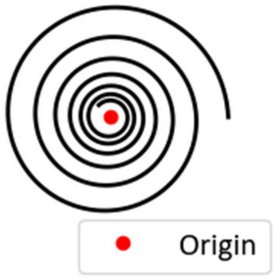
Define $\begin{cases} \tilde{x}(t) = x(t) - \hat{x}(t) \\ \tilde{f}(t) = f(t) - \hat{f}_{neural}(t) \end{cases}$ ⇒ Error dynamics $\dot{\tilde{x}}(t) = -L\tilde{x}(t) + \Delta f(t)$ ⇒ Converged sets

$$\{\tilde{x}(t) \in \mathbb{R}^n : \|\tilde{x}(t)\| \le \gamma/\lambda_m(L)\}$$
$$\{\dot{\tilde{x}}(t) \in \mathbb{R}^n : \|\dot{\tilde{x}}(t)\| \le \gamma\lambda_M(L)/\lambda_m(L) + \gamma\}$$

3. Linear feedback

Toy example – spiral curve

(a) The training set

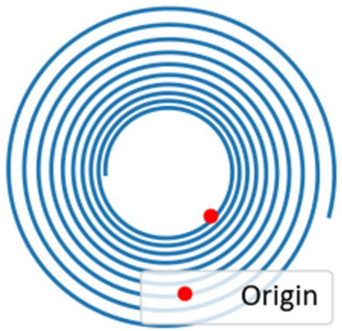


$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \boxed{\begin{bmatrix}-0.1 & 2\\ -2 & -0.1\end{bmatrix}}\begin{bmatrix}x\\ y\end{bmatrix}$$

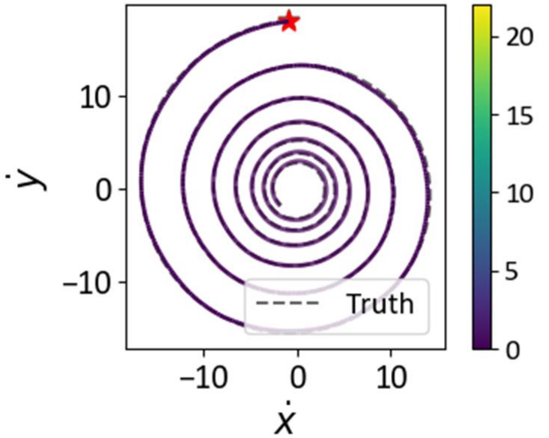
Unknown

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}-0.05 & 3\\ -3 & -0.05\end{bmatrix}\begin{bmatrix}x\\ y\end{bmatrix} + \begin{bmatrix}10\\ 10\end{bmatrix}$$

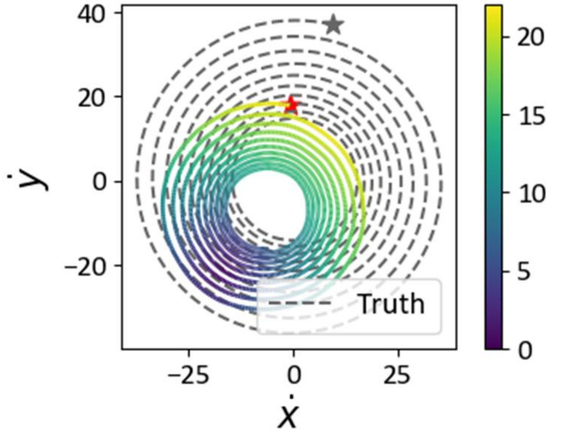
(b) The test set



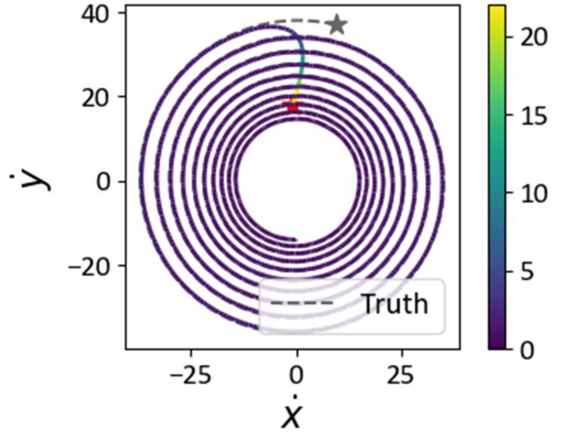
(c) Trained performance



(d) Testing performance of Neural ODE



(e) Testing performance of Feedback NN

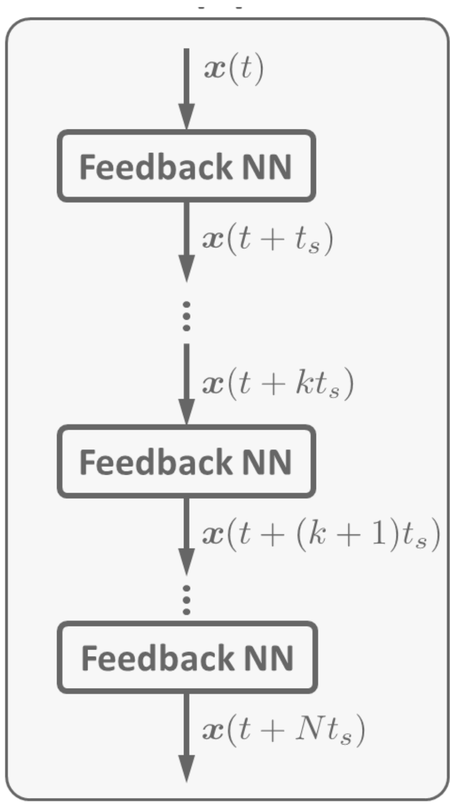


The learnt latent dynamics is corrected accurately

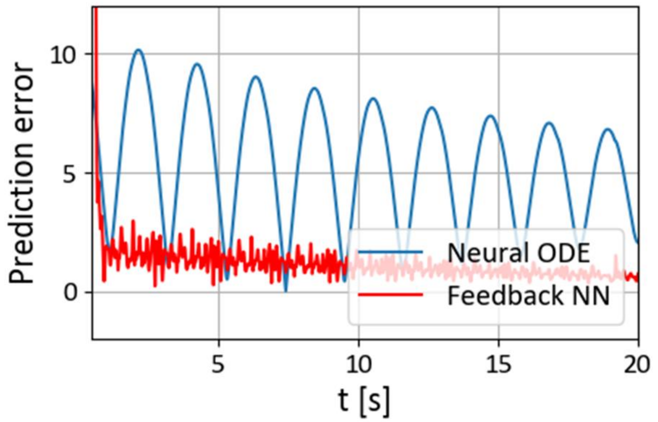
3. Linear feedback

Multi-step prediction:

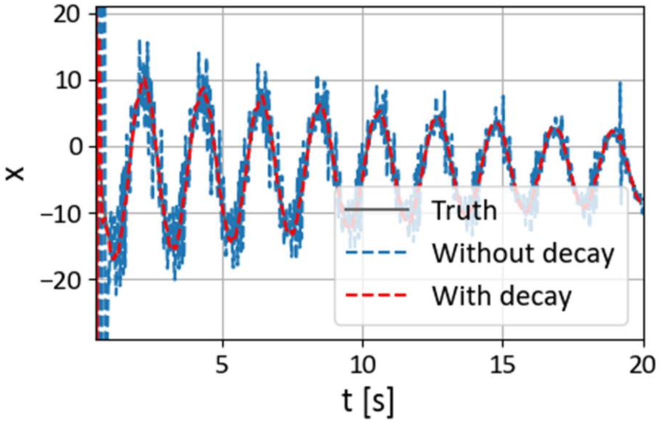
$N = 50$



(f) Multi-step prediction errors in testing



(g) Performance with decay strategy

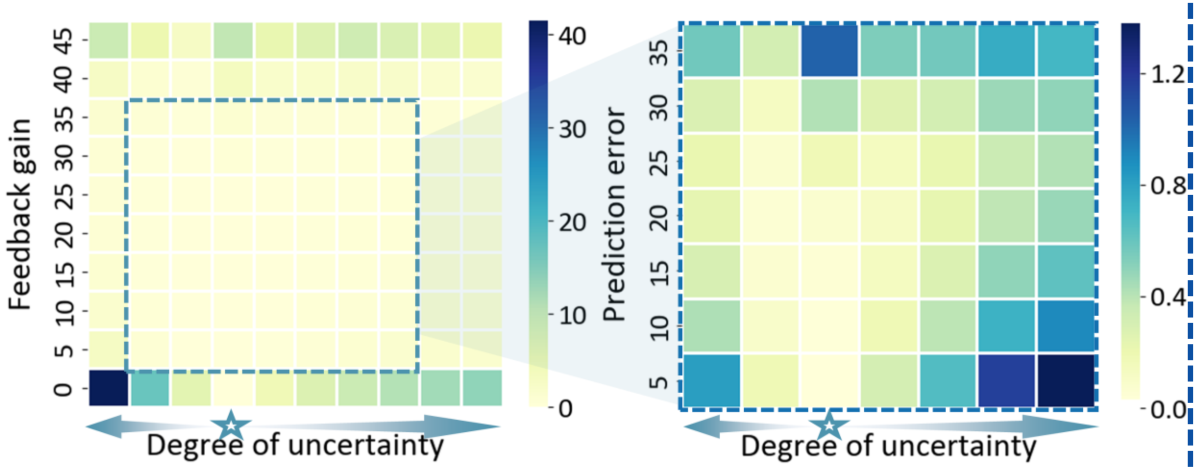


Decaying gains

$$\boldsymbol{L}_{\mathrm{i}} = \boldsymbol{L} \odot e^{-\beta i}$$

Ablation study on linear gain:

Different levels of gains and uncertainties

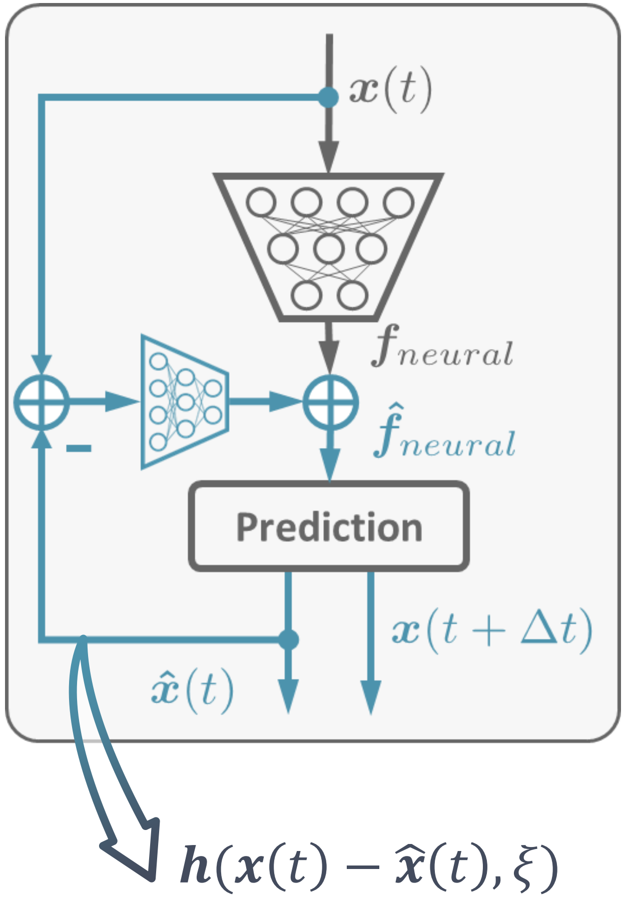


Uncertainty ↑ Prediction error ↑

Gain ↑ Prediction error ↓

Gain ↑↑ Prediction error ↑

4. Neural feedback



$\boldsymbol{h}(\boldsymbol{x}(t) - \widehat{\boldsymbol{x}}(t), \xi)$

nonlinear form

Training procedure:

$$\boldsymbol{\xi}^* = \arg\min_{\boldsymbol{\xi}} \sum_{i=1}^{n_{case}} \sum_{j\in\mathcal{D}_i^{tra}} \left\| \boldsymbol{x}_{i,j}^* - \boldsymbol{x}_{i,j} \right\|$$

$$s.t. \quad \boldsymbol{x}_{i,j} = \boldsymbol{x}_{i,j-1} + T_s\left(\boldsymbol{f}_{neural}(\boldsymbol{x}_{i,j-1}) + \boldsymbol{h}_{neural}\left(\boldsymbol{x}_{i,j-1} - \hat{\boldsymbol{x}}_{i,j-1}, \boldsymbol{\xi}\right)\right)$$

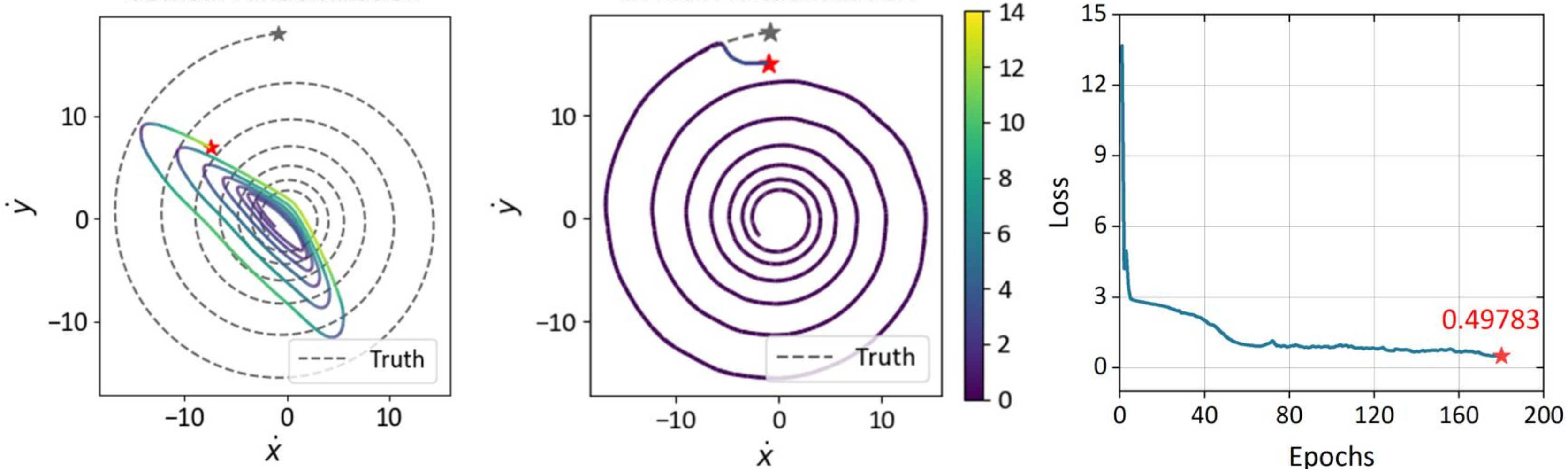
with n_{case} randomized cases through domain randomization.

Training results:

(a) Training Neural ODE through domain randomization

(b) Training feedback part through domain randomization

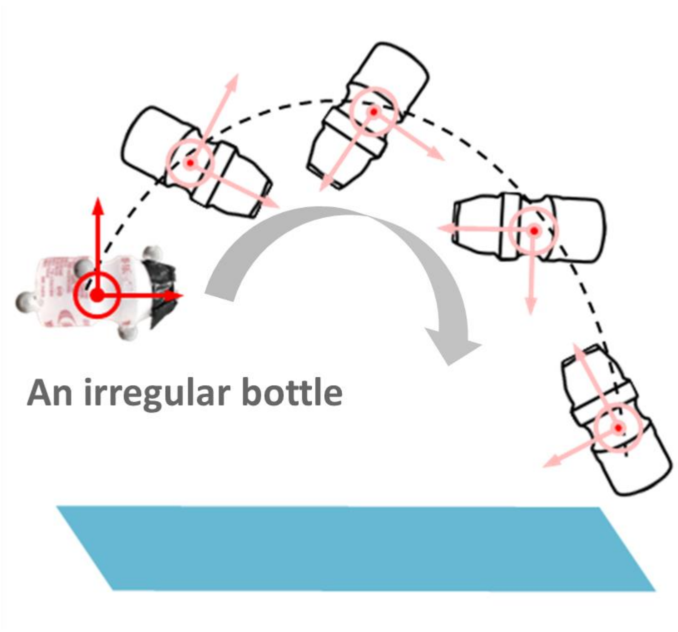
(c) Evaluation of training loss

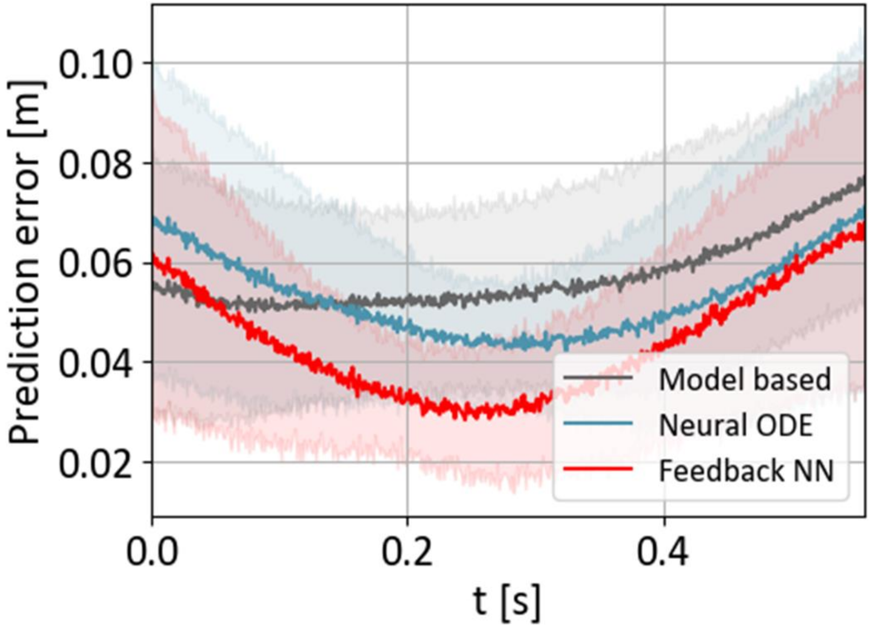


5. Empirical study – Irregular object

Task: Trajectory prediction of an **irregular** object

Challenge: The aerodynamic drag is **intractable**





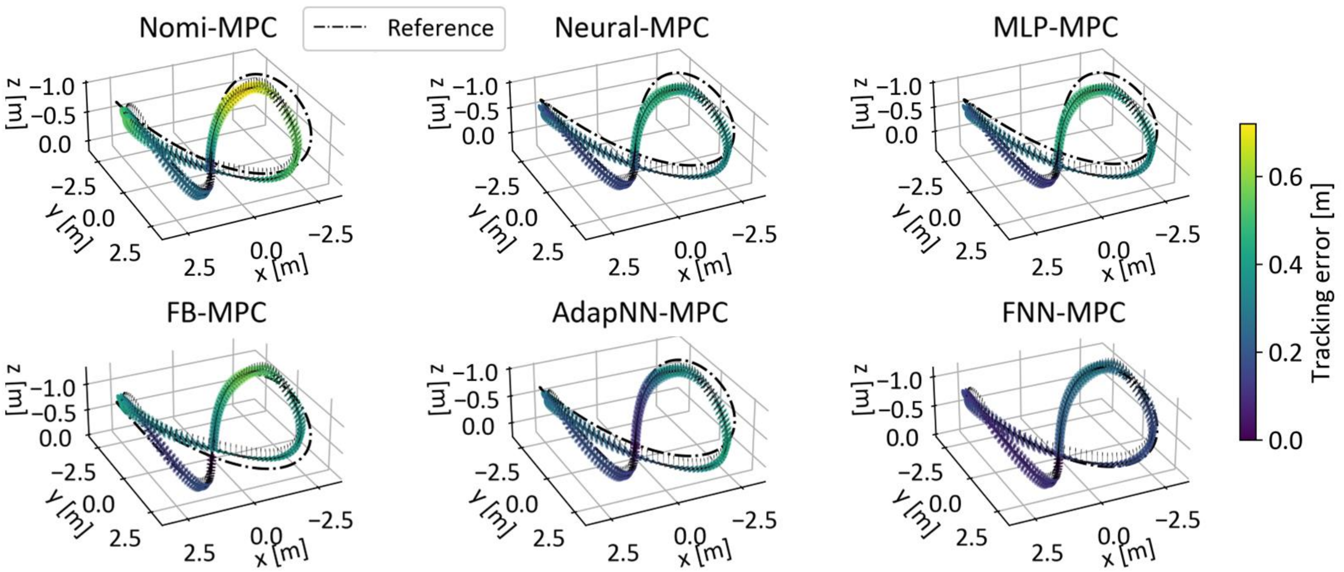
- 21 trajectories for training
- 9 trajectories for testing
- Prediction horizon - $0.5\ s$
- Compared: model-based[1] and learning-based[2]

[1]Muller et al., Quadrocopter ball juggling, *IROS*, 2011.
[2] Chen et al., Neural ordinary differential equations, *NeurIPS*, 2018.
[3] Jia et al., EVOLVER: Online learning and prediction of disturbances for robot control, *TRO*, 2024.

5. Empirical study – Quadrotor flight

Task: Agile trajectory tracking of quadrotor under **multiple disturbances**

Challenge: MPC needs an **accurate** dynamics



	Nomi-MPC	Neural-MPC	MLP-MPC	FB-MPC	AdapNN-MPC	**FNN-MPC**
RMSE [m]	0.248	0.167	0.182	0.203	0.151	**0.093**

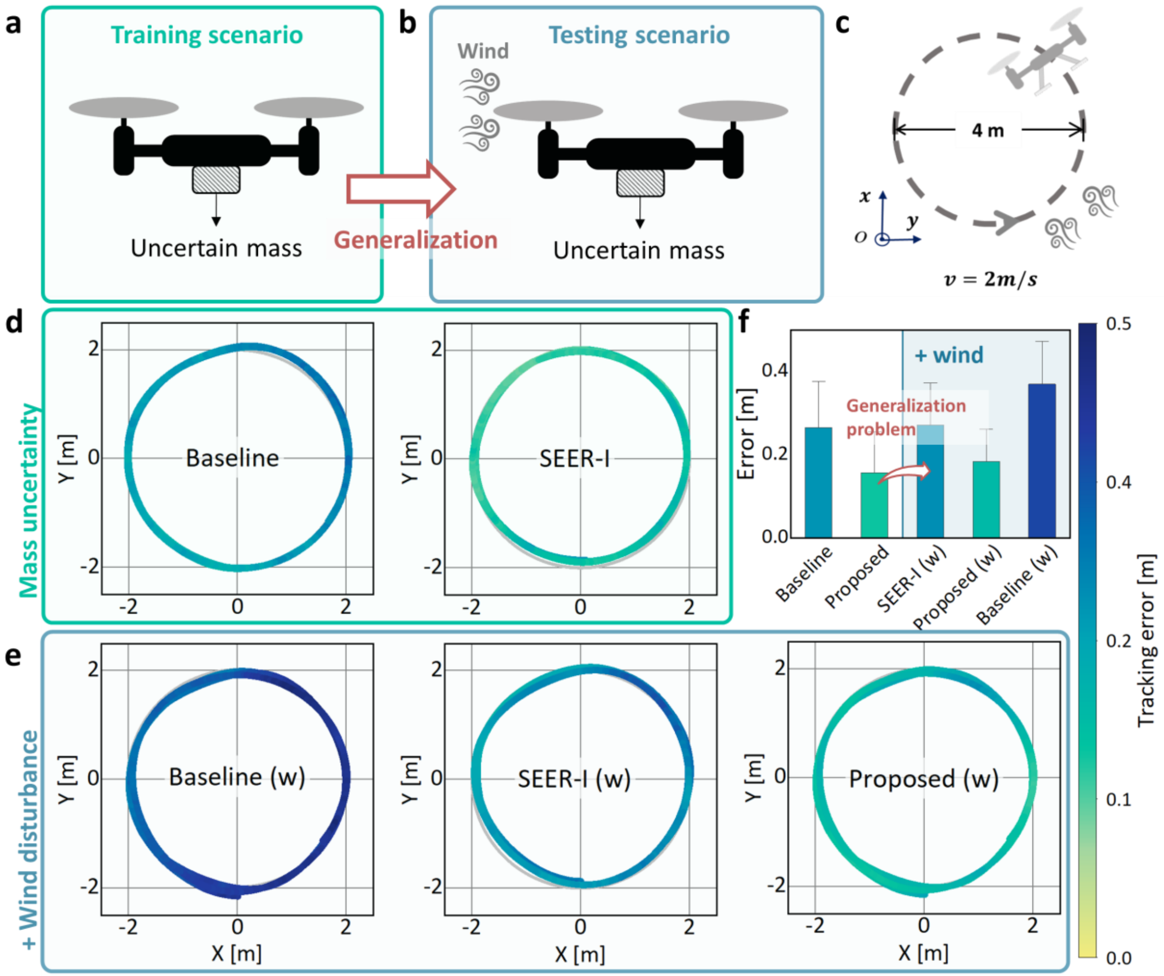
- Learning aerodynamics drag
- Testing under other disturbances like *mass, inertia, drag, force*…
- Prediction horizon - 10
- Compared: model-based, learning-based[1, 2], and adaptive learning-based [3]

[1] Chen et al., Neural ordinary differential equations, *NeurIPS*, 2018.
[2]Saviolo et al., Learning quadrotor dynamics for precise, safe, and agile flight control, *ARC*, 2011.
[3] Cheng et al., Human motion prediction using semi-adaptable neural networks, *TRO*, 2024.

5. Empirical study – Extension

Feedback improve the generalization of other learning methods

- Training under mass uncertainty
- Testing with additional wind
- Flight speed 2 m/s
- Wind speed 5m/s



[1] Jia et al., FORESEER: Recognize and utilize uncertainties by integrating data-based learning and symbolic feedback, 2025.

Thanks for your attention!

Paper

Source Code

Project Site