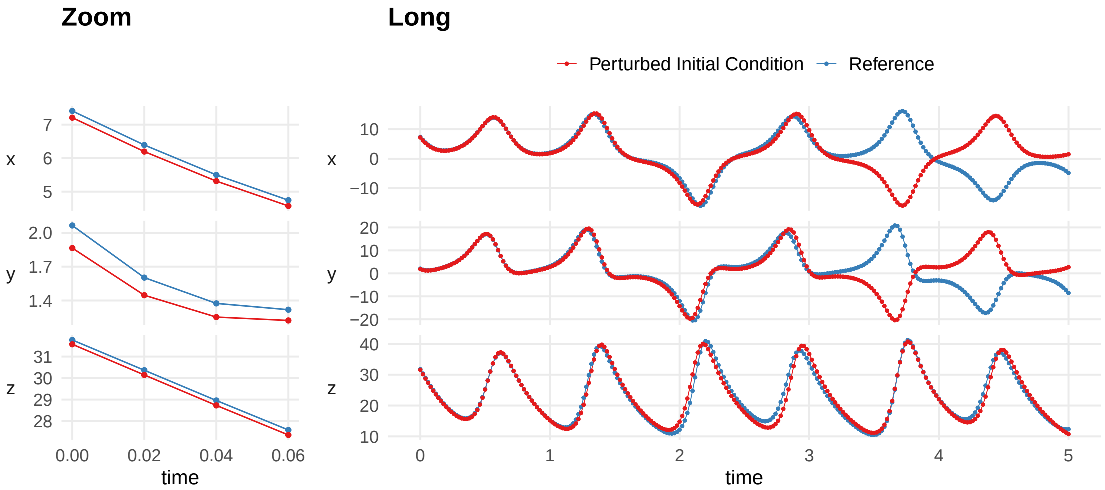
Machine-Precision Prediction of Low-Dimensional Chaotic Systems

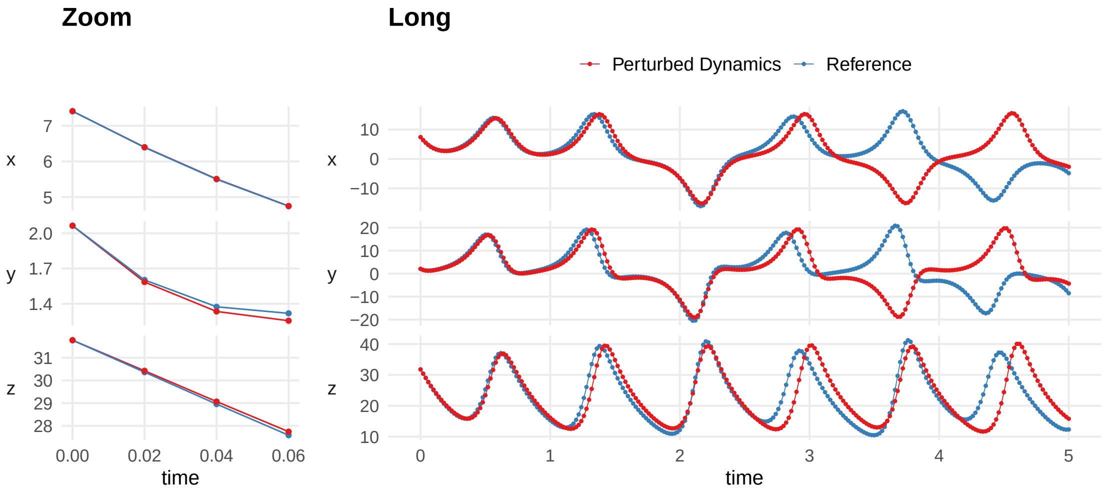
Christof Schötz[1,2] and Niklas Boers[1,2,3]

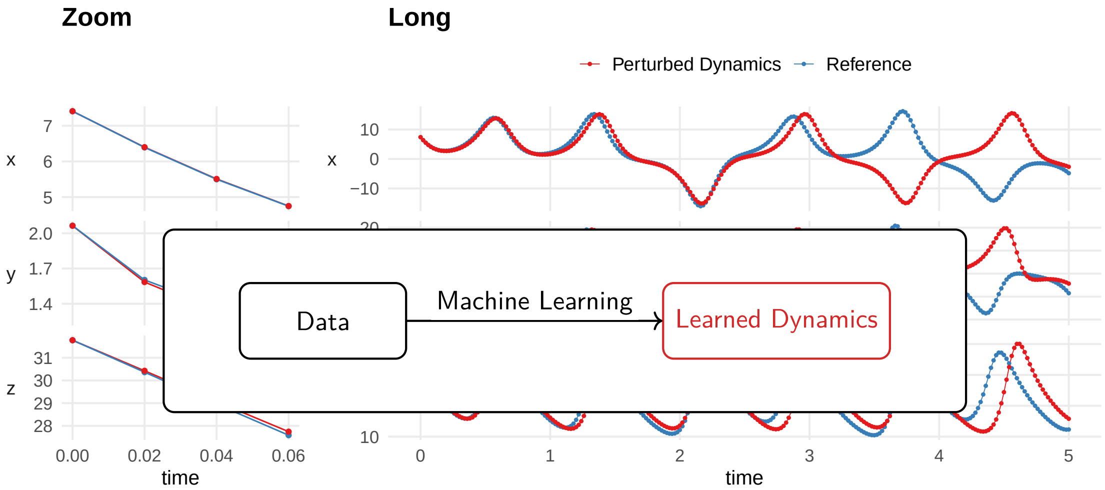
[1]Technical University of Munich
[2]Potsdam Institute for Climate Impact Research
[3]University of Exeter

EGU 2026, Vienna, Austria

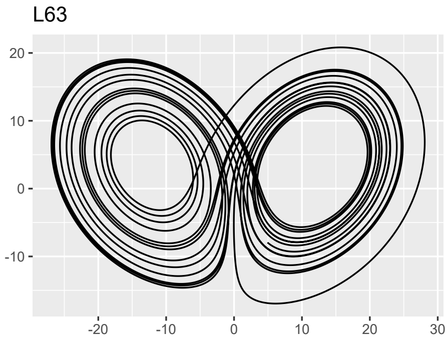
Sharing is encouraged

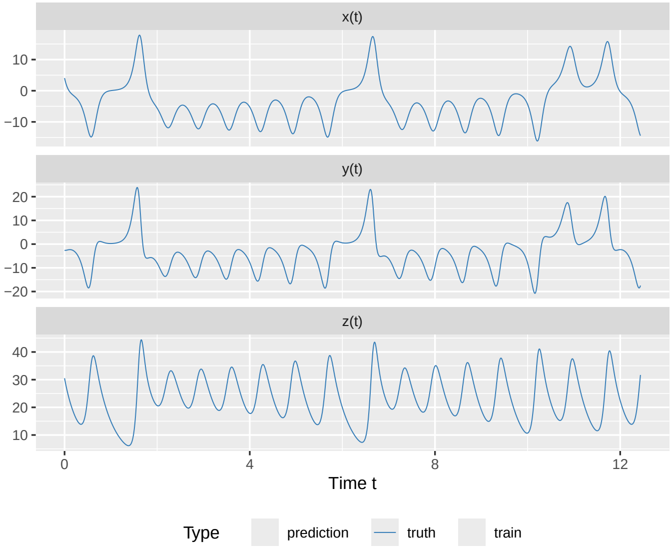

Zoom
Long
Perturbed Initial Condition
Reference
x
y
z
7
6
5
2.0
1.7
1.4
31
30
29
28
0.00
0.02
0.04
0.06
time
10
0
−10
20
10
0
−10
−20
40
30
20
10
0
1
2
3
4
5
time


Zoom
Long
Perturbed Dynamics
Reference
x
y
z
7
6
5
2.0
1.7
1.4
31
30
29
28
0.00
0.02
0.04
0.06
time
10
0
−10
20
10
0
−10
−20
40
30
20
10
0
1
2
3
4
5
time

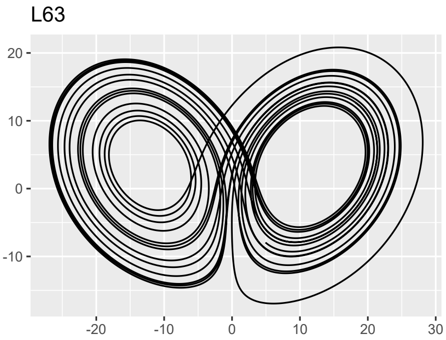

Zoom
Long
Perturbed Dynamics
Reference
x
y
z
7
6
5
2.0
1.7
1.4
31
30
29
28
10
0
−10
10
0.00
0.02
0.04
0.06
time
0
1
2
3
4
5
time
Data
Machine Learning
Learned Dynamics

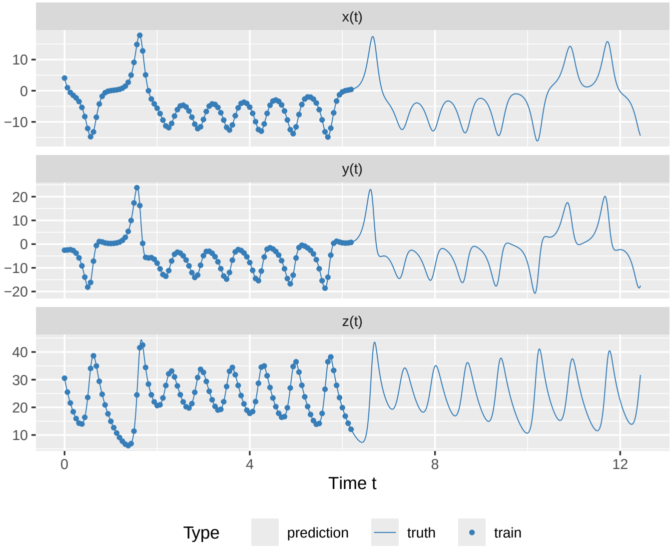
$$\dot{x}(t) = 10(y(t) - x(t))$$
$$\dot{y}(t) = x(t)(28 - z(t)) - y(t)$$
$$\dot{z}(t) = x(t)y(t) - \frac{8}{3}z(t)$$

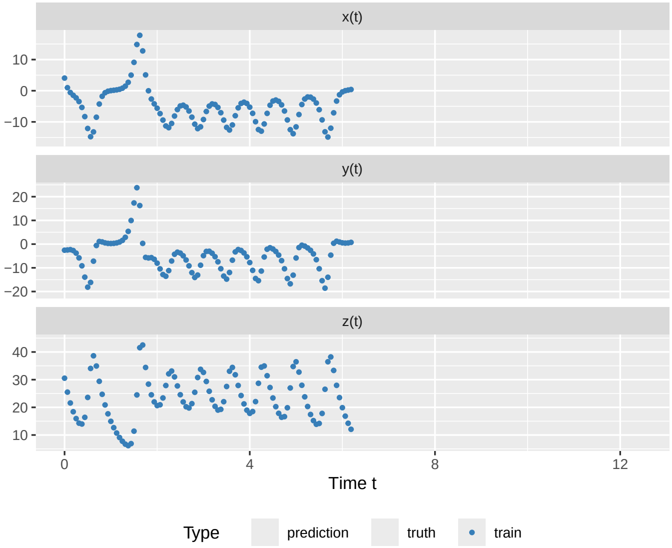




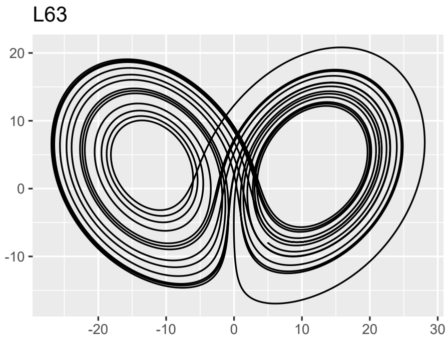
$$\dot{x}(t) = 10(y(t) - x(t))$$
$$\dot{y}(t) = x(t)(28 - z(t)) - y(t)$$
$$\dot{z}(t) = x(t)y(t) - \frac{8}{3}z(t)$$

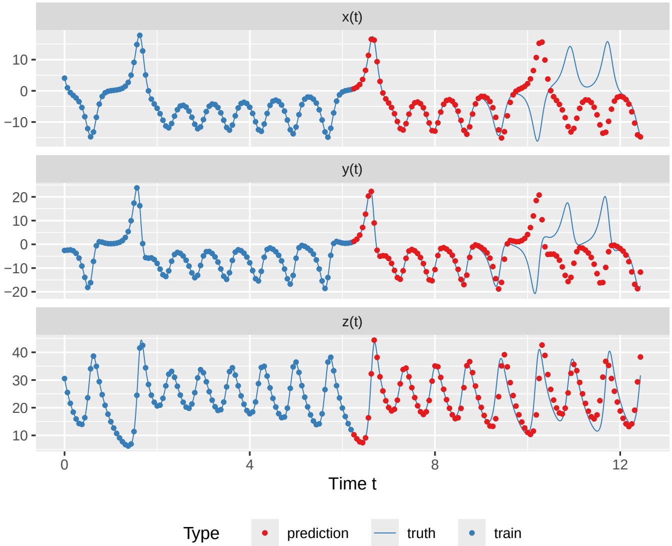




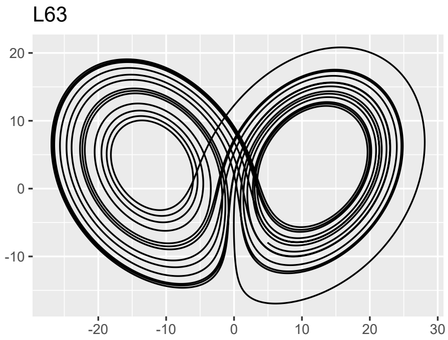

x(t)
10
0
−10
y(t)
20
10
0
−10
−20
z(t)
40
30
20
10
0
4
8
12
Time t
Type
prediction
truth
train

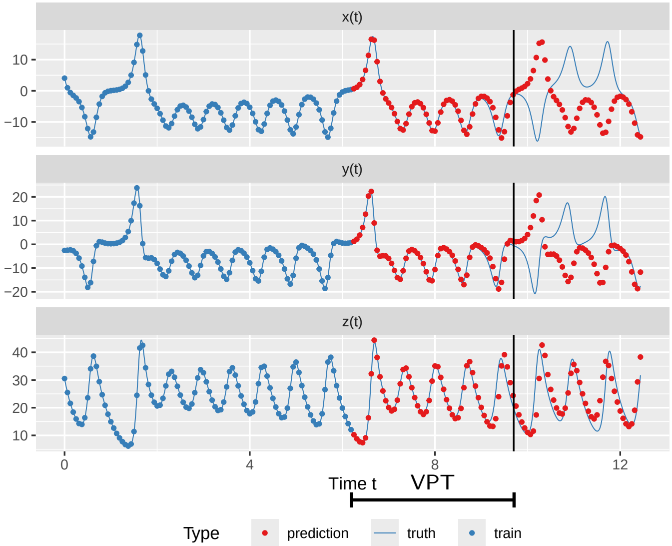
$$\dot{x}(t) = 10(y(t) - x(t))$$
$$\dot{y}(t) = x(t)(28 - z(t)) - y(t)$$
$$\dot{z}(t) = x(t)y(t) - \frac{8}{3}z(t)$$



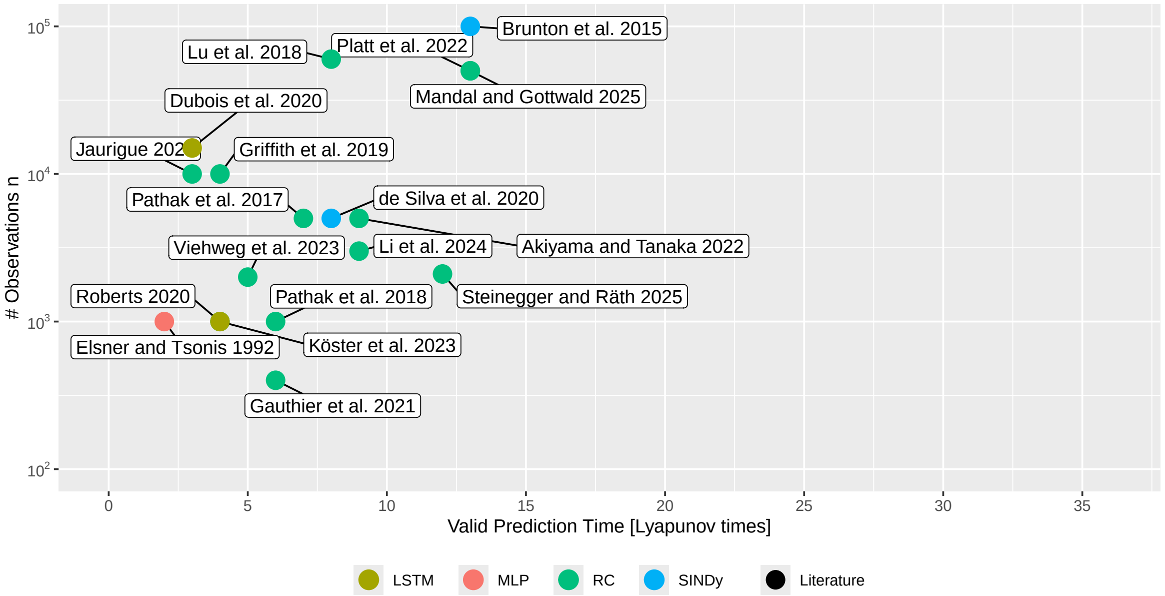


$$\dot{x}(t) = 10(y(t) - x(t))$$
$$\dot{y}(t) = x(t)(28 - z(t)) - y(t)$$
$$\dot{z}(t) = x(t)y(t) - \frac{8}{3}z(t)$$



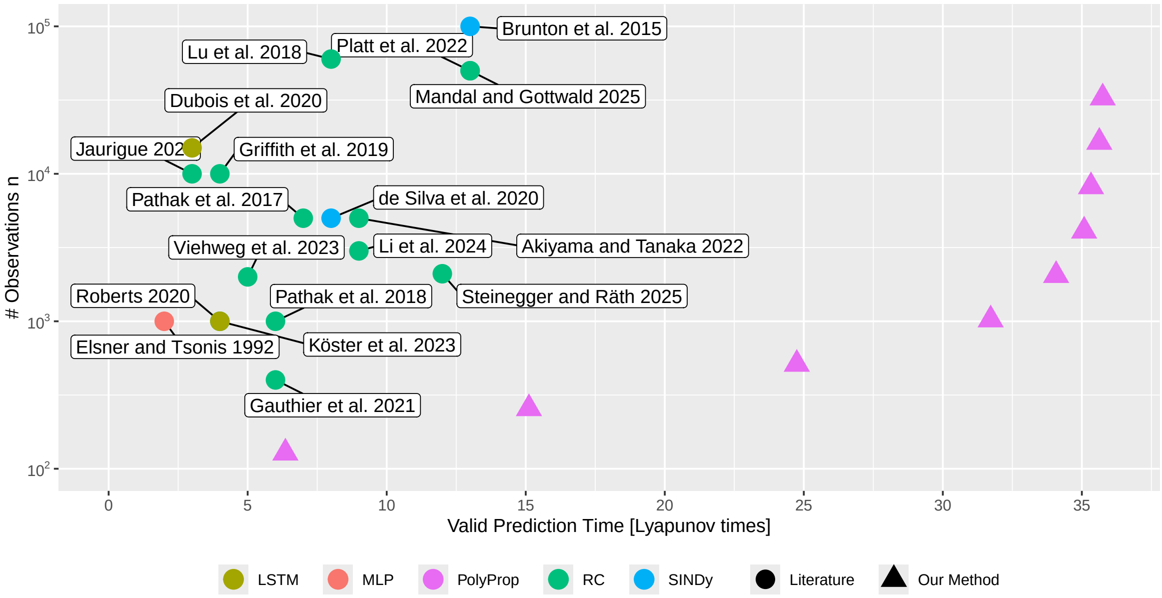


Literature



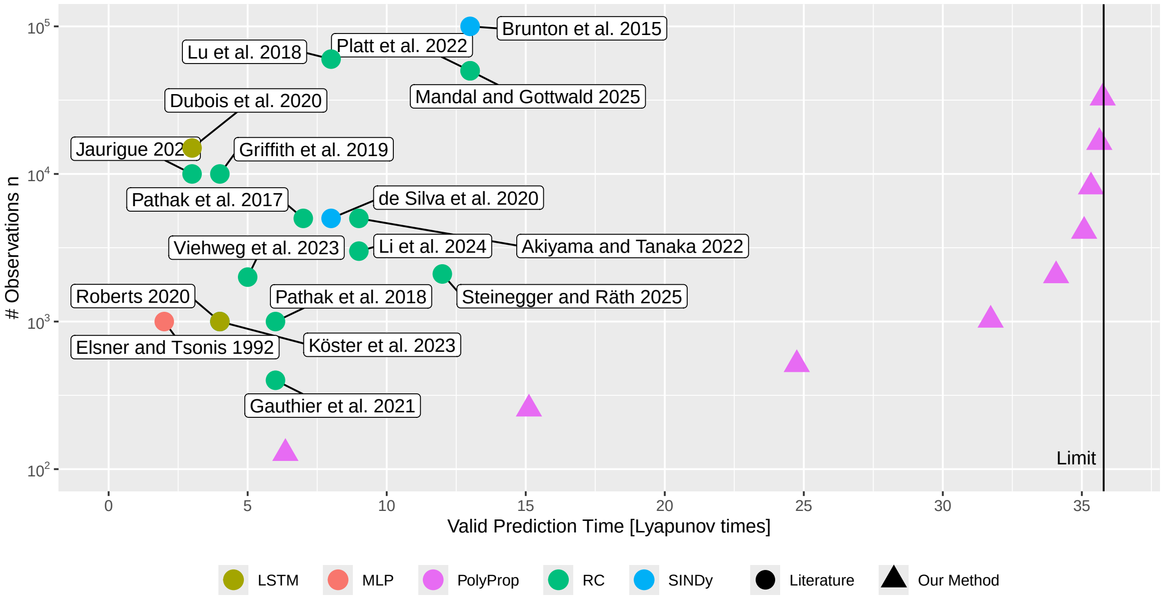
Brunton et al. 2015
Lu et al. 2018
Platt et al. 2022
Mandal and Gottwald 2025
Dubois et al. 2020
Jaurigue 202
Griffith et al. 2019
Pathak et al. 2017
de Silva et al. 2020
Viehweg et al. 2023
Li et al. 2024
Akiyama and Tanaka 2022
Steinegger and Räth 2025
Roberts 2020
Pathak et al. 2018
Elsner and Tsonis 1992
Köster et al. 2023
Gauthier et al. 2021
10^5
10^4
10^3
10^2
Observations n
0
5
10
15
20
25
30
35
Valid Prediction Time [Lyapunov times]
LSTM
MLP
RC
SINDy
Literature

Literature



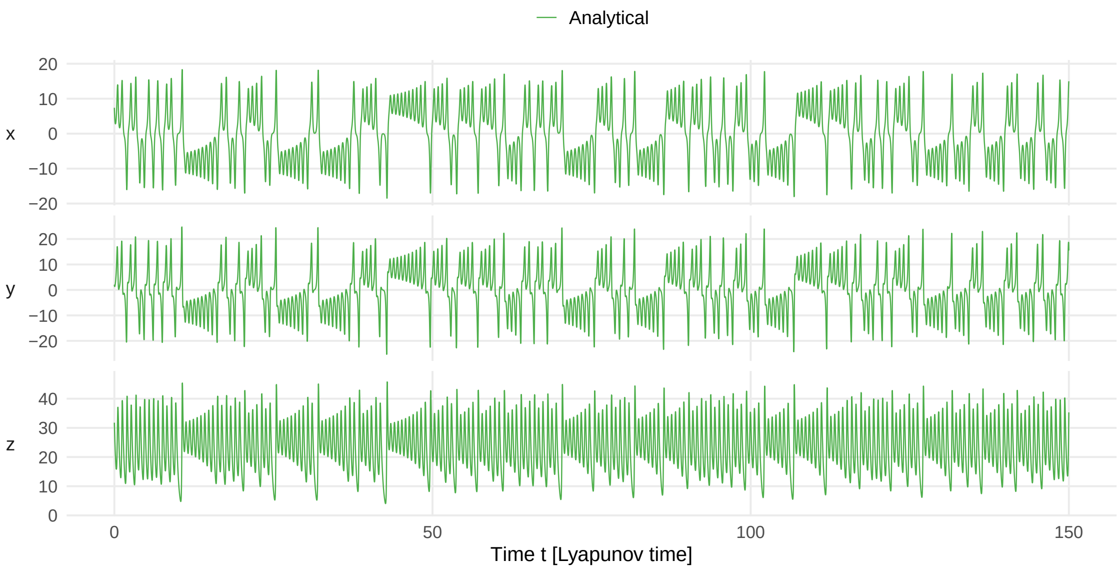
Brunton et al. 2015
Lu et al. 2018
Platt et al. 2022
Mandal and Gottwald 2025
Dubois et al. 2020
Jaurigue 202
Griffith et al. 2019
Pathak et al. 2017
de Silva et al. 2020
Viehweg et al. 2023
Li et al. 2024
Akiyama and Tanaka 2022
Roberts 2020
Pathak et al. 2018
Steinegger and Räth 2025
Elsner and Tsonis 1992
Köster et al. 2023
Gauthier et al. 2021
10^5
10^4
10^3
10^2
Observations n
0
5
10
15
20
25
30
35
Valid Prediction Time [Lyapunov times]
LSTM
MLP
PolyProp
RC
SINDy
Literature
Our Method

Literature

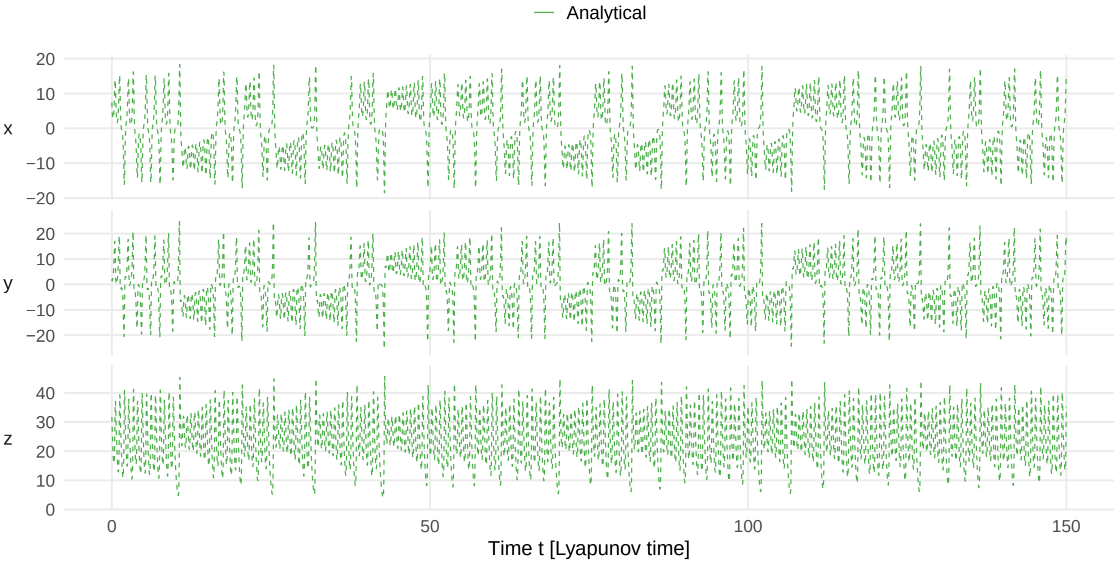


Brunton et al. 2015
Lu et al. 2018
Platt et al. 2022
Mandal and Gottwald 2025
Dubois et al. 2020
Jaurigue 202
Griffith et al. 2019
Pathak et al. 2017
de Silva et al. 2020
Viehweg et al. 2023
Li et al. 2024
Akiyama and Tanaka 2022
Roberts 2020
Pathak et al. 2018
Steinegger and Räth 2025
Elsner and Tsonis 1992
Köster et al. 2023
Gauthier et al. 2021
Limit
Observations n
10^5
10^4
10^3
10^2
0
5
10
15
20
25
30
35
Valid Prediction Time [Lyapunov times]
LSTM
MLP
PolyProp
RC
SINDy
Literature
Our Method

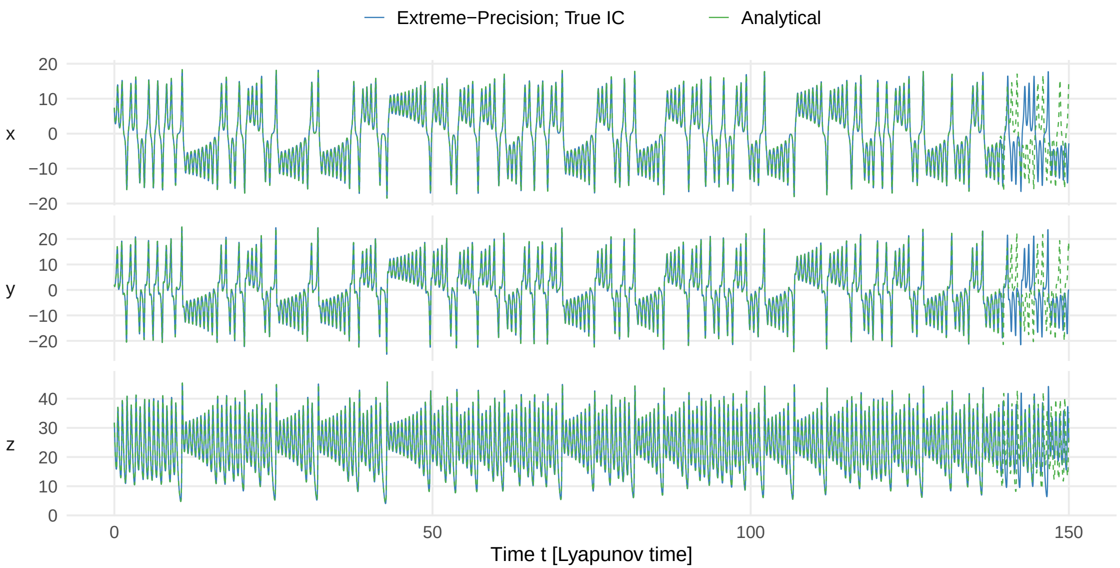
Lorenz 63 – Ground Truth and Machine Precision Limit



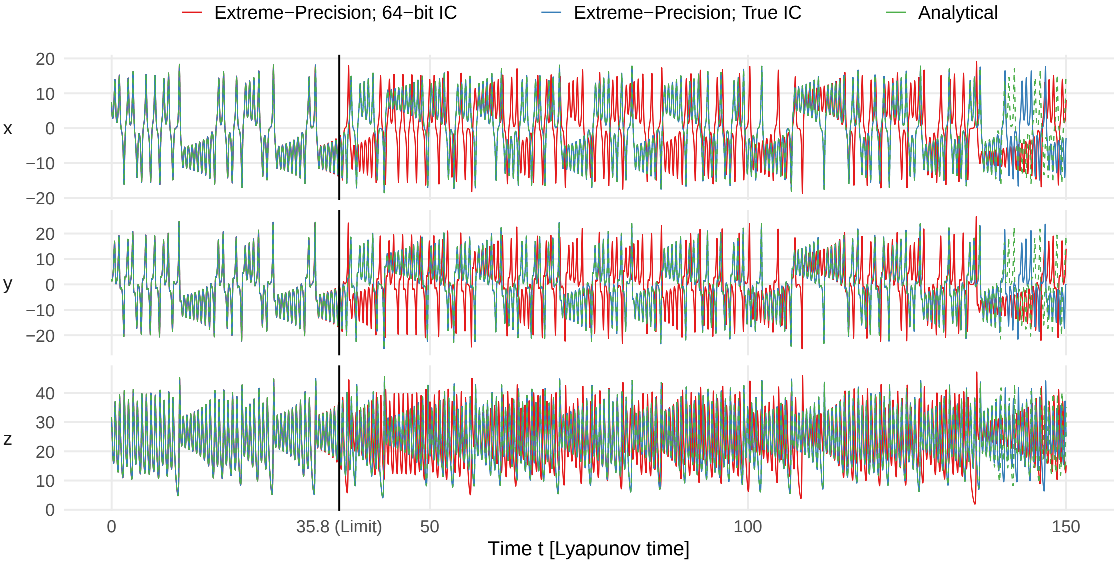
Lorenz 63 – Ground Truth and Machine Precision Limit



Lorenz 63 – Ground Truth and Machine Precision Limit



Lorenz 63 – Ground Truth and Machine Precision Limit



With Initial Condition Error equal to 64-bit Rounding Error:

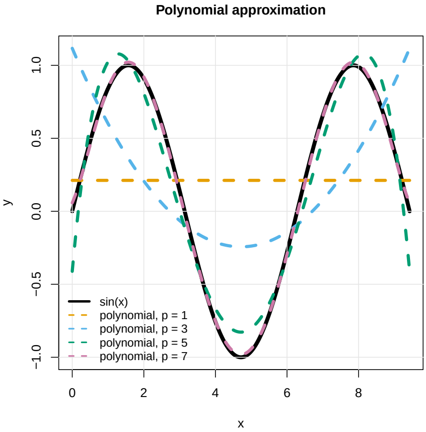
	VPT [Lyapunov time]
Limit	35.8* $\approx -\ln(\mathtt{EPS}_{\mathtt{64bit}})$
PolyProp	35.6*
max precision RK45	31.6
standard RK4	21.6
previous best learned	13

* statistically indistinguishable

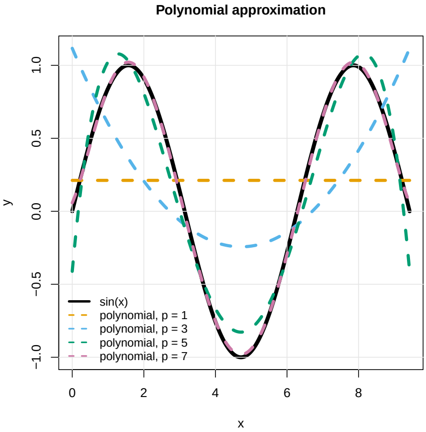
PolyProp does not have access to the ODE but to $n = 2^{14} \approx 16{,}000$ 64-bit observations at time step $\Delta t = 2^{-5} \approx 0.03$

- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$

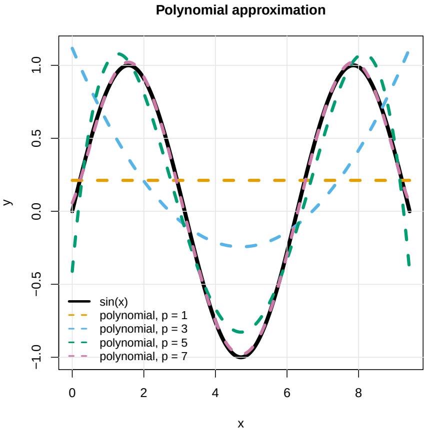
- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$
- **Stone–Weierstrass Theorem** guarantees approximation via **polynomials**



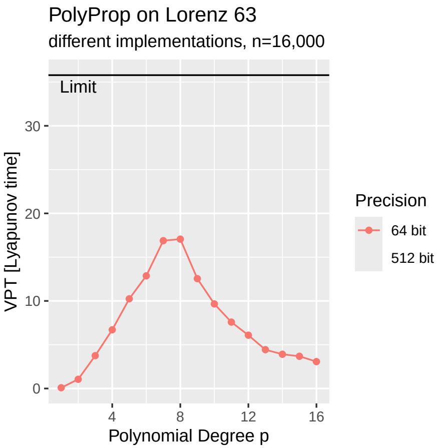
- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$
- **Stone–Weierstrass Theorem** guarantees approximation via **polynomials**
- Polynomial features up to degree p: $\mathrm{poly}_p(x, y, z) = (1, x, y, z, xy, yz, xz, x^2, y^2, z^2, \ldots)$



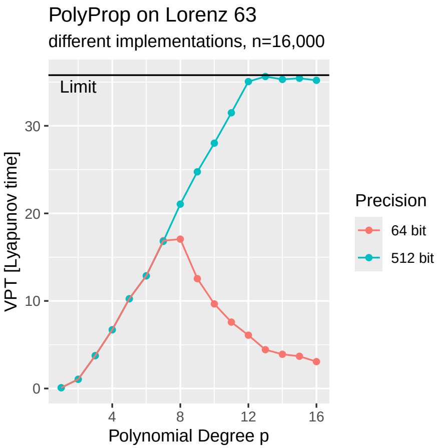
- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$
- **Stone–Weierstrass Theorem** guarantees approximation via **polynomials**
- Polynomial features up to degree p: $\mathrm{poly}_p(x, y, z) = (1, x, y, z, xy, yz, xz, x^2, y^2, z^2, \dots)$
- OLS linear regression: $u(t + \Delta t) = \beta \cdot \mathrm{poly}_p(u(t))$



- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$
- **Stone–Weierstrass Theorem** guarantees approximation via **polynomials**
- Polynomial features up to degree p: $\text{poly}_p(x, y, z) = (1, x, y, z, xy, yz, xz, x^2, y^2, z^2, \ldots)$
- OLS linear regression: $u(t + \Delta t) = \beta \cdot \text{poly}_p(u(t))$
- Naive implementation does not work for $p > 8$: **ill-conditioned!**

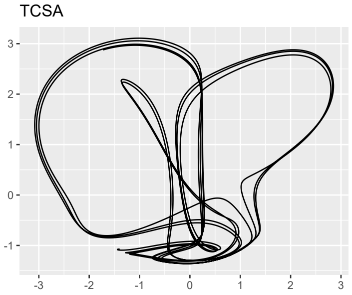


- For $\dot{u}(t) = f(u(t))$ approximate **propagator** $u(t) \mapsto u(t + \Delta t)$
- **Stone–Weierstrass Theorem** guarantees approximation via **polynomials**
- Polynomial features up to degree p: $\text{poly}_p(x, y, z) = (1, x, y, z, xy, yz, xz, x^2, y^2, z^2, \ldots)$
- OLS linear regression: $u(t + \Delta t) = \beta \cdot \text{poly}_p(u(t))$
- Naive implementation does not work for $p > 8$: **ill-conditioned!**
- Solution: Use **512-bit** arithmetic internally (observations stay **64-bit**)



Thomas Cyclically Symmetric Attractor

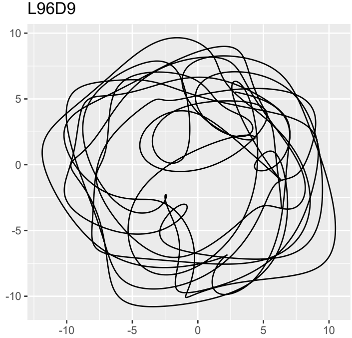
$$\dot{x} = \sin(y) - bx$$
$$\dot{y} = \sin(z) - by \qquad \text{with } b = 0.208$$
$$\dot{z} = \sin(x) - bz$$



Lorenz-96

$$\dot{x}_k = (x_{k+1} - x_{k-2})x_{k-1} - x_k + 8$$

with $x_{-1} = x_{d-1}$, $x_0 = x_d$, $x_{d+1} = x_1$.
Tested in dimensions $d \in \{5, \dots, 9\}$.



Limitations and Conclusion

Limitations

- State dimension: $d < 10$ ok
  $d > 100$ infeasible ($\to$ localization)
- Full state observation required
  (partial obs. $\to$ time delay embedding)
- Noise-free data assumed

Limitations

- State dimension: $d < 10$ ok
  $d > 100$ infeasible ($\rightarrow$ localization)
- Full state observation required
  (partial obs. $\rightarrow$ time delay embedding)
- Noise-free data assumed

Within these limitations,
learning and forecasting
dynamical systems
from **noise-free** observations
is effectively **solved**.

dynamics error $\ll$ initial condition error

Limitations

- State dimension: $d < 10$ ok
  $d > 100$ infeasible ($\rightarrow$ localization)
- Full state observation required
  (partial obs. $\rightarrow$ time delay embedding)
- Noise-free data assumed

Within these limitations,
learning and forecasting
dynamical systems
from **noise-free** observations
is effectively **solved**.

dynamics error $\ll$ initial condition error

Future Reseach:
Focus on **noisy** observations.

Limitations and Conclusion

Limitations

- State dimension: $d < 10$ ok
  $d > 100$ infeasible ($\rightarrow$ localization)
- Full state observation required
  (partial obs. $\rightarrow$ time delay embedding)
- Noise-free data assumed

Within these limitations,
learning and forecasting
dynamical systems
from **noise-free** observations
is effectively **solved**.

dynamics error $\ll$ initial condition error

Future Reseach:
Focus on **noisy** observations.

Thank You!