

# Experimental study of Neural ODE training with adaptive solver for dynamical systems modeling

Alexandre Allauzen<sup>1,2</sup>, Thiago Petrilli Maffei Dardis<sup>2</sup>, Hannah Plath<sup>2</sup>  
<sup>1</sup>LAMSADE CNRS, Dauphine University-PSL, <sup>2</sup>ESPCI-PSL

## TAKE-HOME MESSAGE: WHATCH YOUR STEP SIZE !

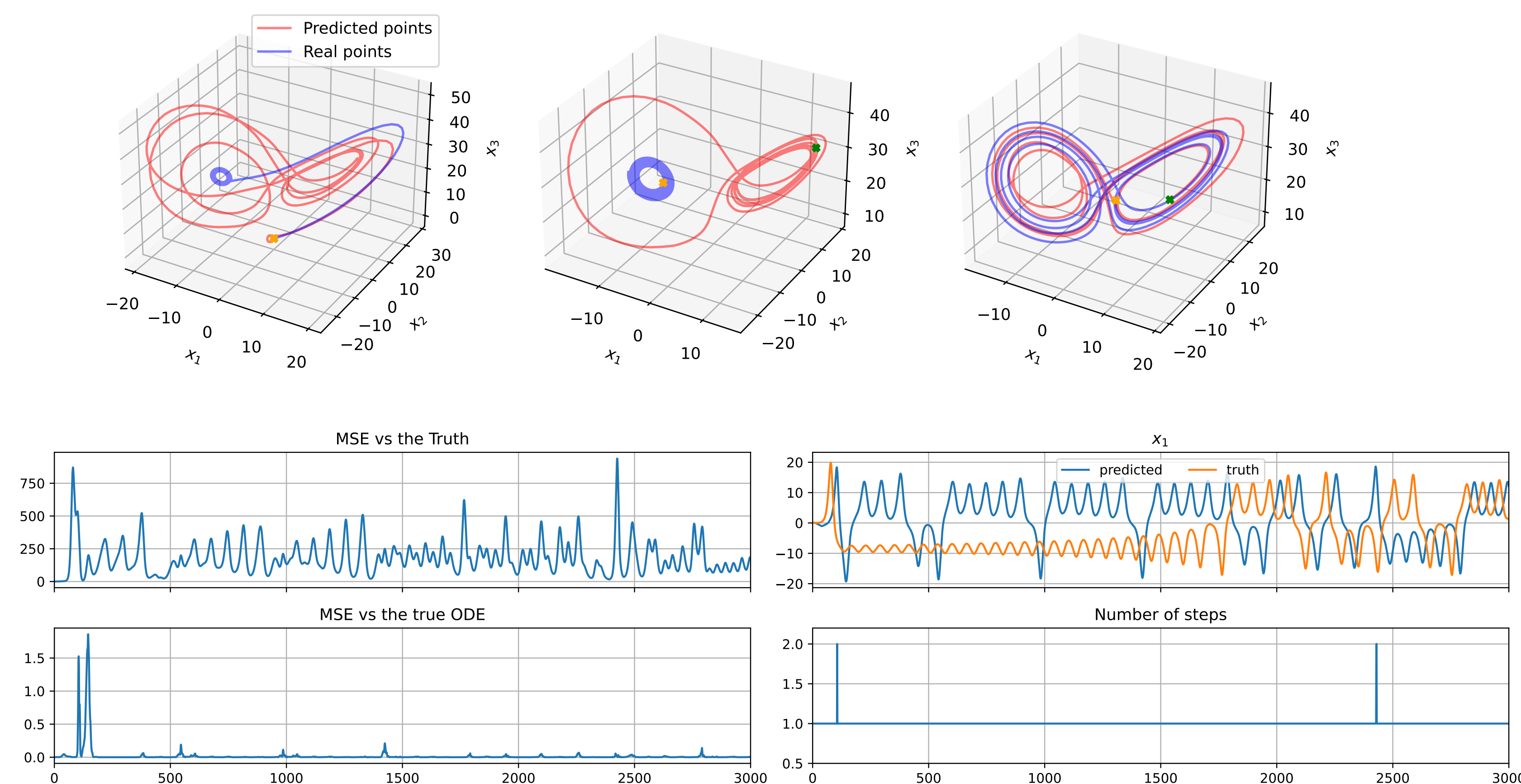
- Neural Ordinary Differential Equations (ODEs) relies on solvers for inference and training.
- Some solvers can **adapt** their evaluation strategy depending on the complexity of the problem.
- By taking the Lorenz'63 system as a showcase, a naive application of the Fehlberg's method does not yield the expected results.
- Adaptive solvers cannot be seamlessly leveraged as a black-box:

- For most of the numerical methods, the step size is adapted given the self-estimated error of the model.
- Neural ODE model is always self-confident, and the “adaptive” ability is under-used.
- A simple workaround:

use the supervision within the solver.

<https://github.com/Allauzen/adaptive-step-size-neural-ode>

## FIRST ROUND



**Figure 1:** Time evolution of (from left to right and top to bottom): the MSE, the evolution of  $x_1$ , the MSE *w.r.t* the true ODE of Lorenz'63, and the number of steps.

## NEURAL ODE AND FEHLBERG'S METHOD

- Given  $\mathcal{D} = (\tilde{\mathbf{x}}_i)_{i=1}^N$ , Neural ODE learns:  $\dot{\mathbf{x}} = f_{\theta}(\mathbf{x})$  ( $f_{\theta}$  is a NNet and  $\theta$  its set of parameters).
- Inference requires a solver:  $\mathbf{x}_i = \text{ODE\_Solve}(f_{\theta}, \tilde{\mathbf{x}}_i)$  and the loss function is

$$\mathcal{L}(\theta, \mathcal{D}) = \sum_{i=1}^N \|\tilde{\mathbf{x}}_i - \mathbf{x}_i\|^2.$$

**Fehlberg's solver:** starting with a step size  $h = 1$ , compute:

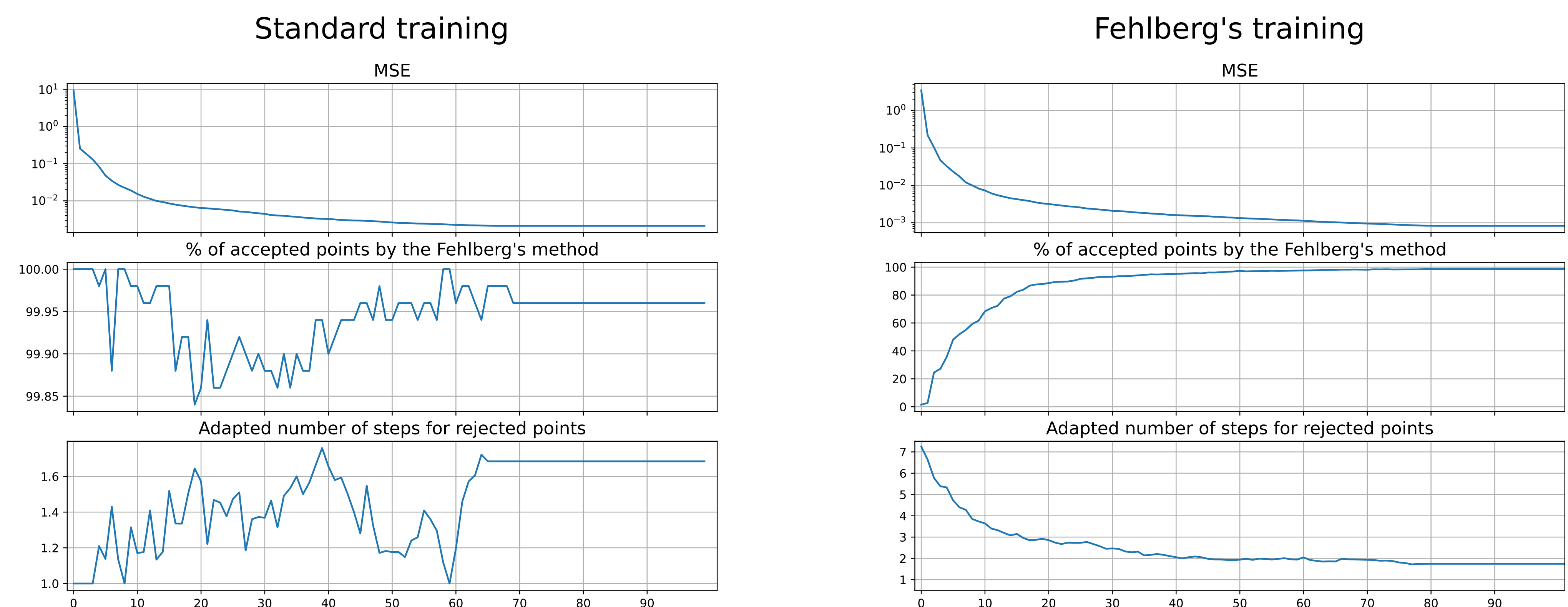
$$f_1 = f_{\theta}(\mathbf{x}_i), \quad f_2 = f_{\theta}(\mathbf{x}_i + hf_1), \quad f_3 = f_{\theta}(\mathbf{x}_i + \frac{h}{4}[f_1 + f_2]) \text{ then}$$

$$A_1 = \mathbf{x}_i + \frac{h}{2}[f_1 + f_2] \text{ (RK2)} \quad A_2 = \mathbf{x}_i + \frac{h}{6}[f_1 + f_2 + 4f_3] \text{ (RK3)}. \quad \rightarrow \quad r = \frac{|A_1 - A_2|}{h} \approx Kh^2$$

With the error rate , then:

- If  $r > \epsilon = 0.1$  (tolerance),  $A_2$  is rejected and restart the computation with a new step size  $h' = S \times h \sqrt{\epsilon/r}$ ,  $S = 0.9$  is a safety factor.
- Otherwise accept  $A_2$  as  $\mathbf{x}_{i+1}$

## FEHLBERG'S TRAINING



**Figure 2:** Time evolution for two training conditions of: the MSE loss; the percentage of accepted hypotheses  $A_2$ ; the new number of steps for the rejected hypotheses (before rounding).

## ACKNOWLEDGMENT

This work was funded by the French National Research Agency (SPEED-20-CE23-0025).