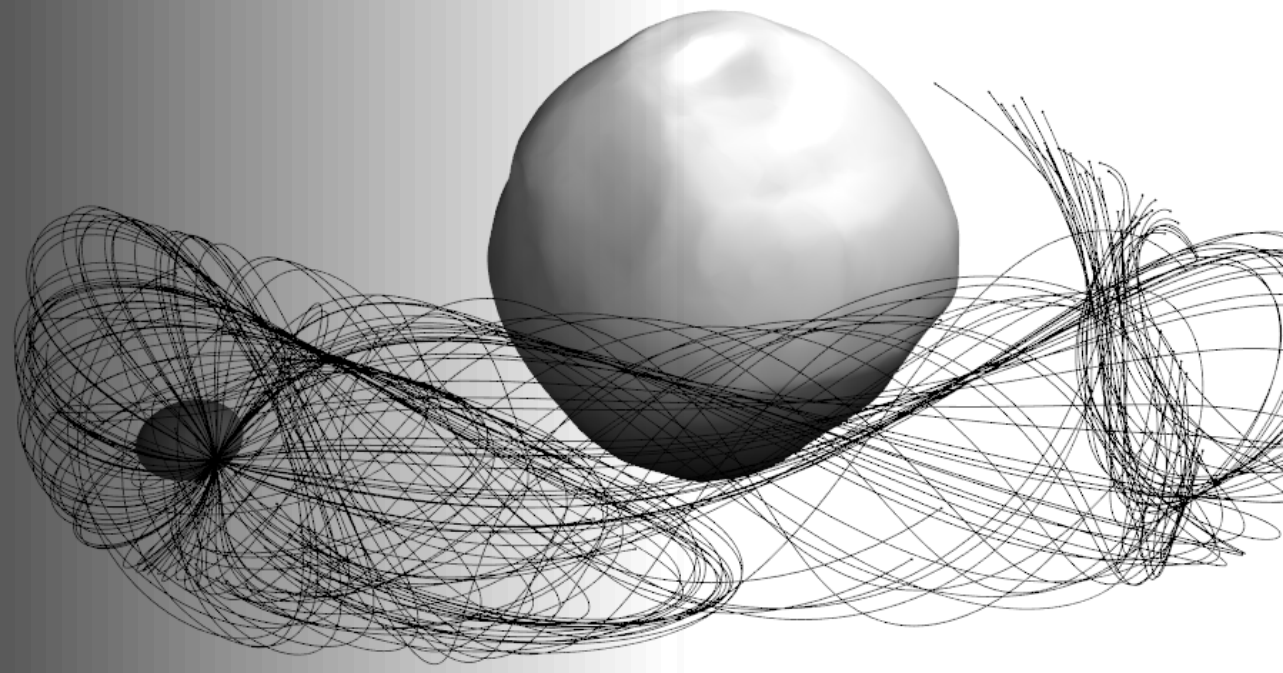
Generative Astrodynamics:

from the generation of periodic solutions to orbit determination

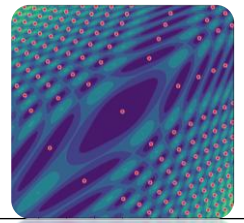
Massimiliano Vasile
Aerospace Centre of Excellence
University of Strathclyde



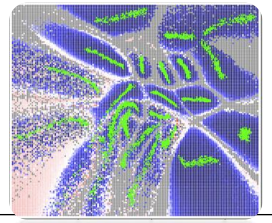
Agenda



Generative Astrodynamics What is it?



Generation of local minima under uncertainty



Generation of Periodic Solutions



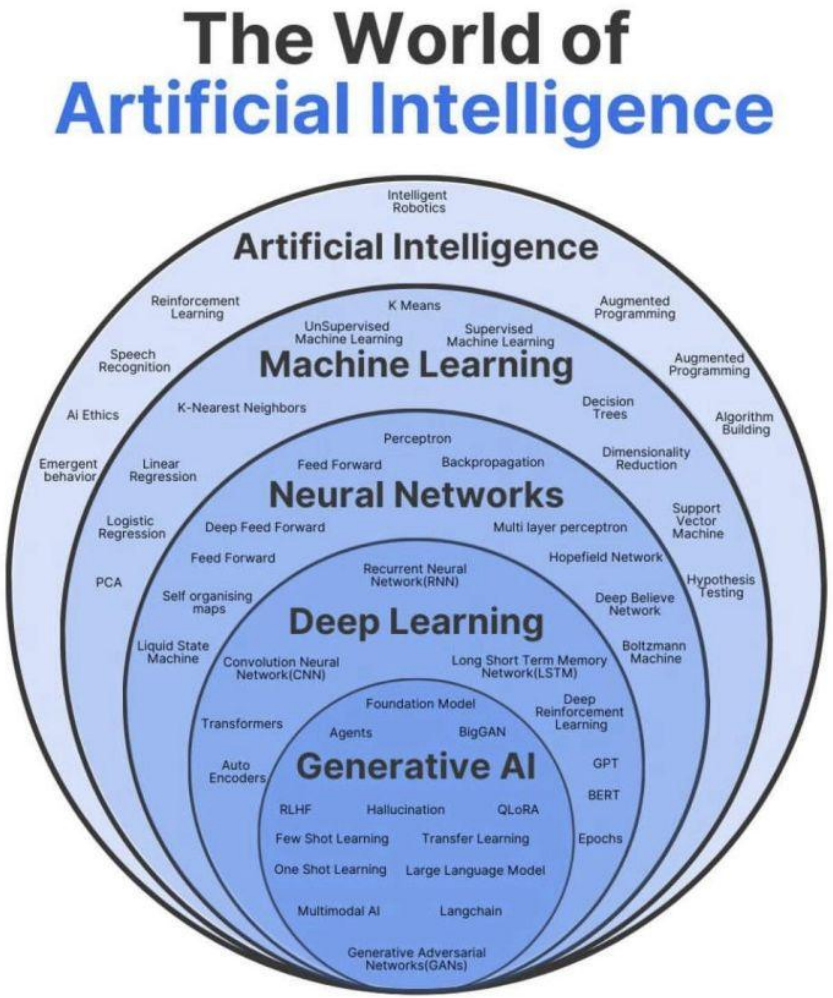
Generative Orbit Determination

Agenda



Generative Astrodynamics What is it?

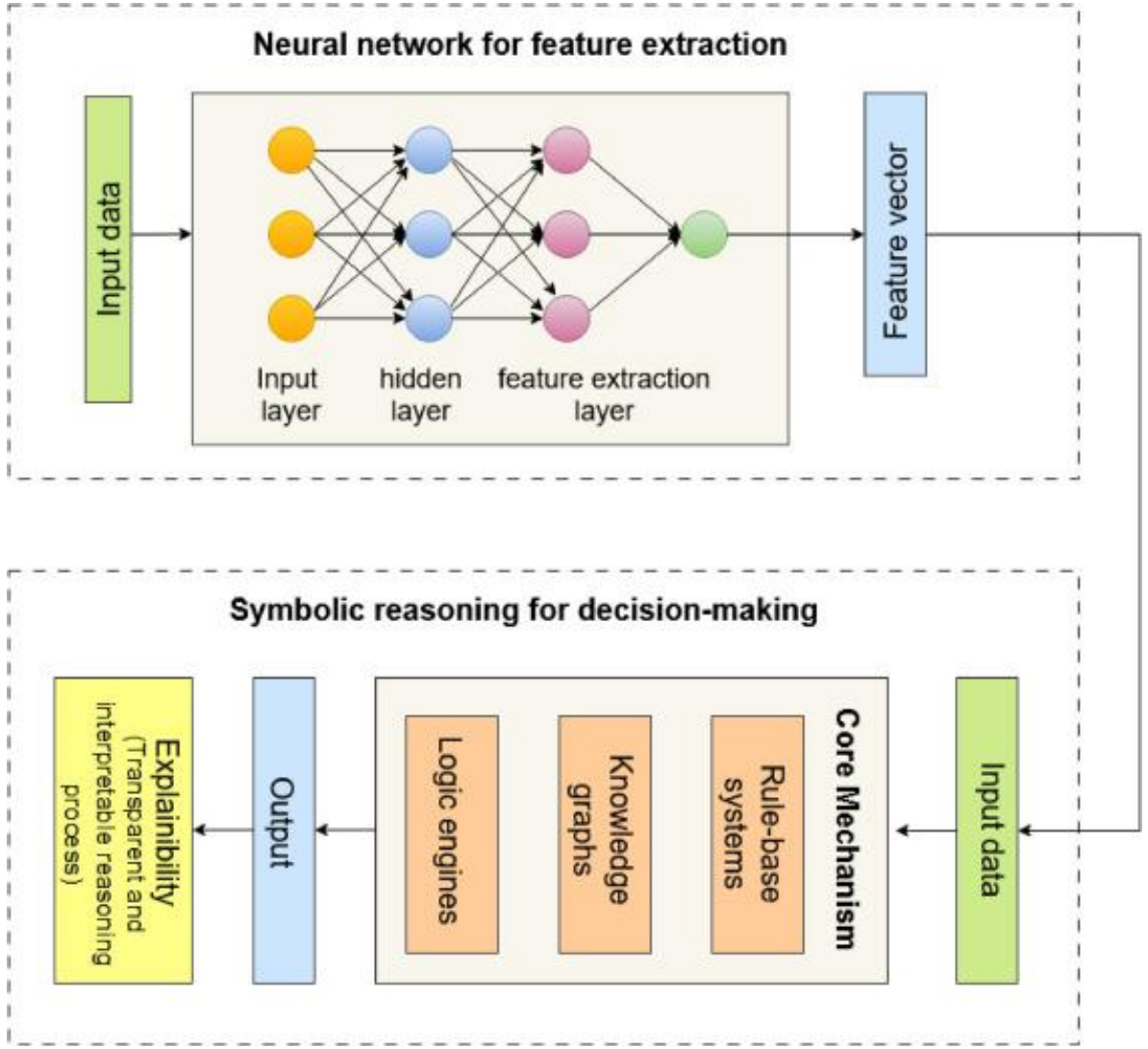
What is AI and why AI is good at math?



- AI is not only machine learning and machine learning is not only neural networks.

$$\text{AI} \supset \text{ML} \supset \text{ANN}$$

What is AI and why AI is good at math?



- Models like Alpha Proof, Alpha Geometry or Alpha Zero use neurosymbolic architectures.
- Part 1 of the architecture is the creative part that generates solutions
- Part 2 of the architecture uses formal logic and a knowledge base to formally verify the solution.
- Both solution and verification are often expressed as programs with steps connected by logic rules.

Now let’s focus on the generative part…

Generative AI – general idea

- Given a known stochastic variable *x* with a complex and unknown distribution we want a transformation to an unknown variable *z* with a known and easy to sample distribution:

$$x = g(z)$$

where *z* is drawn from a known distribution $N(z, \theta)$.

Now if we can find the transformation *g*, we can sample *N* and *generate x*.

The idea of generative AI is to use a neural network to model *g:*

$$g(z) \approx h(z, \omega)$$

where *h* is the network and ω are the weights of the network, and are to be determined.

Generative AI – two possible architectures

There are many ways to build a generator *h*. One is by building an invertible machine learning model so that:

$$x = h(z, \omega)$$

and

$$z = h^{-1}(x, \omega)$$

The other is to create two functions *h* and *f* that model the direct and inverse transformation:

$$x = h(z, \omega)$$

and

$$z = f(x, \alpha)$$

Generative AI – two possible architectures

For clarity we call *z* the **latent variable** and the corresponding space *Z* the latent space.

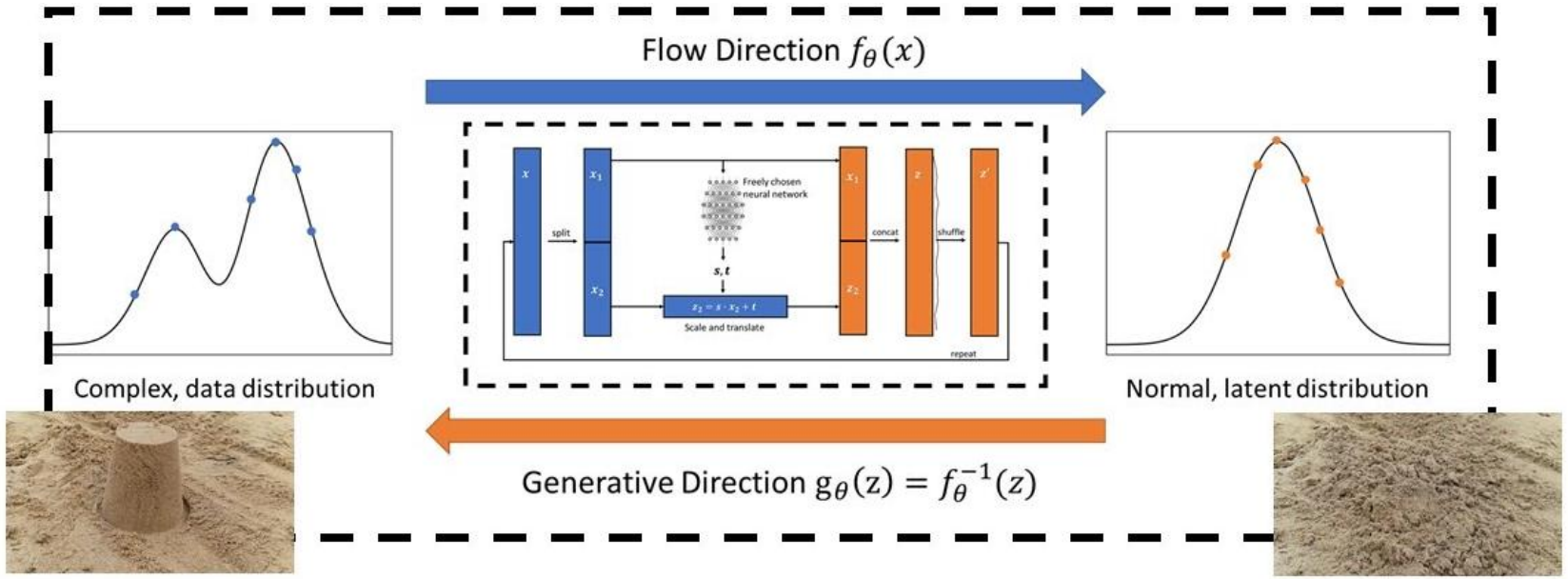
The function *h* is the **machine learning model** and the generative process is divided into a training step and an inference step.

During training we use data set $x \in X$ to find the weights ω of the model

During the inference step we sample z to generate a new x

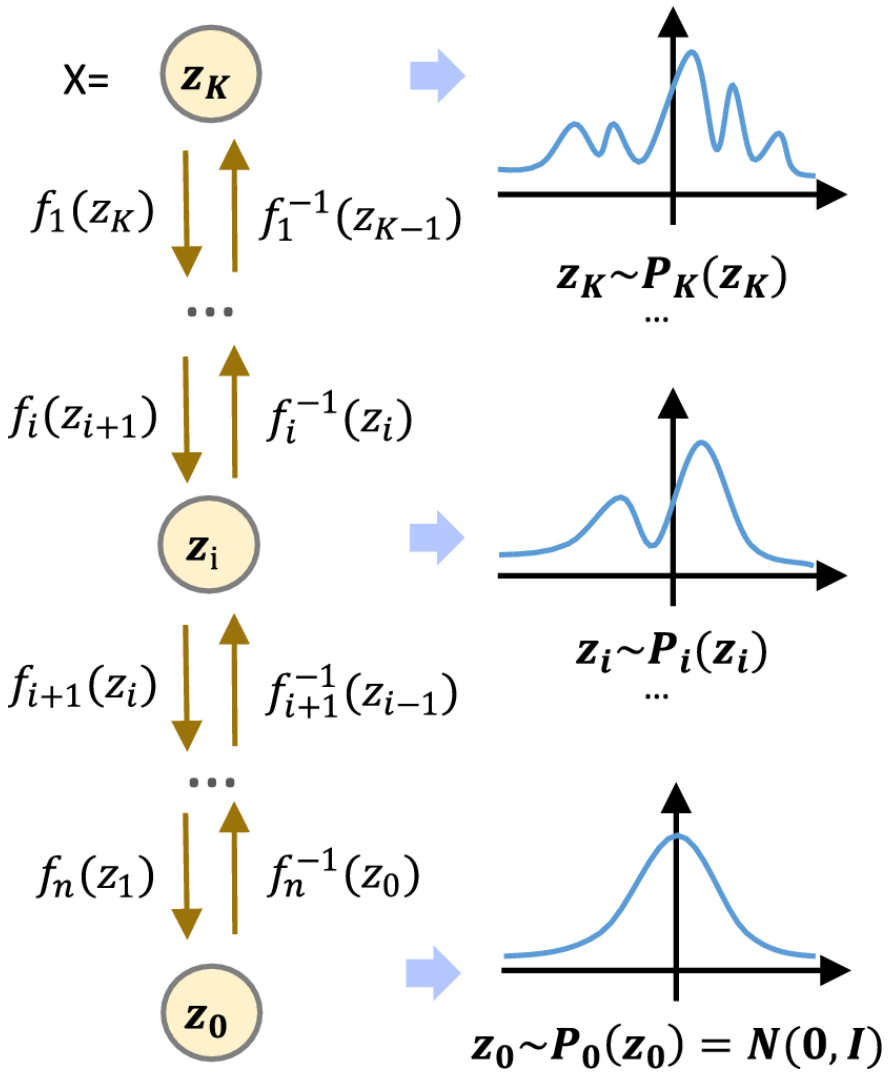
Finding the weights ω means solving an optimization process that minimises a so-called loss function.

Generative AI – Flow-based Model



- The goal is to map a complex distribution into a simple one – flow direction
- We can then sample the simple distribution to generate samples in the complex one – generative direction
- We model the function relating the physical space **x** with the latent space **z** with an invertible machine learning model

Generative AI – Flow-based Model

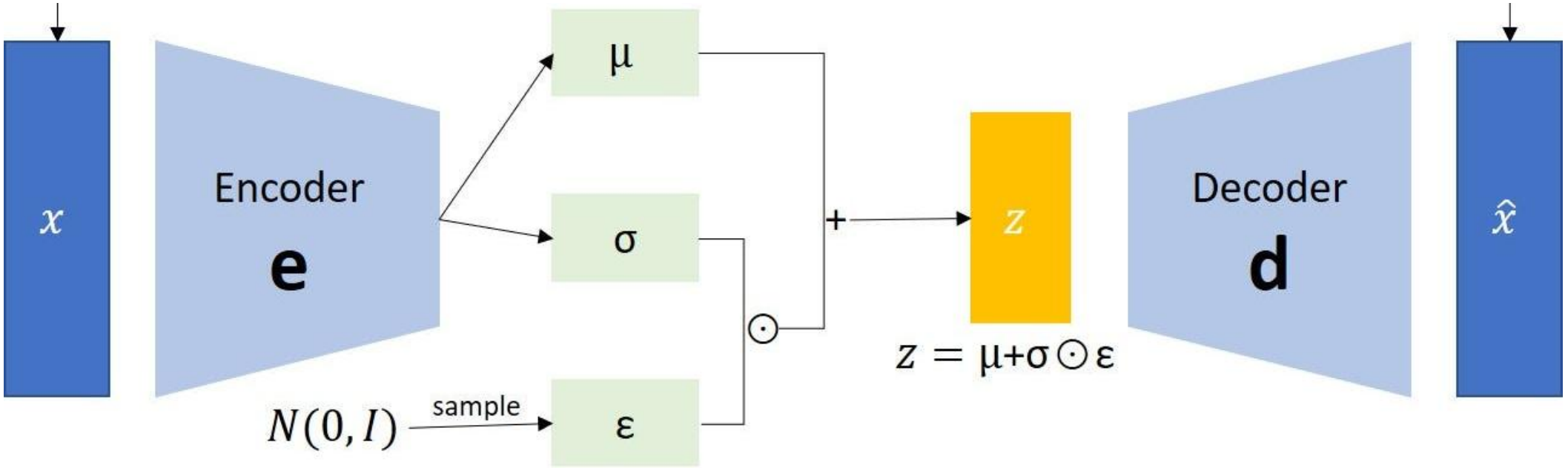


- A normalizing flow model can be seen as a sequence of function compositions in which each function (unknown in general) is approximated with a neural network.
- A key feature is that these transformations are invertible.
- Wecna push the idea to the continuous domain and have a neuralODE or continuous flow:

$$x = F(z_0) = z_T = z_0 + \int_0^T f(z_t, t)dt$$

$$z_0 = F^{-1}(x) = z_T + \int_T^0 f(z_t, t)dt = z_T - \int_0^T f(z_t, t)dt$$

Generative AI – Variational Autoencoder

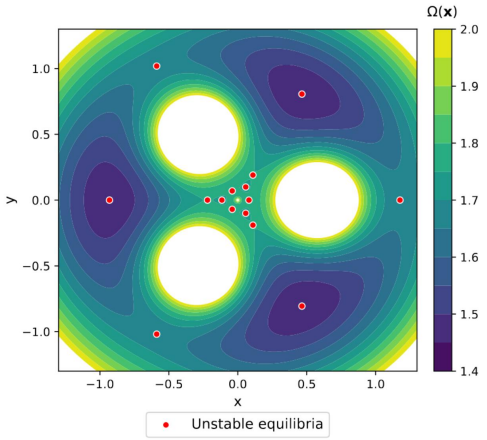


What we want is $\min(\hat{x} - x)$ and we want that the distribution of *z* is as close as possible to a known distribution that we can easily sample.

We then find the weights of the encoder and of the decoder so that *x* is reconstructed by the decoder and the distribution of *z*, generated by the encoder, is as close as possible to the one we want.

So, what is Generative Astrodynamics?

Main Idea



Suppose that a dynamical system has *N* equilibrium solutions and we know only *M*<*N* can we directly generate the remaining *N*-*M* equilibria?

Can we generate equilibria conditional to their type of stability under parametric uncertainty?

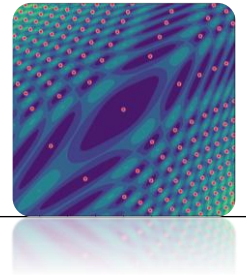
Suppose that a dynamical system has *N* periodic solutions but we know only *M*<*N*. Can we directly generate the remaining *N*-*M* periodic solutions?



EPSRC

AEROSPACE CENTRE of EXCELLENCE

- Research question: can we automatically generate the local minima of a scalar function under uncertainty?

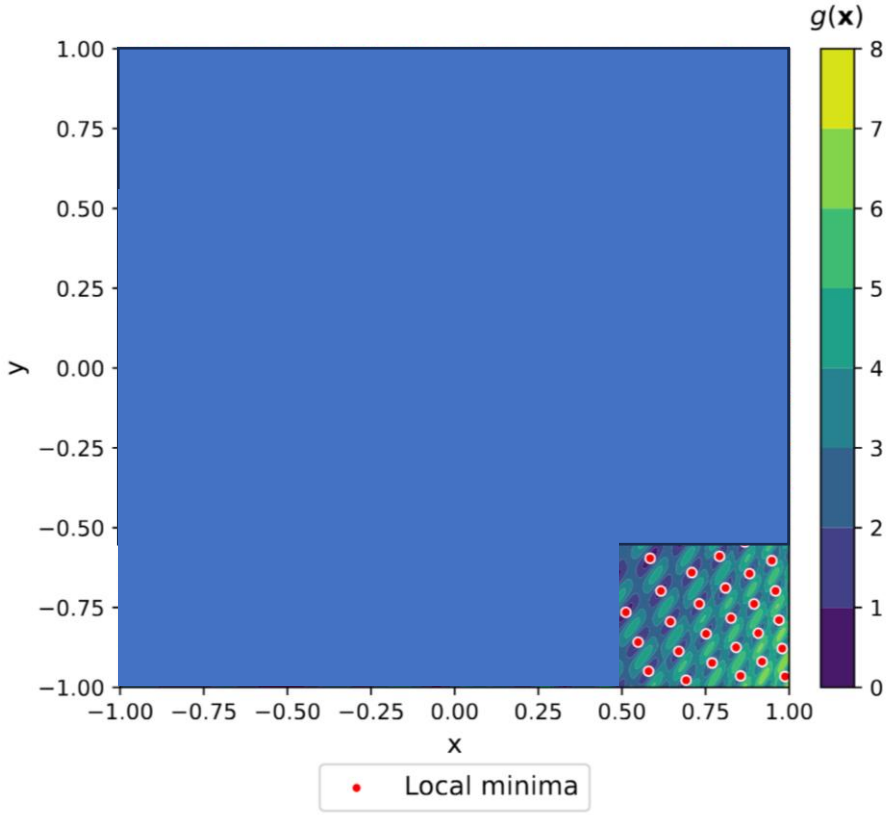


Generation of local minima under uncertainty



Callum Wilson

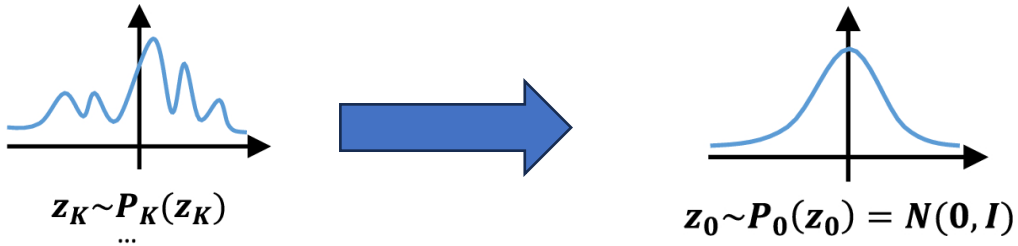
Generating Minima of a Scalar Function



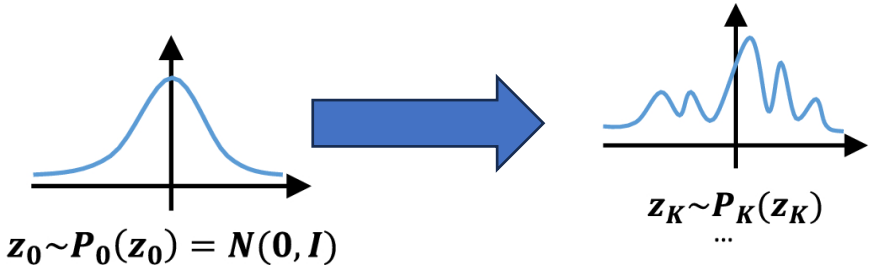
- Say that we have a scalar function $g(\mathbf{x})$ with N minima but we know only M

Idea

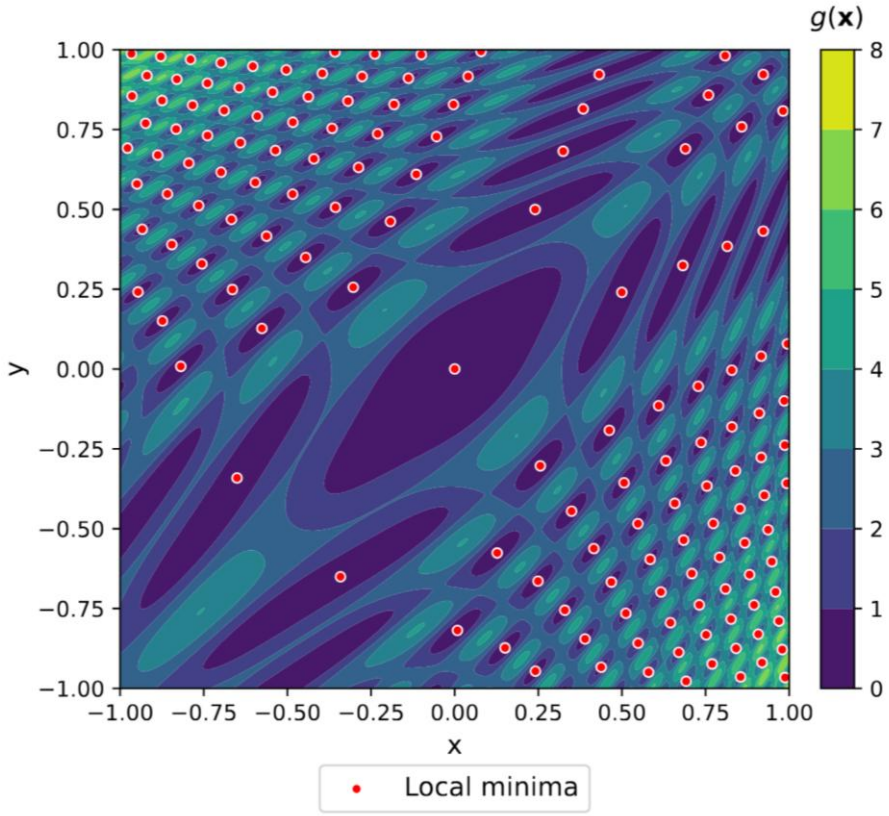
- Train a flow model on the distribution of the known minima



- Sample the latent space and generate new minima



Generating Minima of a Scalar Function



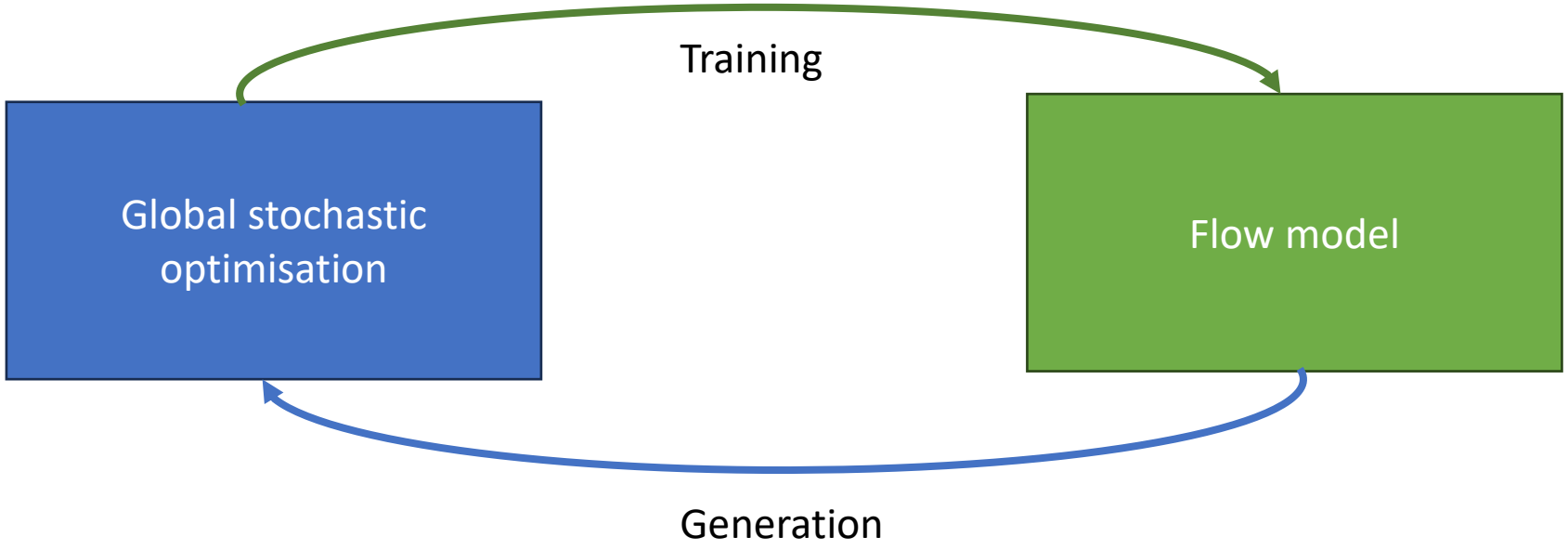
Problem

- **Out of distribution generation** is hard because the observed distribution is only part of the whole distribution

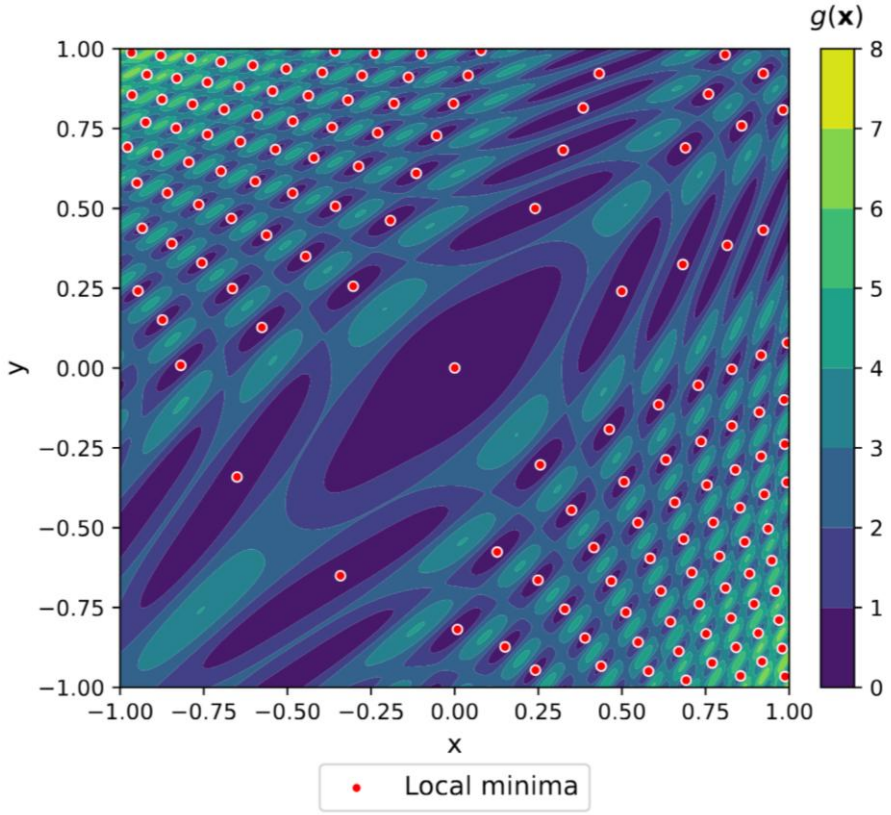
Question: how do we generate outside the distribution of the training set?

Flow Expander via Global Search

1. Train the flow model on the subset of known minima
2. Sample the latent space to generate new minima
3. Add the new sampled minima to the training set
4. Expand the flow model



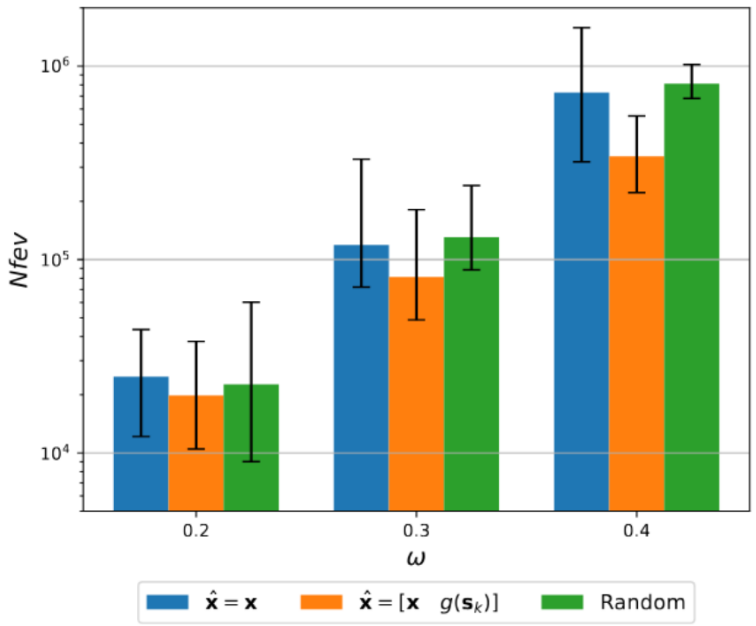
Generating local minima



We can train the model in two ways:

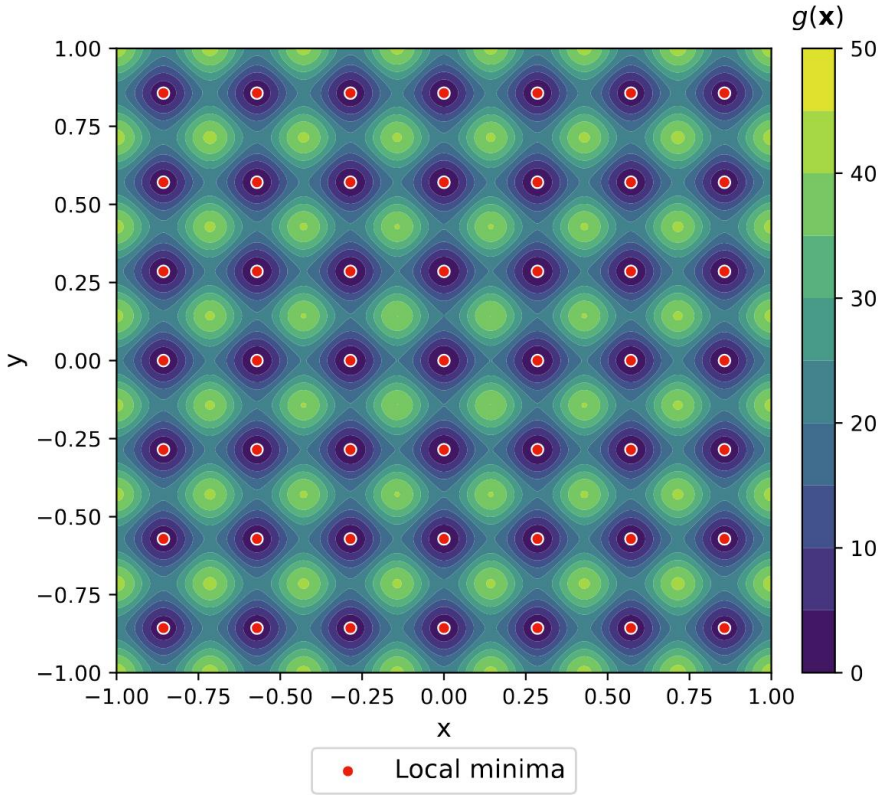
- Only on the distribution of the location of the minima x
- On the location of the minima plus their cost function g(x)

The question is: if we want ALL the minima can we do better than a systematic uniform sampling with local search?



The uncertain parameter omega defines how many local minima exist in the search space, from 29 to 462.

Generating local minima under uncertainty



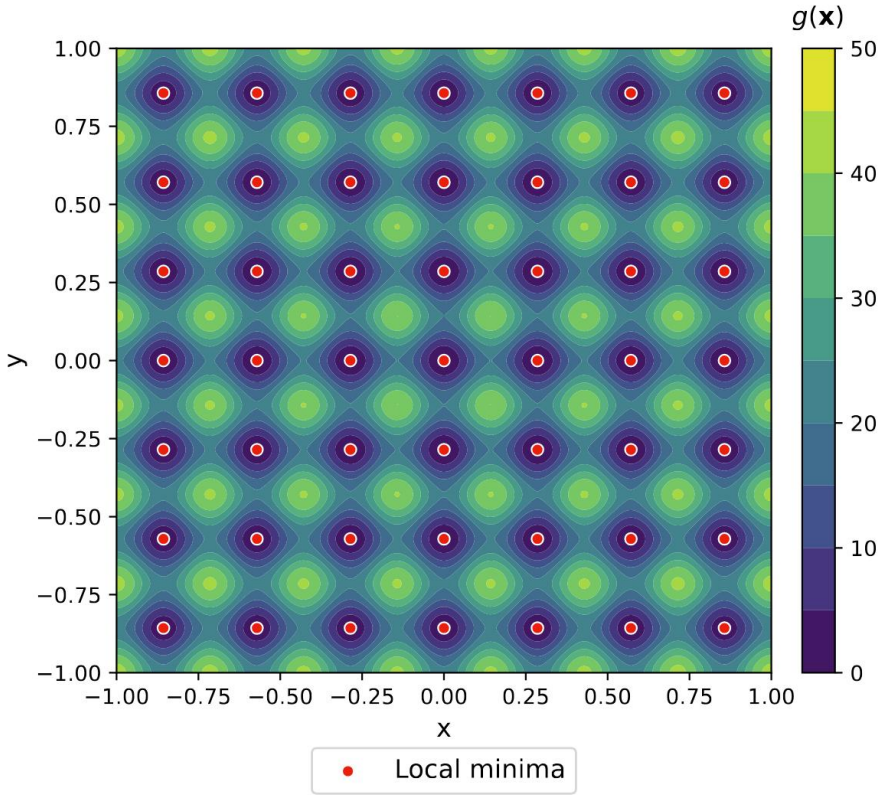
Question.

Can we train the flow model including one or more uncertain parameters that change the number of minima and their nature?

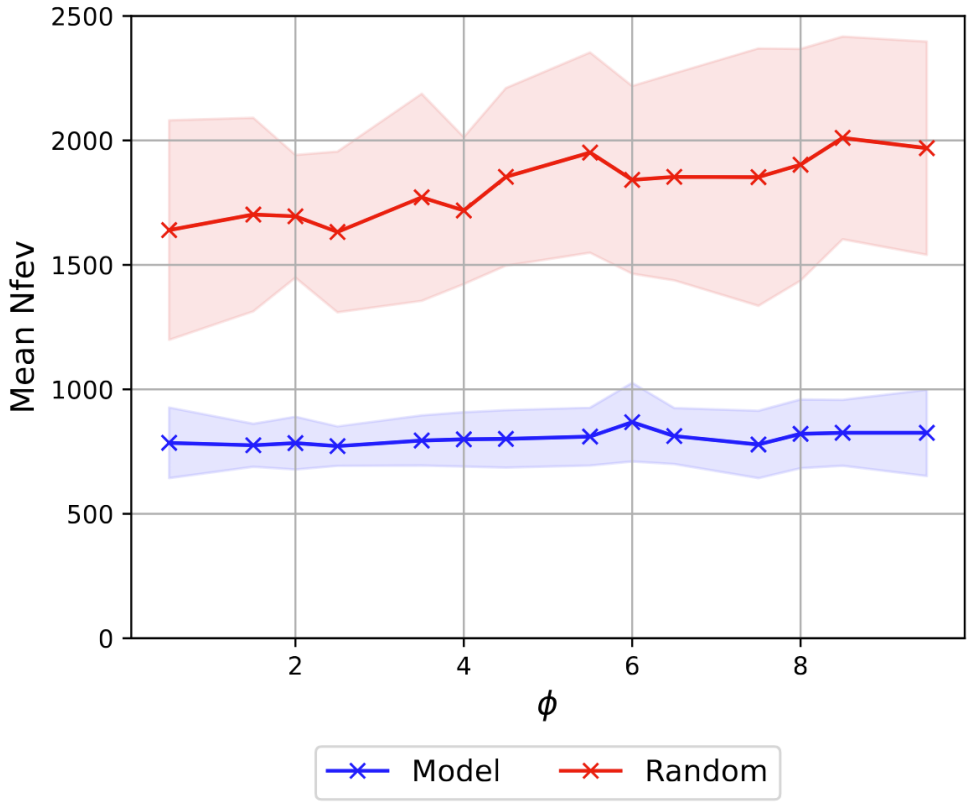
Example

Easy distribution but on top of a variable number of minima, up to 225, the landscape rotates according to the uncertain parameter ϕ

Generating local minima under uncertainty

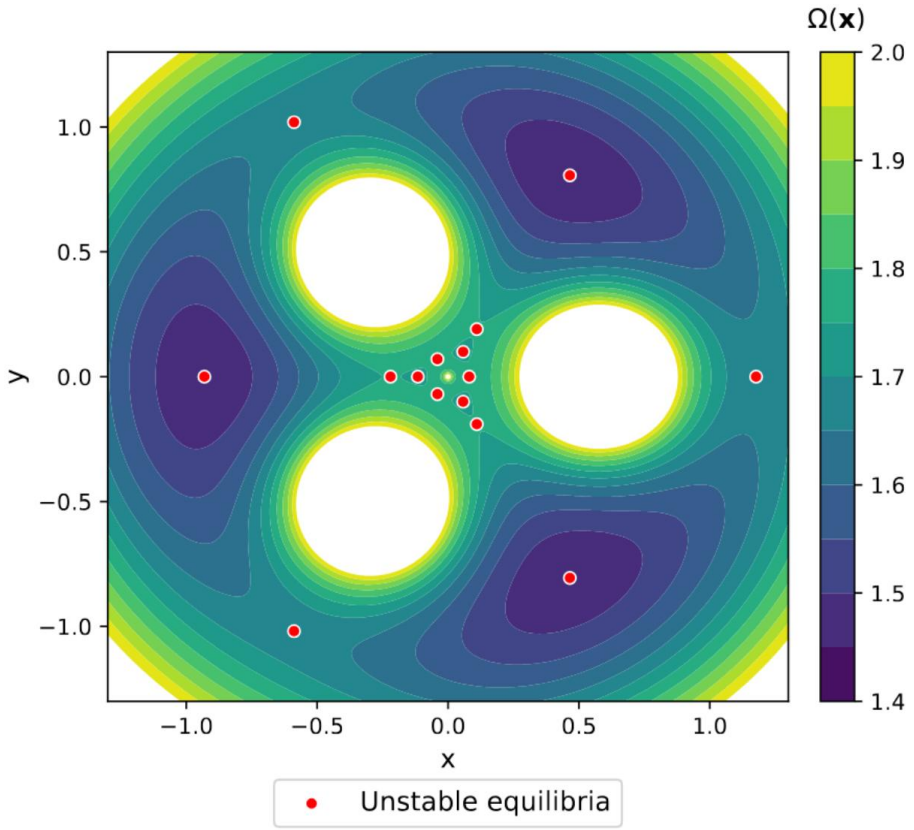


Model trained on minima, cost function and uncertain parameter rotating the search space.



Easy distribution but on top of a variable number of minima, up to 225, the landscape rotates according to the uncertain parameter ϕ

Generating local minima under uncertainty



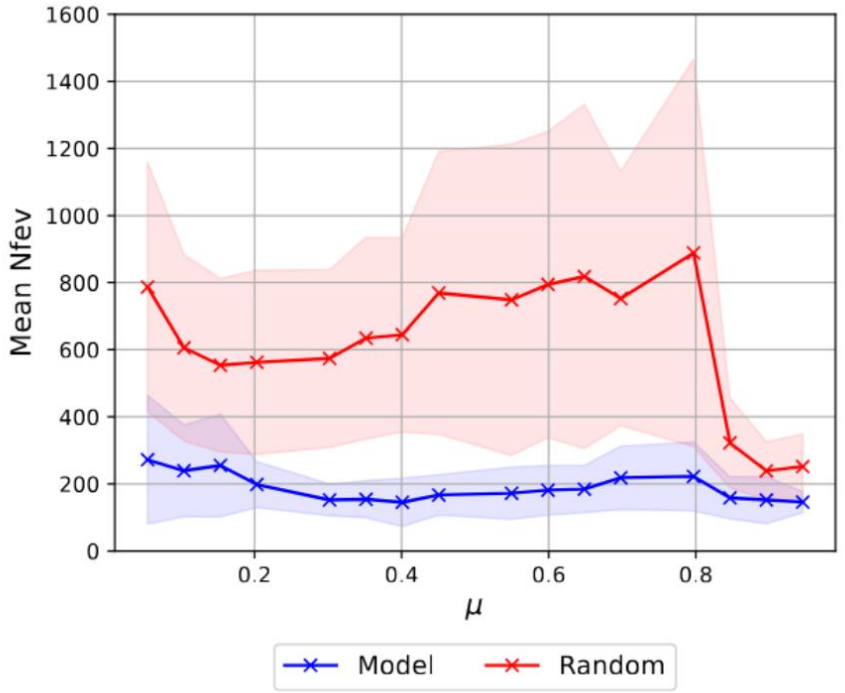
Mass $m_0=\beta m$ all other masses are equal to m.
Effective potential:

$$\Omega(x,y) = \frac{1}{2}\left(x^2+y^2\right) + k\sum_{i=0}^{3}\left[\frac{m_i}{r_i}\right],$$

$$k = \frac{1}{3(1+\beta\sqrt{3})},$$

$$r_i = \sqrt{(x-x_i)^2+(y-y_i)^2},\ i=0,1,2,3.$$

$$\mu = 1/(1+\beta)$$

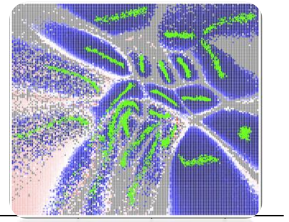


In the case of the circular restricted 5 body problem the number minima and their stability is conditional to μ

References

- Wilson, C., **Vasile, M**. *Generating Stable and Metastable Critical Points in Uncertain Systems via Flow-based Models*, Expert Systems, 2025, 10.13140/RG.2.2.16413.83686
- Wilson, C., & **Vasile, M.** (2024). *Generation and classification of critical points in uncertain N-body problems via machine learning*. Paper presented at 75th International Astronautical Congress , Milan, Italy.

- Research question: can we automatically generate periodic orbits from the time series of known ones?



Generation of Periodic Solutions with a VAE



Walther Litteri



Alvaro Francisco Gil



Dr Victor Rodriguez Fernandes

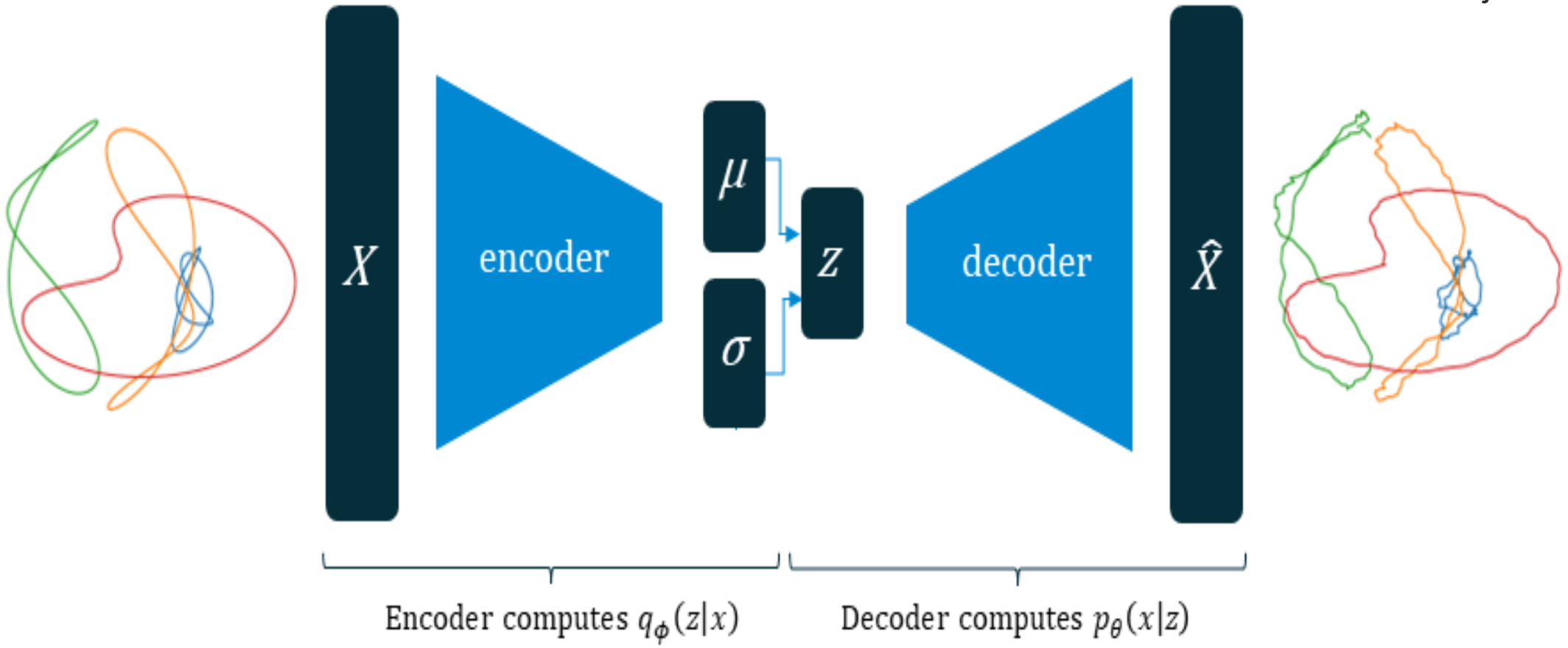
Lagrange Generator

Lower dimensional latent space

Trajectory state vectors

Feature extraction

Generated Trajectories



29

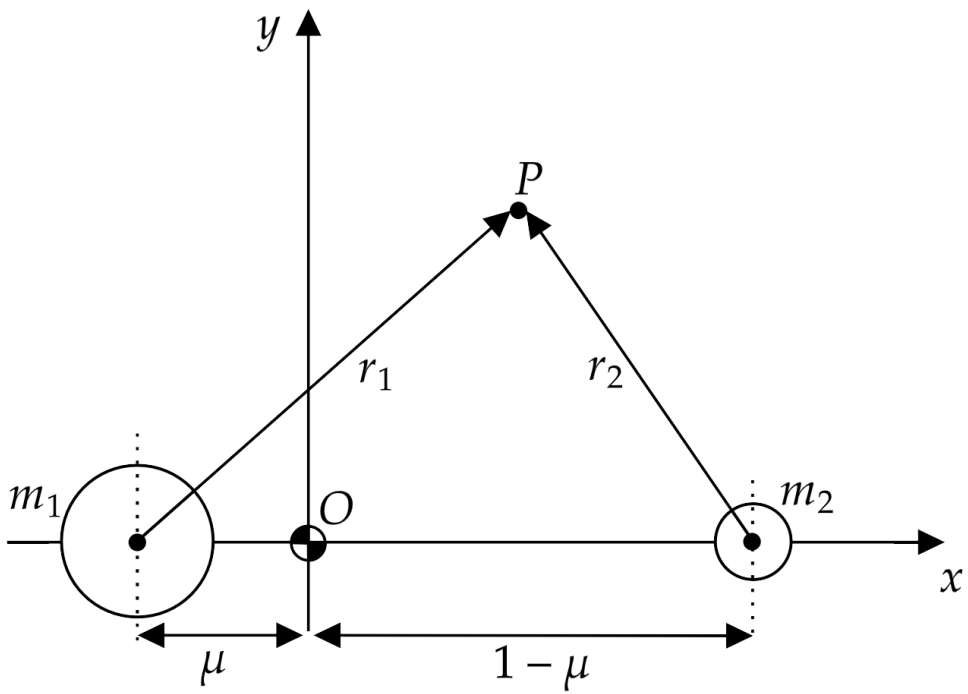
The Restricted Three-Body Problem (CR3BP)
Geometry and model

A **massless particle** P moves around **two primaries**

The primaries move in **circular motion** around their Centre of Mass O

Equations of motion are **normalized** to length, time and mass units

$$\mu = \frac{m_2}{m_1+m_2} = 0.0122 \text{ (Earth-Moon)}$$



Geometry of the Circular-Restricted Three Body Problem.

30

Dataset

Input Dataset

40 families of Periodic Orbits in the Moon CR3BP[1]

44k Initial Conditions (ICs) and orbit features

Trajectory propagation

ICs are integrated for 1 Orbital period over N=100 nodes

1e-13 Absolute and Relative tolerance on the solver[2]

Data Preprocessing

Normalisation with respect to dataset-wide **max and min** features. Resulting **data range from 0 to 1**.

[1]NASA-JPL, *Three-Body Periodic Orbits*. URL: https://ssd.jpl.nasa.gov/tools/periodic_orbits.html. Visited on 14/01/2025

[2]Rackauckas et al., *DifferentialEquations.jl - a performant and feature-rich ecosystem for solving differential equations in Julia,* Journal of Open Research Software, 2017

Training

The training dataset is **divided into batches** consisting of M=32 orbits

Validation Dataset: 5% of the volume

Training is performed for **25 epochs**

Reconstruction Loss term ($\mathcal{L}_r$, MSE)

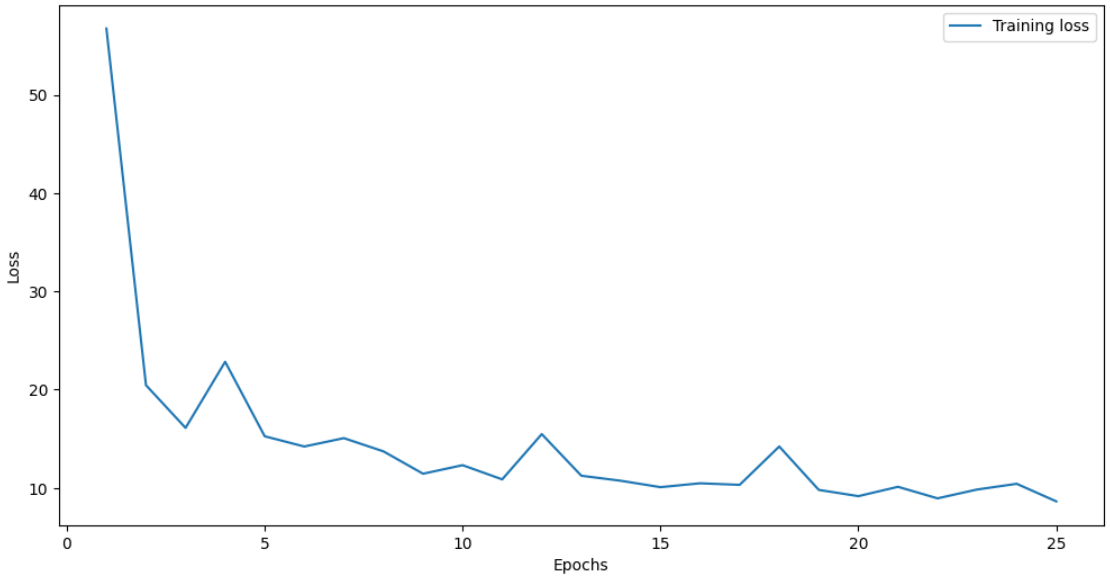
$$\mathcal{L}_r = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{7}\left(x_{ijk} - \hat{x}_{ijk}\right)^2$$

KL-Divergence term ($\mathcal{L}_{KL}$)

$$\mathcal{L}_{KL} = -\frac{1}{2}\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{2}(1 + \tilde{\sigma}_{ij} - \mu_{ij}^2 - e^{\tilde{\sigma}_{ij}})$$

With $M = 32$, $N = 100, \sigma = e^{1/2\tilde{\sigma}}$

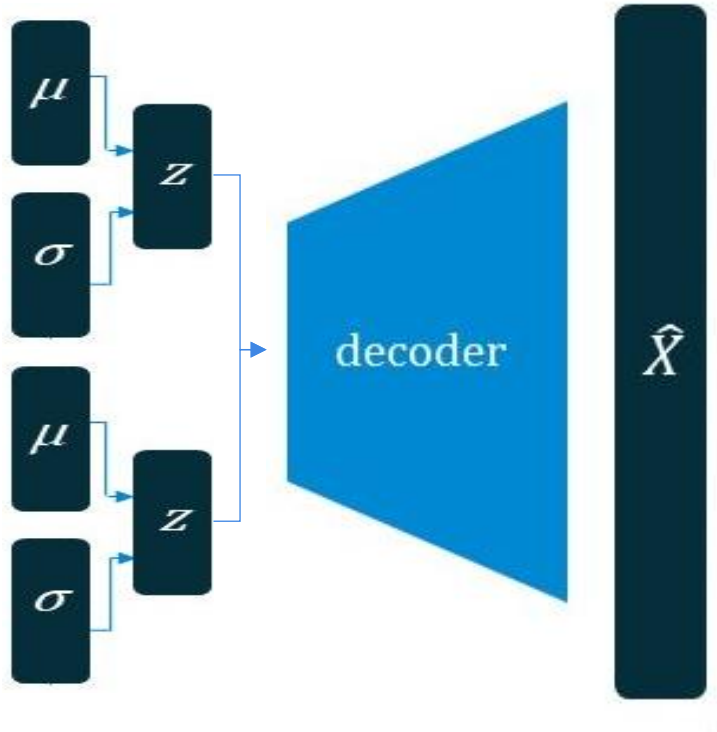
$\beta = 1e - 3$



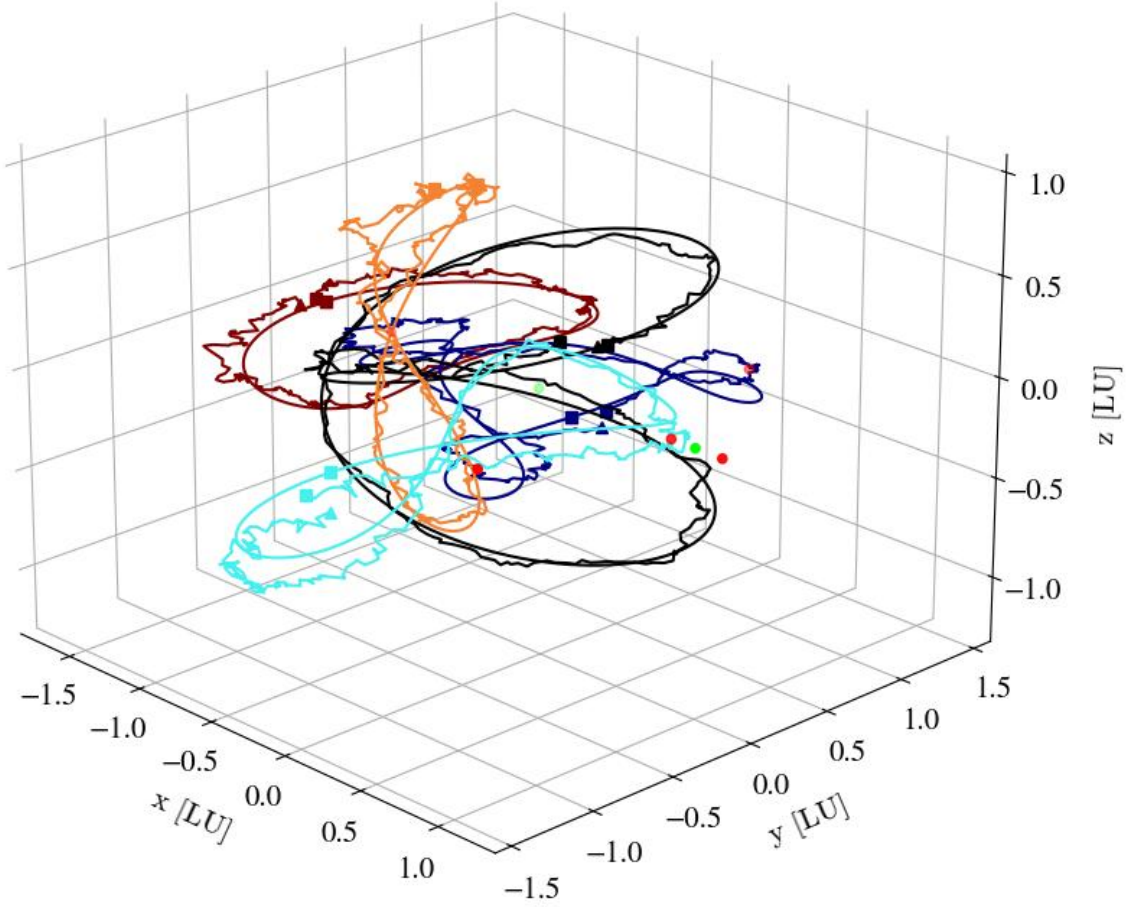
Loss function evolution during training.

Lagrange Generator
Synthetic Orbits

Synthetically generated orbits are locally refined with a shooting method

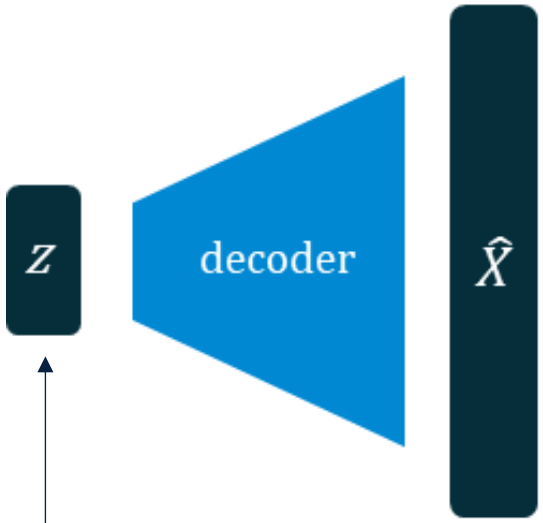


Sampling process and decoding in a VAE architecture.

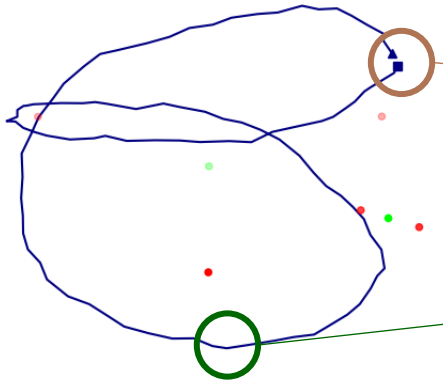


33

Lagrange Generator
Orbit Generation and Refinement



Latent Space Sampling



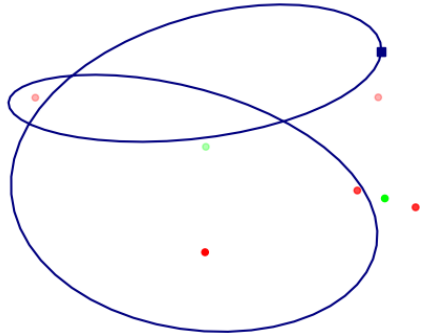
Generated Orbit

Refinement Algorithm
(Differential Correction)

```
enforce periodicity

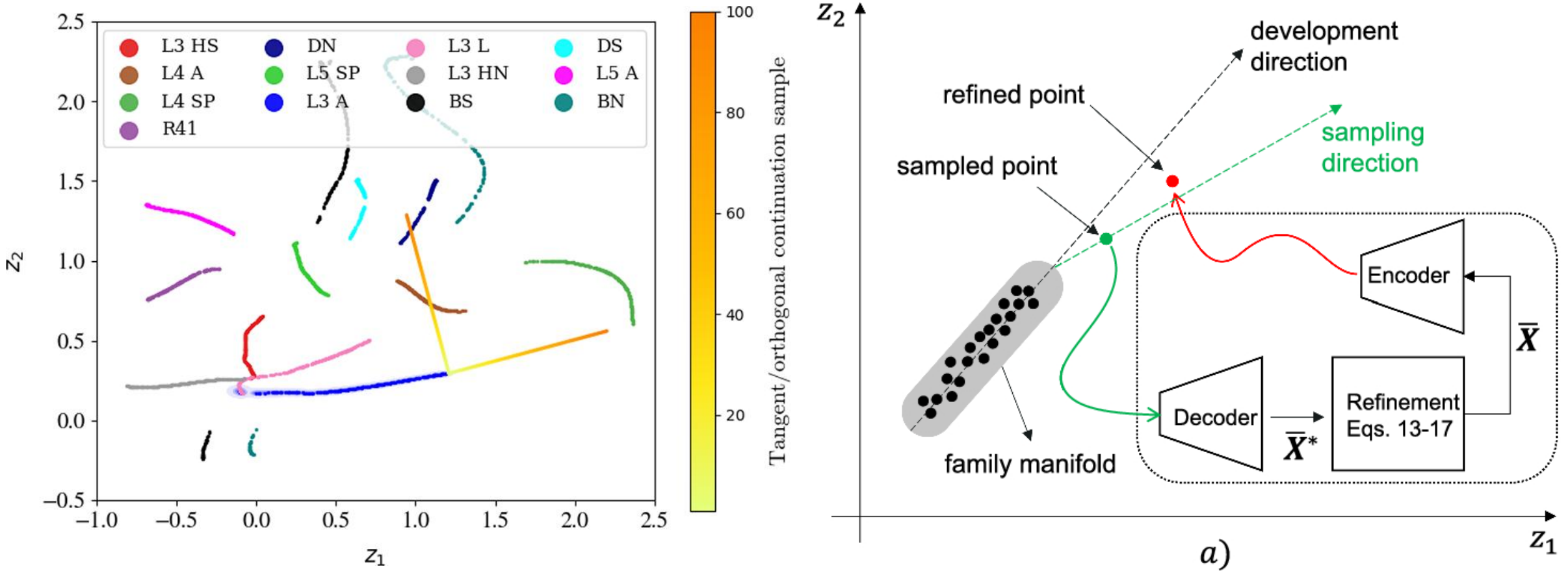
for each trajectory node:
      enforce state continuity

end
```

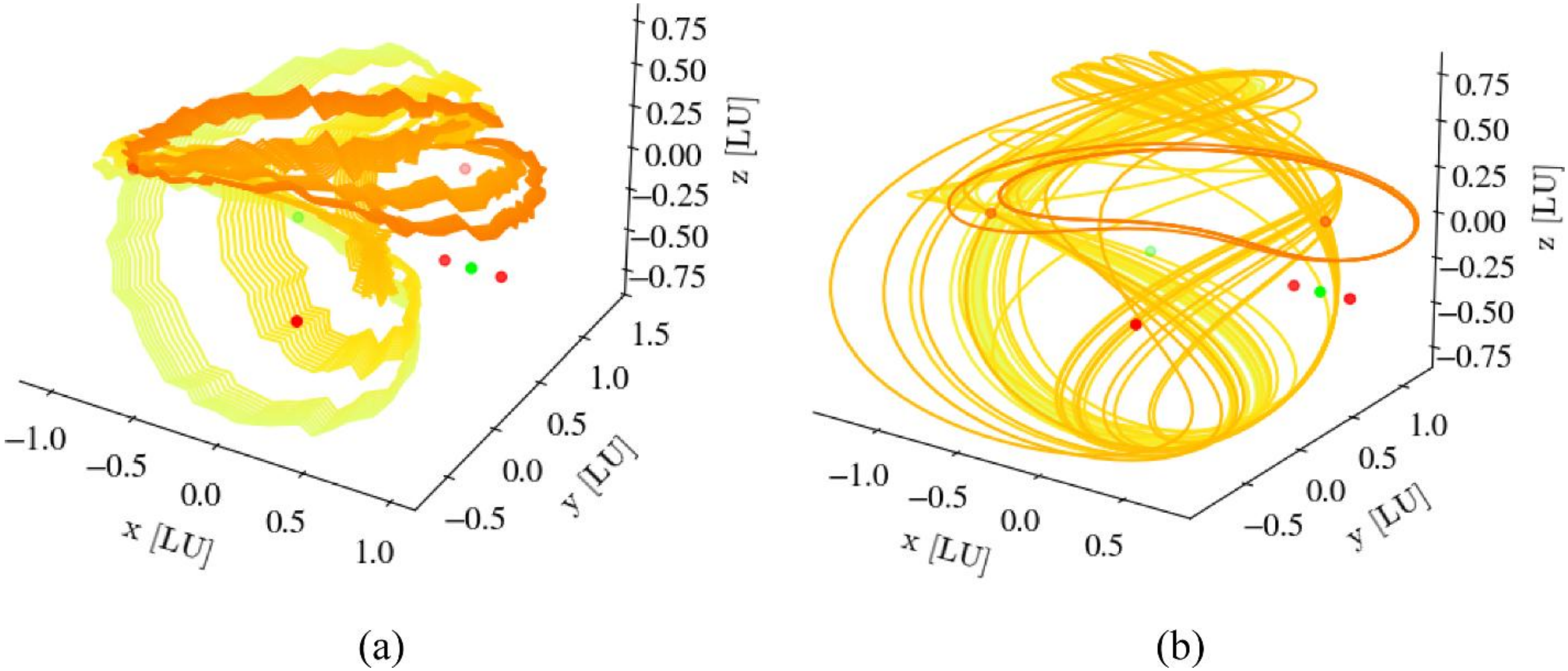


Refined Orbit

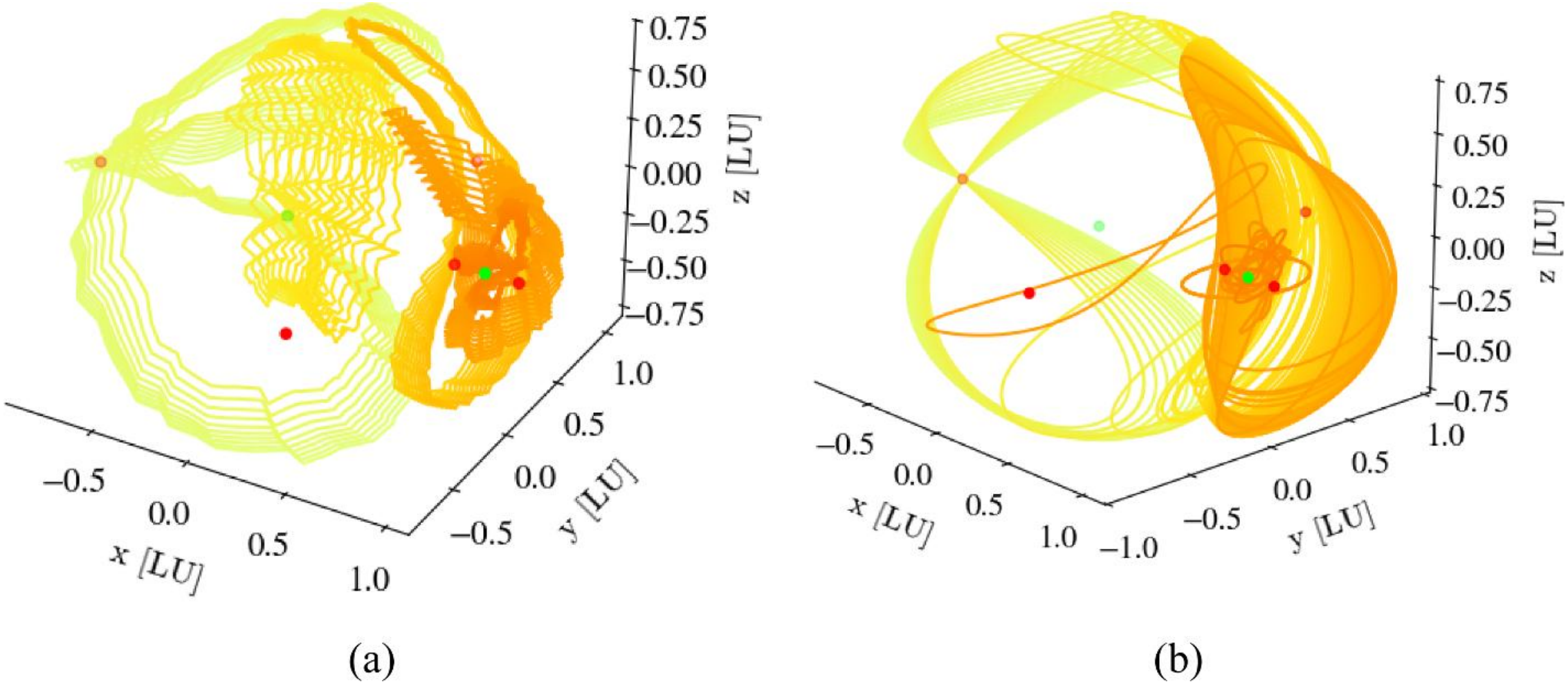
Latent Space and Generation by Continuation



Latent Space and Generation by Continuation

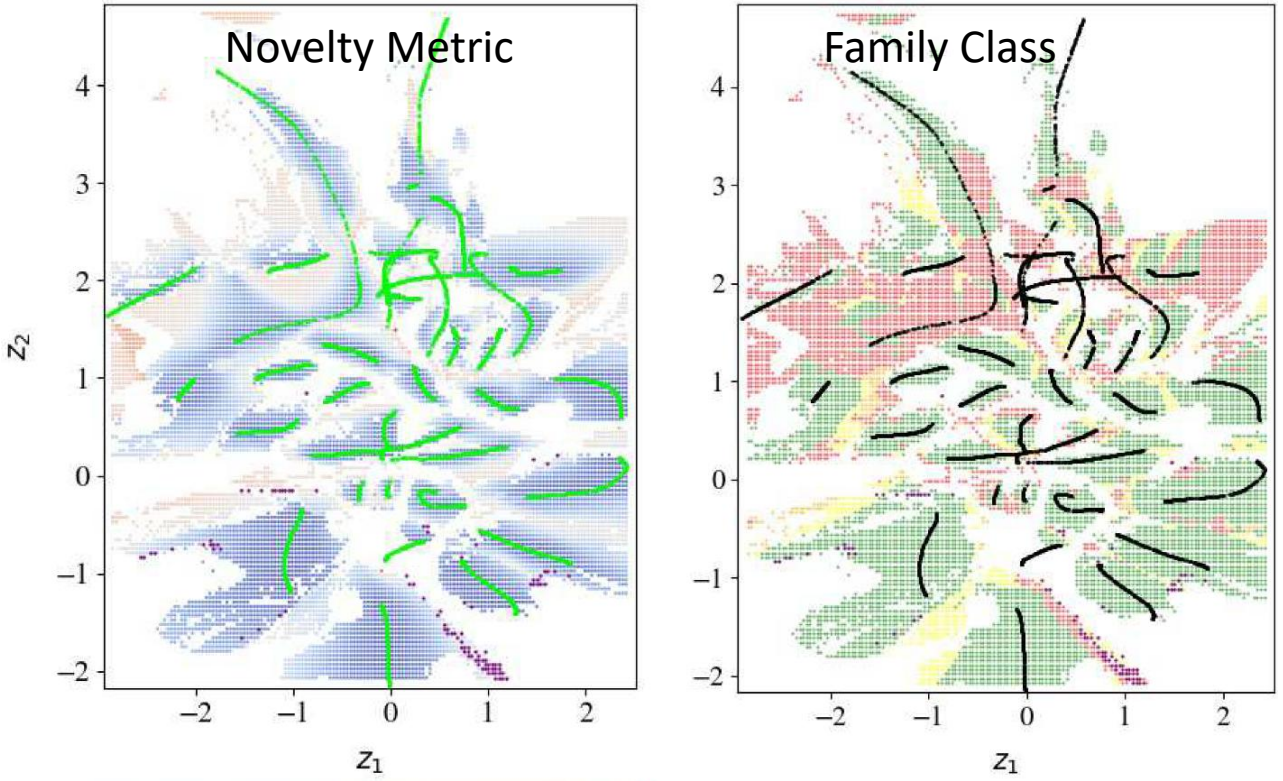


Latent Space and Generation by Continuation



Grid Sampling the Latent Space

- Systematic exploration of the latent space
- Some generated orbits converge to physical orbits with geometric characteristics that define the nearest family.
- Others converge far from the nearest families



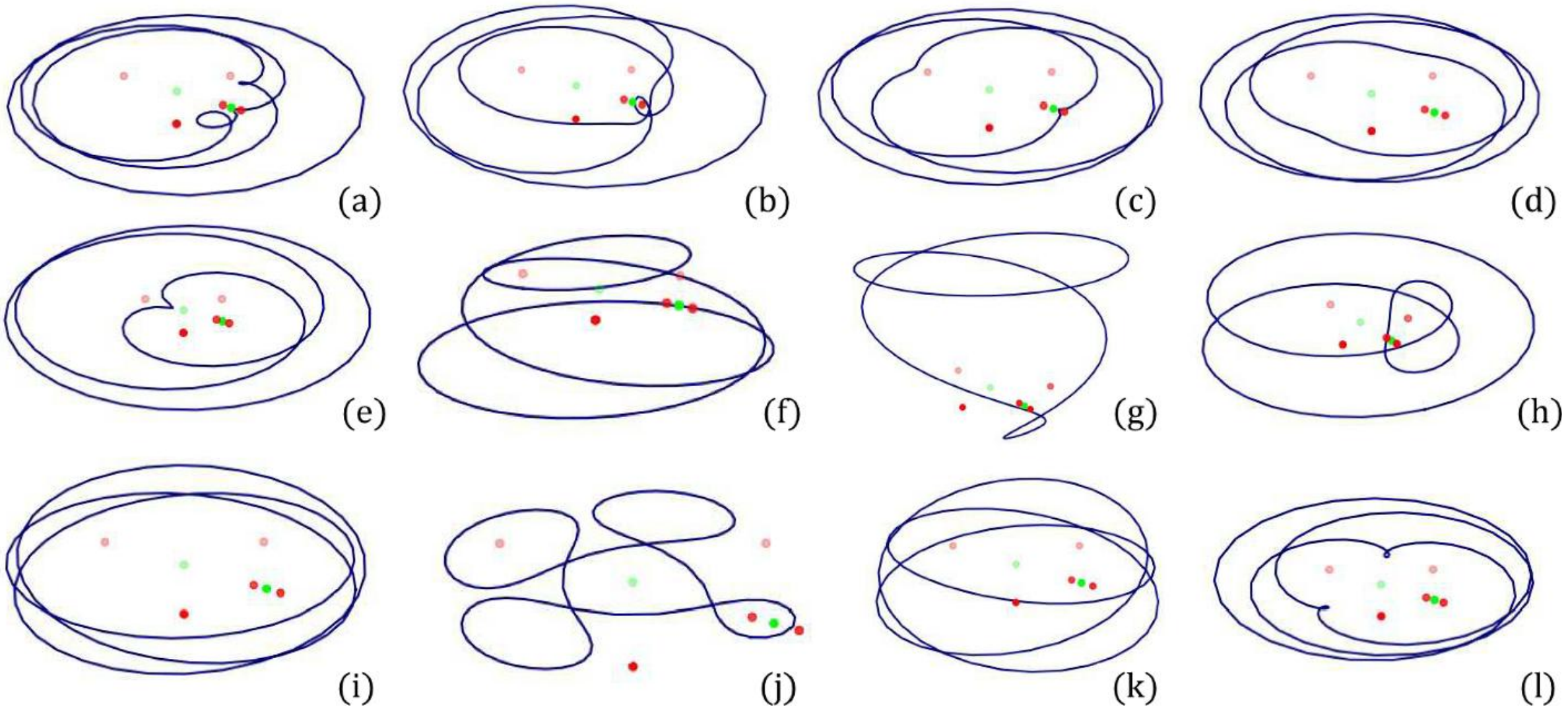
$$D_{\text{nov}} = \delta_1 D_1 + \delta_2 D_2 + \delta_3 D_3,$$

Mahalanobis distance in latent space

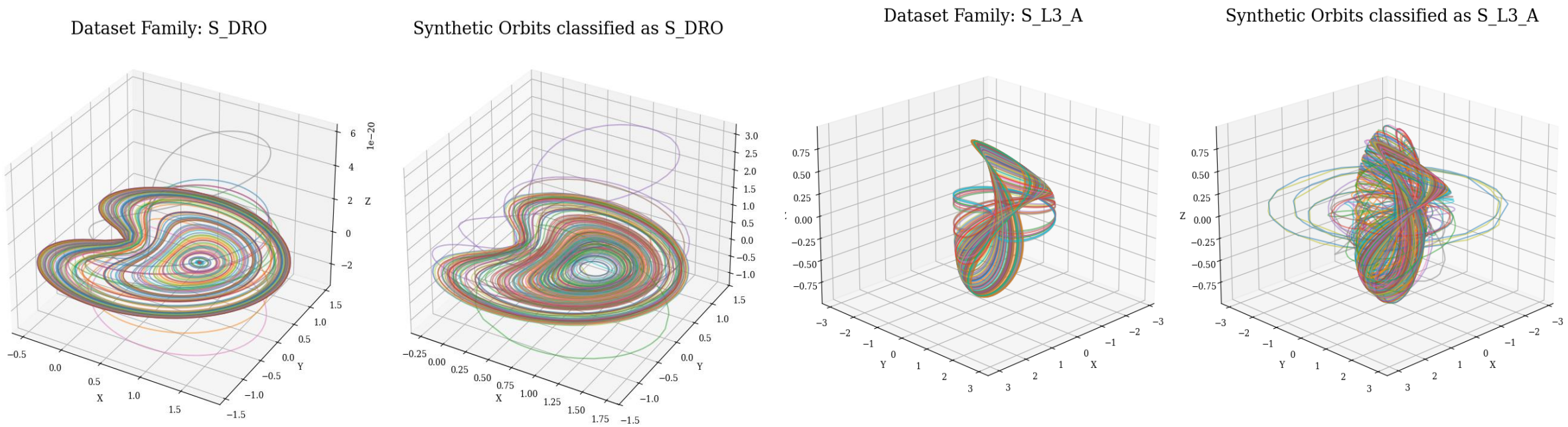
Mahalanobis distance in physical space

Misclassification probability

Grid Sampling the Latent Space



Generative Re-Discovery

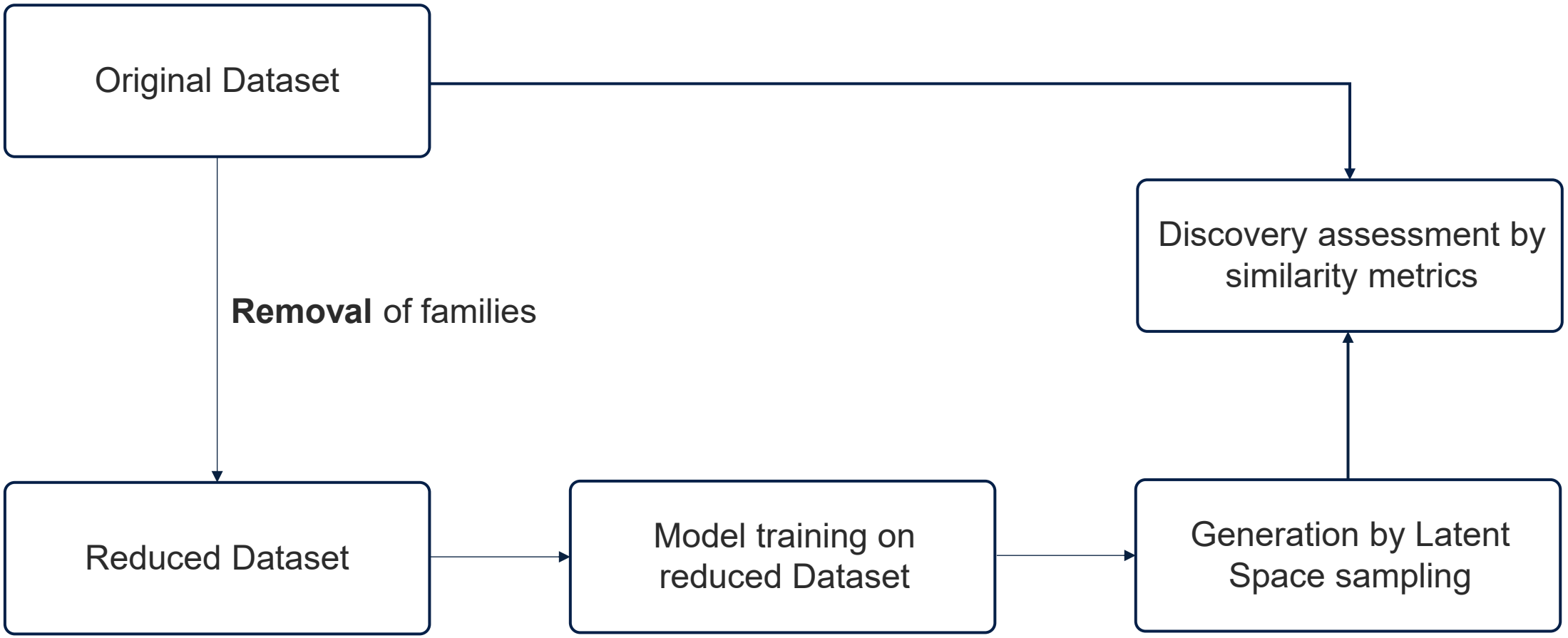


Examples of **generations** of periodic solutions **from a reduced dataset,** and **their classification** to one of the original families.

Results for **Distant Retrograde Orbits** (left couple) and **L3-Axial orbits** (right couple).

Synthetic orbits (right) and original sample (left).

Generative Re-Discovery



References

1. Litteri, W., Francisco Gil, A., Vasile, M. *et al.* Generation of periodic orbits in the restricted three-body problem with a variational autoencoder. *Celest Mech Dyn Astron* **138**, 25 (2026). https://doi.org/10.1007/s10569-026-10299-x

2. Walther Litteri, Alvaro Francisco Gil, Massimiliano Vasile, Victor Rodriguez-Fernandez, and David Camacho. *GENERATIVE ASTRODYNAMICS: TRAJECTORY ANALYSIS AND DESIGN IN THE RESTRICTED THREE-BODY PROBLEM*. 2025 AAS/AIAA Space Flight Mechanics Meeting, Kaua'i, Hawaii, January 19-23 2025At: Kaua'i, Hawaii, USA. https://www.researchgate.net/publication/387938166_GENERATIVE_ASTRODYNAMICS_TRAJECTORY_ANALYSIS_AND_DESIGN_IN_THE_RESTRICTED_THREE-BODY_PROBLEM

3. Alvaro Francisco Gil, Walther Litteri, Victor Rodriguez-Fernandez, David Camacho, Massimiliano Vasile, *Generative Design of Periodic Orbits in the Restricted Three-Body Problem.* SPAICE Conference 2024. https://arxiv.org/abs/2408.03691

4. ESA OrbitGPT project - https://activities.esa.int/4000143521

5. Litteri, W., Francisco Gil, A., **Vasile, M.** et al. *Generation of periodic orbits in the restricted three-body problem with a variational autoencoder*. Celest Mech Dyn Astron 138, 25 (2026). https://doi.org/10.1007/s10569-026-10299-x

- Research question: can we generate a first guess orbital solutions from (conditional to) a number of sparse observations?



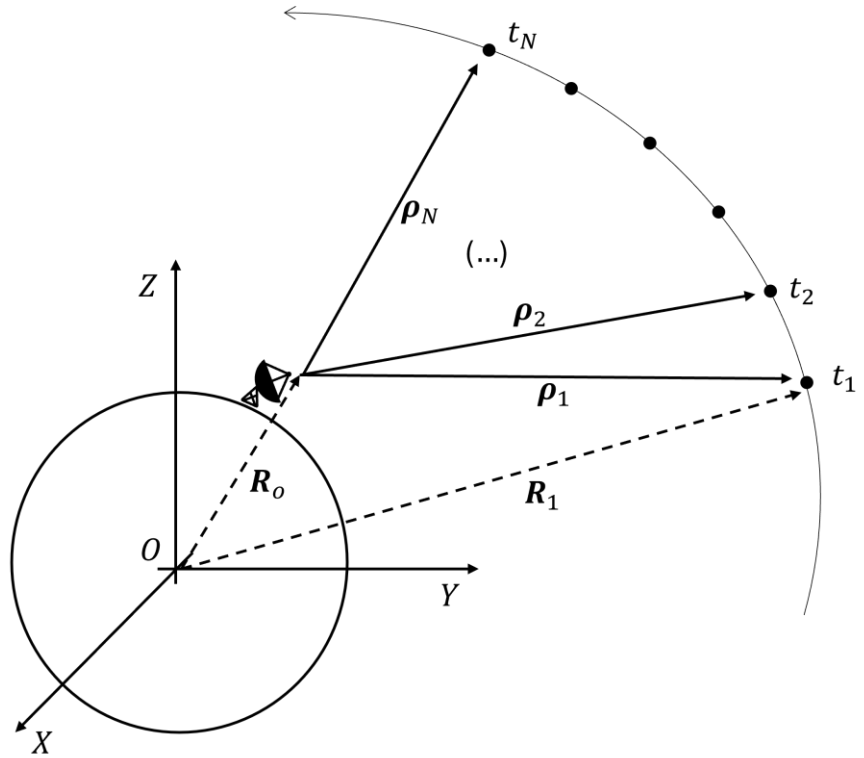
Generative Orbit Determination



Walther Litteri

Orbit Determination
An introduction

The problem: given a set of **observations**, derive the **state** of the spacecraft.



Orbit Determination framework from a topocentric position $\boldsymbol{R}_O$.

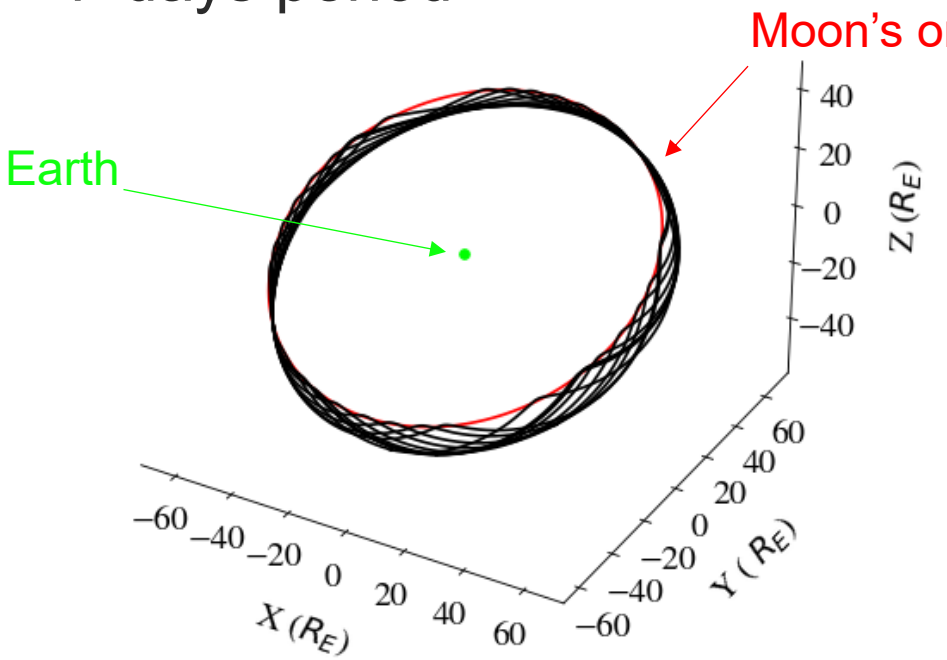


DSA-2 in Cebreros, Spain
European Space Operations Centre.
Photo: ©ESA

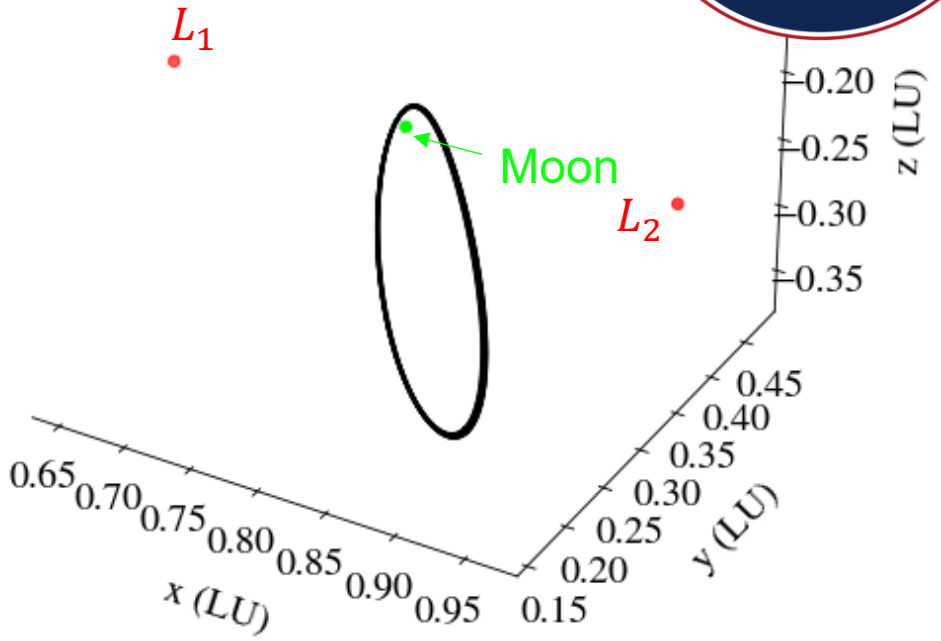
Orbit Determination
Target Orbit

The orbit:

- Near Rectilinear Halo Orbit (NRHO) at Earth-Moon L_2 point
- 7-days period



Geocentric Inertial Reference Frame



Synodic Reference Frame

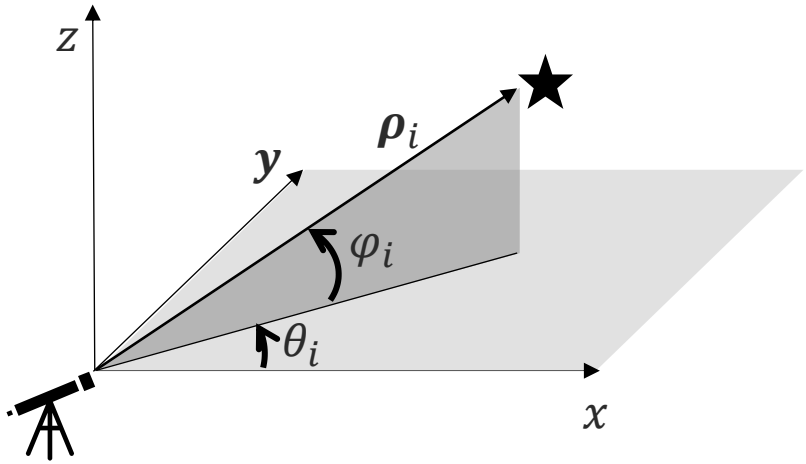
Selection of 10 orbits belonging to the L_2 −NRHO family, propagated for 1 lunar sidereal period.

Orbit Determination
Observation Arcs

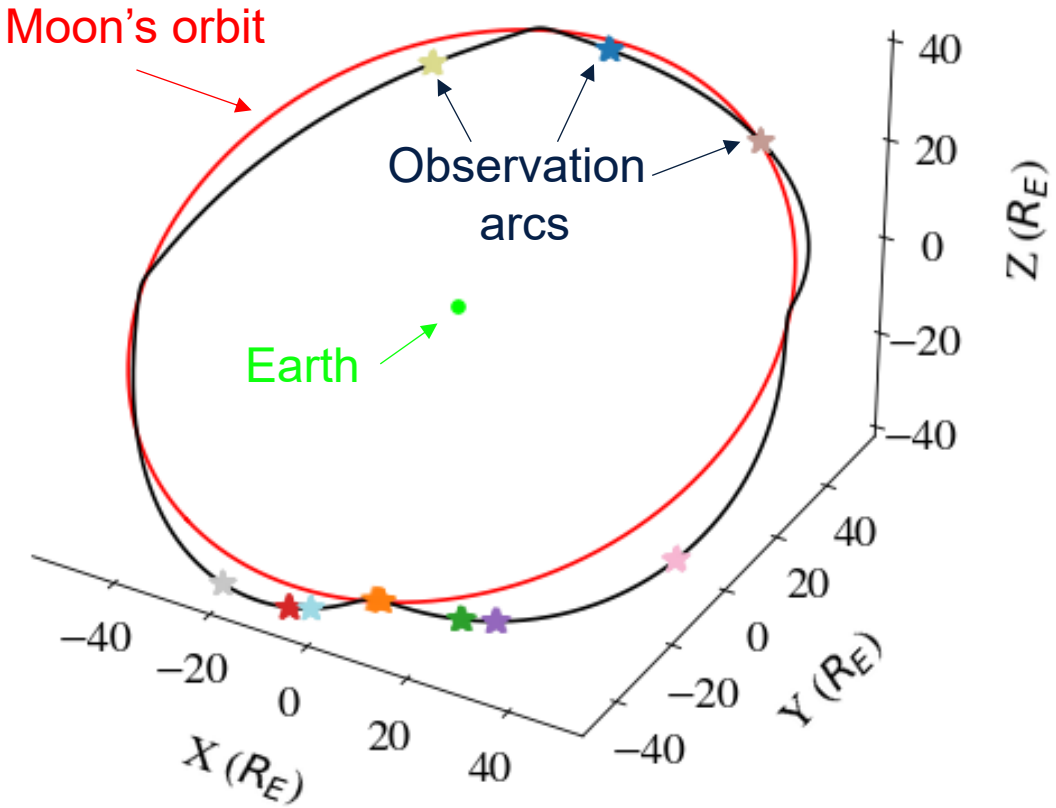
Each observation arc $\boldsymbol{Y}_k$ is composed of $N = 10$ observations with 2 **angular measurements** and with **various cadence** (from 5min to 15 min):

$$\mathbf{Y}_k = [\ldots, t_i, \theta_i, \varphi_i, \ldots] \; i = 1, \ldots, N$$

Dataset of 4119 observation arcs



Observation scenario from topocentric Reference Frame.



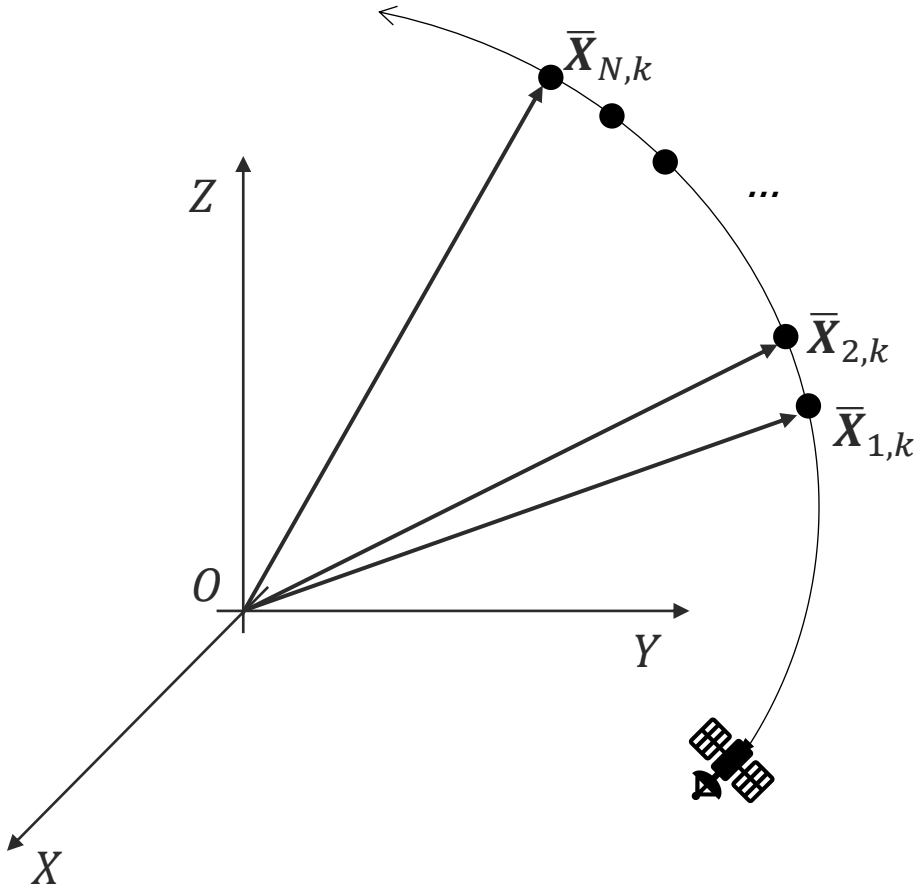
Observation arcs of the target orbit in the Geocentric Inertial Reference Frame.

Orbit Determination
Probabilistic modelling

The **initial state** $\bar{X}_{1,k}$ of the observation arc k is the realisation of a **probability distribution**, **conditioned** to the topocentric **angles-only measurements** $\boldsymbol{Y}_k$.

$$\bar{\mathbf{X}}_{1,k} \sim p(\bar{\mathbf{X}}_{1,k}|\mathbf{Y}_k)$$

$$\mathbf{Y}_k = [\ldots, t_i, \theta_i, \varphi_i, \ldots] \; i = 1, \ldots, N$$



Detail of an observation arc in the Geocentric Inertial Reference Frame.

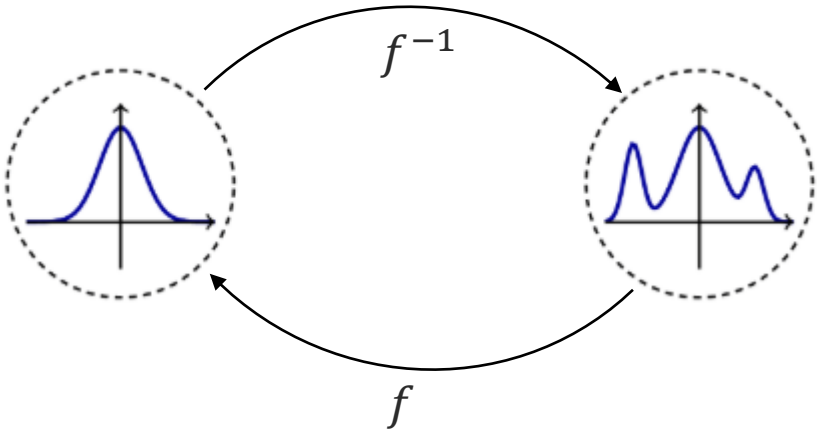
Generative Orbit Determination Framework
Conditional Normalising Flow Models

Learn an invertible and differentiable transformation $f : \mathbb{R}^6 \rightarrow \mathbb{R}^6$

Forward process: $\mathbf{w} = f(\bar{\mathbf{X}}_{1,k}|\mathbf{Y}_k)$

Inverse process: $\bar{\mathbf{X}}_{1,k} = f^{-1}(\mathbf{w}|\mathbf{Y}_k) = g(\mathbf{w}|\mathbf{Y}_k)$

Real Non-Volume Preserving Normalising Flow (**Real-NVP**)



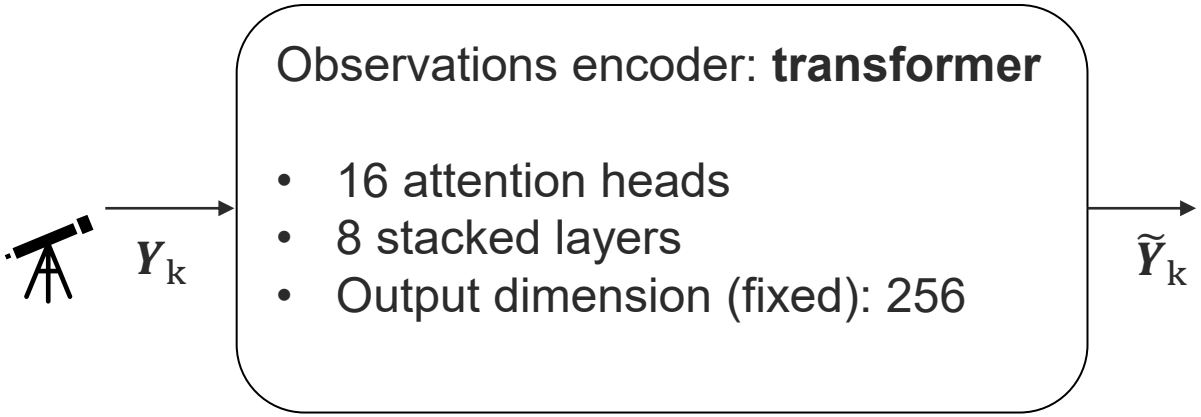
Training Function: conditional log-likelihood

$$\mathcal{L}_{\text{flow}} = -\frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \log p(\bar{\mathbf{X}}_{1,k}|\mathbf{Y}_k)$$

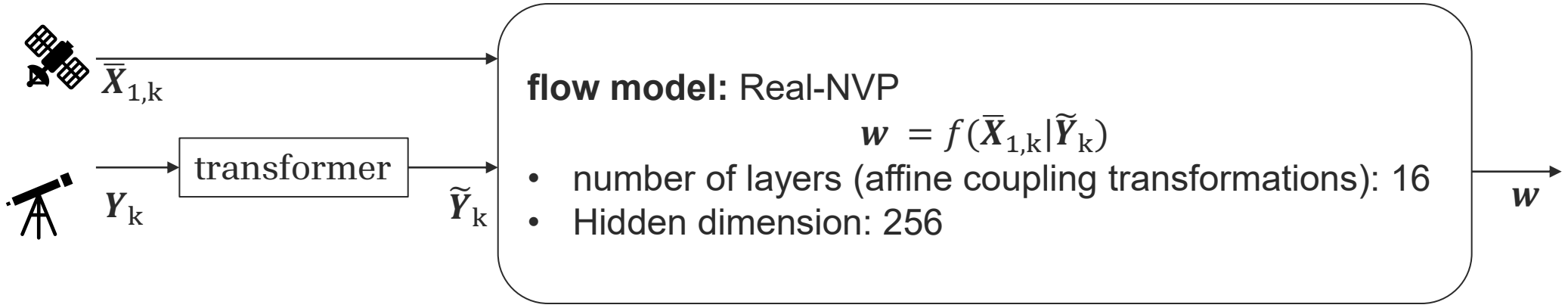
$n_{train} = 3319$

$$\log p(\bar{\mathbf{X}}_{1,k}|\mathbf{Y}_k) = \log p(\mathbf{w}) + \log \left|\mathbf{J}(f(\bar{\mathbf{X}}_{1,k}|\mathbf{Y}_k))\right|$$

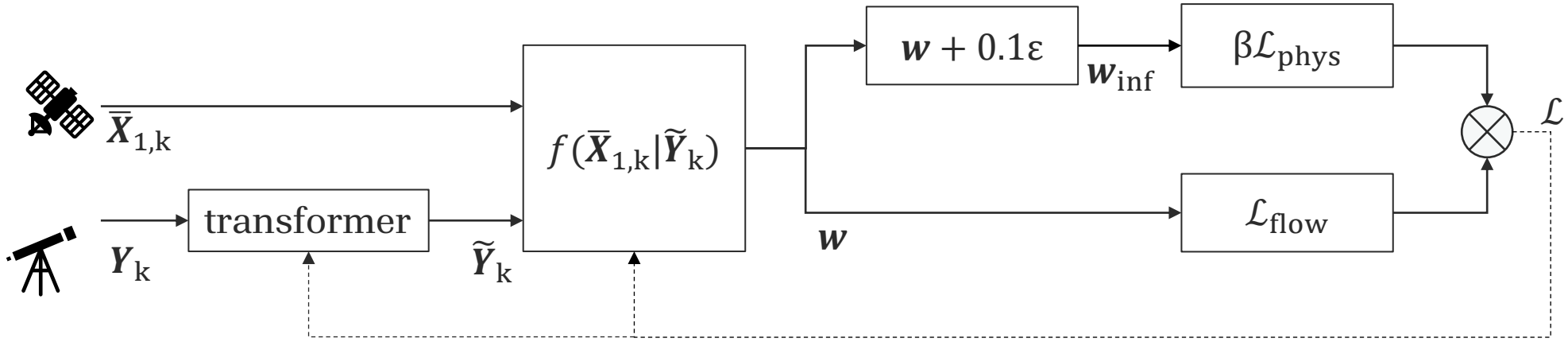
Generative Orbit Determination Framework
Architecture components



Generative Orbit Determination Framework
Architecture components



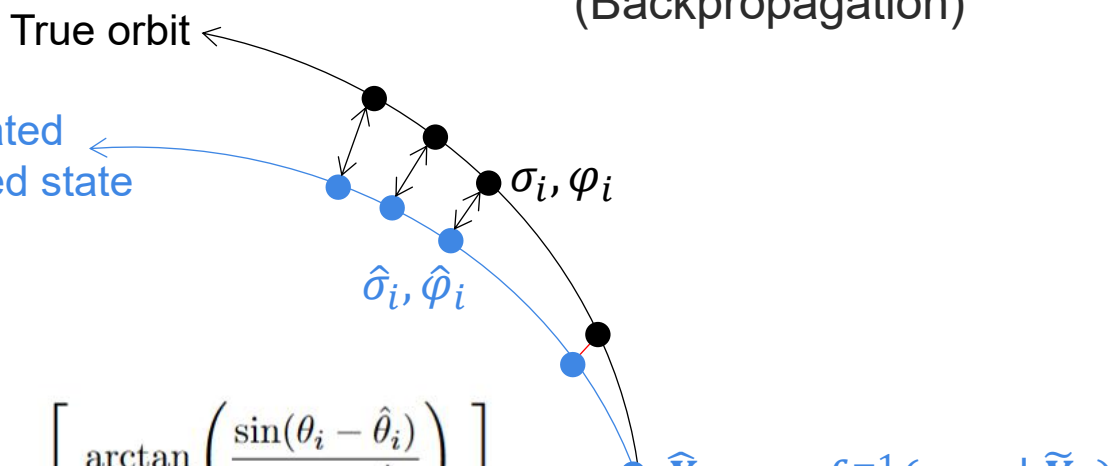
Generative Orbit Determination Framework
Variational Physics-informed Training



$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \beta\,\mathcal{L}_{\text{phys}}$$

$$\mathcal{L}_{\text{phys}} = \frac{1}{n_{\text{train}}} \sum_{k=1}^{n_{\text{train}}} \|\bar{\boldsymbol{\sigma}}_k\|^2 \qquad n_{train} = 3319$$

$$\bar{\boldsymbol{\sigma}}_k = \begin{bmatrix} \boldsymbol{\sigma}_1 \\ \cdots \\ \boldsymbol{\sigma}_i \\ \cdots \\ \boldsymbol{\sigma}_N \end{bmatrix} \qquad \boldsymbol{\sigma}_i = \begin{bmatrix} \arctan\left(\dfrac{\sin(\theta_i - \hat{\theta}_i)}{\cos(\theta_i - \hat{\theta}_i)}\right) \\ \varphi_i - \hat{\varphi}_i \end{bmatrix}$$



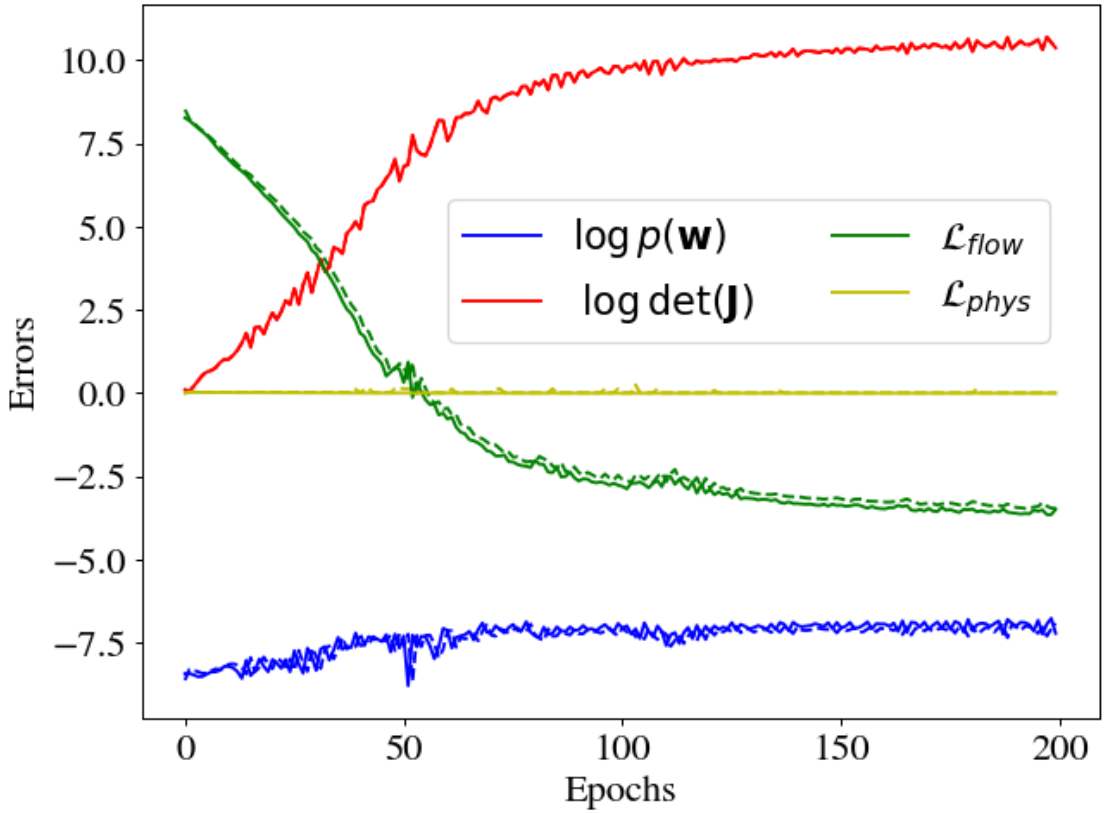
Results
Training Dynamics

Experimental setup

Dataset	Number of samples
Training	3119
Validation	500
Test	500

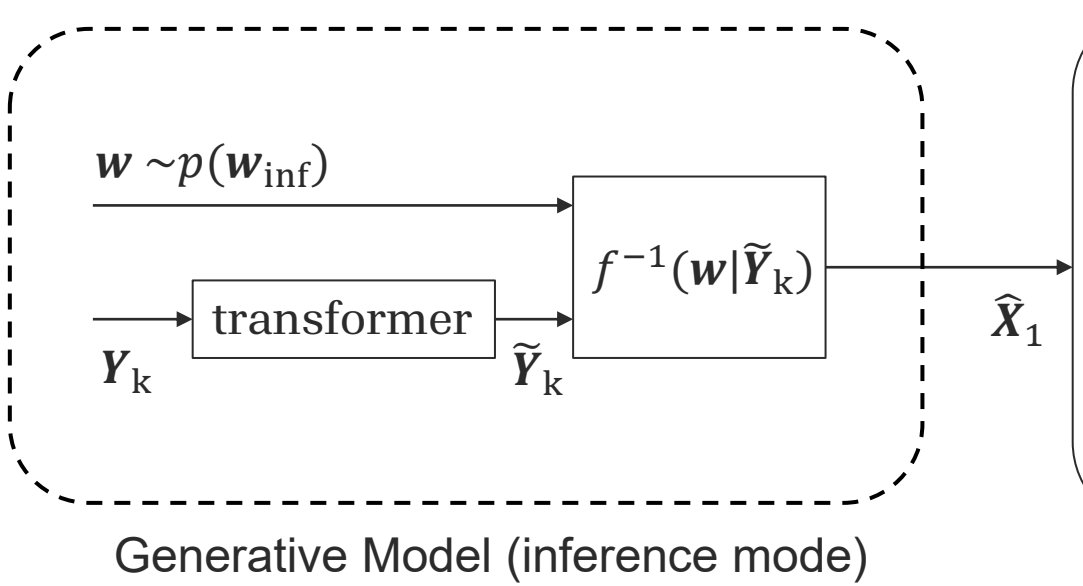
Subset	$\log p(\mathbf{w})$	$\log \lvert\mathbf{J}\rvert$	$\mathcal{L}_{\text{flow}}$	$\mathcal{L}_{\text{phys}}$
Training	-7.0761	10.5315	-3.4513	8.052×10^{-3}
Validation	-6.7635	10.5315	-3.6475	9.090×10^{-4}

Loss Function components at final iteration.



Training dynamics. Training set in dashed line; validation set in dashed line.

Results
Estimation assessment



Generative Model (inference mode)

Refinement approach

Nonlinear Least Squares Estimation (NLSE)

$$\hat{\mathbf{X}}_1^j = \hat{\mathbf{X}}_1^{j-1} - \lambda \mathbf{J}^{\dagger} \boldsymbol{\sigma}^{j-1}, \qquad j = 2, \ldots, N_{\max}$$

- Convergence tolerance: 1×10^{-9}
- Maximum of Iterations: $N_{\max} = 100$
- Tolerance on the Residual: 1×10^{-9}

$\widehat{\boldsymbol{X}}_1^r, \overline{\boldsymbol{\sigma}}^r$

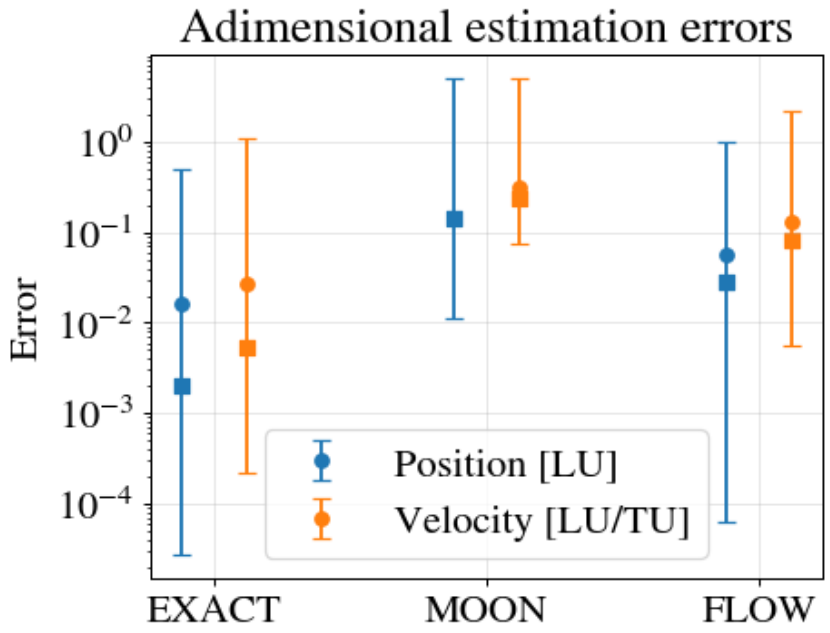
$\overline{\boldsymbol{X}}_1$

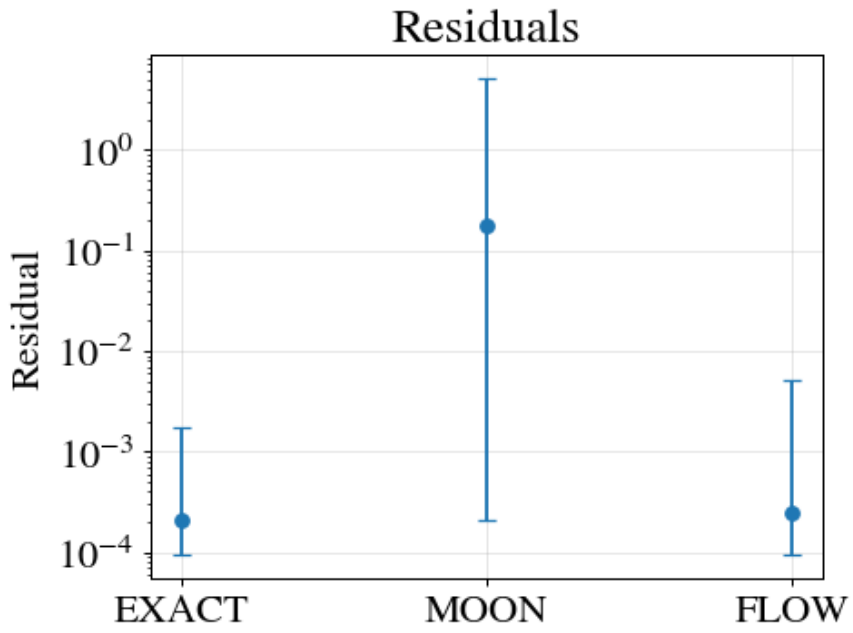
True initial state

Assessment Metrics

- Position error
- Velocity error
- Post-estimation residual

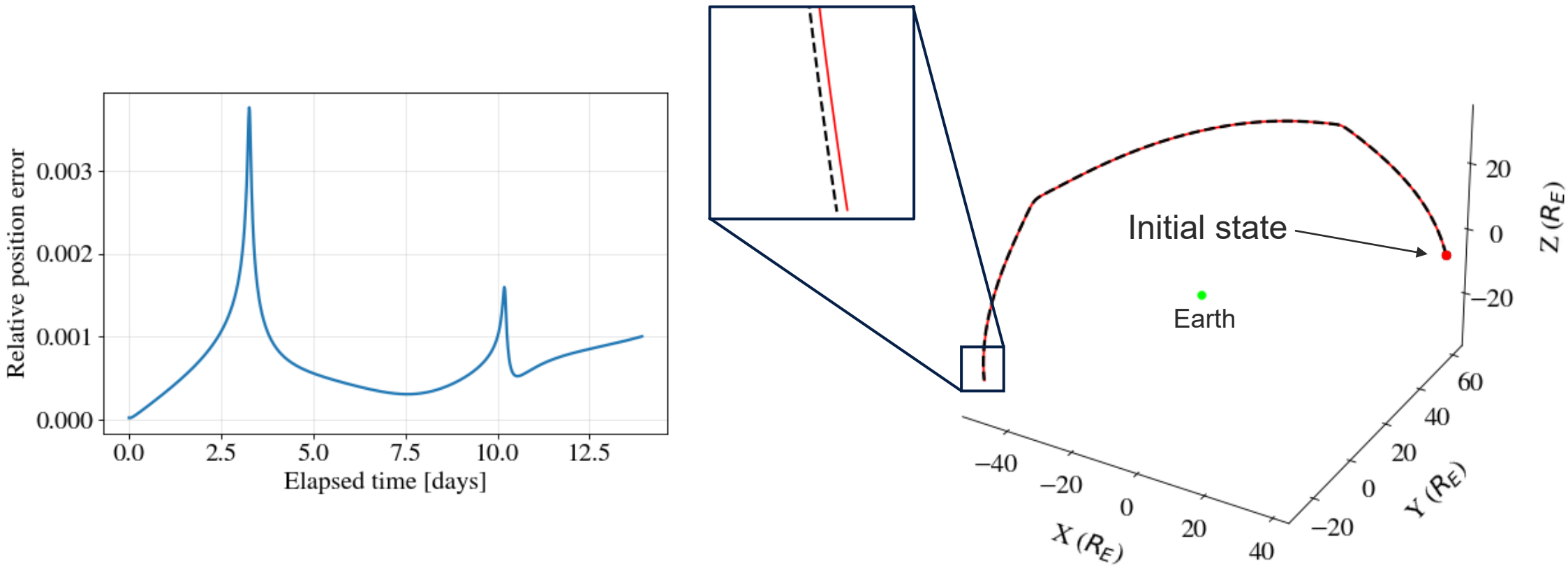
Results
Estimation assessment





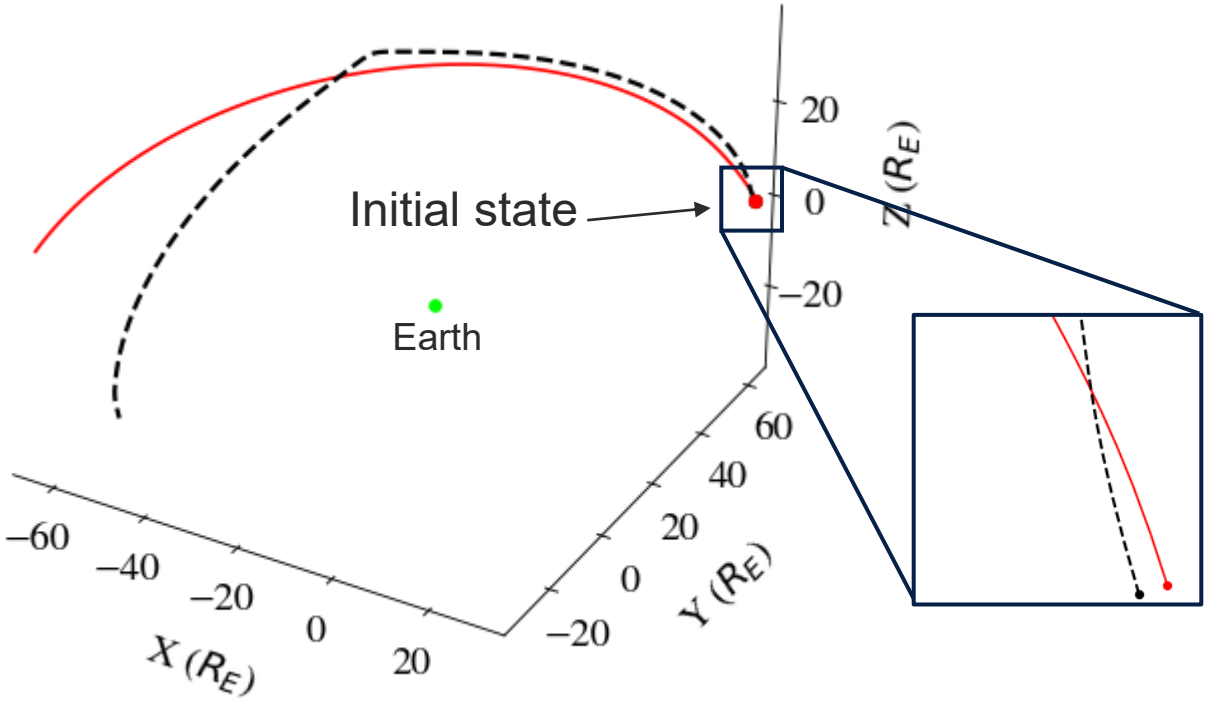
Assessment metrics statistics over the test subset (500 samples).
NLSE is **initialised** with the **true state**, the **Moon**'s state, and $\widehat{X}_1$.
Dots – mean values; squares – median values.

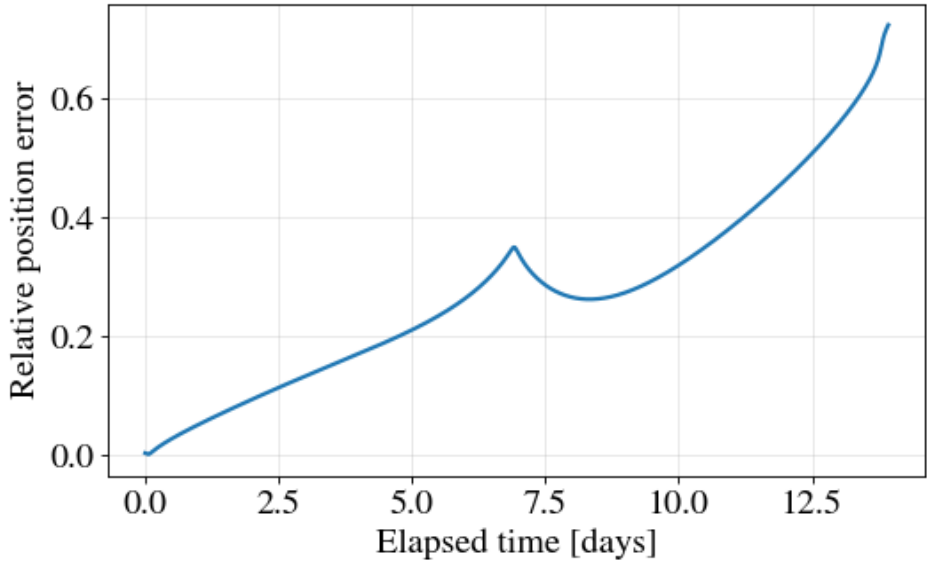
Results
True state NLSE initialisation: propagation estimation



Propagation of the least-position error estimation at initial time instant.
Left: relative position error. Right: trajectory in geocentric Inertial RF.
Black: true orbit. Red: reconstructed orbit.

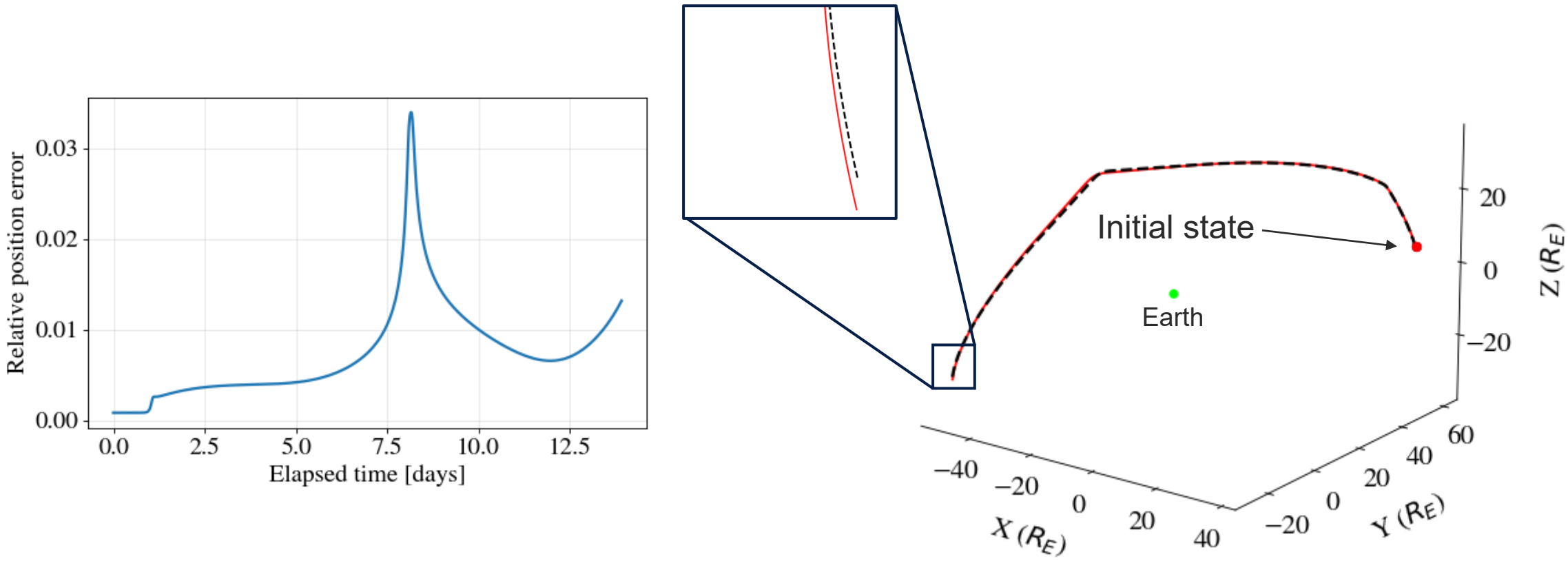
Results
Moon's state NLSE initialisation: propagation estimation





Propagation of the least-position error estimation at initial time instant.
Left: trajectory in geocentric Inertial RF. Right: relative position error.
Black: true orbit. Red: reconstructed orbit.

Results
Flow Model state NLSE initialisation: propagation estimation



Propagation of the least-position error estimation at initial time instant.
Left: relative position error. Right: trajectory in geocentric Inertial RF.
Black: true orbit. Red: reconstructed orbit.

References

- Walther Litteri, Massimiliano Vasile (2026) Physics-informed Conditional Normalizing Flows for Angles-only Cislunar Orbit Determination, https://arxiv.org/abs/2606.30936

Questions?





University of Strathclyde Glasgow

The University of Strathclyde is a charitable body, registered in Scotland, with registration number SC015263