Searching for Special Lagrangian cycles in Calabi-Yau Manifolds

FABIAN RUEHLE (NORTHEASTERN UNIVERSITY & IAIFI)

ESI Vienna
The Unreasonable Effectiveness of Toric Geometry
07/09/2026

Based on:

[Ahmed, Qi, FR: 260x.xxxxx]

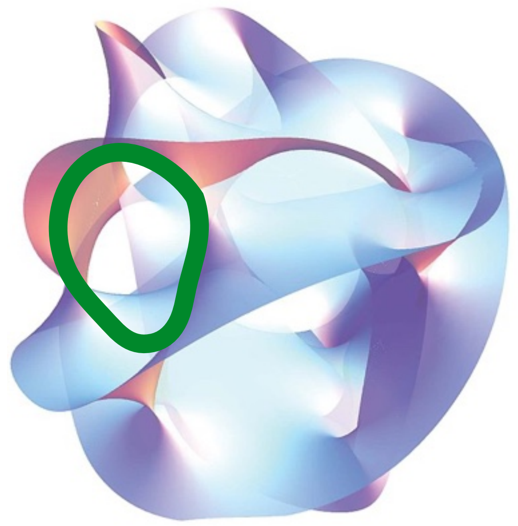
[Larfors, Lukas, FR, Schneider: 2205.13408]

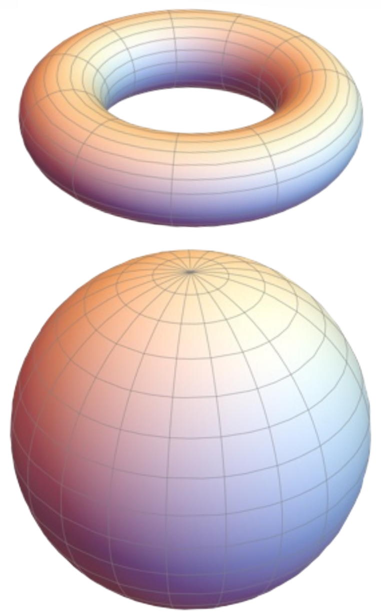
[Larfors, Lukas, FR, Schneider: 2111.01436]

[FR: Data Science Applications to String Theory (2020)]

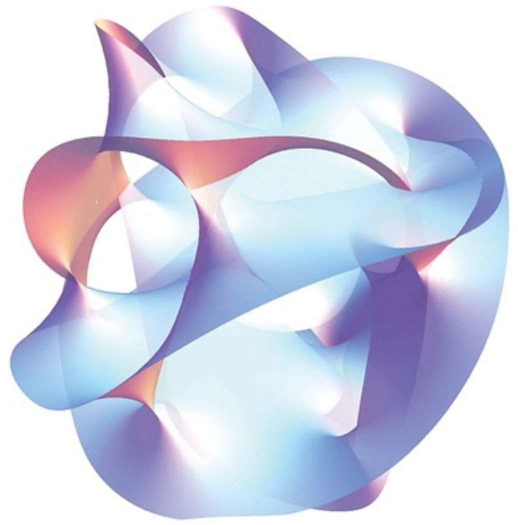
Outline

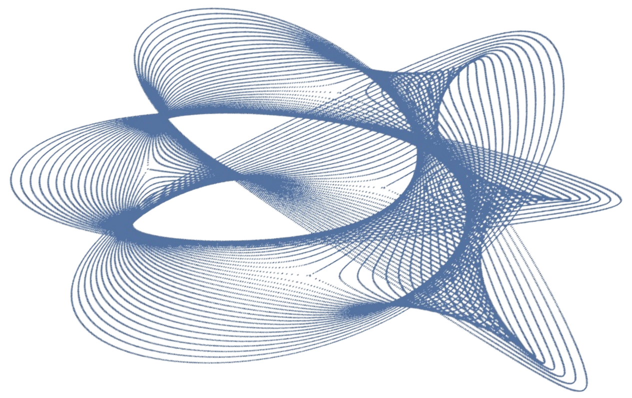
1 Introduction to sLags



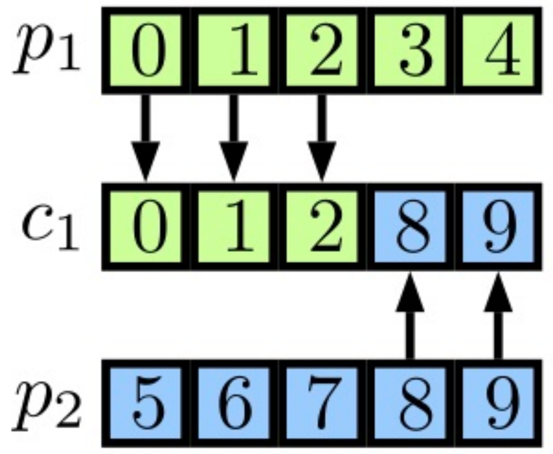


2 CY Metrics

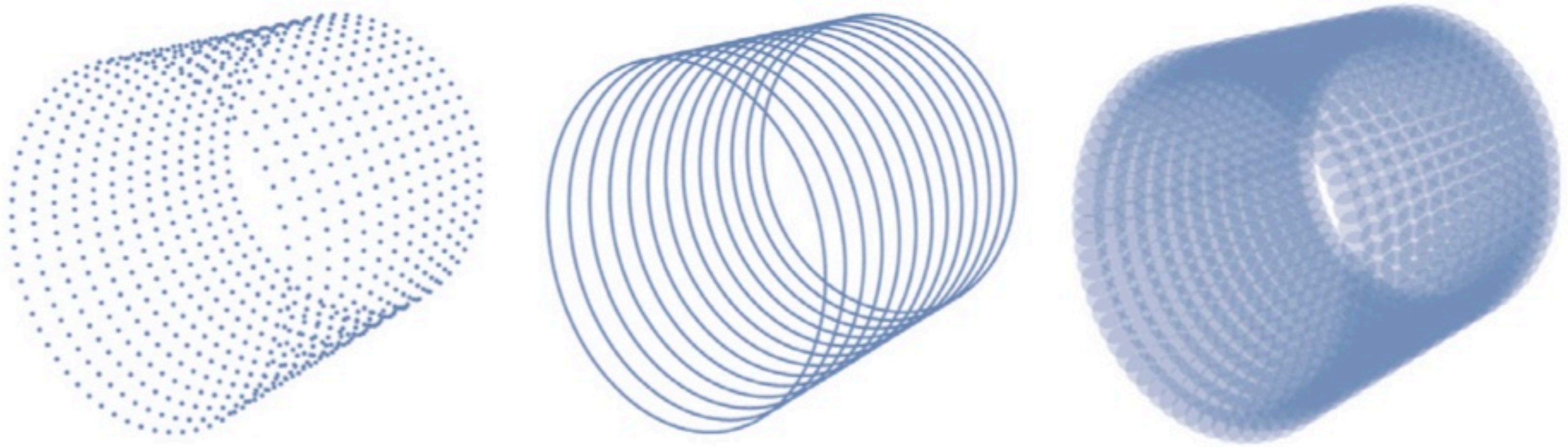


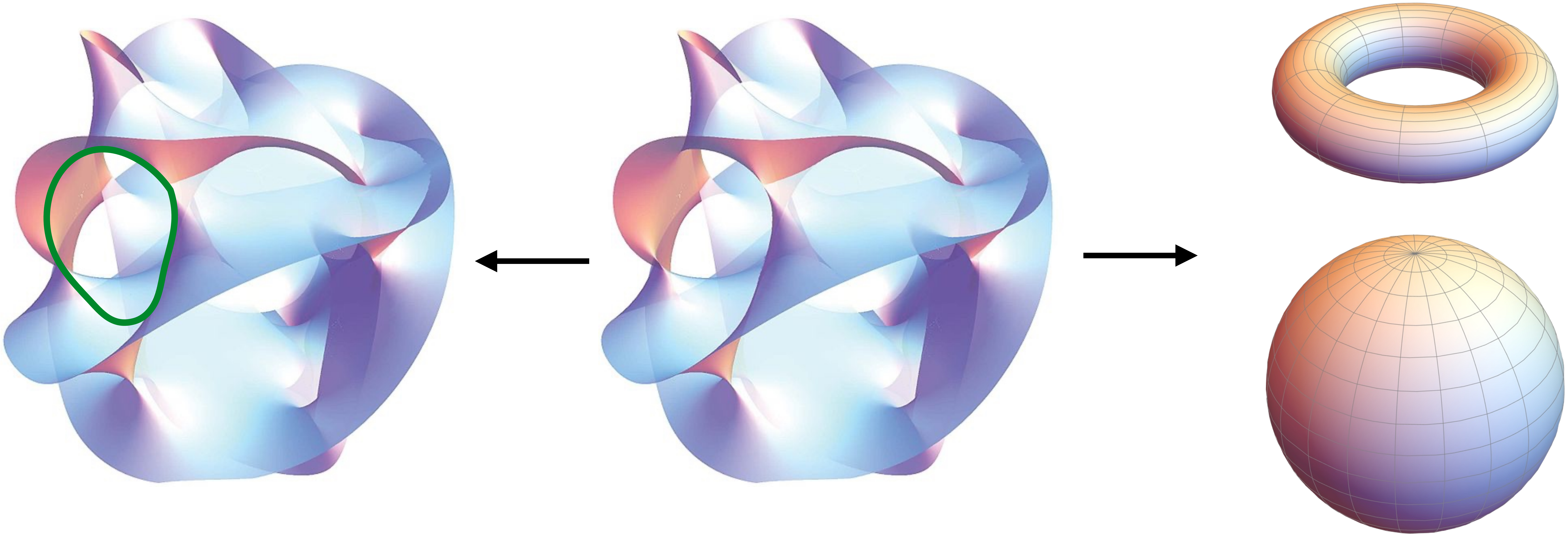


3 Evolutionary Algorithms



4 Persistent Homology





Introduction to Special Lagrangian Cycles

sLags - Definition

- On a Calabi-Yau threefold, a special Lagrangian cycle $\mathcal{L}$ is a real three-dimensional submanifold that satisfies two conditions:
  - **Special:** The pullback of the holomorphic threefrom has constant phase

$$\mathrm{Im}(e^{i\theta}\Omega|_{\mathcal{L}}) = 0$$

  - **Lagrangian:** The pullback of the Kähler form vanishes

$$J|_{\mathcal{L}} = 0$$

- Important in String Theory (BPS) and Math (SYZ)

Example: Quintic T3

- Focus on Dwork Quintic: $z_0^5 + z_1^5 + z_2^5 + z_3^5 + z_4^5 - 5\psi z_0 z_1 z_2 z_3 z_4$
- In their seminal Mirror Symmetry paper, Candelas et.al. describe the following three-cycle **[Candelas, de la Ossa, Green, Parkes `91]**

$$B_2 = \{z_k \mid z_0 = 1, |z_1| = |z_2| = |z_3| = \delta, z_4 \text{ is given by the solution } p(z) = 0 \text{ that tends to } 0 \text{ as } \psi \to \infty\}$$

- Let us study this cycle in more detail. Note that $z_j = \delta e^{i\theta_j} \quad \Rightarrow \quad dz_j = i z_j d\theta_j$
  - Special: $\Omega = \dfrac{dz_1 \wedge dz_2 \wedge dz_3}{(\partial p)/(\partial z_4)} = \dfrac{(i)^3 z_1 z_2 z_3 d\theta_1 \wedge d\theta_2 \wedge d\theta_3}{5z_4^4 - 5\psi z_1 z_2 z_3} \xrightarrow{\psi \to \infty} \dfrac{i}{5\psi} d\theta_1 \wedge d\theta_2 \wedge d\theta_3$
  - Lagrangian: $J|_{\mathcal{L}} = \dfrac{i}{2} g_{i\bar{j}} dz^i \wedge d\bar{z}^{\bar{j}} = -\dfrac{i}{2} g_{i\bar{j}} \delta^2 e^{i(\theta^i - \theta^j)} d\theta^i \wedge d\theta^j = 0$

Example: Quintic RP3

- Becker-Becker-Strominger identify a family of $\mathbb{RP}^3$ at $\psi = 0$

[Harvey, Lawson `82; Becker, Becker, Strominger `95]

$$\mathcal{L} = \{z_k \mid z_0 = 1, z_1 = r_1, z_2 = r_2, z_3 = r_3, z_4 \text{ solves } p(z) = 0\}$$

- Using similar arguments to the ones we showed before, this is also a sLag
- Besides these examples, I am not aware of any other explicitly known sLags on proper CY threefold
- We want to construct new examples using search+optimization techniques, and check the topology using topological data analysis

Ansatz for sLags

- We make the following ansatz for the equations that define the sLag

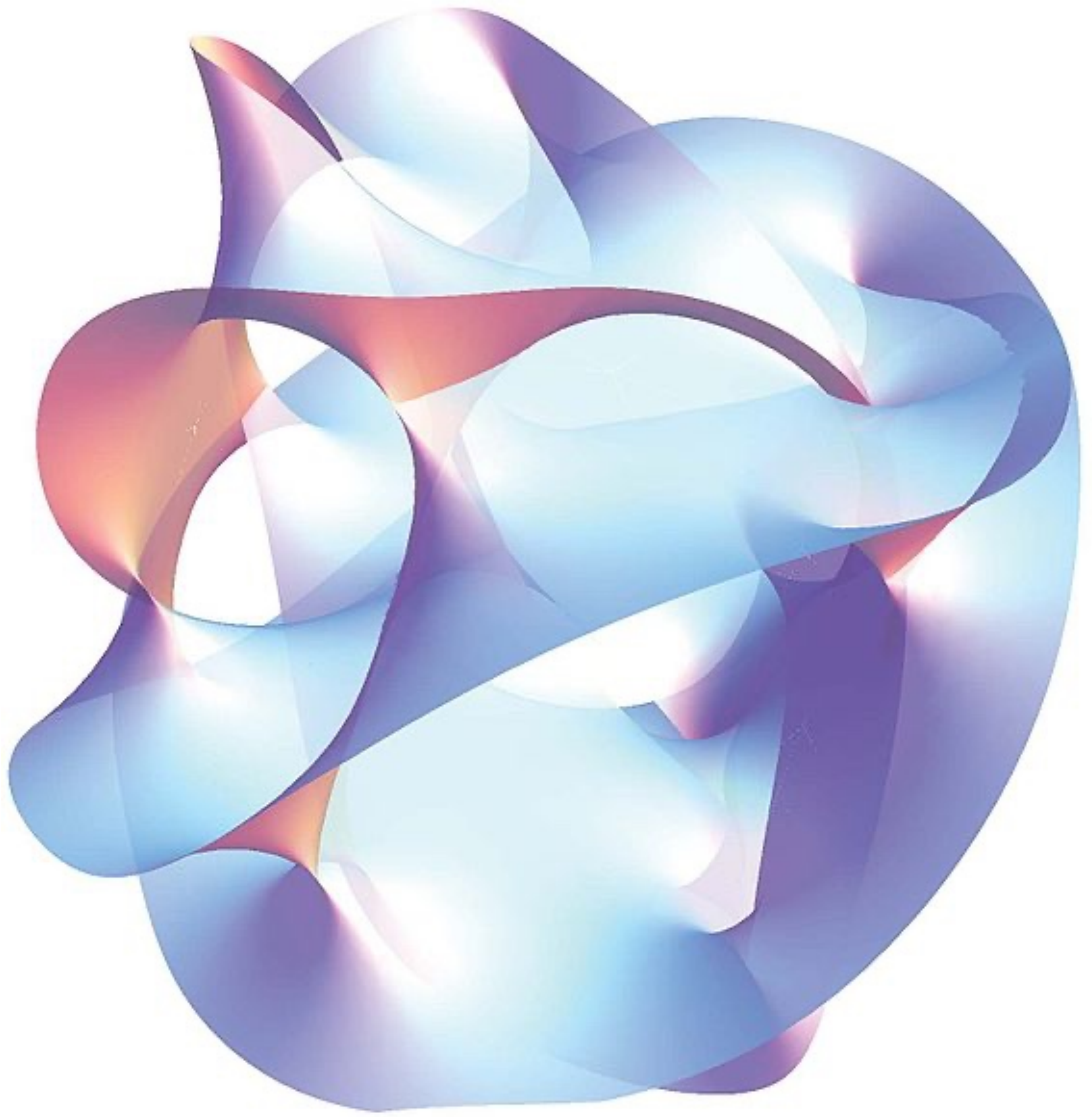
$$\vec{z}^{\,\dagger} H_i\, \vec{z} = 0\,, \quad H_i \text{ are } 5\times 5 \text{ hermitian}\,, \quad i = 1,2,3$$

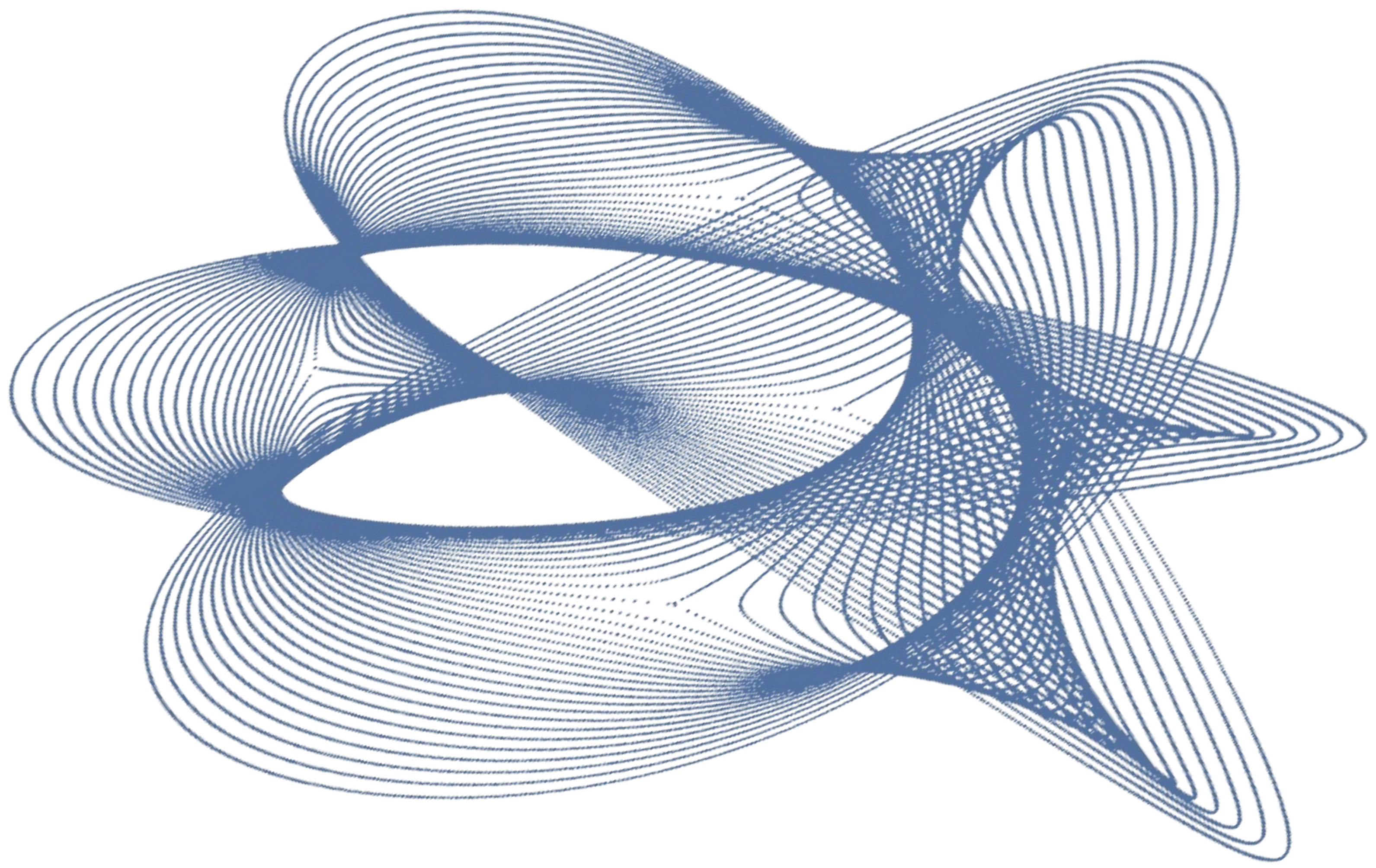
- These are a set of 3 parameterized (1,1) bi-holomorphic polynomials
- In addition, we impose the CY (hypersurface) condition $p(z) = 0$
- This gives 3 real constraints, hence defines a real three-cycle
- Note that the two sLags discussed above can be written this way
- The ansatz can be generalized to $\vec{s}^{\,\dagger} H_i\, \vec{s} = 0\,, s \in \Gamma(\mathcal{O}_{\mathbb{P}^4}(k))\,, \quad k > 1$
  The ansatz at the next-highest k contains the previous ansatz

Algorithm for finding sLags

The optimization algorithm proceeds as follows:

1. Find points on the CY hypersurface
2. Find/approximate CY metric (optional, but required for string applications)
3. Arbitrarily initialize the entries in H_i
4. Select the points from step 1 that are closest to the 3-cycle and use these as initial points for Newton's algorithm to "project" them onto the three-cycle
5. Compute the special and Lagrangian condition
6. Move the sLag by changing the parameters in H_i such that the conditions in 5 are closer to being satisfied
7. Repeat steps 4 - 6 until no improvement
8. Use the H_i to initialize the H_i for the next-higher ansatz $k \to k+1$ and repeat 4 - 8





Calabi-Yau Metrics

Finding points on the CY

- Note that on every CY: $J \wedge J \wedge J = \kappa \Omega \wedge \bar{\Omega}$ (since $h^{3,0} = h^{0,3} = h^{3,3} = 1$)
- We want a uniform point sample on the CY
- We can get a point sample that is uniform with respect to the (pullback of the) FS metric using a theorem by [Shiffman,Zelditch `98] , but this requires working with a Veronese-type embedding of the Kähler cone sections and can be unfeasible at large $h^{1,1}$ or for toric models
- Alternatively, we can use Markov-Chain Monte Carlo to get a point sample that is uniform w.r.t. the CY metric (by sampling from $\Omega \wedge \bar{\Omega}$)

Approximating the CY Metric

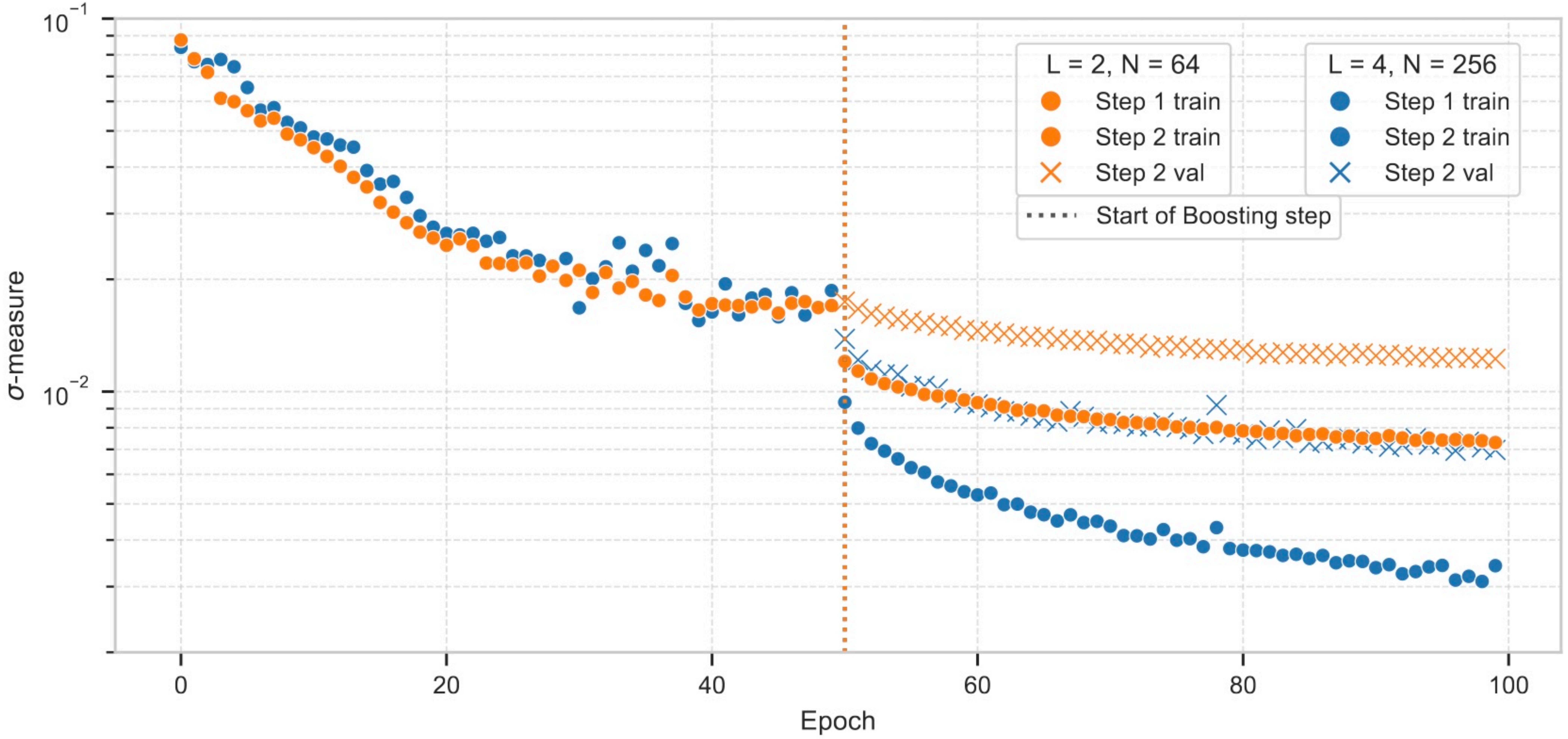
- To approximate the CY metric, we use so-called Physics Informed Neural Networks (PINNs)
- We make an ansatz for the CY metric (or rather, its Kähler potential):

$$K_{\text{CY}} = K_{\text{FS}} + \phi = \ln \sum |z_i|^2 + \phi \qquad \Rightarrow \qquad g_{\text{CY}} = g_{FS} + \partial\bar{\partial}\phi$$

- Here, ϕ is a real function parameterized by a Neural Network (which itself is a parameterized map from $\blacksquare\blacksquare^4 \to \blacksquare$), and we take derivatives of the NN w.r.t. its inputs
- To get the NN to represent the correction that makes the metric CY, we adjust its parameters by minimizing the Monge-Ampere loss

$$J \wedge J \wedge J = \kappa \Omega \wedge \bar{\Omega} \qquad \Rightarrow \quad \mathcal{L}_{\text{MA}} = \frac{1}{N} \sum_{a=1}^{N} \left| 1 - \kappa \frac{J^3(z_a)}{|\Omega(z_a)|^2} \right|$$

Approximating the CY Metric



[Lust, FR, Schreier, to appear]

Evolutionary Algorithms

Genetic Algorithms 101

[FR `20]

1. Define a genome/alleles (parameters in H_i)
2. Initialize the first generation with a random population (collection of individuals)
3. Calculate the fitness (how close they are to defining a sLag) for each individual
4. Cross-breed the fittest individual by exchanging parts of the genome (parameters of H_i) to produce offspring for the next generation
5. Randomly mutate the offspring (random assignment of values to parameters in H_i)
6. Repeat steps 3 - 5 for the next generation

[FR `17; Cole, Schachner, Shiu `19; Cole, Krippendorf, Schachner, Shiu `21; Abel, Constantin, Harvey, Lukas `21 `22; Abel, Constantin, Harvey, Lukas, Nutricati `23; Berglund, He, Heyes, Hirst, Jejjala `23]

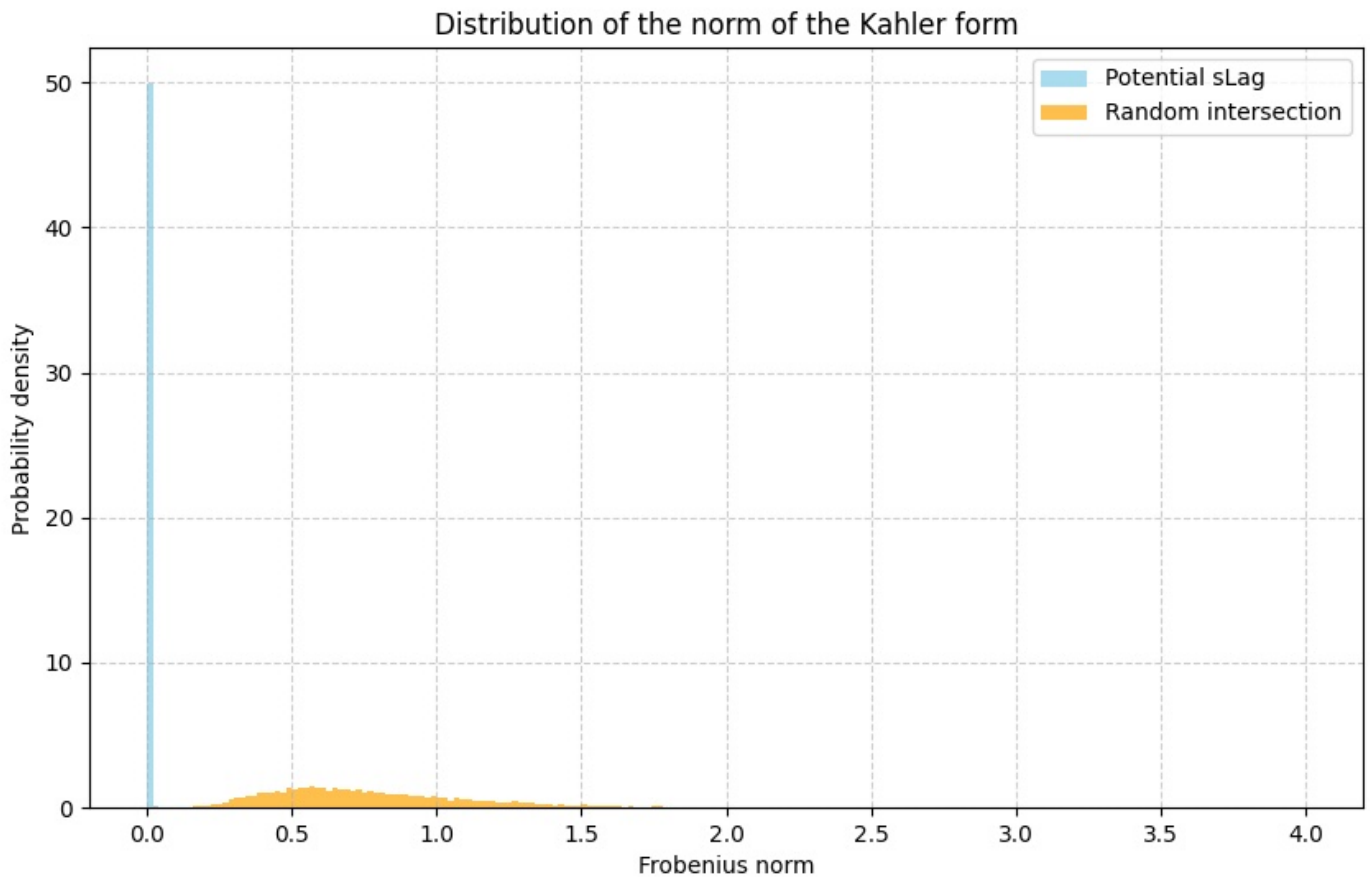
Fitness Function - Special Property

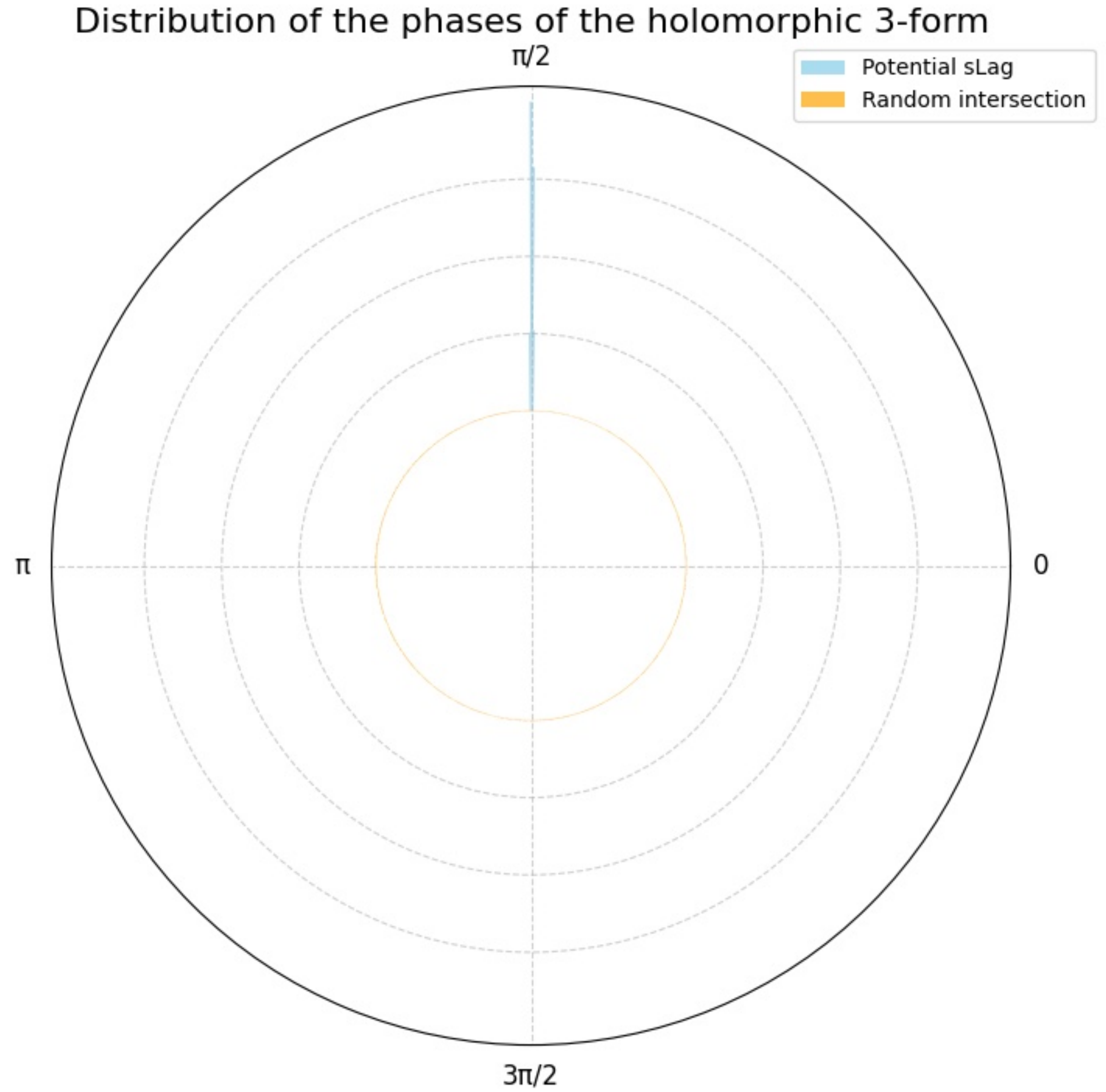
- **Special property:**
  - Compute the phases $\Omega|_{\mathcal{L}}$ for a set of points on $X \cap \mathcal{L}$
  - Bin them and compute the entropy $S = -\sum_x p(x) \log p(x)$ from the histogram
  - Define the fitness $f_{\text{special}} = 1 - \frac{S}{S_{\max}}$, where $S_{\max}$ is the entropy of the uniform distribution
  - Note that by definition $f_{\text{special}} \in [0,1]$, where 0 corresponds to a uniform distribution of phases and 1 corresponds to constant phase (all in the same bin)

Fitness Function - Lagrangian Property

- **Lagrangian property:**
  - Compute the Kähler forms $J|_{\mathcal{L}}$ for a set of points on $X \cap \mathcal{L}$
  - Compute the Frobenius norm $L = \frac{1}{N}\sum_{n=1}^{N}\sum_{i,j}|J_{i\bar{j}}(x^{(n)})|$
  - Define the fitness $f_{\mathrm{Lag}} = e^{-kL}\,,\ k \in \mathbb{R}_+$
  - Note that by definition $f_{\mathrm{Lag}} \in]0,1]$, with 0 corresponding to infinite Frobenius norm and 1 corresponding to $J|_{\mathcal{L}} = 0$
- **Total fitness:** $f_{\mathrm{tot}} = f_{\mathrm{special}} \times f_{\mathrm{Lag}}$
  - Note that by definition $f_{\mathrm{tot}} \in]0,1]$

Example - T^3



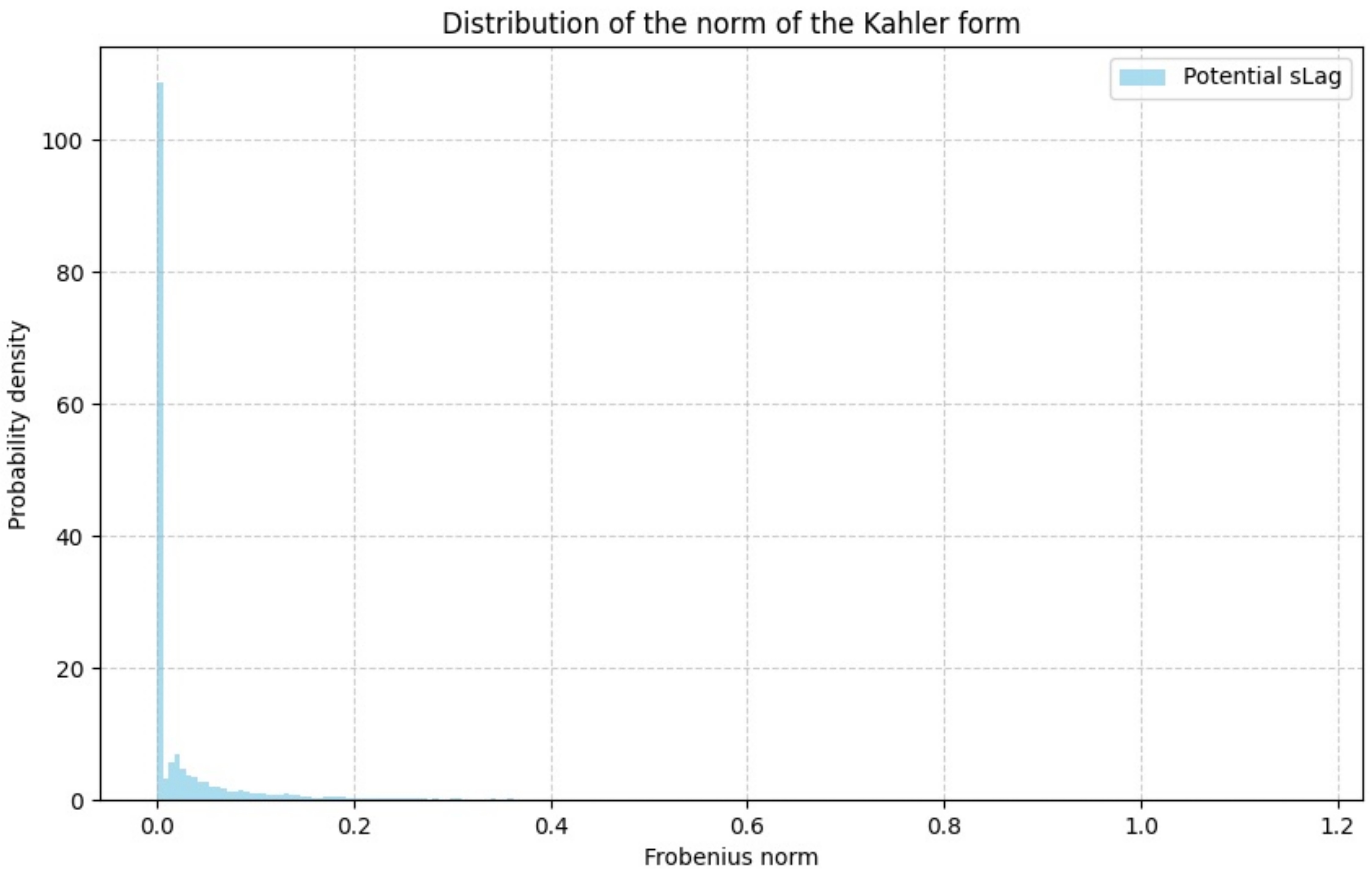


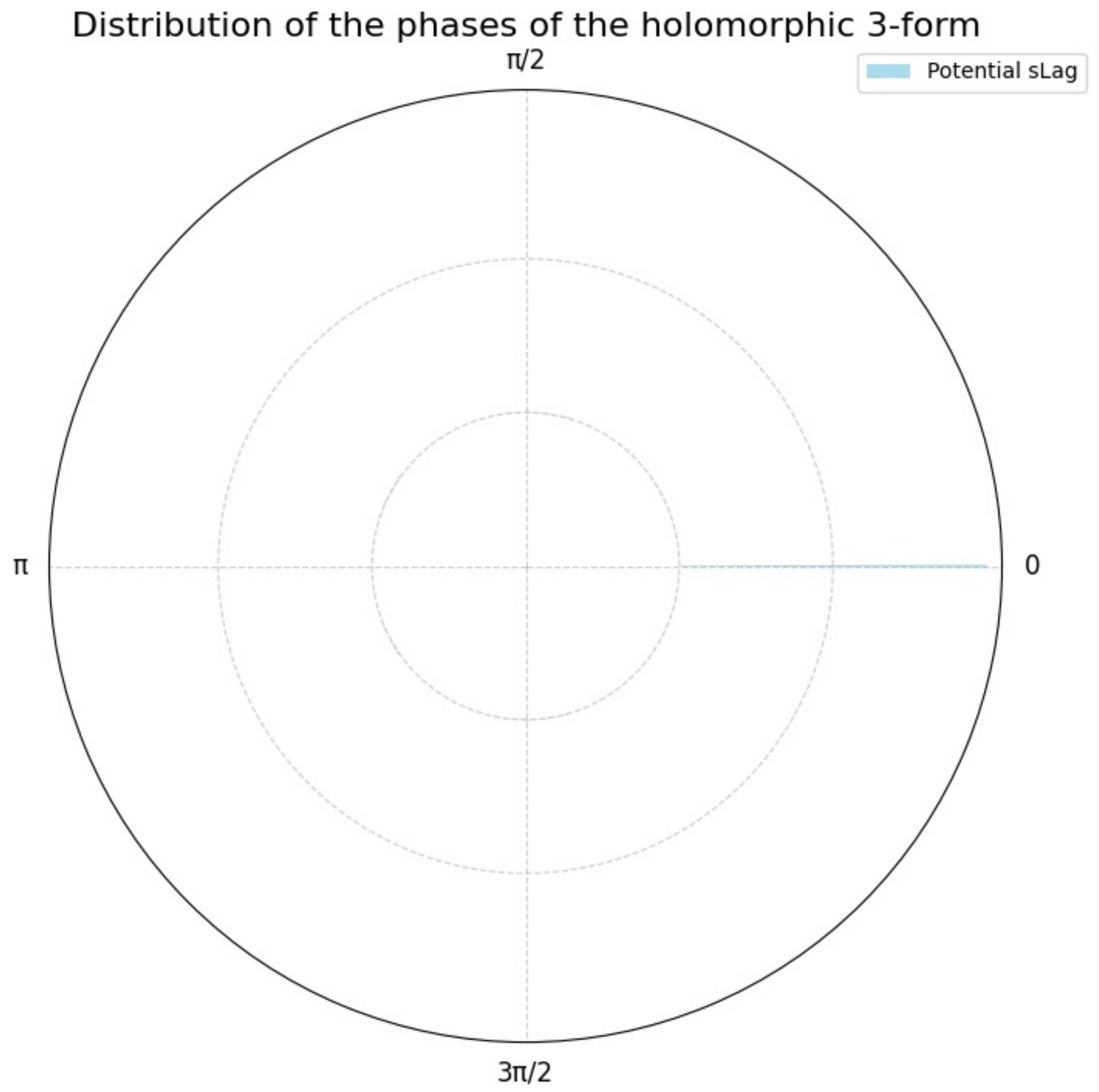
Lagrangian Fitness: 0.99999

Special Fitness: 1.0000

Total Fitness (at $\psi \geq 200$): 1.00

Example - $\mathbb{RP}^3$



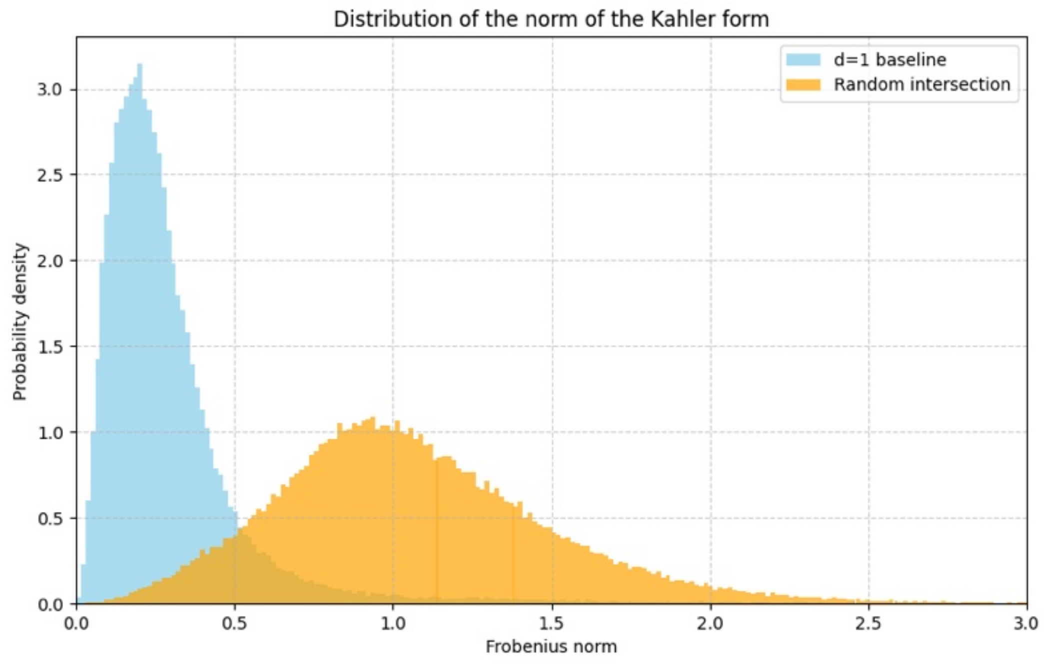


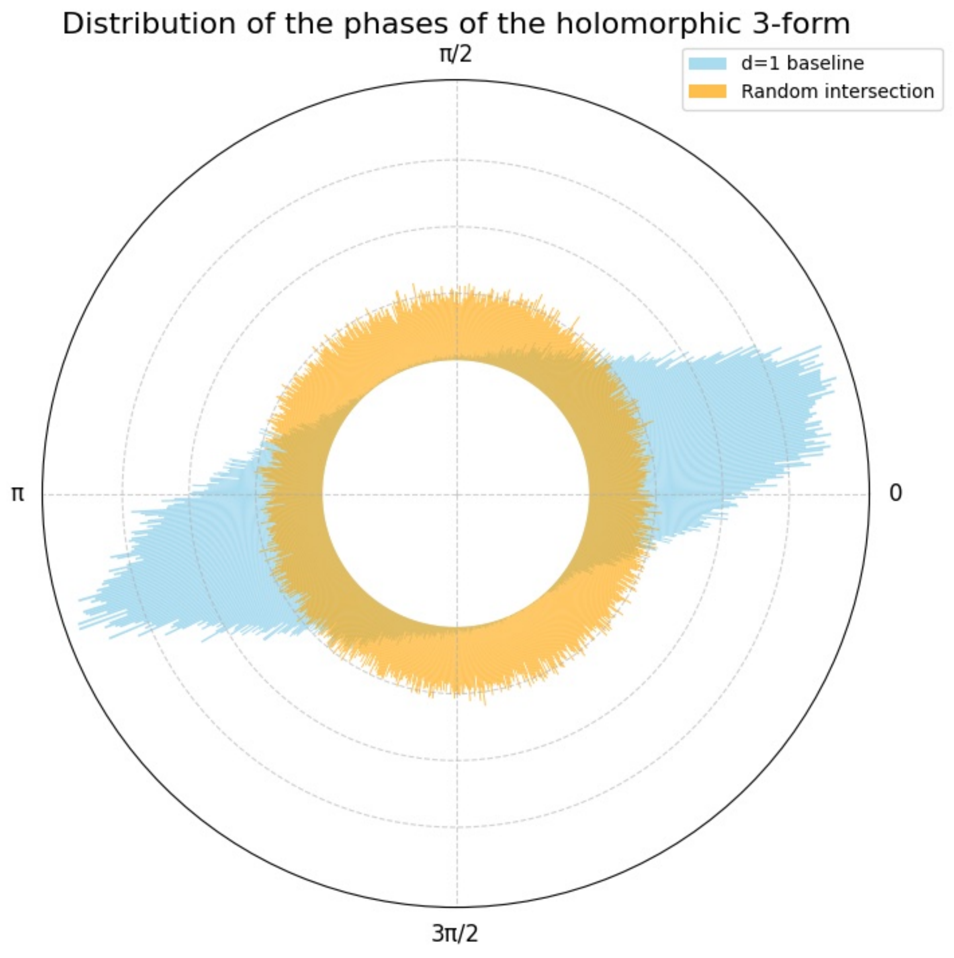
Lagrangian Fitness: 1.0000

Special Fitness: 1.0000

Total Fitness (at $\psi = 0$, branch where $\mathrm{Im}(z_4) = 0$): 1.0000

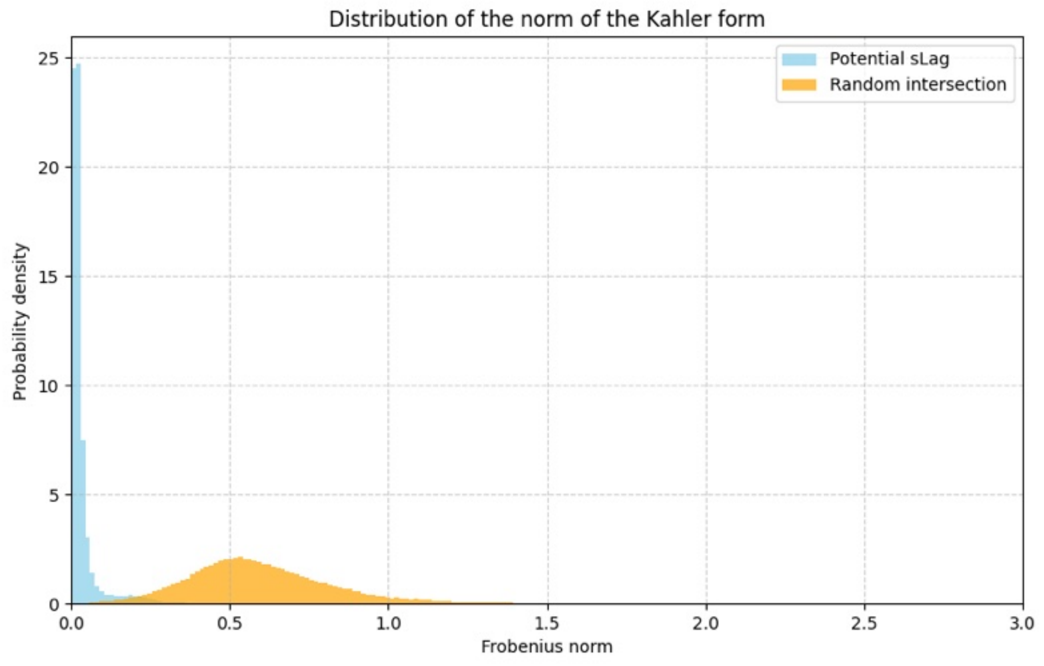
New sLag candidate

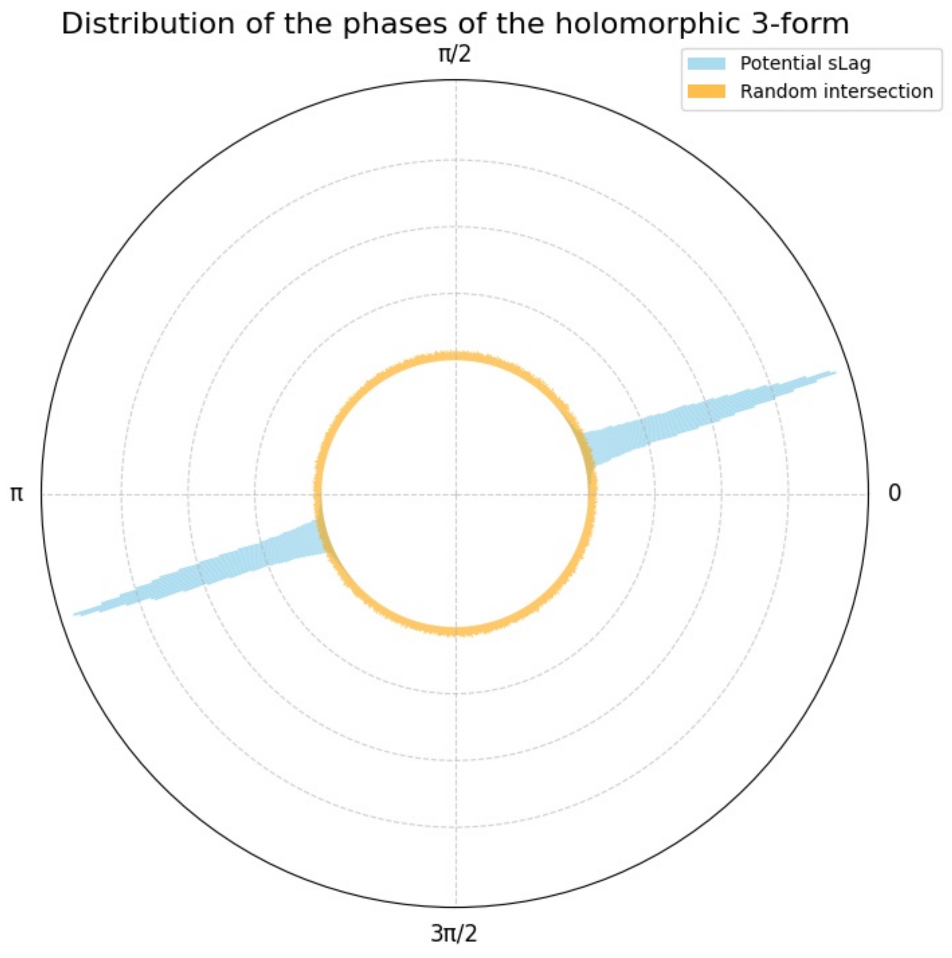




Lagrangian Fitness: 0.23

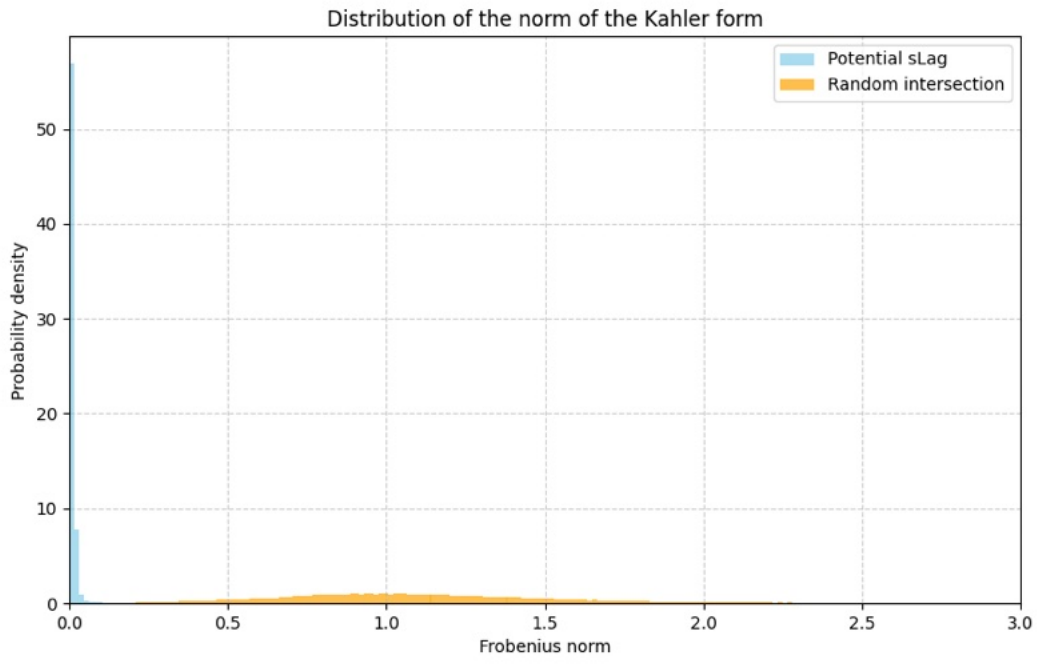
Special Fitness: 0.15

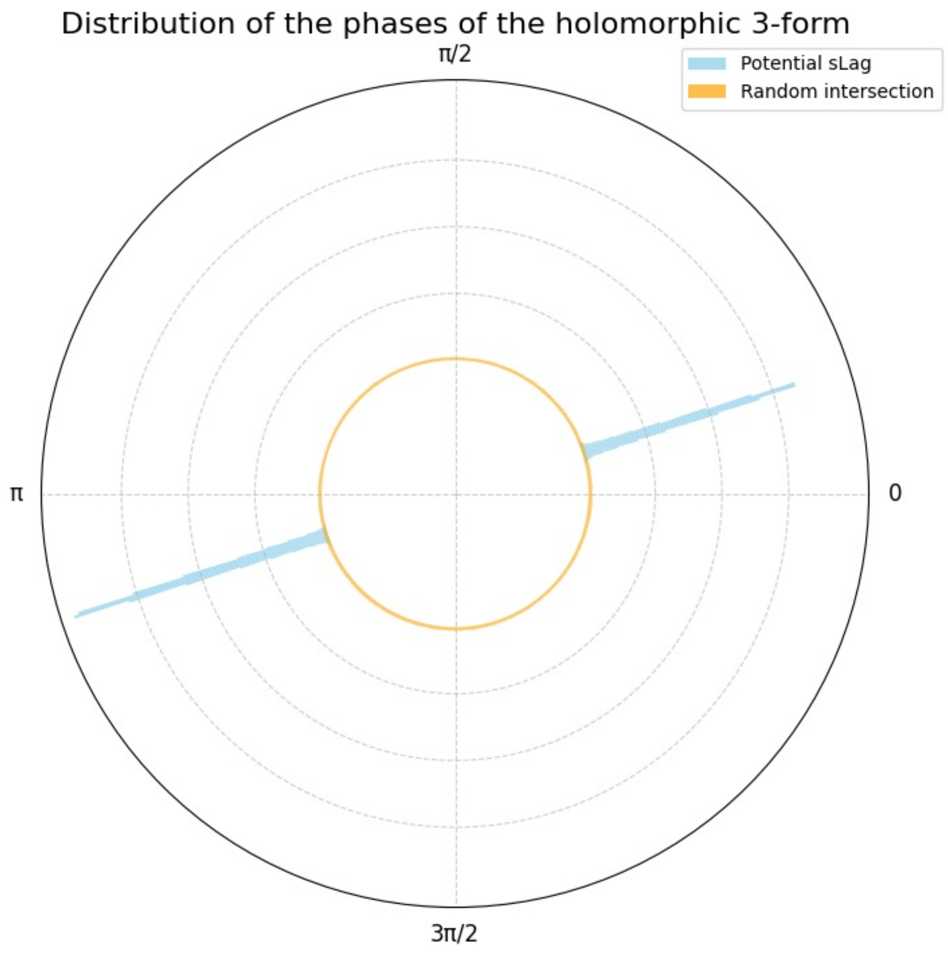




Lagrangian Fitness: 0.80

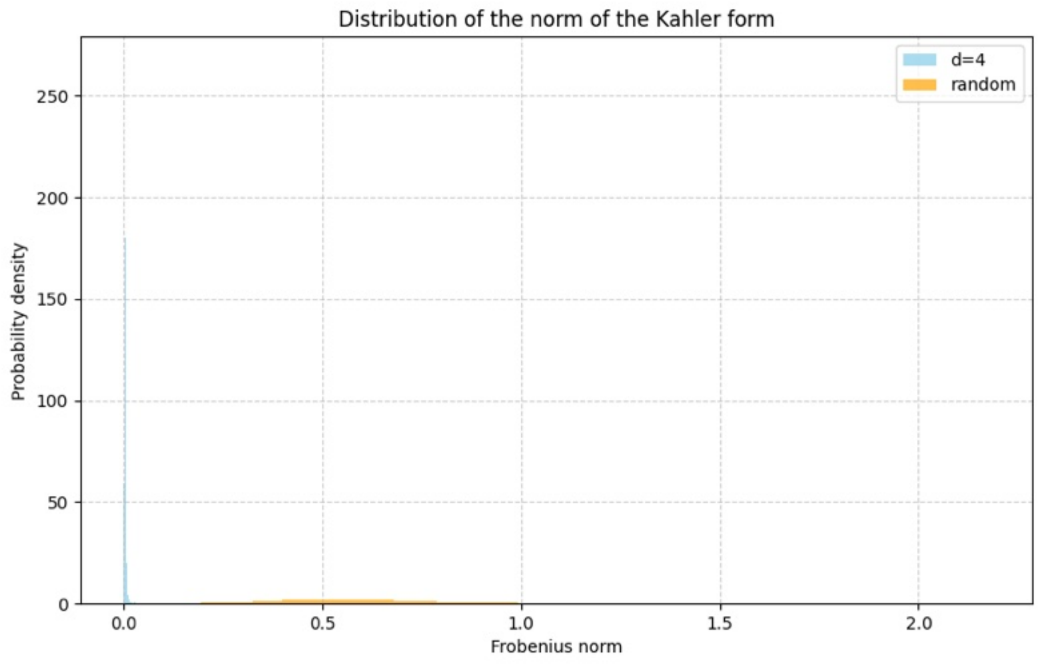
Special Fitness: 0.61

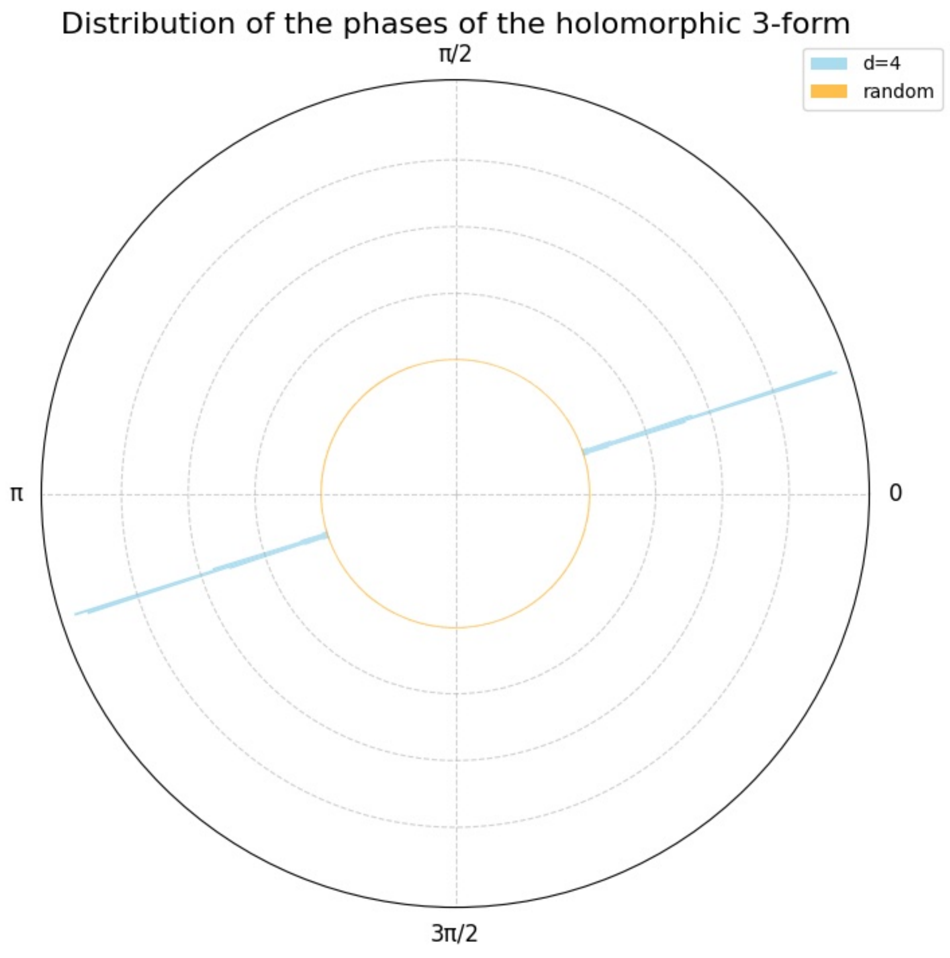




Lagrangian Fitness: 0.96

Special Fitness: 0.78

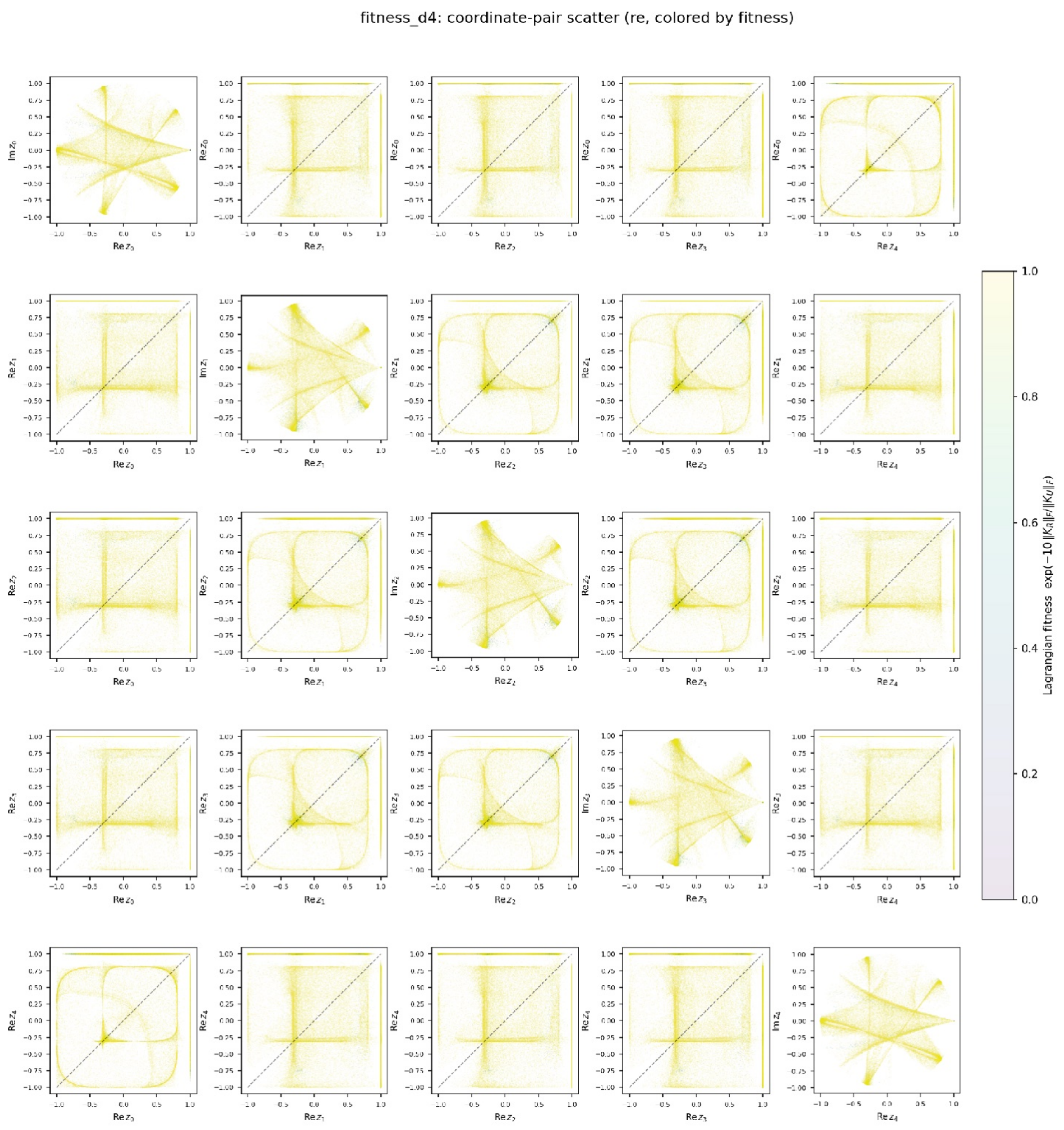


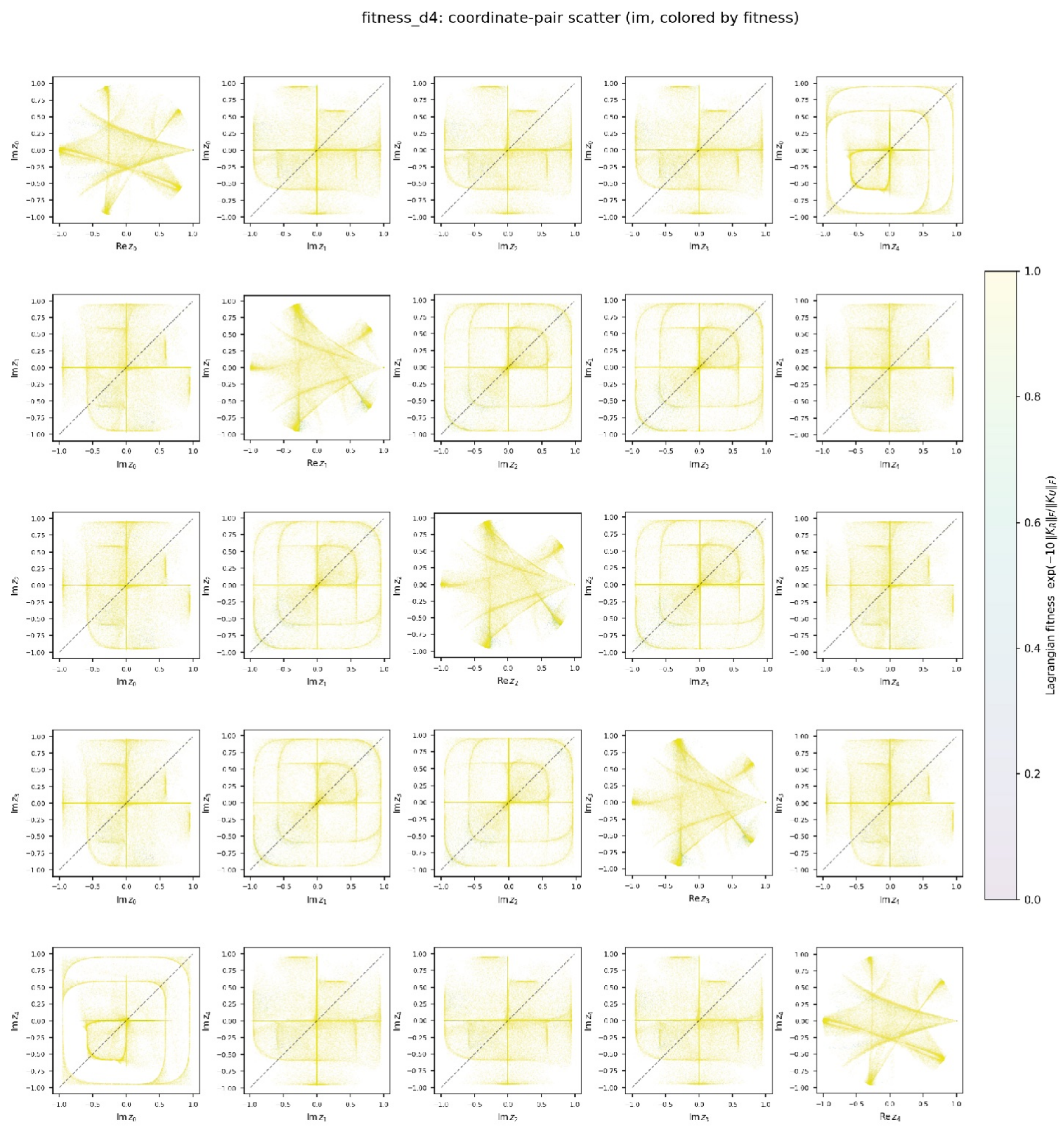


Lagrangian Fitness: 0.97

Special Fitness: 0.96

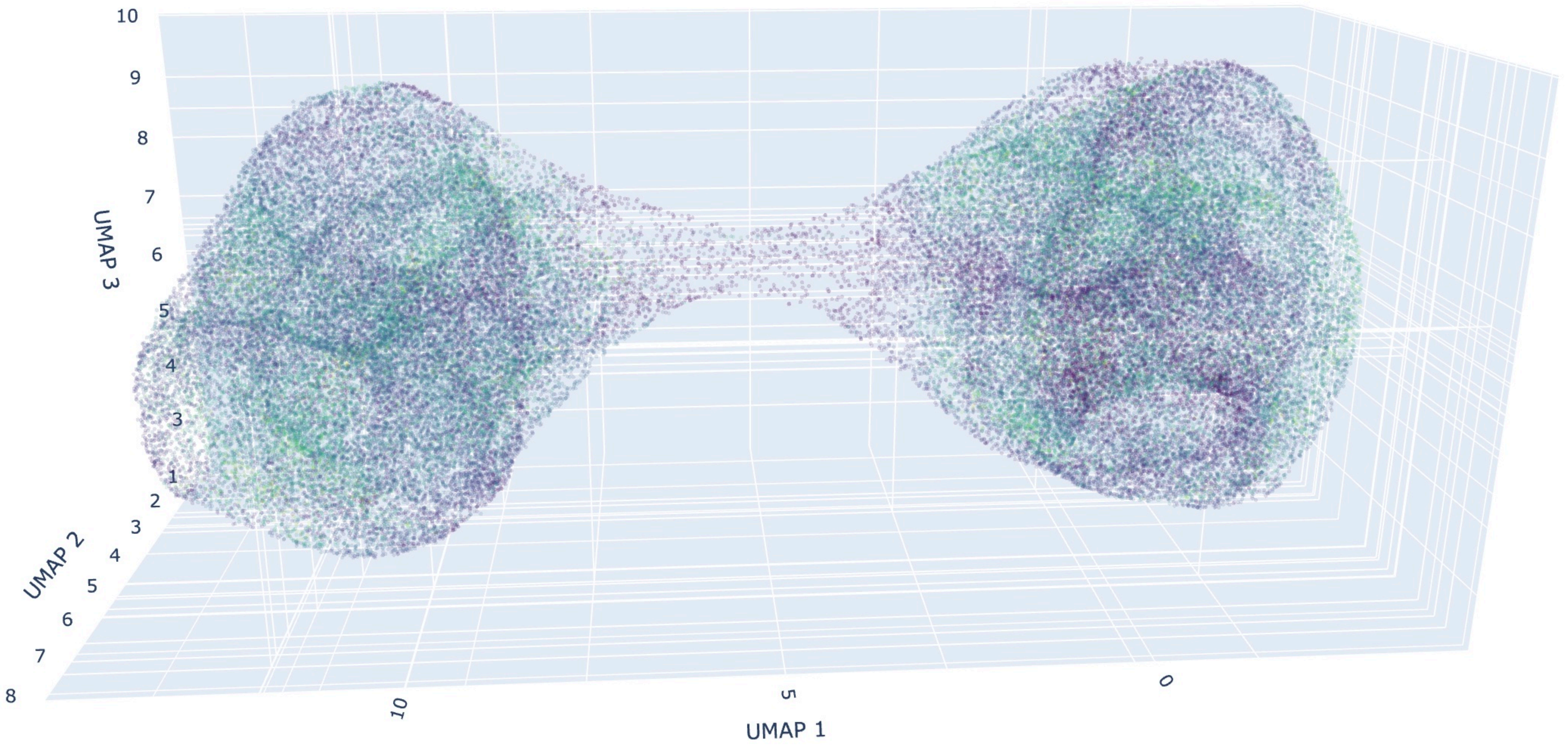
New sLag candidate - coordinate visualization





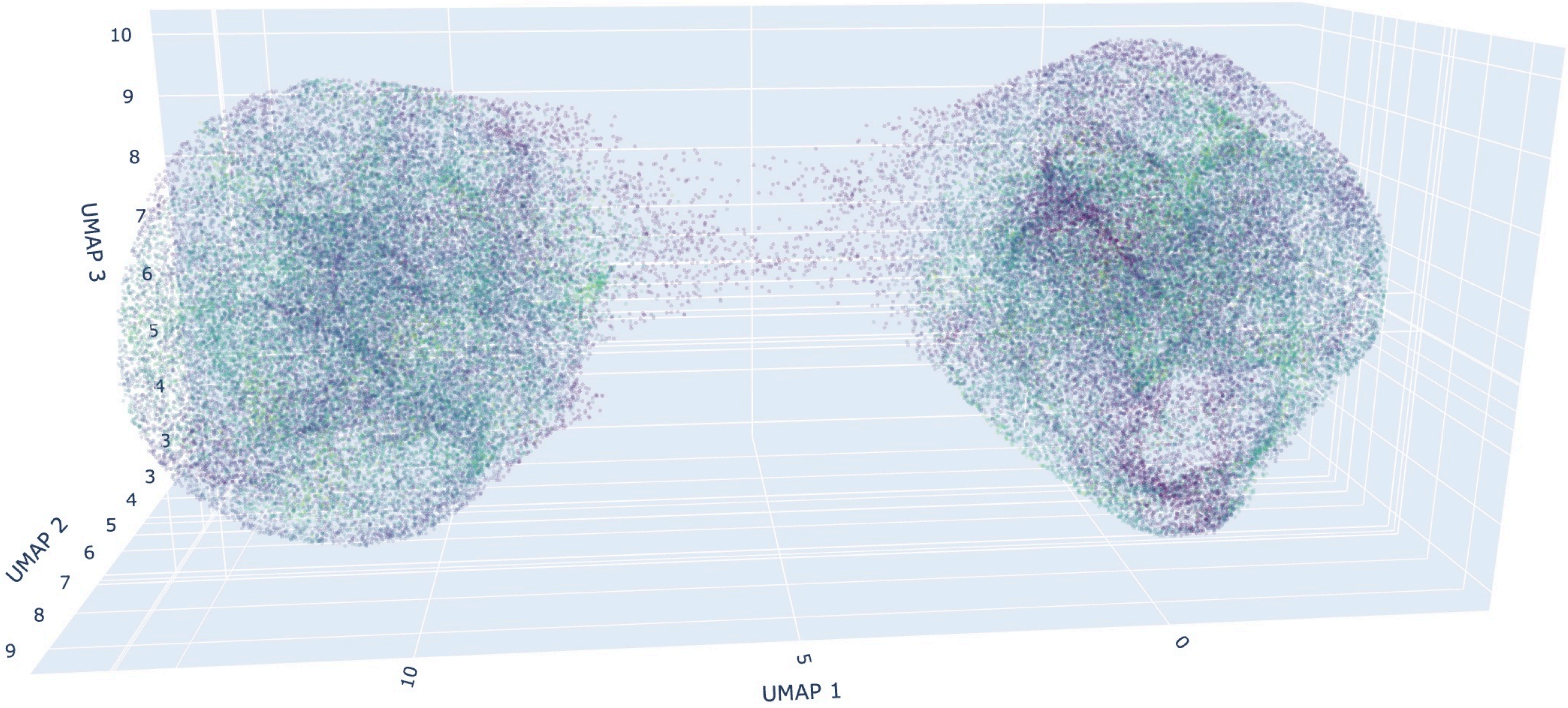
New sLag candidate - UMAP visualization

d=4 from d=1, GD step 0



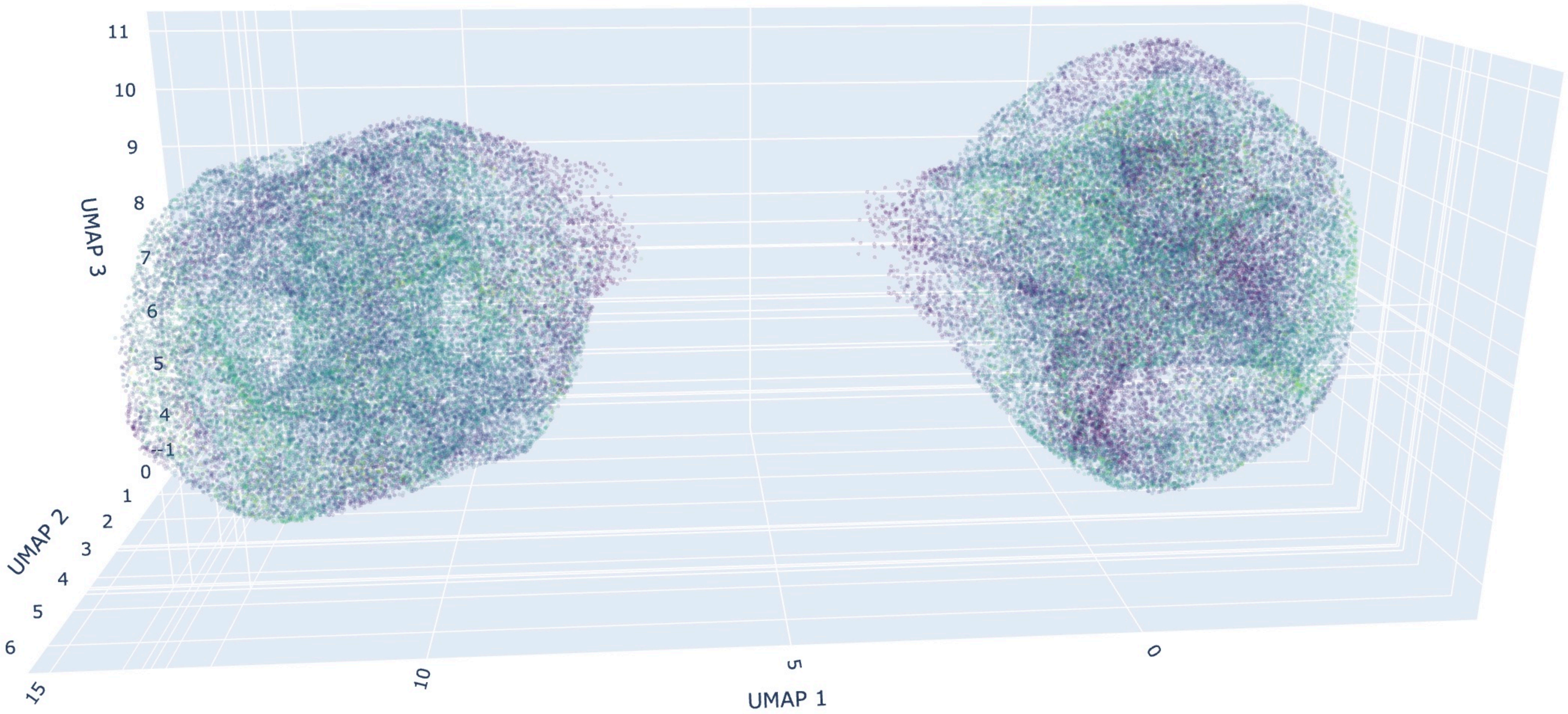
New sLag candidate - UMAP visualization

d=4 from d=1, GD step 1



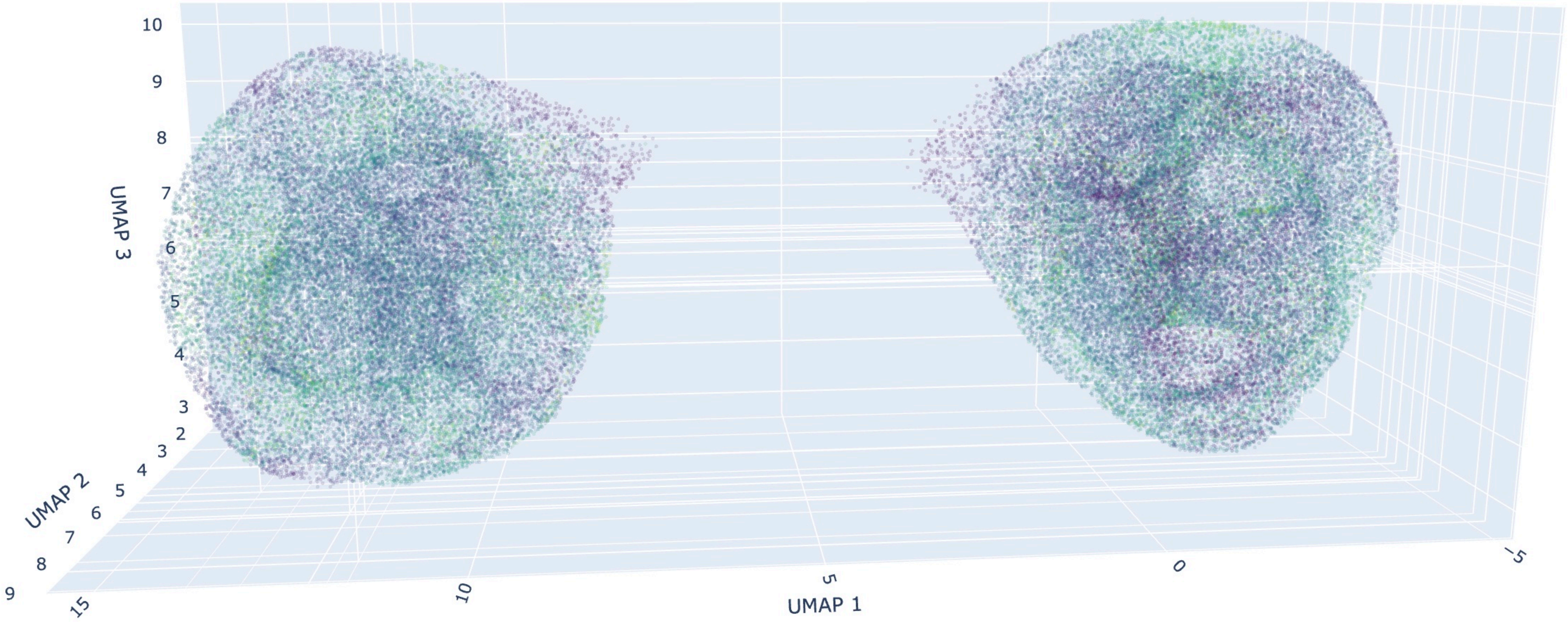
New sLag candidate - UMAP visualization

d=4 from d=1, GD step 2



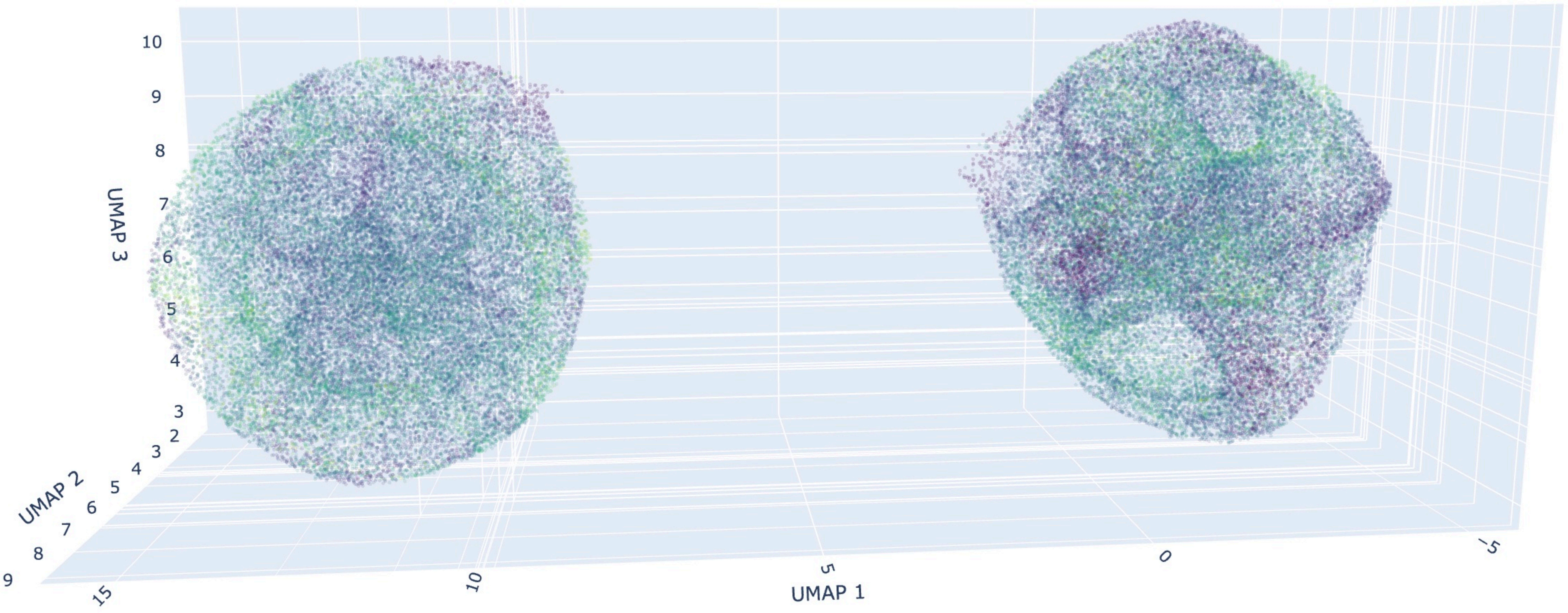
New sLag candidate - UMAP visualization

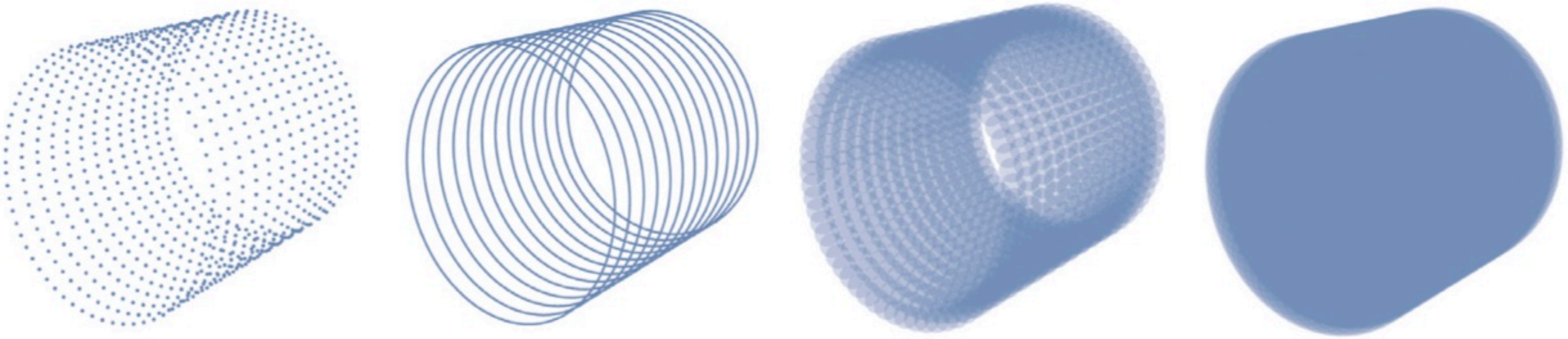
d=4 from d=1, GD step 3

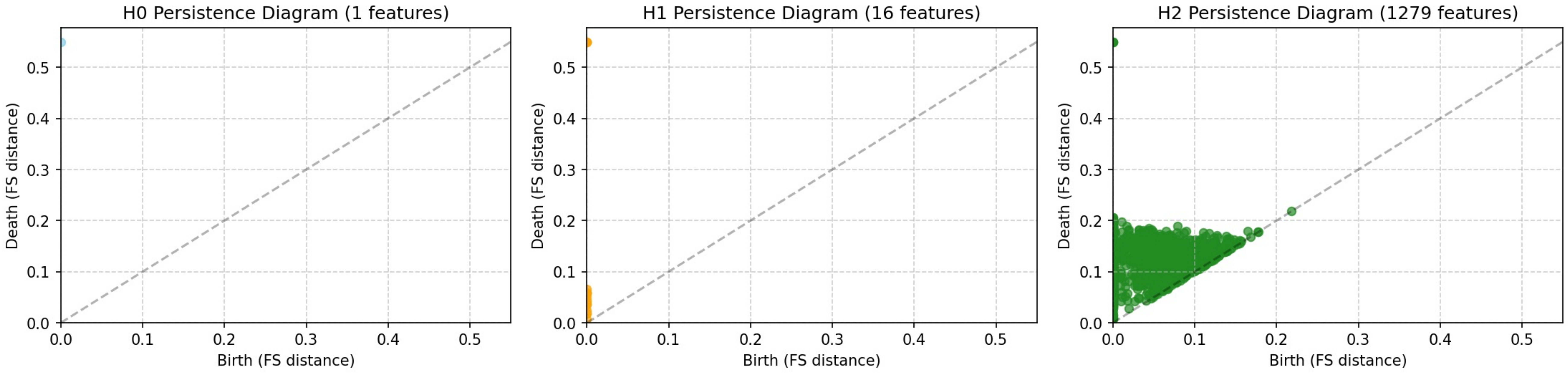


New sLag candidate - UMAP visualization

d=4 from d=1, GD step 4



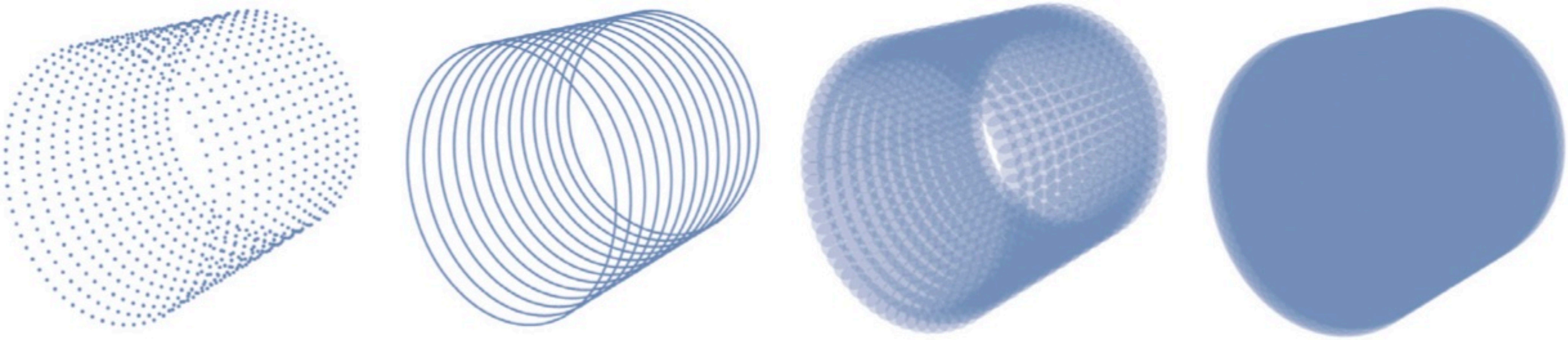




Persistent Homology

Topology of Point Clouds

- In order to find the topology of the sLag, we use Persistent Homology, a tool from Topological Data Analysis [FR `20]



(a) Points. (b) Circles. (c) Cylinder surface. (d) Ball.

Persistent Homology

- Mathematically, we do the following:

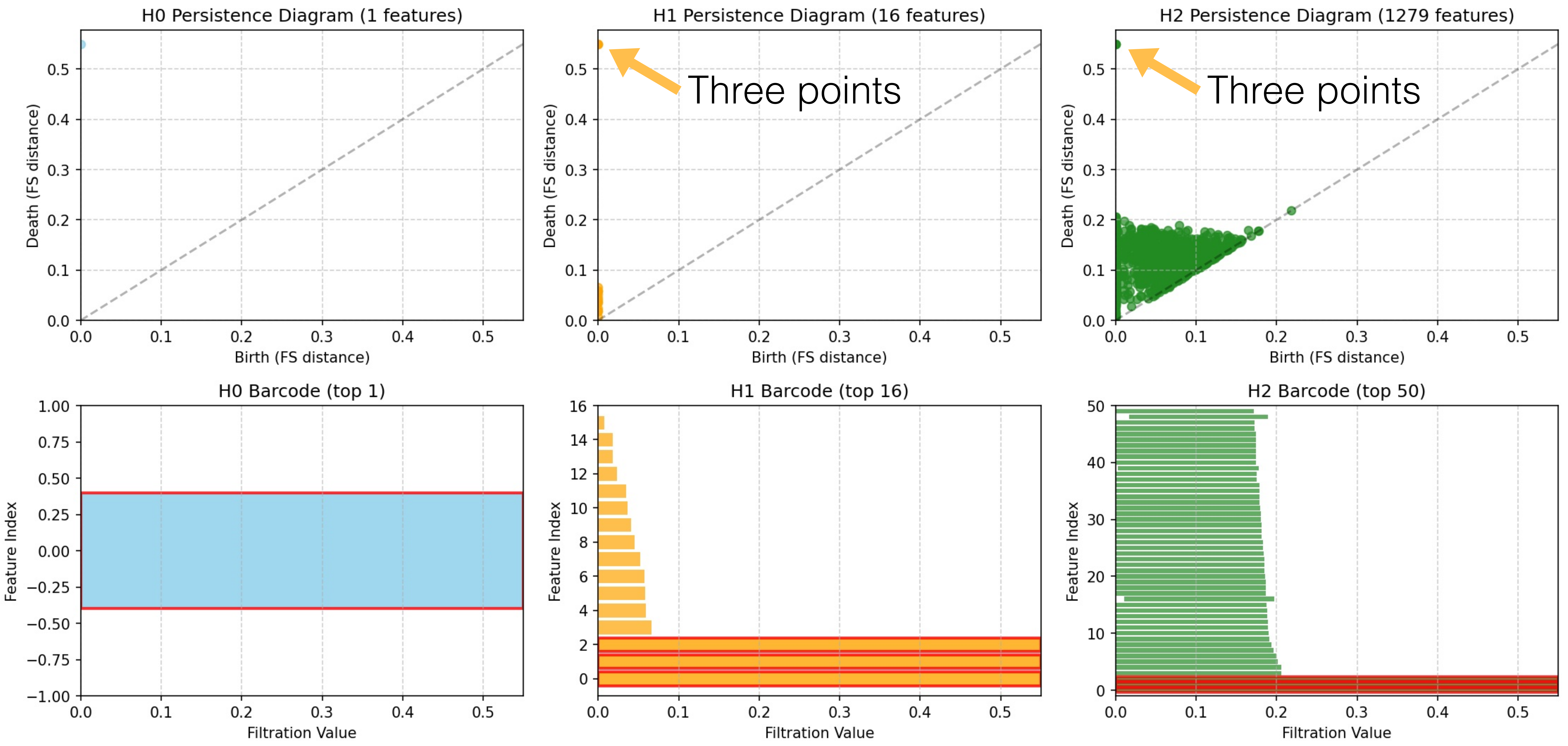
1. Replace points by balls and vary their radii R (filtration by R)
2. Overlapping balls are joined
3. Record the radius at which cycles form (birth)
4. As radii increase, these cycles die (either because they are filled in, or because new, higher-dimensional cycles form such that this cycle becomes a boundary cycle and hence trivial in homology)
5. We assign the topology based on the cycles that persist the longest (over the widest range of R)

[Cirafici `16; Cole, Shiu `18]

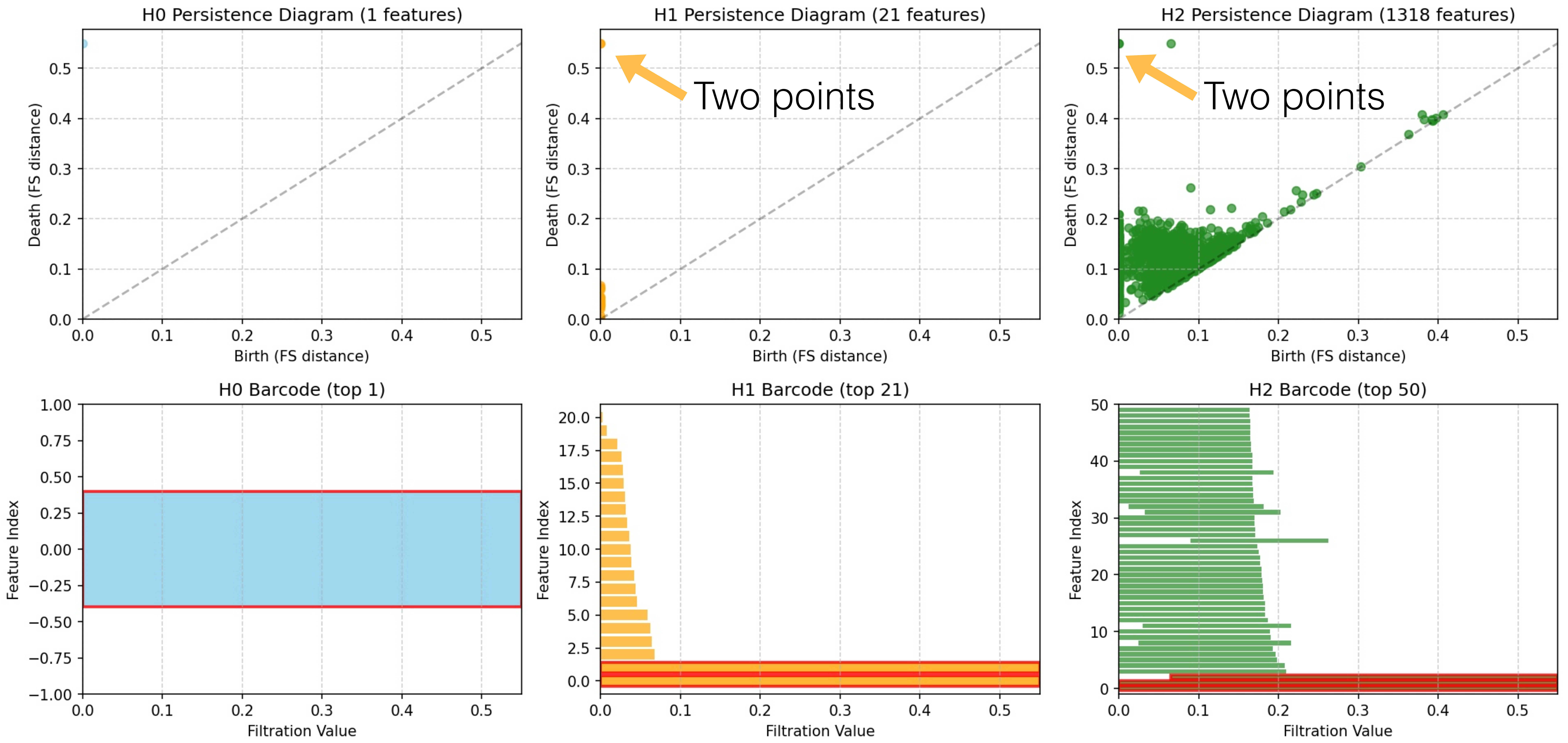
Persistent Homology - Details

- Step 4 (birth/death of cycles) requires explicit construction of (co)-boundary maps in the exact sequence defining the homology
- For computational efficiency, we compute this over $\mathbb{Z}_p$, p prime
- For the distance between points we use the geodesic distance computed w.r.t. the Fubini-Study ambient space metric
- By computing $H_\bullet(\mathcal{L}, \mathbb{Z}_p)$ for different p, we can recover integer homology and the torsion part
- This is automated in the package ripser
- Side remark: This can also be applied to find torsion cycles in CY threefolds more generally, which is very difficult to do numerically [FR, to appear]

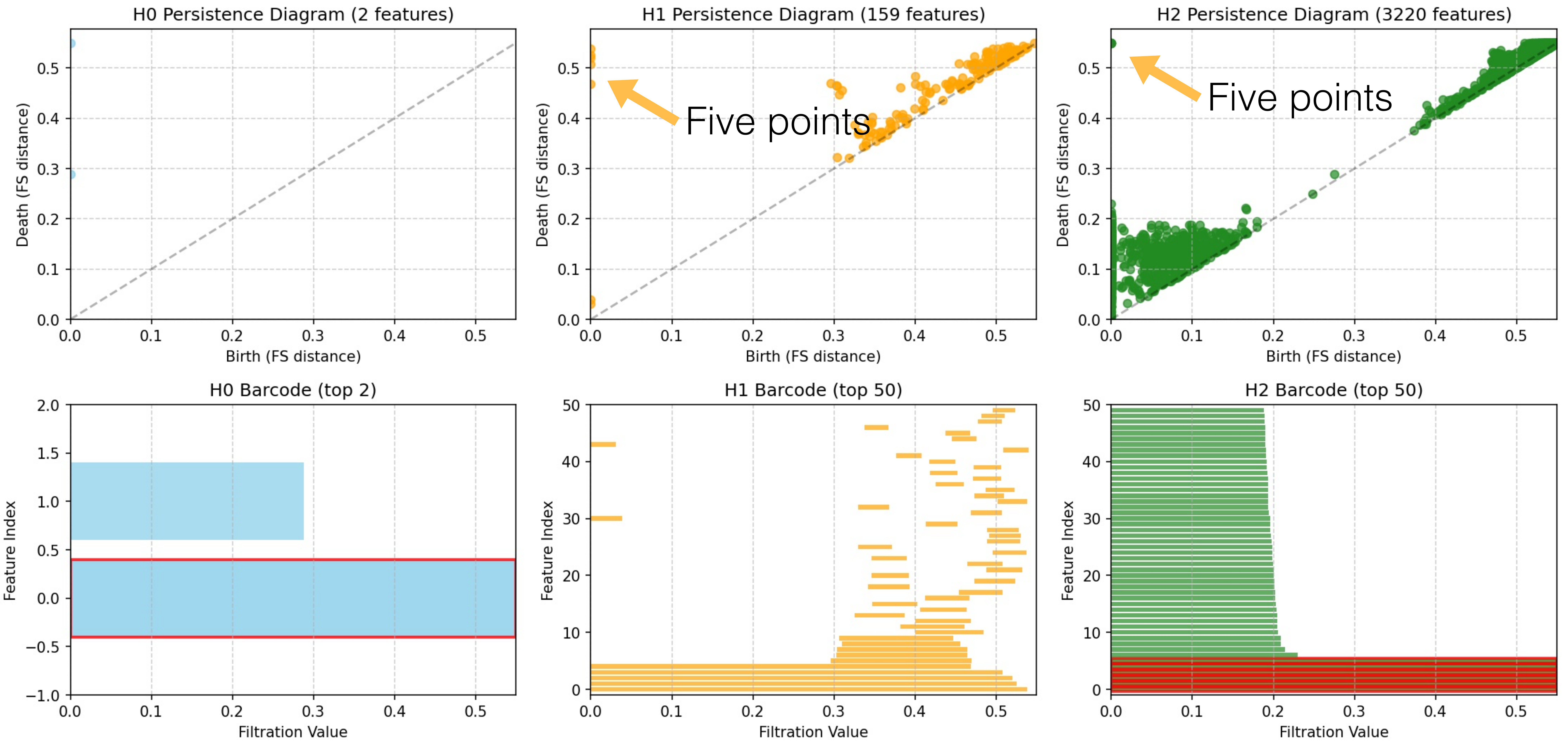
Persistent Homology - Cluster 1

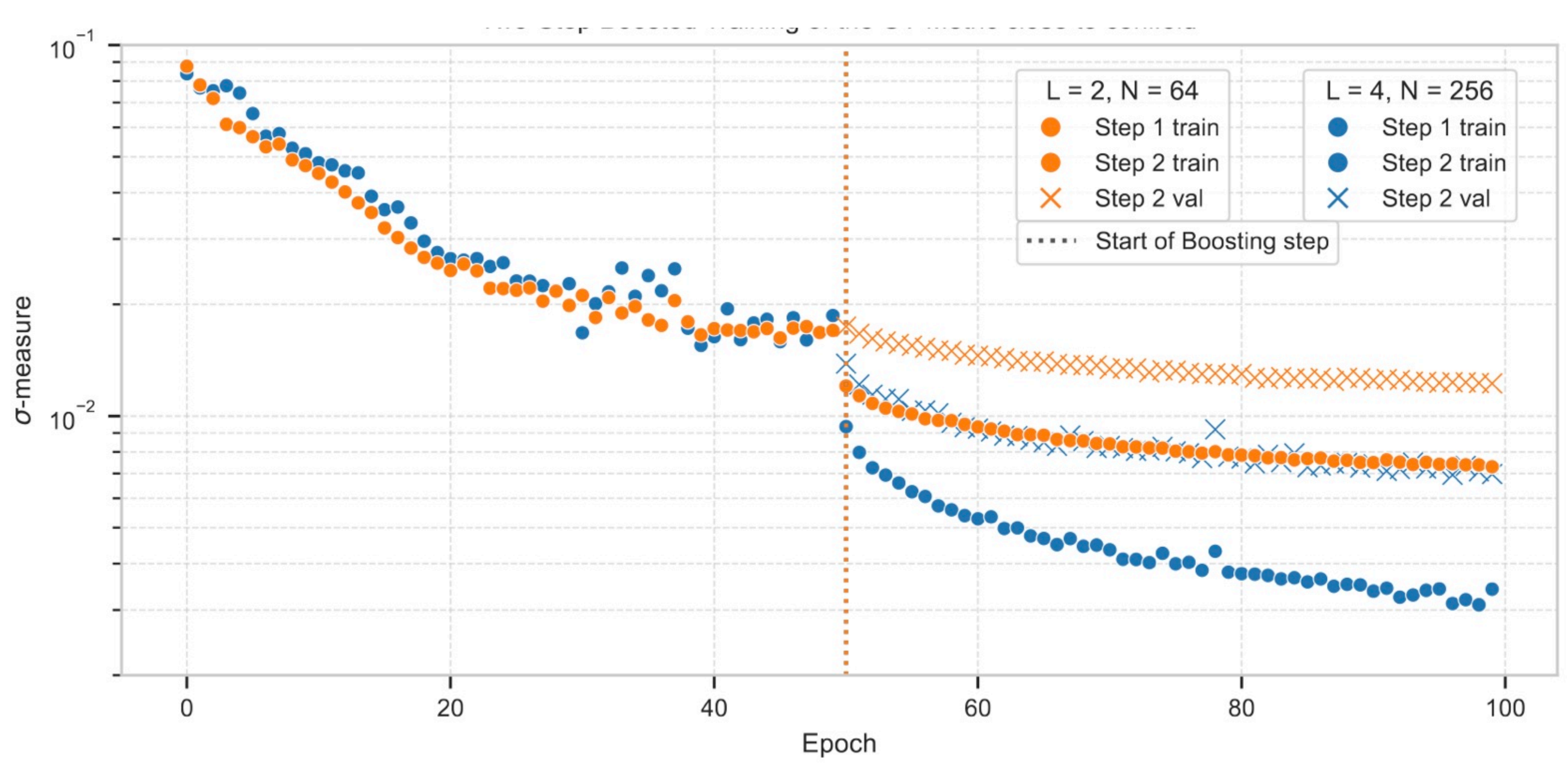


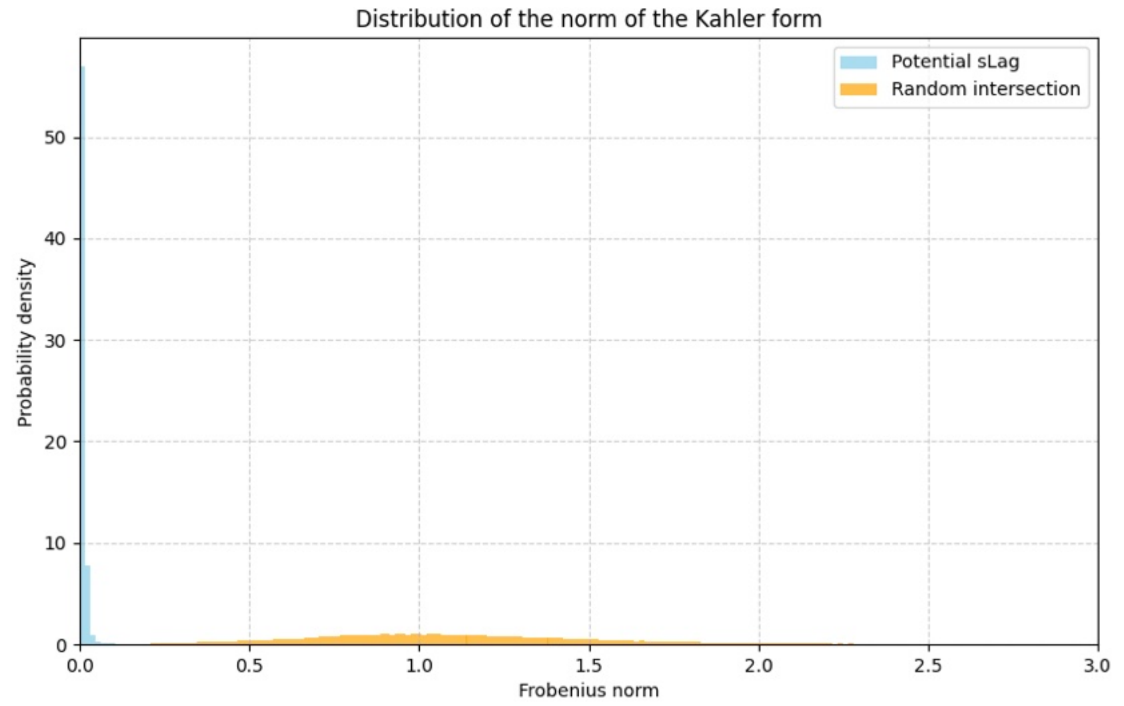
Persistent Homology - Cluster 2

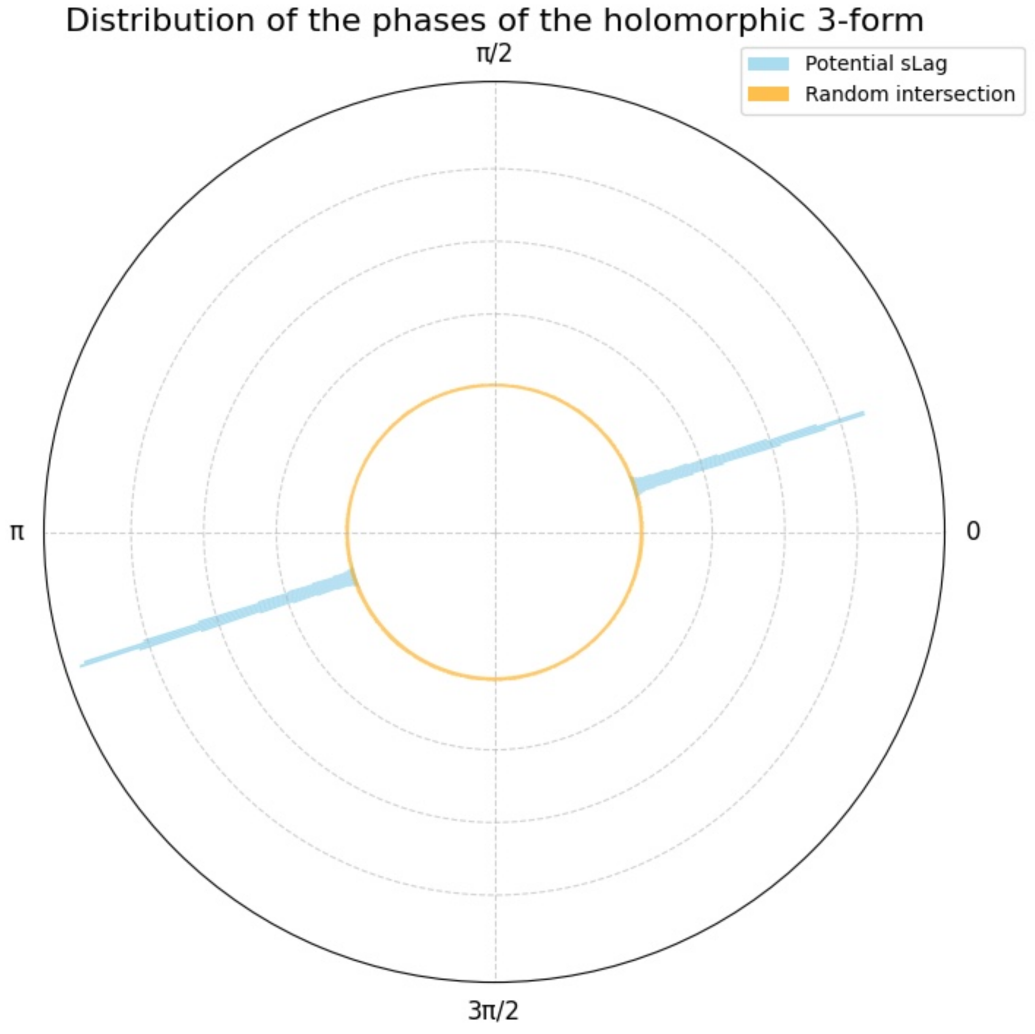


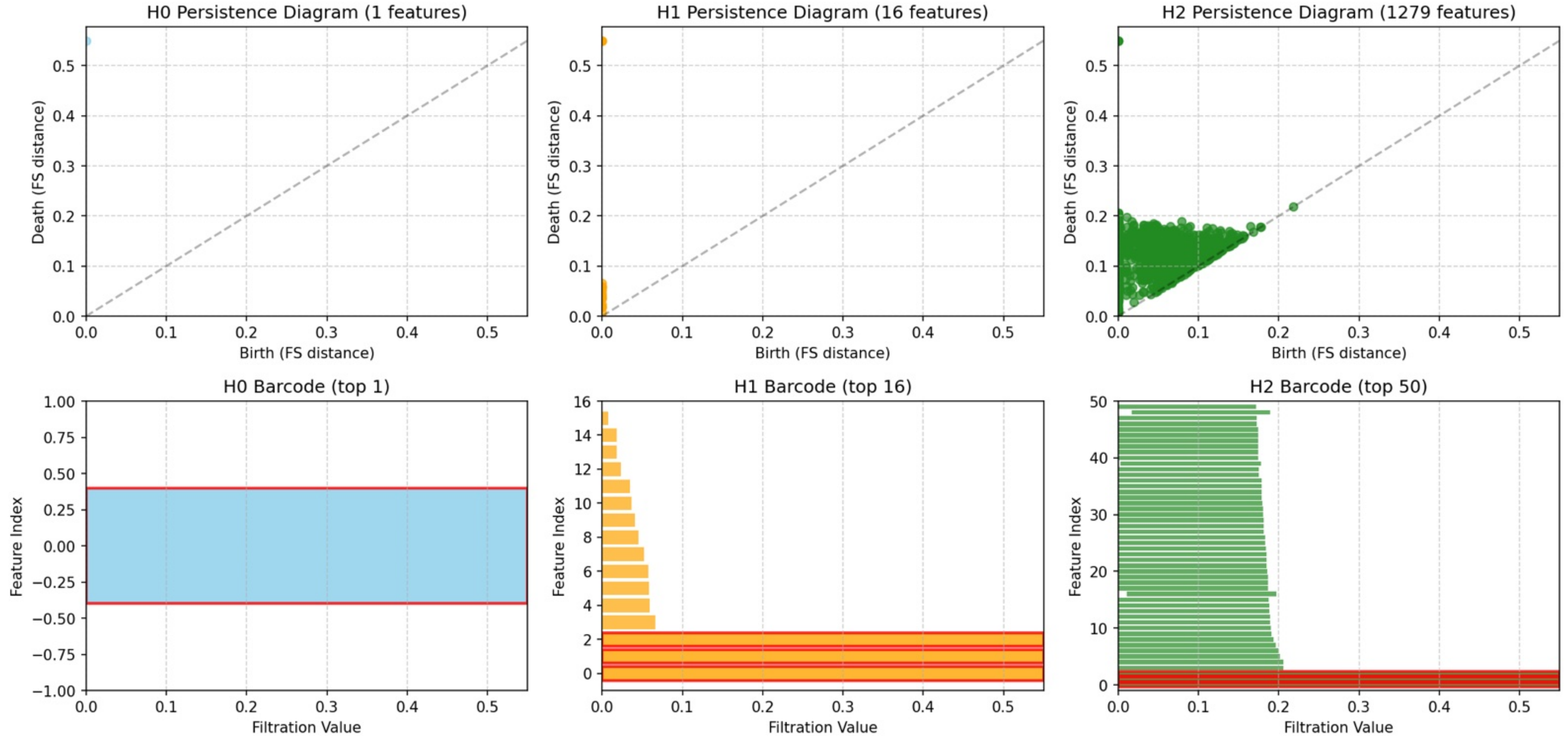
Persistent Homology - Both











Conclusions

Conclusions

- Despite their importance, essentially no examples for sLags are known
- Use Data Science Techniques to find new candidate sLags on the Quintic
  - Point sampling techniques to find points
  - PINNs to find the NN metric
  - Genetic algorithms to find equations that define sLags (no diff needed)
  - TDA to find the topology of the sLag
- We reproduced known examples (RP^3 and T^3), and identified a new potential example

Conclusions

- Despite their importance, essentially no examples for sLags are known
- Use Data Science Techniques to find new candidate sLags on the Quintic
  - Point sampling techniques to find points
  - PINNs to find the NN metric
  - Genetic algorithms to find equations that define sLags (n
  - TDA to find the topology of the sLag
- We reproduced known examples (RP^3 and T ew potential example

Thank you!