Neural Network Approximations for Calabi-Yau Metrics

Damián Mayorga Peña
WITS

BIRS-CMO, Oaxaca 11.11.2021

The Upshot

In this talk I discuss one of the recent **Machine Learning approaches to obtain numerical approximations to Ricci flat Calabi-Yau metrics.** Instead of approximating the Kähler potential, we approximate the metric directly by an array neural networks. We apply this approach to the quartic K3, the Dwork quintics and Tian-Yau manifold.

Collaborators

Vishnu Jejjala

WITS

Challenger Mishra

Cambridge

Arghya Chattopadhyay

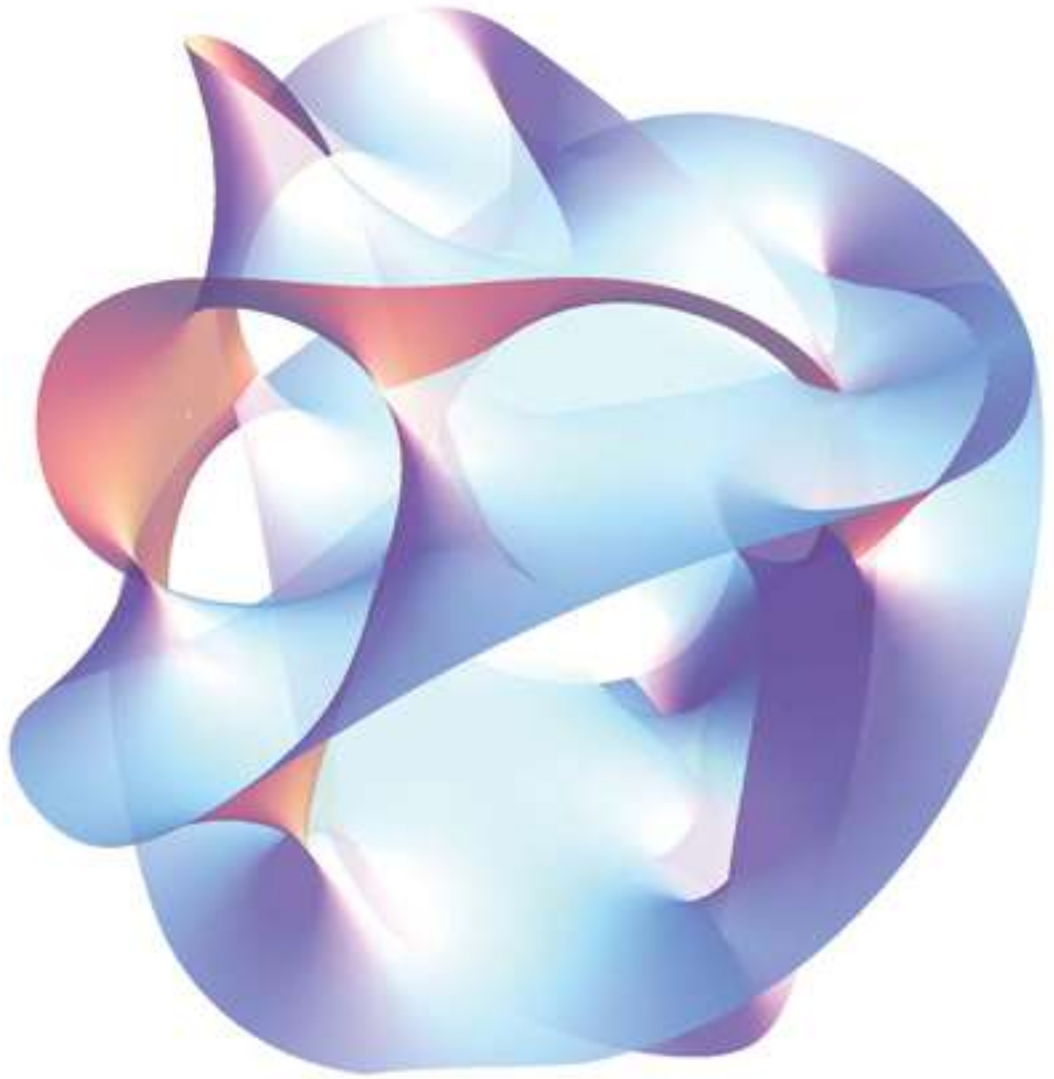
WITS

Based on **arXiv: 2012.15821**, **arXiv: 21mm.nnnn**

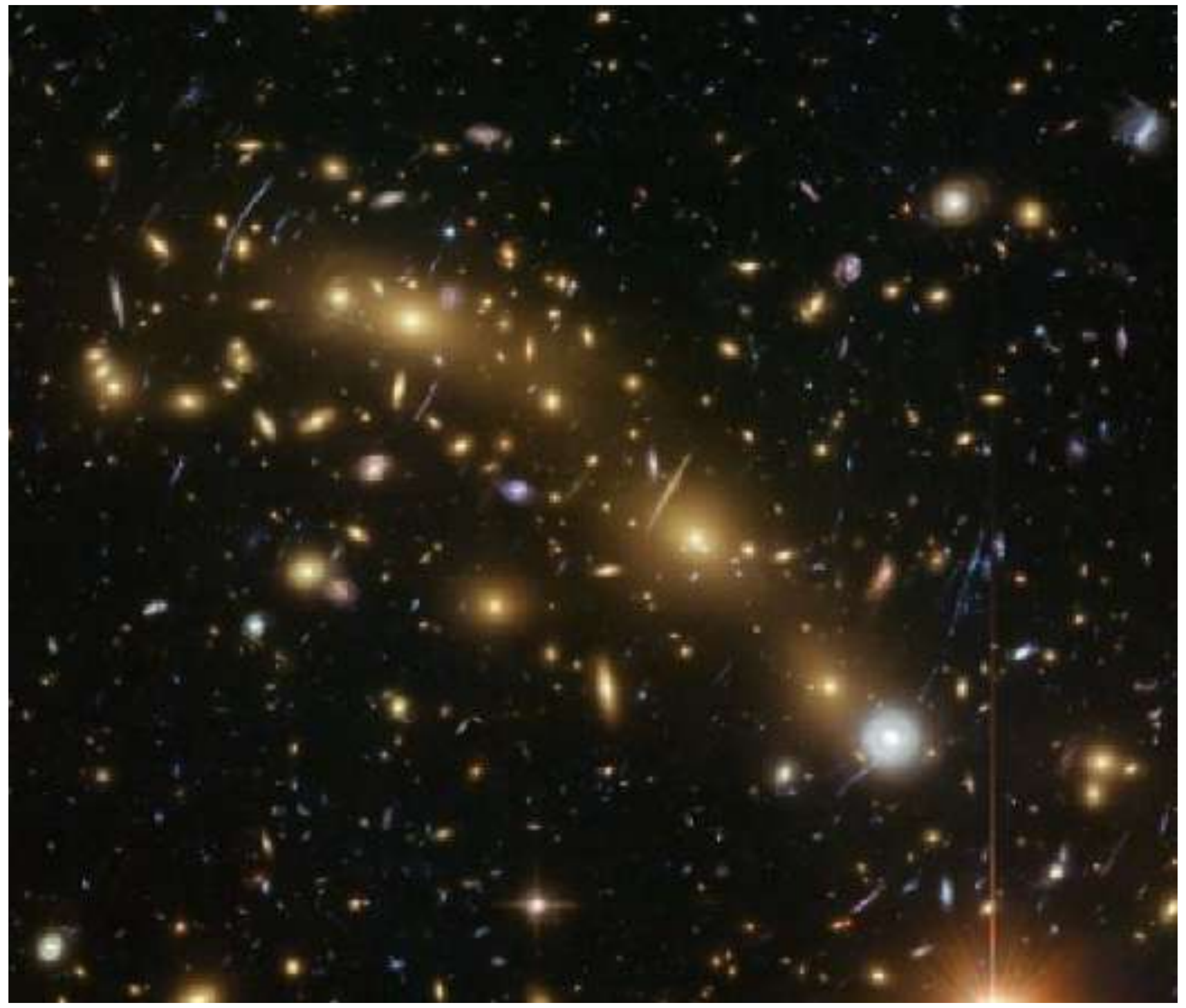
Calabi-Yau manifolds

In string compactifications one is interested in obtaining low energy effective field theories with some remnant supersymmetry

Total Space-time (10D)



Calabi-Yau manifold



Our usual 4D space-time

Calabi-Yau manifolds

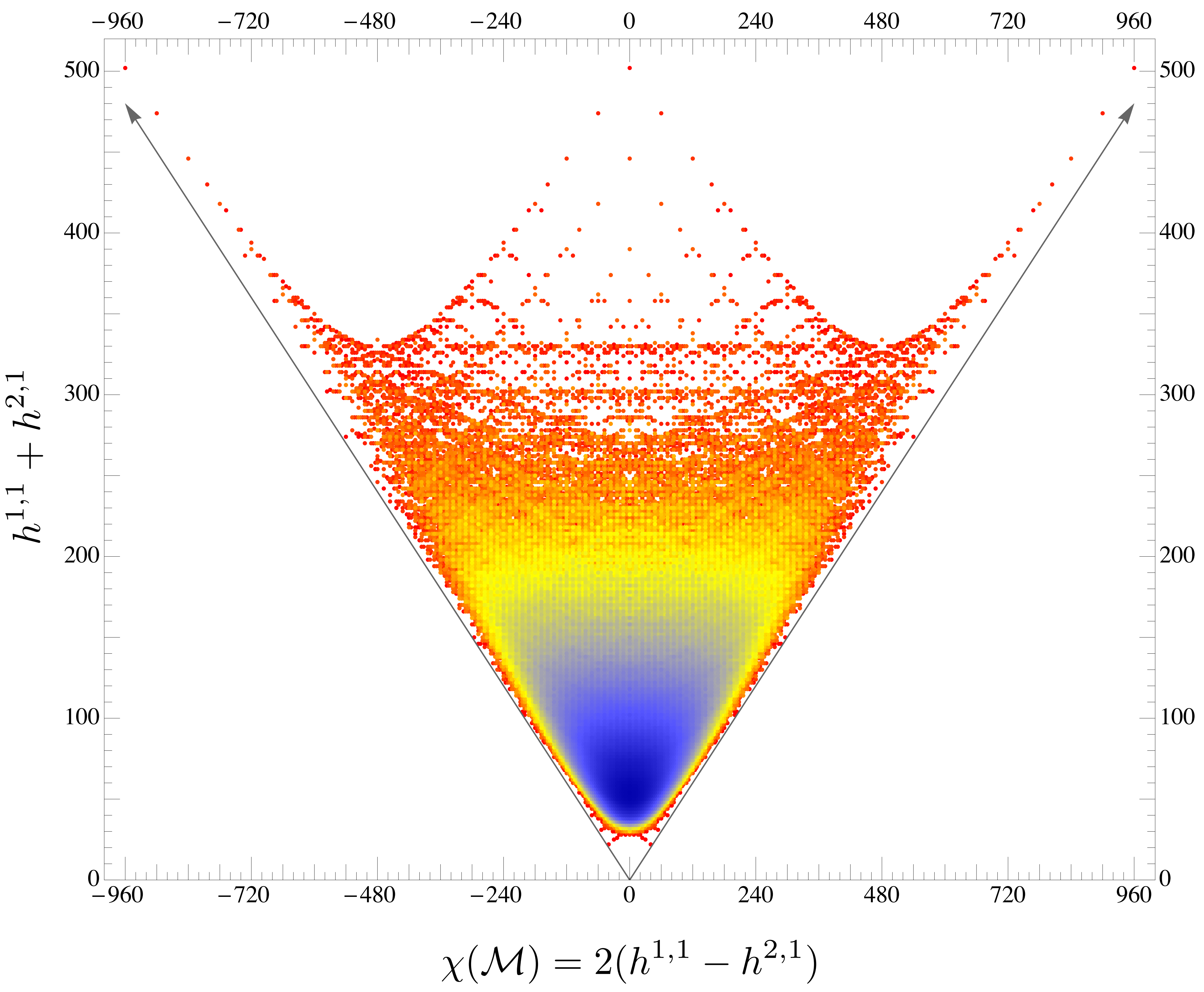
The total parameter space of a CY threefold:

- $h^{1,1}(\mathcal{M})$ dimension of the Kähler moduli space.
- $h^{2,1}(\mathcal{M})$ dimension of the Complex structure moduli space.

- Calabi-Yau threefolds come in mirror pairs $(\mathcal{M}, \mathcal{M}')$, satisfying

$$h^{1,1}(\mathcal{M}) = h^{2,1}(\mathcal{M}') \qquad h^{2,1}(\mathcal{M}) = h^{1,1}(\mathcal{M}')$$

In simple terms, complex structure and Kähler structure get interchanged. This is the basic idea behind Mirror Symmetry.



Calabi-Yau awesomeness!

@mateofarinella

Calabi-Yau manifolds

A (compact) Calabi-Yau manifold of complex dimension n is a Kähler manifold $(\mathcal{M}, g, J)$ satisfying any of the following equivalent properties:

- The first Chern class of $\mathcal{M}$ is zero.
- $\mathcal{M}$ has a Kähler metric with vanishing Ricci curvature.
- $\mathcal{M}$ has a nowhere vanishing holomorphic n-form.
- $\mathcal{M}$ has a Kähler metric with local holonomy $SU(n)$

Calabi-Yau Manifolds

Some Examples: Calabi-Yaus constructed hypersufaces in projective spaces

-K3 (Fermat Quartic): $z_1^4 + z_2^4 + z_3^4 + z_4^4 = 0 \subset \mathbb{P}^3$,

-Fermat Quintic $z_1^5 + z_2^5 + z_3^5 + z_4^5 + z_5^5 = 0 \subset \mathbb{P}^4$

-Dwork $z_1^5 + z_2^5 + z_3^5 + z_4^5 + z_5^5 - 5\psi z_1 z_2 z_3 z_4 z_5 = 0 \subset \mathbb{P}^4 \ , \quad \psi^5 \neq 1$

-Tian Yau

$$\left[\begin{array}{c||ccc} \mathbb{P}^3 & 3 & 0 & 1 \\ \mathbb{P}^3 & 0 & 3 & 1 \end{array}\right]^{14,\ 23}_{\chi=-18} \iff \left\{\begin{array}{ccc} \alpha^{ijk} z_i z_j z_k & = & 0\ , \\ \beta^{ijk} w_i w_j w_k & = & 0\ , \\ \gamma^{ij} z_i w_j & = & 0\ . \end{array}\right.$$

-Schoen

$$\left[\begin{array}{c||cc} \mathbb{P}^1 & 1 & 1 \\ \mathbb{P}^2 & 3 & 0 \\ \mathbb{P}^2 & 0 & 3 \end{array}\right]^{19,19}_{\chi=0} \simeq \left[\begin{array}{c||cccccc} \mathbb{P}^1 & 0 & 0 & 0 & 0 & 1 & 1 \\ \mathbb{P}^2 & 1 & 1 & 0 & 0 & 1 & 0 \\ \mathbb{P}^2 & 1 & 1 & 0 & 0 & 1 & 0 \\ \mathbb{P}^2 & 0 & 0 & 1 & 1 & 0 & 1 \\ \mathbb{P}^2 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}\right]^{19,19}_{\chi=0} .$$

Machine Learning Calabi-Yau Metrics

Donaldson's Algorithm

A valid Kähler potential can be obtained generalizing the Fubini-Study metric to polynomials of a higher degree

$$K^{(k)}(z,\bar{z}) = \frac{1}{k\pi}\log(h^{\alpha\bar{\beta}}s_\alpha \bar{s}_{\bar{\beta}})$$

where the s_α form a basis for holomorphic polynomials over $\mathcal{M}$ up to degree k. The task is to find a Hermitean matrix $h^{\alpha\bar{\beta}}$ for every k, that gives the best approximation to the Ricci flat metric.

Take N_k to be the dimension of $\{s_\alpha\}$ and define

$$H_{\alpha\bar{\beta}} = \frac{N_k}{\mathrm{Vol}_\Omega}\int_{\mathcal{M}} d\mathrm{Vol}_\Omega \left(\frac{s_\alpha \bar{s}_{\bar{\beta}}}{h^{\alpha\bar{\beta}}s_\alpha \bar{s}_{\bar{\beta}}}\right) \qquad d\mathrm{Vol}_\Omega = \Omega \wedge \bar{\Omega}$$

Now take $h^{\alpha\bar{\beta}} = (H_{\alpha\bar{\beta}})^{-1}$ and proceed iteratively until the metric stabilizes. In this manner one obtains the "balanced metric" at degree k.

As the polynomial degree increases the metric $g_{a\bar{b}}$ obtained from the Kähler potential with the balanced metric approaches the desired Ricci flat metric.

Machine Learning Calabi-Yau Metrics

To date we do not have an analytic expression for a Ricci flat Calabi Yau metric.
With the exception of K3. Kachru, Tripathy, Zimmet'20'21

-The metric can be accessed numerically.

Headrick, Wiseman'05 Anderson, Braun Karp, Ovrut'10
Headrick, Nassar'13, Cui, Gray'19

-More recently, Machine Learning techniques have been used for this endeavour. Ashmore, Ovrut, He'19 Anderson, Gerdes, Gray, Krippendorf, Raghuram, Rühle'20
Douglas, Lakshminarasimhan, Qui'20, Jejjala, DM, Mishra'20, Ashmore, Rühle'21
Ashmore, Deen, He, Ovrut'21, Larfors, Lukas, Rühle, Schneider'21

Being a Kähler manifold, the Hermitian metric g can be derived from a Kähler potential

$$g_{a\bar{b}} = \partial_a \partial_{\bar{b}} K(z^a, \bar{z}^{\bar{b}})$$

The Kähler form is given by:

$$J = \frac{\mathrm{i}}{2} g_{a\bar{b}}\, dz^a \wedge \bar{z}^{\bar{b}}$$

The Ricci tensor is obtained as

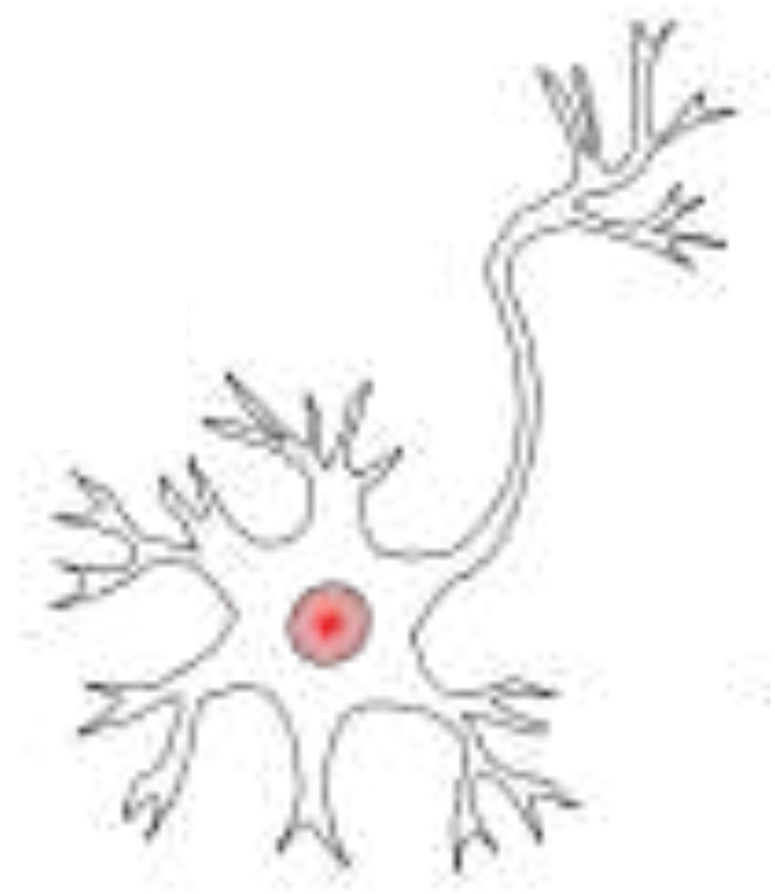
$$R_{a\bar{b}} = \partial_a \partial_{\bar{b}} \log \det g$$

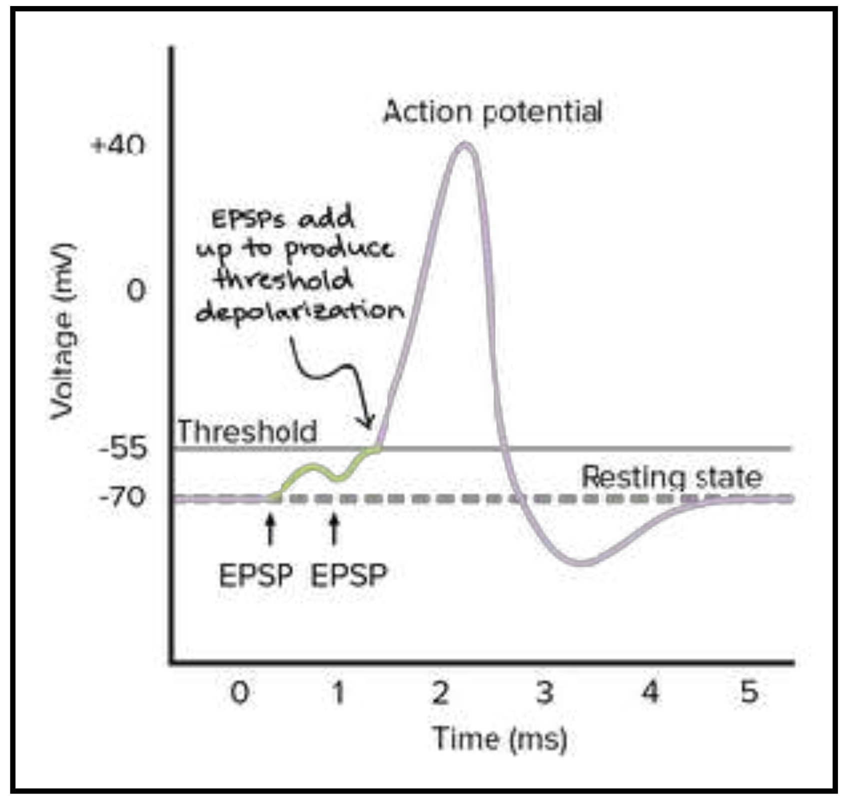
Simplest case: The Fubini-Study metric in the ambient space

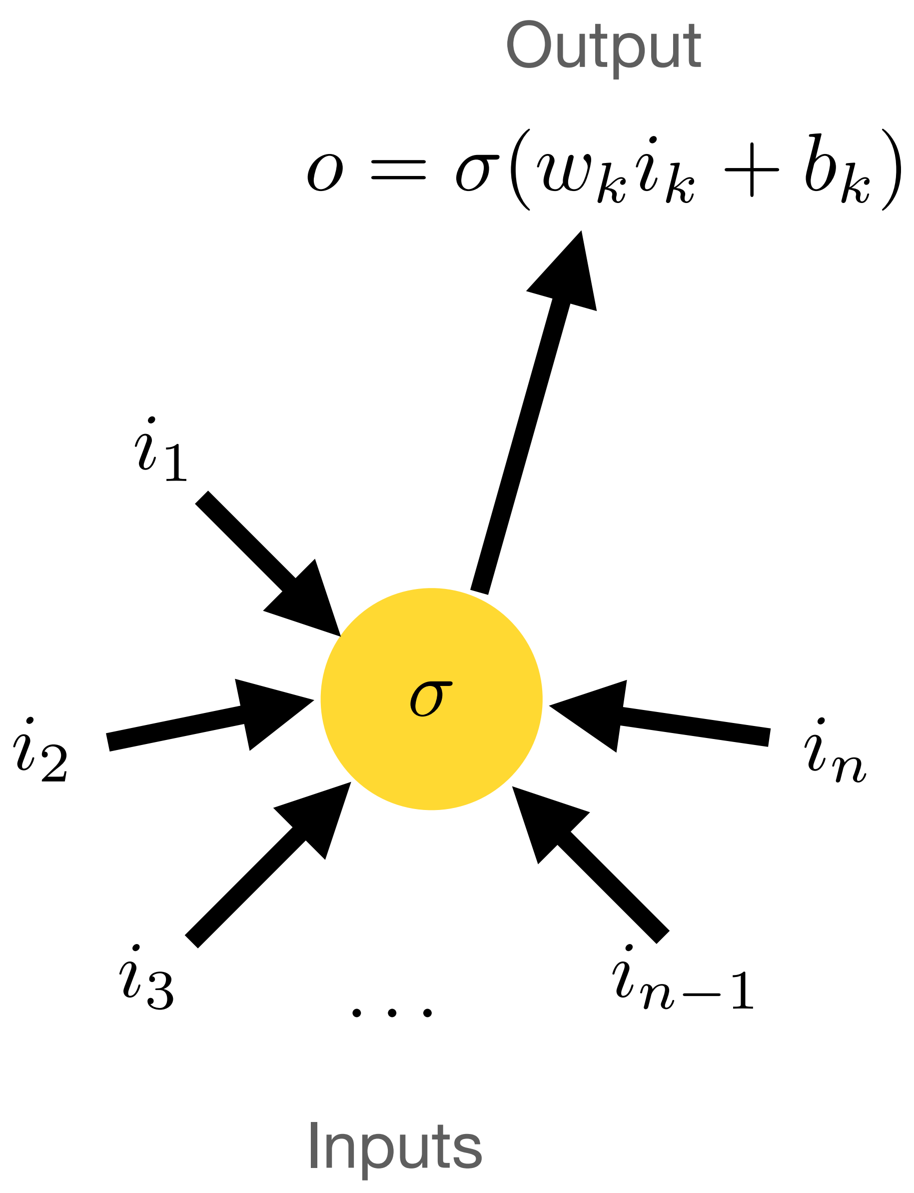
$$K_{FS} = \frac{1}{\pi} \log(z \cdot \bar{z})$$

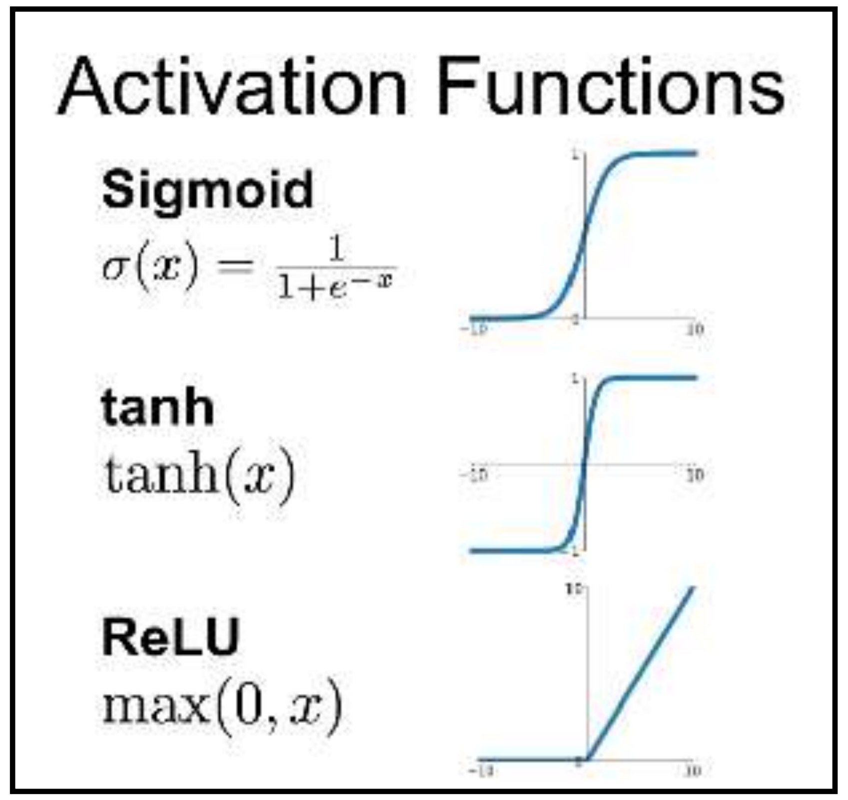
restricted to the hypersurface (CICY).

Machine Learning Calabi-Yau metrics

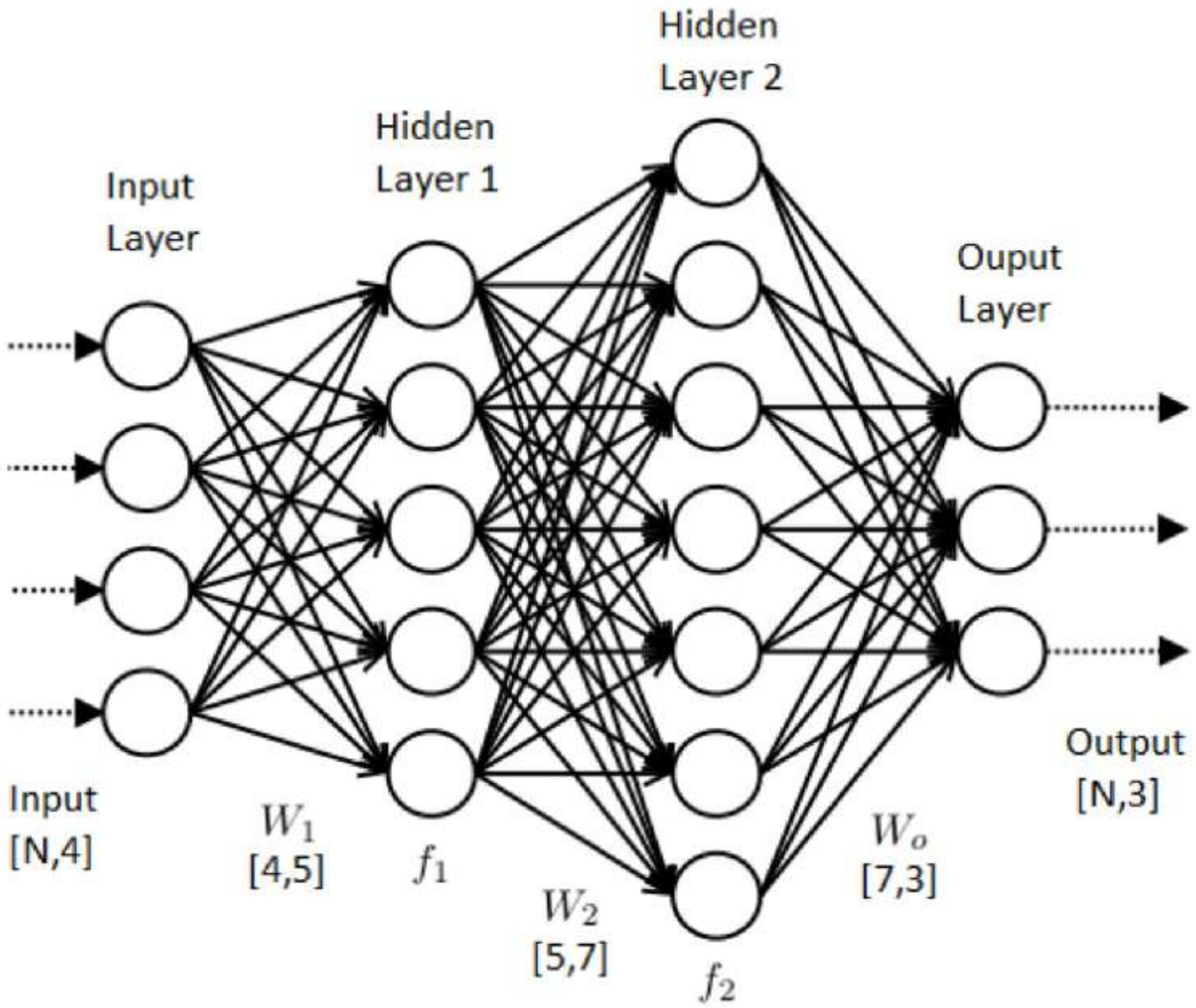








A paradigm in ML are **Artificial Neural Networks**: arrays of artificial neurons that emulate the human brain.



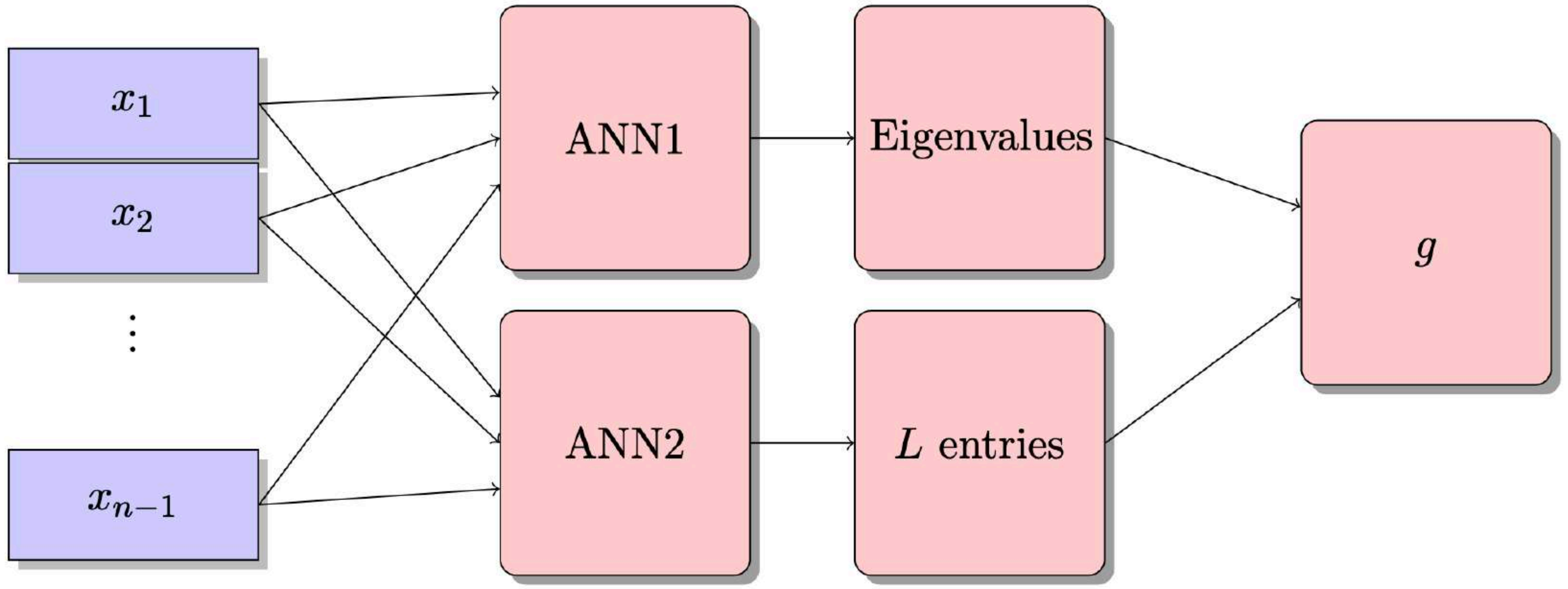
Machine Learning Calabi-Yau Metrics

A Hermitian metric can be written in the LDL decomposition.

$$g = L\,D\,L^{\dagger}$$

With L a lower triangular matrix with 1s in the diagonal and $D = \mathrm{diag}(e_1,\, e_2,\, \ldots, e_n) \quad e_i > 0$

Approximate the metric as a combination of neural networks



Since the eigenvalues have to be positive. we take exponentiate the outputs of ANN1 $e_i = \mathrm{Exp}(o_i^{(1)})$ and use the outputs of ANN2 to construct L in K3 for example,

$$L = \begin{pmatrix} 1 & 0 \\ o_1^{(2)} + \mathrm{i}o_2^{(2)} & 1 \end{pmatrix}$$

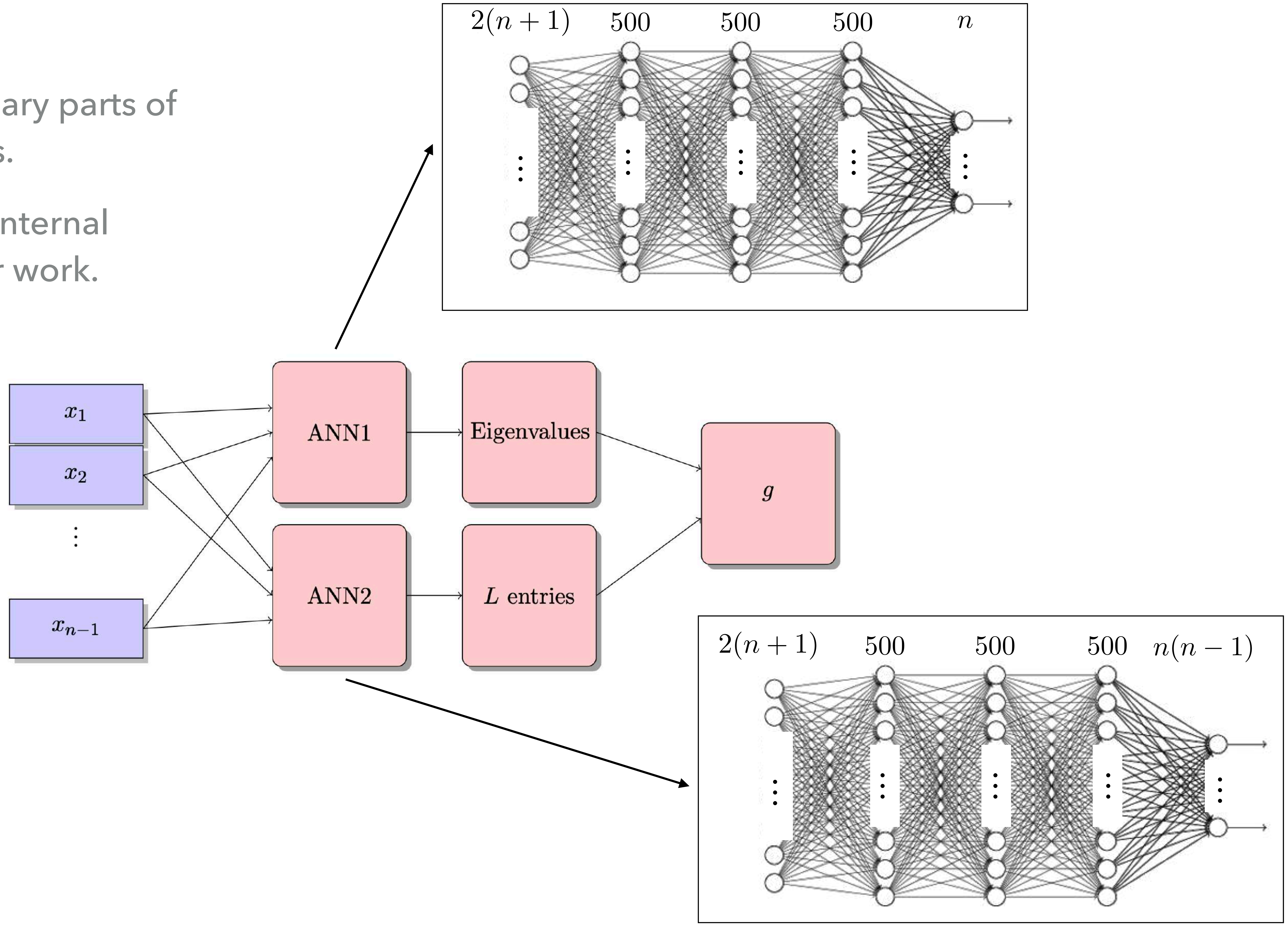
Machine Learning Calabi-Yau Metrics

We have taken real and imaginary parts of the affine coordinates as inputs.

The number of neurons in the internal layers was kept throughout our work.

In all of the experiments three activation functions were used: Logistic Sigmoid, ReLU and Tanh.

The data was prepared in Mathematica and the neural networks were implemented in PyTorch.



Machine Learning Calabi-Yau Metrics

The loss function is constructed for the full network ensemble, and it is minimized for g approaching the Flat metric. It is constructed based on three properties

-Local Flatness

$$\sigma = \frac{1}{\mathrm{Vol}_\Omega} \int_{\mathcal{M}} d\mathrm{Vol}_\Omega \left| 1 - \frac{\mathrm{Vol}_\Omega}{\mathrm{Vol}_J} \cdot \frac{J^n}{\Omega \wedge \bar{\Omega}} \right| .$$

-Kählerity

$$\kappa = \frac{\mathrm{Vol}_J^{1/n}}{\mathrm{Vol}_\Omega} \int_{\mathcal{M}} d\mathrm{Vol}_J \, |k|^2, \quad |k|^2 = \sum_{a,b,\bar{c}} |k_{ab\bar{c}}|^2, \qquad k_{ab\bar{c}} = \partial_a g_{b\bar{c}} - \partial_b g_{a\bar{c}} \; .$$

-Patch Matching

$$\mu = \frac{1}{N_p!} \sum_{m',l'} \sum_{m.l \neq m',l'} \frac{1}{\mathrm{Vol}_\Omega} \int_{\mathcal{M}} d\mathrm{Vol}_J \, |M(m',l';m,l)|^2,$$

With the total Loss function being

$$\mathrm{Loss} = \alpha_\sigma \sigma + \alpha_\kappa \kappa + \alpha_\mu \mu \, .$$

Point Sampling

Braun, Brelidze, Douglas, Ovrut'07
Anderson, Braun Karp, Ovrut'10
Ashmore, He, Ovrut'19

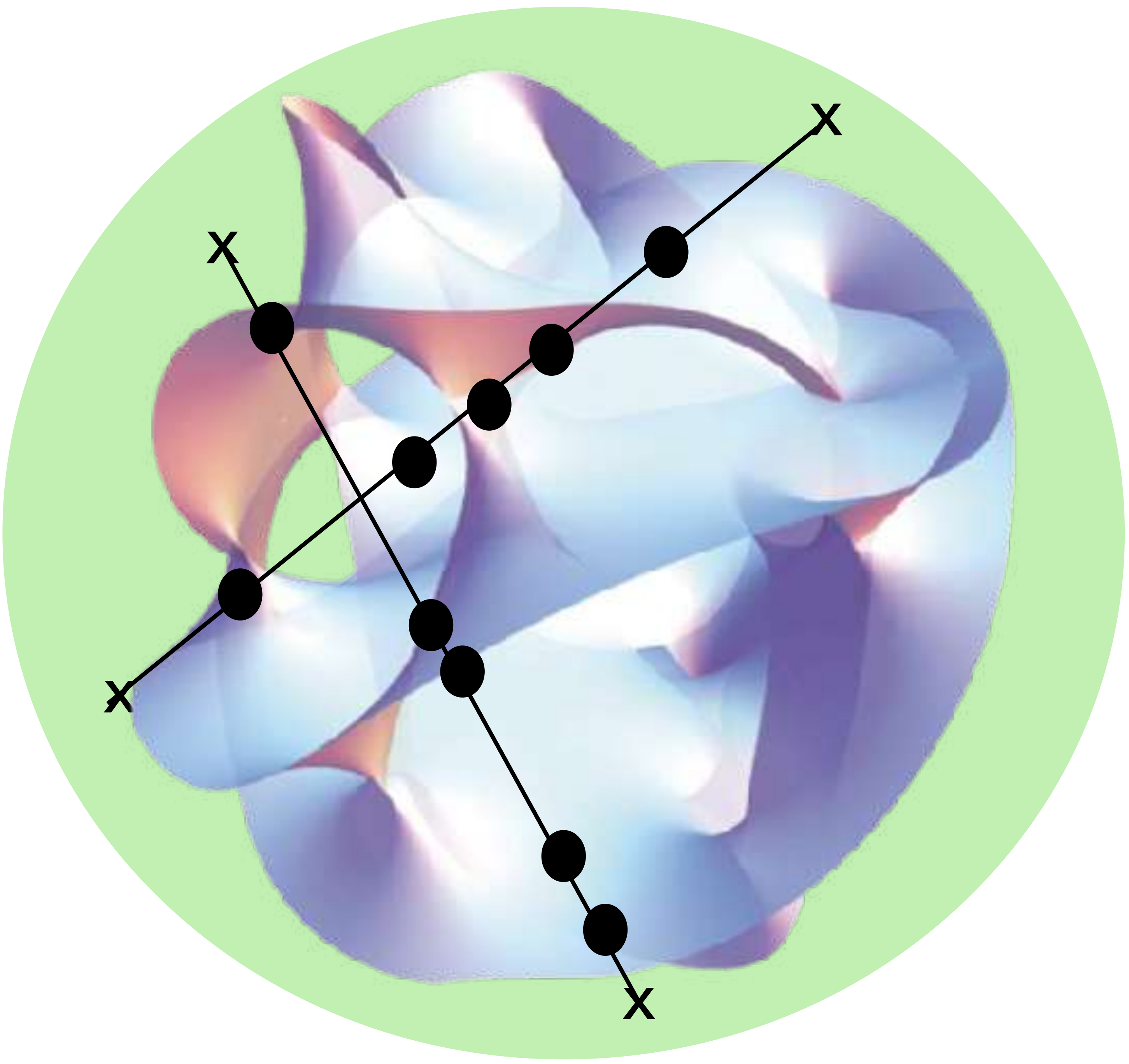
Building Lines in $\mathbb{P}^n$

- Start with selecting random points in $[-1,1]^{2(n+1)}$ and use each point to build a complex vector $v \in \mathbb{C}^{n+1}$. Discard those vectors with $|v| > 1$
- Project v onto the surface of the unitary sphere S^{2n+1}
- As points in $\mathbb{P}^n$, the rescaled v's are uniformly distributed with respect to the $SU(n+1)$ symmetry of the Fubini-Study metric in $\mathbb{P}^n$
- Use any two unitary 's to construct a line

$$L_{ij} = \{v_i + \lambda v_j \,|\, \lambda \in \mathbb{C}\}$$

- The sample points in the hypersurface of interest are obtained as

$$L_{ij} \cap \{p = 0\}$$



Point Sampling

Slight Modification for Tian-Yau Manifold

-Sample points in each $\mathbb{P}^3$ identically as before.

-Use the points to obtain a line and a plane

$$L_{ij} = \{v_i + \lambda v_j \,|\, \lambda \in \mathbb{C}\} \subset \mathbb{P}^3_1$$
$$P_{ijk} = \{v_i + \alpha\, v_j + \beta v_k \,|\, \alpha\,, \beta \in \mathbb{C}\} \subset \mathbb{P}^3_2$$

-Take the points in the Tian-Yau manifold as

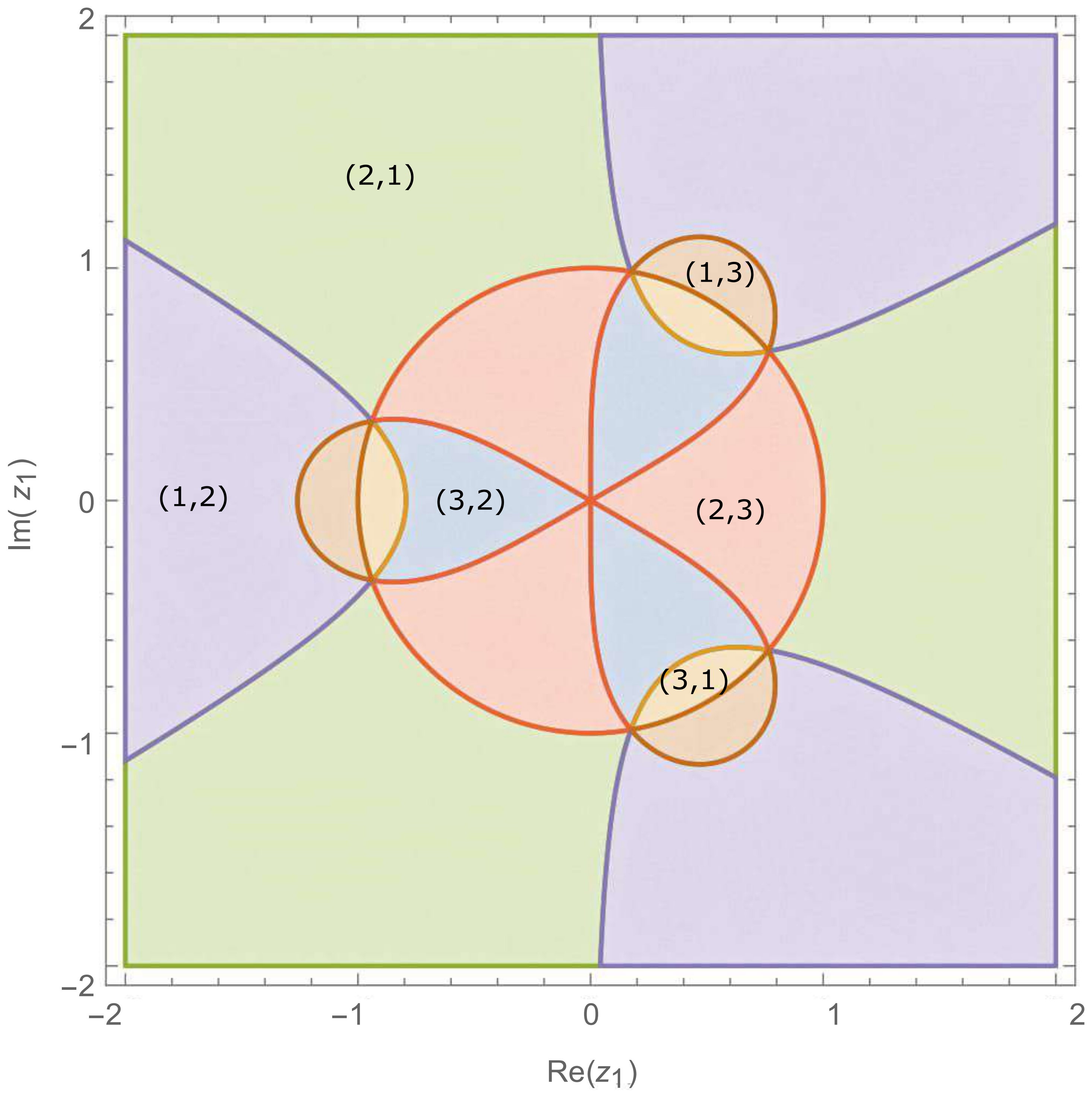
$$(L_{ij} \cup P_{klm}) \cap \{p_1 = p_2 = p_3 = 0\}$$

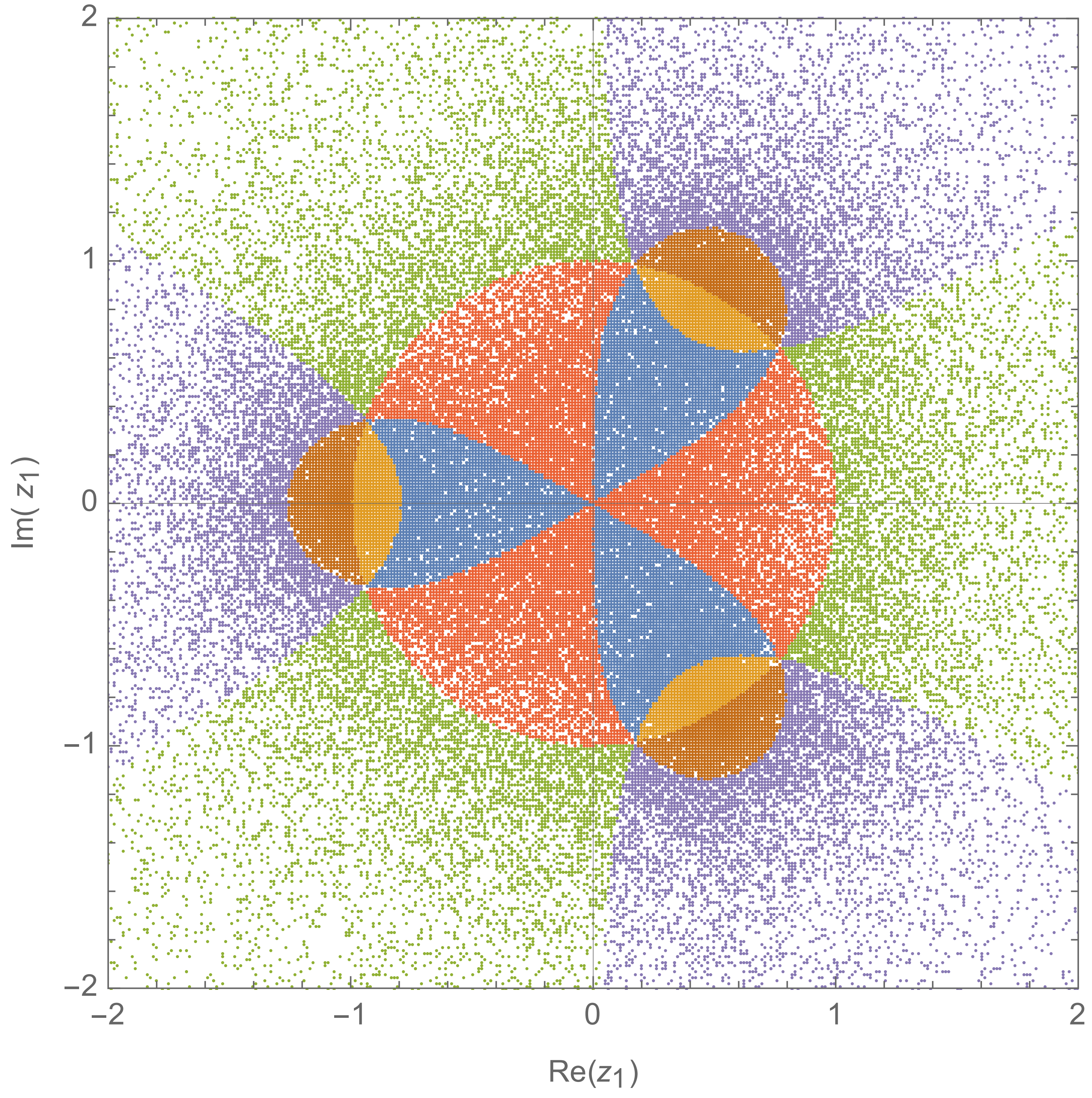
(plus $\mathbb{P}^3_1 \leftrightarrow \mathbb{P}^3_2$)

Sampling points in the torus

$$z_1^3 + z_2^3 + z_3^3 = 0 \subset \mathbb{P}^2$$

For the torus we have six patches fixed upon choice of the affine coordinate and the dependent coordinate. For each point in the torus there is a preferred patch.



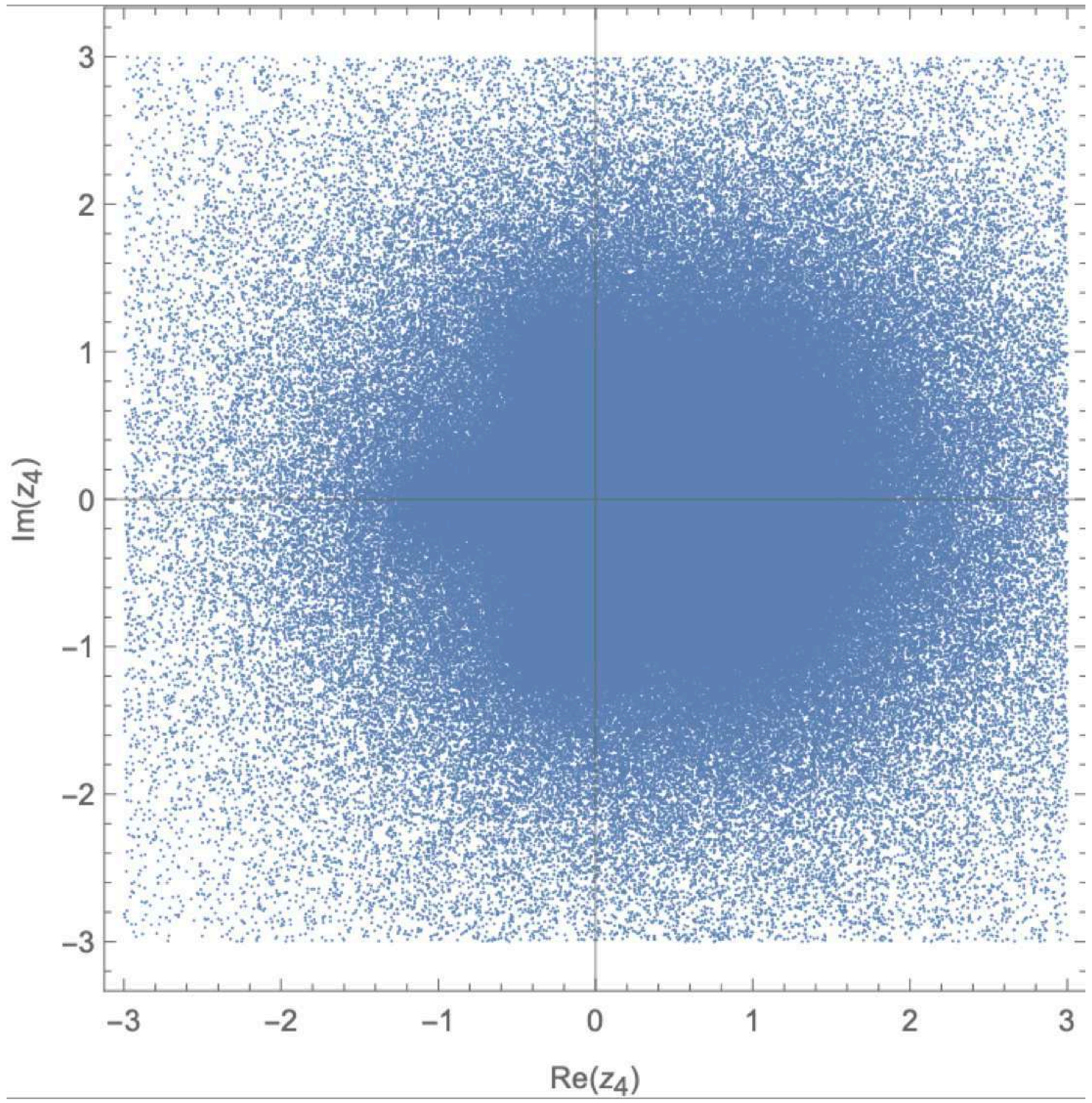


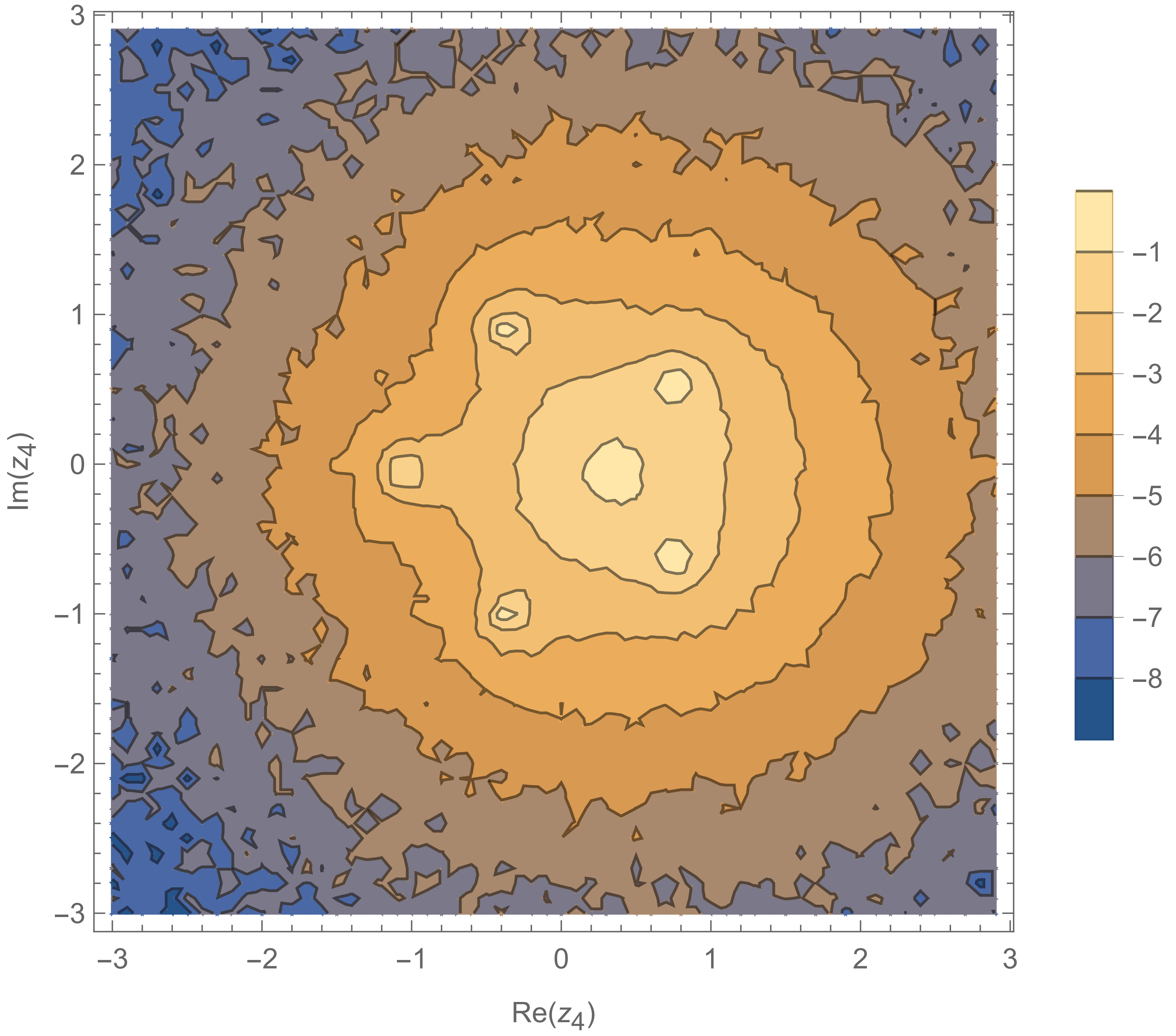
Point Sampling

Sampling points in the Fermat Quintic

Similarly as in the torus case the different patches (i, j) are labeled in terms of the affine coordinate $z_i = 1$ and a dependent coordinate z_j obtained from $p = 0$

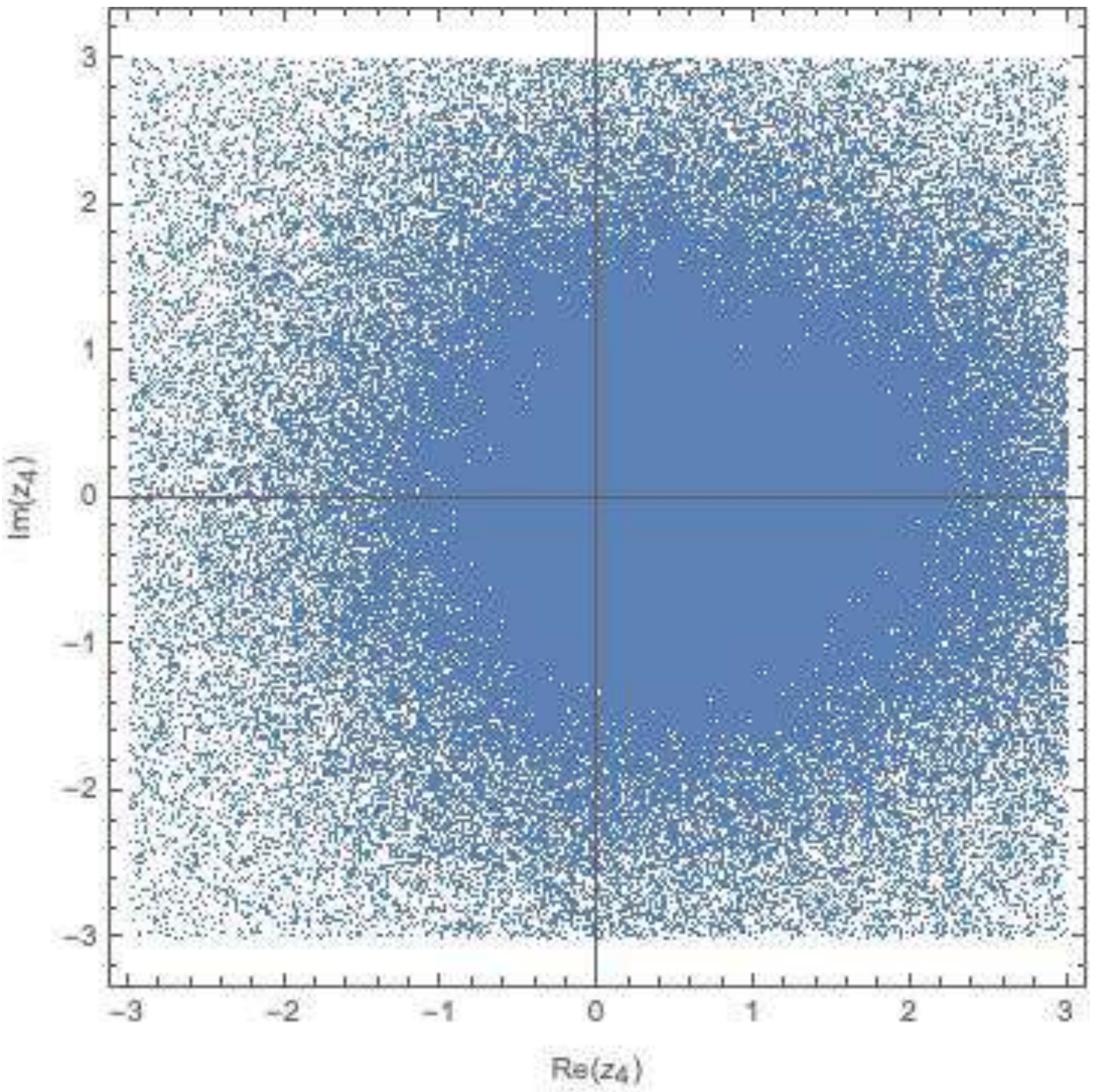
There are 20 patches, all equivalent up to permutations of coordinates.

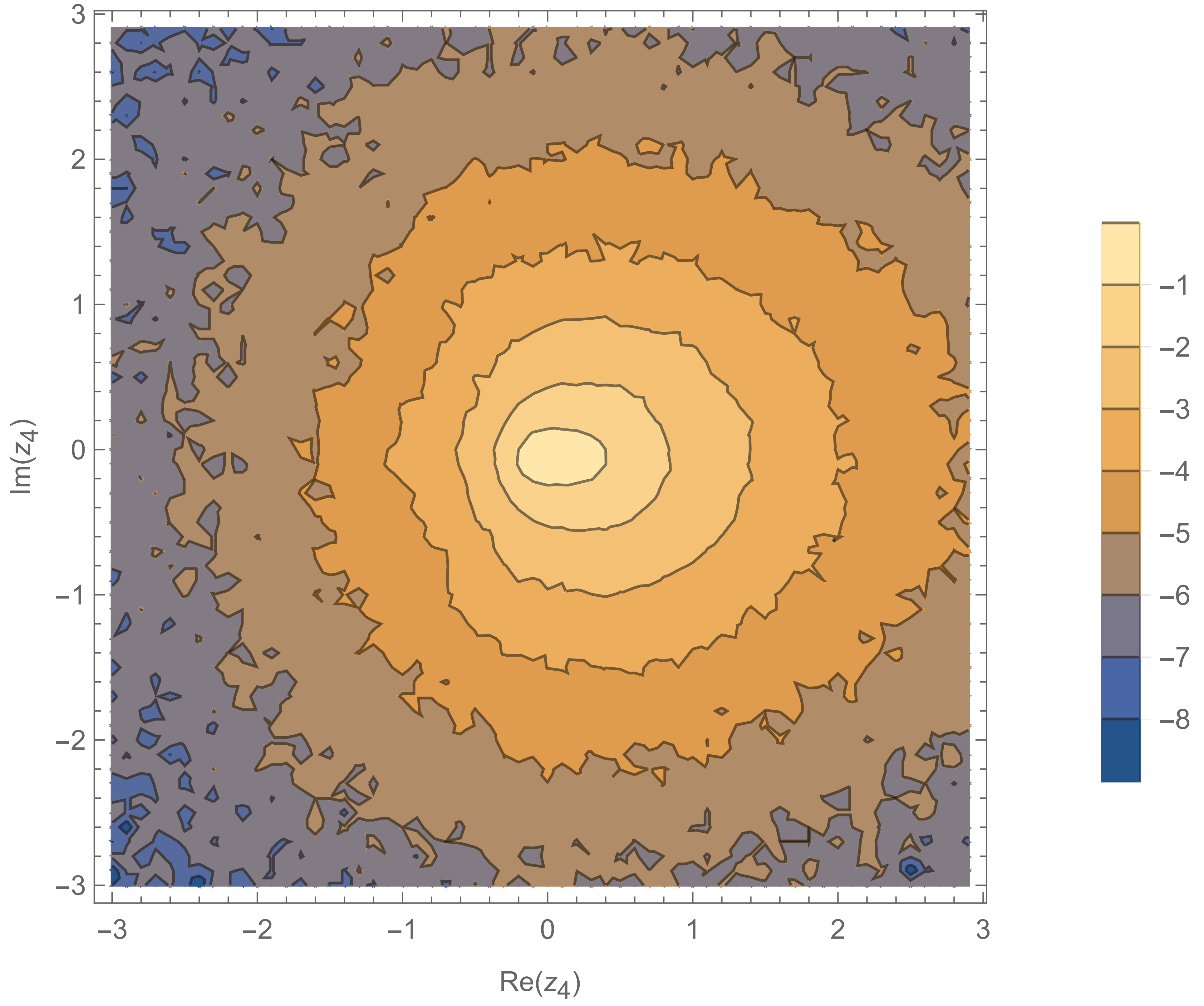




Point Sampling

Sampling points in the Dwork Quintic $\psi = -1/5$





Point Sampling

Sampling points in the Tian-Yau manifold

We considered the following equations defining the Tian-Yau manifold

$$p_1 = \frac{1}{3}(z_1^3 + z_2^3 + z_3^3 + z_4^3)$$

$$p_2 = \frac{1}{3}(z_5^3 + z_6^3 + z_7^3 + z_8^3)$$

$$p_3 = z_1 z_5 + z_2 z_6 + z_3 z_7 + z_4 z_8$$

For $z_i \in \mathbb{P}_1^3 \quad i = 1, ..., 4$ and $z_i \in \mathbb{P}_2^3 \quad i = 5, ..., 8$ The patches for this case are given in terms of five indices $(i, j; k, l, m)$ corresponding to two affine coordinates and three dependent ones that solve for $p_1 = p_2 = p_3 = 0$

This leads us to 192 patches. Considering all symmetries leaving the defining equations invariant, we are left with 4 inequivalent classes of patches. In general, the metrics in inequivalent patches do not agree up to permutations.

Point Sampling

Numerical Integration

- Patches intersect over zero measure sets, hence numerical integration would be a sum over points in different patches if these would be uniformly distributed with respect to the Calabi-Yau metric.

- The sampling method provides points uniformly distributed with respect to the Fubini-Study metric restricted to the Calabi-Yau.

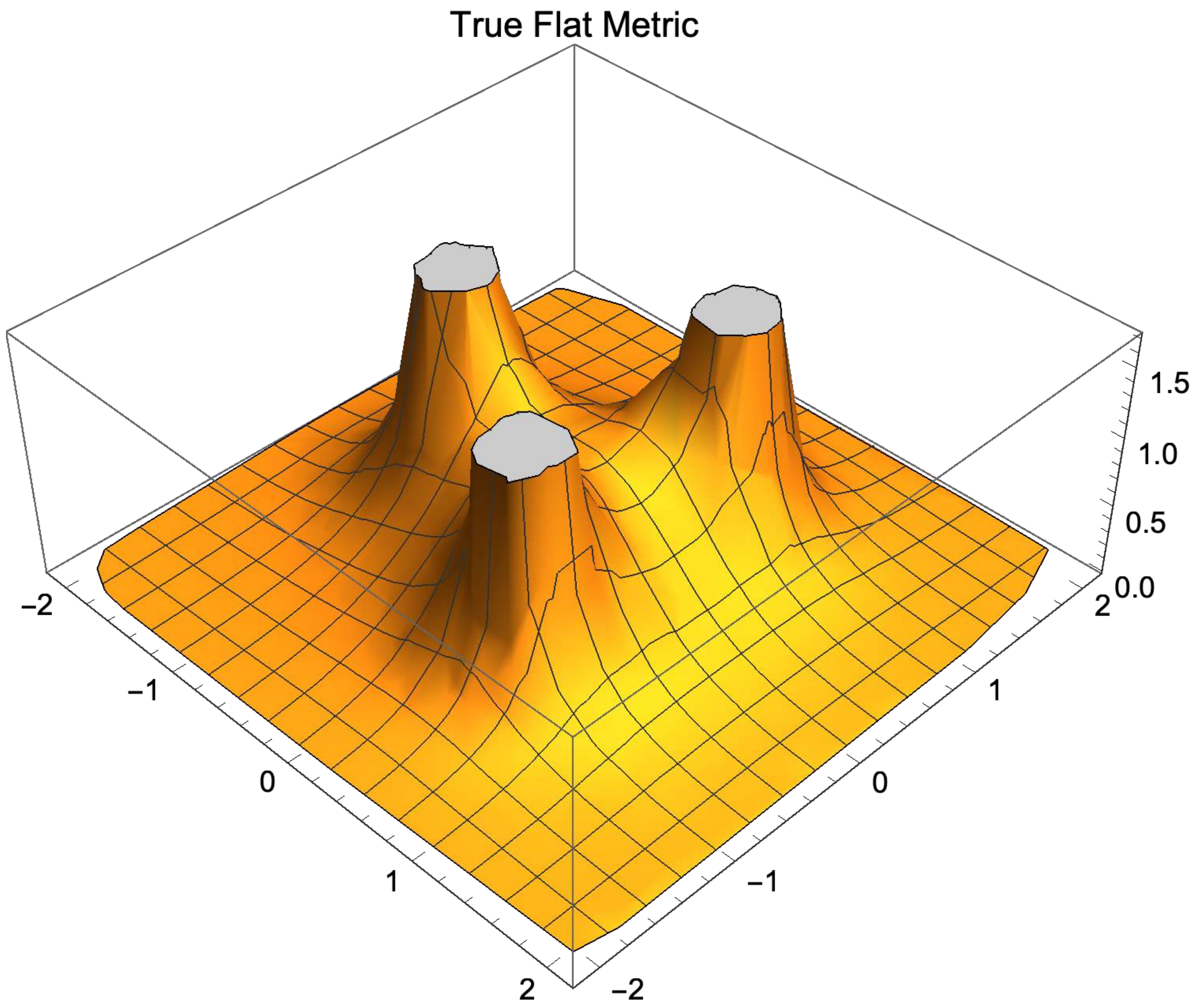
- Numerical integration requires to weight the sample points in order to obtain meaningful quantities

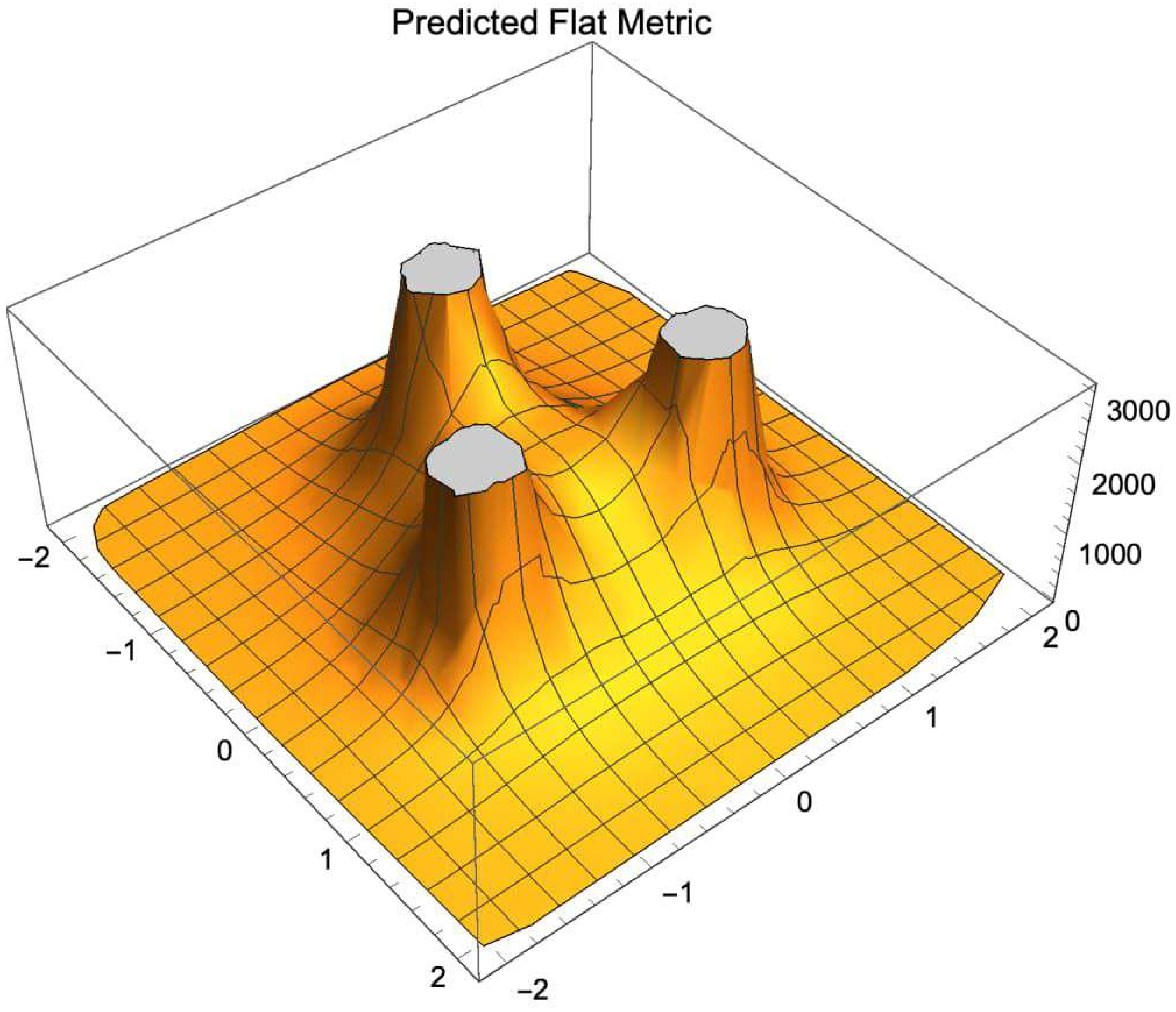
$$\int_{\mathcal{M}} d\mathrm{Vol}_{\Omega}\, f(z,\bar{z}) = \int_{\mathcal{M}} d\mathrm{Vol}_{FS} \left(\frac{d\mathrm{Vol}_{\Omega}}{d\mathrm{Vol}_{FS}} \right) f(z,\bar{z})$$

$$\int_{\mathcal{M}} d\mathrm{Vol}_{\Omega}\, f(z,\bar{z}) = \frac{1}{N} \sum_{l=1}^{M} w(p_l) f(p_l) \qquad w(p_l) = \frac{d\mathrm{Vol}_{\Omega}(p_l)}{d\mathrm{Vol}_{FS}(p_l)}$$

Calabi-Yau metrics (results)

The Torus

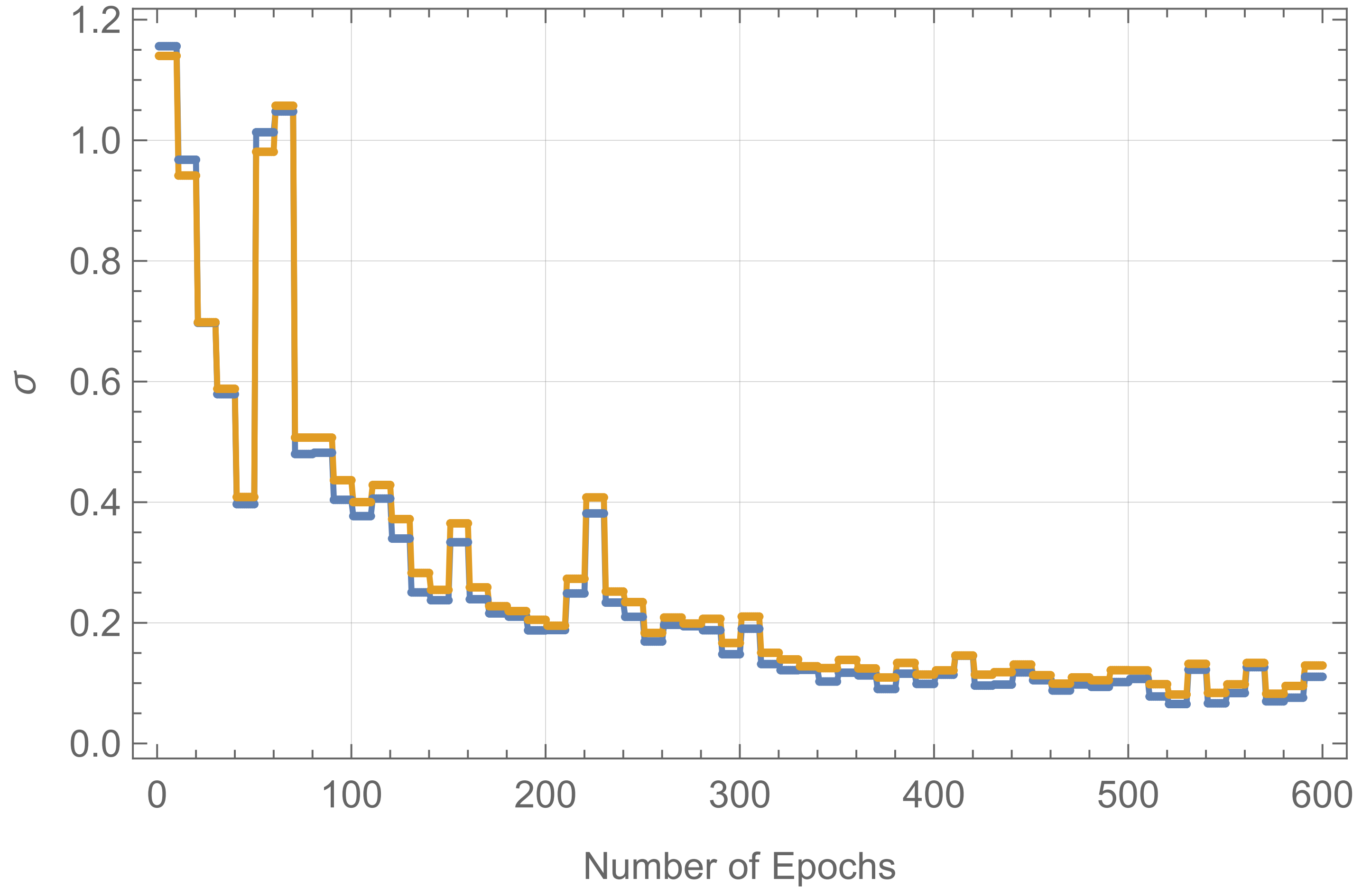




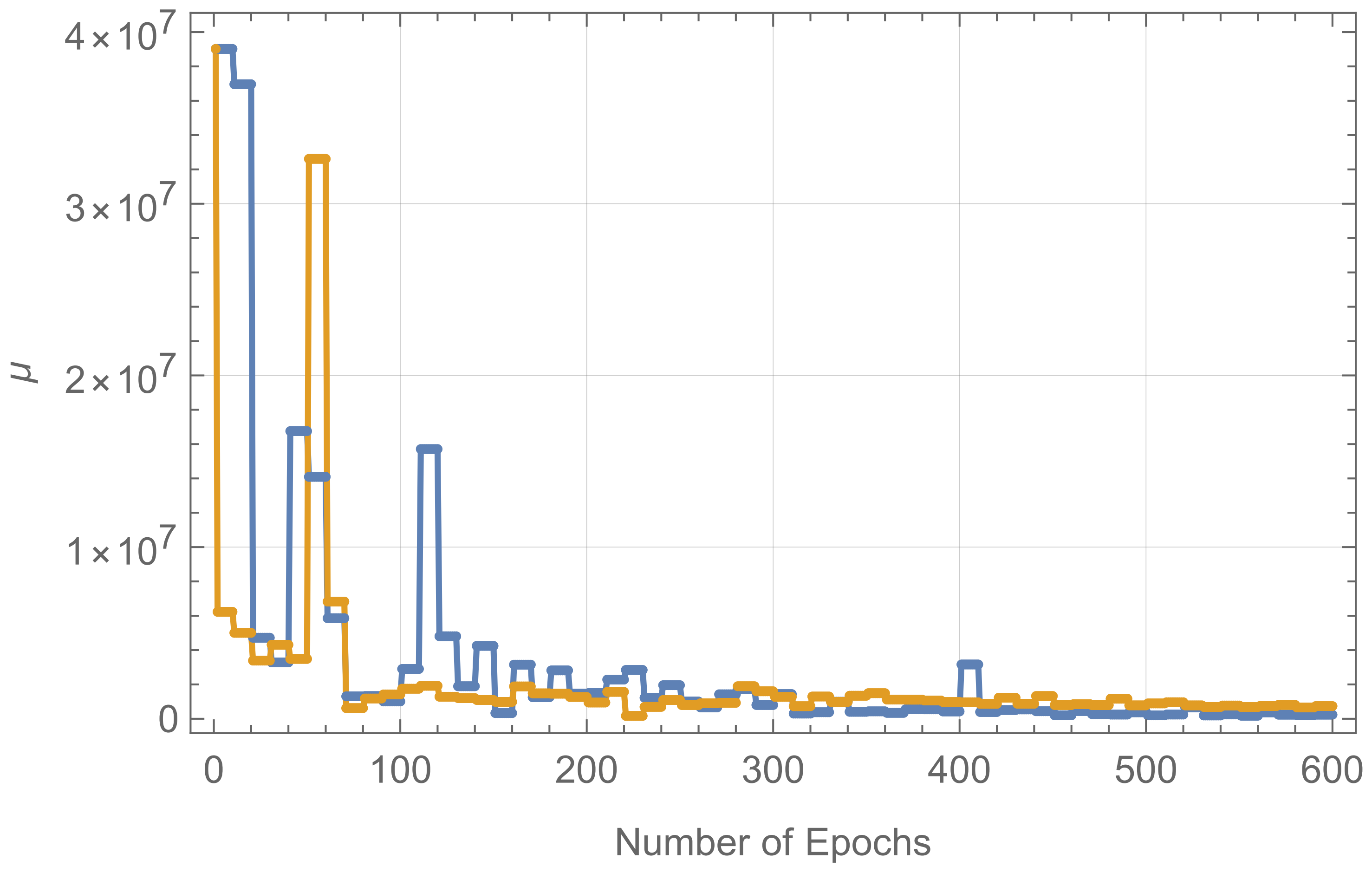
Calabi-Yau Metrics (Results)

The Quartic K3

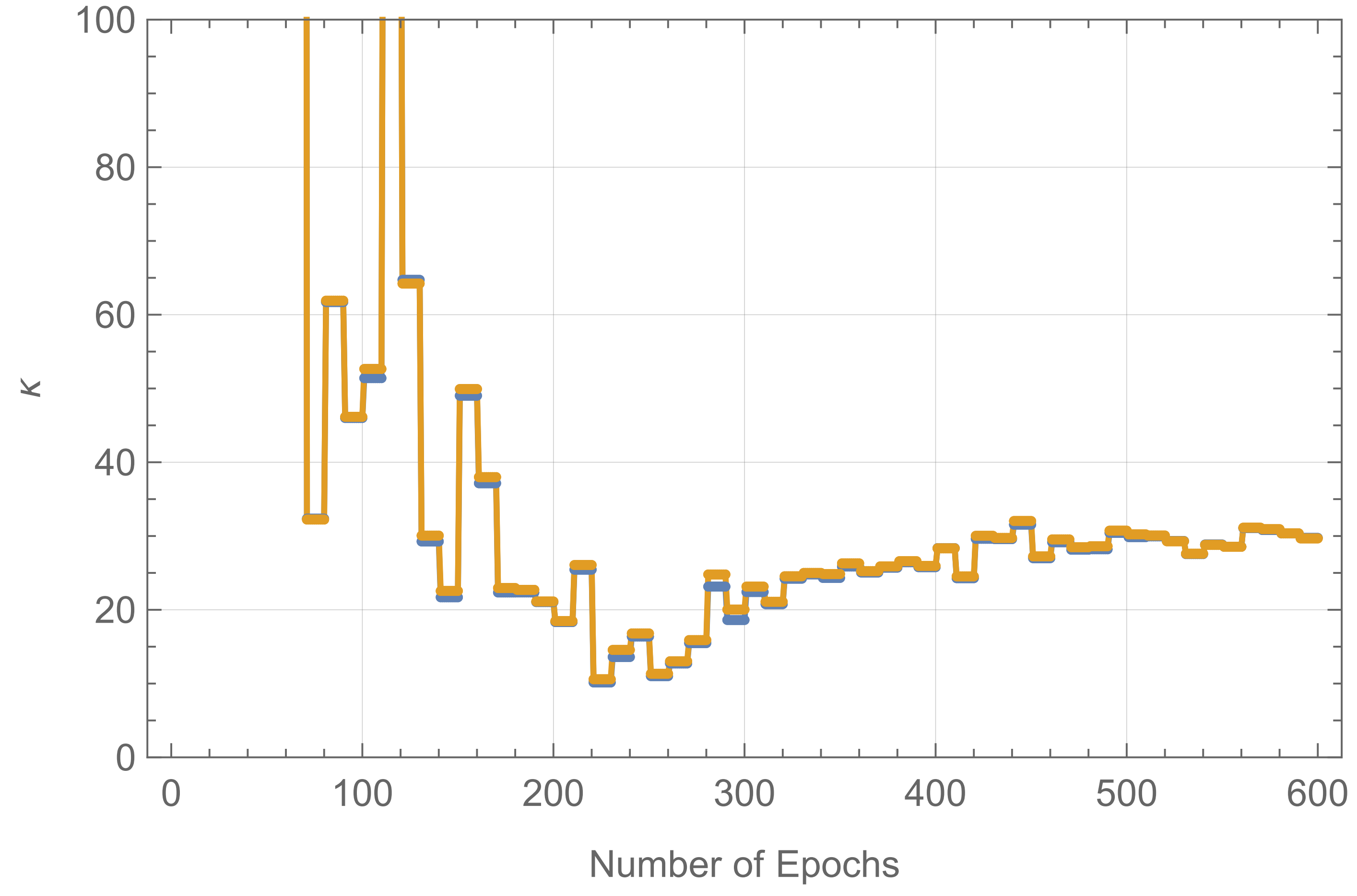
σ



μ



κ



We reach sigma values after training of 0.18. Ignoring the kaehlericity loss and the mu loss this could be compared k=6 in Donaldson's algorithm.

Calabi-Yau Metrics (Results)

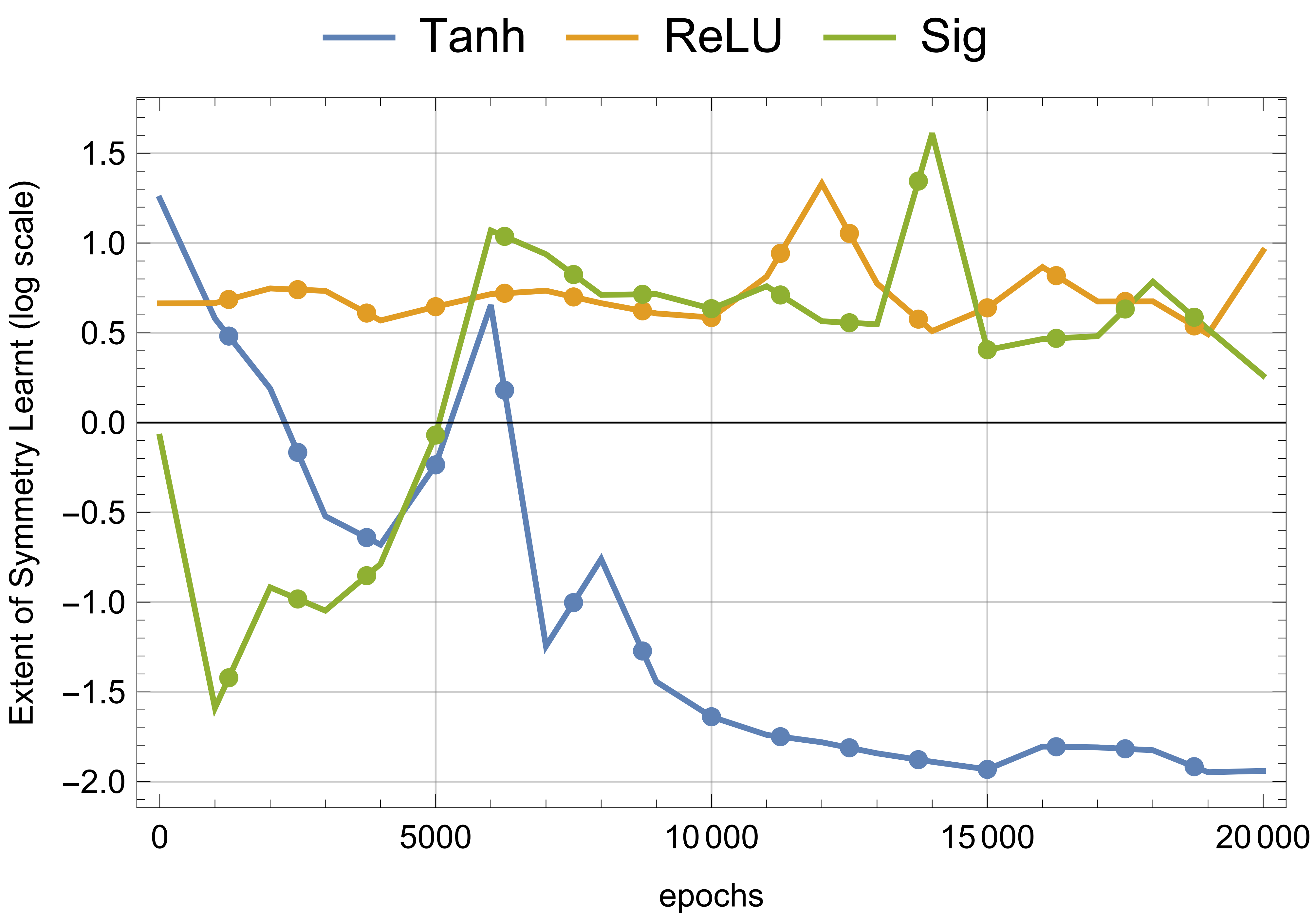
K3: Test of Symmetry Learning

$$\text{K3}: \quad z_1^4 + z_2^4 + z_3^4 + z_4^4 = 0 \subset \mathbb{P}^3$$

$$z_p \to \omega_p \, z_p \; ; \quad \text{with} \quad p \in \{1,2,3,4\} \quad \text{and} \quad \omega_p \in \mathbb{Z}_4$$

$$\tau : \quad z_1 \mapsto i \, z_1 \; , \; z_2 \mapsto - z_2 \; , \; z_3 \mapsto -i \, z_3 \; .$$

$$\delta_n(\tau) := \frac{1}{N} \sum_z \text{abs} \left(\frac{g_{NN}(\theta_n; z) - g_{NN}(\theta_n; \tau.z)}{g_{NN}(\theta_n; z)} \right)$$



Calabi-Yau metrics (results)

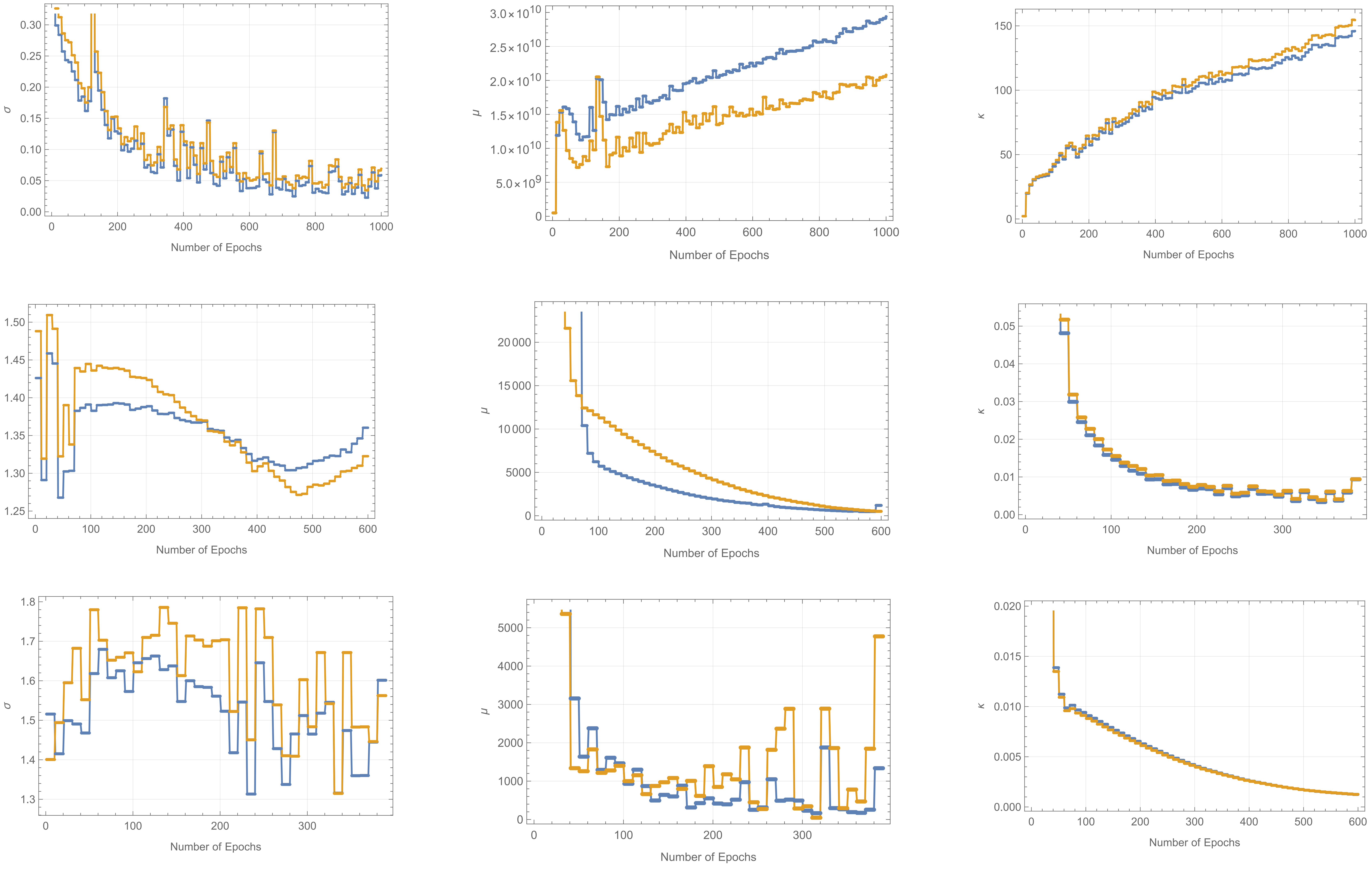
The Fermat Quintic

σ | μ | κ

Training with σ only

Training with μ only

Training with κ only



Calabi-Yau metrics (results)

The Fermat Quintic

σ μ κ

28.000 Points

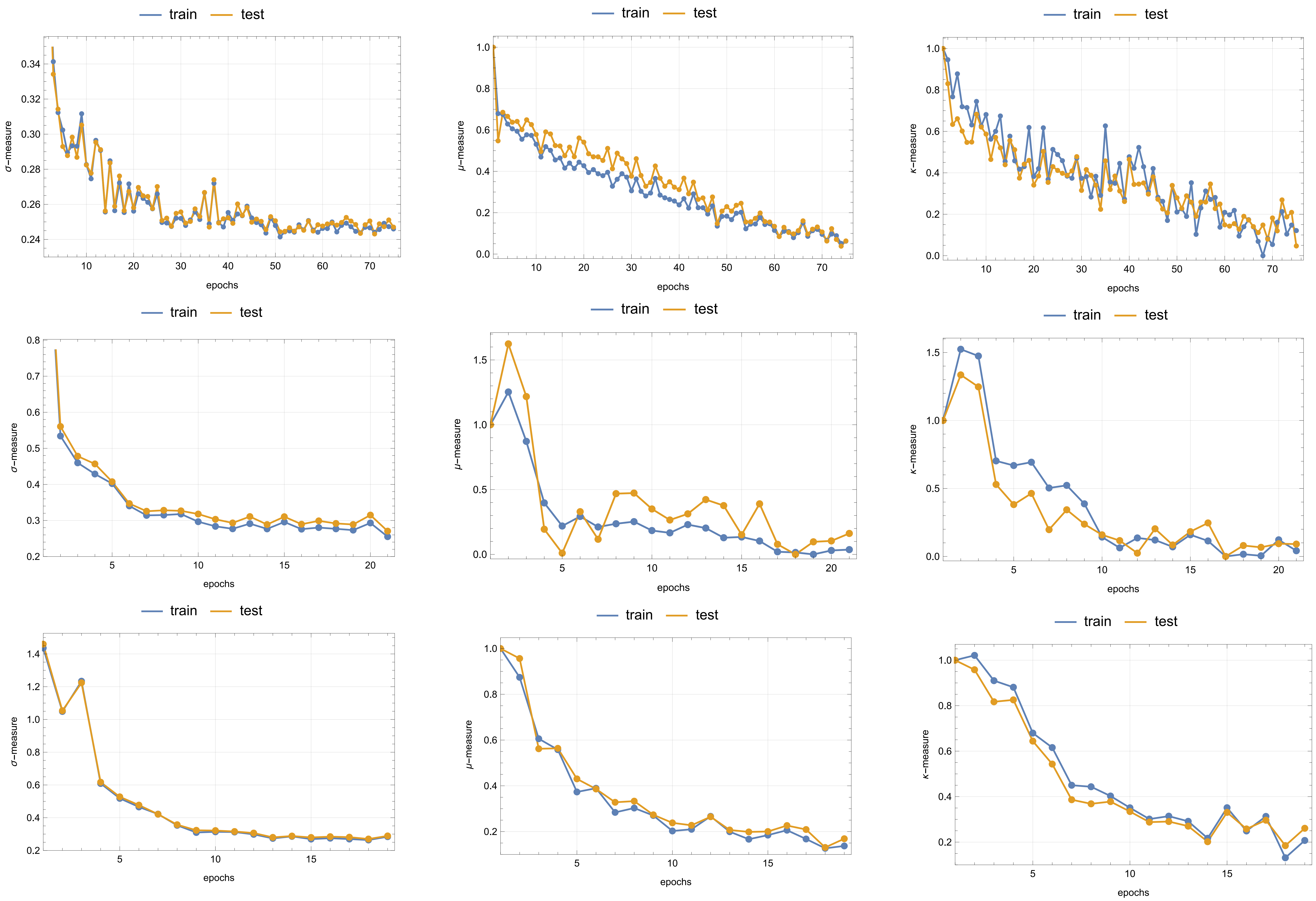
Learning Rate 10^{-4}

10.000 Points

Learning Rate 10^{-4}

10.000 Points

Learning Rate 10^{-3}



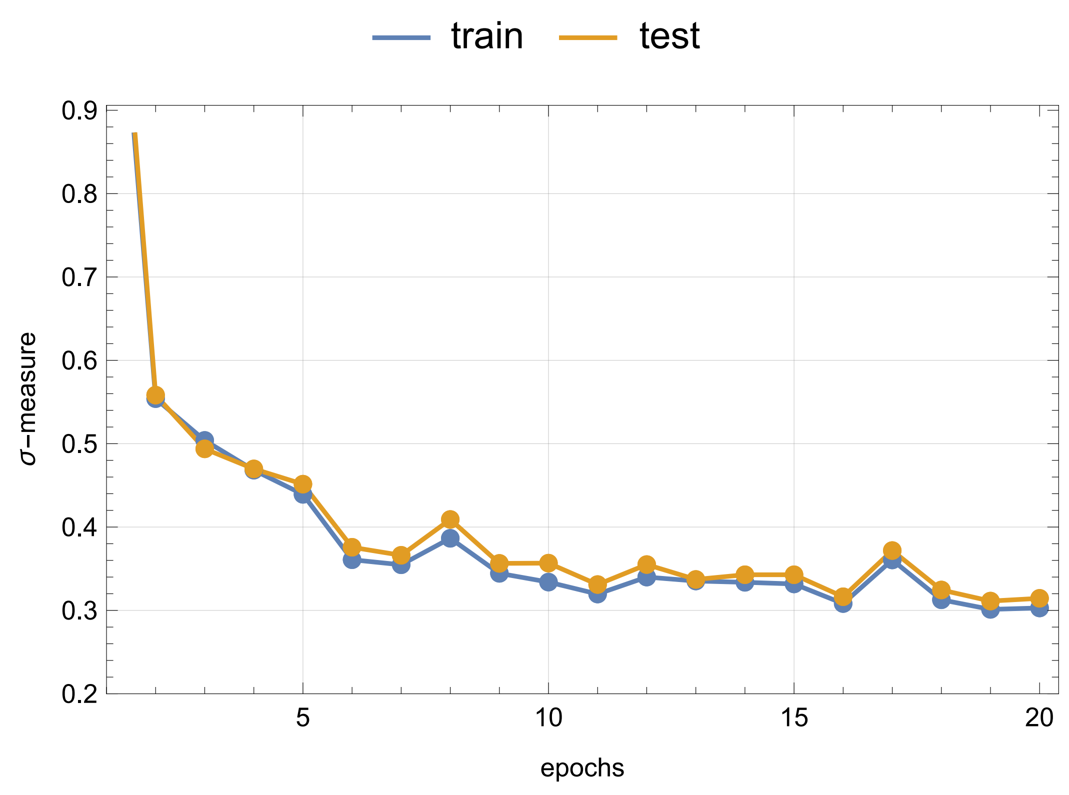
Calabi-Yau metrics (results)

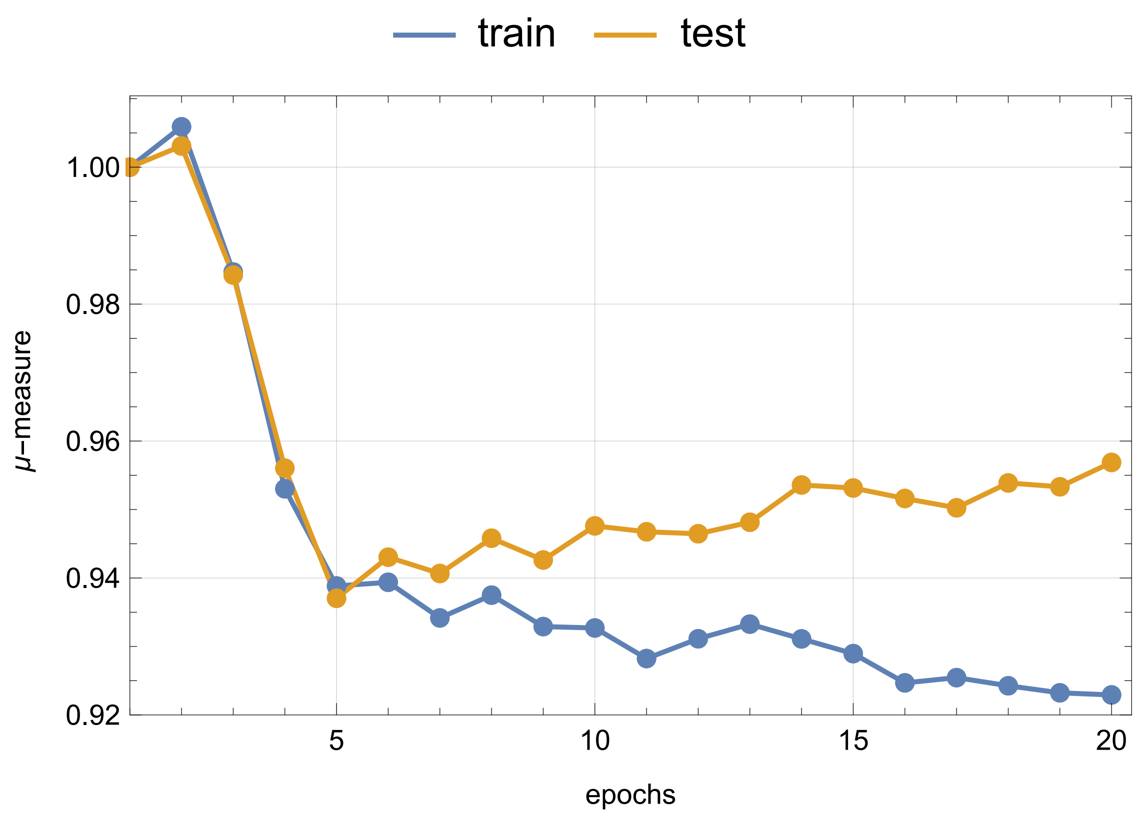
The Dwork Quintic $\psi = -1/5$

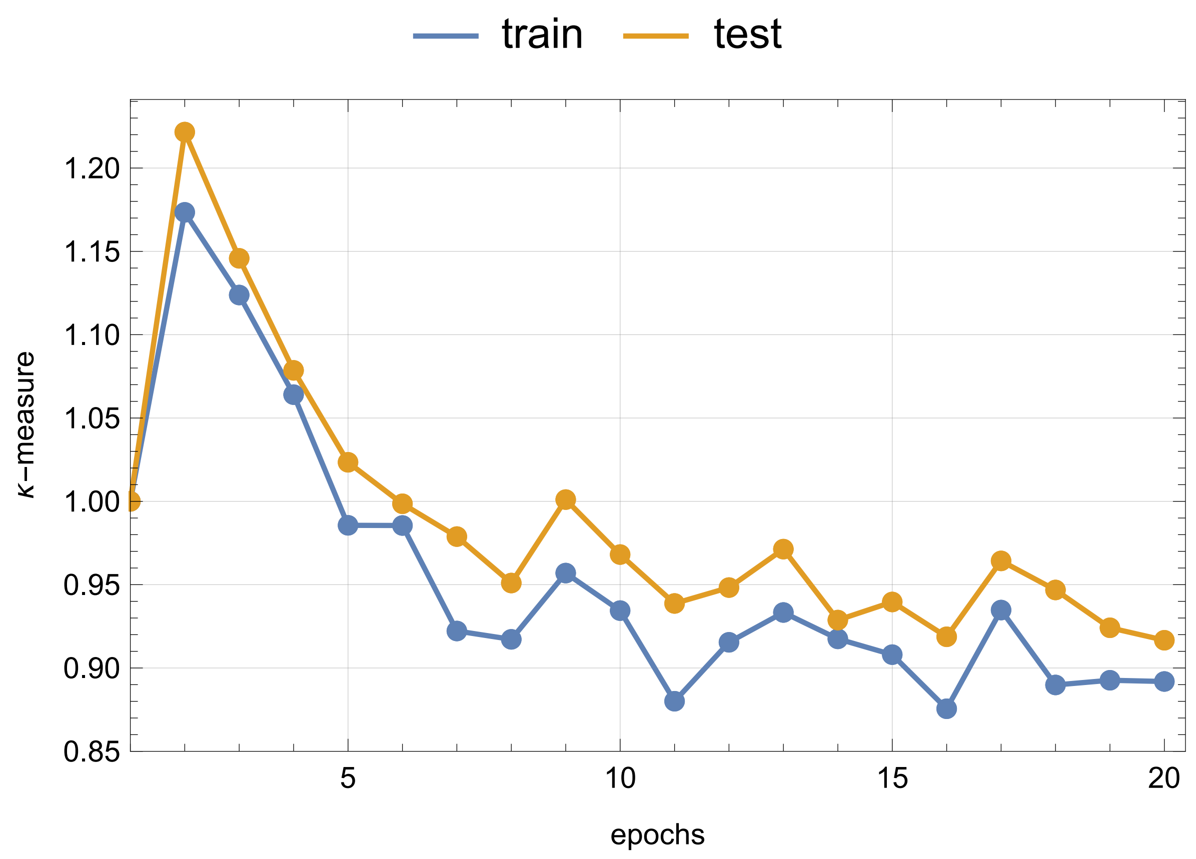
σ μ κ

10.000 Points

Learning Rate 10^{-4}

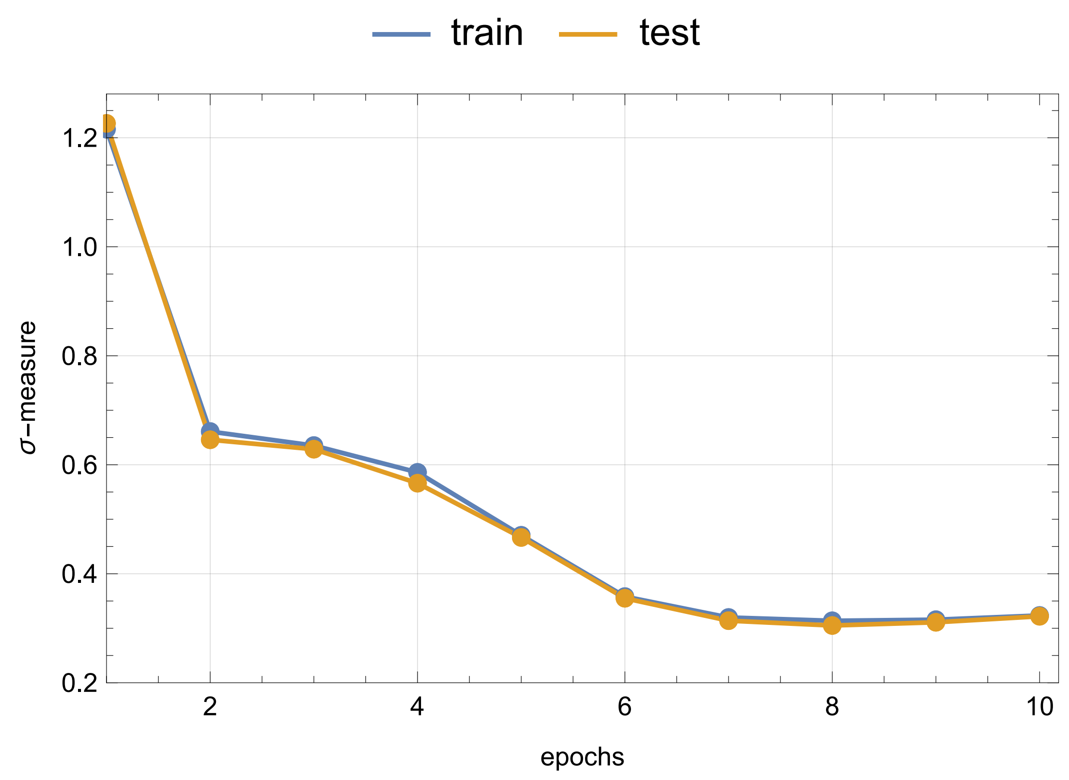


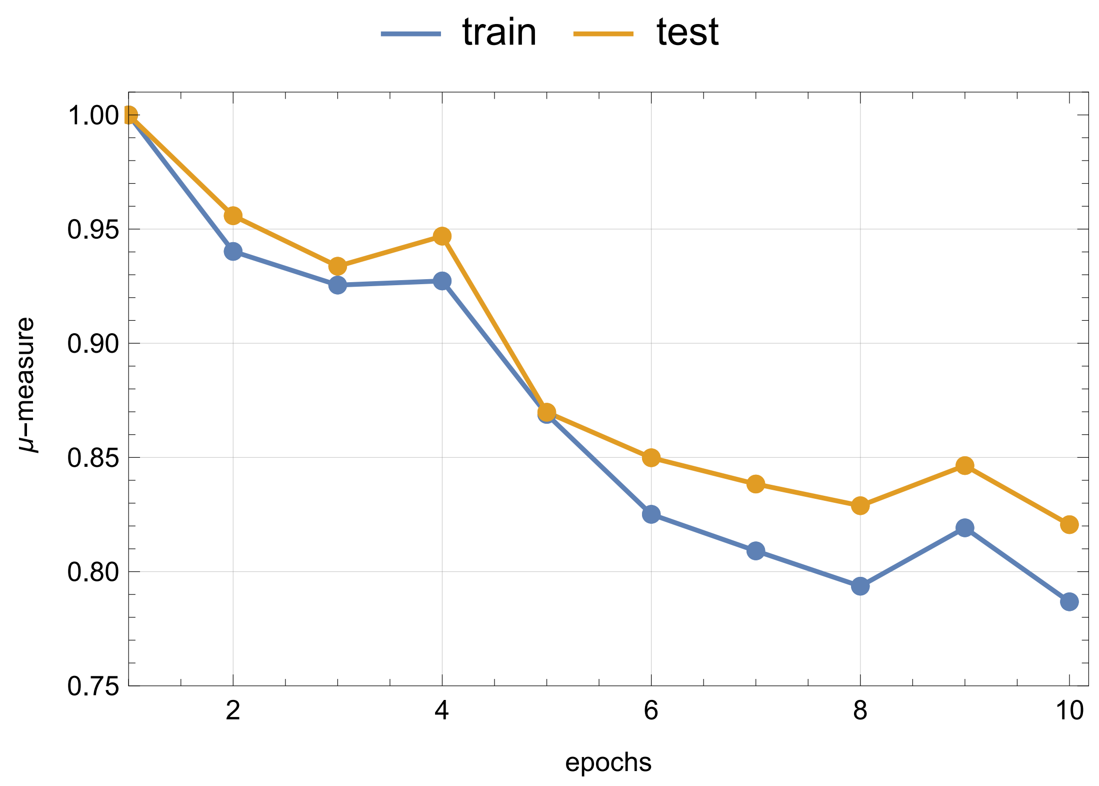


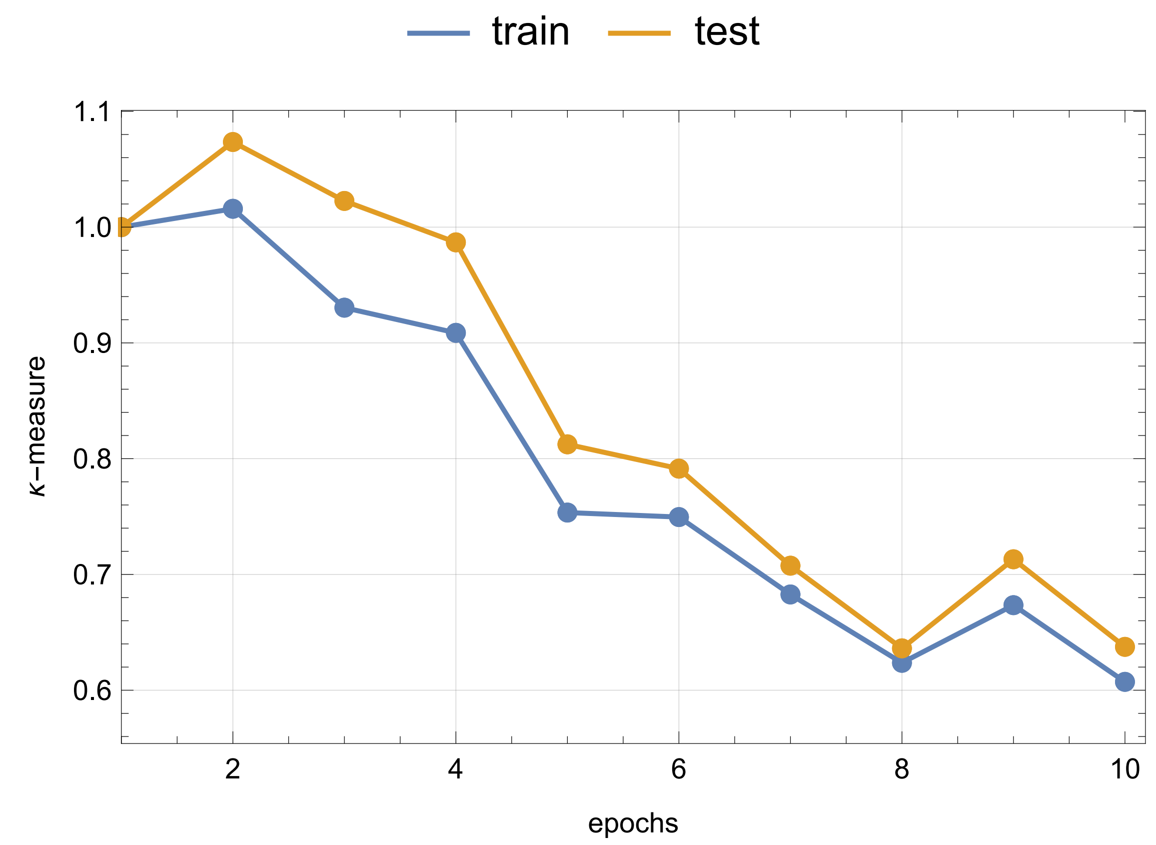


10.000 Points

Learning Rate 10^{-3}



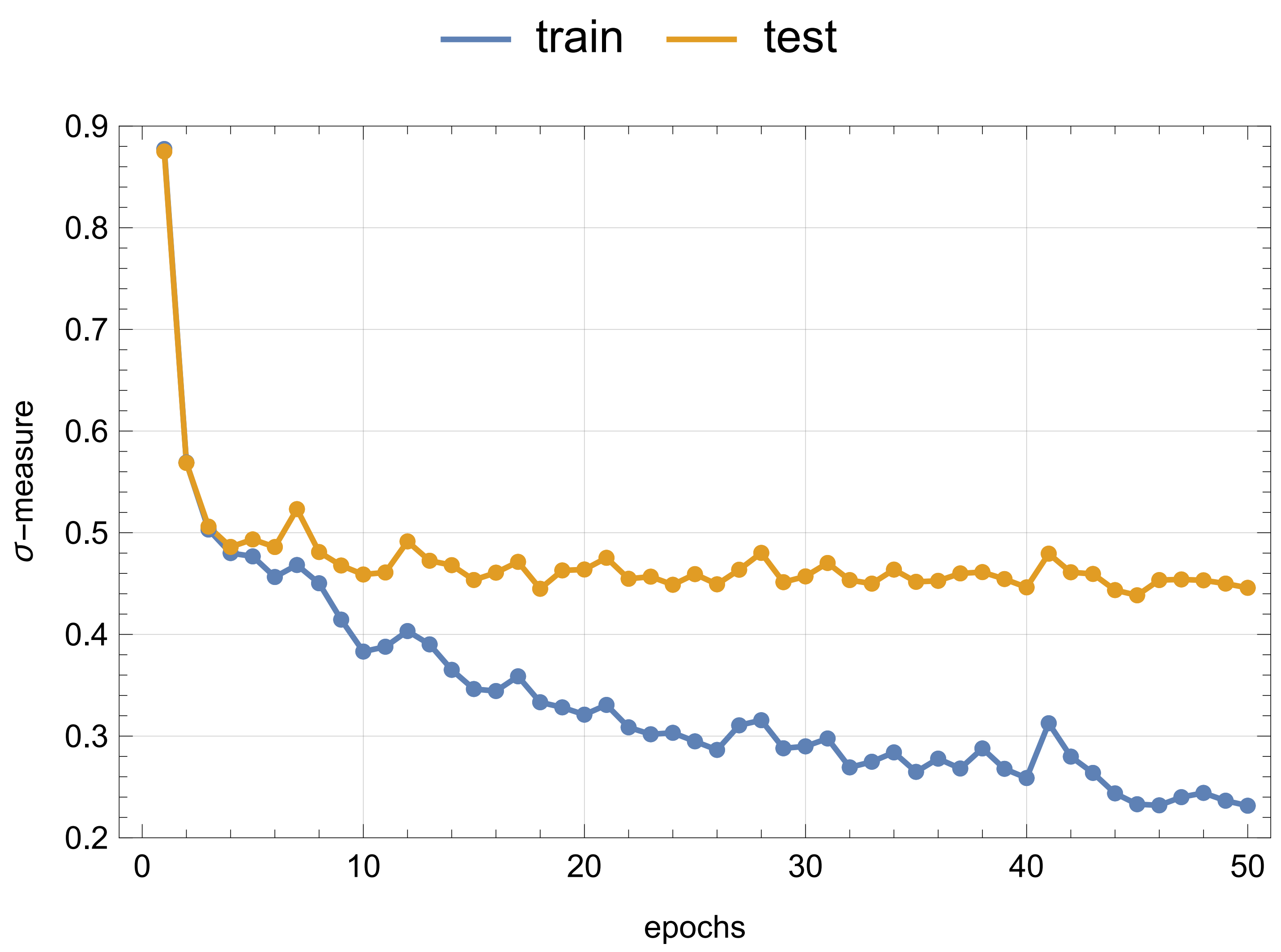




Calabi-Yau metrics (results)

Tian-Yau

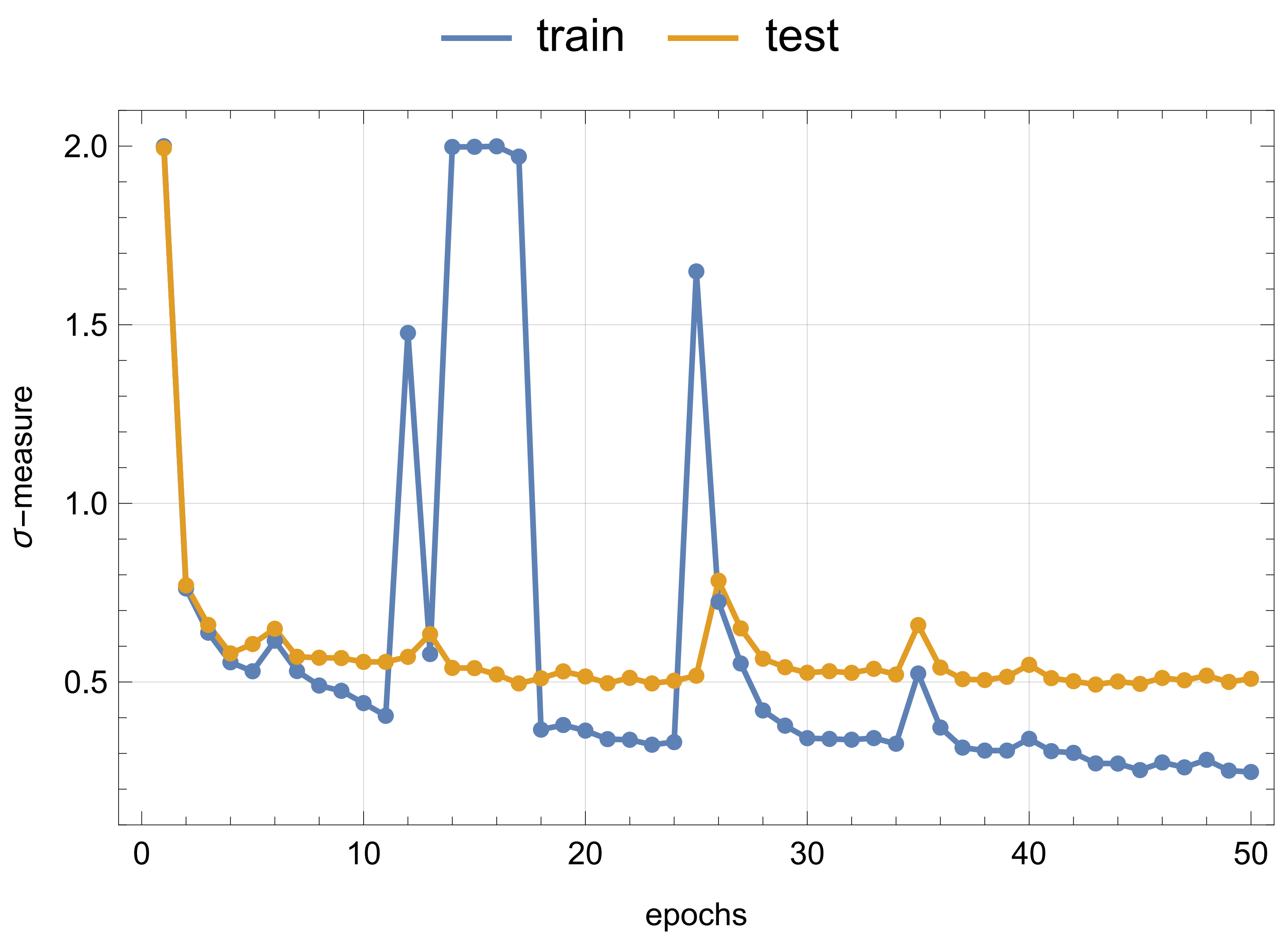
For this example we need to modify the architecture as we need to predict simultaneously the metric on four different inequivalent patches. As a preliminary approach we consider the same architecture as before and train on sigma only.



5.000 Points

ReLU Activation Function

Learning Rate 10^{-5}



5.000 Points

Tanh Activation Function

Learning Rate 10^{-5}

Calabi-Yau metrics (results)

Tian-Yau

Issues:

-Need to train on full Loss, include other measures beyond one patch class. (Results coming soon)

-We do not know the Kahler class. As we approach zero net loss we are getting closer to a flat Kahler metric but we do not know which one. (Possible issues with sampling)

Main Interest:

-Quotienting Tian-Yau by a freely acting $\mathbb{Z}_3$

$$(z_1, z_2, z_3, z_4) \to (z_1, \alpha^2 z_2, \alpha z_3, \alpha z_4)$$
$$(z_5, z_6, z_7, z_8) \to (z_5, \alpha z_6, \alpha^2 z_7, \alpha^2 z_8)$$

With α a cubic root of 1, leads to a three generation heterotic model. Symmetry simplifies Yukawa coupling computations (as well as field normalizations). This could be an interesting setting to check how good the metric approximations must be in order to provide reliable Yukawa info.

Greene, Kirklin, Miron, Ross'86
Candelas, Kalara'87

Comments on Ricci-Flow

Finding the Ricci-flat metric in the class of g amounts to solving the Monge-Ampère equation

$$(g+\partial\bar{\partial}\phi)^n = e^f g^n$$

For smooth and real ϕ and $e^f g^n = \Omega \wedge \bar{\Omega}$

One can think of ϕ to have some parametric dependence $\phi(\lambda)$, in such a way that for some $\lambda = T$ one obtains the desired Ricci flat metric. This can be thought of as a flow in the space of metrics correlating suitably between $g(\lambda)$ and $f(\lambda)$.

This reminiscent of Ricci flow. Ricci flow is the gradient flow of the Einstein-Hilbert action and is governed by

$$\frac{\partial g}{\partial \lambda} = -R$$

-Short term existence of solutions for real manifolds.

-Long term existence guaranteed for Kähler manifolds.

Ricci flow starting from an arbitrary Kähler metric converges to the Ricci-flat metric in its class. The Kähler class is preserved throughout the flow.

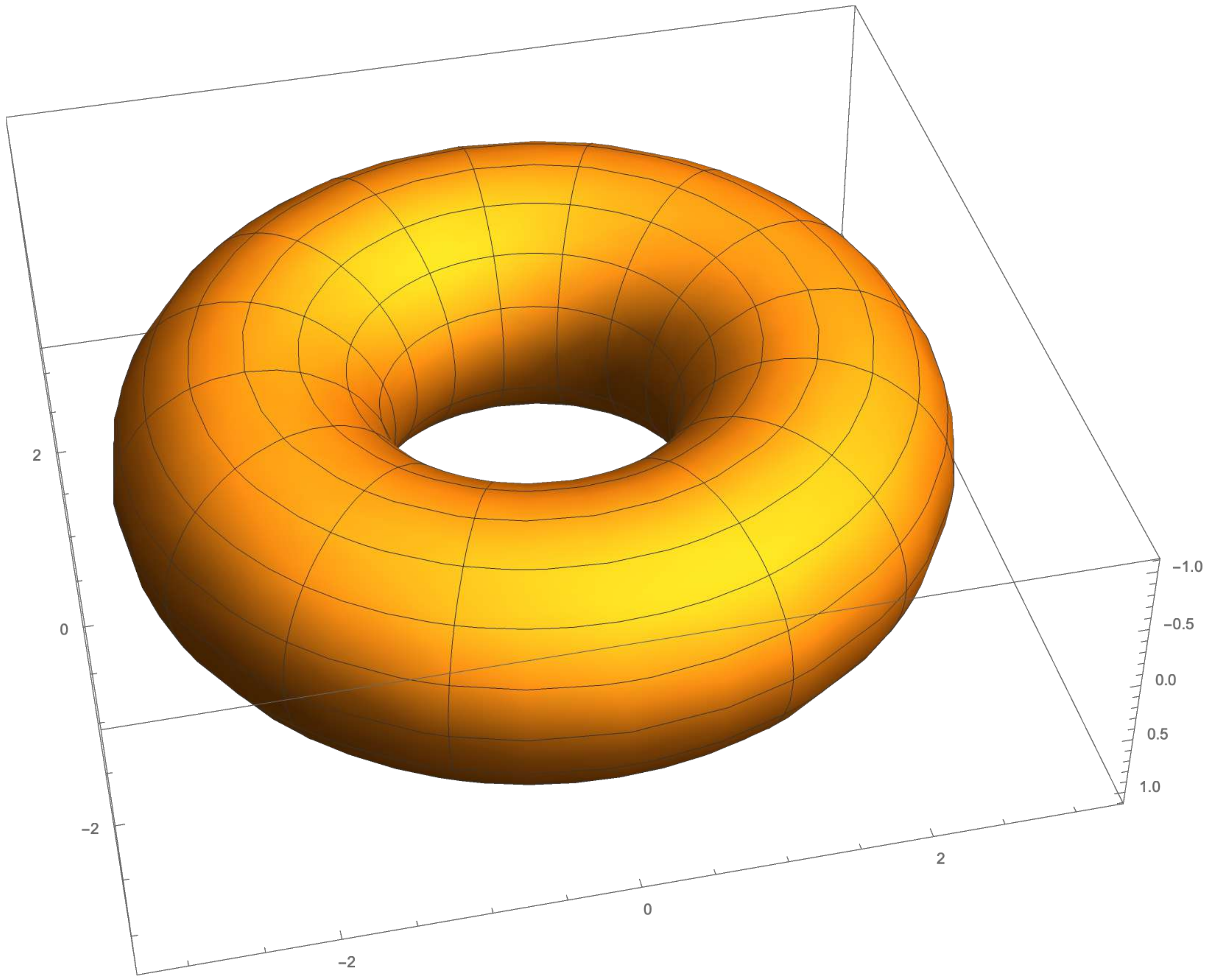
Cao’85

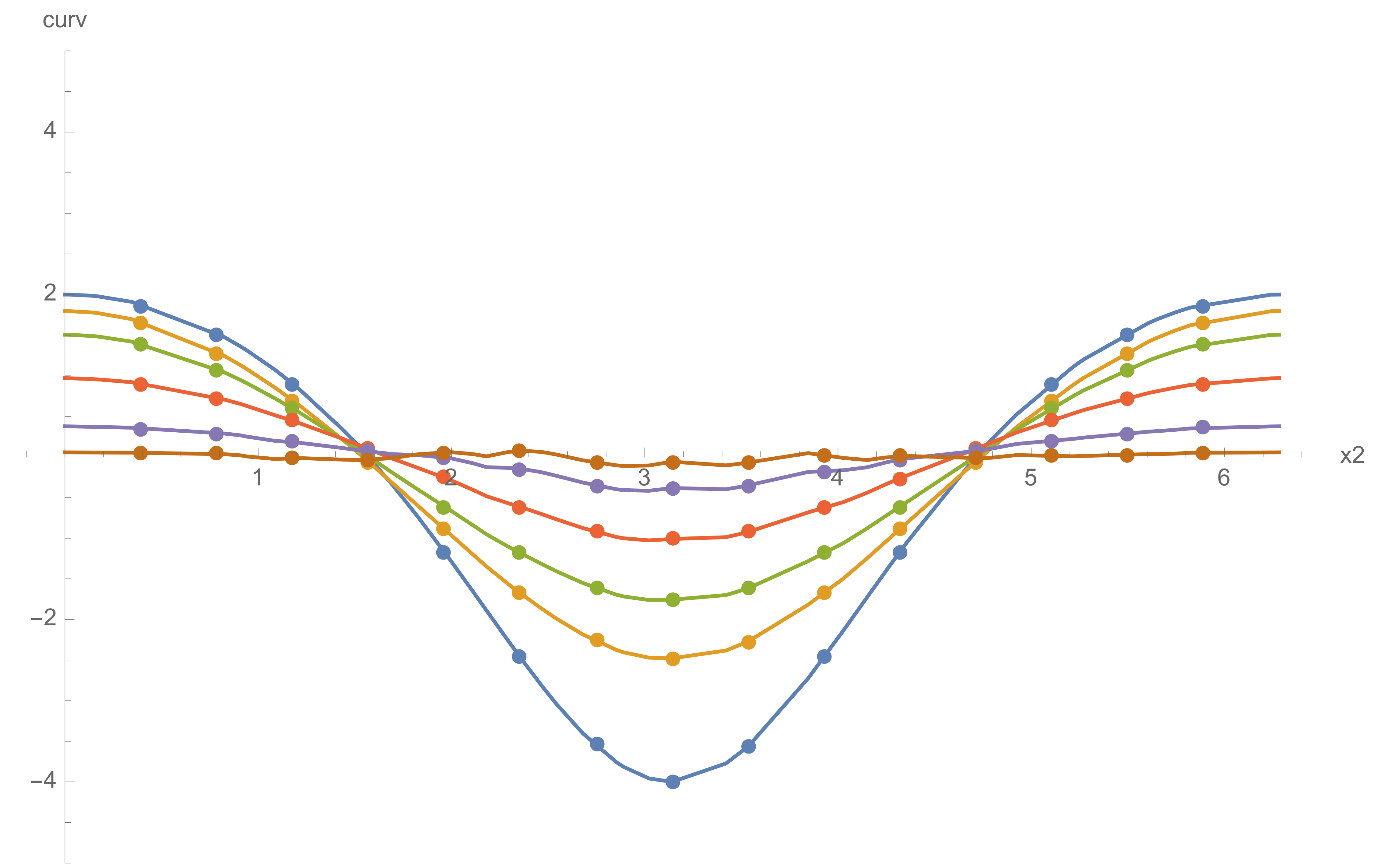
Comments on Ricci-Flow

-Solving Ricci flow implies obtaining a family of metrics, instead of just the desired Ricci-flat one.

-Numerical solutions usually involve iteration errors that propagate as λ evolves.

-One needs a (positive) representative of the Kähler class to start with.





Comments on Ricci-Flow

Instead of looking for a Ricci flow solution, one could instead for a “potential” that has the Ricci-flat metric as a minimum.

Consider for example Perelman’s entropy functional

$$\mathcal{F}(g,f) = \int_{\mathcal{M}} d\mu\ e^{-f}\left(R + |\nabla f|^2\right) = \int_{\mathcal{M}} dm\ \left(R + |\nabla f|^2\right)\ ,$$

Where physical speaking we have introduced the dilaton f. Here f is responsible for keeping the differential volume fixed.

The variation of the entropy functional gives the following modified Ricci flow

$$\frac{\partial}{\partial\lambda} g_{a\bar{b}} = -(\mathrm{Ric}_{a\bar{b}} + \nabla_a \nabla_{\bar{b}} f) \qquad *$$

Together with $\frac{\partial}{\partial\lambda}(d\mu e^{-f}) = 0$

$$\frac{\partial}{\partial\lambda} f = -\Delta f - R \qquad **$$

At first glance one might solve * for an initial condition on the metric, but then, looking at the evolution of the dilaton, one observes that it follows a backward heat Equation, with no solution guaranteed for an initial condition.

Comments on Ricci-Flow

The previous system can be recast back to the original Ricci flow with a decoupled equation for the dilaton

$$\frac{\partial}{\partial\lambda} g_{a\bar{b}} = -\mathrm{Ric}_{a\bar{b}}\,,$$

$$\frac{\partial}{\partial\lambda} f \; = \; -\Delta f + |\nabla f|^2 - R\,.$$

This system has a solution for initial $g(0)$ and final $f(T)$ in the interval $[0,T]$

Along the flow, $\mathcal{F}$ grows monotonically

$$\frac{d}{d\lambda}\mathcal{F} = 2\int_{\mathcal{M}} d\mu\; e^{-f}|\mathrm{Ric}_{a\bar{b}} + \nabla_a\nabla_{\bar{b}} f|^2.$$

Inspired by this we can think of loss functions that get minimized for the Ricci-flat metric. One possibility could be

$$\mathrm{Loss} = \int_{\mathcal{M}} d\mu\; e^{-f}|\mathrm{Ric}_{a\bar{b}} + \nabla_a\nabla_{\bar{b}} f|^2.$$

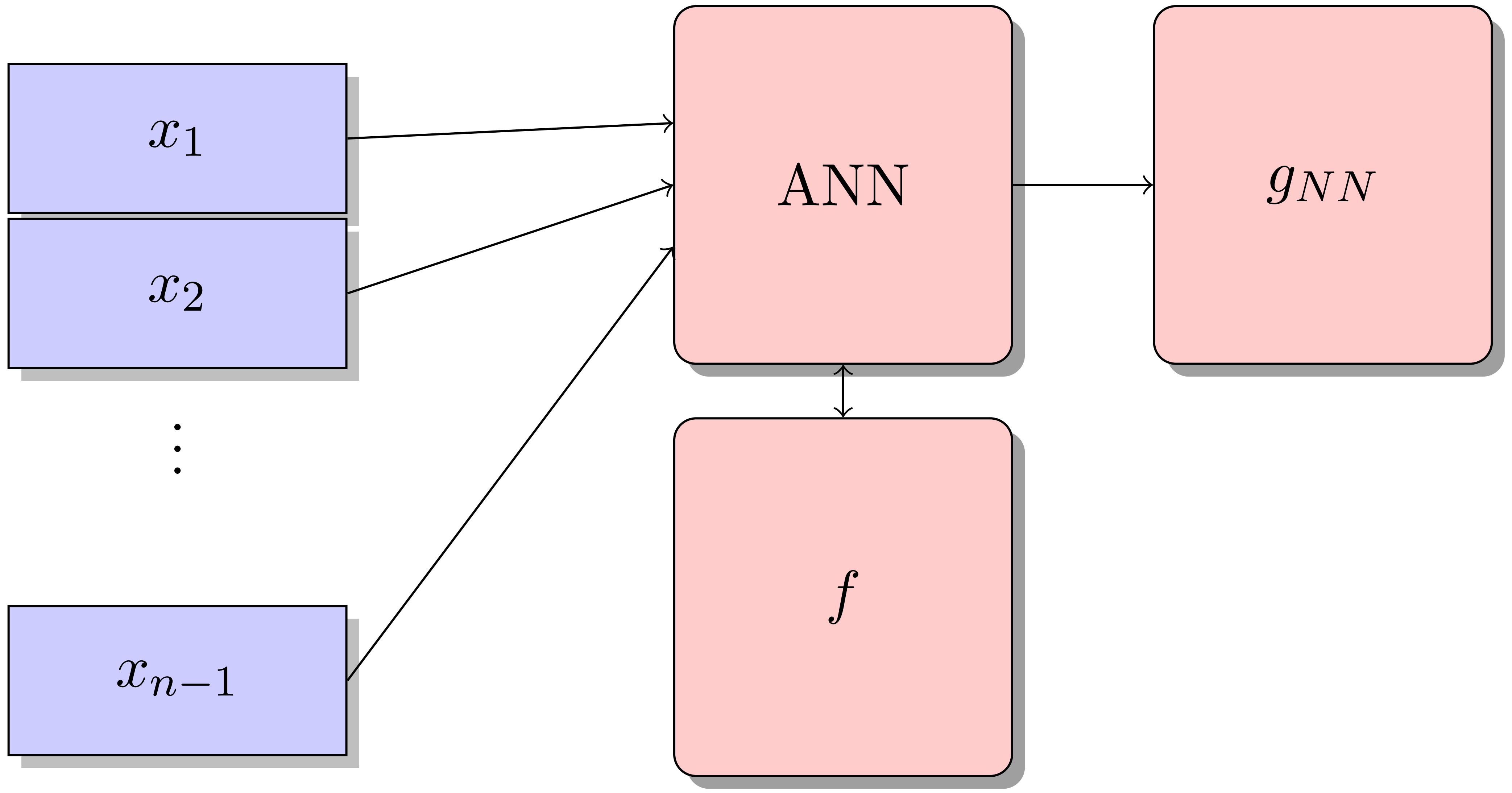
Having a neural network approximating g and keeping f as obtained from the constraint on differential volumes.

Comments on Ricci-Flow

Consider the metric Ansatz

$$g(\lambda) = g(0) + (1 - e^{-\lambda})g_{NN}$$

With g_{NN} a neural network approximation metric, λ proportional to the number of training epochs and $g(0)$ a Kähler metric. In addition to the new loss function that would replace the sigma loss one must include the Kähler as well as the patch-matching losses.



Drawbacks:

-Away from 0, the metrics are not strictly Kähler. In fact we observe that as training evolves, f starts diverging at some points.

-Flow can be corrected adding an extra penalty for increasing f.

Final Remarks

-A need for interpretability: Can we deduce analytic expressions for the Calabi-Yau metrics?

-As symmetry learning hints to the best architecture/activation function. Can we use symmetries for architecture selection?

-The structure of our approach suggests that it can be extended to other Calabi-Yau manifolds, in particular: CICYs, toric constructions. See Fabian's talk


Xquixhe pe laatu’
/ Muchas Gracias