Machine learning in CY geometry

Strings 2024

CERN, June 5

Magdalena Larfors, Uppsala University

Based on collaborations with A. Lukas, F. Ruehle, R. Schneider (2111.01436, 2205.13408)
L. Anderson, J. Gray (2312.17125)
Y. Hendi, M. Walden (2406.----)

Why use machine learning?

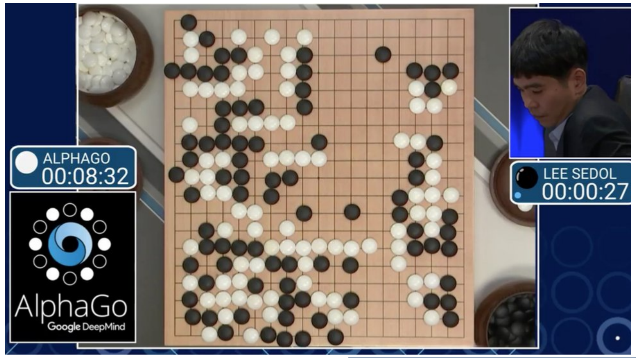
It works.

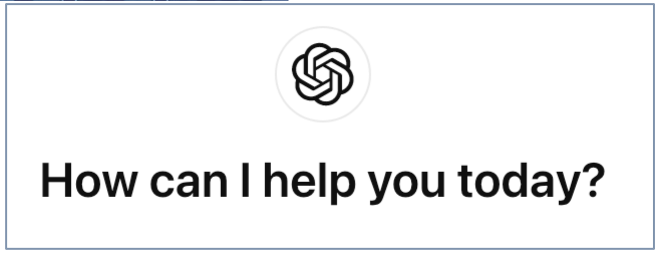
- Automate tasks
- Solve hard problems

Recent successes driven by

- better software (neural nets, optimizers)
- better hardware (GPUs)
- more data (… and more money/energy for training)
- user-friendly ML libraries (TensorFlow, JAX, PyTorch,…)

Label	Prediction
Cat	0.98
Dog	0.02
Cow	0.00





Why use ML in string theory?

- Build string vacuum with {Standard Model, dS, scale separation, ..}
  - Can ML pick good geometries? Speed up hard computations? Find vacua?
- Swampland program
  - Can ML help classify UV-complete effective field theories?
- Numerics: ML for conformal bootstrap, ML of CY metrics
- Learn mathematical structures (perhaps of relevance for physics)
- Physics-inspired models to explain how ML works

… *progress on all of these topics, driven by many researchers*

Reviews: Ruehle:20, Bao, He, Heyes, Hirst:22, Anderson, Gray, ML:23

CY geometry: Ricci flat metrics

CY Theorem: Let X be an n-dimensional compact, complex, Kähler manifold with vanishing first Chern class.
Then in any Kähler class $[J]$, X admits a unique Ricci flat metric g_{CY}.

Calabi:54, Yau:78

- For n>2, *no analytical expression* for g_{CY}.

K3: Kachru-Tripathy-Zimet:18

- Solve $R_{ij}(g) = 0$ — 4^{th} order, non-linear PDE. Very hard.
- Equivalent to — 2^{nd} order PDE for function ϕ.
  Hard, but may solve numerically on examples

CY geometry: Ricci flat metrics

CY Theorem: Let X be an n-dimensional compact, complex, Kähler manifold with vanishing first Chern class.
Then in any Kähler class $[J]$, X admits a unique Ricci flat metric g_{CY}.

Kähler form J_{CY} satisfies

- $J_{CY} = J + \partial\bar{\partial}\phi$ — same Kähler class; ϕ is a function
- $J_{CY} \wedge J_{CY} \wedge J_{CY} = \kappa\, \Omega \wedge \bar{\Omega}$ — Monge-Ampere equation (κ constant)
  2nd order PDE for ϕ

- Sample points on CY; compute J, Ω, κ; solve MA eq numerically

Numerical CY metrics

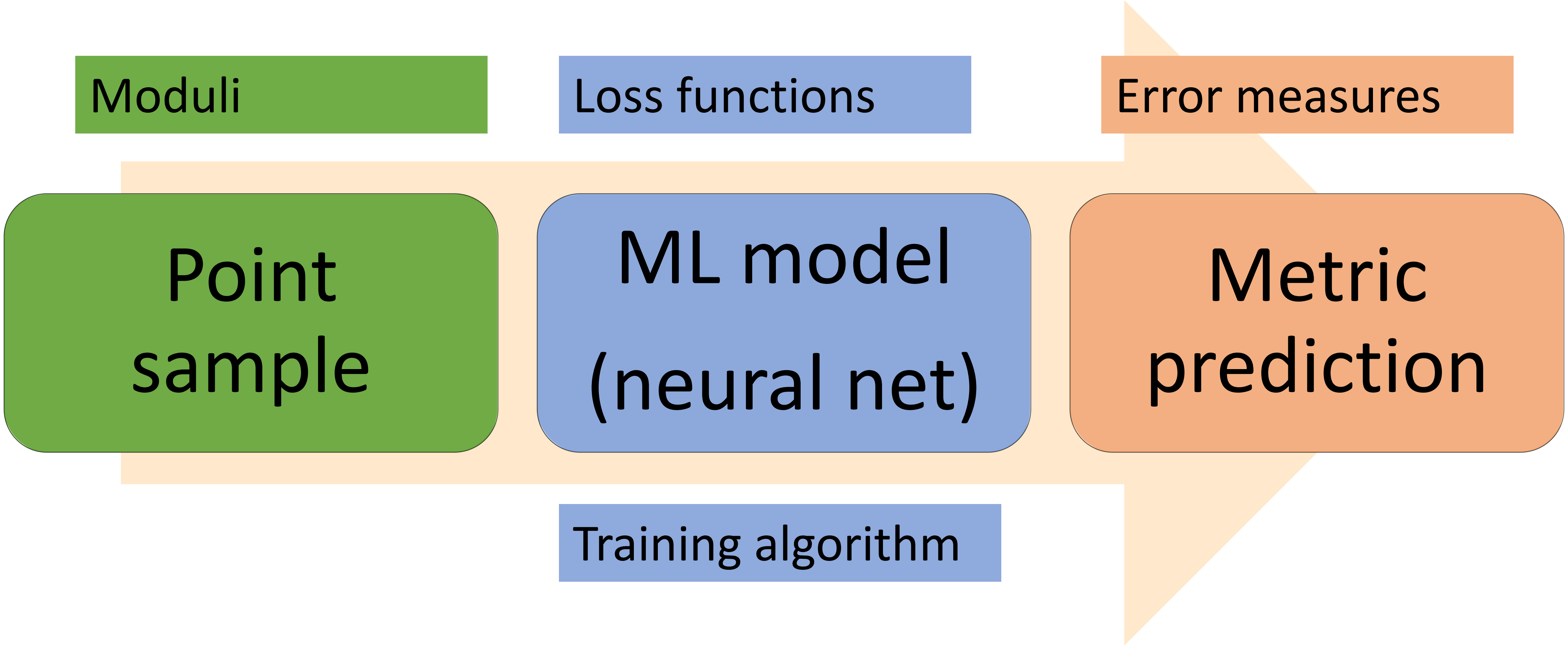
Algebraic CY metrics

- $K_k(z,\bar{z}) = \frac{1}{k}\sum \ln H_{a\bar{b}} p^a \bar{p}^{\bar{b}}$
  spectral basis of polynomials
- Solve for $H_{a\bar{b}}$ using
  - Donaldson algorithm
    Donaldson:05, Douglas-et.al:06,
    Douglas-et.al:08, Braun-et.al:08,
    Anderson-et.al:10, ...
  - Functional minimization
    Headrick–Nassar:13, Cui–Gray:20,
    Ashmore–Calmon–He–Ovrut:21
  - … or machine learning

Machine Learning CY metrics

- Neural Networks are universal approximators
  Cybenko:89, Hornik:91,
  Leshno et.al:93, Pinkus:99
- Train ML model to approximate CY metric, or Kähler potential
  Ashmore–He–Ovrut:19,
  Douglas–Lakshminarasimhan–Qi:20,
  Anderson–et.al:20,
  Jejjala–Mayorga–Pena:20 ,
  ML-Lukas-Ruehle-Schneider:21, 22
  Ashmore–Calmon–He–Ovrut:21,22,
  Berglund-et.al:22 ,
  Gerdes–Krippendorf:22, ...

Machine Learning implementation



1. Generating a point sample

On example CY need random set of points, sampled w.r.t. known measure

Leading algorithm: CY is hypersurface in $\mathbb{P}^n$ Douglas et. al: 06

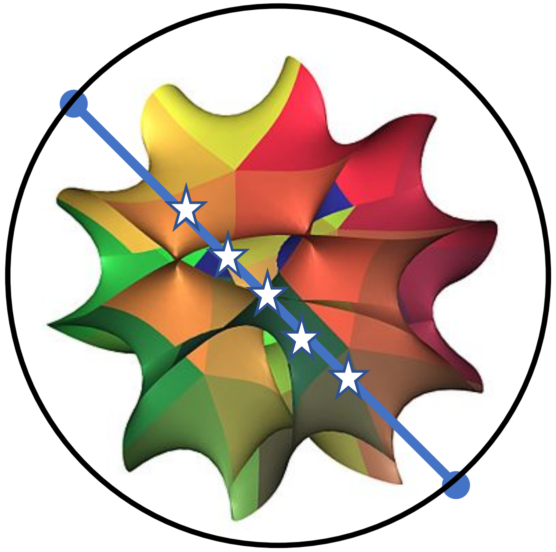
- Sample 2 pts on $\mathbb{P}^n$, connect with line & intersect → $n+1$ pts
- Shiffman-Zelditch theorem: distributed w.r.t. $dvol_{FS}$

Generalizes to CICYs and CYs from Kreuzer-Skarke list

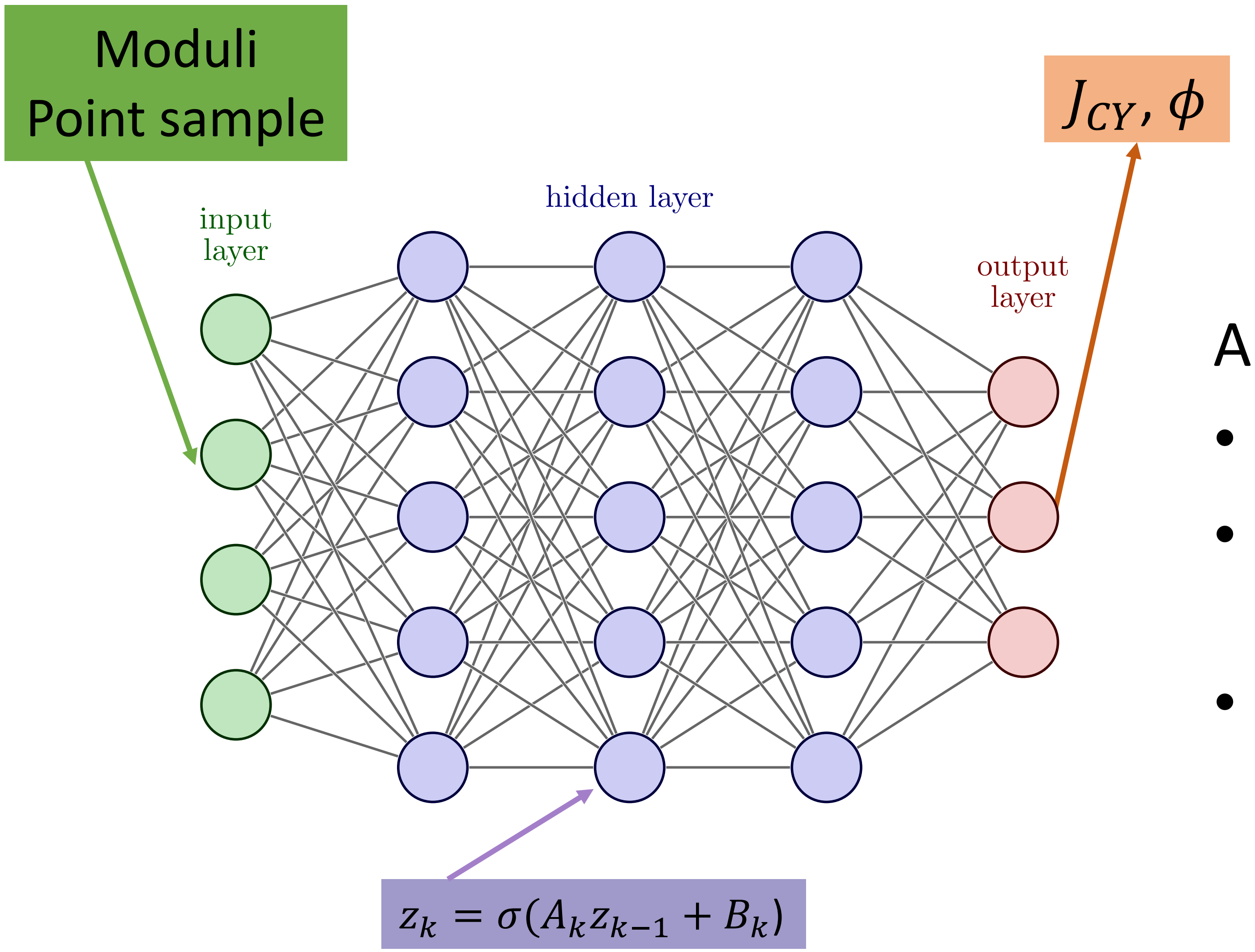
Douglas et.al: 07, ML, Lukas, Ruehle, Schneider: 21,22

- Fast point generators of ML packages

MLgeometry, cymetric, cyjax



2. Setting up the ML model



Architectural choices

- What to predict?
- Encode constraints in NN or loss? (global, complex, Kähler…)
- Flexibility vs. precision

ML models - choice of architecture

1. Learn metric
   Anderson-et.al.:20,
   Jejjala–Mayorga–Pena:20
   ML-Lukas-Ruehle-Schneider:21, 22
2. Learn Kähler potential (ϕ)
   Anderson-et.al.:20,
   Douglas–Lakshminarasimhan–Qi:20,
   Ashmore–Calmon–He–Ovrut:21,22,
   ML-Lukas-Ruehle-Schneider:21, 22,
   Berglund-et.al.:22
3. Learn Donaldson's H matrix
   Anderson-et.al.:20,
   Gerdes–Krippendorf:22

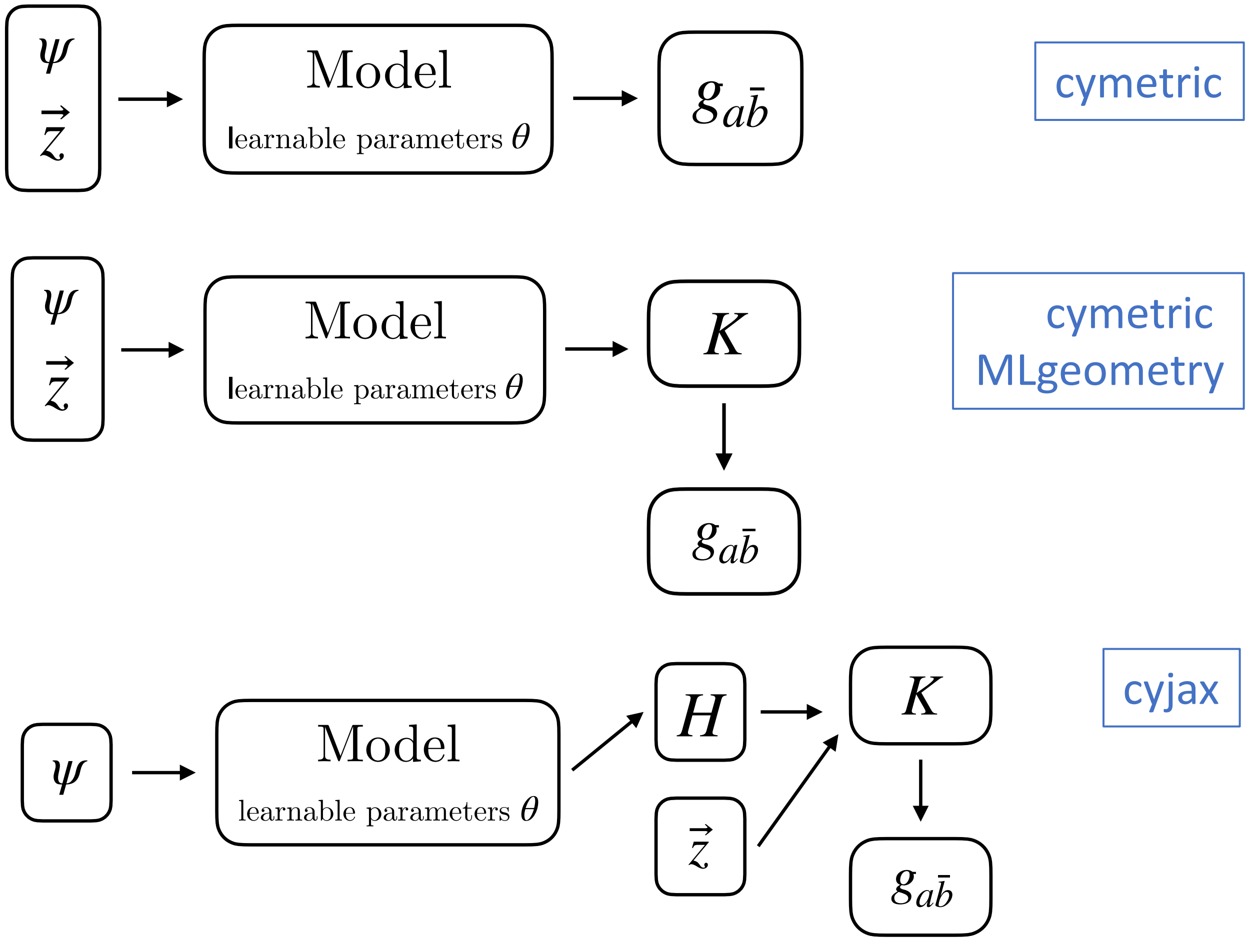
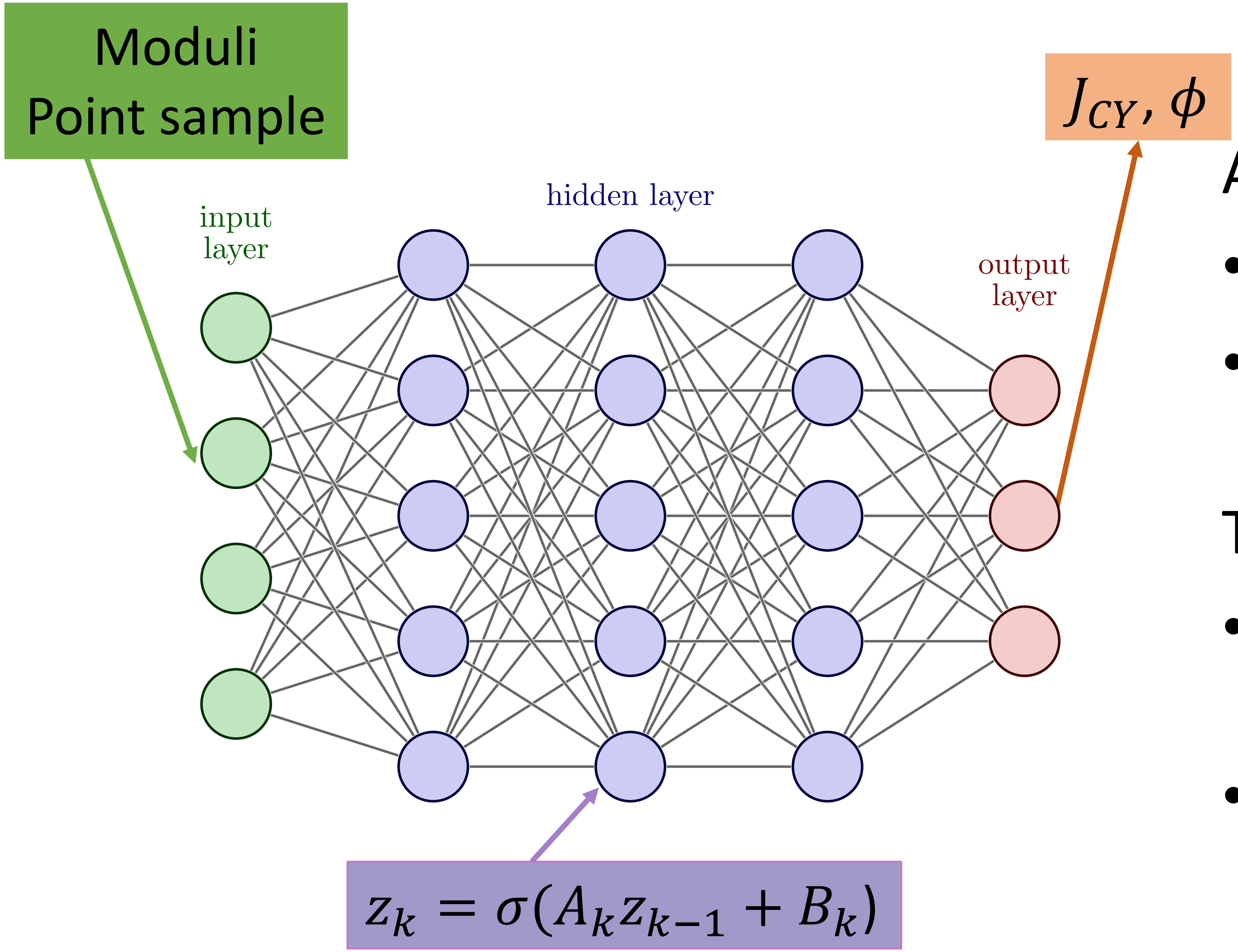


Figure adapted from Anderson et al:20

3. Train the ML model



Architectural choices

- What to predict?
- Encode constraints in NN or loss?

Then train

- Adapt layer weights to minimize loss functions
- Stochastic gradient descent

Loss functions encode math constraints

- Train the network to get unknown Ricci-flat metric (in given Kähler class)
- Use semi-supervised learning
    1. Encode mathematical constraints as custom loss functions
    2. Train network (adapt layer weights) to minimize loss functions
- Satisfy Monge-Ampere eq → minimize Monge-Ampere loss

$$\mathcal{L}_{\text{MA}} = \left\| 1 - \frac{1}{\kappa} \frac{\det g_{\text{pr}}}{\Omega \wedge \bar{\Omega}} \right\|_n$$

- Less rigid metric ansatz → more loss functions (Kähler, transition)

4. Check accuracy

- After training, check that MA eq holds and Ricci tensor is zero

Check via established benchmarks:

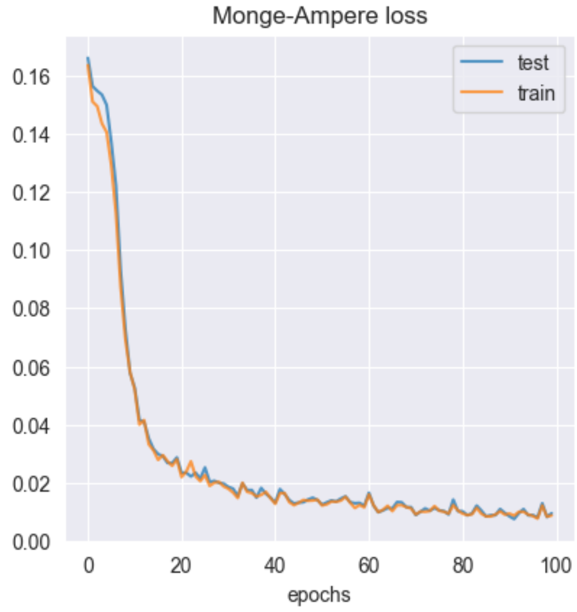
$$\sigma = \frac{1}{\mathrm{Vol}_{\mathsf{CY}}} \int_X \left| 1 - \kappa \, \frac{\Omega \wedge \overline{\Omega}}{(J_{\mathsf{pr}})^3} \right| \; , \; \mathcal{R} = \frac{1}{\mathrm{Vol}_{\mathsf{CY}}} \int_X |R_{\mathsf{pr}}| \; .$$

- For CY manifolds with more than one Kähler class, checks of volume and line bundle slopes ensures this stays fixed.

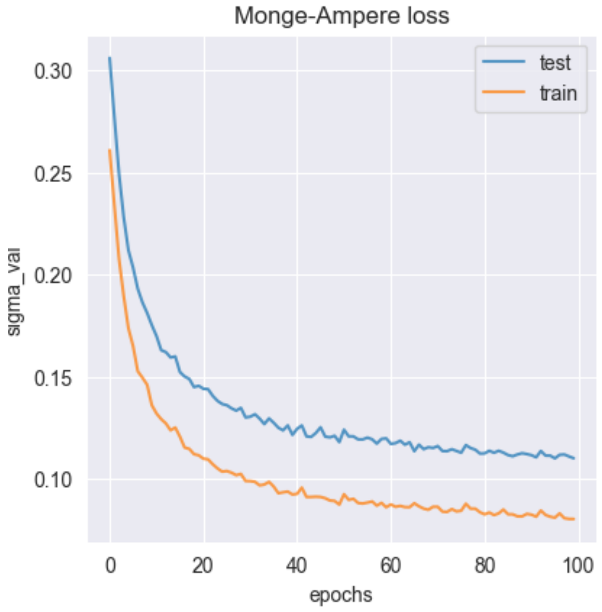
Experiments: Fermat vs. generic quintic

Anderson, Gray, ML:23

Monge-Ampere loss

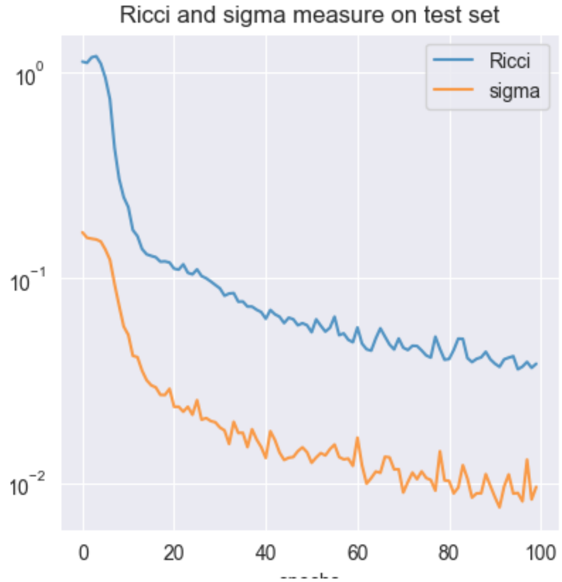


Fermat

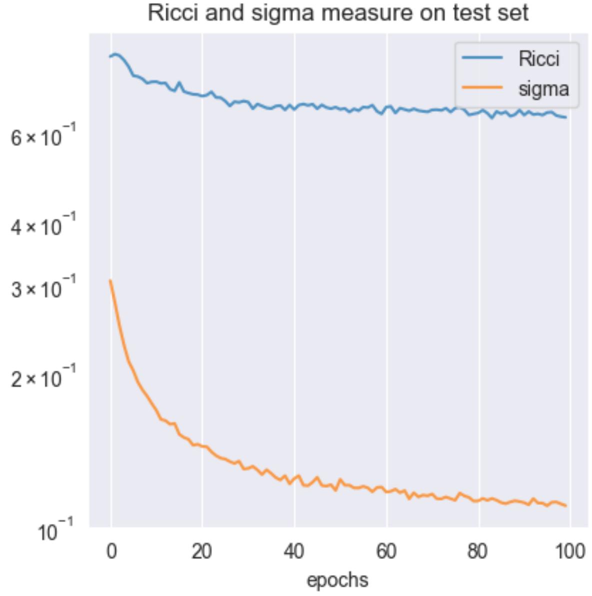


Generic

Error measures



Fermat



Generic

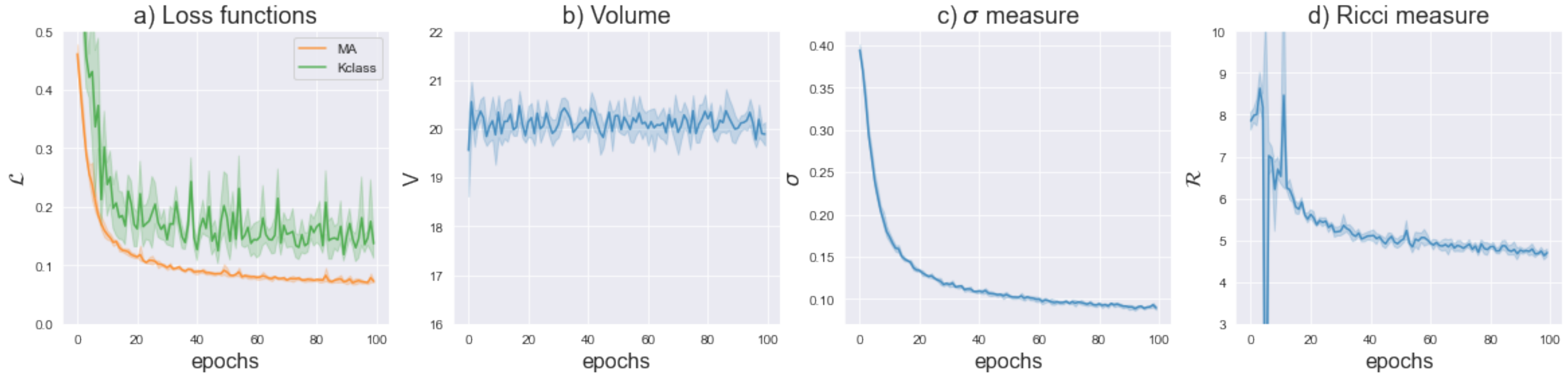
Cymetric, 100 000 points, ϕ model, 3 64-node layers, GELU, default loss parameters, Adam, batch (64, 50000)

ML methods are less sensitive to symmetry

Experiments: KS CY example

ML,Lukas, Ruehle,Schneider:22

- $h^{1,1} = 2, h^{2,1} = 80$ hypersurface from Kreuzer-Skarke database



Toric ϕ-model, default loss, 200 000 points

NN width 256, depth 3, GELU, batch (128, 10000), SGD w. momentum

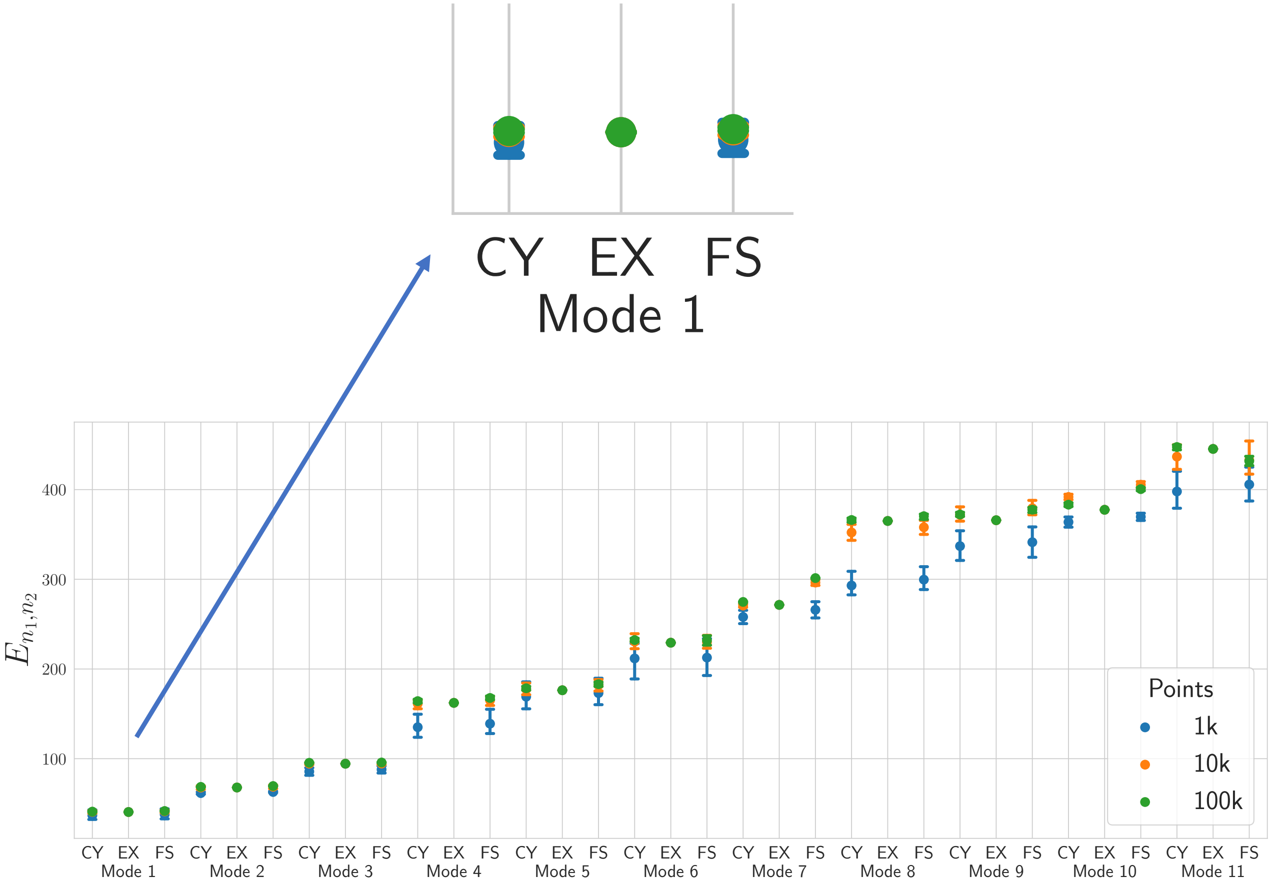
ML methods work on both CICY and KS CYs

Accuracy and benchmarks

Ahmed & Ruehle:23

Improve accuracy

- Larger point sample
- Wider/deeper NN
- Train longer



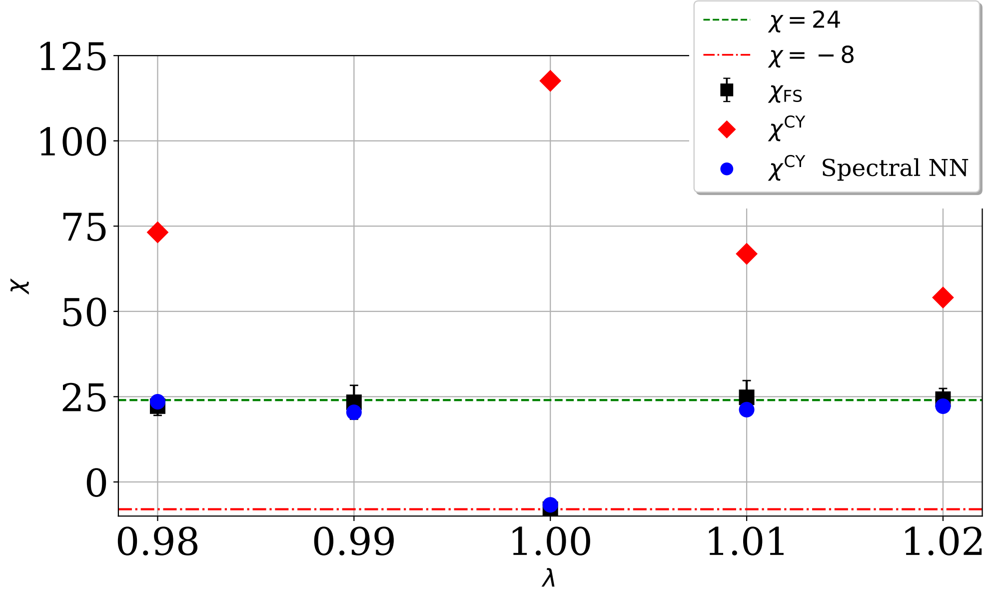
- Benchmark cymetric cubic CY in $\mathbb{P}^2$ (a.k.a. the torus)
- Spectrum of Δ_{CY}

Accuracy, performance and architecture

- Is the control by loss functions enough?

ML models which always give global ϕ

- Algebraic metric, using spectral basis
  Anderson et al : 20, Douglas et al : 20, Gerdes & Krippendorf:22, ...
- Combining cymetric with “spectral layer” improves accuracy and performance
  Berglund et al:22



Berglund et al:22 CY 2-fold; singular at 1

$$(z_0, \ldots, z_n) \mapsto \begin{pmatrix} \frac{z_0\bar{z}_0}{|z|^2} & \frac{z_0\bar{z}_1}{|z|^2} & \cdots & \frac{z_0\bar{z}_n}{|z|^2} \\ \frac{z_1\bar{z}_0}{|z|^2} & \frac{z_1\bar{z}_1}{|z|^2} & \cdots & \frac{z_1\bar{z}_n}{|z|^2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{z_n\bar{z}_0}{|z|^2} & \frac{z_n\bar{z}_1}{|z|^2} & \cdots & \frac{z_n\bar{z}_n}{|z|^2} \end{pmatrix}$$

ML G-invariant CY metrics

Hendi, ML, Walden:24 (work in progress)

- Let X be smooth CY, G discrete symmetry w.o fixed points
  Want: Ricci-flat metric on X/G
- Traditional approach: restrict spectral basis to invariant polynomials

Douglas et al:08, … Butbaia et al:24

Alternative: design G-invariant ML model $\phi(g \cdot z) = \phi(z)$

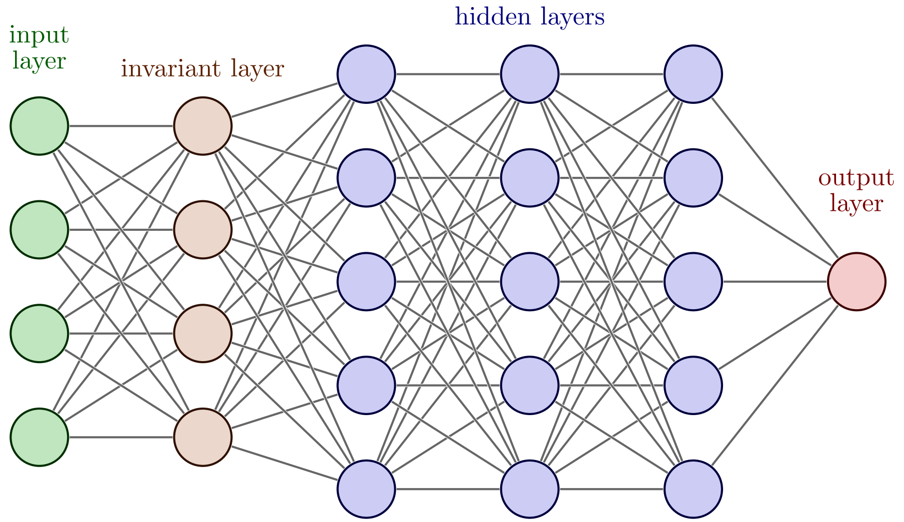
- Geometric Deep Learning: symmetry & performance
  Bronstein et al:17,21,..
- Universal approximator theorem for invariant NNs Yarotsky:22,..
- Invariance can be imposed in several ways in ML
  In NN, just need one invariant layer

$$\phi(z) = \phi(\sigma(A_k(\ldots \sigma(A_1(InvLay(z)\ \ldots)$$

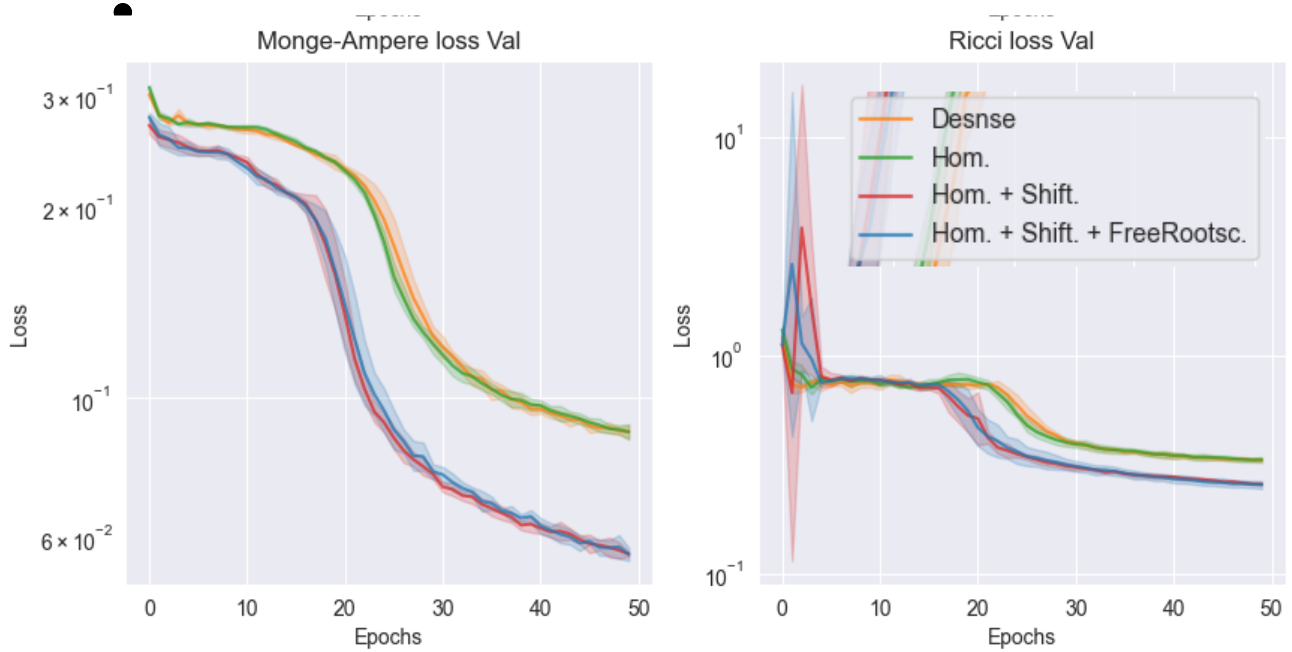
CY metric on smooth quintic quotient

Hendi, ML, Walden: 24 (work in progress)

- Ricci-flat metric on $\frac{X}{G}$
- ϕ-model of cymetric with non-trainable layer



- Invariant layer projects data to fundamental domain of G
  Aslan, Platt, Sheard:22, Kaba et.al. 23
-



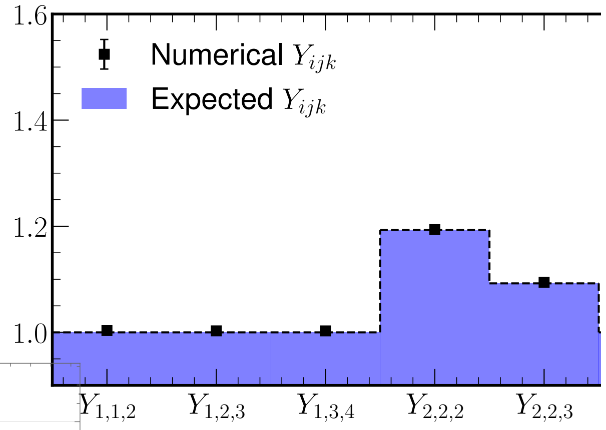
Applications

Physical Yukawa couplings

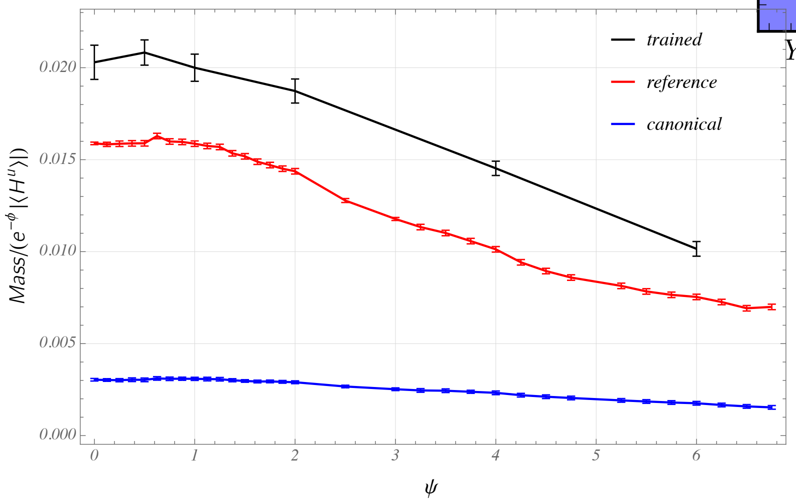
- Heterotic string: matter fields come from gauge bundle
- In “standard embedding” models, physical Yukawa couplings known
  Strominger:85,Greene,et.al.86,87, Candelas:88, Distler, Greene:88,…
- Not true for other gauge choices
- Use ML to compute
  - Ricci-flat CY metric
  - HYM connection
  - Harmonic representatives

Butbaia, Mayorga-Pena, Tan, Berglund, Hubsch, Jejjala, Mishra :24

Constantin, Fraser-Taliente, Harvey, Lukas, Ovrut:24



Standard embedding



Line bundle sum

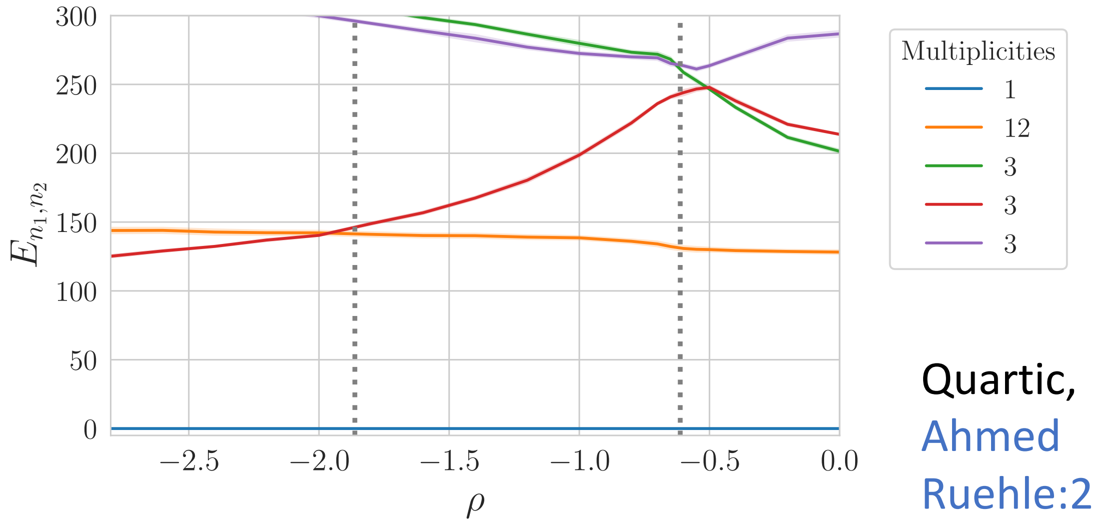
Test swampland distance conjecture

Ashmore:20, Ashmore & Ruehle:21 Ahmed & Ruehle:23

- Compute moduli-dependent spectrum of Δ_{CY} in example CY:s

1. Compute the moduli space metric (using either analytic [20] or numeric [21] techniques)
2. Compute the geodesics and the geodesic distances in moduli space
3. Compute the CY metric along the moduli space geodesics
4. Compute the massive spectrum from the CY metric
5. Fit a function to the masses and compare with the prediction from the SDC

- Level crossing & number theory



Quartic, Ahmed Ruehle:23

Conclusion and outlook

- ML models learn Ricci flat metrics on CICY and KS CY manifolds.
- Mathematical constraints: encoded in NN or in loss functions
- Performant ML packages: cymetric, MLgeometry, cyjax
- Architecture determines accuracy, performance, generality
- Physics applications:
  - Yukawa couplings Butbaia-et.al:24, Constantin-et.al:24
  - Swampland distance conjecture, Ashmore:20, Ashmore & Ruehle:21 Ahmed & Ruehle:23

Outlook:

- Moduli-dependent CY metrics Anderson-et.al:20, Gerdes-Krippendorf:22
- Beyond CY: G2 metrics, G-structure manifolds, ...

Thank you for listening!