Machine learning in CY geometry

String Pheno 2024

Padova, June 26

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 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? Or test conjectures?
- Numerics: ML for conformal bootstrap, ML of CY metrics
- Learn mathematical structures (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

- Talks by Stefano Lanza, Kit Fraser-Taliente, Justin Tan, Lucas Tsun Yin Leung,…

This talk in a nutshell

- ML is useful for CY geometry
  - Ricci flat metrics
  - Hermitian Yang-Mills connections
  - Harmonic representatives

See talks by Kit Fraser-Taliente and Justin Tan

- ML methods work for
  - generic CY manifolds
  - at given point in moduli space
- Symmetries can be encoded using invariant ML models

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>1, *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

- $K_k(z,\bar{z}) = \frac{1}{k}\sum \ln H_{a\bar{b}} p^a \bar{p}^{\bar{b}}$
  p^a spectral basis (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
    Anderson–et.al:20, Gerdes–Krippendorf:22,

Machine Learning CY metrics

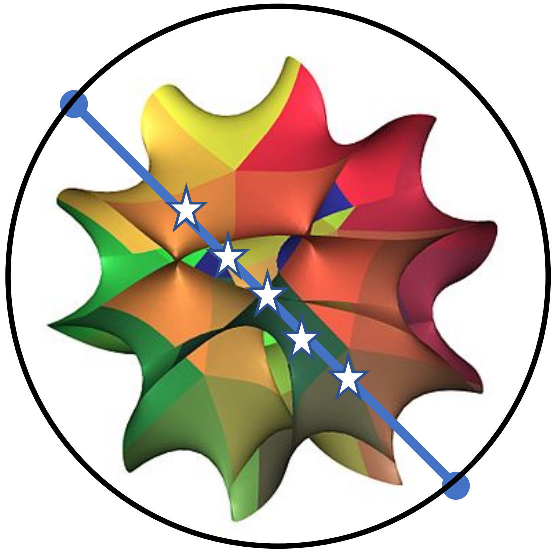
- Neural Networks are universal approximators
  Cybenko:89, Hornik:91, Leshno et.al:93, Pinkus:99,…
- Train ML model to approximate CY metric, or Kähler potential
  Ashmore–He–Ovrut:19, Douglas–Lakshminarasimhan–Qi:20, Anderson–et.al:20, Jejjala–Mayorga–Pena:20 , ML-Lukas-Ruehle-Schneider:21, 22 Ashmore–Calmon–He–Ovrut:21,22, Berglund-et.al:22 , Halverson-Ruehle:23, Douglas-Platt-Qi-24

1. Generate 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. $dvol_{FS}$



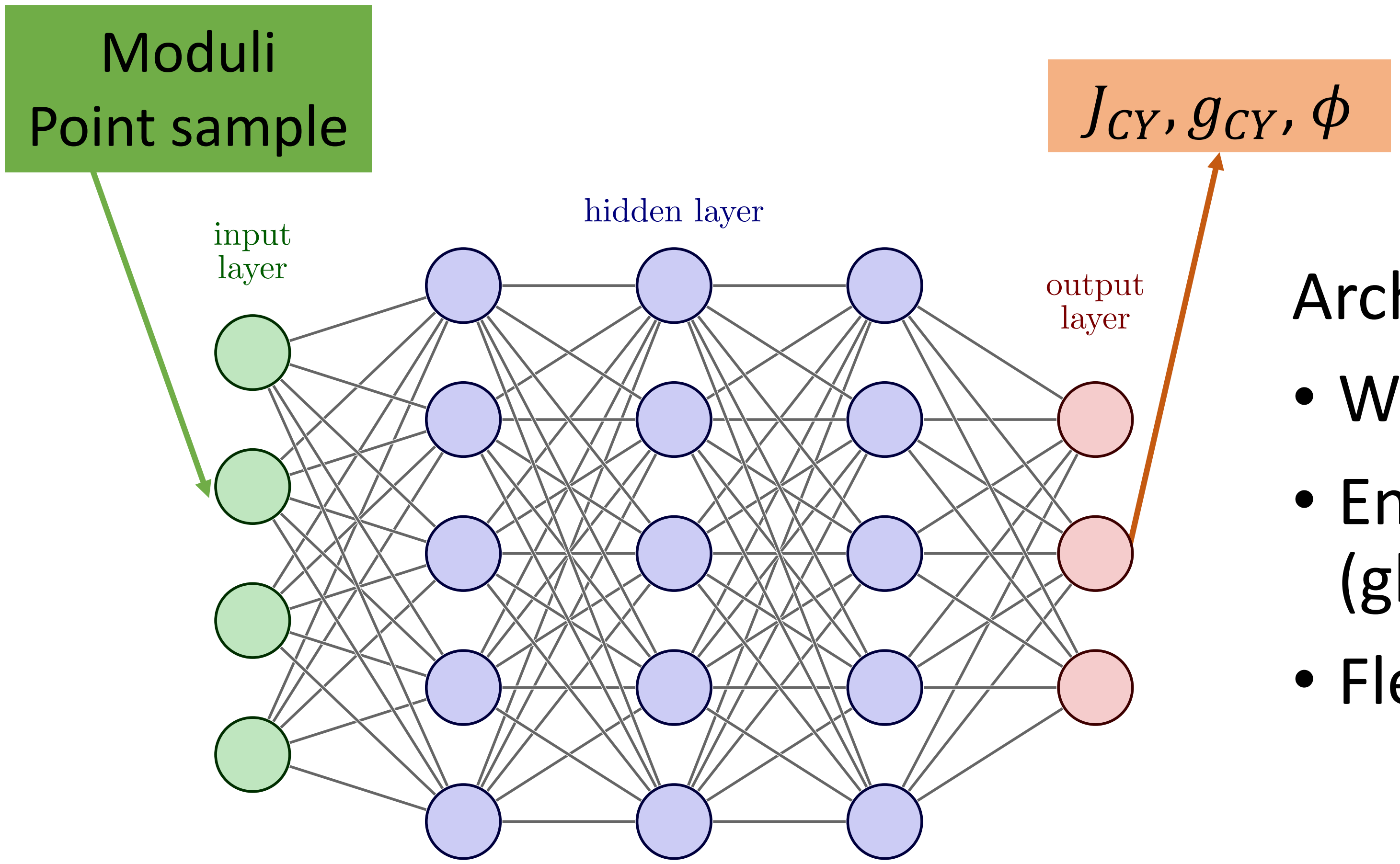
Generalizes to CICYs and CYs from Kreuzer-Skarke list

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

- Fast point generators included in ML packages

MLgeometry, cymetric, cyjax

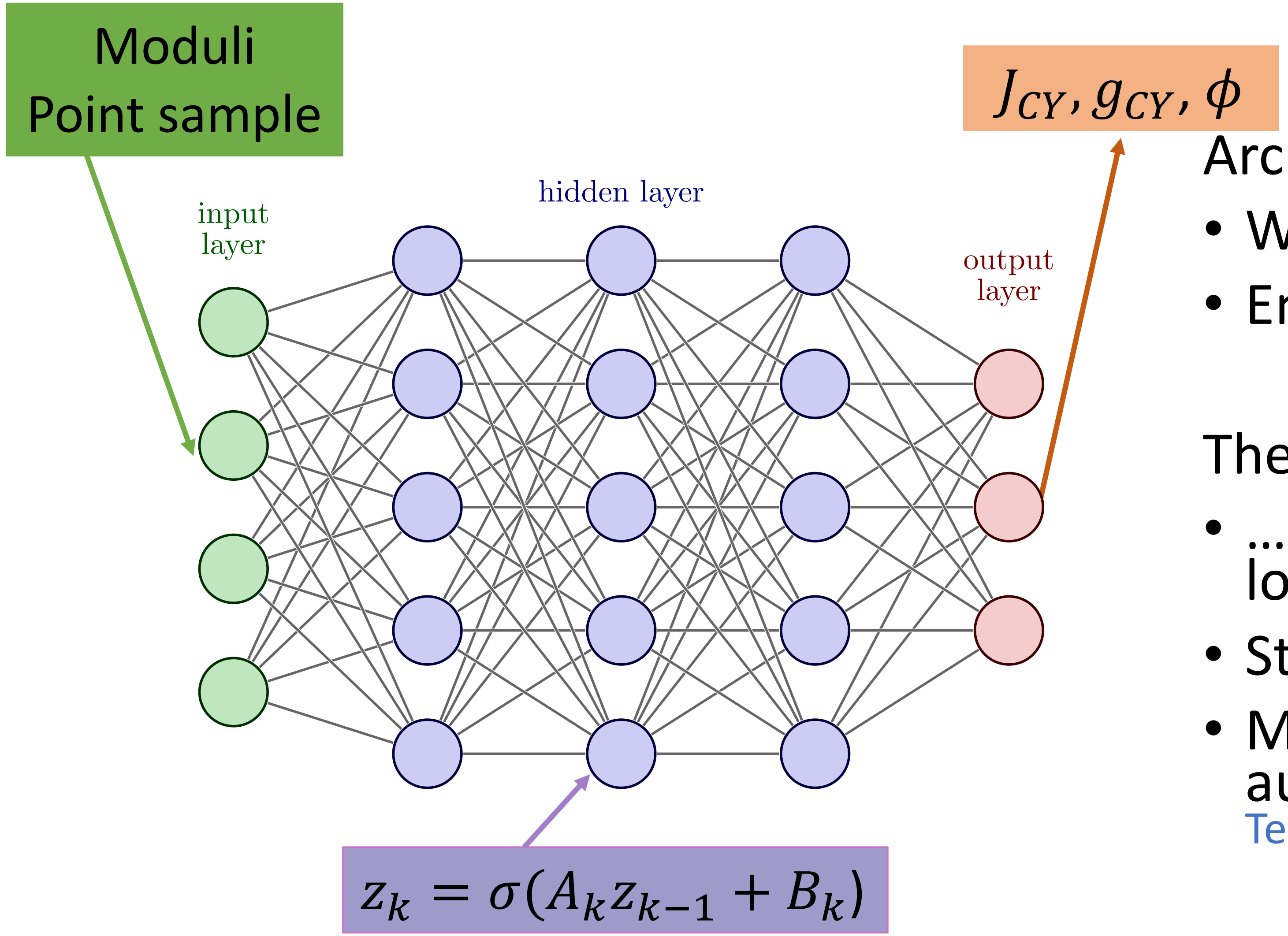
2. Set up the ML model



Architectural choices

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

3. Train the ML model



Architectural choices

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

Then train

- … i.e. adapt layer weights to minimize loss functions
- Stochastic gradient descent
- ML libraries w. highly optimized automatic differentiation
  TensorFlow, JAX, PyTorch

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

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

- Free metric ansatz → more loss functions (Kähler, transition, K-class)

4. Check accuracy

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

Check via established benchmarks:

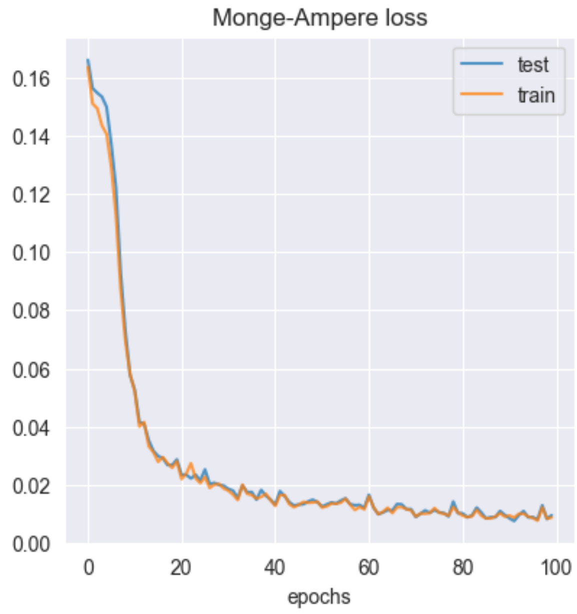
$$\sigma = \frac{1}{\mathrm{Vol}_{\mathsf{CY}}} \int_X \left| 1 - \kappa \, \frac{\Omega \wedge \overline{\Omega}}{(J_{\mathsf{pr}})^3} \right| \, , \; \mathcal{R} = \frac{1}{\mathrm{Vol}_{\mathsf{CY}}} \int_X |R_{\mathsf{pr}}| \, .$$

- Checking topological quantities, like volume and line bundle slopes, ensures metric prediction has good global properties .

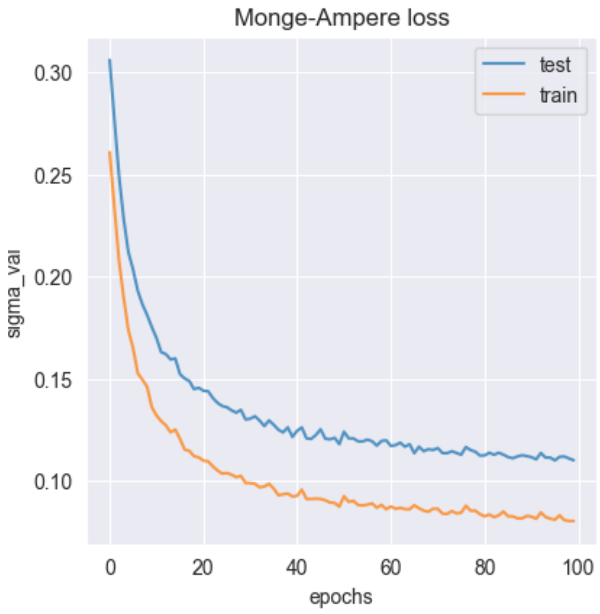
Experiments: Fermat vs. generic quintic

Anderson, Gray, ML:23

Monge-Ampere loss

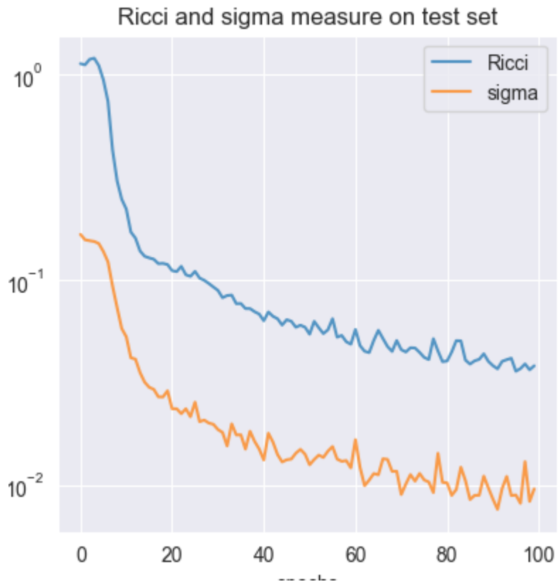


Fermat

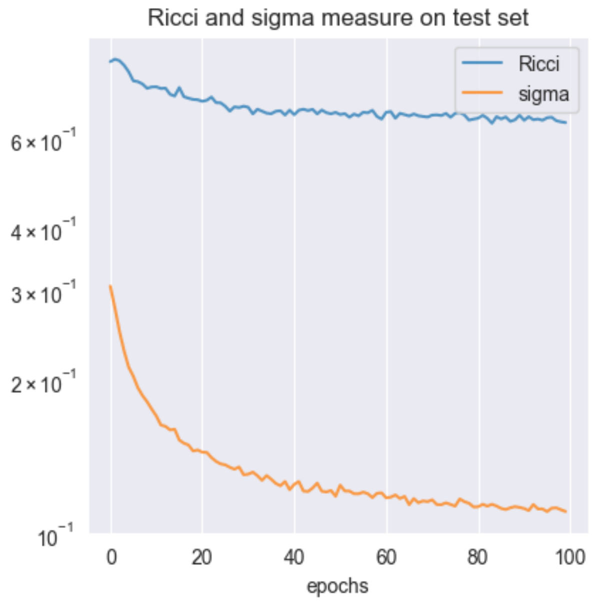


Generic

Error measures



Fermat



Generic

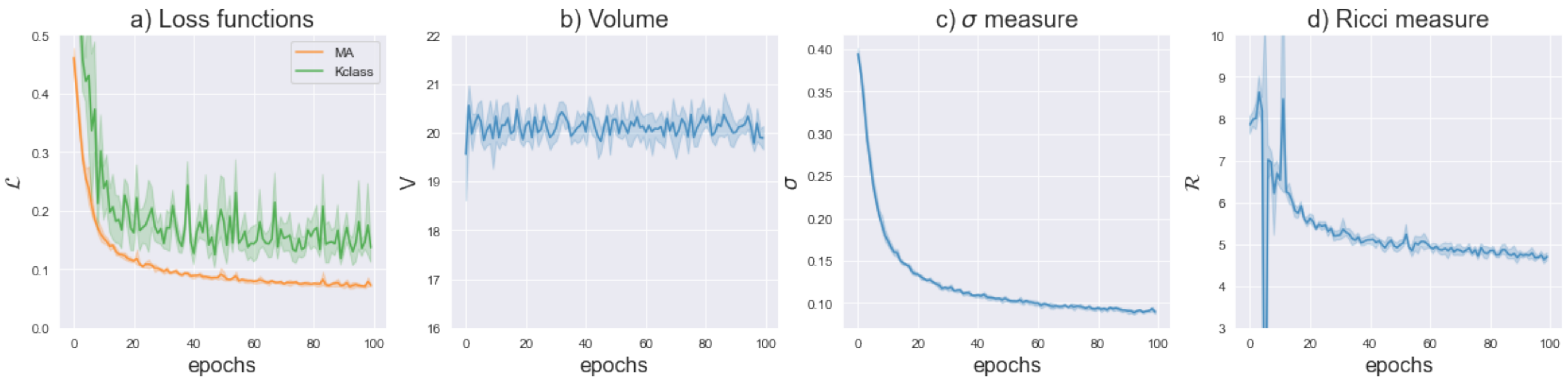
cymetric, ϕ-model, 100 000 points, 3 64-node layers, GELU, default loss parameters, Adam, batch (64, 50000)

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



cymetric, 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

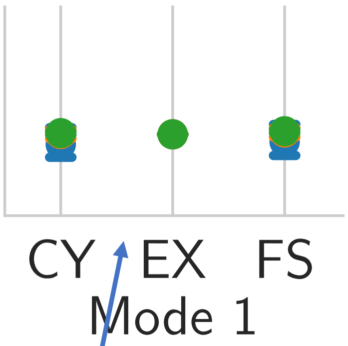
Ahmed & Ruehle:23

Accuracy improves with

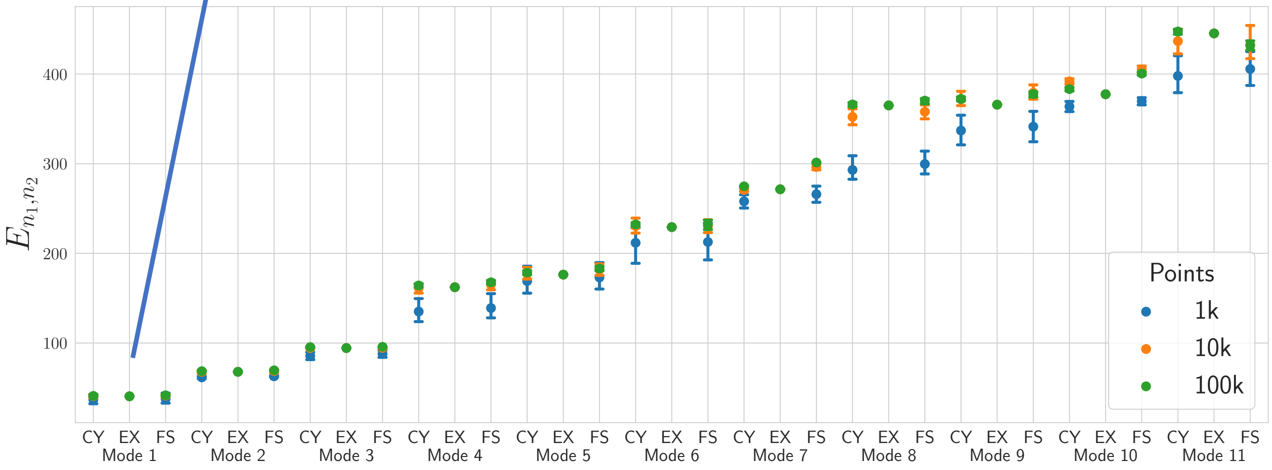
- Larger point sample
- Wider/deeper NN
- Train longer

and/or

- Change model architecture
  - cymetric ϕ-model + Spectral Layer
    Berglund et al:22, Butbaia et al:24



Benchmark cymetric
on the torus
Spectrum of Δ_{CY}



Application: Heterotic Standard-Like Models

Building blocks

- Ricci-flat Calabi Yau manifold X
- Vector bundle V satisfying Hermitian Yang-Mills eq.
- Discrete symmetry group G (break GUT)

- Many examples! E.g. 35 000 SLMs found with $V = \bigoplus L_i$
  Anderson et.al:11,12,13, …
  with RL/gen.alg. ML-Schneider:20, Constantin et.al: 21, Abel et al:21,23,…
- Next step toward SM: compute normalized Yukawa couplings
  Butbaia-et.al:24, Constantin-et.al:24

See talks by Kit Fraser-Taliente and Justin Tan

Application: Heterotic Standard-Like Models

Building blocks

- Ricci-flat Calabi Yau manifold X
- Vector bundle satisfying HYM eq.
- Discrete symmetry $G \rightsquigarrow$ smooth quotient CY X/G
  - allows to break GUT using Wilson lines
  - symmetries: permutations, discrete phase rotations, shifts of input z_i

- How can we restrict the model prediction to group invariant metrics?
- Can we use ML for Ricci flat metric on quotient CY?

ML G-invariant CY metrics

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

- Let X be smooth CY, G symmetry, $g_{CY} = g_{FS} + \partial\bar{\partial}\phi$
- ML model which approximates $\phi(z)$ is G-invariant if

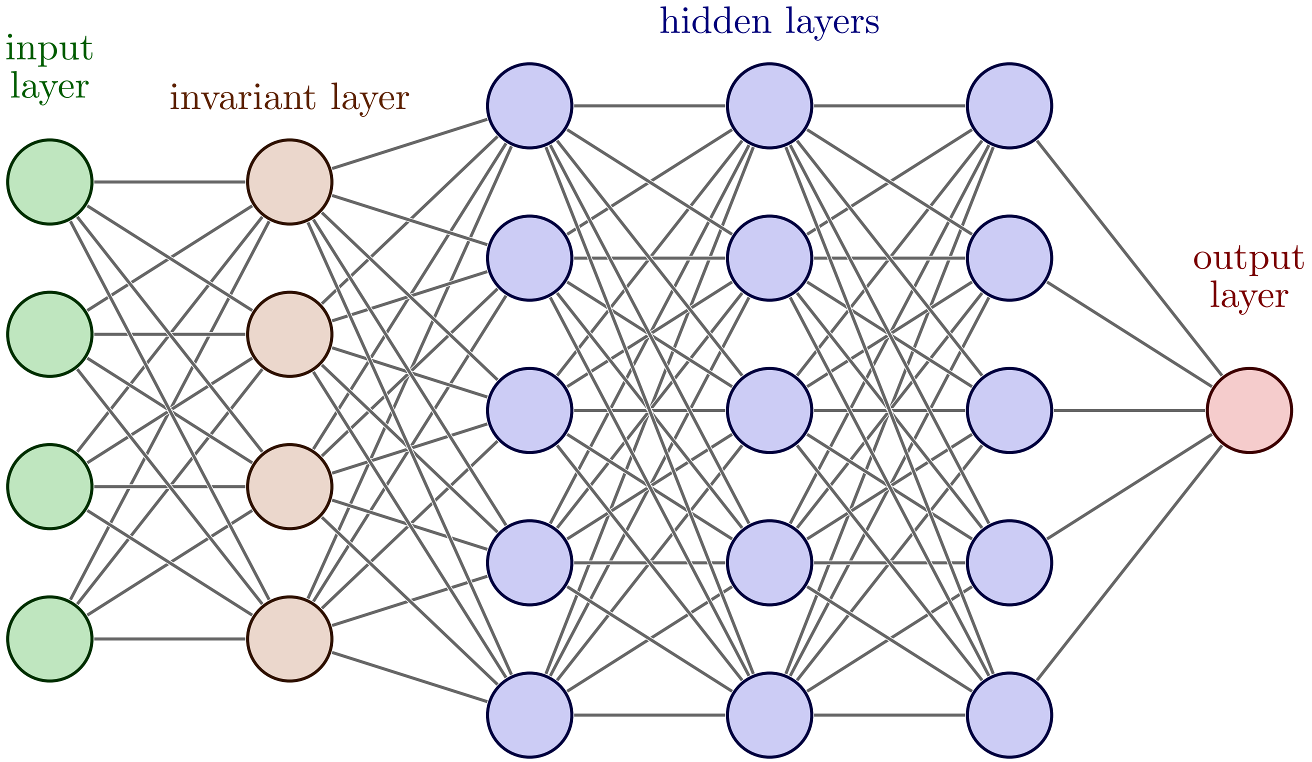
$$\phi(g \cdot z) = \phi(z)$$

- With spectral basis, ϕ invariant if expanded in invariant polynomials
  Donaldson:05, Headrick-Nassar:13, Douglas et al:08, …
- For ML, any G-invariant layer makes model invariant
  - Invariant NNs are Universal approximators for invariant functions Yarotsky:22,..
  - Invariant ML models can be constructed in many ways
  - Geometric Deep Learning: symmetry, performance & interpretability
    Bronstein et al:17,21,..

Invariance through non-trainable layers

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

- Expand input in group invariant polynomials
  Yarotsky:22
  $\rightsquigarrow$ Spectral layer
- G-canonicalization: Invariant layer projects input to fundamental domain of G
  Aslan, Platt, Sheard:22, Kaba et.al. 23

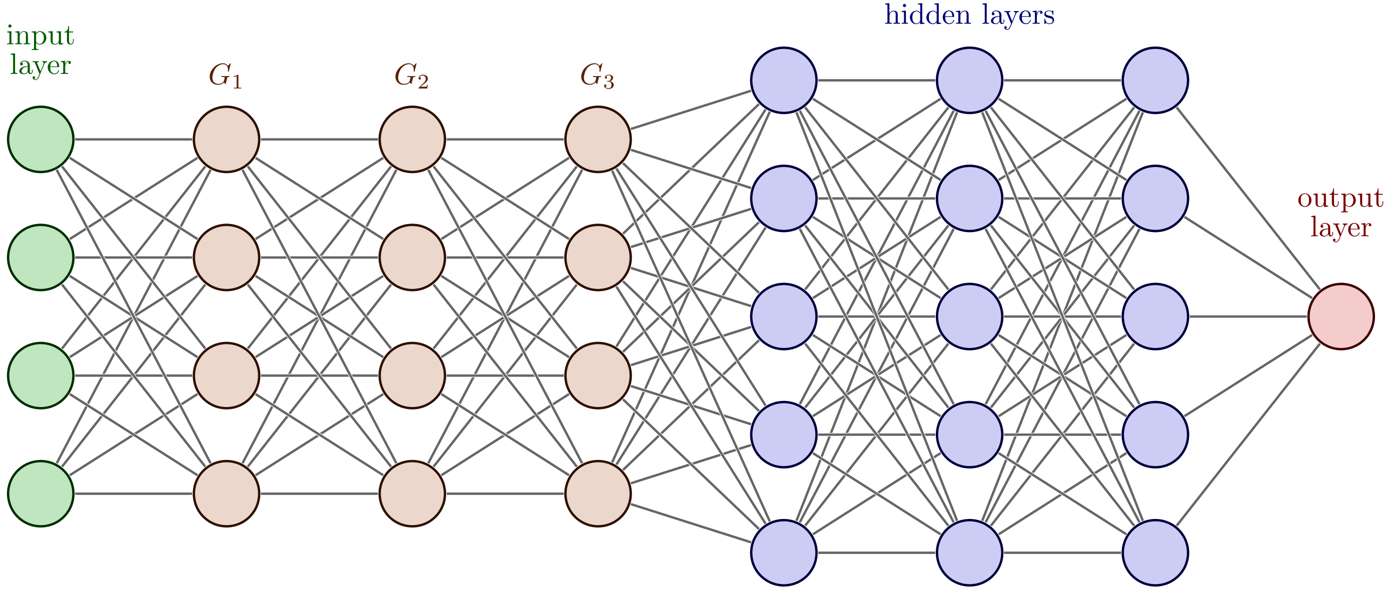


Invariance through non-trainable layers

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

G-canonicalization:

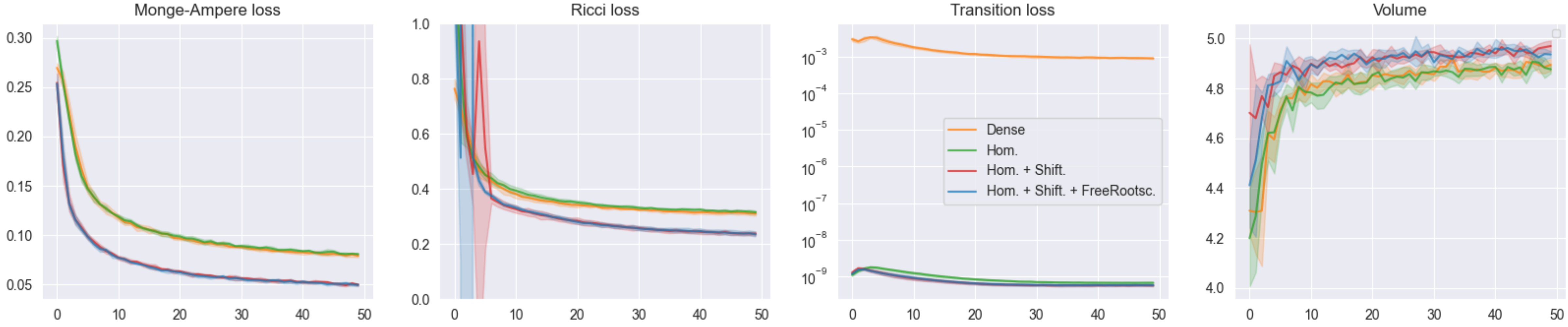
- Invariant layer projects data to fundamental domain of G
- Modular and stackable (given compatibility condition)
- Easily included in ML models for CY metrics (we use cymetric)



CY metric on smooth quintic quotient

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

- Ricci-flat metric on $X/\mathbb{Z}_2^2$
- Symmetries (shifts and phase rotation) act freely
- ϕ-model of cymetric with 2 canonicalization layers



Conclusion and outlook

- Can learn Ricci flat metrics on CICY and KS CY manifolds, for given moduli.
- Mathematical constraints: encoded in NN or in loss functions
- Performant ML packages: cymetric, MLgeometry, cyjax,…
- Architecture determines accuracy, performance, generality.

Applications and generalizations

  - Yukawa couplings Butbaia-et.al:24, Constantin-et.al:24
  - Swampland distance conjecture Ashmore:20, Ashmore & Ruehle:21 Ahmed & Ruehle:23
  - Moduli-dependent CY metrics Anderson-et.al:20, Gerdes-Krippendorf:22
  - Warped CY metrics, G-structure geometry
  - (towards) G2 metrics Douglas-Platt-Qi:24

Conclusion and outlook

- Can learn Ricci flat metrics on CICY and KS CY manifolds, for given moduli.
- Mathematical constraints: encoded in NN or in loss functions
- Performant ML packages: cymetric, MLgeometry, cyjax,…
- Architecture determines accuracy, performance, generality.

Applications and generalizations

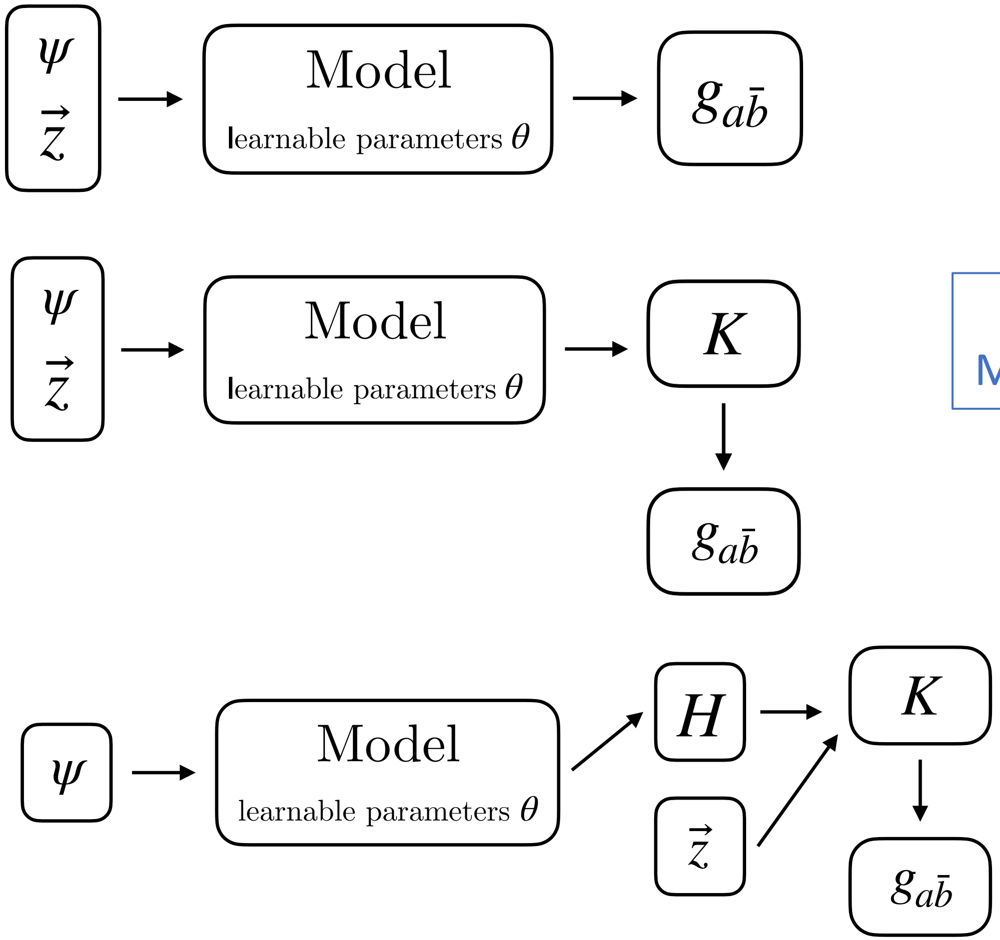
- Yukawa couplings Butbaia-et.al:24, Constantin-et.al:24
- Swampland distance conjecture Ashmore:20, Ashmore & Ruehle:21 Ahmed & Ruehle:23
- Moduli-dependent CY metrics Anderson-et.al:20, Gerdes-Krippendorf:22
- Warped CY metrics, G-structure geometry
- (towards) G2 metrics Douglas-Platt-Qi:24

Thank you for listening!

Additional slides

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



cymetric

cymetric
MLgeometry

cyjax

Figure adapted from Anderson et al:20

Homogeneous rescaling invariance

- Often work with homogeneous coordinates of ambient space
- Rescaling invariant $\phi(z) = \phi(\lambda\, z)$
- True *exactly* for algebraic metric, using spectral basis
  Anderson et al : 20, Douglas et al : 20, Gerdes & Krippendorf:22, ...
- The models of the cymetric package are only approximately invariant.
- Combining cymetric ϕ −model with "spectral layer" gives invariant model
  Berglund et al:22

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

Application: Heterotic Standard-Like Models

Building blocks

- Ricci-flat Calabi Yau manifold X
- Vector bundle V satisfying Hermitian Yang-Mills eq.

$$F \wedge \Omega = 0 = F \wedge J \wedge J$$

- Discrete symmetry (break GUT)

- Hard differential equation. Requires g_{CY}
- Solution exists iff V is polystable
  Donaldson:85, Uhlenbeck–Yau:86
- Can solve numerically…
  Douglas et.al.:06 Anderson et.al:10,11
- … with ML
  Ashmore–Deen–He–Ovrut:21
  ML-Lukas-Ruehle-Schneider:22,
  Butbaia et.al:24, Constantin et. al:24

Application: Heterotic Standard-Like Models

Building blocks

- Ricci-flat Calabi Yau manifold X
- Vector bundle V satisfying Hermitian Yang-Mills eq.

$$F \wedge \Omega = 0 = F \wedge J \wedge J$$

- Discrete symmetry (break GUT)

- Hard differential equation. Requires g_{CY}
- Solution exists iff V is polystable
  Donaldson:85, Uhlenbeck–Yau:86
- Can solve numerically
  Douglas et.al.:06 Anderson et.al:10,11,
  Ashmore–Deen–He–Ovrut:21
  ML-Lukas-Ruehle-Schneider:22
  Butbaia et.al:24, Constantin et. al:24

- Next step: compute normalized Yukawa couplings

See talks by Cristoforo Fraser-Taliente and Justin Tan