Metrics from Machine Learning

14.12.2020, string_data2020 CERN

Sven Krippendorf (sven.krippendorf@physik.uni-muenchen.de, @krippendorfsven)

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN based on (2012.04656):

Moduli dependent Calabi-Yau and SU(3)-structure metrics from Machine Learning

in collaboration with:

Lara Anderson

Mathis Gerdes
(applying for PhDs)

James Gray

Nikhil Raghuram

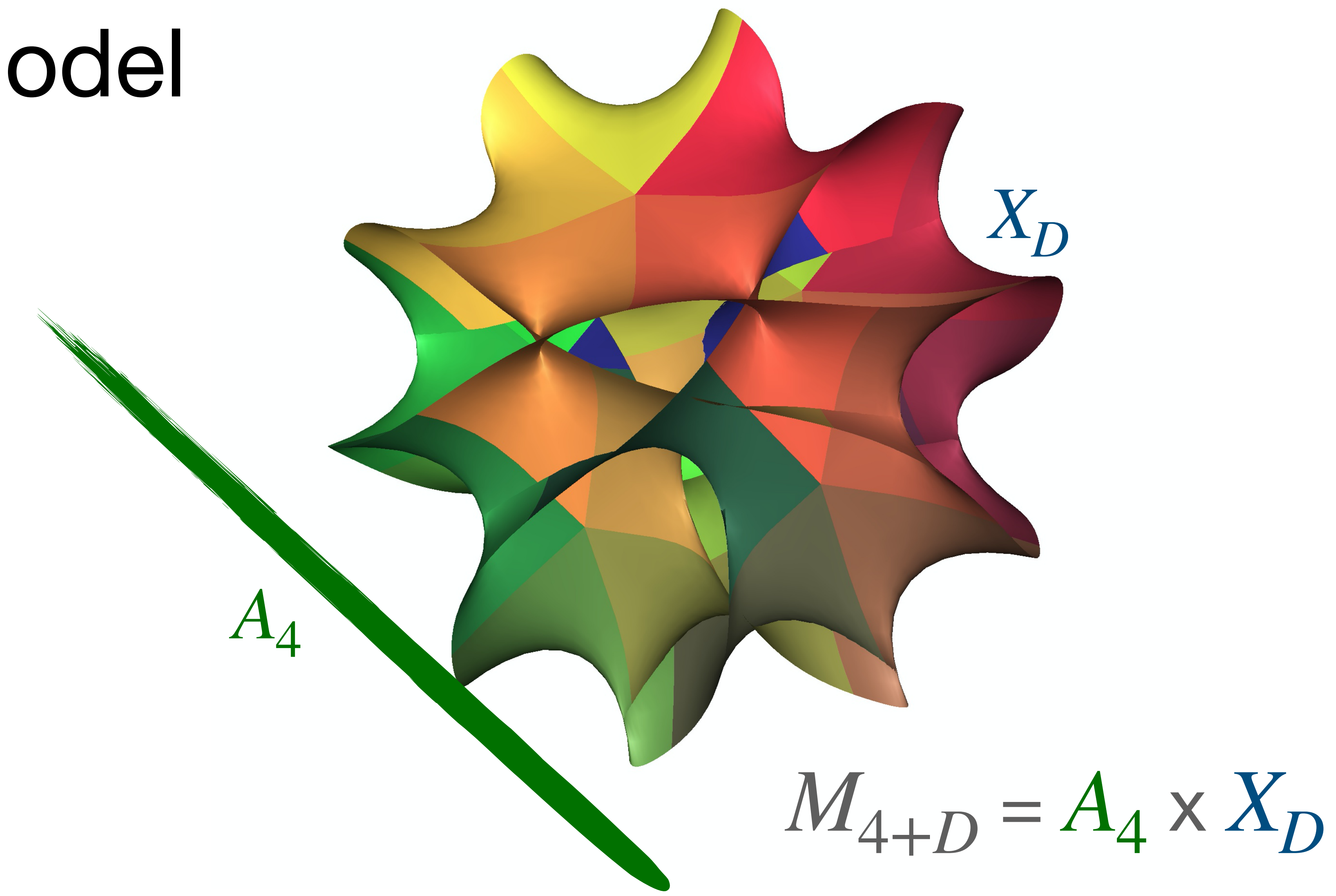
Fabian Ruehle

Metrics matter

- The metric is key in any extra-dimensional physics model

$$S = \int_{M_{4+D}} d^{4+D}x \, \sqrt{-\det g_{4+D}} \; R(g_{4+D})$$

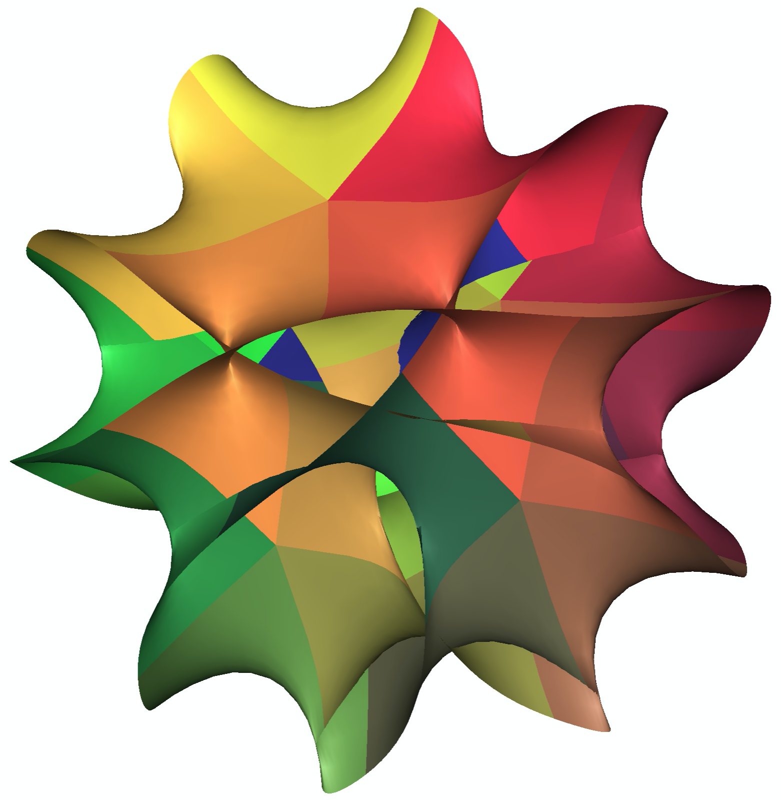
combined metric



- String compactifications are no exception to this. For instance:
  1. Matter kinetic terms (soft-terms, cf. 0906.3297)
  2. Moduli potential (D3-brane inflation [probing directly CY-moduli space])
  3. Massive string spectrum (distance conjecture)

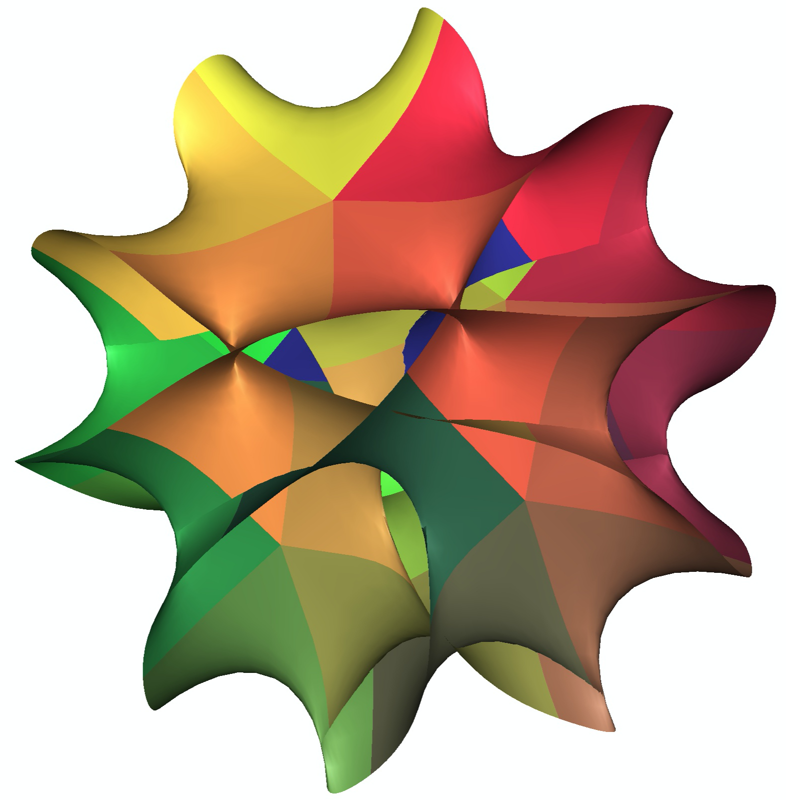
Which Metrics?

6D metrics relevant for string theory



Which Metrics?

6D metrics relevant for string theory



- String Theory EOM for 4D $\mathcal{N} = 1$ Minkowski vacua require a Ricci-flat Kähler metric (Candelas, Horowitz, Strominger, Witten 1985)
- Which compact spaces do exist with a Ricci-flat Kähler metric?

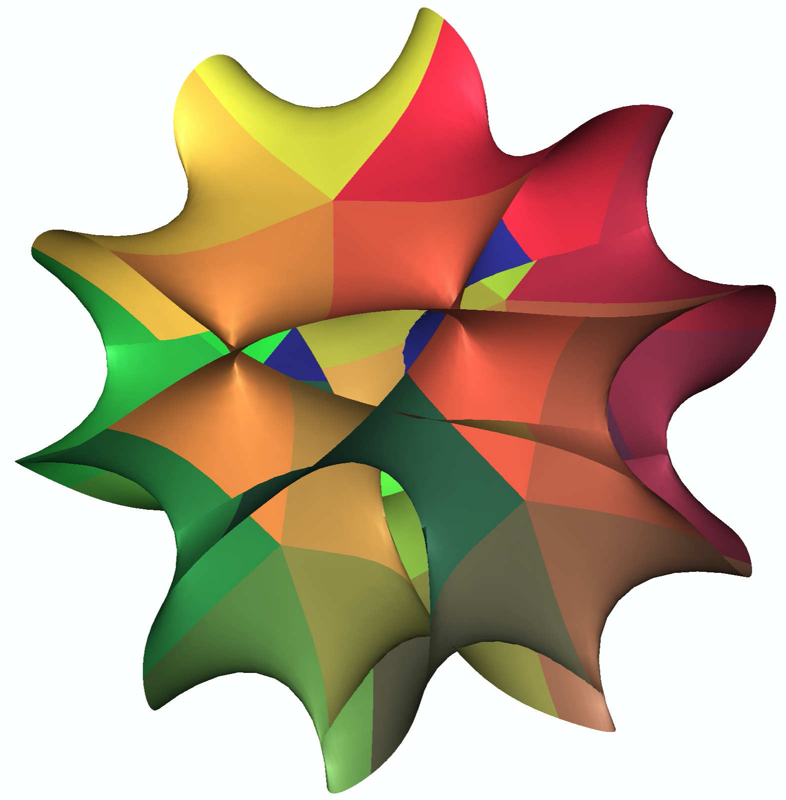
  Calabi-Yau manifolds
  (Example today: Quintic hypersurface in $\mathbb{P}^4$)

Quintic hypersurface in $\mathbb{P}^4$:

$$p_\psi(\vec{z}) = \sum_{i=0}^{d+1} z_i^{d+2} + \psi \prod_{i=0}^{d+1} z_i = 0$$

Which Metrics?

6D metrics relevant for string theory



- String Theory EOM for 4D $\mathcal{N} = 1$ Minkowski vacua require a Ricci-flat Kähler metric (Candelas, Horowitz, Strominger, Witten 1985)
- Which compact spaces do exist with a Ricci-flat Kähler metric?

  Calabi-Yau manifolds
  (Example today: Quintic hypersurface in $\mathbb{P}^4$)

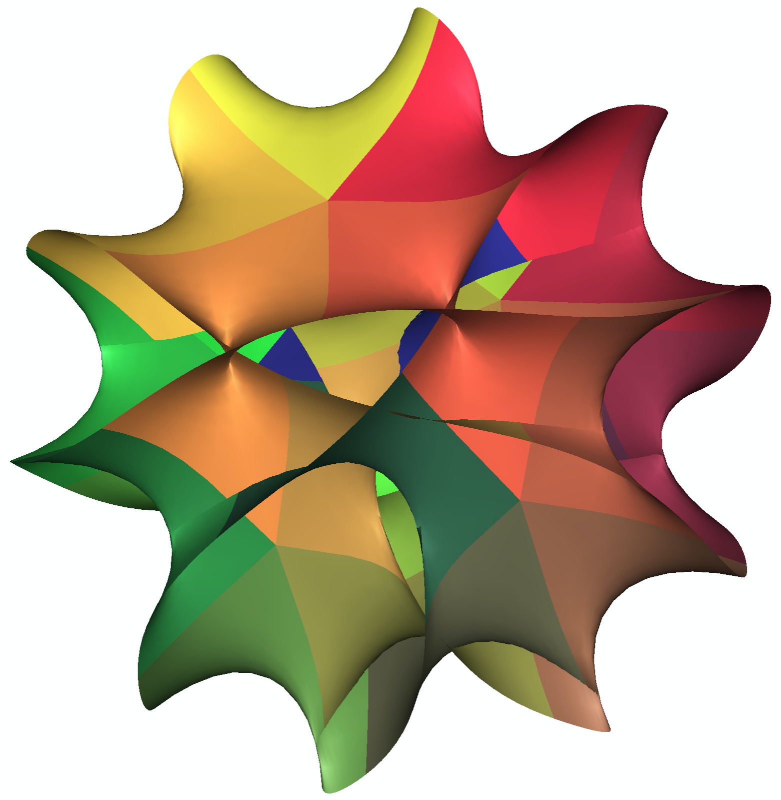
Quintic hypersurface in $\mathbb{P}^4$:

$$p_\psi(\vec{z}) = \sum_{i=0}^{d+1} z_i^{d+2} + \psi \prod_{i=0}^{d+1} z_i = 0$$

- Yau (1977) showed the existence of such a unique Ricci-flat Kähler metric, but without explicit constructions.
- This talk is about how to get such metrics with ML.

Which Metrics?

6D metrics relevant for string theory



- String Theory EOM for 4D $\mathcal{N} = 1$ Minkowski vacua require a Ricci-flat Kähler metric (Candelas, Horowitz, Strominger, Witten 1985)
- Which compact spaces do exist with a Ricci-flat Kähler metric?

  Calabi-Yau manifolds
  (Example today: Quintic hypersurface in $\mathbb{P}^4$)

- Yau (1977) showed the existence of such a unique Ricci-flat Kähler metric, but without explicit constructions.
- This talk is about how to get such metrics with ML.
- One ansatz is by using algebraic metrics.

Quintic hypersurface in $\mathbb{P}^4$:

$$p_\psi(\vec{z}) = \sum_{i=0}^{d+1} z_i^{d+2} + \psi \prod_{i=0}^{d+1} z_i = 0$$

Algebraic metrics:

$$K = 1/2\pi \ln(\mathbf{k})$$

$$\mathbf{k} = \sum_{\alpha,\bar{\beta}=0}^{N_k} s_\alpha(\vec{z})\ H_{\alpha\bar{\beta}}\ \bar{s}_{\bar{\beta}}(\vec{\bar{z}})$$

$$g_{a\bar{b}} = \partial_a \bar{\partial}_{\bar{b}} K = \frac{1}{2\pi} \frac{\mathbf{k}\mathbf{k}_{a\bar{b}} - \mathbf{k}_a \mathbf{k}_{\bar{b}}}{\mathbf{k}^2}$$

Metrics are hard without ML

6D metrics relevant for string theory

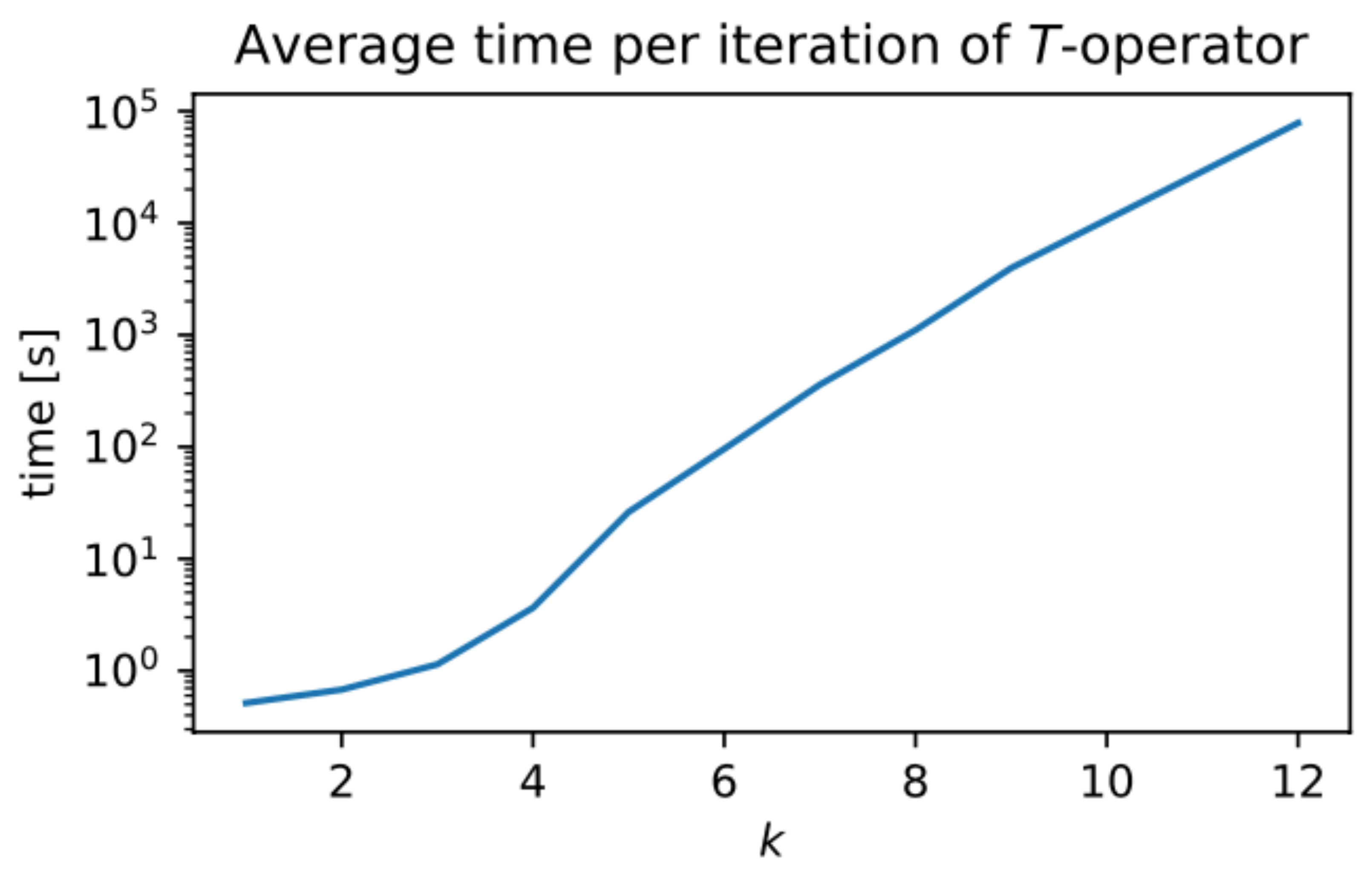
- Finite distance methods "fail" (Headrick, Wiseman 2009)

- Spectral methods simplify, but they are currently inefficient:
    1. Single point in moduli space
    2. High accuracies become expensive

  (Donaldson, Braun, Belidze, Douglas, Ovrut, Karp, Cui, Gray, Lukic, Ashmore, He; Kachru, Tripathy, Zimet; Headrick and Nasar)

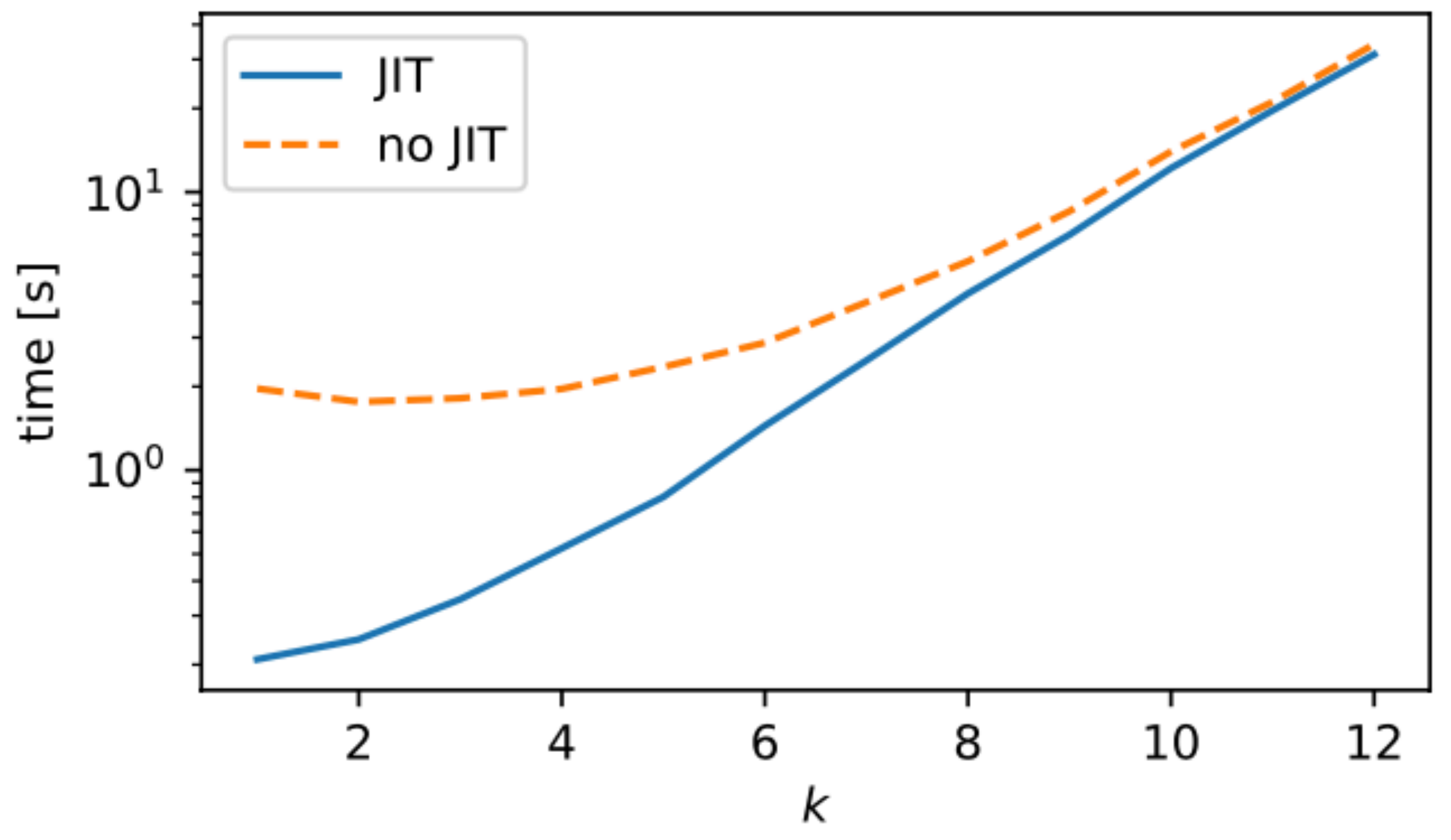
- How about non-Kähler solutions?

- Target on a practical level: metric with reasonable accuracy for one string compactification ~ O(1 day) [impossible with non ML algorithms]



k ~ accuracy of spectral resolution

Time to check accuracy of solution σ



Which metric?

What is the optimisation problem

$$J \wedge J \wedge J \sim \det g$$

$$\Omega = \frac{1}{\partial p_\psi(\vec{z})/\partial z_b} \bigwedge_{\substack{c=1,\dots,d \\ c \neq a,b}} dz_c$$

Which metric?

What is the optimisation problem

1. Ricci-flatness: (Induced FS is not Ricci-flat):
   Ricci tensor: $R_{i\bar{j}} = -\partial_i \bar{\partial}_{\bar{j}} \log \det g$
   Cheaper alternative (less derivatives) via Monge-Ampere equation:

$$J \wedge J \wedge J = \kappa\, \Omega \wedge \bar{\Omega} \qquad \rightarrow \qquad \mathscr{L}_{\mathrm{MA}} = \frac{1}{\int_X \Omega \wedge \bar{\Omega}} \int_X \left| 1 - \frac{1}{\kappa} \frac{J^3}{\Omega \wedge \bar{\Omega}} \right|$$

$$J \wedge J \wedge J \sim \det g$$

$$\Omega = \frac{1}{\partial p_\psi(\vec{z})/\partial z_b} \bigwedge_{\substack{c=1,\ldots,d \\ c \neq a,b}} dz_c$$

Which metric?

What is the optimisation problem

1. Ricci-flatness: (Induced FS is not Ricci-flat):
   Ricci tensor: $R_{i\bar{j}} = -\partial_i \bar{\partial}_{\bar{j}} \log \det g$
   Cheaper alternative (less derivatives) via Monge-Ampere equation:

$$J \wedge J \wedge J = \kappa\, \Omega \wedge \bar{\Omega} \qquad \rightarrow \quad \mathscr{L}_{\mathrm{MA}} = \frac{1}{\int_X \Omega \wedge \bar{\Omega}} \int_X \left| 1 - \frac{1}{\kappa} \frac{J^3}{\Omega \wedge \bar{\Omega}} \right|$$

2. Kählerity:

$$dJ = 0 \quad \leftrightarrow \quad g_{i\bar{j},k}\, dz_i \wedge d\bar{z}_{\bar{j}} \wedge dz_k = 0 = g_{i\bar{j},\bar{k}}\, dz_i \wedge d\bar{z}_{\bar{j}} \wedge d\bar{z}_{\bar{k}}$$

$$c_{ijk} = g_{i\bar{j},k} - g_{k\bar{j},i} = 0, \qquad \rightarrow \quad \mathscr{L}_{\mathrm{dJ}} = \sum_{i,j,k} ||\mathrm{Re}(c_{ijk})||_n + ||\mathrm{Im}(c_{ijk})||_n$$

$$J \wedge J \wedge J \sim \det g$$

$$\Omega = \frac{1}{\partial p_\psi(\vec{z})/\partial z_b} \bigwedge_{\substack{c=1,\dots,d \\ c \neq a,b}} dz_c$$

Which metric?

What is the optimisation problem

1. Ricci-flatness: (Induced FS is not Ricci-flat):
   Ricci tensor: $R_{i\bar{j}} = -\partial_i \bar{\partial}_{\bar{j}} \log \det g$
   Cheaper alternative (less derivatives) via Monge-Ampere equation:

$$J \wedge J \wedge J = \kappa\, \Omega \wedge \bar{\Omega} \quad \rightarrow \quad \mathscr{L}_{\text{MA}} = \frac{1}{\int_X \Omega \wedge \bar{\Omega}} \int_X \left| 1 - \frac{1}{\kappa} \frac{J^3}{\Omega \wedge \bar{\Omega}} \right|$$

2. Kählerity:

$$dJ = 0 \quad \leftrightarrow \quad g_{i\bar{j},k}\, dz_i \wedge d\bar{z}_{\bar{j}} \wedge dz_k = 0 = g_{i\bar{j},\bar{k}}\, dz_i \wedge d\bar{z}_{\bar{j}} \wedge d\bar{z}_{\bar{k}}$$

$$c_{ijk} = g_{i\bar{j},k} - g_{k\bar{j},i} = 0, \quad \rightarrow \quad \mathscr{L}_{\text{dJ}} = \sum_{i,j,k} ||\text{Re}(c_{ijk})||_n + ||\text{Im}(c_{ijk})||_n$$

3. Well defined across different coordinate patches:

$$g^{(j)} = T_{ij} \cdot g^{(i)} \cdot T_{ij}^\dagger, \quad T_{ij} = \partial \vec{z}^{(i)} / \partial \vec{z}^{(j)} \quad \rightarrow \quad \mathscr{L}_{\text{Transition}} = \frac{1}{d} \sum_{k,j} \left|\left| g_{\text{NN}}^{(k)}(\vec{z}) - T_{jk}(\vec{z}) \cdot g_{\text{NN}}^{(j)}(\vec{z}) \cdot T_{jk}^\dagger(\bar{\vec{z}}) \right|\right|_n$$

How to measure accuracy?

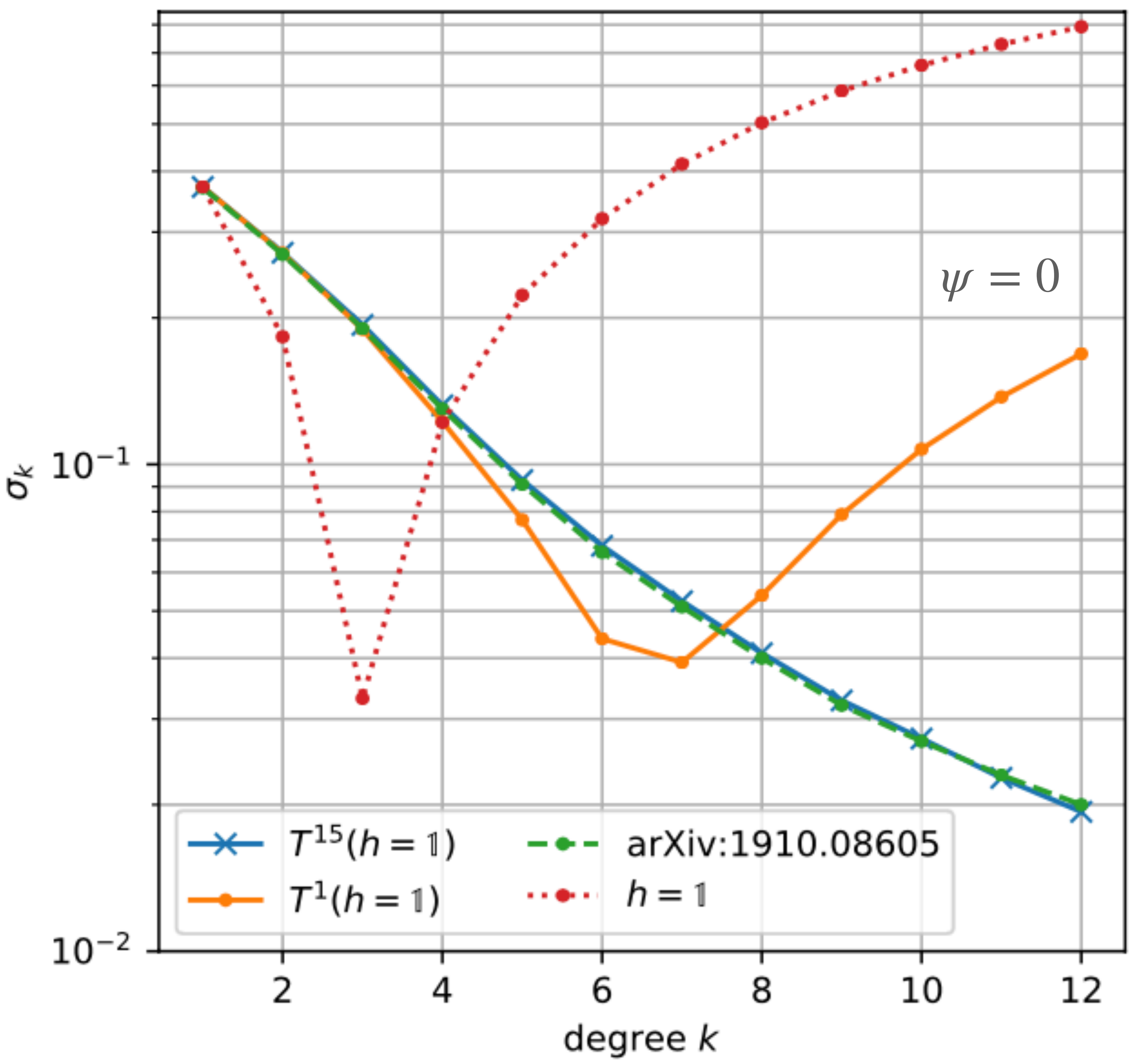
- Monte-Carlo sampling:

$$\int_{X_\psi} f\, d\mathrm{Vol}_{\mathrm{CY}} = \int_{X_\psi} f\ \frac{d\mathrm{Vol}_{\mathrm{CY}}}{dA} dA \approx \frac{1}{N}\sum_{i=1}^{N} f(\vec{z}_i) w(\vec{z}_i)$$

$$w(\vec{z}_i) = \frac{d\mathrm{Vol}_{\mathrm{CY}}}{dA}\Big|_{\vec{z}_i}, \qquad d\mathrm{Vol}_{\mathrm{CY}} \propto \Omega \wedge \bar{\Omega}\,, \quad dA \propto i_p^* \omega^{\mathrm{FS}}_{\mathbb{P}^4}$$

- Use this to evaluate σ-accuracy:

$$\sigma = \frac{1}{\int_X \Omega \wedge \bar{\Omega}} \int_X \left| 1 - \frac{1}{\kappa}\ \frac{J^3}{\Omega \wedge \bar{\Omega}} \right|$$



Neural networks to the rescue?

- Can NN give good approximations?
- Motivation beyond universal approximation scheme (NN can be shown to give good and accurate predictions to PDEs):
  - Solutions to high-dimensional Schrödinger equations (Rupp, Tkatchenko, Müller, von Lilienfeld 2012, …)
  - Black-Scholes PDE (Grohs, Hornung, Jentzen, von Wurstemberger 2018, …)
  - Approximation rates of NNs to solutions of PDEs (Kutyniok, Petersen, Raslan, Schneider 2019, …)

Why can Neural Networks be useful?

- Searching for a function with particular properties, such as:

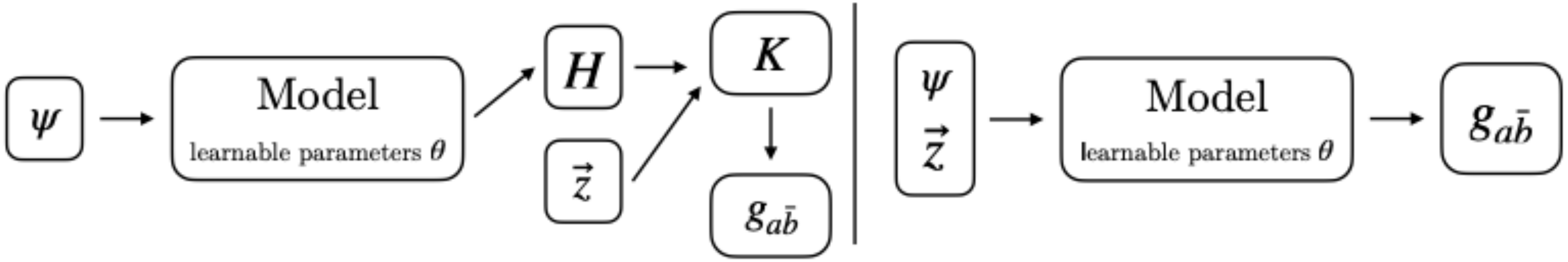
$$1.\ \mathscr{L}_{MA}(g_{a\bar{b}}) = 0, \qquad 2.\ \mathscr{L}_{dJ}(g_{a\bar{b}}) = 0, \qquad 3.\ \mathscr{L}_{\text{overlap}}(g_{a\bar{b}}) = 0$$

- If $g_{a\bar{b}}$ is the output of a NN, we need to be able to calculate derivatives of this network to evaluate (not just to optimise) loss functions.

- Auto-differentiation readily allows to do this and implementations in standard packages can do this (here: Pytorch, TensorFlow/Keras, JAX).

Our approach

- Reporting here on learning H and the metric directly



- Results for quintic. Generalization to other examples straight-forward.

Our approach

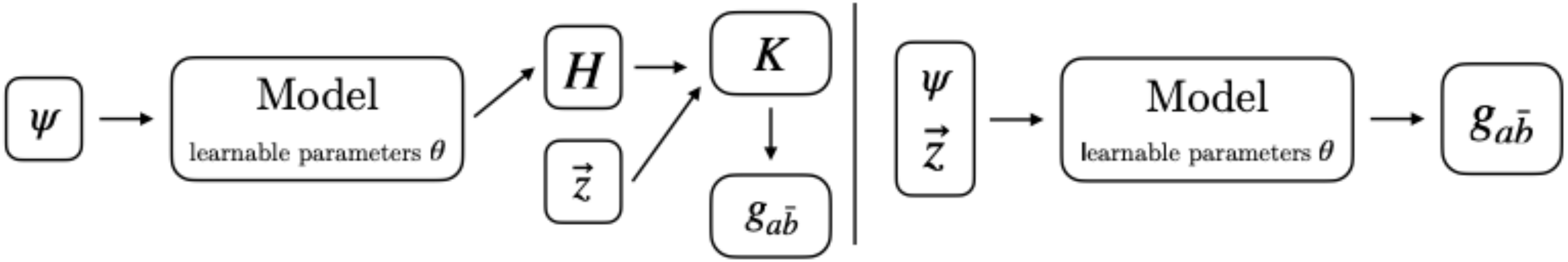
Algebraic metrics:

$$K = 1/2\pi \ln(\mathbf{k})$$

$$\mathbf{k} = \sum_{\alpha,\bar{\beta}=0}^{N_k} s_\alpha(\vec{z})\ H_{\alpha\bar{\beta}}\ \bar{s}_{\bar{\beta}}(\vec{\bar{z}})$$

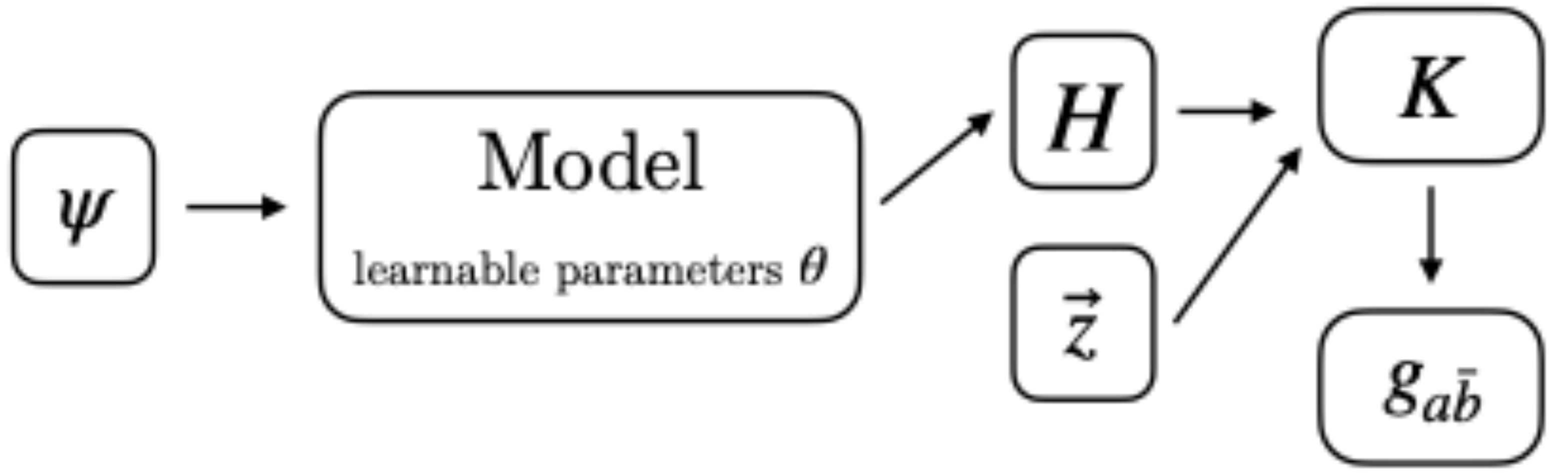
$$g_{a\bar{b}} = \partial_a \bar{\partial}_{\bar{b}} K = \frac{1}{2\pi} \frac{\mathbf{k}\mathbf{k}_{a\bar{b}} - \mathbf{k}_a \mathbf{k}_{\bar{b}}}{\mathbf{k}^2}$$

- Reporting here on learning H and the metric directly



- Results for quintic. Generalization to other examples straight-forward.

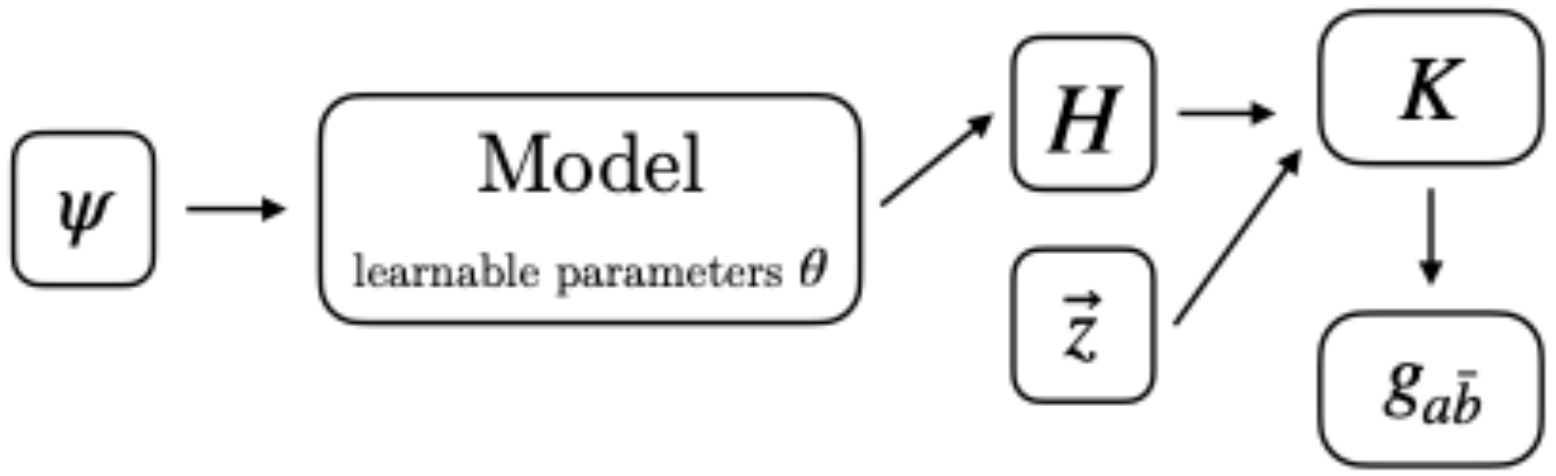
Learning H



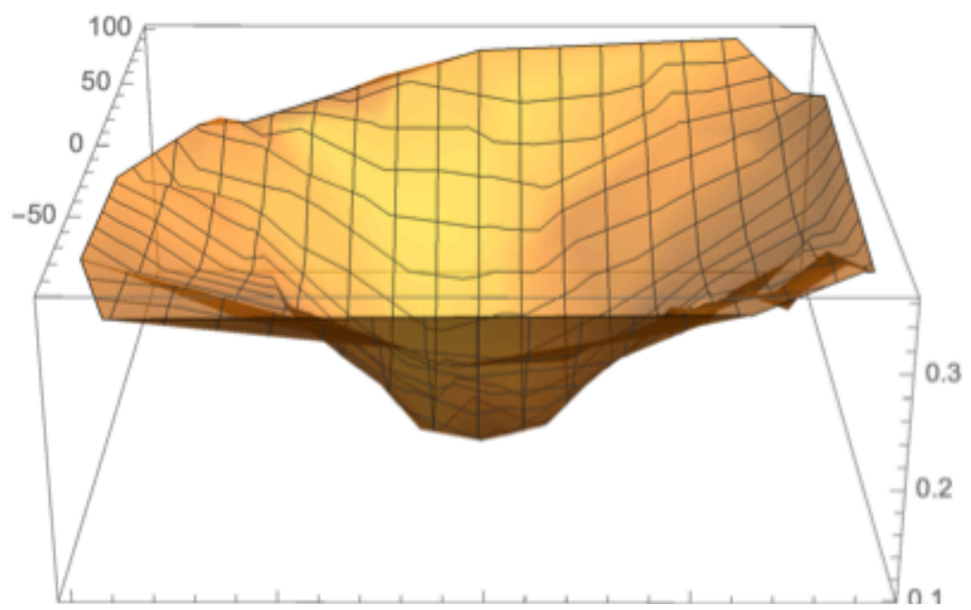
- Overlap and Kähler conditions automatically satisfied
- 2 Strategies:
  1. Use Donaldson data to formulate regression problem
  2. Use σ-measure to train H directly
- Standard few layer feedforward (dense) neural networks, ADAM optimiser.

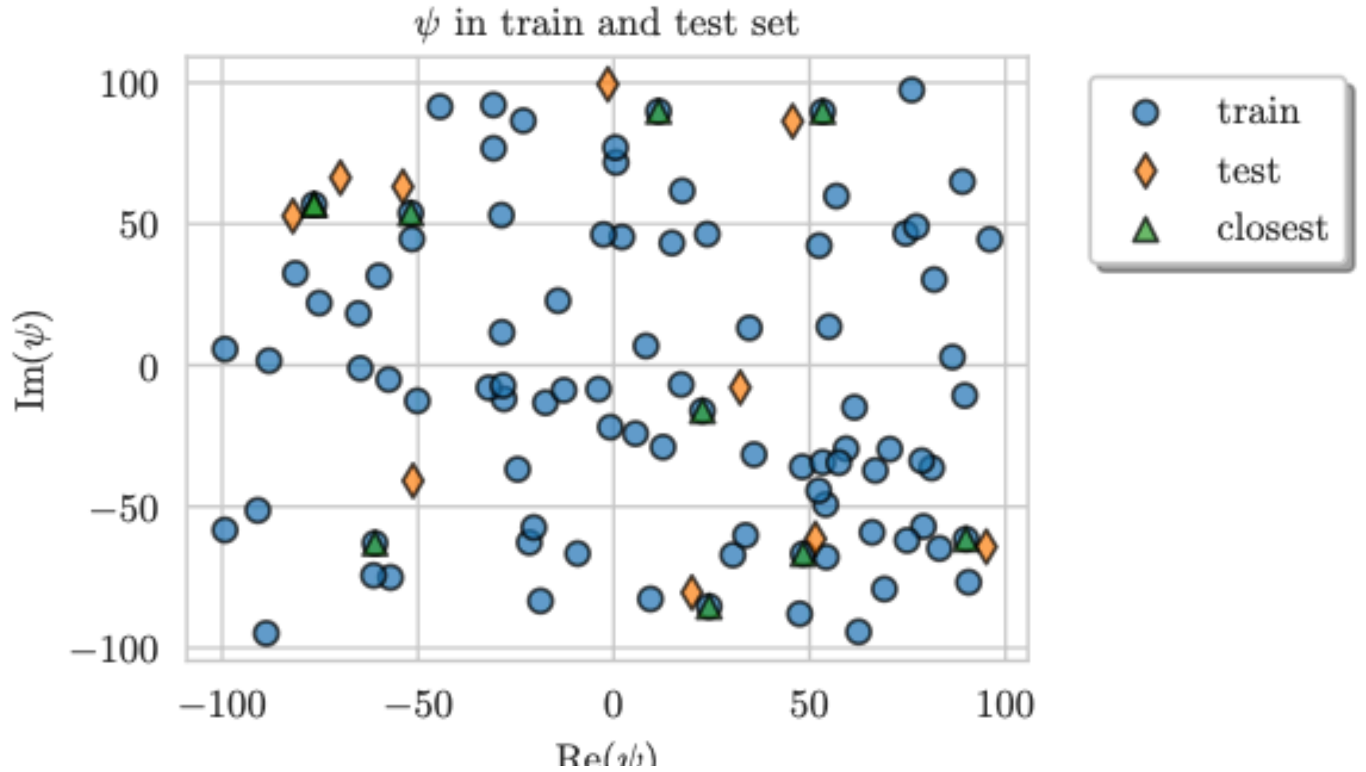
Learning H

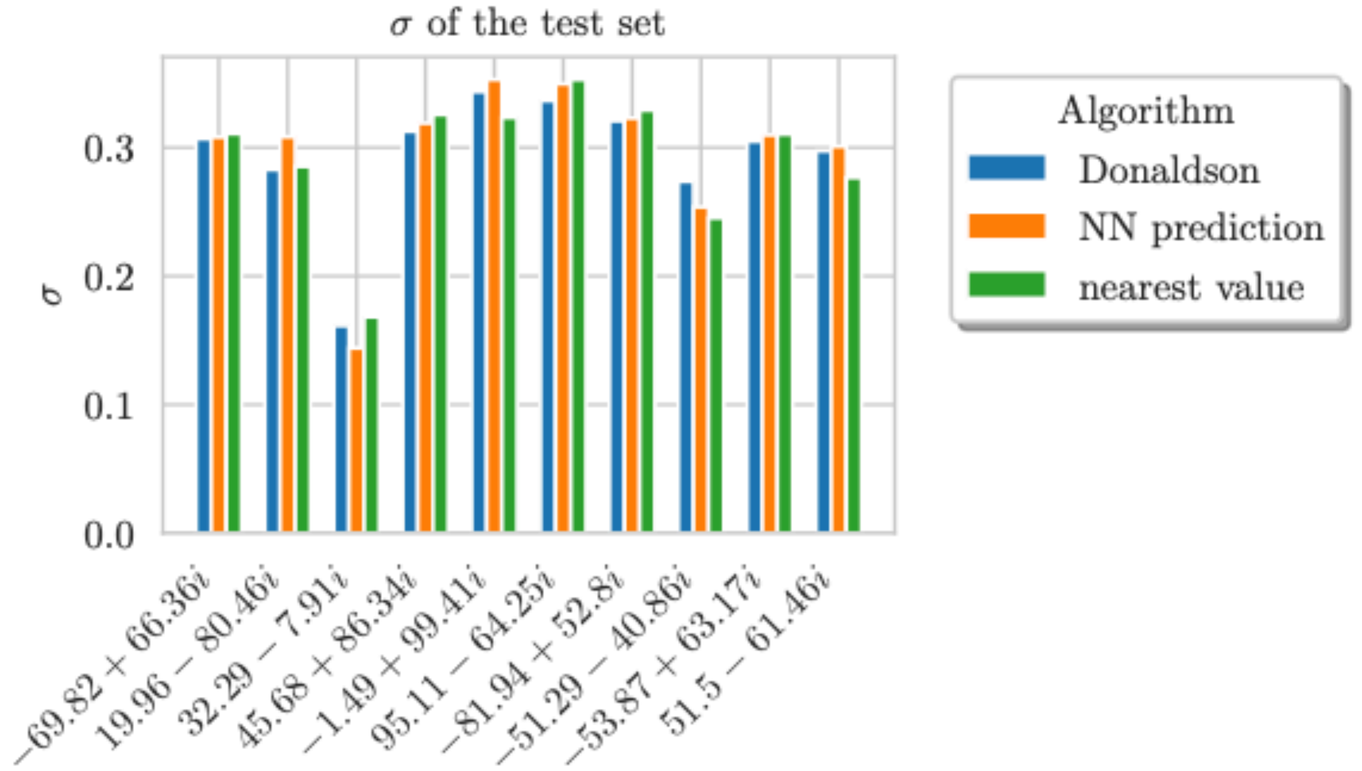
Optimising with Donaldson

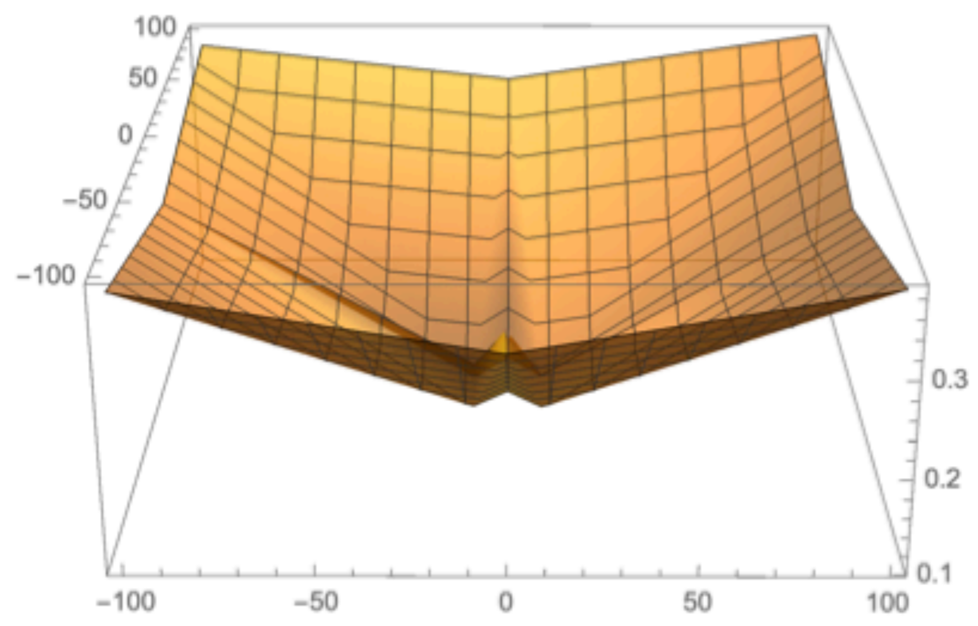


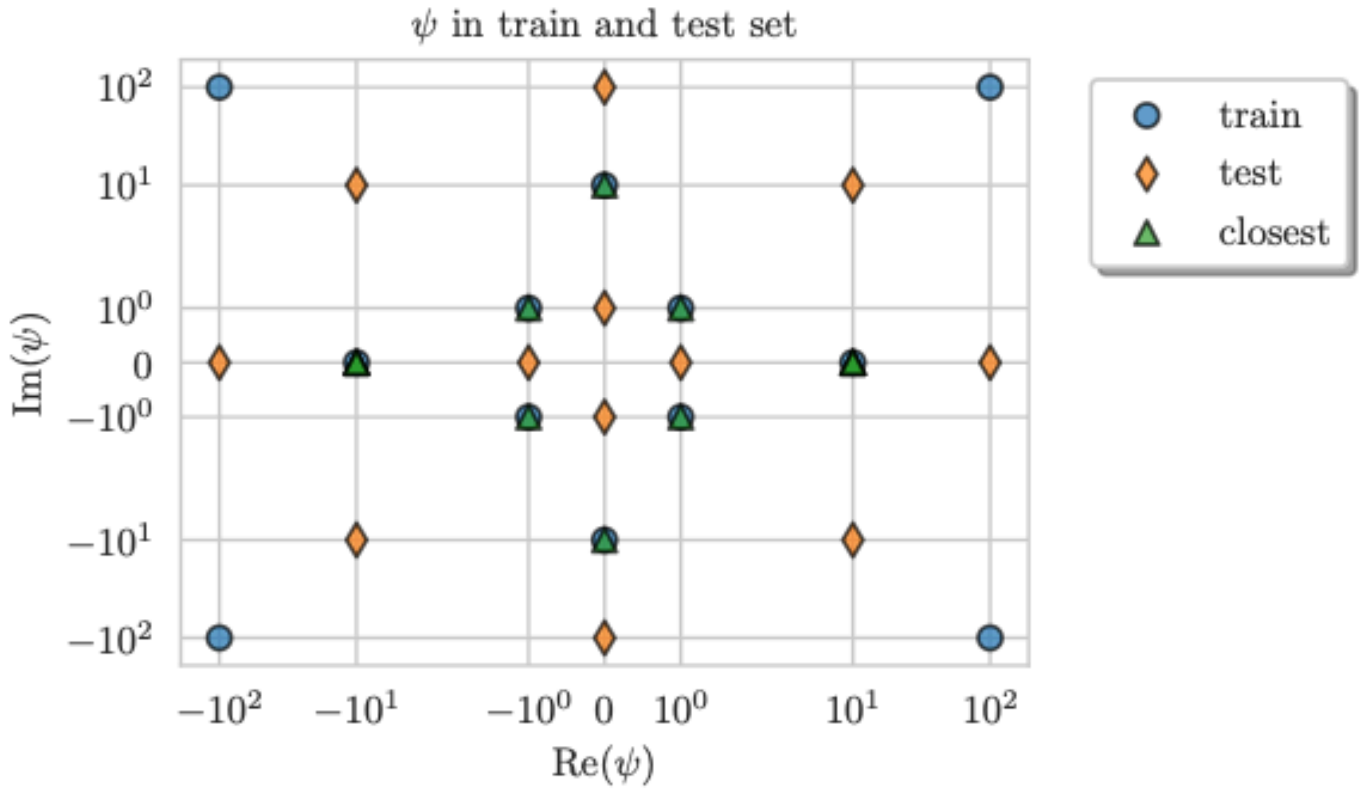
- Network learns interpolation and shows good performance even outside of trained area
- Experiment k=3 (dims of H: 35x35)

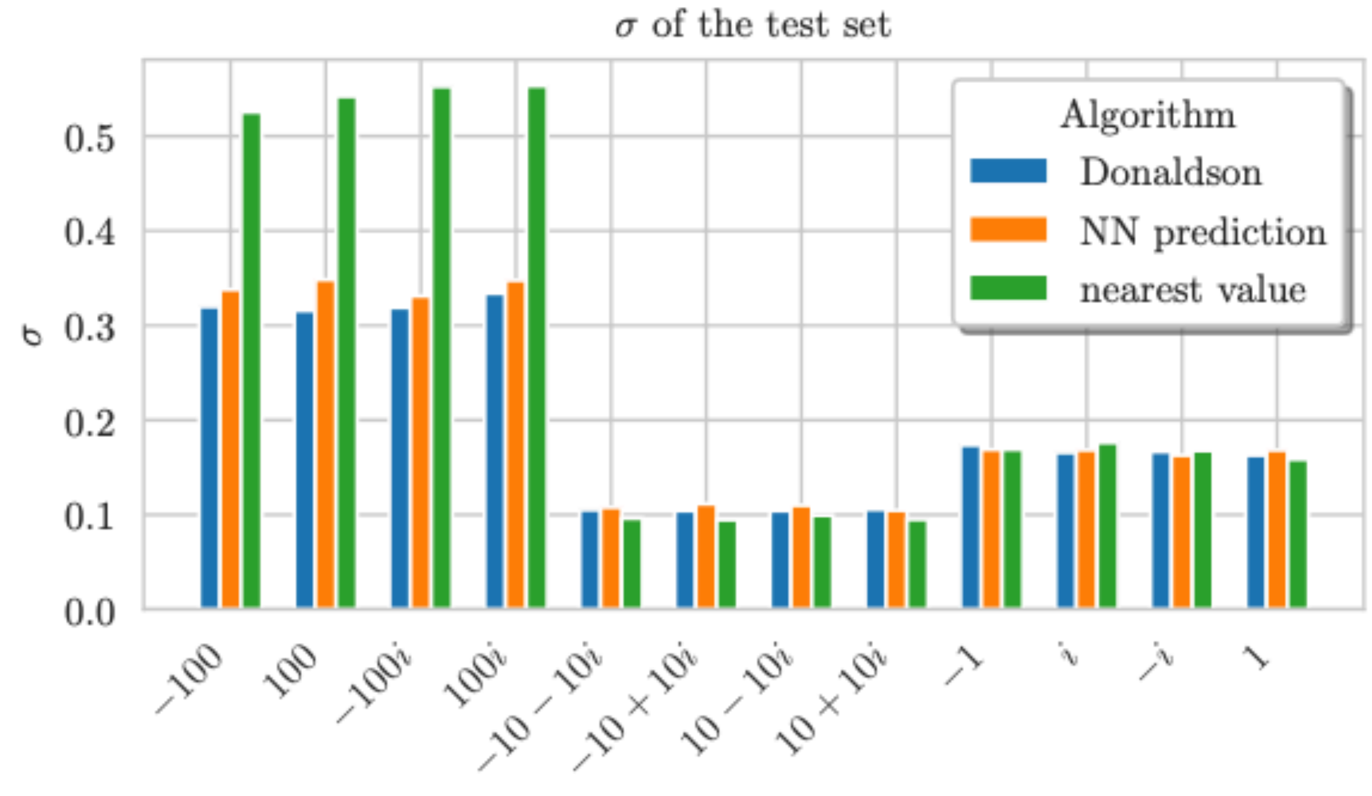






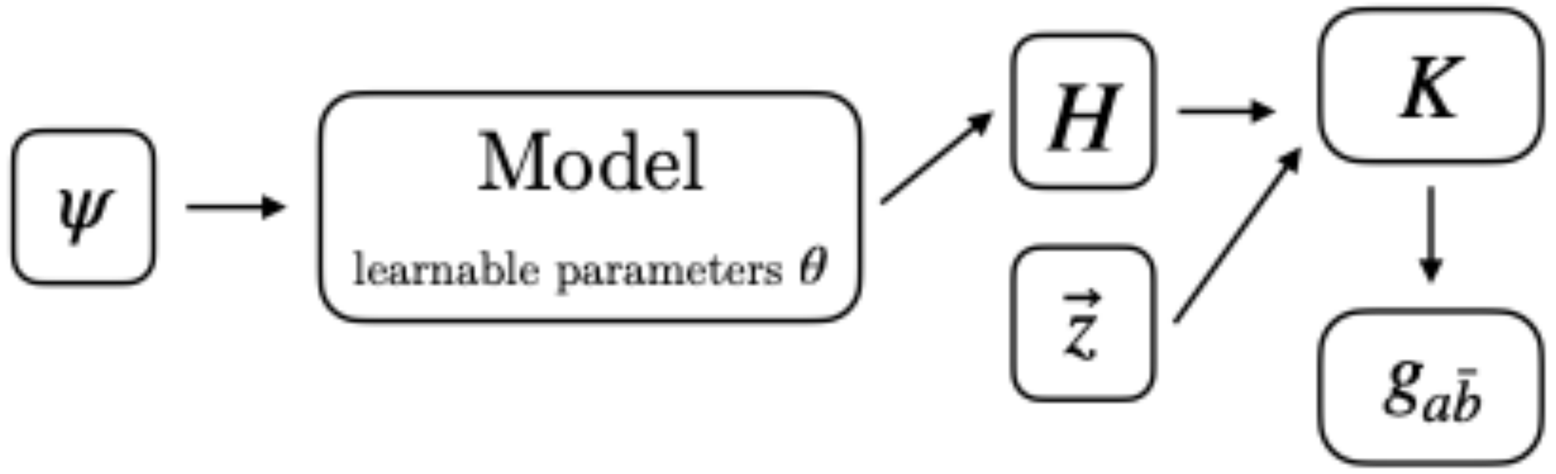






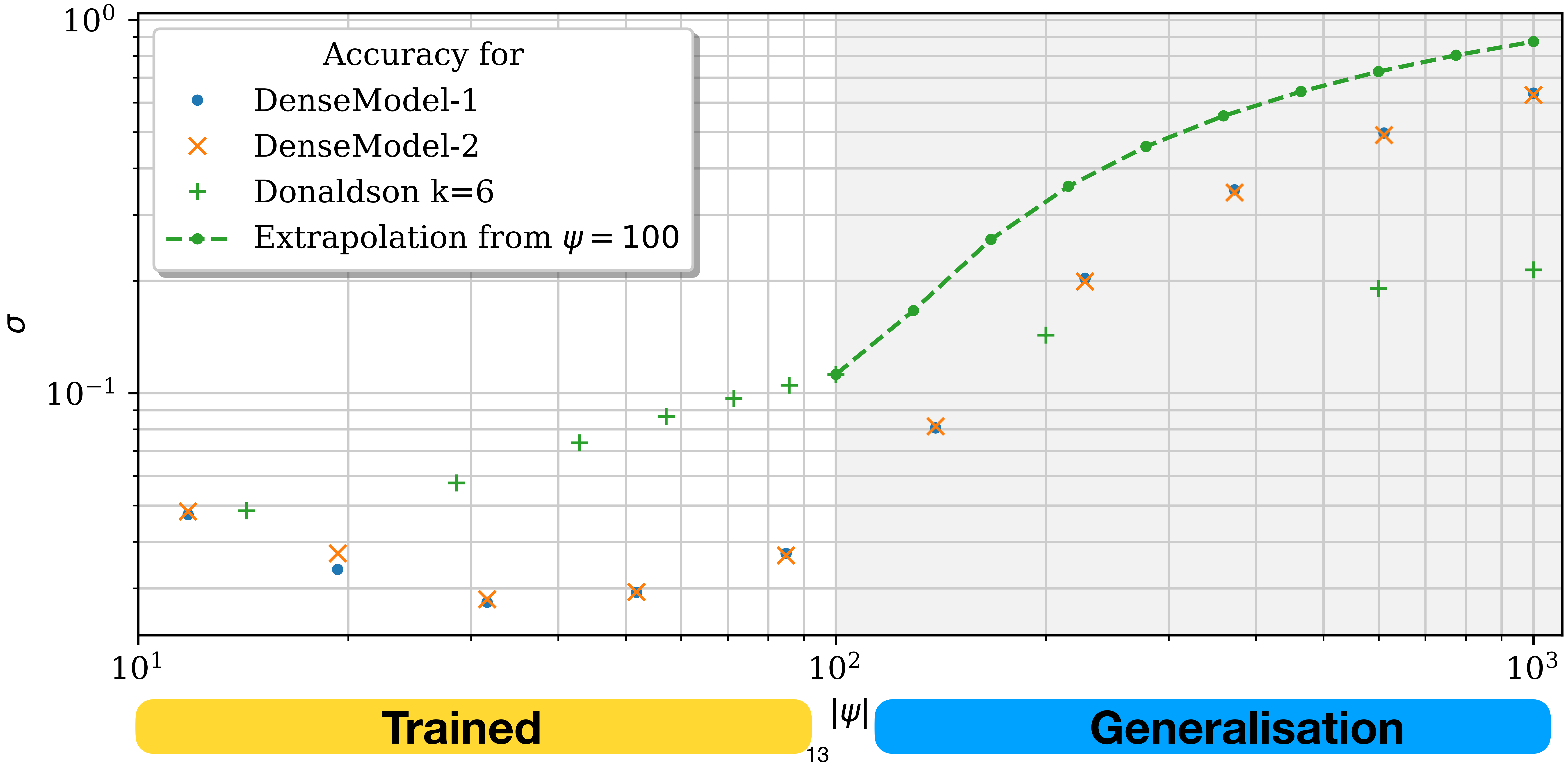
Learning H

Optimising with σ (no Donaldson)

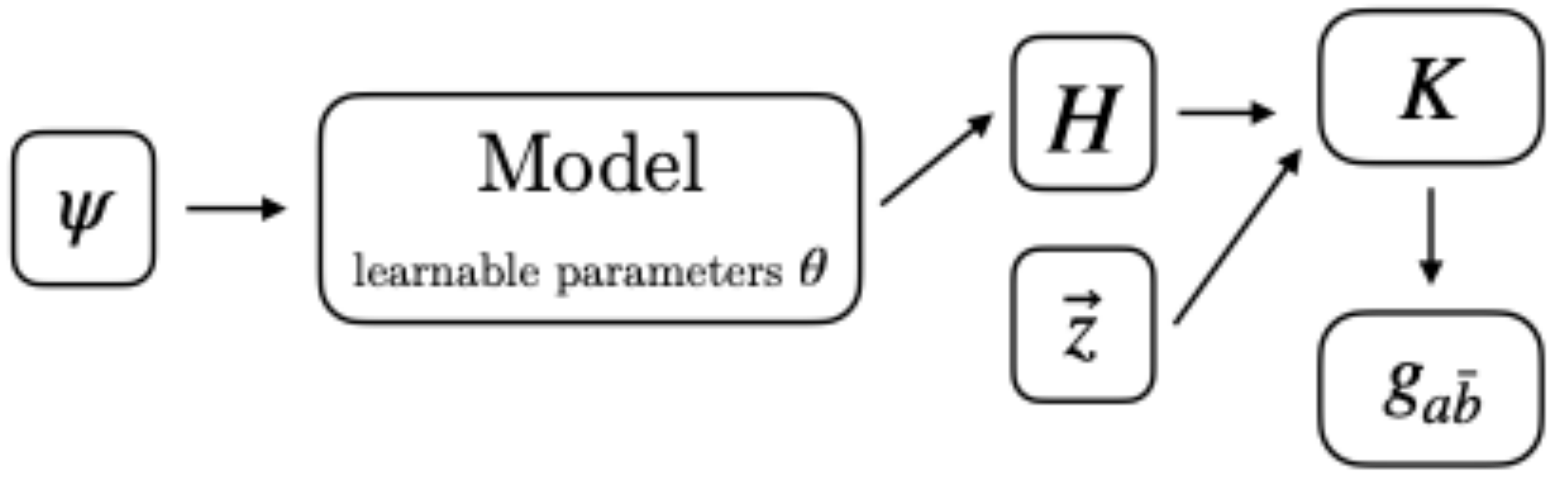


- k=6 (42025 components in H), sampling fast and always using new points

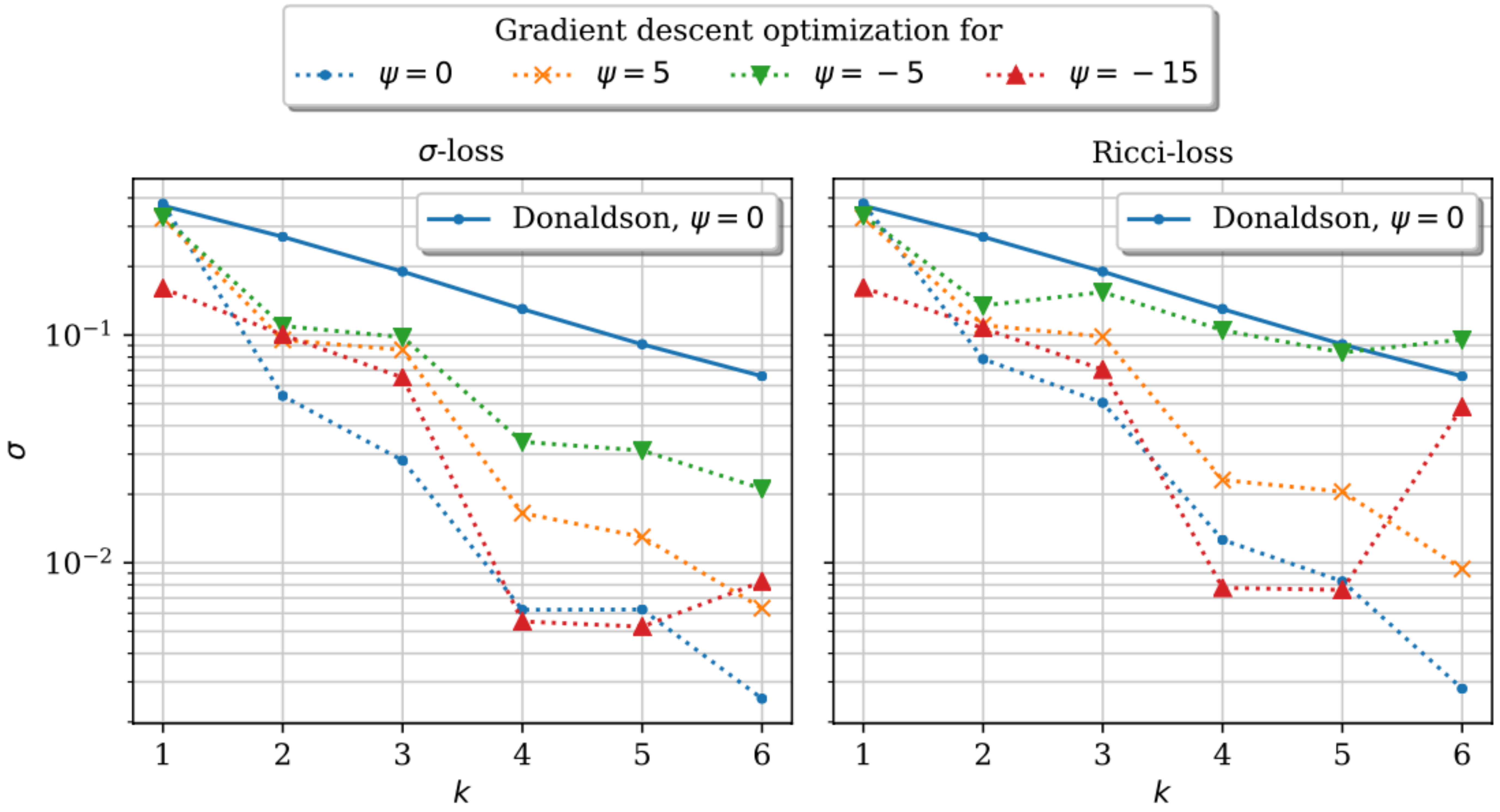
$$\sigma = \frac{1}{\int_X \Omega \wedge \bar{\Omega}} \int_X \left| 1 - \frac{1}{\kappa} \frac{J^3}{\Omega \wedge \bar{\Omega}} \right|$$



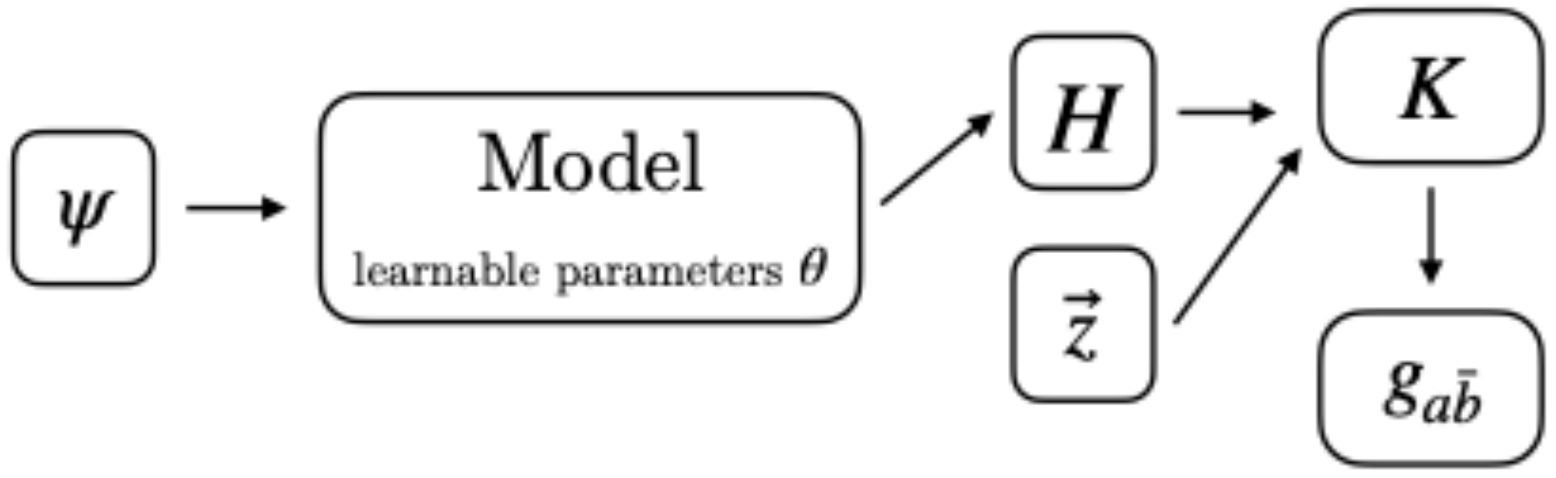
Learning H



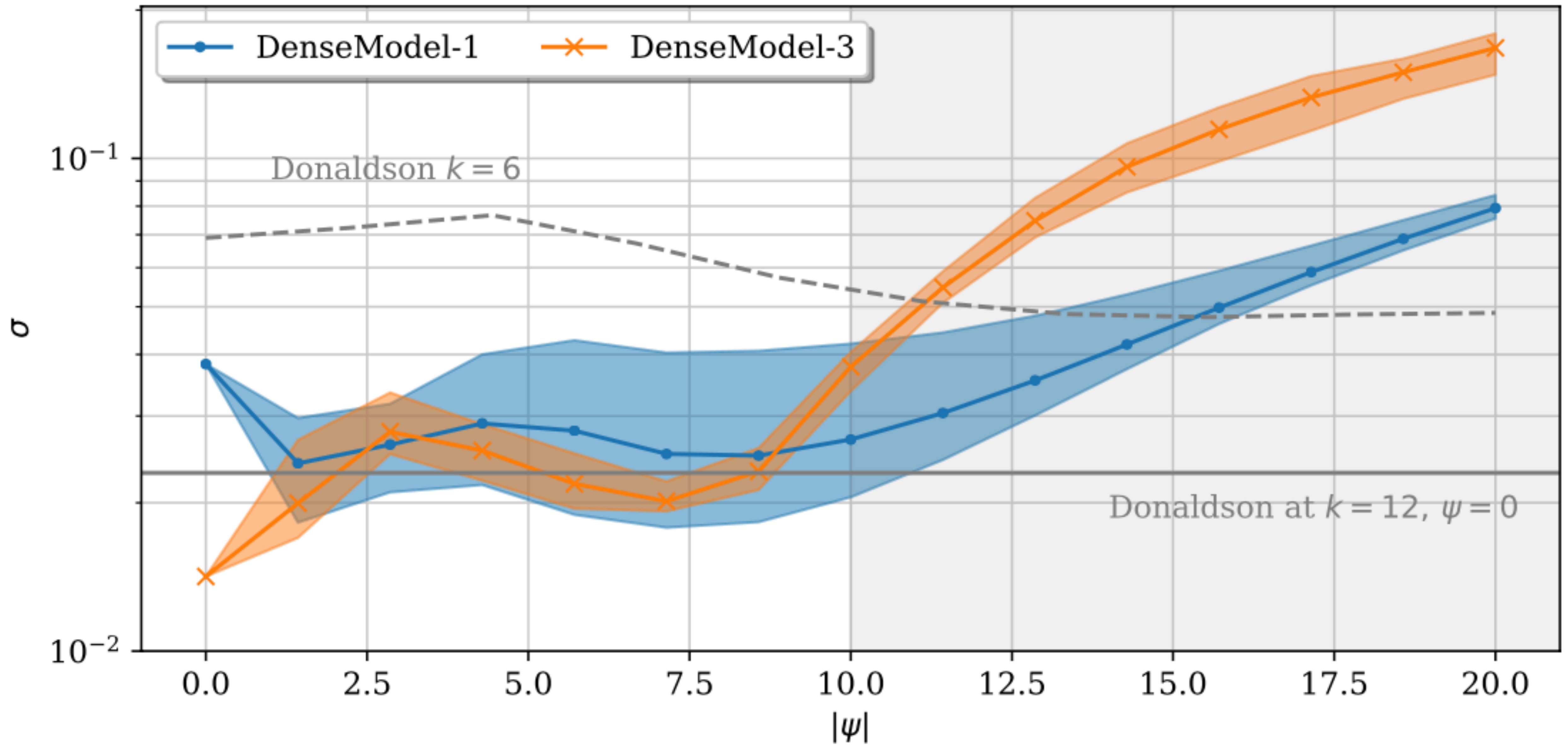
- Instead of σ we can also use R directly (2 additional derivatives, more expensive).
- Accuracy sensitive to architecture and range we train for.
- Metric eigenvalues close to metric eigenvalues obtained from Donaldson at higher k!
- Interesting structures in H obtained from Donaldson.

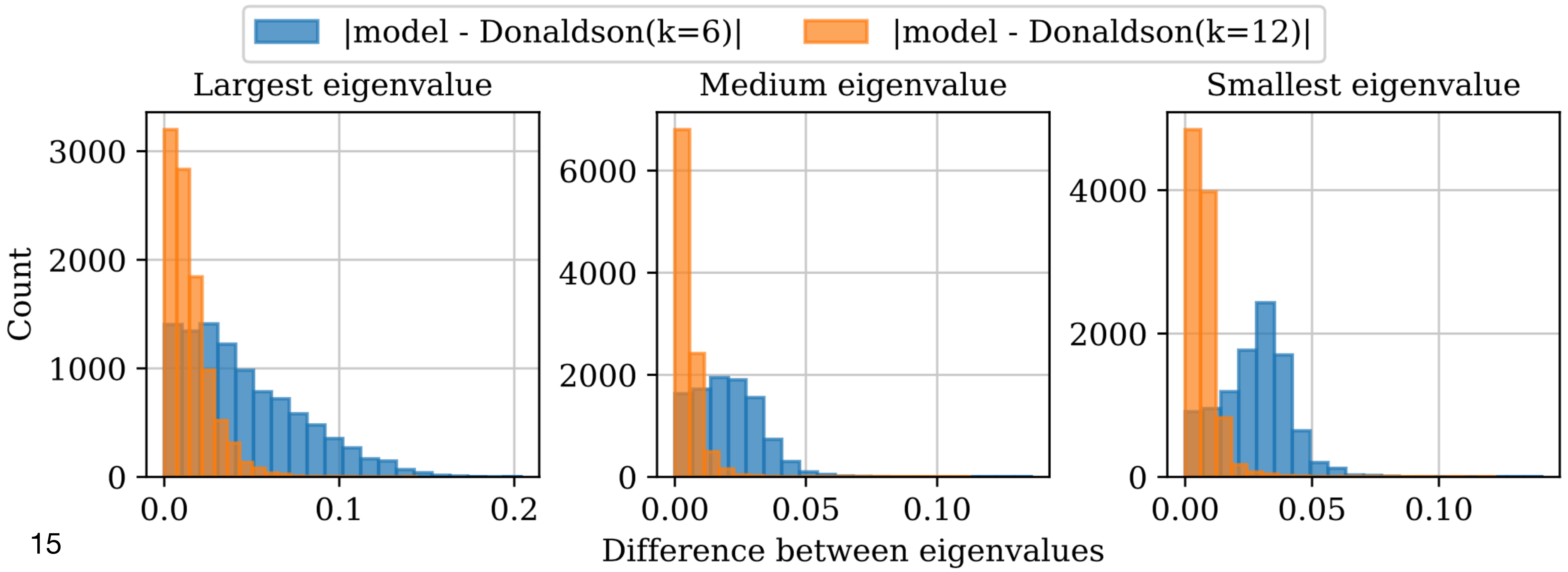


Learning H

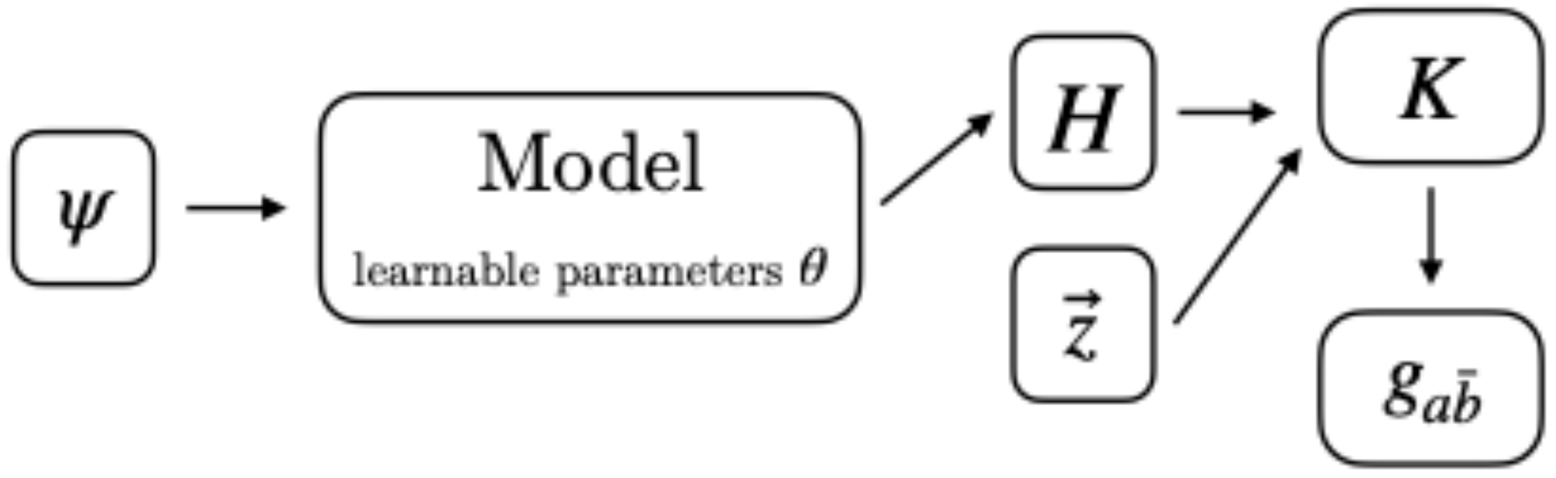


- Instead of σ we can also use R directly (2 additional derivatives, more expensive).
- Accuracy sensitive to architecture and range we train for.
- Metric eigenvalues close to metric eigenvalues obtained from Donaldson at higher k!
- Interesting structures in H obtained from Donaldson.

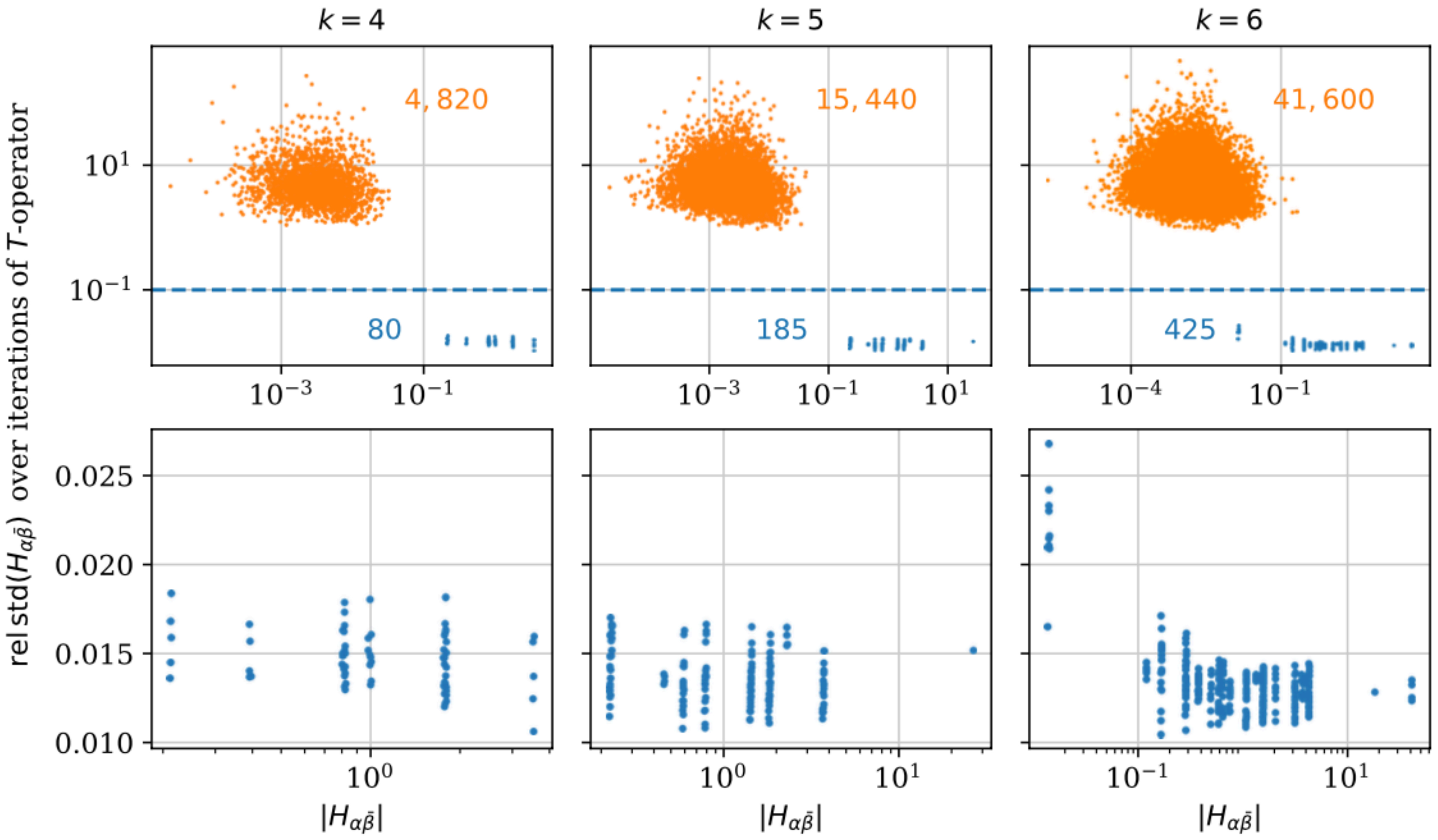




Learning H

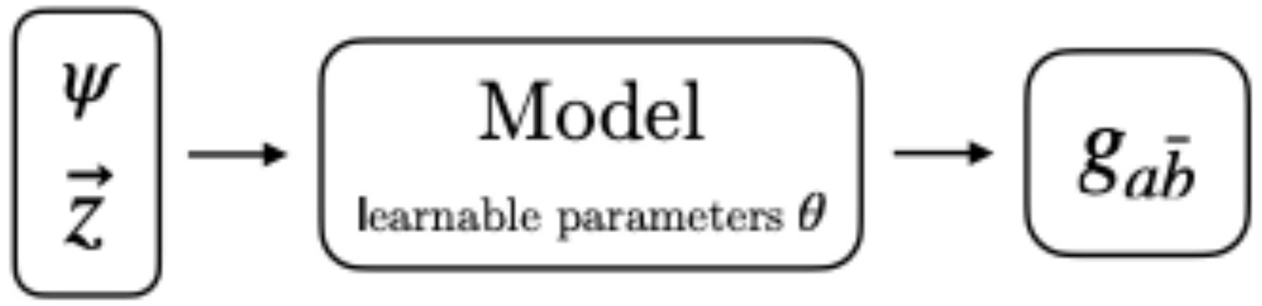


- Instead of σ we can also use R directly (2 additional derivatives, more expensive).
- Accuracy sensitive to architecture and range we train for.
- Metric eigenvalues close to metric eigenvalues obtained from Donaldson at higher k!
- Interesting structures in H obtained from Donaldson.



$$\mathbb{Z}^{(1)}_{d+2} \;:\; [z_0 : z_1 : \ldots : z_{d+1}] \mapsto [\alpha^0 z_0 : \alpha^1 z_1 : \ldots : \alpha^{d+1} z_{d+1}]\,, \qquad \alpha = e^{2\pi i/(d+2)}$$
$$\mathbb{Z}^{(2)}_{d+2} \;:\; [z_0 : z_1 : \ldots : z_{d+1}] \mapsto [z_1 : z_2 : \ldots : z_{d+1} : z_0]$$

Learning g



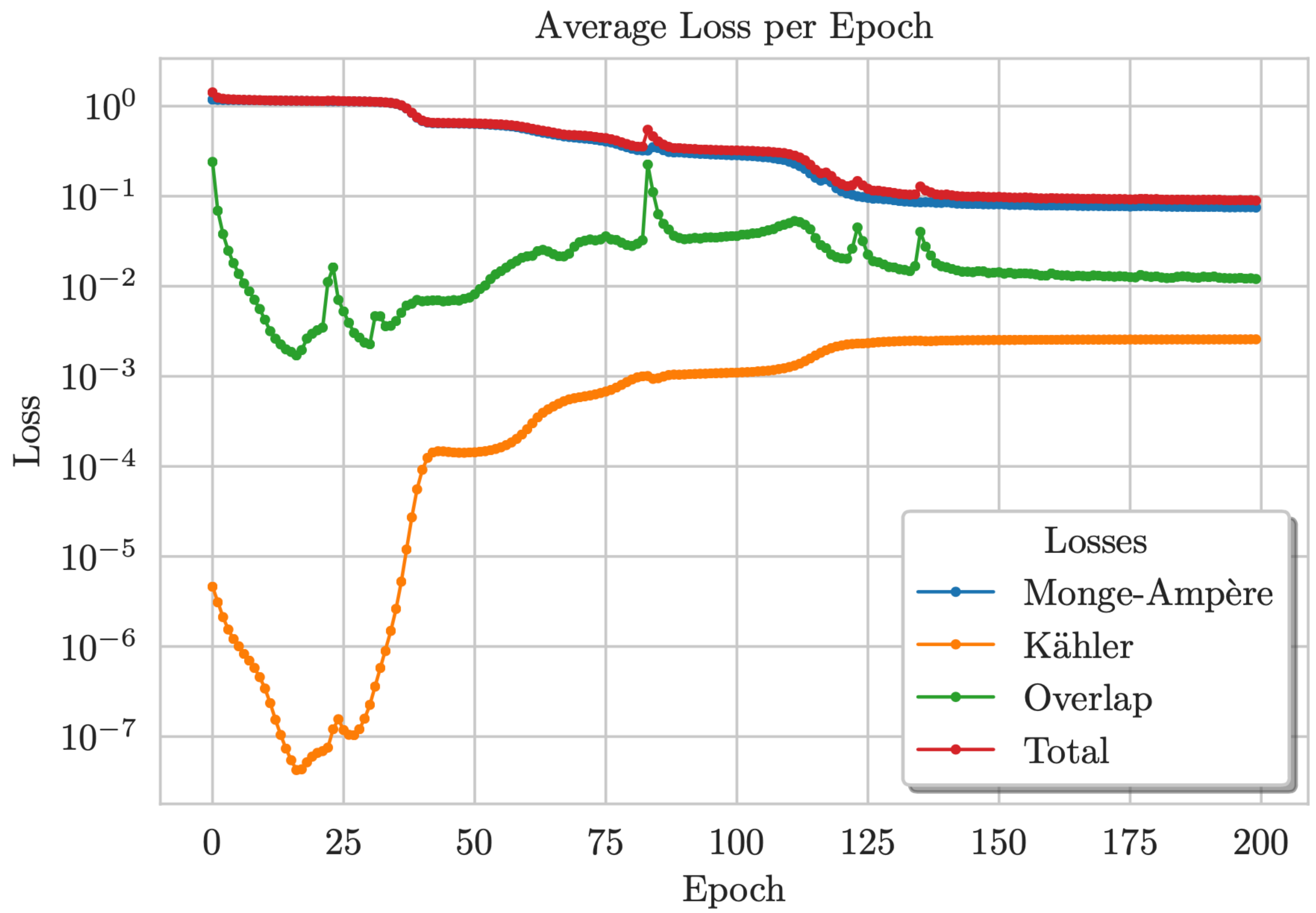
- Now: $dJ = 0$ and overlap need to be checked. $\vec{z}$ part of input.

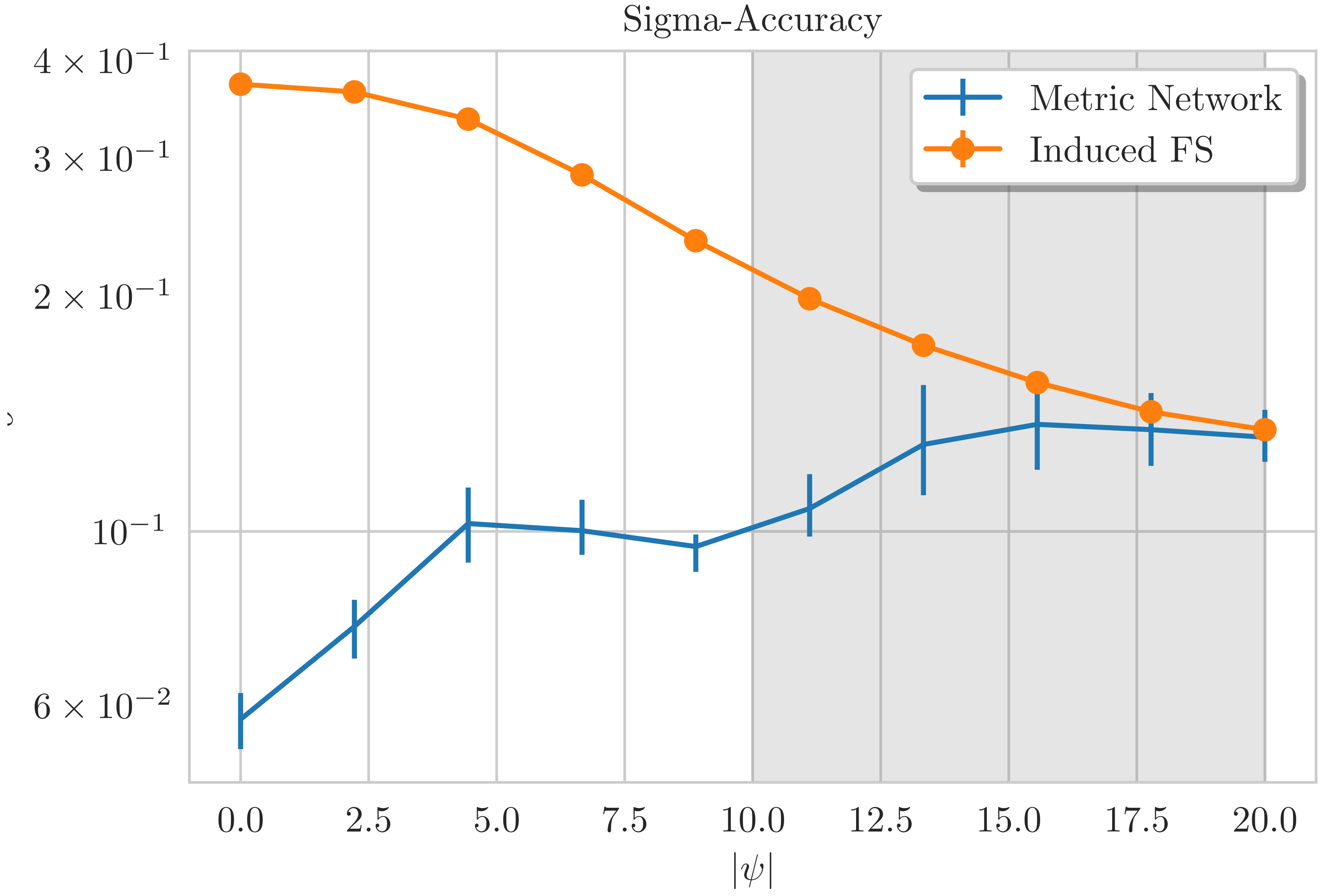
$$\mathscr{L} = \lambda_1 \mathscr{L}_{\text{MA}} + \lambda_2 \mathscr{L}_{\text{dJ}} + \lambda_3 \mathscr{L}_{\text{overlap}}$$

- Neural networks as perturbation to induced Fubini-Study metric (satisfying overlap and Kählerity condition)

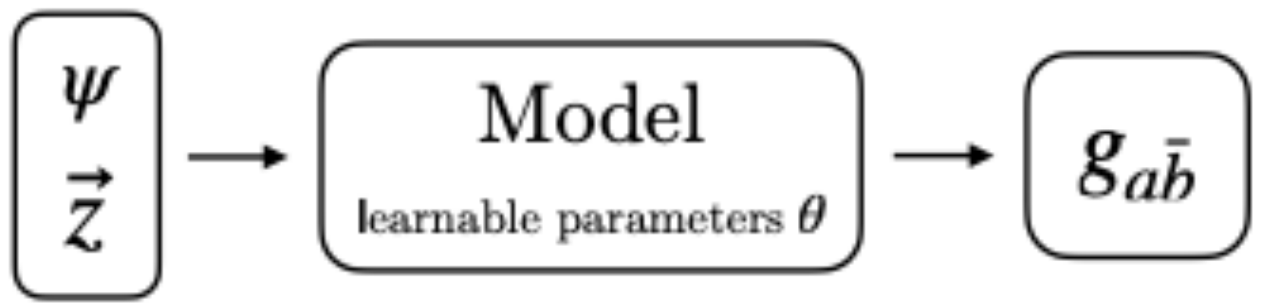
$$g_{\text{CY}} = g_{\text{FS}}(1 + g_{\text{NN}})$$

- Standard feed-forward neural network with relatively small initialisations.
- Metric networks converge, σ-accuracy improves, deviating from Fubini-Study





Learning g



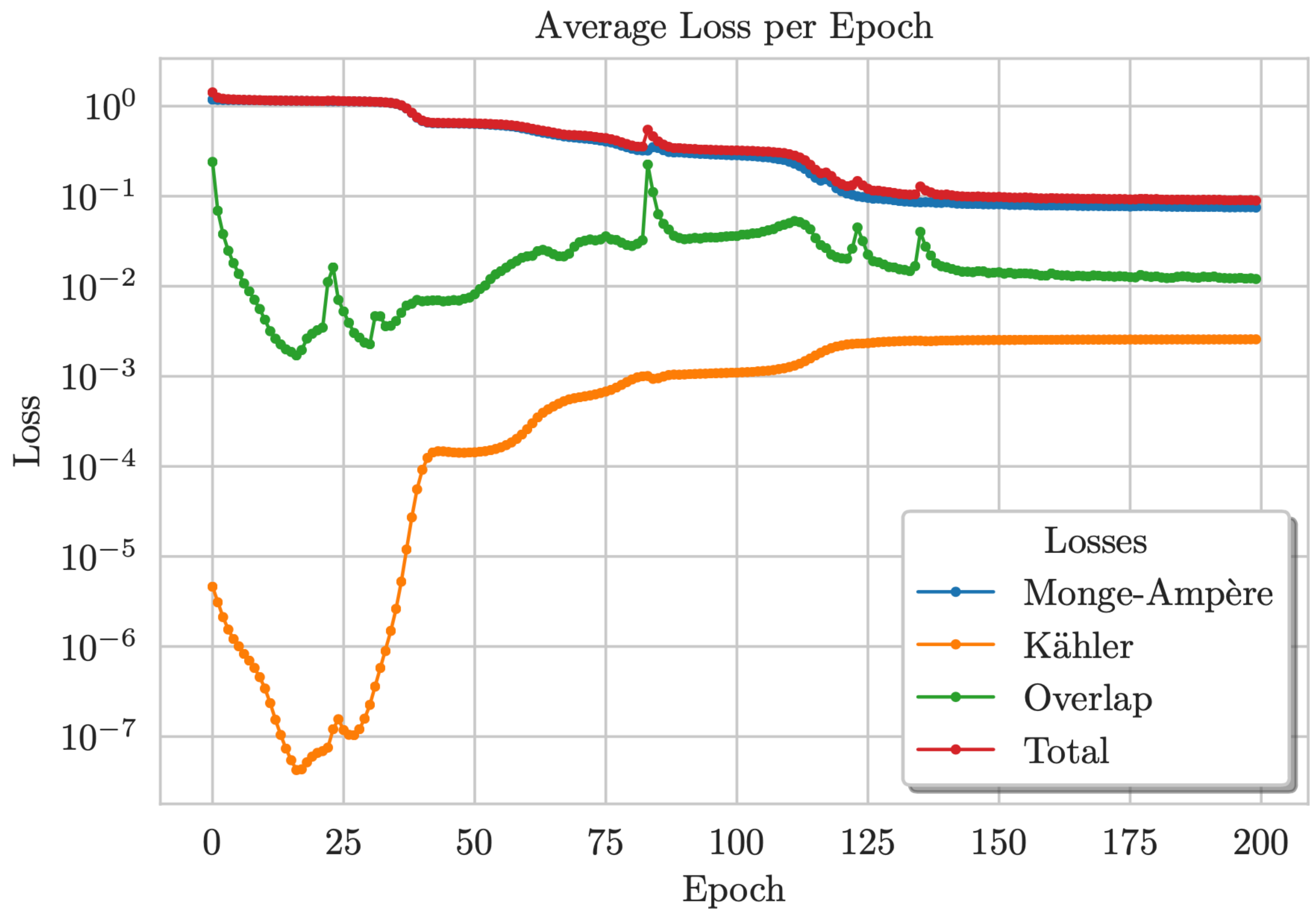
- Now: $dJ = 0$ and overlap need to be checked. $\vec{z}$ part of input.

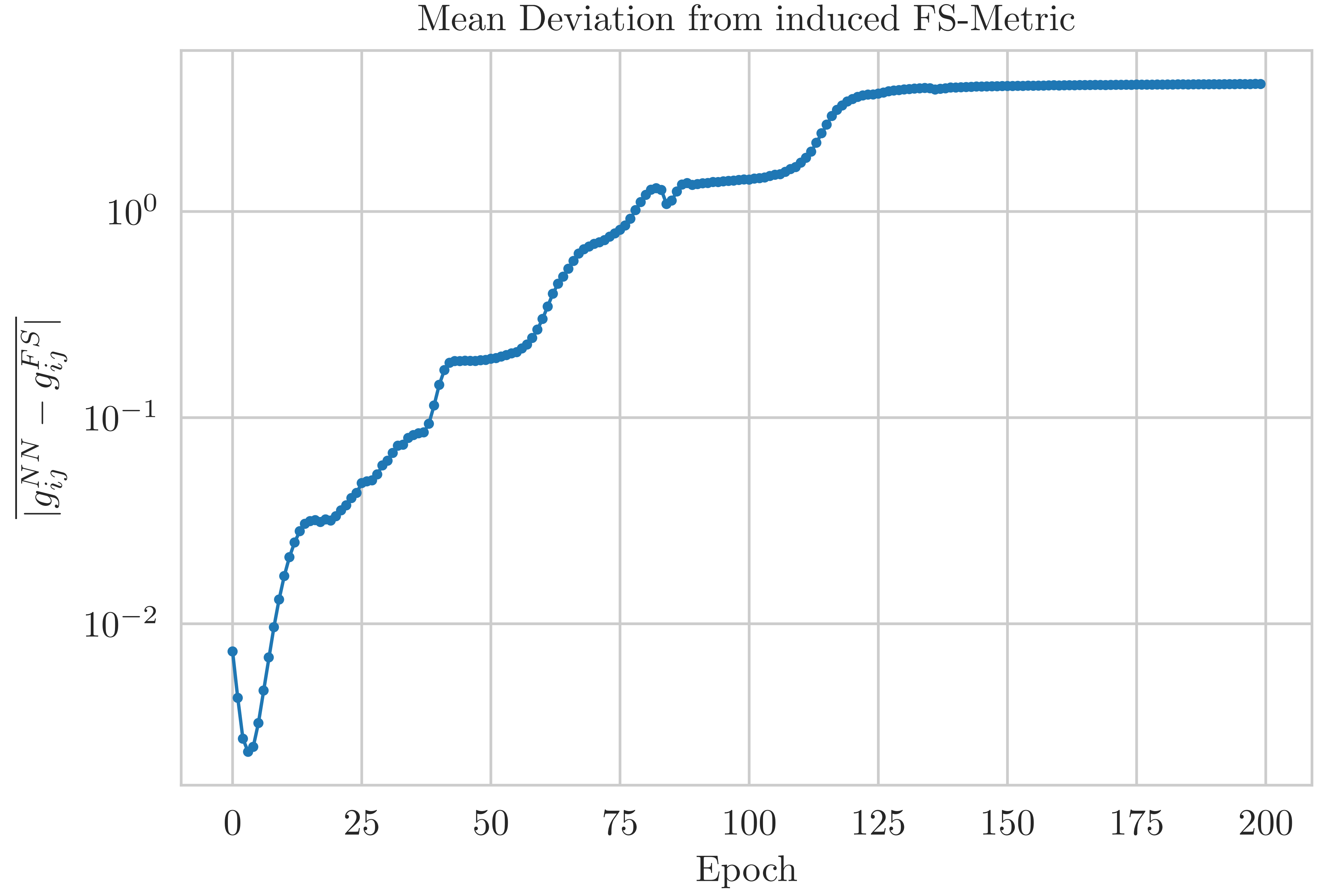
  $$\mathscr{L} = \lambda_1 \mathscr{L}_{\mathrm{MA}} + \lambda_2 \mathscr{L}_{\mathrm{dJ}} + \lambda_3 \mathscr{L}_{\mathrm{overlap}}$$

- Neural networks as perturbation to induced Fubini-Study metric (satisfying overlap and Kählerity condition)

  $$g_{\mathrm{CY}} = g_{\mathrm{FS}}(1 + g_{\mathrm{NN}})$$

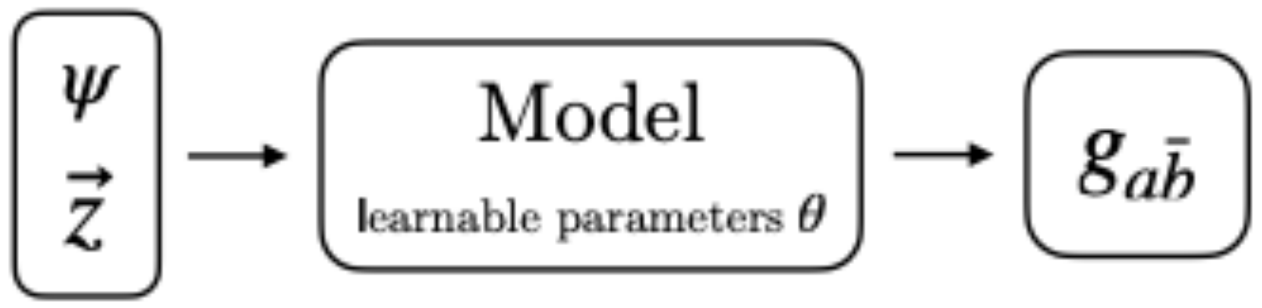
- Standard feed-forward neural network with relatively small initialisations.
- Metric networks converge, σ-accuracy improves, deviating from Fubini-Study

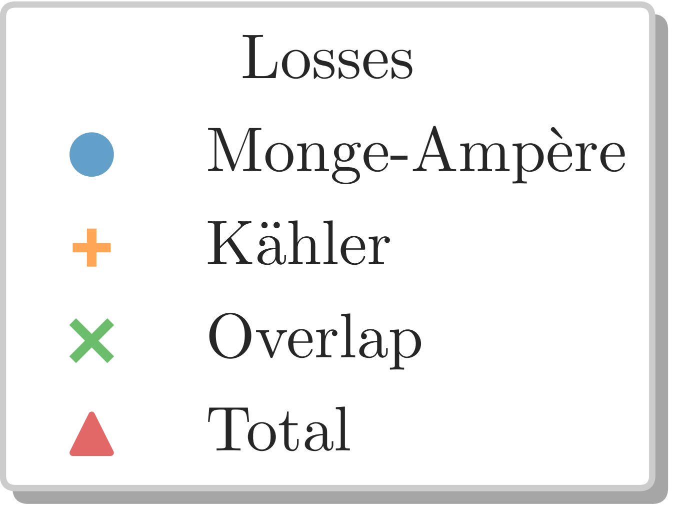




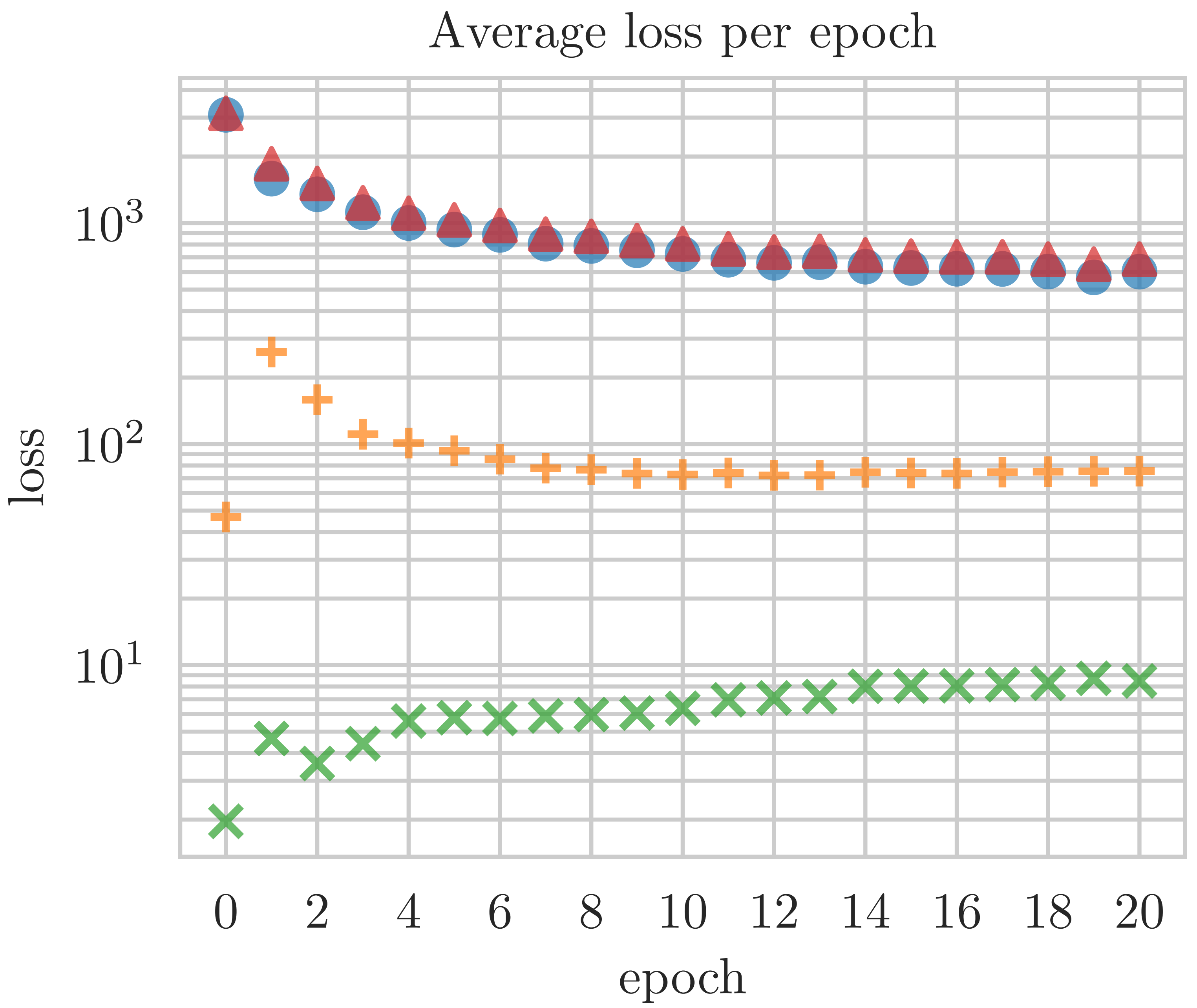
Metric loss components

Influence of different components

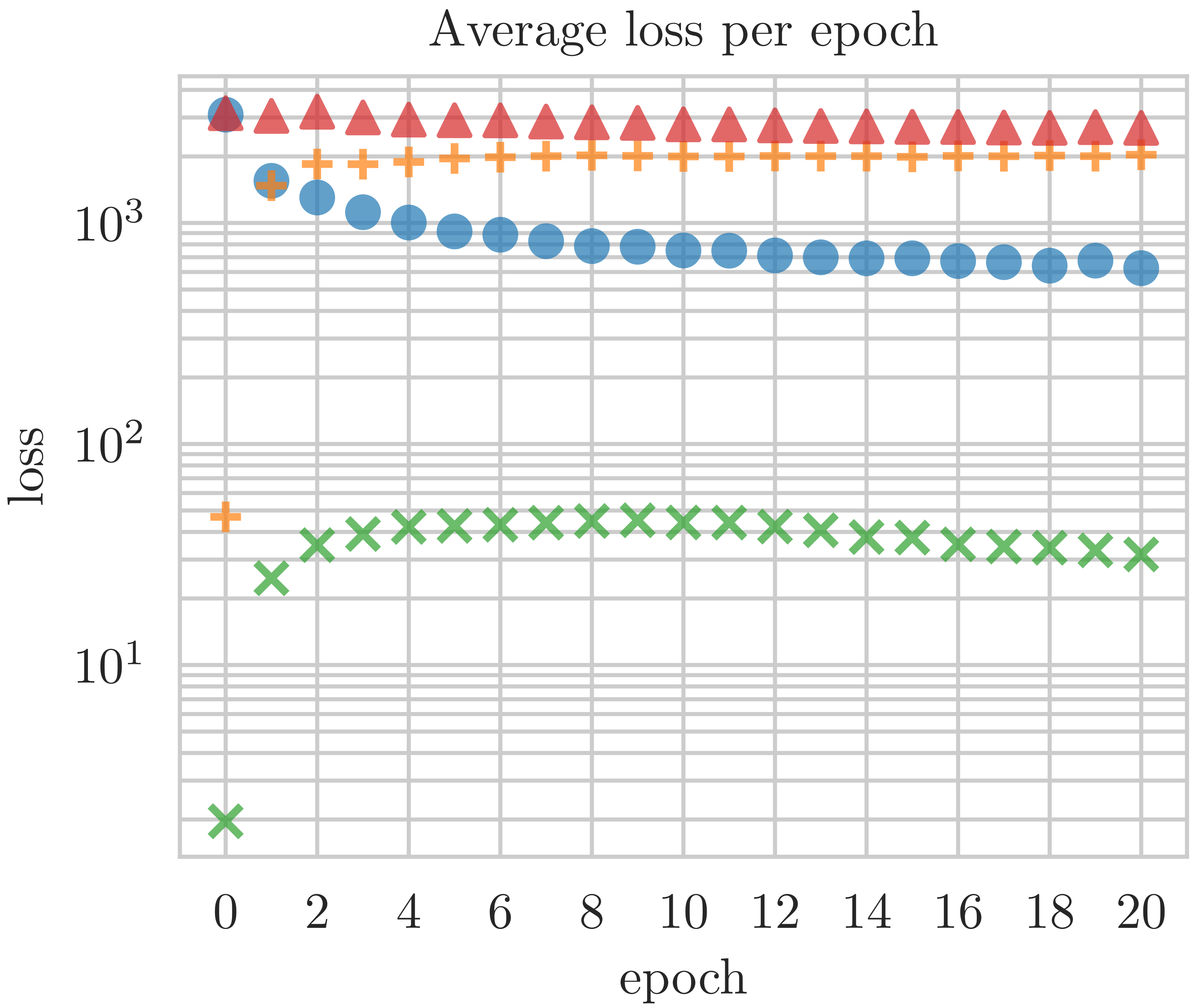




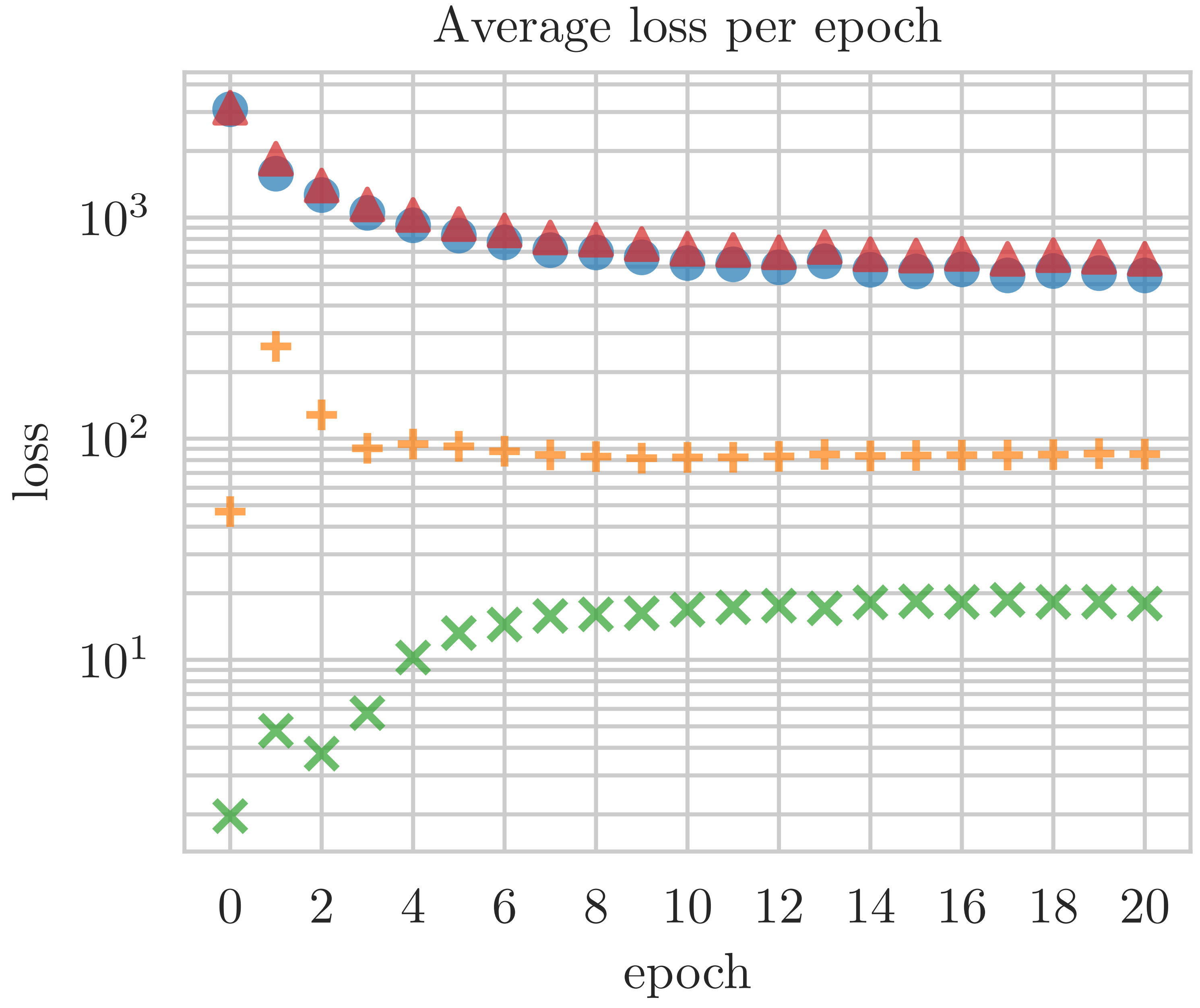
all losses



no Kähler loss ($\lambda_2 = 0$)



no overlap loss ($\lambda_3 = 0$)



$\psi = 10$

$$\mathscr{L} = \lambda_1 \mathscr{L}_{\mathrm{MA}} + \lambda_2 \mathscr{L}_{\mathrm{dJ}} + \lambda_3 \mathscr{L}_{\mathrm{overlap}}$$

g-Networks for New Classes of Solutions

- Approach of learning g allows to ask for metrics with different properties (not covered with previous numerical approaches).
- Philosophy: modified loss functions, additionally learned outputs.
- Can we augment the landscape of metrics to G2 and SU(3) structure manifolds? Phenomenologically necessary, otherwise missing large parts of string theory constructions; unexplored mathematical structures.
- Here: example SU(3) structure manifolds

$$J \wedge J \wedge J = \frac{3}{4} i\Omega \wedge \overline{\Omega} \ , \ J \wedge \Omega = 0, \ dJ = -\frac{3}{2}\text{Im}(W_1\overline{\Omega}) + W_4 \wedge J + W_3 \ ,$$

$$d\Omega = W_1 J \wedge J + W_2 \wedge J + W_5 \wedge \Omega, \ W_3 \wedge J = W_3 \wedge \Omega = W_2 \wedge J \wedge J = 0$$

Our SU(3) structure metrics

- Example of subset of torsion classes (Strominger-Hull system):

$$W_1 = W_2 = 0,\ W_4 = \frac{1}{2}W_5 = d\phi,\ W_3 \text{ arbitrary}$$

- Simple ansatz for metric and 3-form:

$$J = \sum_i^m a_i J_i \ ,\ \Omega = A_1\Omega_0 + A_2\overline{\Omega}_0$$

$$|A_1|^2 + |A_2|^2 = \sum_{i,j,k=1}^m \Lambda_{ijk} a_i a_j a_k\ ,\ J_i \wedge J_j \wedge J_k = \frac{3}{4} i \Lambda_{ijk} \Omega_0 \wedge \overline{\Omega}_0$$

- Special ansatz for W_i:

$$W_1 = W_2 = W_3 = 0\ ,\ W_5 = 2W_4 = 2d(\ln a_1)\ ,\ a_1 = \frac{1}{\pi^3} \frac{|\nabla p_\psi|^2}{(\sum |X_a|^2)^4}$$

SU(3) structure experiment

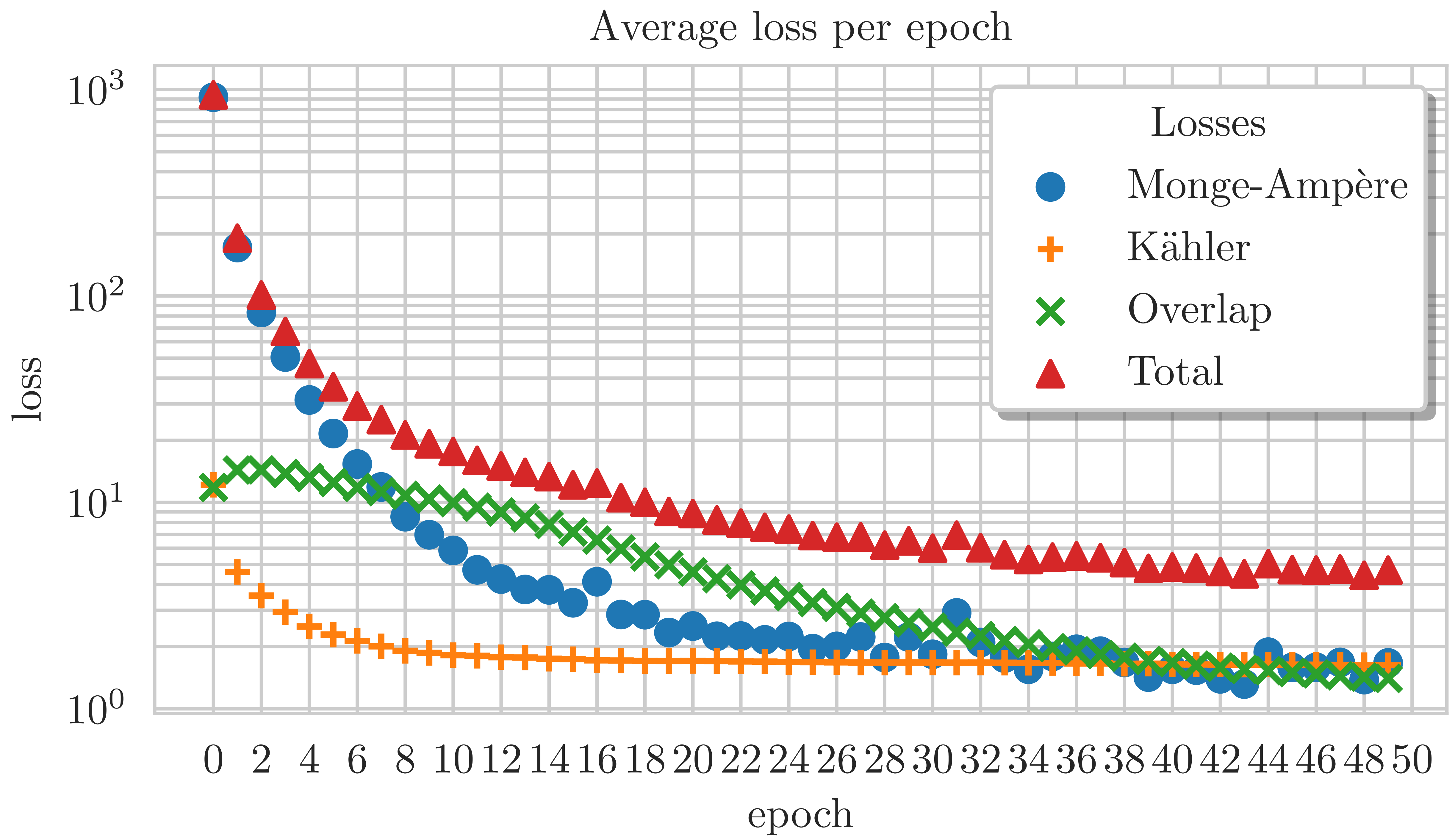
- Ansatz with known solution:

$$W_4 = d \log a_1$$

- Adapted Kähler loss:

$$\mathscr{L}'_{W_4} = ||dJ - d \ln a_1 \wedge J||_n$$

- Does the network converge to known solution? Yes.



Status of Metrics

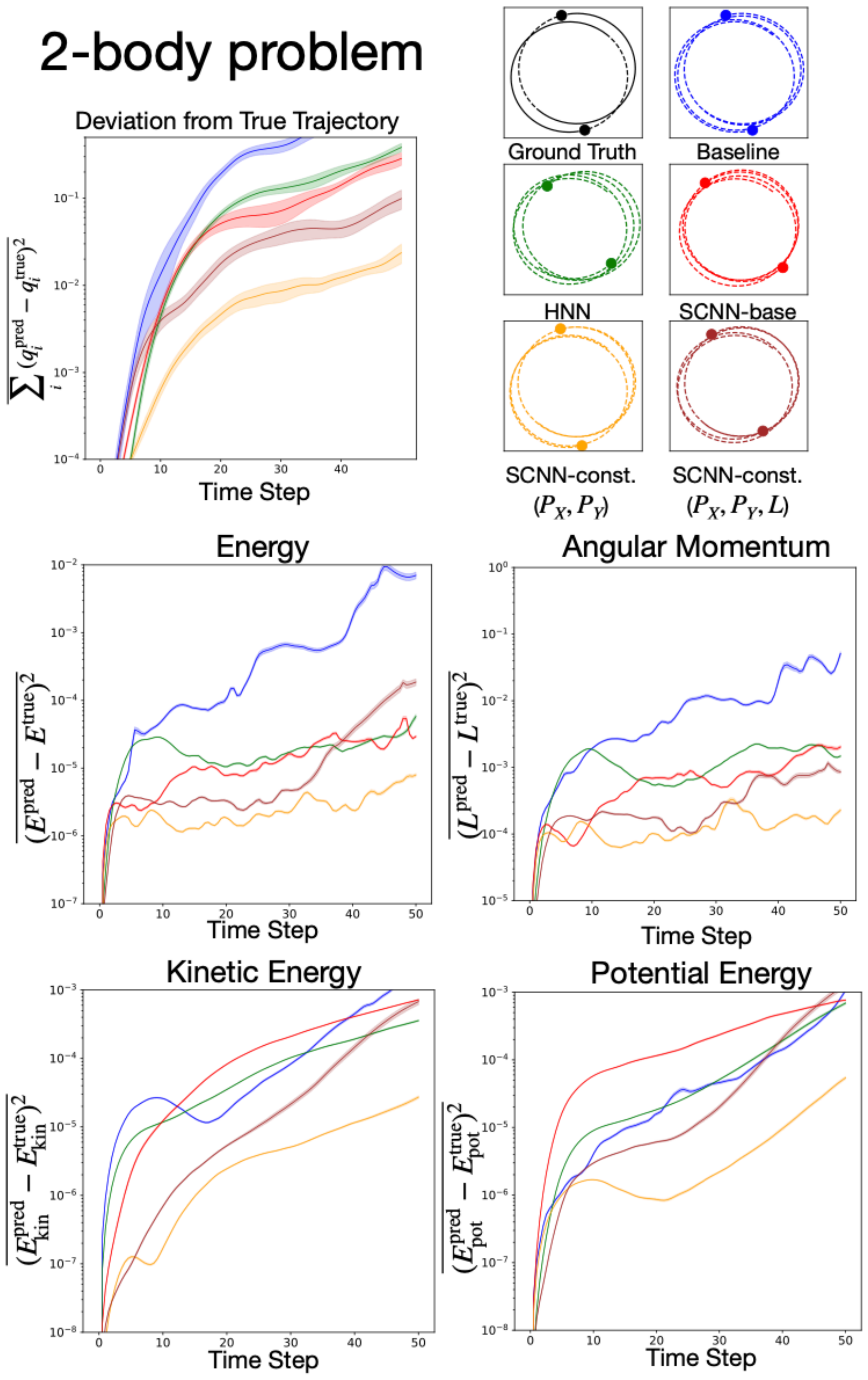
- Very little is known on interesting EFT questions due to the lack of results on the compactification metrics
- Generically: good accuracy requires computational effort, largely unfeasible with previous methods (e.g. single point in moduli space ~ day on desktop computer [k=12])

	Donaldson, Headrick & Nassar	Kähler potential	Metric Directly
Fixed point in Moduli Space	✔	✔	✔
Moduli Dependence	✗ (interpolation)	✔	✔
Non Kähler	✗	✗	✔
Analytic	✗	✗	✗

Analytic formulae from NN?

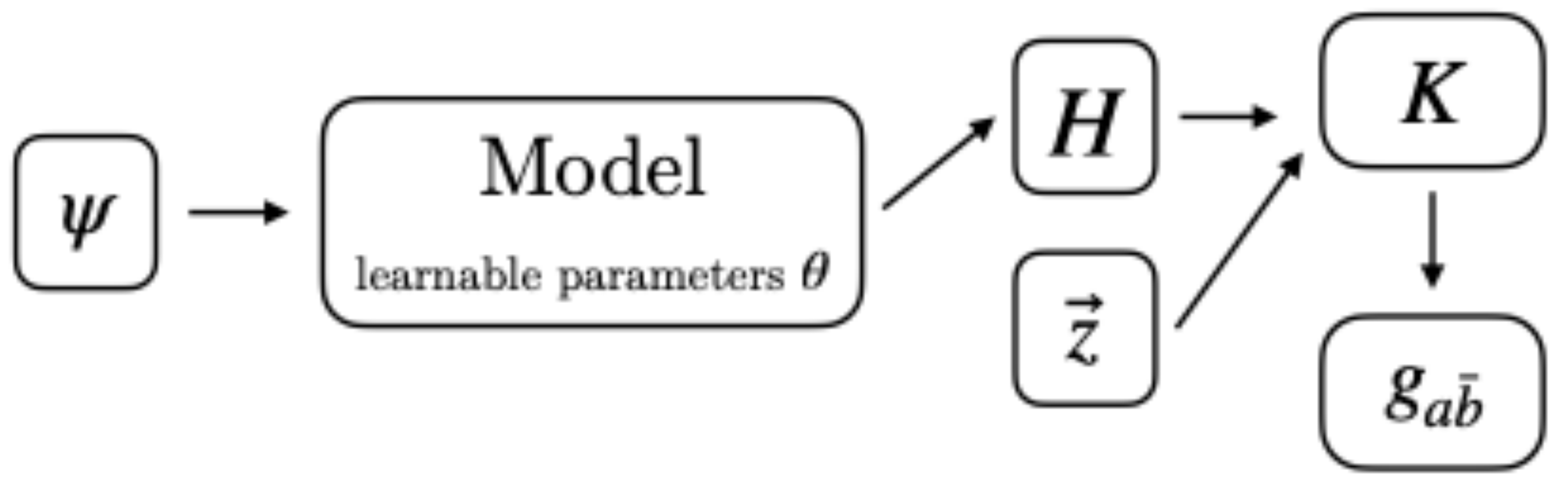
- Can we find analytic expressions for these metrics?
- Hopeful, as other physics examples show that it is possible
  (Cranmer, Xu, Battaglia, Ho; Sahoo, Lampert, Martius; Wetzel, Melko, Scott, Panju, Ganesh)
- Example (work with Marc Syvaeri): Inferring Hamiltonian and Conserved Quantities from simulated data of physical systems

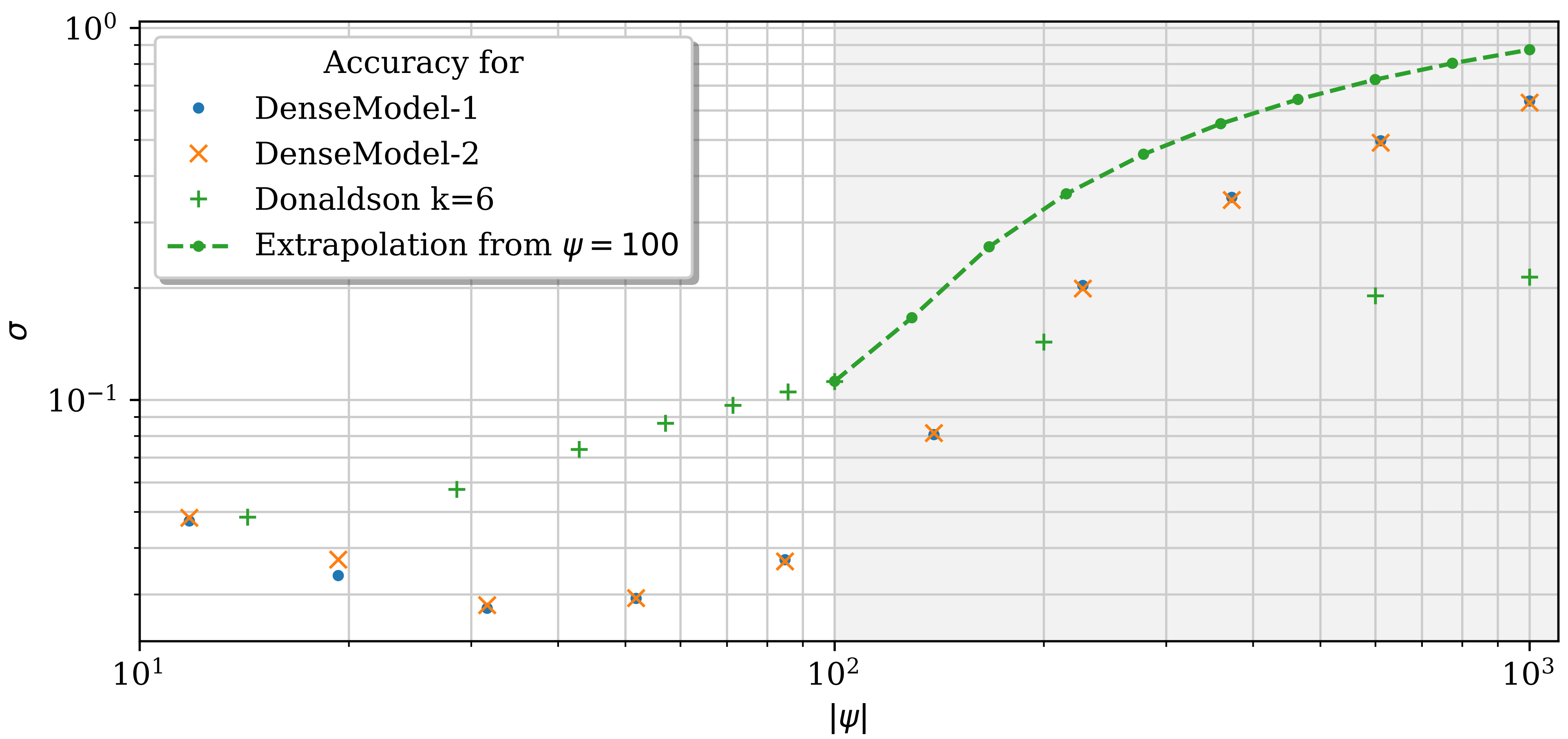
$$
\begin{aligned}
P_{c1} &= -\,4.21\ p_{x1} - 4.21\ p_{x2} - 1.26\ p_{y1} - 1.29\ p_{y2} \qquad (0.03)\ , \\
P_{c2} &= -\,0.93\ p_{x1} - 0.92\ p_{x2} - 3.23\ p_{y1} - 3.22\ p_{y2} \qquad (0.03)\ , \\
L &= -\,1.07\ q_{x1}p_{y1} + 0.88\ q_{x1}p_{y2} + 0.93\ q_{x2}p_{y1} - 1.03\ q_{x2}p_{y2} \\
&\quad + 1.01\ q_{y1}p_{x1} - 0.89\ q_{y1}p_{x2} - 0.92\ q_{y2}p_{x1} + 0.99\ q_{y2}p_{x2} \qquad (0.10)\ .
\end{aligned}
$$

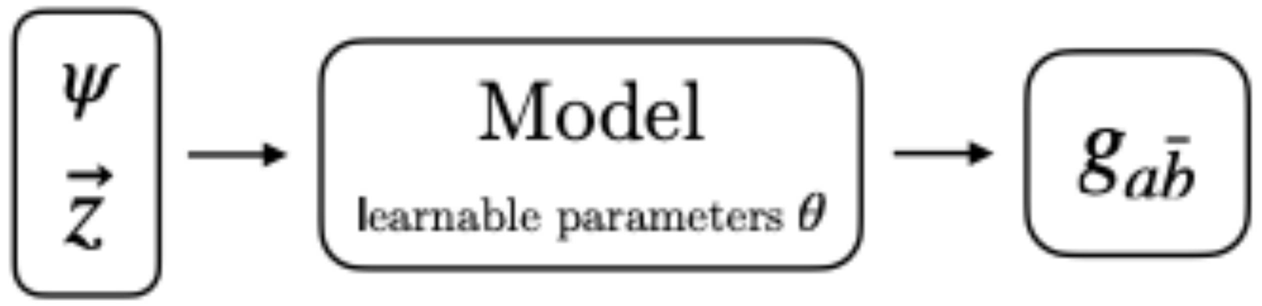


Conclusions

- Learning CY metrics with NNs works and is more efficient.
- Moduli dependent metrics (here: complex structure)
- Auto-differentiation for loss functions depending on derivatives of metric
- New types of metrics are within reach (SU(3) structure, G2)
- Applications in physics and mathematics, e.g.: EFTs in string theory (non-holomorphic quantities), SYZ-conjecture (are CYs T^3-fibrations at large CS)
- A lot of physics and mathematics ahead for future string_data meetings!



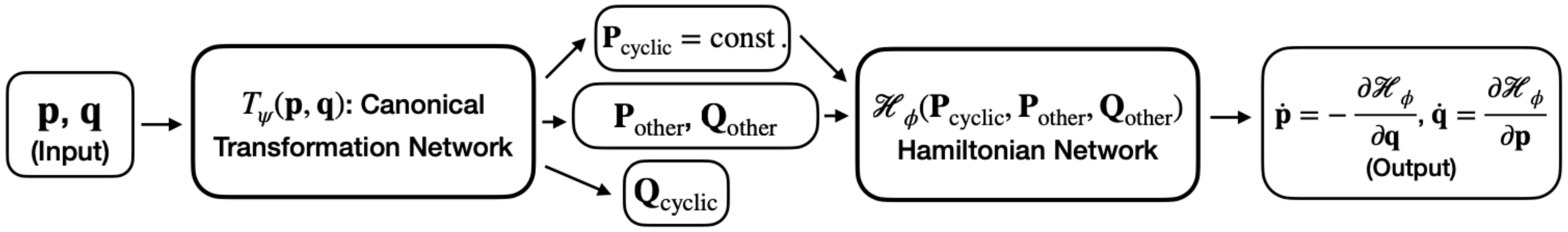




Thank you!

Learning Hamiltonian and Conserved Quantities

Additional information



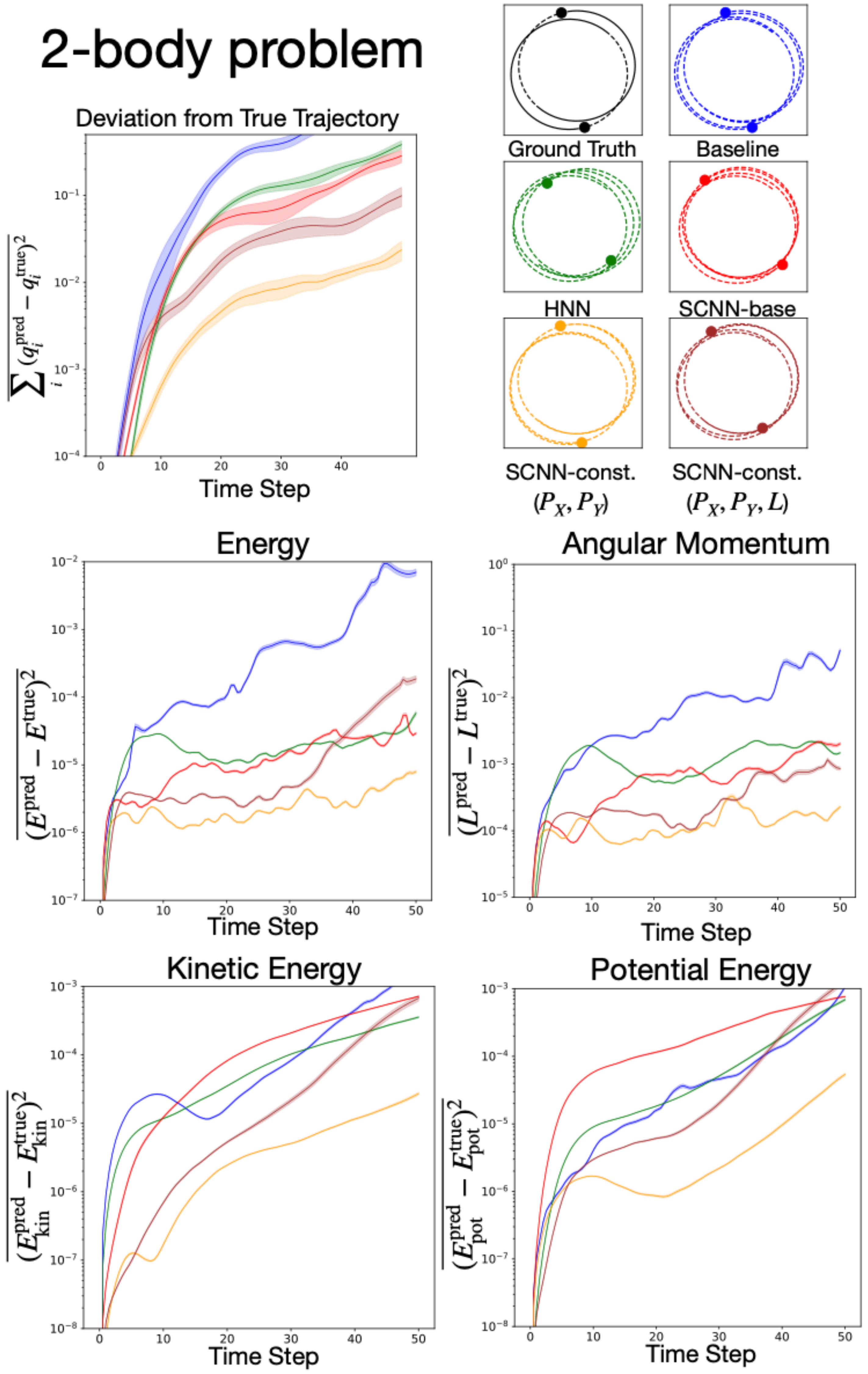
$$\mathcal{L}_{\text{HNN}} = \sum_{i=1}^{N\cdot d} \left\| \frac{\partial \mathcal{H}_\phi(\mathbf{P},\mathbf{Q})}{\partial p_i} - \frac{dq_i}{dt} \right\|_2 + \left\| \frac{\partial \mathcal{H}_\phi(\mathbf{P},\mathbf{Q})}{\partial q_i} + \frac{dp_i}{dt} \right\|_2 .$$

$$\mathcal{L}_{\text{HQP}}^{(n)} = \sum_{i=1}^{n} \left\| \frac{dP_i}{dt} \right\|_2 + \left\| \frac{dQ_i}{dt} - \frac{\partial \mathcal{H}_\phi(\mathbf{P},\mathbf{Q})}{\partial P_i} \right\|_2 + \beta \sum_{i=n+1}^{N\cdot d} \left\| \frac{dP_i}{dt} + \frac{\partial \mathcal{H}_\phi(\mathbf{P},\mathbf{Q})}{\partial Q_i} \right\|_2 + \left\| \frac{dQ_i}{dt} - \frac{\partial \mathcal{H}_\phi(\mathbf{P},\mathbf{Q})}{\partial P_i} \right\|_2$$

$$\mathcal{L}_{\text{Poisson}} = \sum_{i,j=1}^{N\cdot d} \| \{Q_i, P_j\} - \delta_{ij} \|_2 + \sum_{i,j>i}^{N\cdot d} \| \{P_i, P_j\} \|_2 + \| \{Q_i, Q_j\} \|_2$$

$$P_x = p_{x1} + p_{x2} \,, \qquad P_y = p_{y1} + p_{y2} \,,$$
$$L = (p_{x1} - p_{x2})(q_{y1} - q_{y2}) - (p_{y1} - p_{y2})(q_{x1} - q_{x2}) \,.$$

$$\mathcal{H} = \frac{p_{x1}^2}{2m_1} + \frac{p_{y1}^2}{2m_1} + \frac{p_{x2}^2}{2m_2} + \frac{p_{y2}^2}{2m_2} - \frac{g}{\|\mathbf{q}_1 - \mathbf{q}_2\|_2} \,.$$



Our SU(3) structure metrics

- General solutions:

$$W_1 = 0$$

$$W_2 = -i\bar{\partial}A_1 \lrcorner \Omega_0 + i\partial A_2 \lrcorner \overline{\Omega}_0 + i\frac{\bar{\partial}(A_1 + \bar{A}_2)}{A_1 + \bar{A}_2} \lrcorner A_1 \Omega_0 - i\frac{\partial(\bar{A}_1 + A_2)}{\bar{A}_1 + A_2} \lrcorner A_2 \overline{\Omega}_0$$

$$W_3 = \sum_i (da_i - W_4) \wedge J_i$$

$$W_4 = \frac{1}{2} \sum_i J_i \lrcorner (da_i \wedge J_i)$$

$$W_5 = \frac{\bar{\partial}(A_1 + \bar{A}_2)}{A_1 + \bar{A}_2} + \frac{\partial(\bar{A}_1 + A_2)}{\bar{A}_1 + A_2}$$

$W_1 = W_2 = 0,\ W_4 = \frac{1}{2}W_5 = d\phi,\ W_3$ arbitrary

Network Layouts

Layer	Number of Nodes	Activation	Number of Parameters
input	3	–	–
hidden 1	100	leaky ReLU	400
hidden 2	1000	leaky ReLU	101 000
hidden 3	1000	leaky ReLU	1 001 000
output	N_k^2	identity	$1000 \times N_k^2 + N_k^2$

Layer	Number of Nodes	Activation	Number of Parameters
input	17	–	–
hidden 1	100	leaky ReLU	1800
hidden 2	100	leaky ReLU	10 100
hidden 3	100	leaky ReLU	10 100
output	d^2	identity	101 d^2

Layer	number nodes	Activation	Regularization	Initialization
input	10	–	–	–
hidden 1	1000	ReLU	L2(10^{-6})	$\mathcal{N}_k(0, 10^{-4})$, $\mathcal{N}_k(0, 10^{-3})$
hidden 2	1000	ReLU	L2(10^{-6})	$\mathcal{N}_k(0, 10^{-4})$, $\mathcal{N}_k(0, 10^{-3})$
hidden 3	1000	ReLU	L2(10^{-6})	$\mathcal{N}_k(0, 10^{-4})$, $\mathcal{N}_k(0, 10^{-3})$
output	9	–	L2(10^{-4})	$\mathcal{N}_{k,b}(0, 10^{-2})$