Machine Learning Symmetries in Physics from First Principles

Konstantin Matchev

University of Alabama

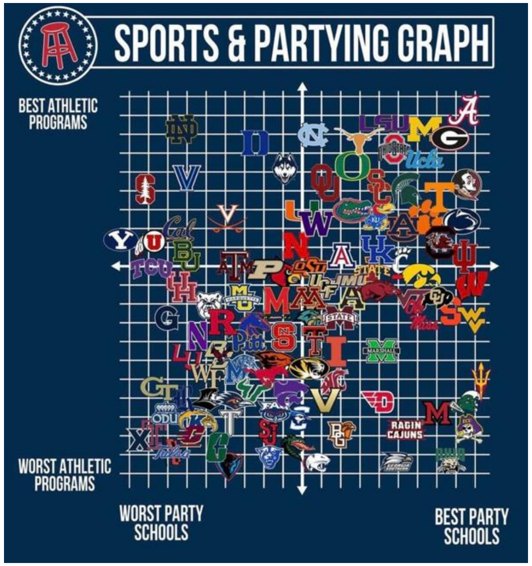
With: Roy Forestano, Katia Matcheva, Alexander Roman, Eyup Unlu, Sarunas Verner

Based on: arXiv:2301.05638, Machine Learning: Science and Technology (2023) **4** 025027
arXiv:2302.00806, Symmetry (2023) **15** (7) 1352
arXiv:2302.05383, Physics Letters B (2023) **844** 138086
arXiv:2307.04891, Physics Letters B, (2023) **847** 138266
arXiv:2309.07860, Physics Letters B, (2023) **847** 138306

Physics Seminar
University of Rome
May 27 2025

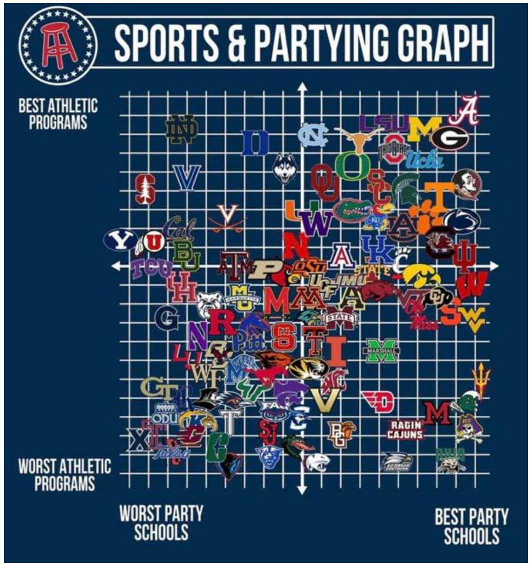
Three great things about UA

3. Better football



Three great things about UA

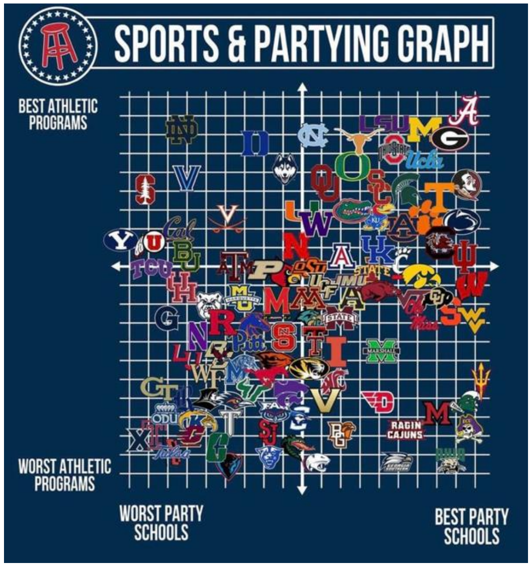
3. Better football



2. Cuter mascot

Three great things about UA

3. Better football



2. Cuter mascot

1. UA strives to be the Quantum AI University

INSTITUTES & CENTERS

UA Launches New Center for AI Research and Development

UA is setting a new standard for AI research as part of a campus-wide collaboration.

References

Deep learning symmetries and their Lie groups, algebras and subalgebras from first principles

Oracle preserving latent flows

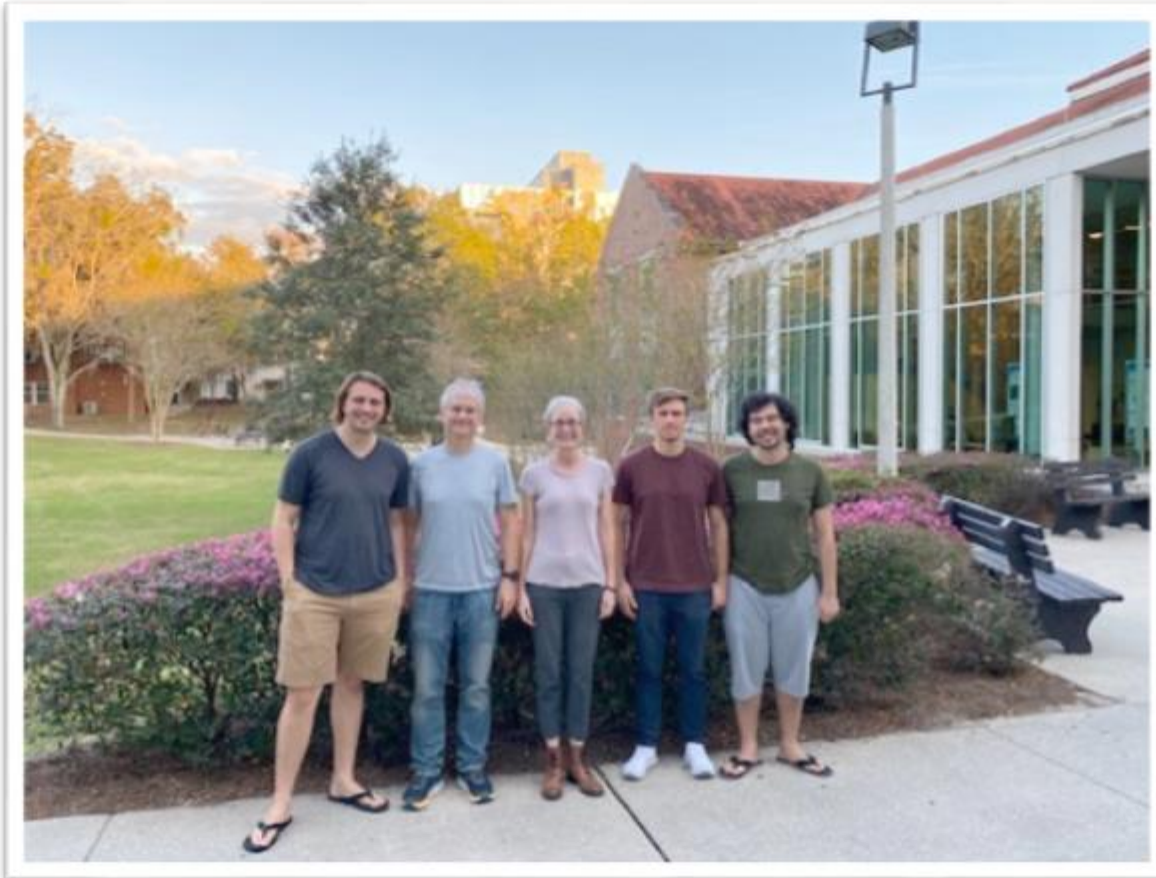
Discovering sparse representations of Lie groups with machine learning

Accelerated discovery of machine-learned symmetries: deriving the exceptional groups G_2, F_4 and E_6

Identifying the group-theoretic structure of machine-learned symmetries

Python code on github

Who are we?



Florida, January 2023

Pittsburgh, Pheno, May 2023

This talk is being given

- by a phenomenologist, who is a theorist and a member of an LHC experiment.
  - what is a "phenomenologist"?

The experimentalist asks:

The theorist answers:

Is it possible to have a theory model which gives signature X? → Yes.

Are there any well motivated such models? → No. / You bet. Let me tell you about those. Actually I have a paper…

Is there any Monte Carlo which can simulate those models? → I'm the wrong person to ask. Ask a phenomenologist.

Slide circa 2005

Nowadays the tables have turned

- The stream of LHC data has changed the picture

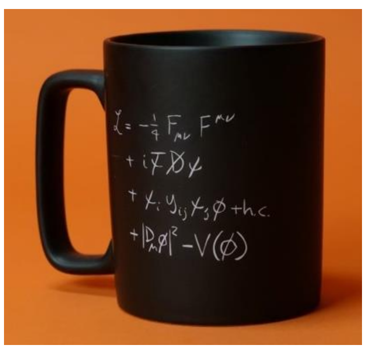
Experimentalist answers:		**The theorist asks:**
Yes.	←	Can LHC be sensitive to model X?
Not this particular model. In our note we only show MSUGRA plots.	←	Is there any analysis which is looking for this model?
Manpower. But talk to a phenomenologist...	←	Why not?! It's a great model.

Slide circa 2015

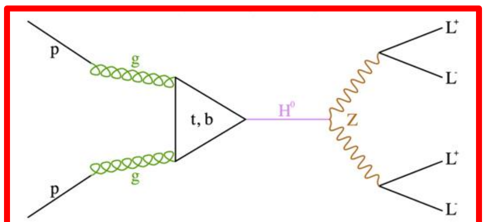
The simulation pipeline in HEP

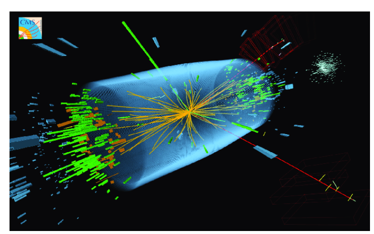
Theory Model → Model Interpreter → Parton-level Calculator → Event Generator → Detector Simulator → Data Analysis

- Symmetries
- Representations
- Fields (basis)
- Lagrangian
- Aesthetics

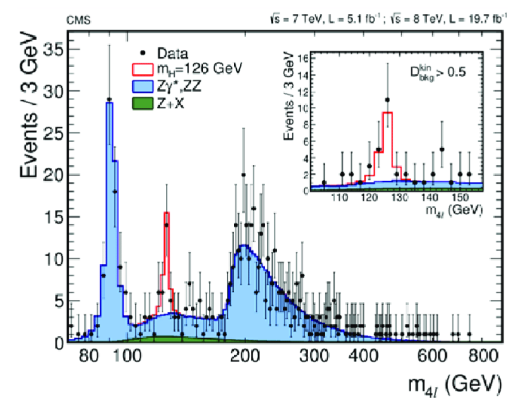


- RGEs
- Vacuum structure
- Mass spectrum
- Feynman rules
- Feynman diagrams
- Cross sections (rates)
- Branching fractions (rates)
- MC event generation
- Fragmentation
- Hadronization
- Higher Order Corrections

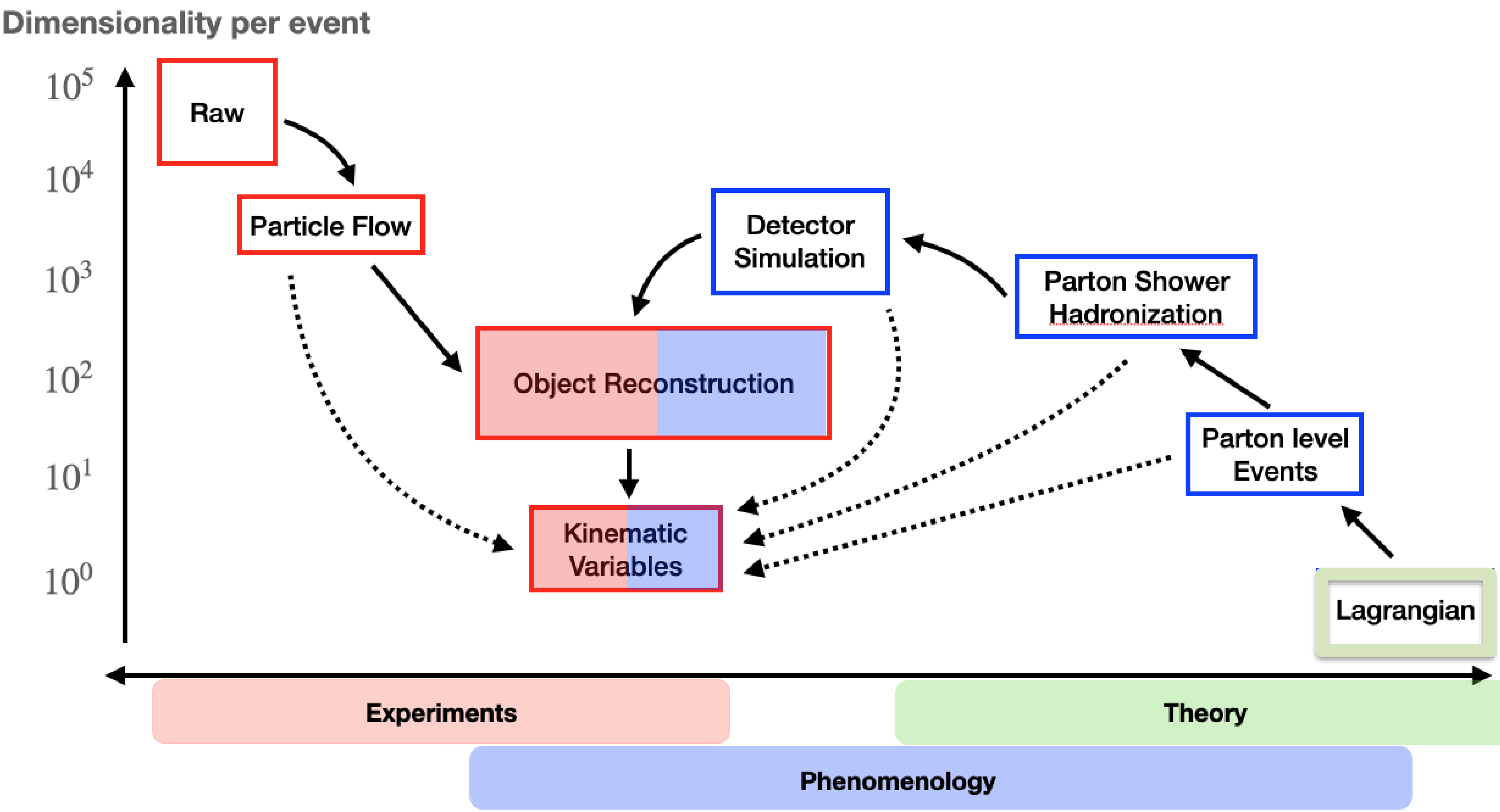




- Resolution
- RecoObjects
- Histograms
- Backgrounds
- Selection cuts
- ROC curve
- Error analysis



The curse of dimensionality



Franceschini,Kim,Kong,KM,Park,Shyamsundar Rev.Mod.Phys. 2023

NOW BACK TO THE COLLOQUIUM

Outline

- Why Machine Learning? My personal story
- Symmetries in every day life, in math, and in physics
  - Group theory from a physicist's point of view
- Discovering continuous symmetries with machine learning (little or no Lie group theory background needed)
  - Basic ingredients
    - Math: vector spaces, operations, invariants
    - ML: features, neural networks, labels (oracles)
  - Loss functions
- Examples (too many to fit in a single talk)
  - Orthogonal groups O(n) (Euclidean rotations)
  - Lorentz transformations in 4D space-time
  - Unitary groups U(n)
  - Exceptional groups: G_2, F_4, E_6.
  - Symmetries in a latent (reduced dimensionality) space

Machine Learning versus Data Science

- Data science is necessarily an **interdisciplinary** paradigm
  - The battle of the Venn diagrams

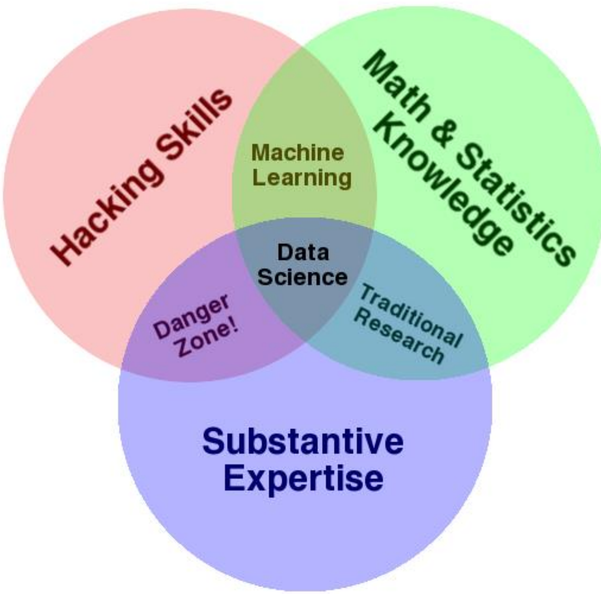
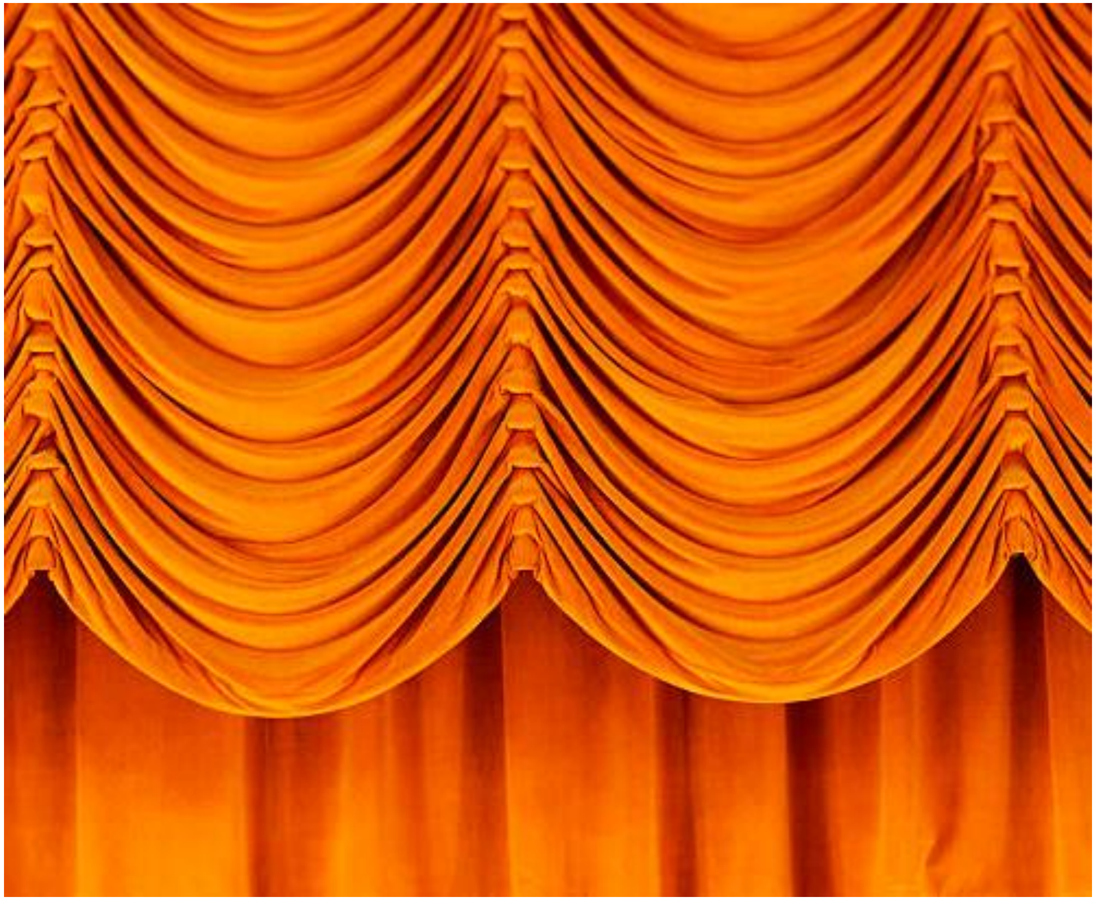


Figure 1a: Conway's Data Scientist Venn Diagram

http://drewconway.com/



Machine Learning versus Data Science

- Data science is necessarily an **interdisciplinary** paradigm
  - The battle of the Venn diagrams

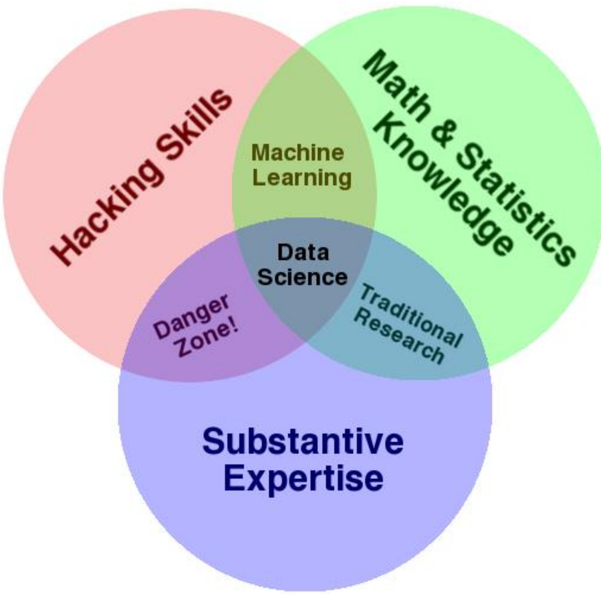


Figure 1a: Conway's Data Scientist Venn Diagram

http://drewconway.com/

Joel Grus

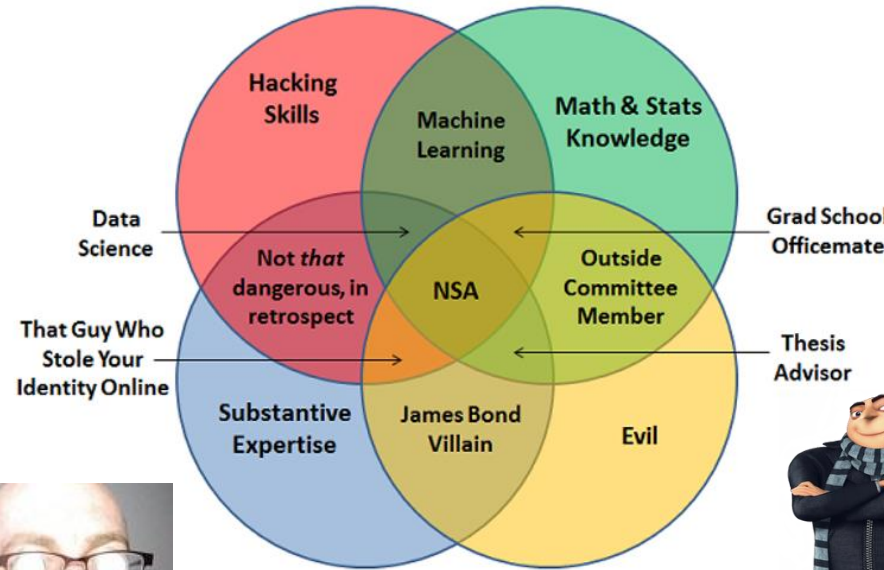


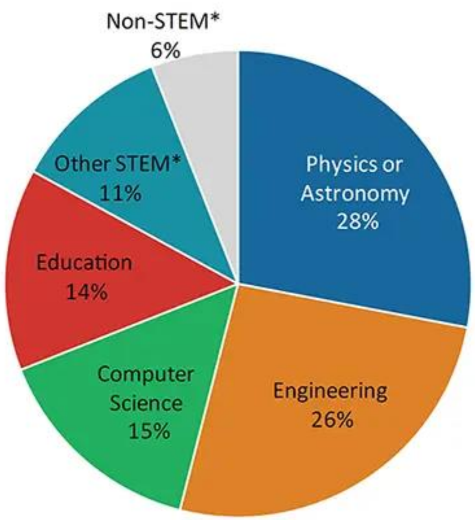
Figure 1b: Grus' Data Scientist Venn Diagram

Gru

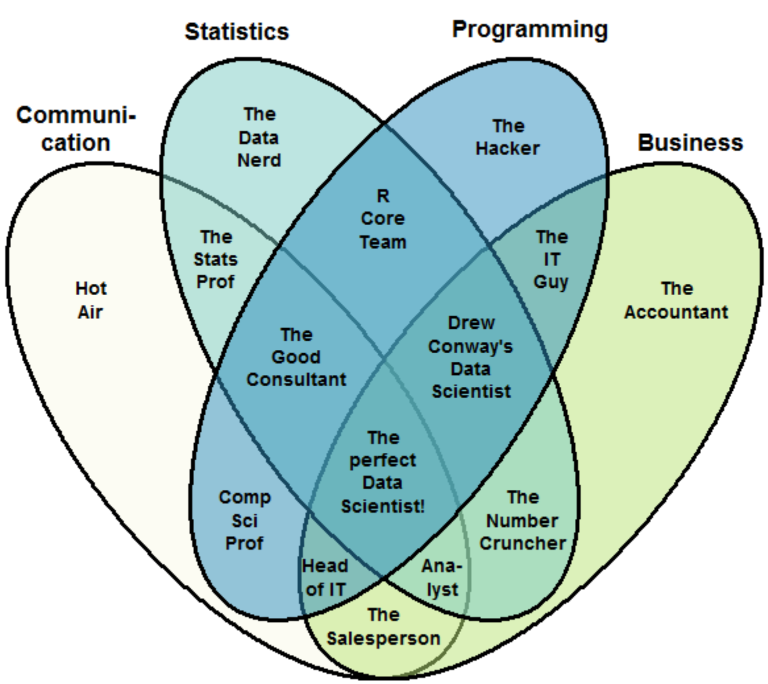
The Job Market

- Many of our students do not end up going into academia
  - **Data science skills** are important
  - **Communication skills** are also important

Fields of Employment of Exiting Physics Masters One Year After Degree, Classes of 2016, 2017, & 2018 Combined



The Data Scientist Venn Diagram



My ML course @ UF

https://datascience.stackexchange.com/users/2853/stephan-kolassa

Ariel Machine Learning Data Challenge

- In 2022, the Ariel Consortium organized a Machine Learning Data Challenge at the NeurIPS conference
  - Given an observed exoplanet transmission spectrum, find the planet's physical parameters and atmospheric composition
  - Our team **"Gators"** won first place and a $2,500 prize!
- In 2023 the "Gators" were again one of the winning teams


Ariel DATA CHALLENGE

CERTIFICATE OF ACHIEVEMENT

THIS CERTIFICATE IS AWARDED TO:

GATORS

FOR BEING THE 1ST PLACE WINNER OF THE REGULAR TRACK AND 2ND PLACE WINNER OF THE LIGHT TRACK OF THE ARIEL DATA CHALLENGE 2022. WE PRESENT YOU WITH THIS AWARD CERTIFICATE TO COMMEMORATE YOUR VALUABLE CONTRIBUTION TOWARDS ADVANCING THE STATE OF ART IN PLANETARY SCIENCE

07 December 2022
DATE

PROFESSOR GIOVANNA TINETTI

ASSOCIATE PROFESSOR INGO WALDMANN

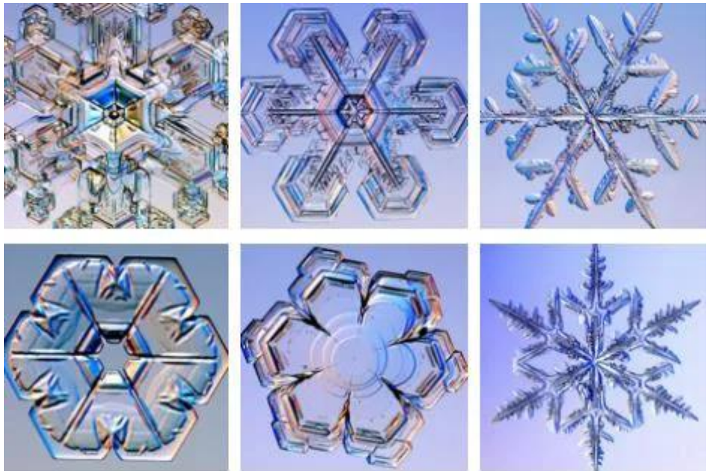
Yip Kai Hou.
DR GORDON YIP

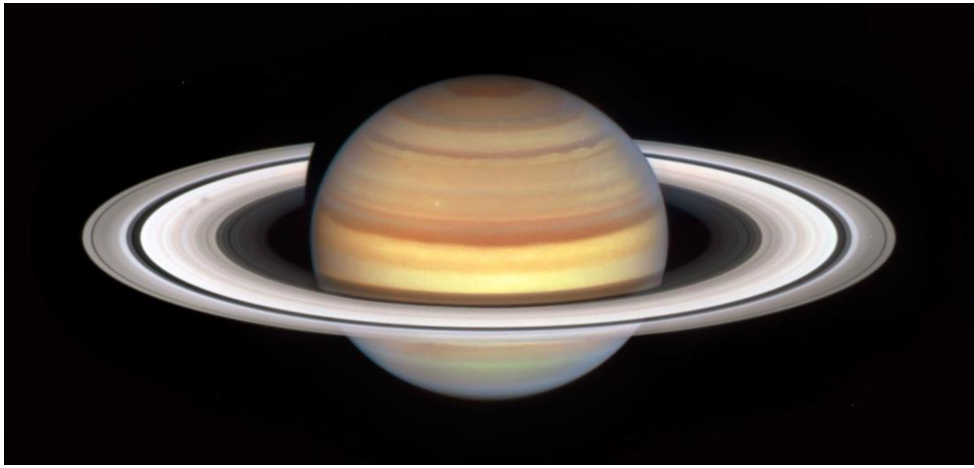
UCL DiRAC UK SPACE SPACEFLUX esa cea The Alan Turing Institute

Symmetries in every day life

☐Discrete

☑Continuous





Johnston et al., “*Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution*”, PNAS (2022)

Symmetry in science

- **Math:** "the term **symmetry** is used to refer to an object that is invariant under some transformations" (Wikipedia)
  - Key concepts: object, transformation, invariance
- **Physics:** "*it is only slightly overstating the case to say that physics is the study of symmetry*" (P. Anderson 1972)
  - Noether's theorem: ***every continuous symmetry of the action of a conservative system has a corresponding conservation law***.

Symmetry		Conservation law
Rotation	⟷	Angular momentum
Space translation	⟷	Momentum
Time translation	⟷	Energy


E. Noether

Group Theory: Basics

- Math definition: A group is a set of elements G with a binary operation "." obeying:
  - **Closure:** c=a.b belongs to G for all a,b in G.
  - **Unit element:** e: e.a=a.e=a for all a in G.
  - **Inverse element:** a^{-1}: $a.a^{-1} = a^{-1}.a=e$ for all a in G.
  - **Associativity:** (a.b).c=a.(b.c) for all a,b,c in G.
- Less rigorous definitions often used in physics:
  - Define the transformations explicitly. For example, "U(3) is the group of transformations represented with 3x3 unitary matrices".
  - Define the invariant quantity explicitly. For example, "O(3) is the group of transformations preserving the length of a 3-vector". This will be our approach here.

ML Symmetry Formalism

- We need to define the object, the transformation and the group invariant.

Vector Space **x** → **Map** → Vector Space **x'**

Function (Oracle) $\phi(x)=\phi(x')$

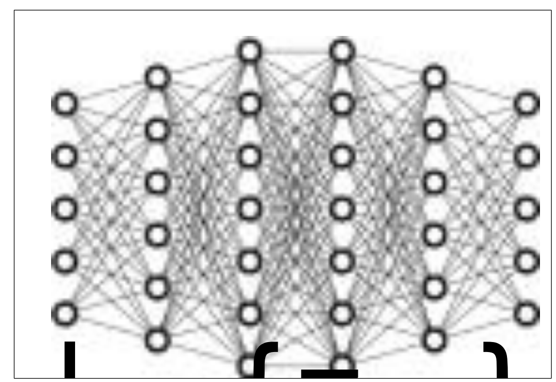
- Let's translate this setup into the machine learning terminology

m samples

n features | *k* labels

$$\begin{array}{cccc|cccc} x_1^{(1)}, & x_1^{(2)}, & \ldots, & x_1^{(n)} & y_1^{(1)}, & \ldots, & y_1^{(k)} \\ x_2^{(1)}, & x_2^{(2)}, & \ldots, & x_2^{(n)} & y_2^{(1)}, & \ldots, & y_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ x_m^{(1)}, & x_m^{(2)}, & \ldots, & x_m^{(n)} & y_m^{(1)}, & \ldots, & y_m^{(k)} \end{array}$$

Neural network



This part (the data) will be given to us

We will be building this

Symmetry parametrization

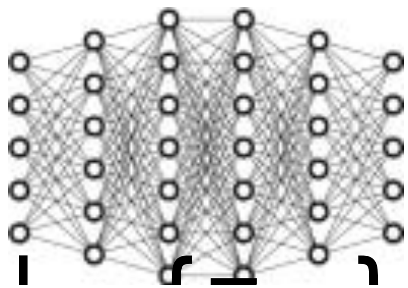
Linear

x → x'=(I+εG)x

I: identity matrix

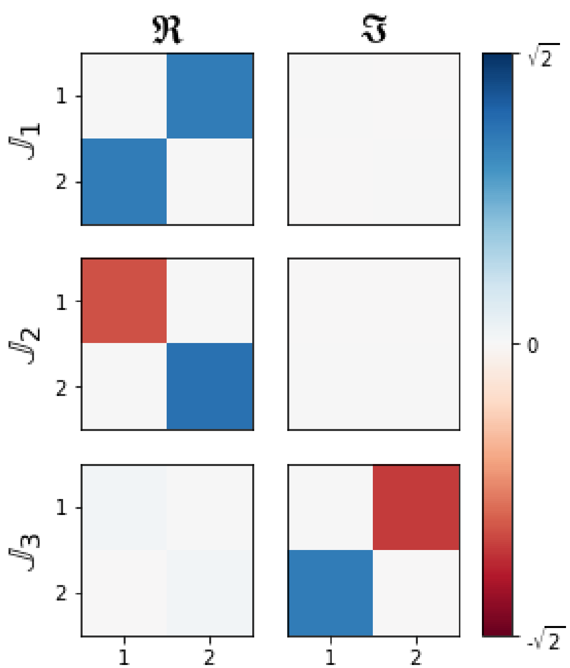
G: matrix to be learned

Non-linear

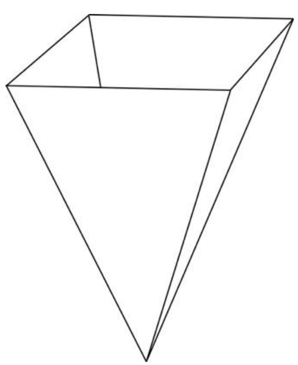
x →  → x'

Neural network whose parameters are to be learned

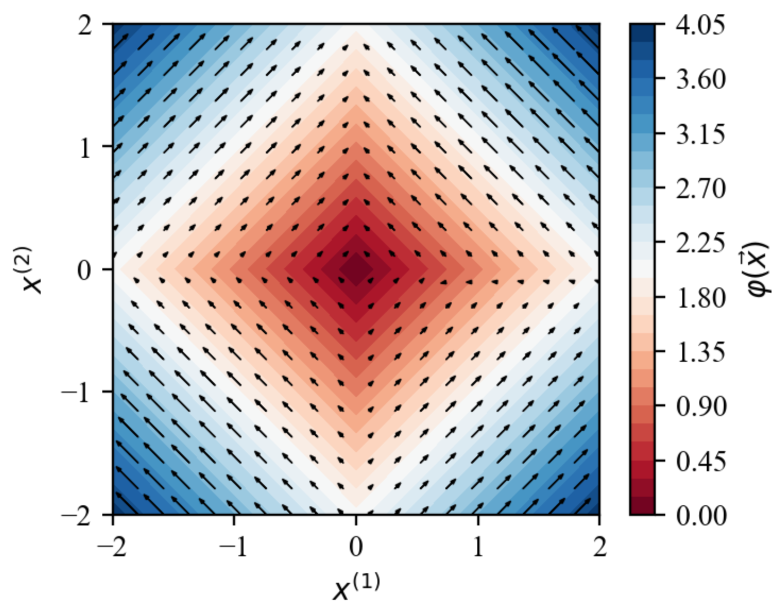
Visualizing the learned generators: SU(2) example (Pauli matrices)



oracle function



Visualizing the learned transformation



Oracle Zoo

O(n)

$$\varphi_O(\mathbf{x}) \;\equiv\; |\mathbf{x}|^2 = \sum_{j=1}^{n} [x^{(j)}]^2, \quad x^{(j)} \in \mathbb{R}$$

U(n)

$$\varphi_U(\mathbf{x}) \;\equiv\; \sum_{j=1}^{n} \left(x^{(j)}\right)^* x^{(j)}, \quad x^{(j)} \in \mathbb{C}.$$

SO(1,3)

$$\varphi_L(\mathbf{x}) \;\equiv\; \left(x^{(1)}\right)^2 - \left(x^{(2)}\right)^2 - \left(x^{(3)}\right)^2 - \left(x^{(4)}\right)^2, \quad x^{(j)} \in \mathbb{R}$$

Squeeze map

$$' \,(x) = x^{(1)} \, x^{(2)}$$

G_2

$$'^{(1)}_{G_2}(x) = \overset{X7}{\underset{i=1}{}} (x^{(i)})^2$$

$$'^{(2)}_{G_2}(x_1, x_2, x_3) = \overset{X7}{\underset{i,j,k=1}{}} D_{ijk} \, x_1^{(i)} \, x_2^{(j)} \, x_3^{(k)}$$

F_4

$$\varphi^{(1)}_{F_4}(\mathbf{x}) = \mathrm{Tr}\,\mathfrak{h}_3 = \sum_{a=1}^{3} r_a$$

$$\varphi^{(2)}_{F_4}(\mathbf{x}) = \mathrm{Tr}\,\mathfrak{h}_3^2 = \sum_{a=1}^{3} \left(r_a^2 + 2\,|\mathbf{o}_a|^2 \right)$$

$$\varphi^{(3)}_{F_4}(\mathbf{x}) = \det \mathfrak{h}_3 = r_1 r_2 r_3 - \sum_{a=1}^{3} r_a \, |\mathbf{o}_{4-a}|^2 + 2\,\mathrm{Re}\,(\mathbf{o}_3 \mathbf{o}_2^* \mathbf{o}_1)$$

Loss Functions

- **Question:** How does the machine learn?
- **Answer:** by minimizing the loss function! The loss computes the difference b/n the current and the expected outputs of an algorithm
- We can easily incorporate constraints into the loss function

Constraint $A(p)=B$ → Loss function $[A(p)-B]^2$

- Our loss function will contain several contributions, one for each desired symmetry property
  - **Invariance** (the transformation preserves the oracle)
  - **Normalization** (the transformation is non-trivial)
  - **Orthogonality** (the found symmetries are different from each other)
  - **Closure** (the found symmetries form a group)
  - **Sparsity** (the representations are simple and easy to interpret)

Contributions to the Loss Function

Invariance

$$L_{\text{inv}}(\mathbb{G},\{\mathbf{x}\}) = \frac{1}{m\varepsilon^2}\sum_{i=1}^{m}\left[\vec{\varphi}(\mathbf{x}_i + \varepsilon\mathbb{G}\cdot\mathbf{x}_i) - \vec{\varphi}(\mathbf{x}_i)\right]^2$$

Normalization

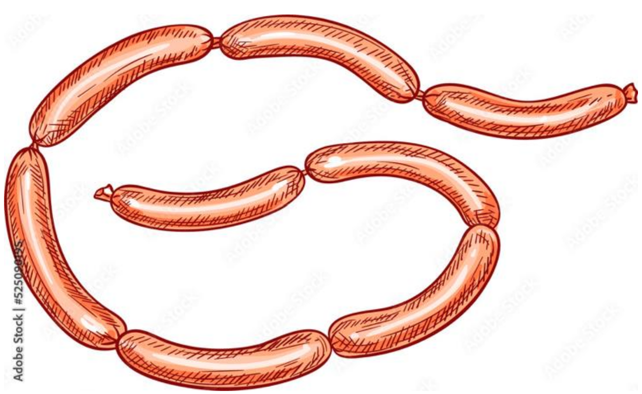
$$L_{\text{norm}}(\mathbb{G}) = \left[\text{Tr}\left(\mathbb{G}\cdot\mathbb{G}^T\right) - 1\right]^2$$

Sparsity

$$L_{\text{sp}}(\mathbb{G}) = \sum_{j,k=1}^{n}\sum_{j',k'=1}^{n}\left|\mathbb{G}^{(jk)}\mathbb{G}^{(j'k')}\right|\left(1 - \delta_{jj'}\delta_{kk'}\right)$$

Orthogonality

$$L_{ortho}(\mathbb{G},\{\mathbf{x}_i\}) = \sum_{\alpha<\beta}^{N_g}\left[\text{Tr}(\mathbb{G}_\alpha^T\mathbb{G}_\beta)\right]^2$$

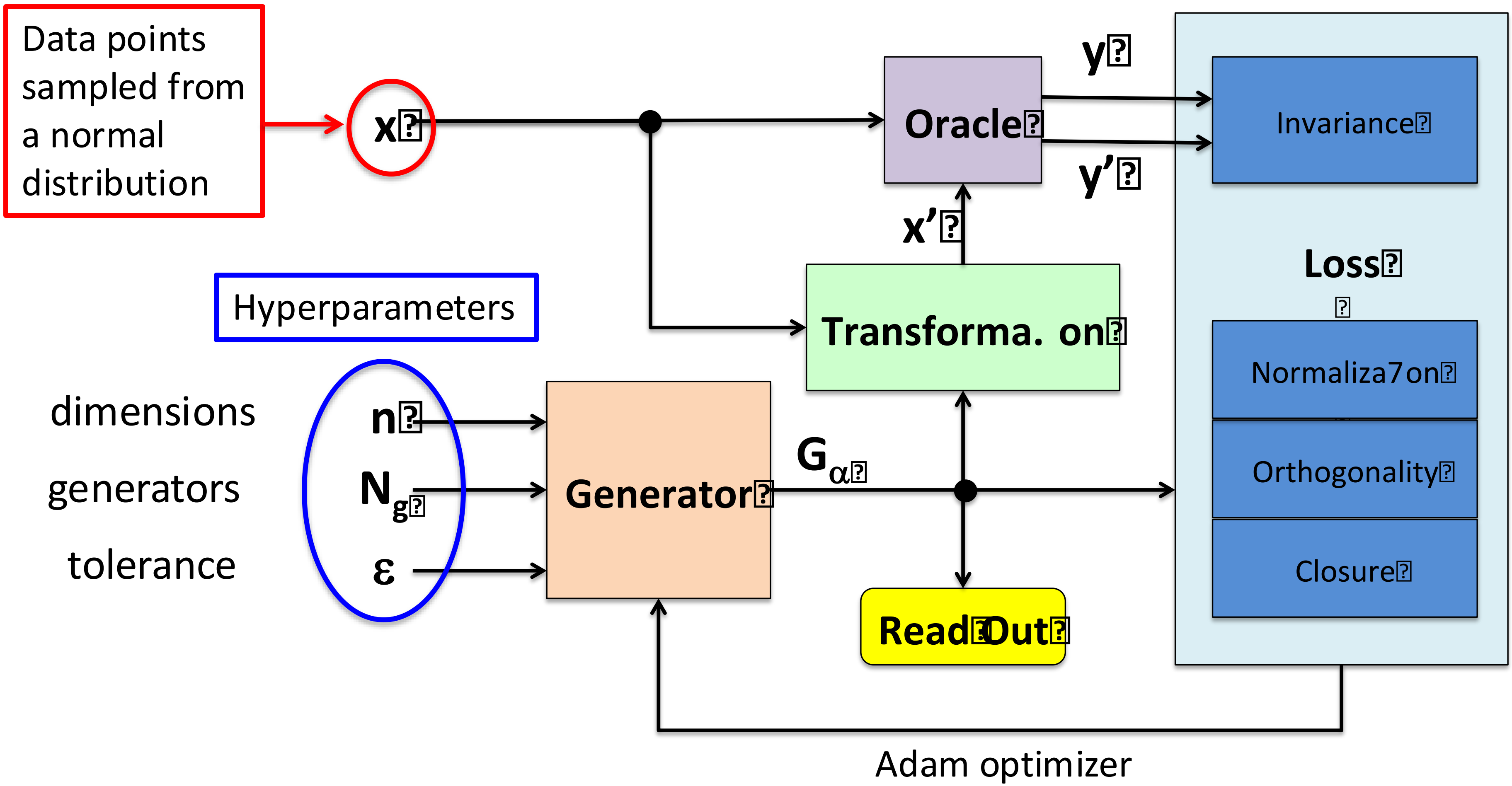


Closure

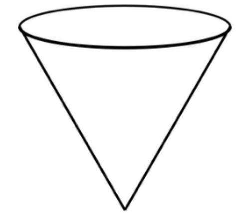
$$L_{closure}(a_{[\alpha\beta]\gamma}) = \sum_{\alpha<\beta}\text{Tr}\left(\mathbb{C}_{[\alpha\beta]}^T\mathbb{C}_{[\alpha\beta]}\right)$$

$$\mathbb{C}_{[\alpha\beta]}(a_{[\alpha\beta]\gamma}) \equiv \left[\mathbb{G}_\alpha,\mathbb{G}_\beta\right] - \sum_{\gamma=1}^{N_g}a_{[\alpha\beta]\gamma}\mathbb{G}_\gamma$$

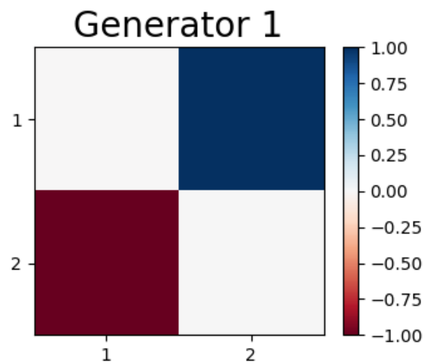
The Basic Algorithm



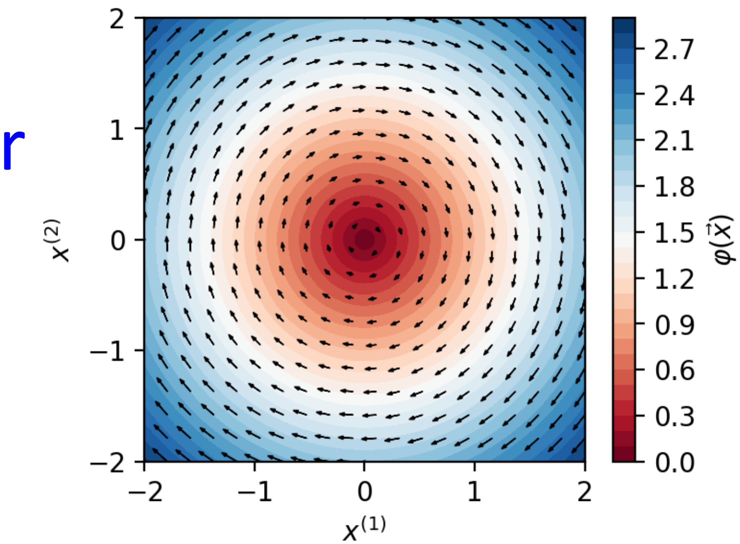
First Toy Example: 2D Rotations



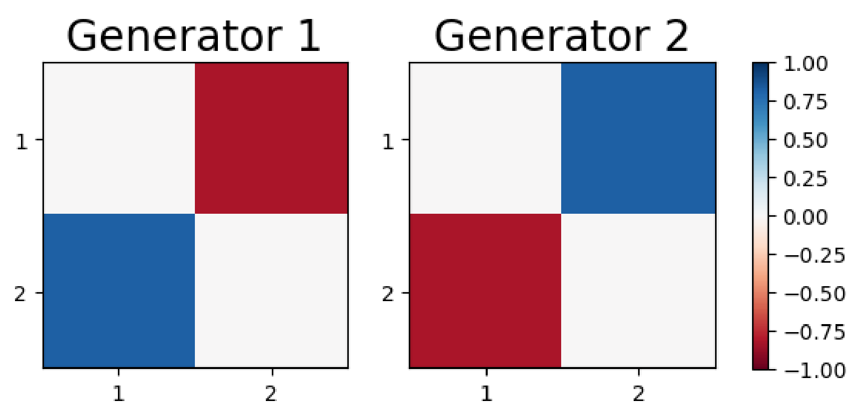
- Oracle: L2 norm of a vector
- n=2 dimensions; N_g=1 generator

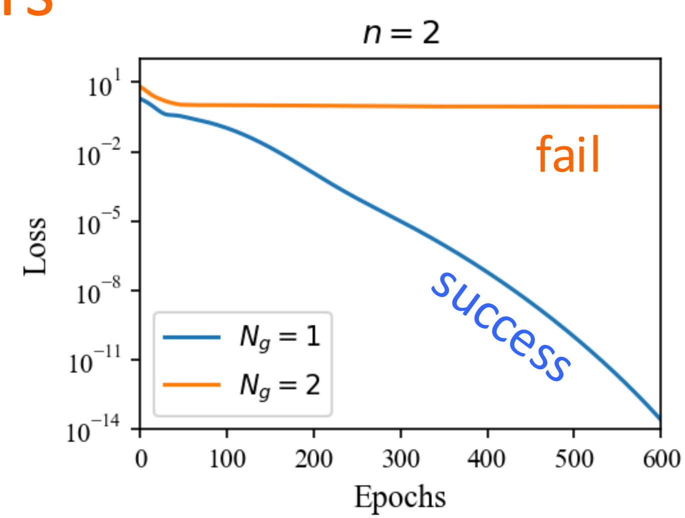


Orthogonal matrices are anti-symmetric!

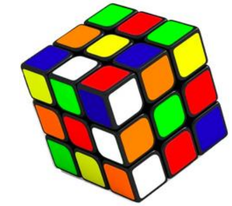


- n=2 dimensions; N_g=2 generators

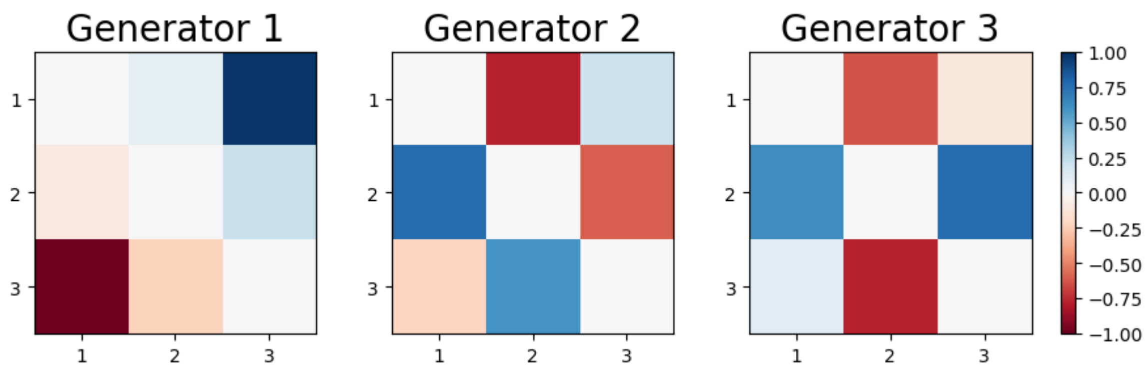


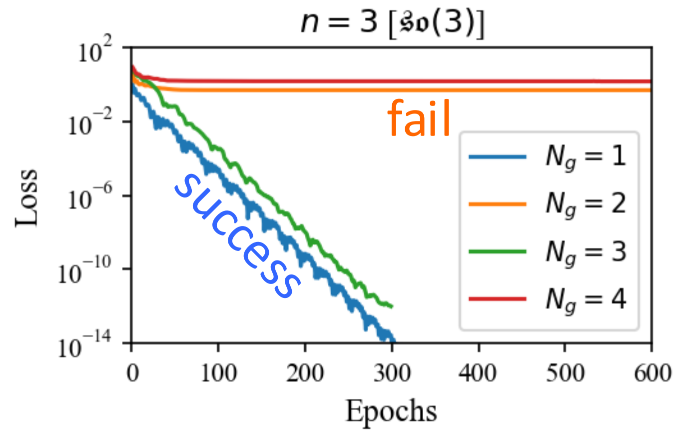


Rotations in 3D

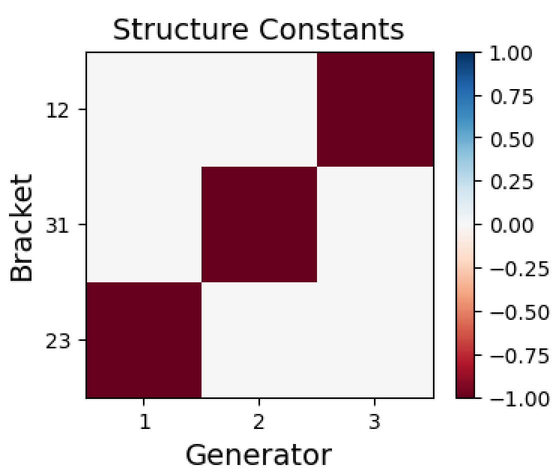


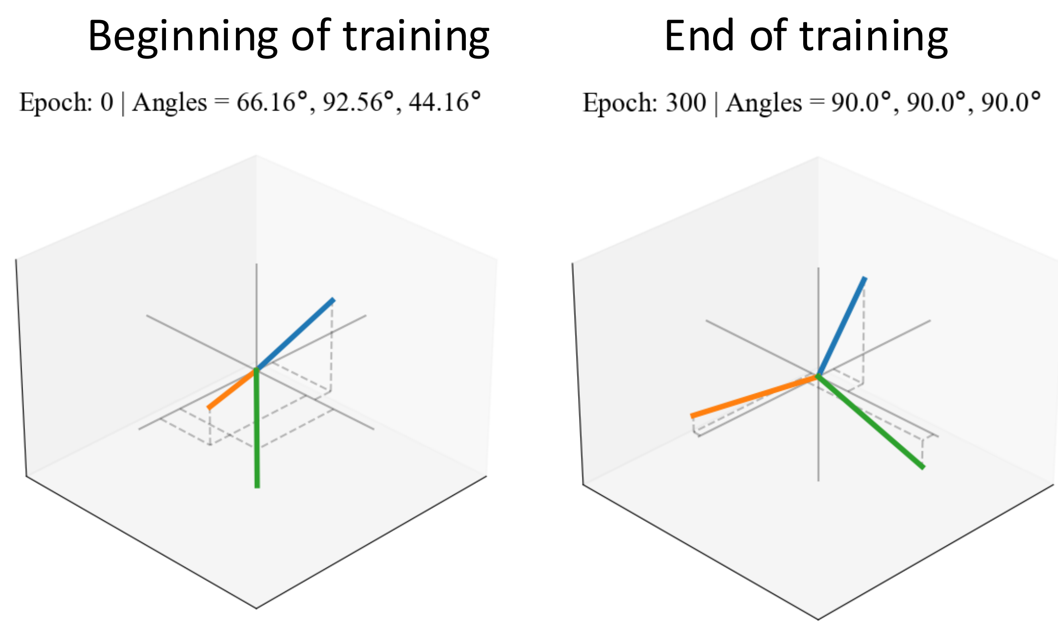
- The training is successful for N_g=1 or N_g=3 generators





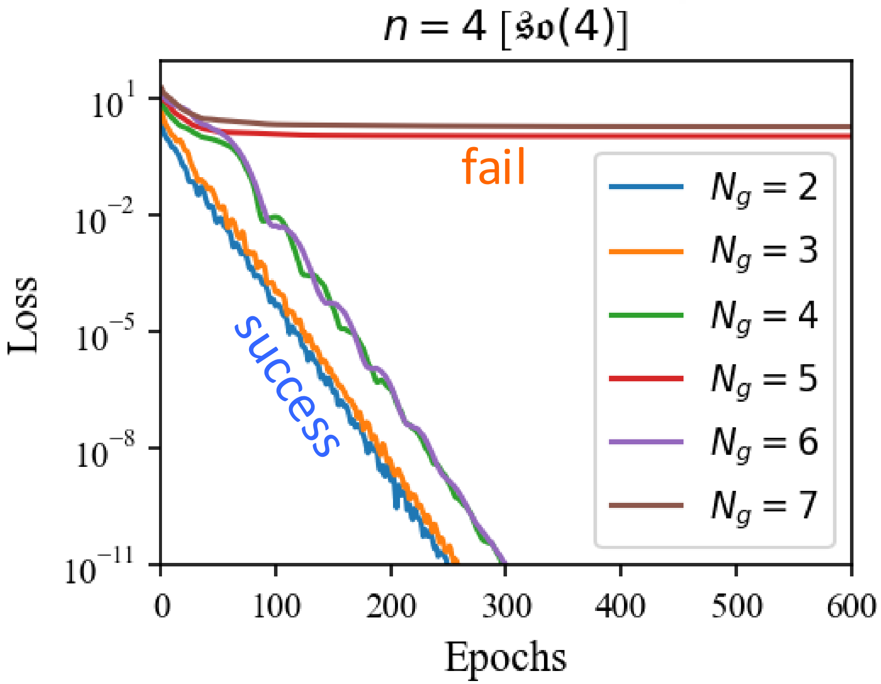
$$[\mathbf{J}_\alpha, \mathbf{J}_\beta] = \sum_{\gamma=1}^{N_g} a_{[\alpha\beta]\gamma} \mathbf{J}_\gamma$$

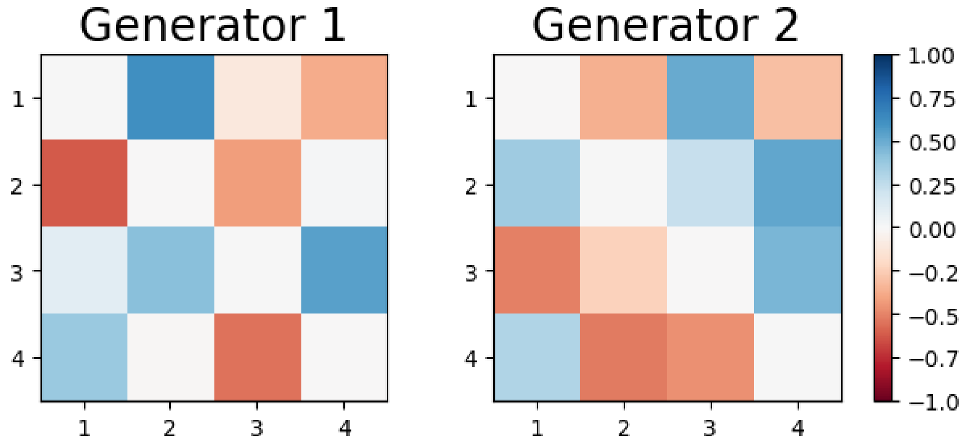


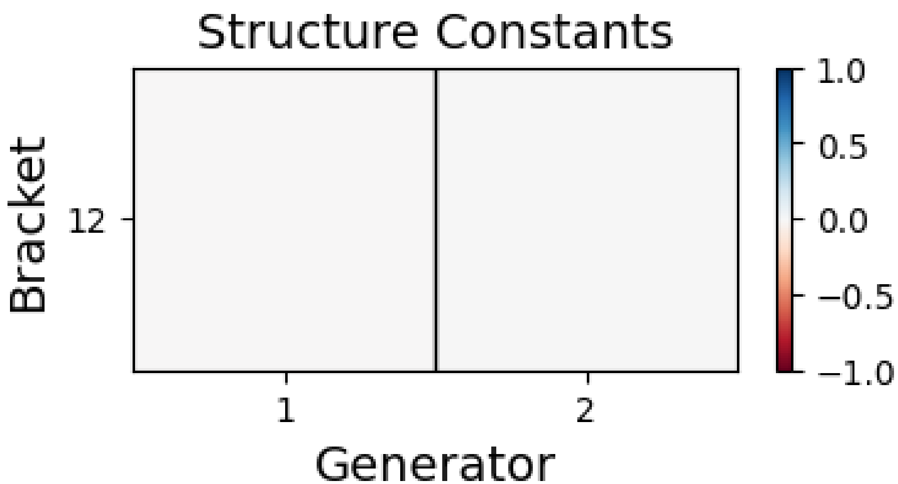


Rotations in 4D

- The training is successful for N_g=1,2,3,4,6 generators
- The full symmetry group SO(4) is rank 2
  - There exists an Abelian (Cartan) subalgebra with N_g=2 generators

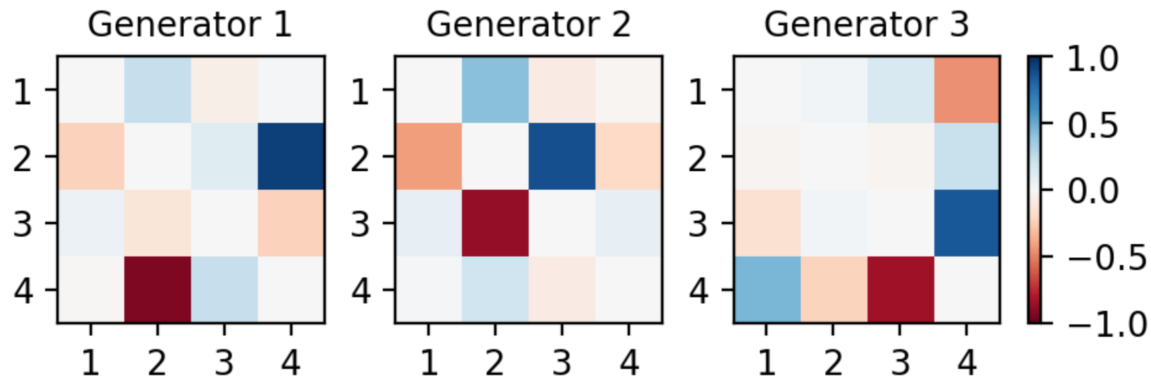


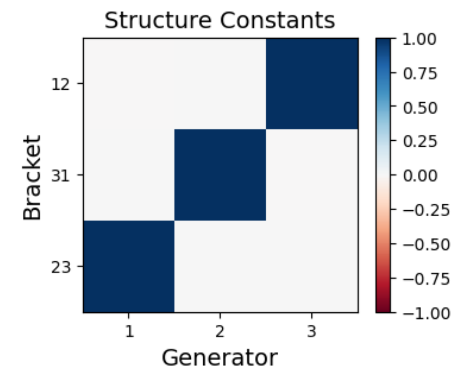




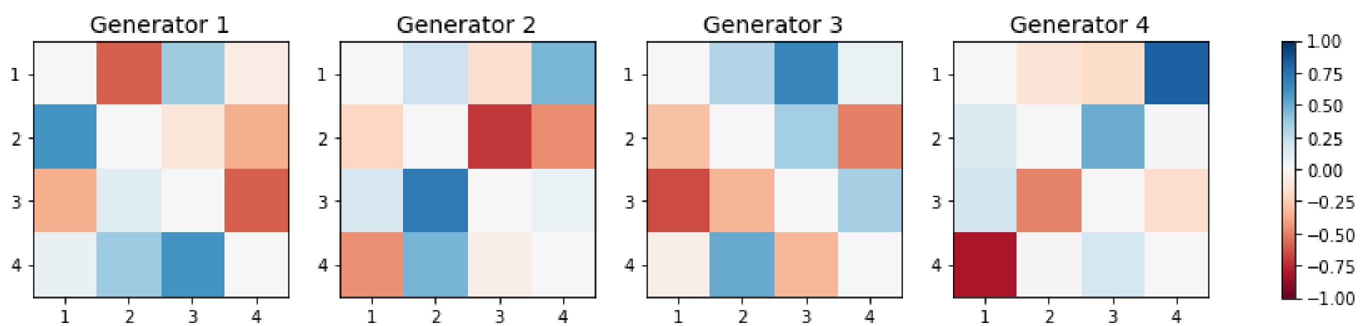
Rotations in 4D: other subalgebras

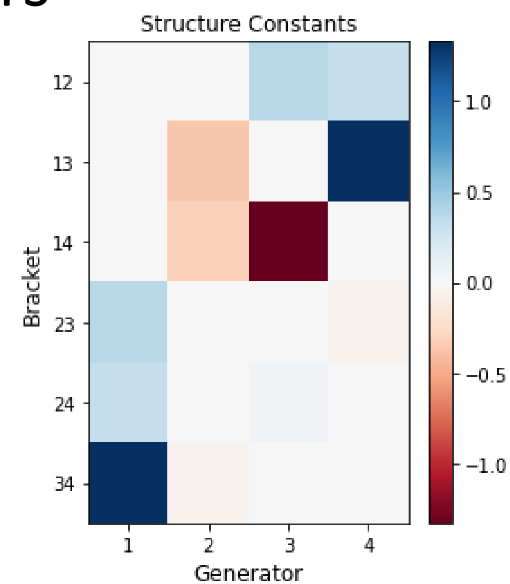
- The training is successful for N_g=3 generators
  - SO(3) is a subgroup of SO(4)





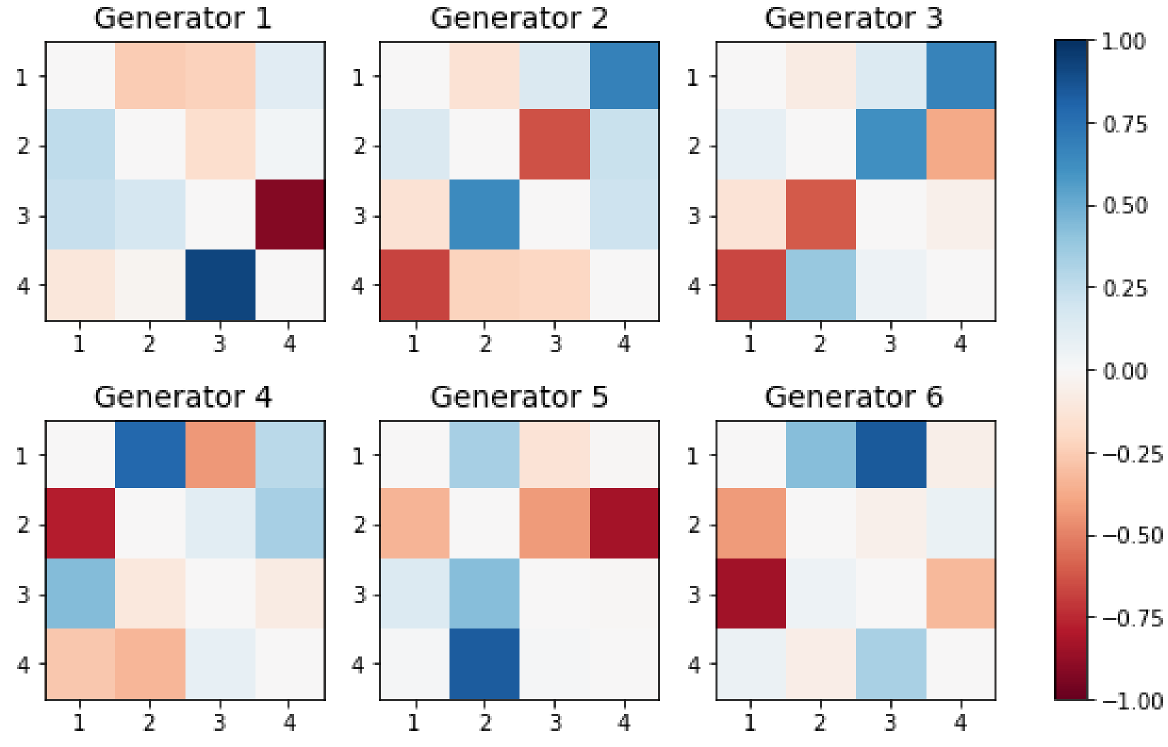
- The training is successful for N_g=4 generators
  - SO(3)xSO(2) is a subgroup of SO(4)

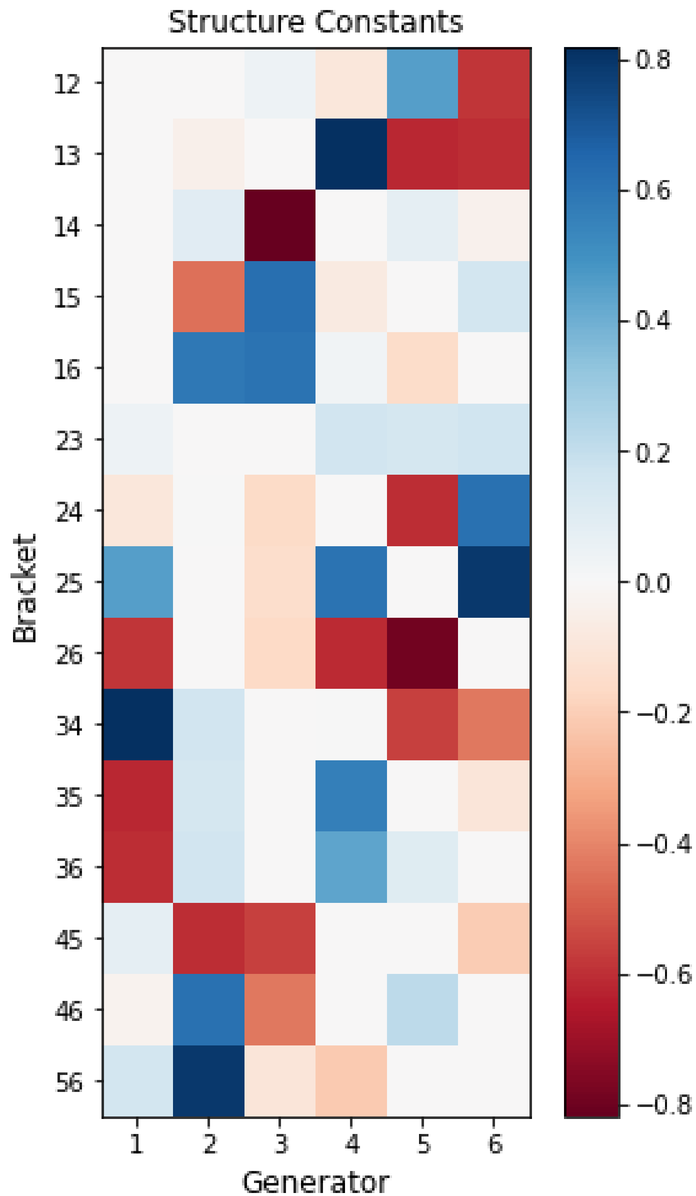




Rotations in 4D: full so(4) algebra

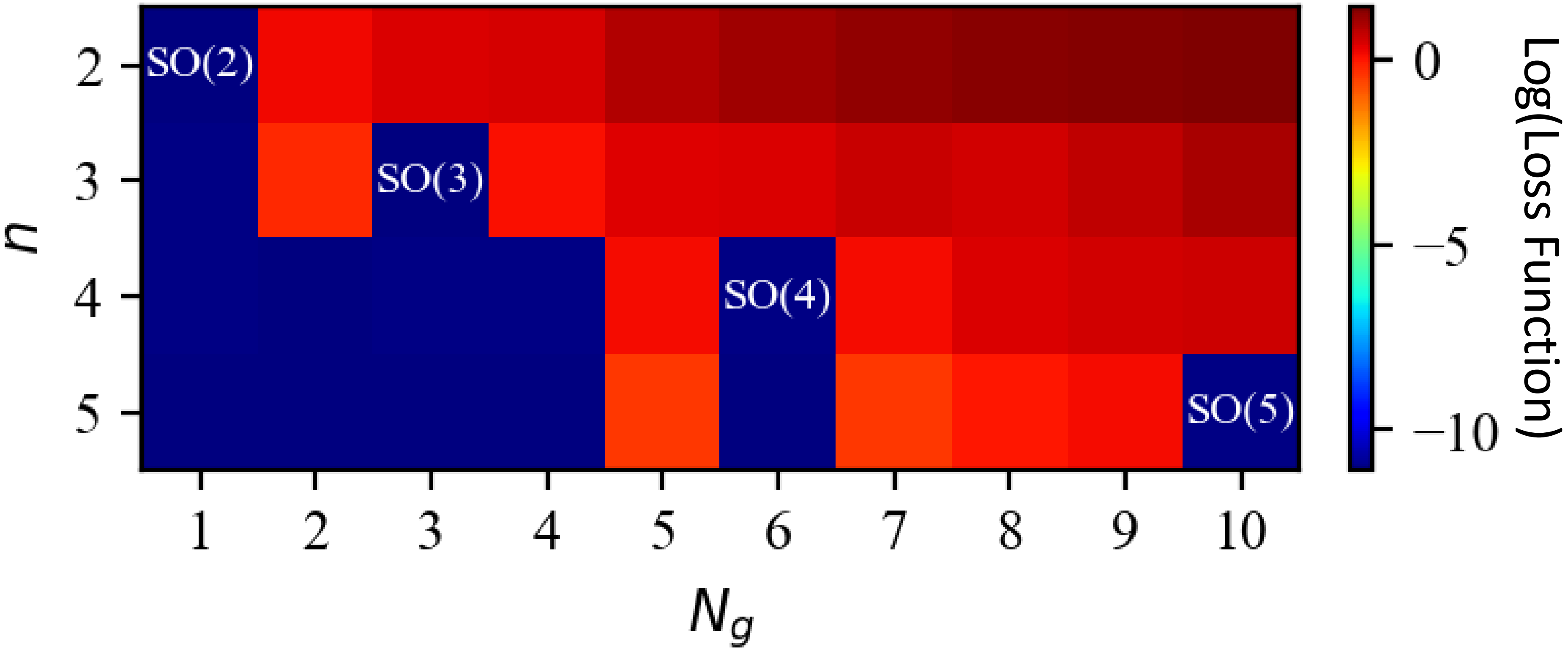
- N_g=6 is the maximum number of generators which
  - are symmetries
  - are orthonormal
  - form a closed algebra



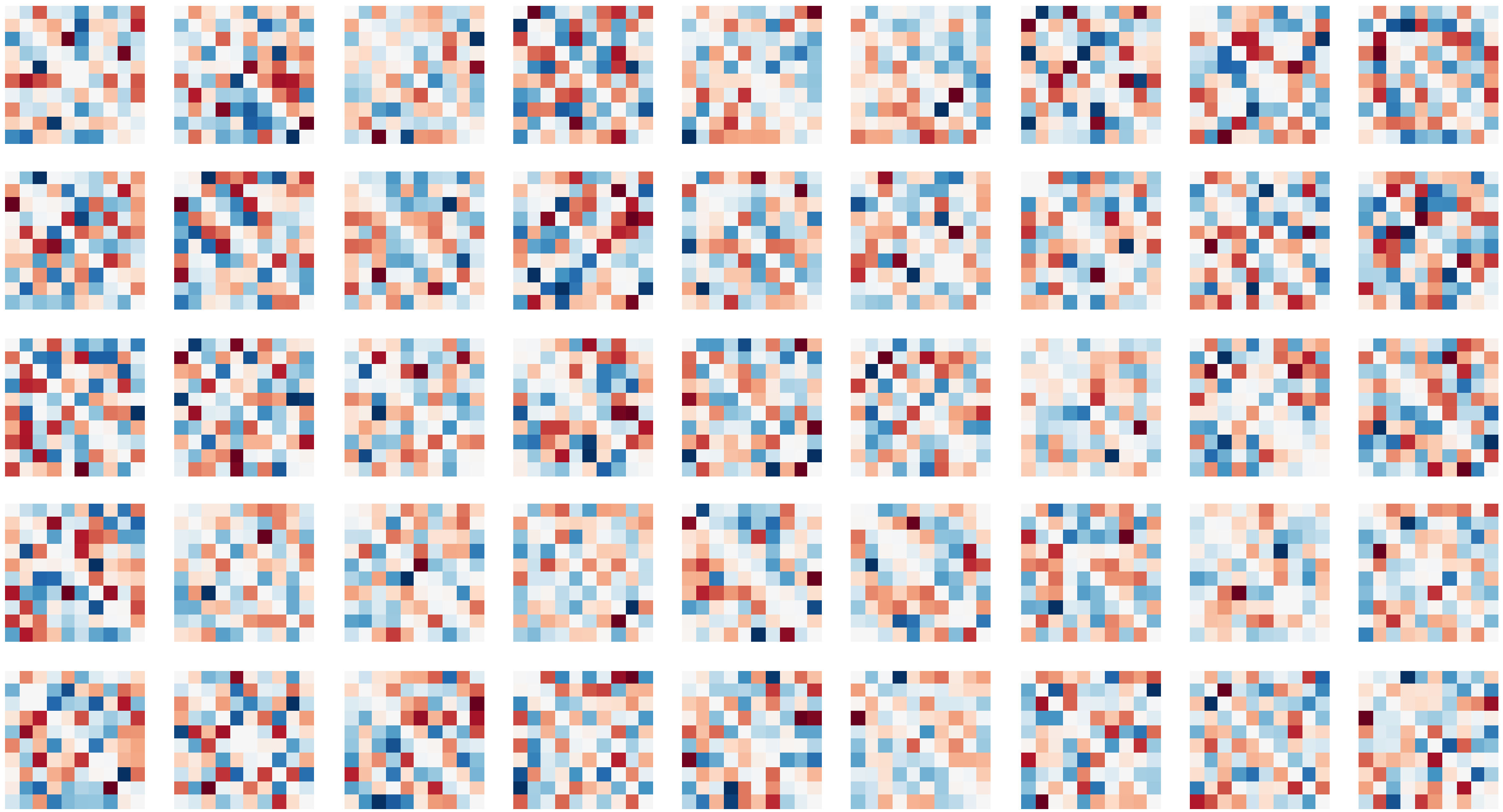


SO(n) Summary

- Heat map of the Log-Loss for
  - different number of dimensions n (y-axis)
  - different number of generators N_g (x-axis)
- Low values of the loss (blue squares) imply valid algebras

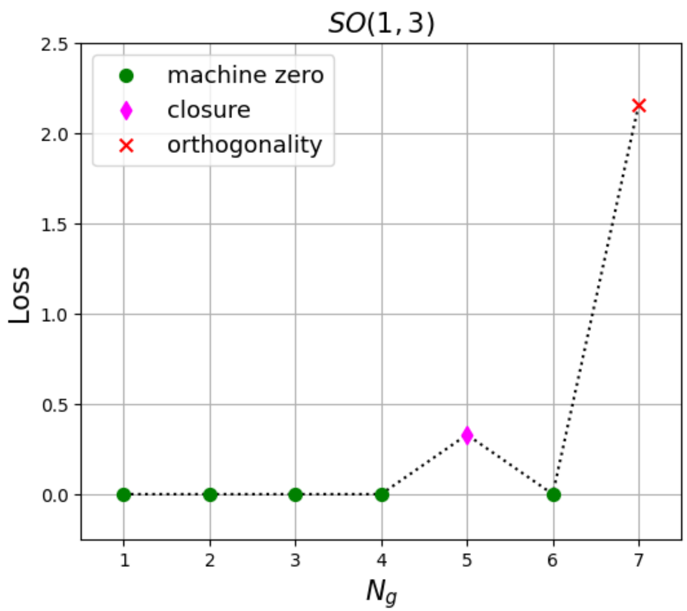


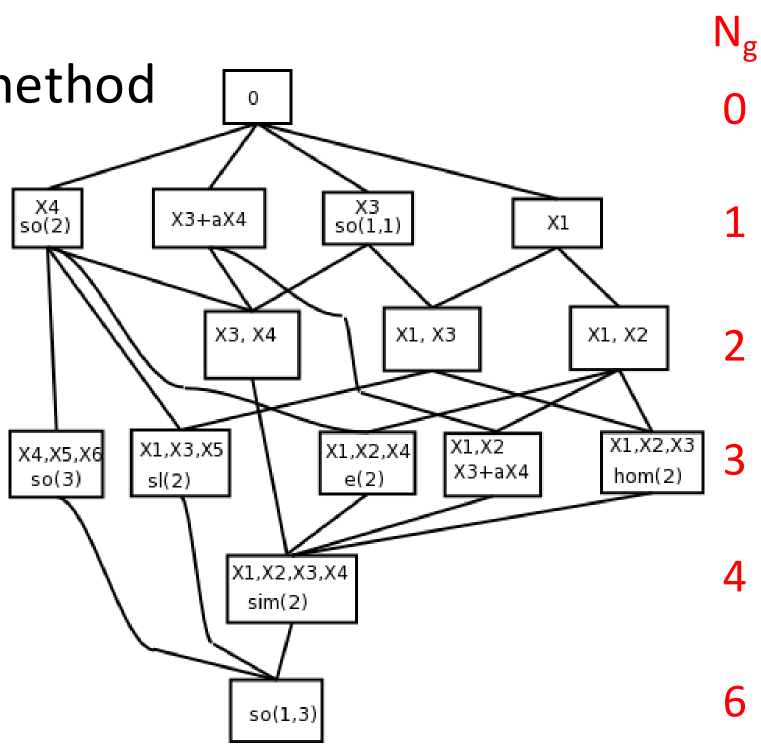
Tour de Force: SO(10) Generators



Lorentz Group

- **Oracle:** the Minkowski metric $t^2-x^2-y^2-z^2$
- The Lorentz group is the cornerstone of relativity
- Being **SO(1,3)**, it resembles SO(4), but has richer subalgebra structure
  - **N_g=2:** both an Abelian and a non-Abelian subalgebra
  - **N_g=3:** five different types of non-Abelian subalgebras
  - **N_g=4:** one subalgebra
  - **N_g=6:** the full Lorentz algebra
- All those results were confirmed with our method

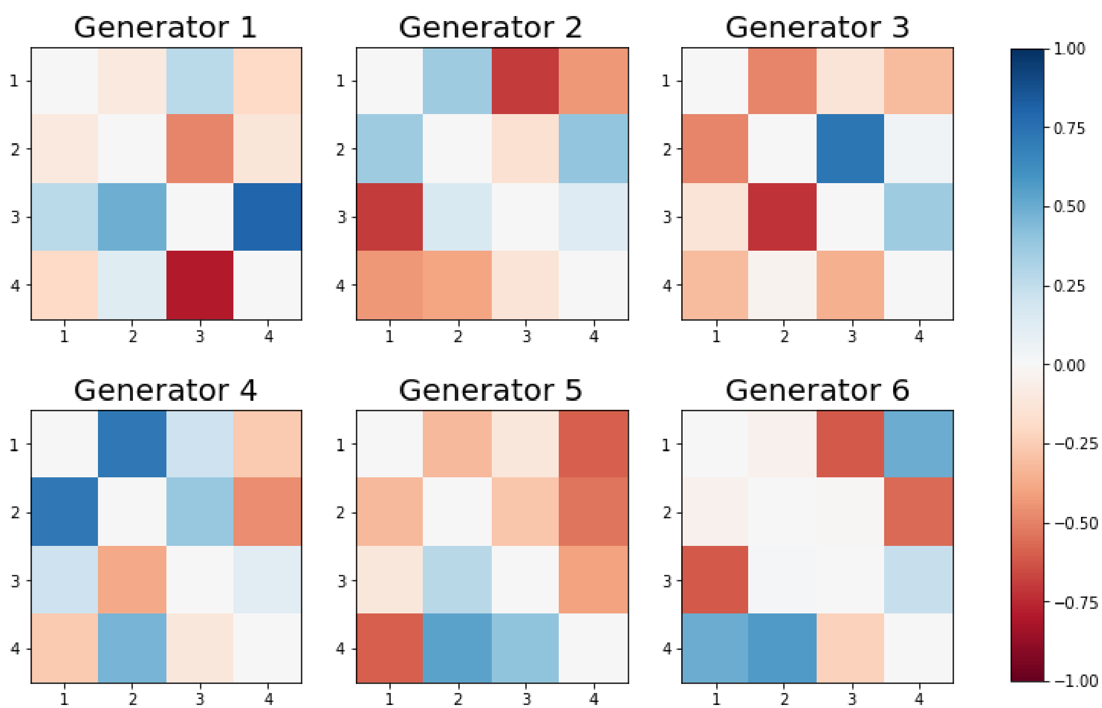


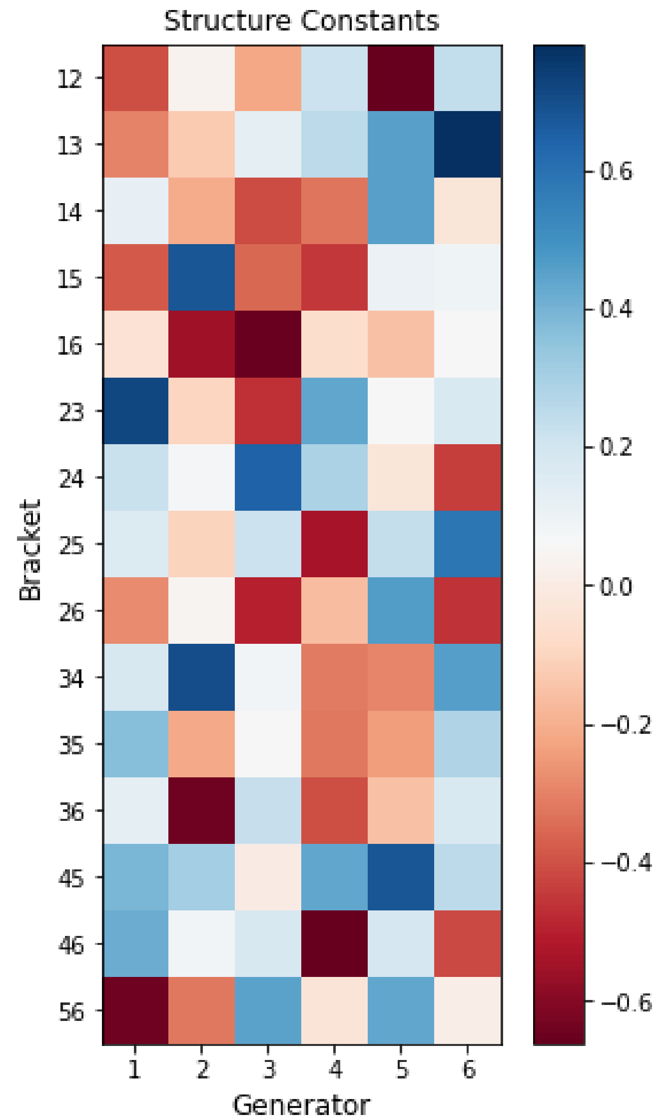


Source: wikipedia

Example: the full Lorentz algebra

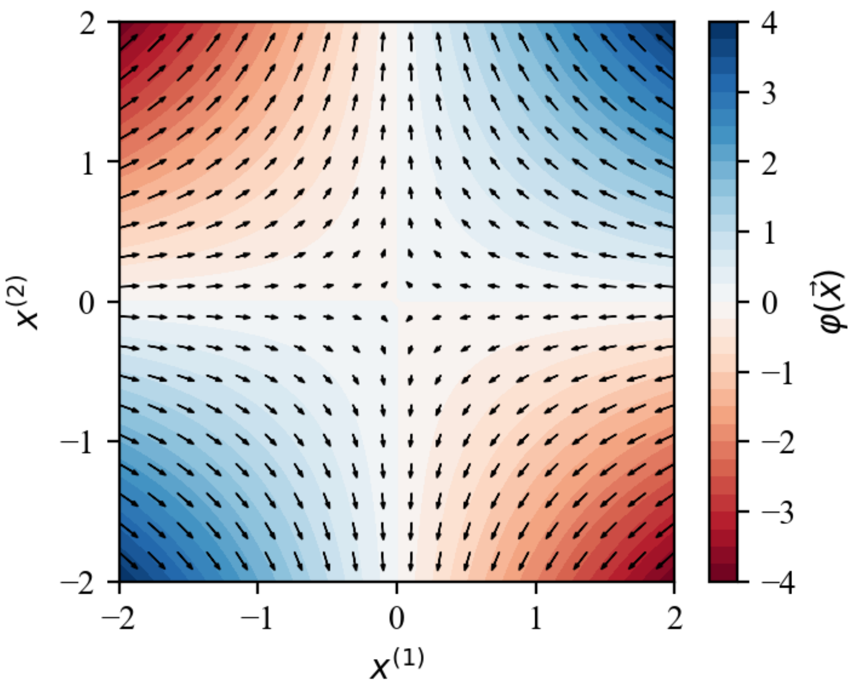
- The N_g=6 generators describe:
  - Boosts: symmetric matrix, non-zero entries in 0i position, i=1,2,3
  - Rotations: anti-symmetric matrix, non-zero entries in ij position
- The learned generators are generic mixtures of those canonical generators





Squeeze mapping

- Squeeze mapping is an example of an equi-areal symmetry transformation
- The oracle in this case is $\phi(x_1,x_2)=x_1x_2$.
  - The area of a rectangle with corners at the origin and at (x_1,x_2)
  - The equipotential contours are hyperbolas $x_2=a/x_1$
- A deep neural network can learn a finite transformation (represented with the arrows in the vector flow)



- A shallow neural network can learn the infinitesimal generator J

$$J = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$$

(Midpoint) Summary

- **Taking stock of what we have been able to accomplish**
  - **Given:** some dataset with labels given by an oracle
  - **Derive:** the closed (sub)algebra of N_g symmetry generators
- **Comments and observations**
  - The learned generator depends on the initial configuration (seed)
  - The learned generators are some general linear combinations of the canonical sparse generators found in the textbooks
  - The training can get rather slow for large N_g
- **Open questions, refinements and potential improvements***
  - Can we **speed up** the training?
  - How can we learn the canonical **sparse** generators?
  - Can we preserve **more than one oracle** function simultaneously?
    - Examples: the exceptional groups G_2, F_4; the MNIST digits.
  - Can the oracle be a more complicated (**discontinuous**) function?
  - How can we **identify** what symmetry group has been learned?
  - How to decide if the symmetry is **trivial** or not? (irrelevant features?)

*all those points were addressed in the follow-up papers

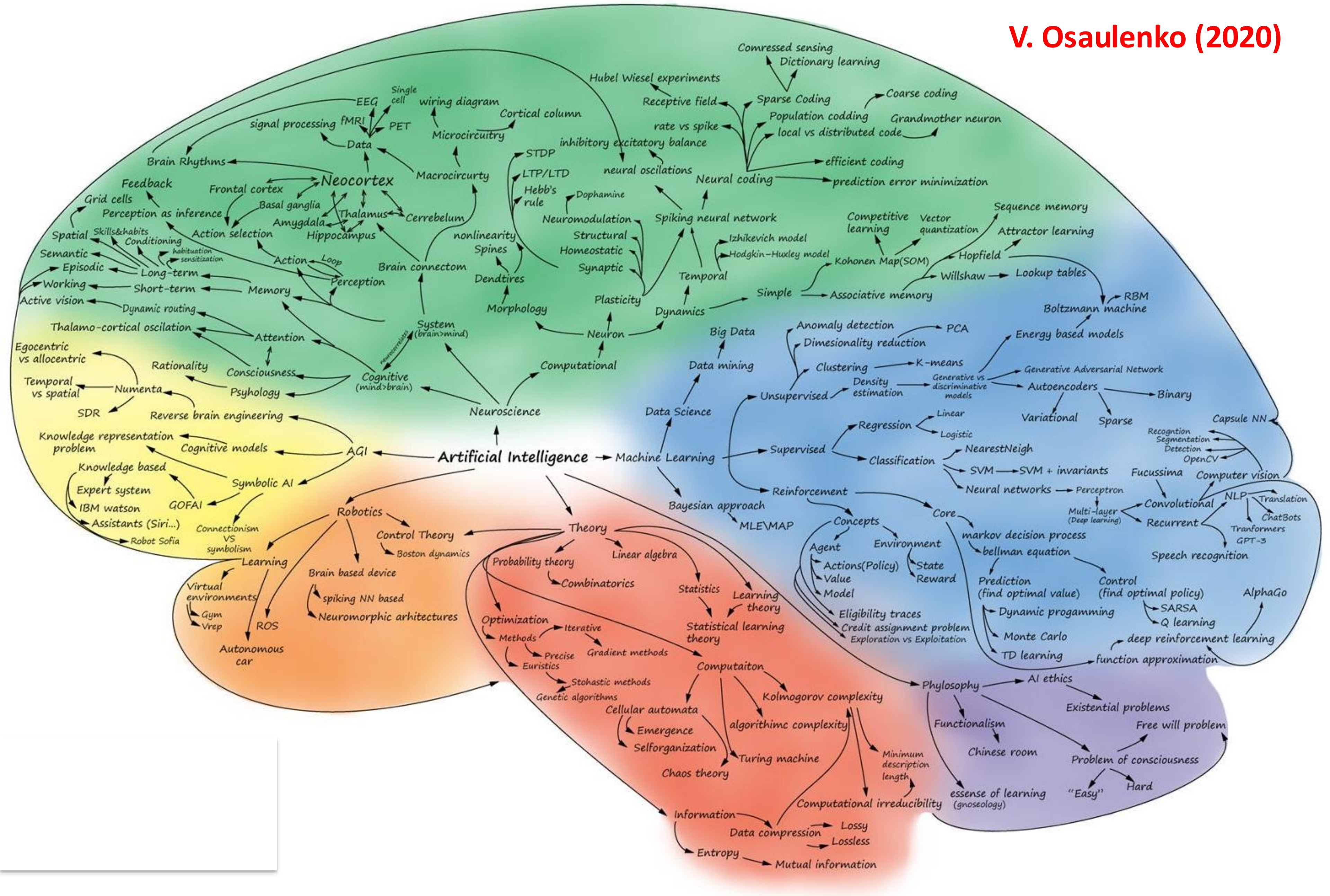
Further Reading

- S. Krippendorf and M. Syvaeri, “Detecting Symmetries with Neural Networks,” 2003.13679.
- Z. Liu and M. Tegmark, “Machine Learning Conservation Laws from Trajectories,” 2011.04698.
- G. Barenboim, J. Hirn and V. Sanz, “Symmetry Meets AI,” 2103.06115.
- B. M. Dillon, G. Kasieczka, H. Olischlager, T. Plehn, P. Sorrenson and L. Vogel, “Symmetries, Safety, and Self-supervision,” 2108.04253.
- Z. Liu and M. Tegmark, “Machine Learning Hidden Symmetries,” 2109.09721.
- K. Desai, B. Nachman and J. Thaler, “Symmetry Discovery with Deep Learning,” 2112.05722.
- S. Craven, D. Croon, D. Cutting and R. Houtz, “Machine Learning a Manifold,” 2112.07673.
- A. Moskalev, A. Sepliarskaia, I. Sosnovik, A. Smeulders, “Liegg: Studying Learned Lie Group Generators,” 2210.04345.
- A. Gabel, V. G. Klein, R. Valperga, J. S. W. Lamb, K. N. Webster, R. Quax and E. Gavves. “Learning Lie Group Symmetry Transformations with Neural Networks.” 2307.01583.

Prompt: Students cheering the end of a boring colloquium on Machine Learning (Picasso style)

The AI Map

V. Osaulenko (2020)

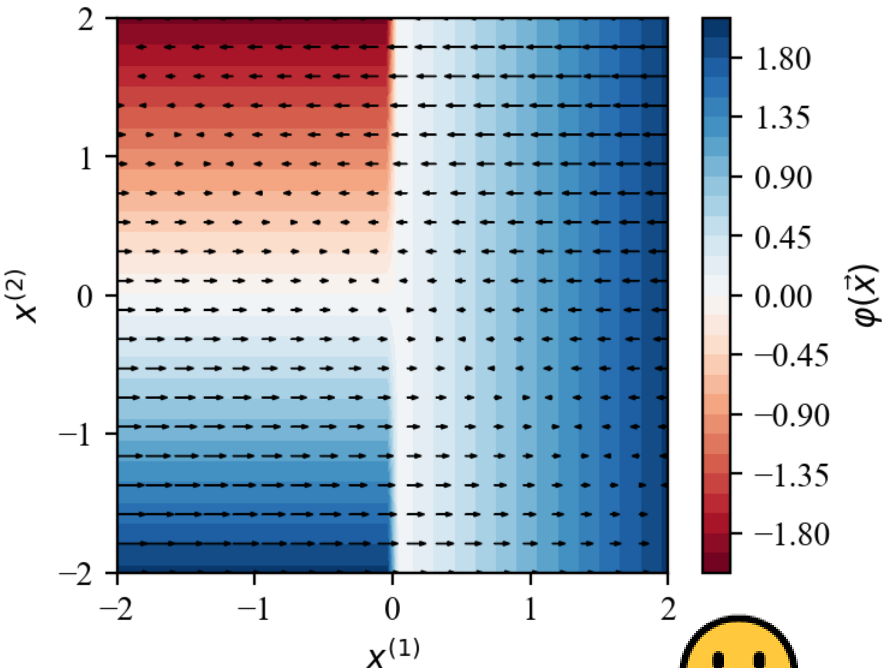


Discontinuous oracles – example I

- Discontinuous oracle functions lead to nonlinear transformations
  - require parametrization of the symmetry with a deep neural network

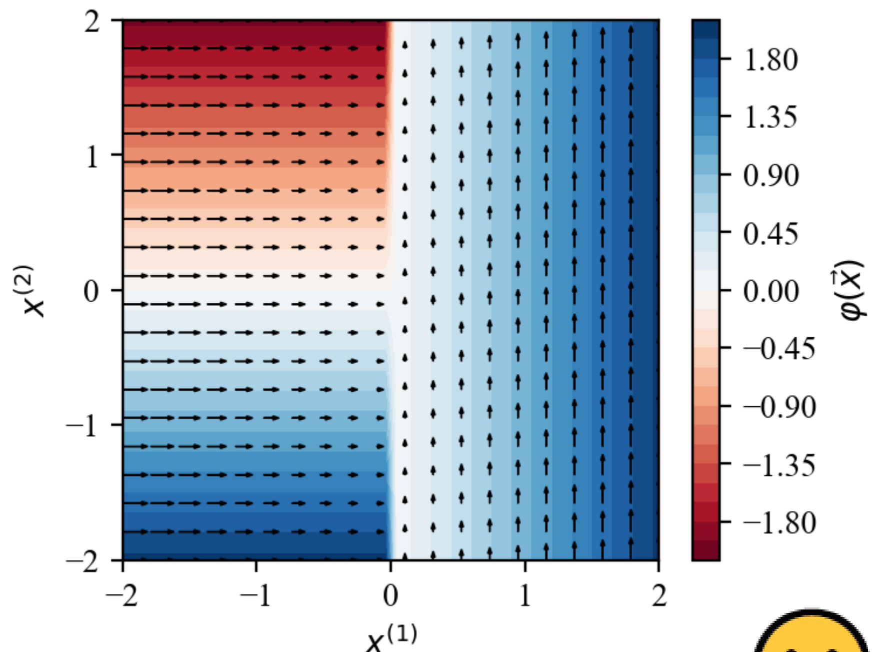
$$' (x) = \begin{cases} -x^{(2)}, & \text{for } x^{(1)} < 0, \\ +x^{(1)}, & \text{for } x^{(1)} \geq 0. \end{cases}$$

Shallow network
(no hidden layers)



Arrows cross contours

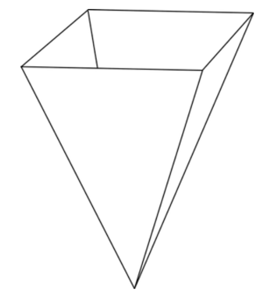
Deep network
(3 hidden layers)



Arrows aligned with contours

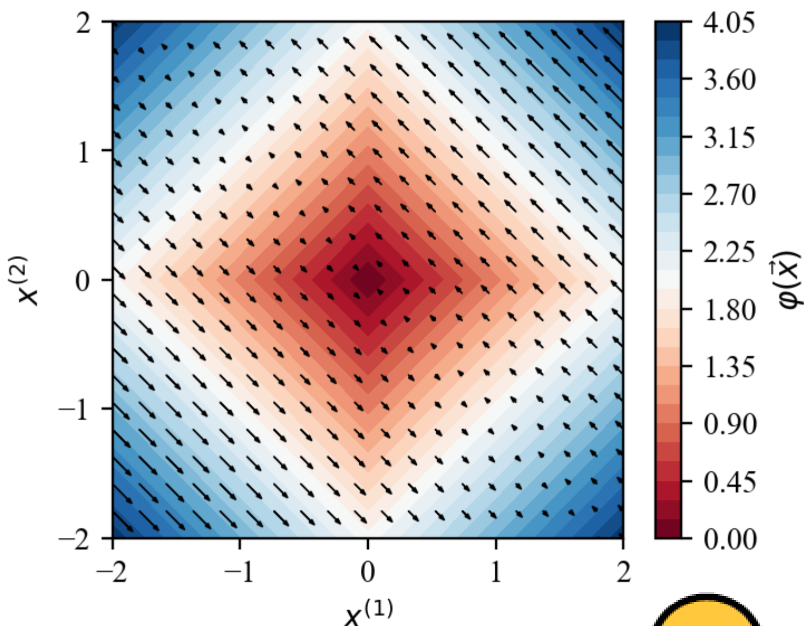
Discontinuous oracles – example II

- Another example: Manhattan (L1) distance oracle
  - Continuous, but not continuously differentiable



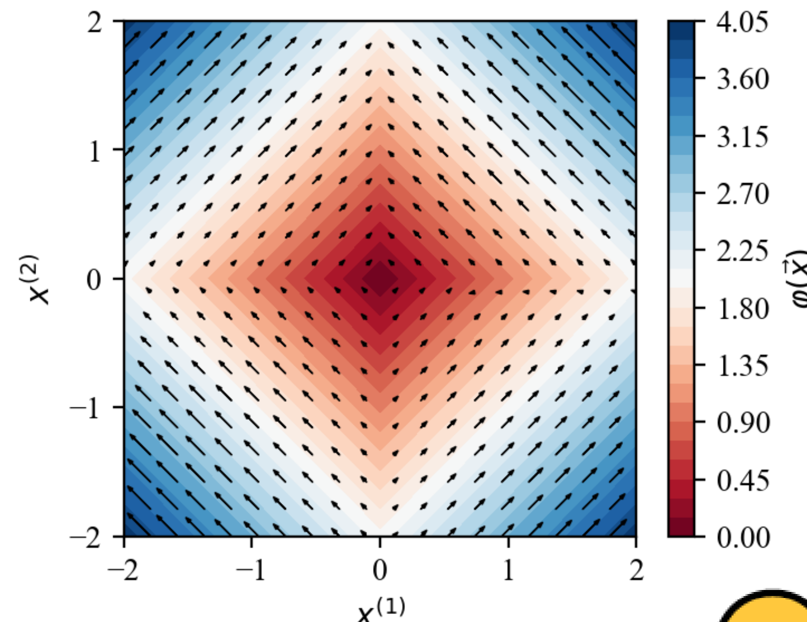
$$' (x) = |x^{(1)}| + |x^{(2)}|$$

Shallow network
(no hidden layers)



Arrows cross contours

Deep network
(3 hidden layers)



Arrows aligned with contours

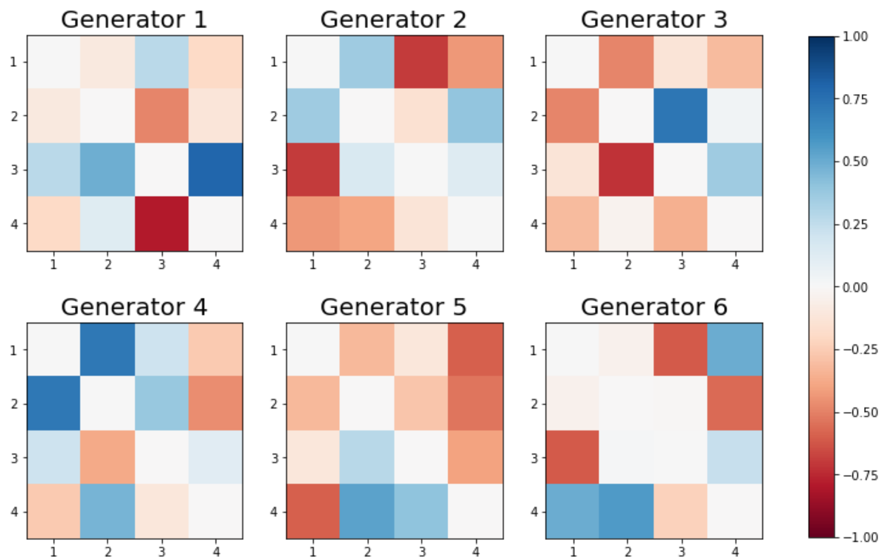
Learning Sparse Representations

- To find the canonical sparse form of the symmetry generators, include the following additional term to the loss function

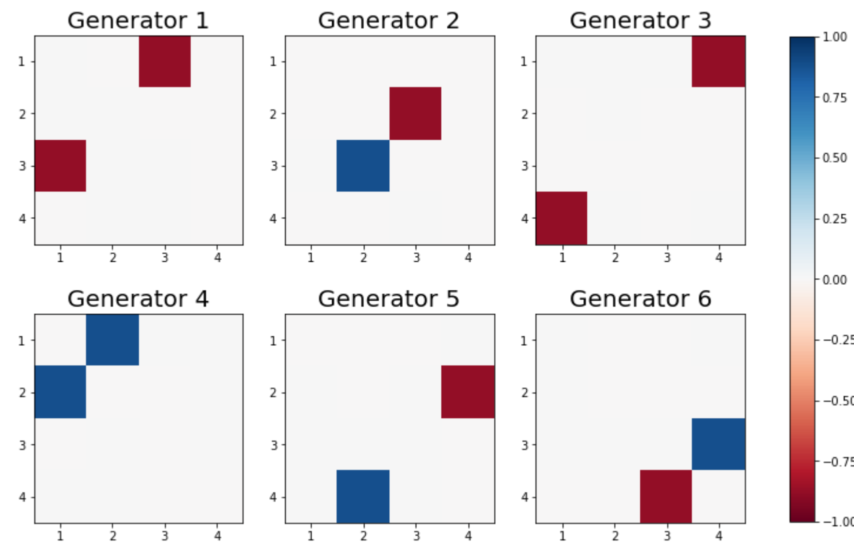
$$L_{\text{sp}}(\mathbb{G}) = \sum_{j,k=1}^{n} \sum_{j',k'=1}^{n} \left|\mathbb{G}^{(jk)}\mathbb{G}^{(j'k')}\right| \left(1 - \delta_{jj'}\delta_{kk'}\right)$$

- It encourages learning a sparser form of the generators
  - Example: the Lorentz group algebra found earlier

Before adding the sparsity loss



After adding the sparsity loss



Learning Sparse Representations

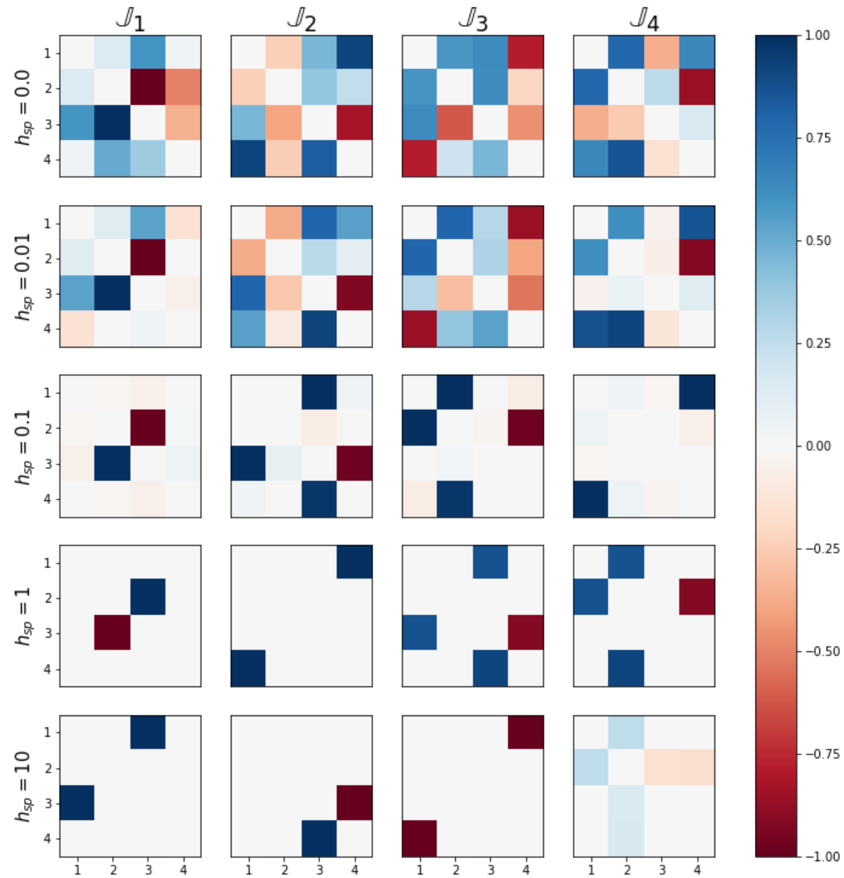
- The relative contribution of the **sparsity loss term** needs to be just right, otherwise it is too much of a good thing.
  - Example: the N_g=4 subgroup of the Lorentz group

No sparsity loss term

Negligible sparsity loss term

Just the right amount

Sparsity loss dominates

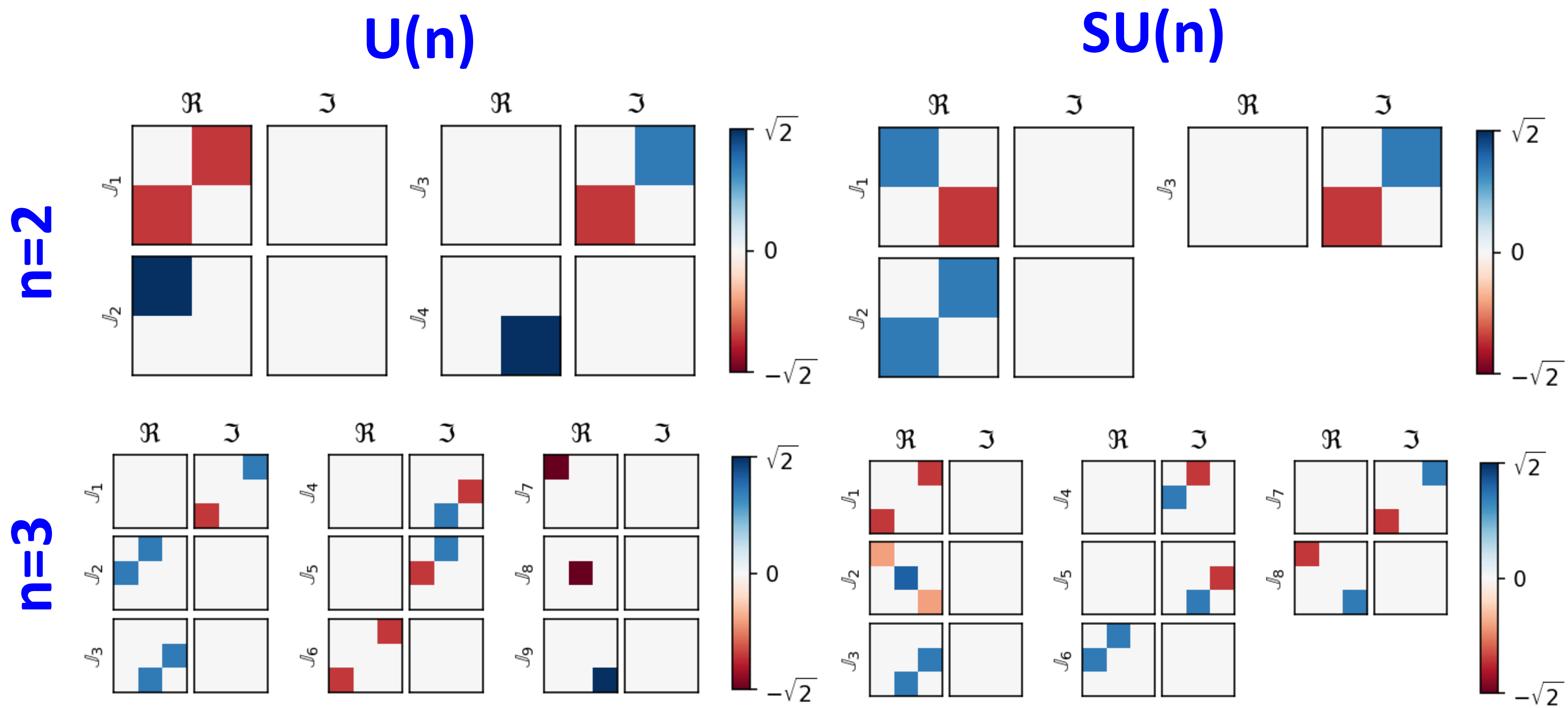


Sparse Representations: U(n) and SU(n)

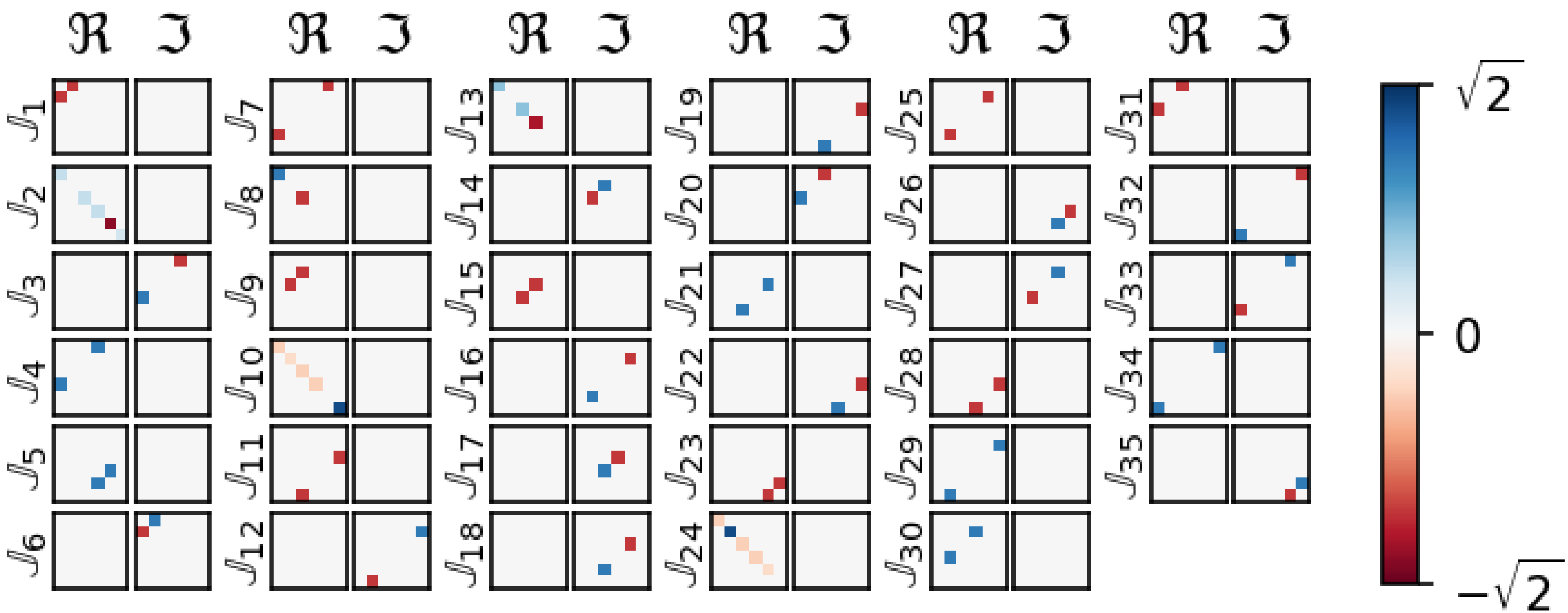
- The oracle for unitary groups:

$$\varphi_U(\mathbf{x}) \equiv \sum_{j=1}^{n} (x^{(j)})^* x^{(j)}, \quad x^{(j)} \in \mathbb{C}.$$

- The sparse representations were learned successfully



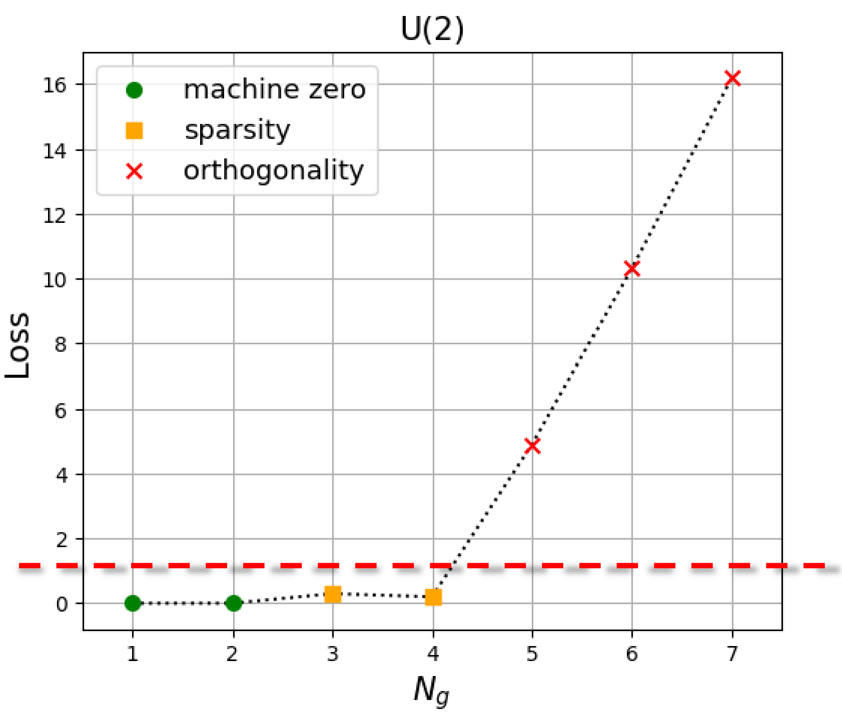
Tour de Force: SU(6)

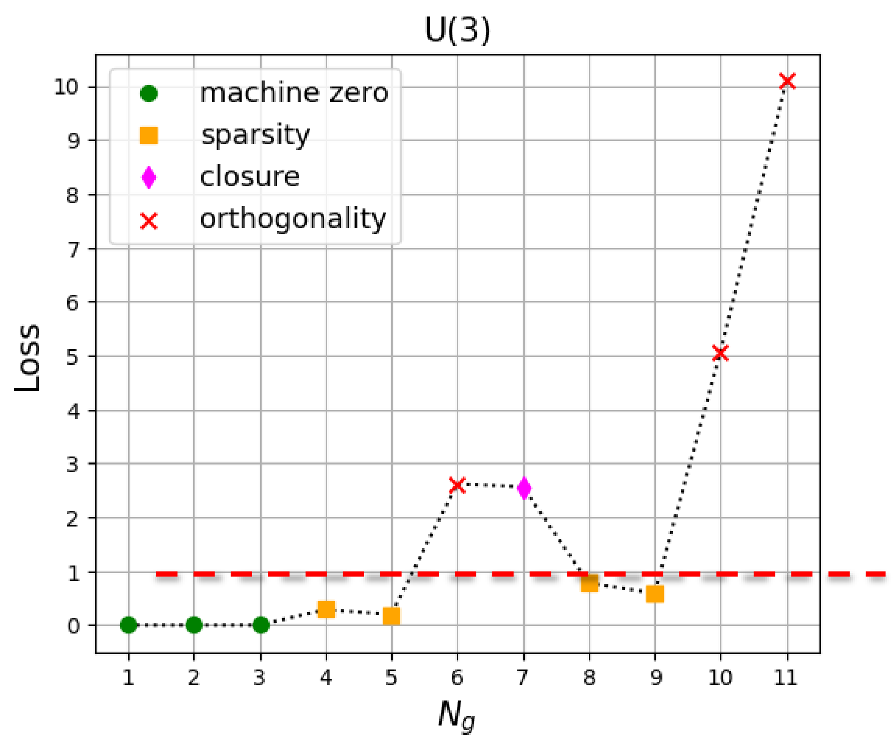


- Total number of generators: 6^2-1 =35
  - SU(6) is rank=5 => 5 real diagonal generators (2,8,10,13,24)
  - 15 symmetric real matrices
  - 15 anti-symmetric purely imaginary matrices

Unitary groups: subgroup structure

- The value of the trained loss function is an indicator of the presence or absence of a valid sub-algebra





Probing the subalgebra structure

- Test for all possible decompositions into factors

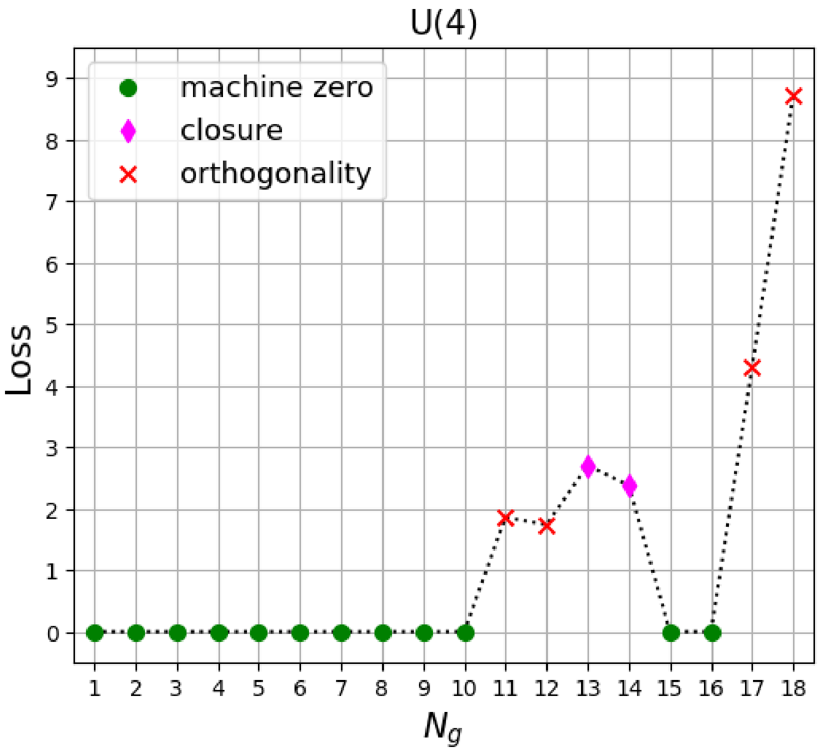
$$\mathfrak{h} = \mathfrak{h}_1 \oplus \mathfrak{h}_2 \oplus \cdots \oplus \mathfrak{h}_h$$

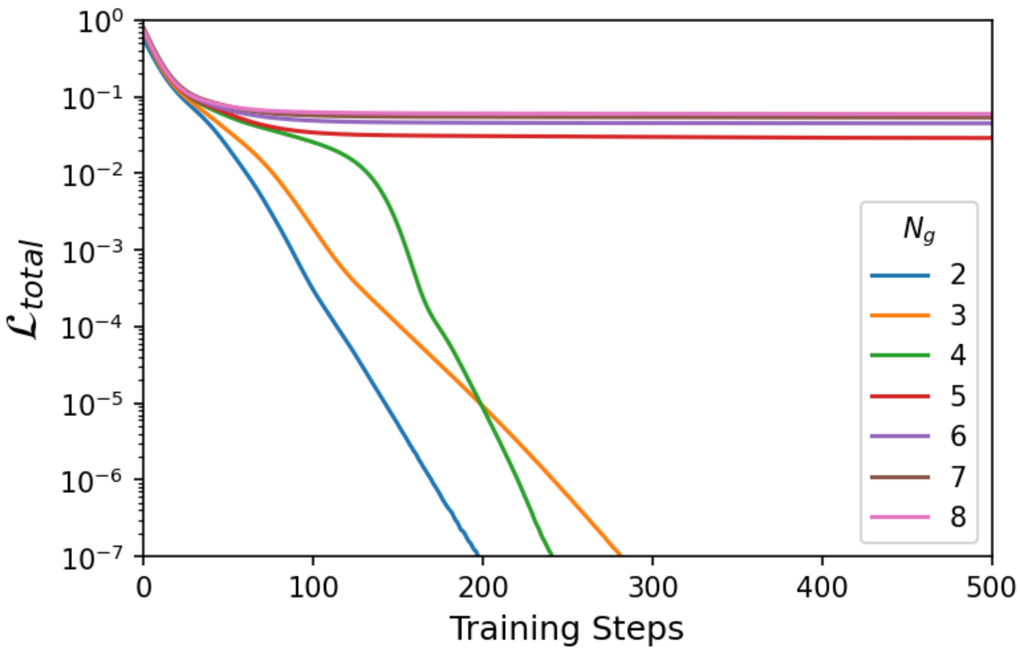
	Number of subalgebra factors h			
$N_{\mathfrak{h}}$	1	2	3	4
1	$\mathfrak{h}_1^{(1)}$	—	—	—
2	$\mathfrak{h}_1^{(2)}$	$\mathfrak{h}_1^{(1)} \oplus \mathfrak{h}_2^{(1)}$	—	—
3	$\mathfrak{h}_1^{(3)}$	$\mathfrak{h}_1^{(2)} \oplus \mathfrak{h}_2^{(1)}$	$\mathfrak{h}_1^{(1)} \oplus \mathfrak{h}_2^{(1)} \oplus \mathfrak{h}_3^{(1)}$	—
4	$\mathfrak{h}_1^{(4)}$	$\mathfrak{h}_1^{(3)} \oplus \mathfrak{h}_2^{(1)}$ $\mathfrak{h}_1^{(2)} \oplus \mathfrak{h}_2^{(2)}$	$\mathfrak{h}_1^{(2)} \oplus \mathfrak{h}_2^{(1)} \oplus \mathfrak{h}_3^{(1)}$	$\mathfrak{h}_1^{(1)} \oplus \mathfrak{h}_2^{(1)} \oplus \mathfrak{h}_3^{(1)} \oplus \mathfrak{h}_4^{(1)}$
⋮	⋮			

Cartan subalgebra

Less trivial example: u(4)

- The transformations are described with 4x4 complex matrices
- There exist valid subalgebras with the following number of generators:
  - 1 through 10, 15 and 16
- The Cartan subalgebra has 4 generators (the rank is 4)





The subalgebra structure of u(4)

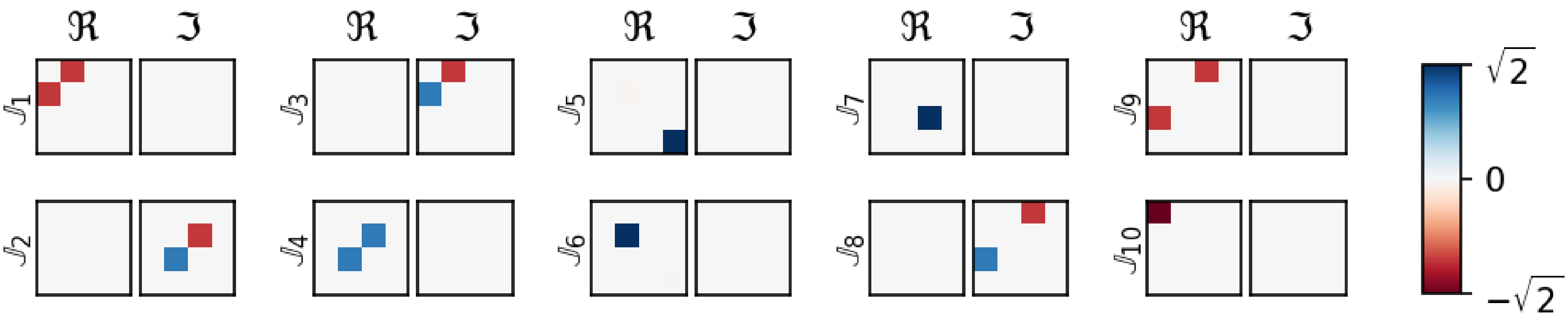
$$\mathfrak{h} = \mathfrak{h}_1 \oplus \mathfrak{h}_2 \oplus \cdots \oplus \mathfrak{h}_h$$

- Color code:
  - Green: valid decomposition
  - Orange: no such decomposition
- Different rows represent different (nonisomorphic) algebras
- Only up to 4 factors are possible
  - The rank is 4
- The circled examples are illustrated on the next slide

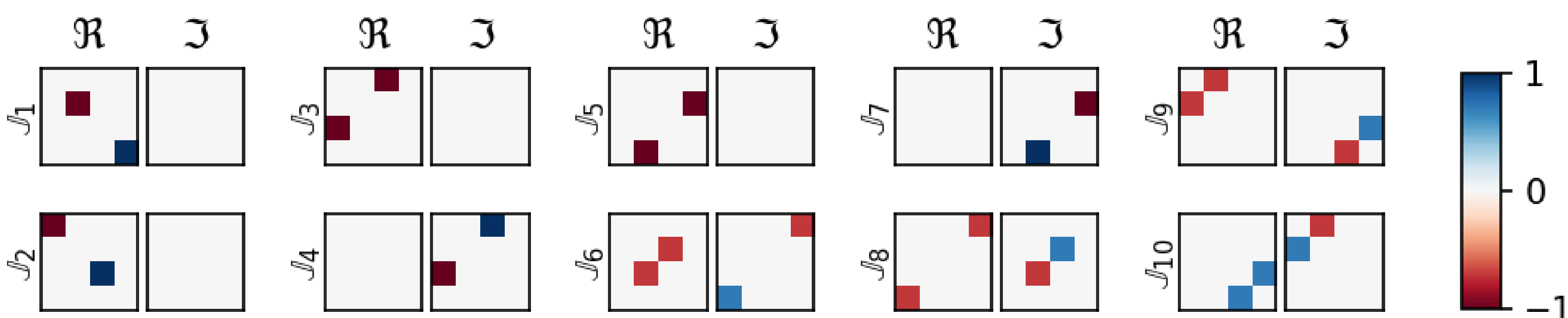
	Number of subalgebra factors h			
$N_{\mathfrak{h}}$	1	2	3	4
1	u_1	—	—	—
2	u_1^2	$u_1 \oplus u_1$	—	—
3	u_1^3 su_2	$u_1^2 \oplus u_1$	$u_1 \oplus u_1 \oplus u_1$	—
4	u_1^4 u_2	$u_1^3 \oplus u_1$ $su_2 \oplus u_1$	$u_1^2 \oplus u_1 \oplus u_1$	$u_1 \oplus u_1 \oplus u_1 \oplus u_1$
5	$\rightarrow$	$u_2 \oplus u_1$	$su_2 \oplus u_1 \oplus u_1$	
6	$\rightarrow$ so_4	$\rightarrow$ $su_2 \oplus su_2$	$u_2 \oplus u_1 \oplus u_1$	$su_2 \oplus u_1 \oplus u_1 \oplus u_1$
7	$\rightarrow$	$u_2 \oplus su_2$	$su_2 \oplus su_2 \oplus u_1$	
8	$\rightarrow$ su_3	$u_2 \oplus u_2$	$u_2 \oplus su_2 \oplus u_1$	$su_2 \oplus su_2 \oplus u_1 \oplus u_1$
9	u_3	$su_3 \oplus u_1$		
10	$\rightarrow$ sp_4	$u_3 \oplus u_1$	$su_3 \oplus u_1 \oplus u_1$	
11				
12				
13				
14				
15	su_4			
16	u_4	$su_4 \oplus u_1$		

The two u(4) subalgebras with 10 generators

- su(3) x u(1) x u(1)

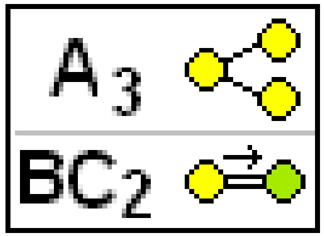


- sp(4) ~ so(5)



The so(5)~sp(4) subgroup of u(4)

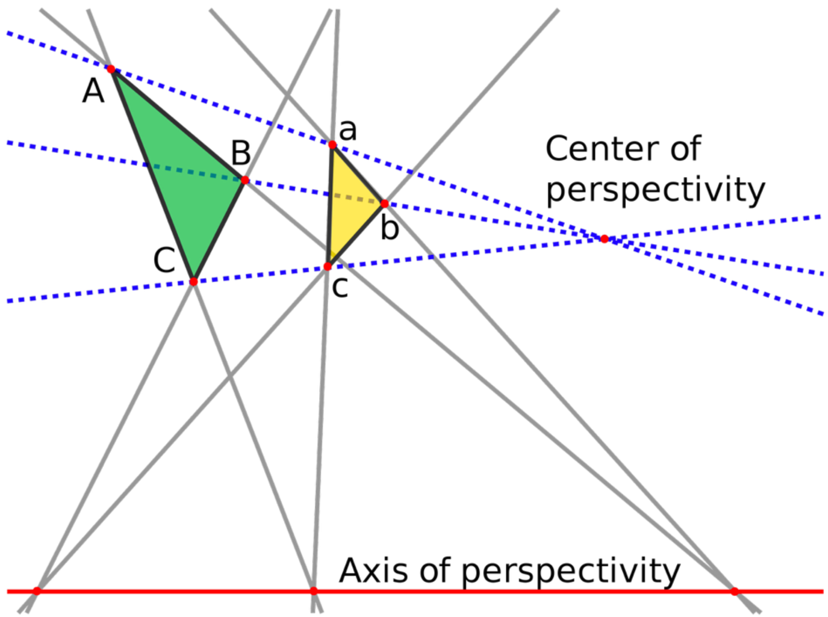
Eugene Dynkin
(1924-2014)



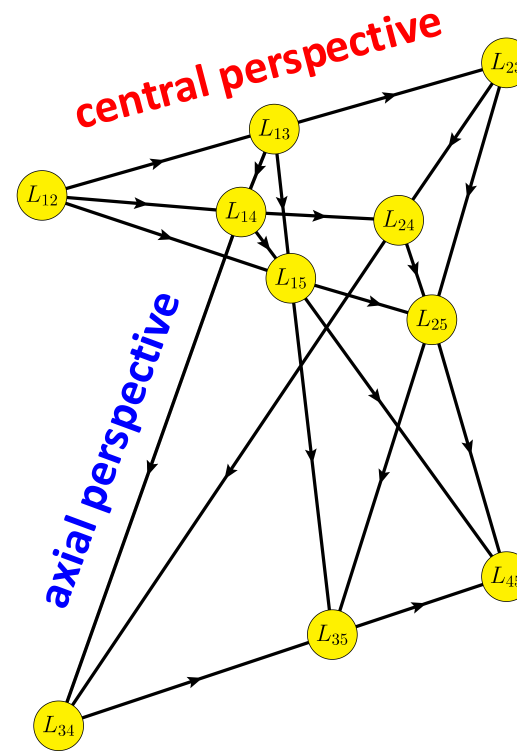
$A_3 = u(4)$

$B_2 = so(5)$
$C_2 = sp(4)$

Girard Desargues
(1591-1661)



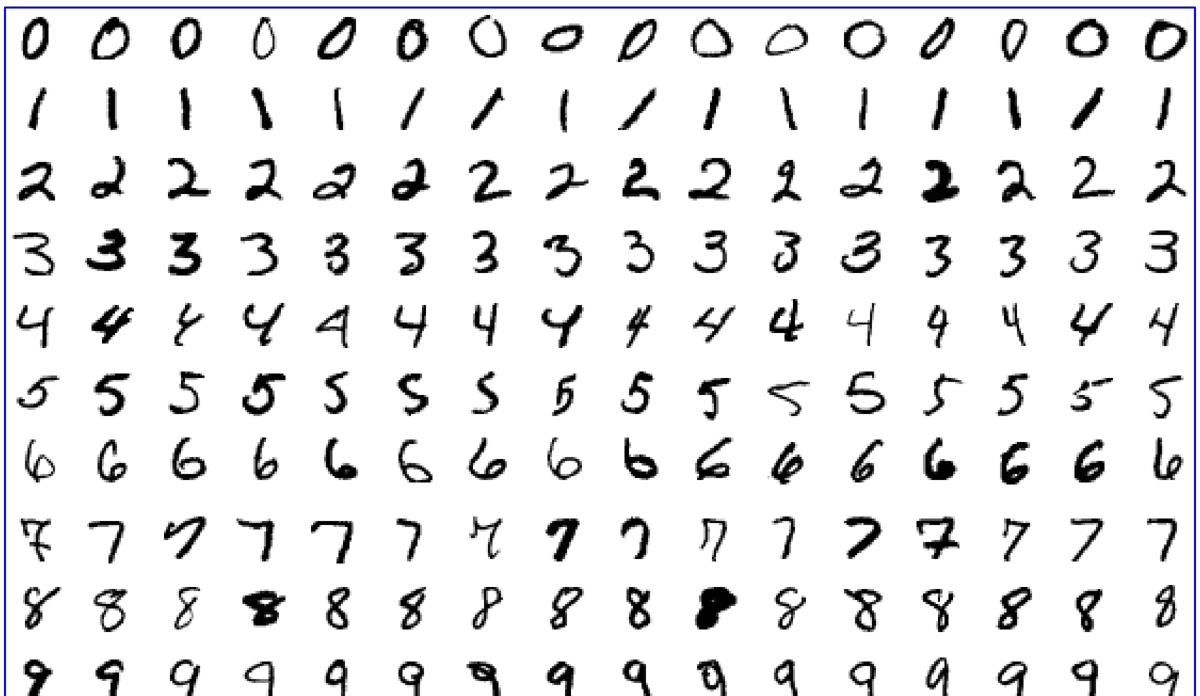
so(5) algebra



Desargues configuration

Symmetries of the MNIST Digits

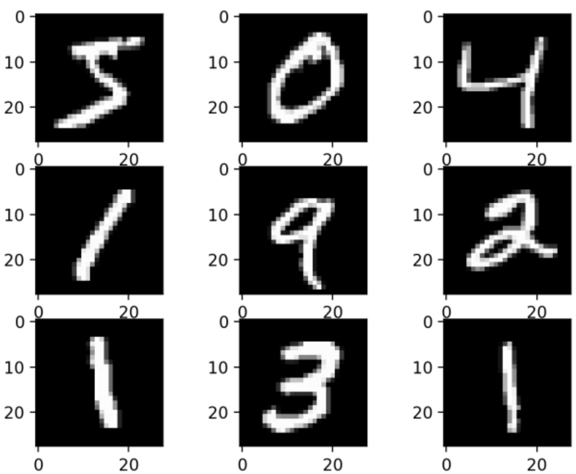
- The MNIST dataset is a widely used database for training and testing various image processing machine learning algorithms
- 70,000 handwritten digits as images of 28x28 pixels
- Typical applications
  - Multi-class classification (10 possible categories)
  - Data compression and manifold learning



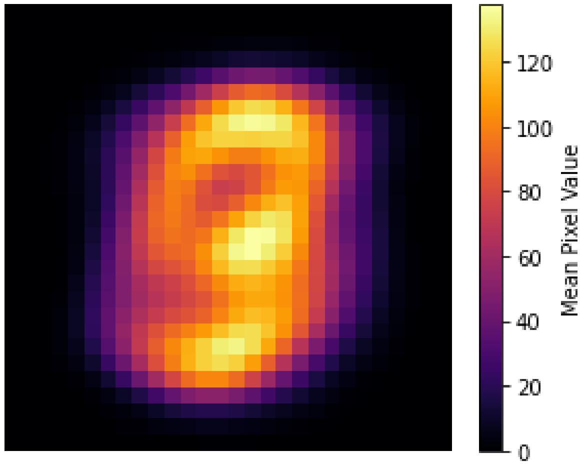
MNIST information content

- Not all of the 28x28=784 features (pixel values) carry equal amount of information
  - The information is mostly contained in the central pixels
- It is possible to find (almost) equivalent, reduced dimensionality representations of this data
  - **Latent space:** an embedding of the original features in a compressed representation
- We can try to look for symmetries in the latent space

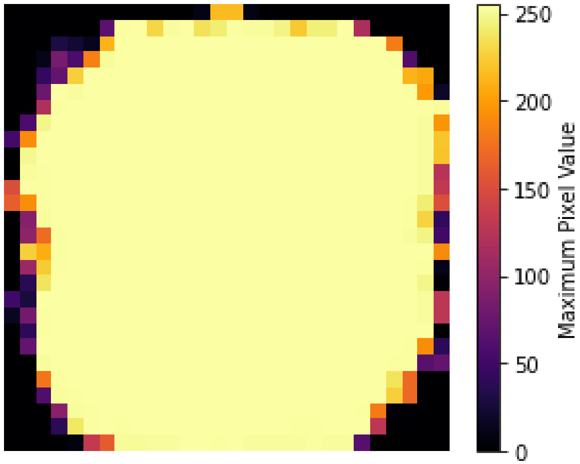
Typical images



Average pixel value



Maximum pixel value

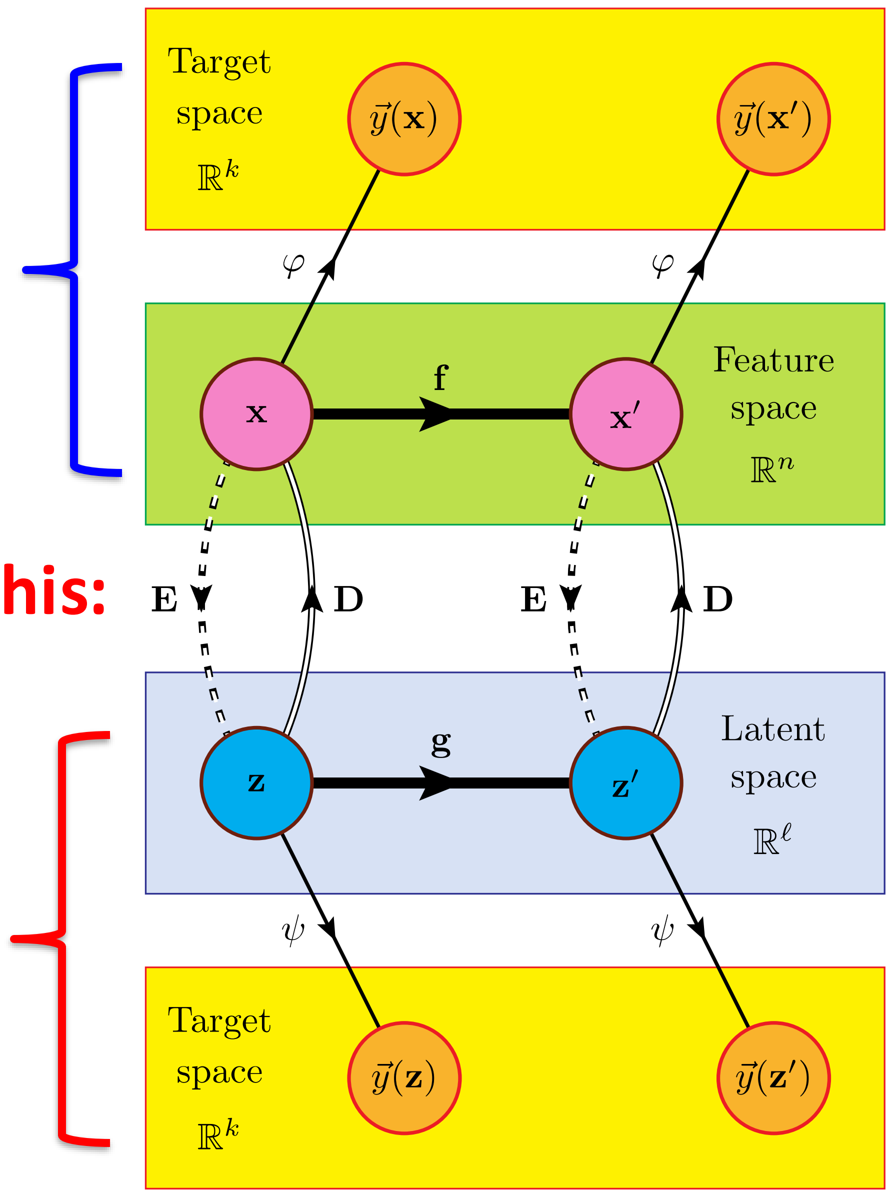


The Basic Idea

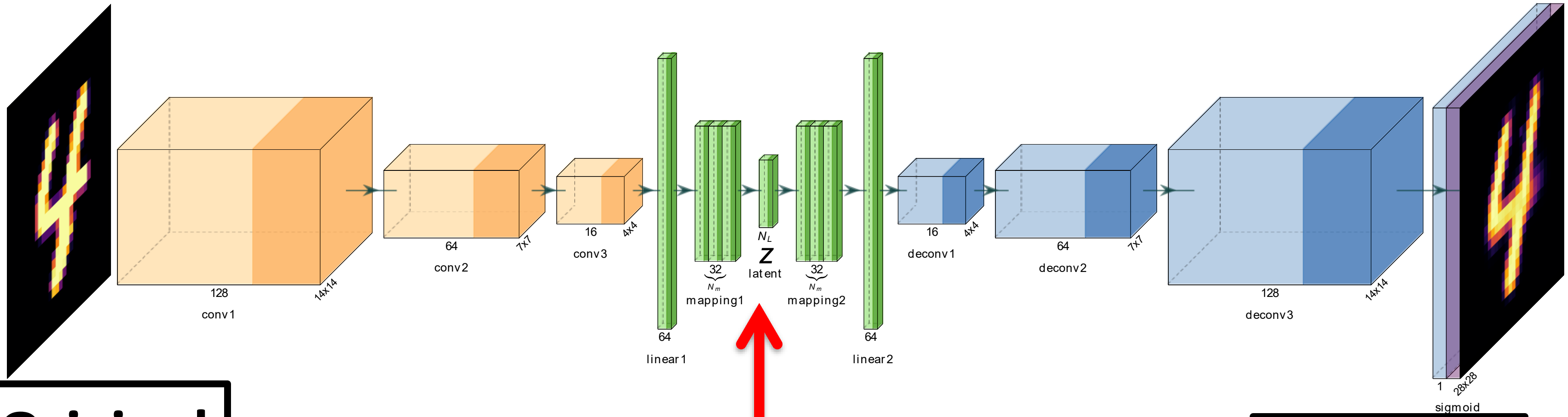
Previously we were doing this:

But first need to build this:

Now we'll be doing this:



Autoencoder Architecture



Original feature space x

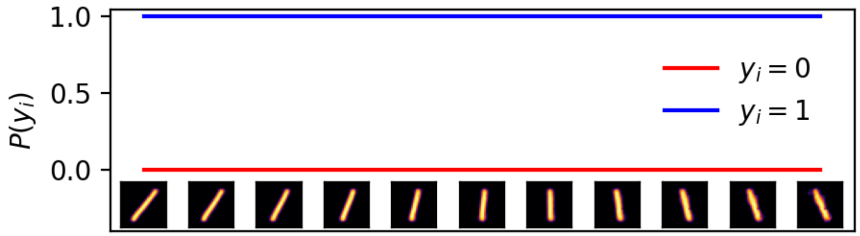
Latent space z

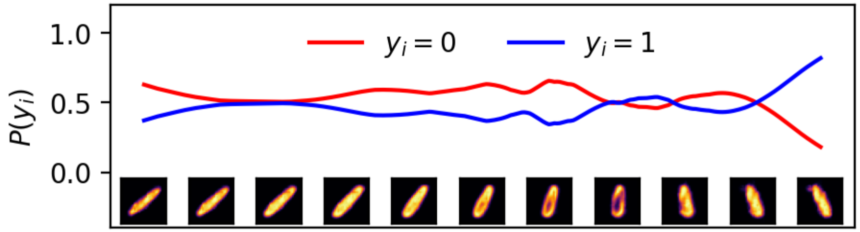
Decoded feature space x'

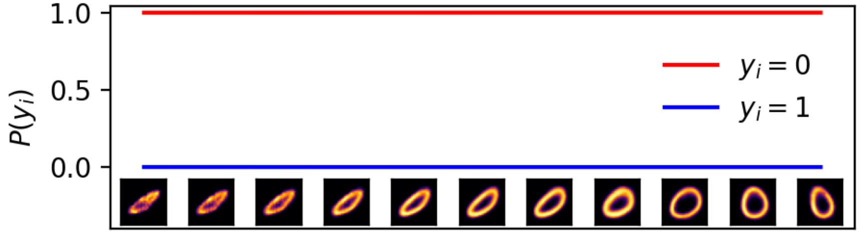
- Let's start with a toy example (next slide)
  - Keep only the 0 and 1 digits from the MNIST dataset
  - Compress to a **2D latent space**
  - Train a **binary classifier** (oracle) to produce the labels (logits)
  - Find the 1-parameter **symmetry transformation in the latent space**
  - Decode to **the feature space** to see what the transformation is doing

Symmetries of the digits 0 and 1

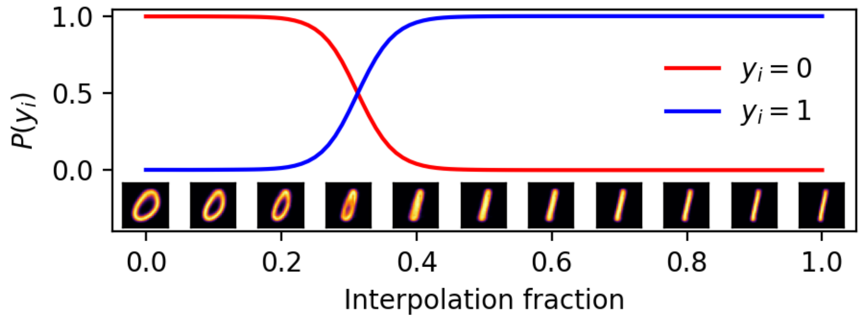
- Symmetry direction



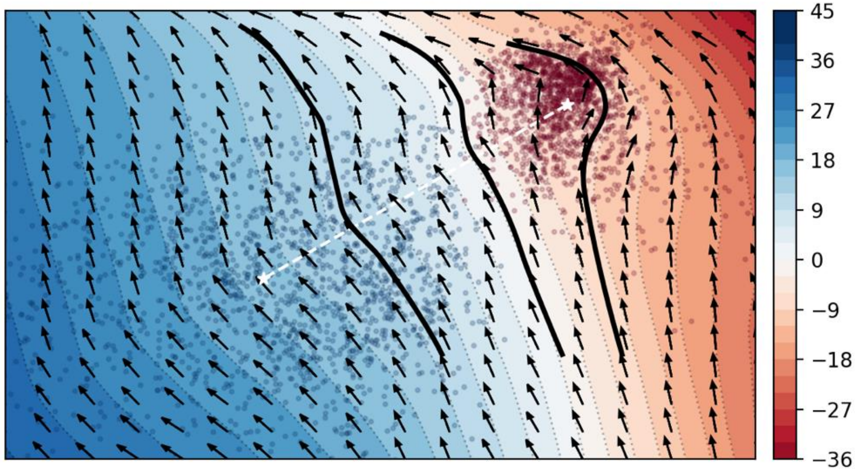




- Orthogonal direction

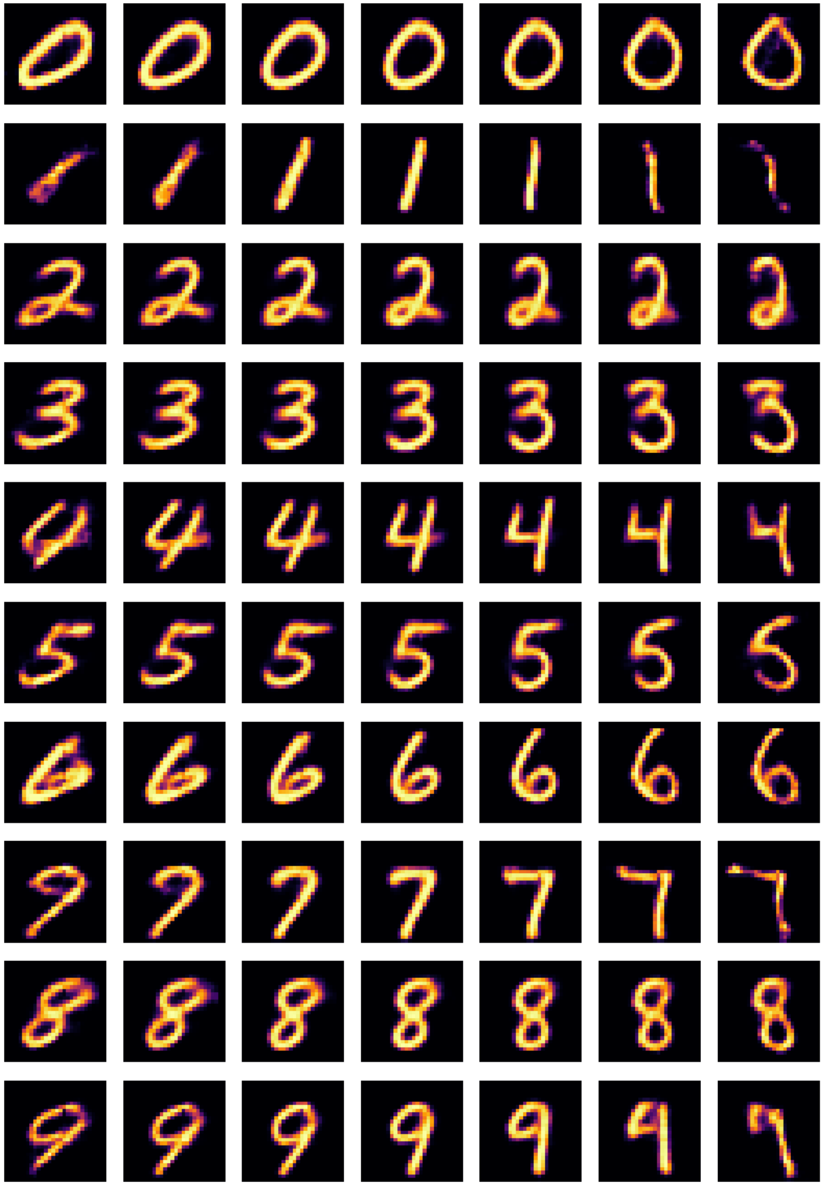


Latent space z



Symmetry deformations of the digits

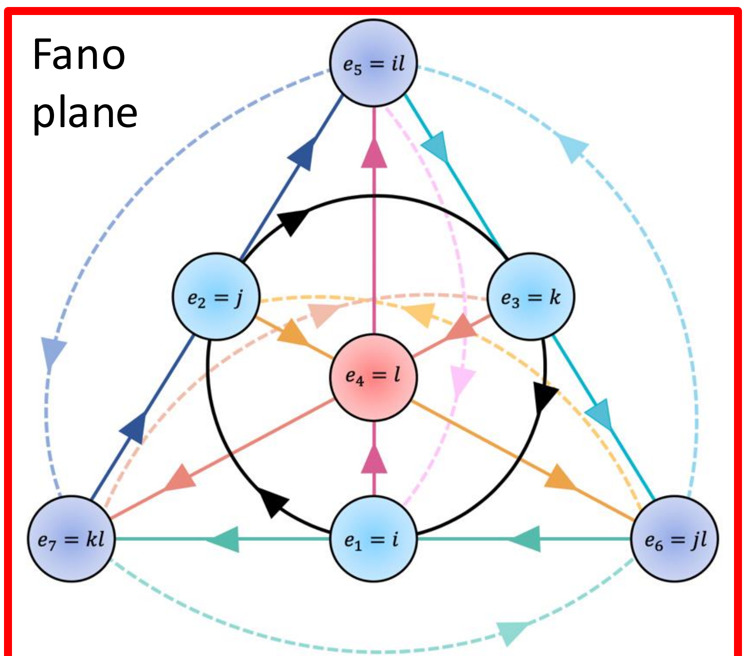
- Now the real exercise:
  - Keep all 10 digits
  - Compress to 16D **latent space**
  - Train a **10-class classifier**
  - Find the **symmetries**
- Starting from the platonic digits, follow a symmetry streamline in + and – direction
- The resulting images are depicted to the right
- The images in each row have equal values for all 10 logits (oracles), and are therefore classified the same way



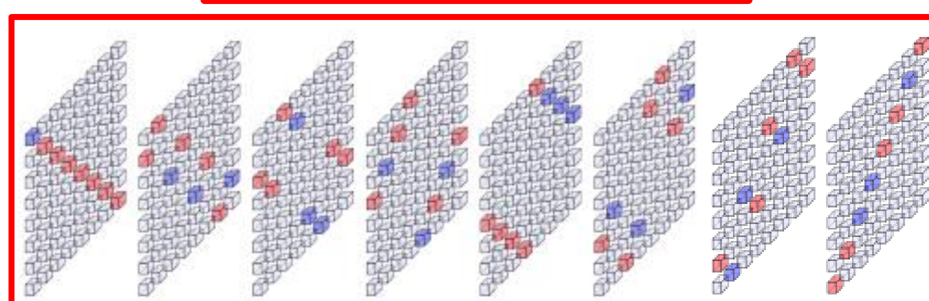
Refresher on Octonions

imagine.art

- Generalizations of the complex numbers
  - 7 imaginary units: {i, j, k, l, il, jl, kl}.
- Inherit the familiar operations:
  - multiplication, conjugation, norm, inverse
- Related to exceptional structures in mathematics
  - E.g.: Jordan algebra h_3 -> F_4 and E_6 exceptional Lie groups



$$h_3 = \begin{pmatrix} r_1 & o_1 & o_2 \\ \overleftarrow{o_1} & r_2 & o_3 \\ \overleftarrow{o_2} & \overleftarrow{o_3} & r_3 \end{pmatrix}$$

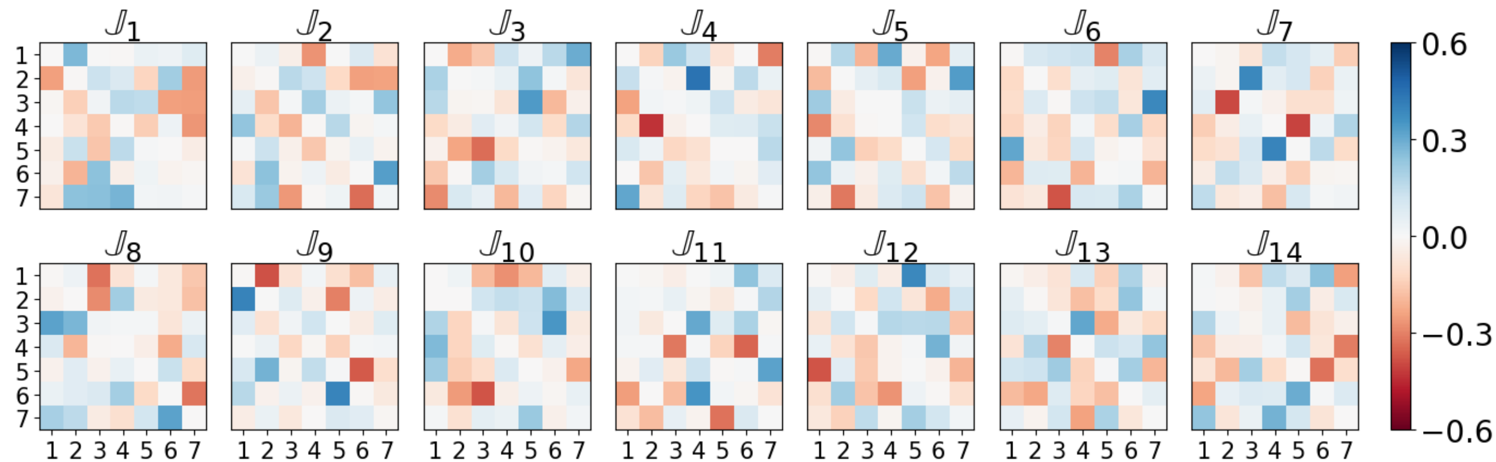


Real component of a triple octonion product

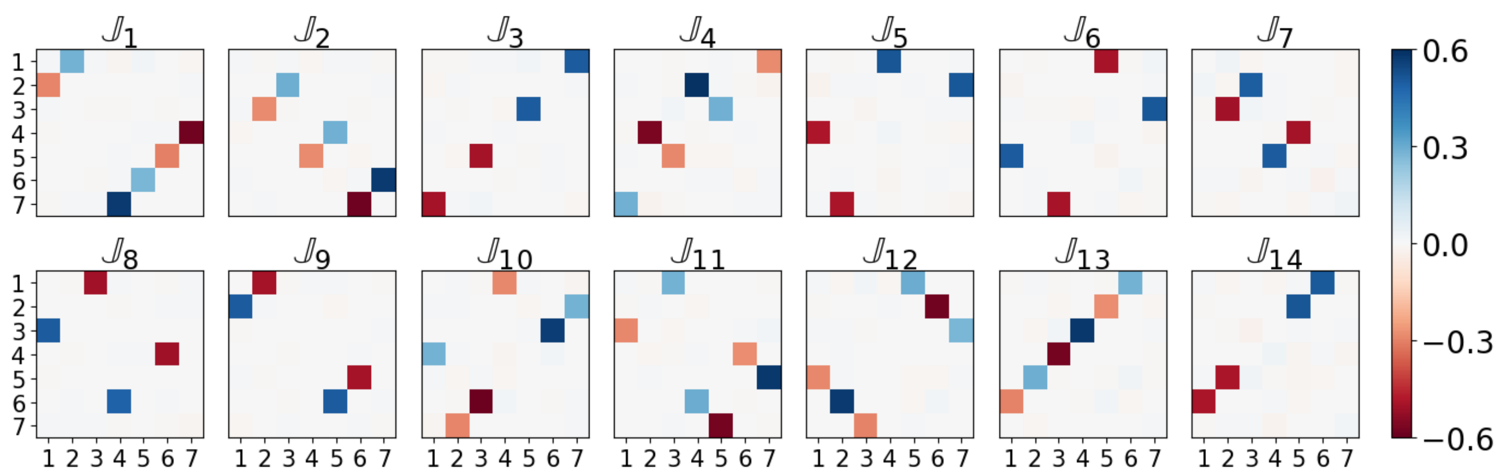
Exceptional Groups: G_2

Two oracles: $\phi^{(1)}_{G_2}(x) = \sum_{i=1}^{7} (x^{(i)})^2$ + $\phi^{(2)}_{G_2}(x_1, x_2, x_3) = \sum_{i,j,k=1}^{7} \Delta_{ijk}\, x_1^{(i)}\, x_2^{(j)}\, x_3^{(k)}$

Non-sparse version



Sparse version*



*sparser than on Wikipedia

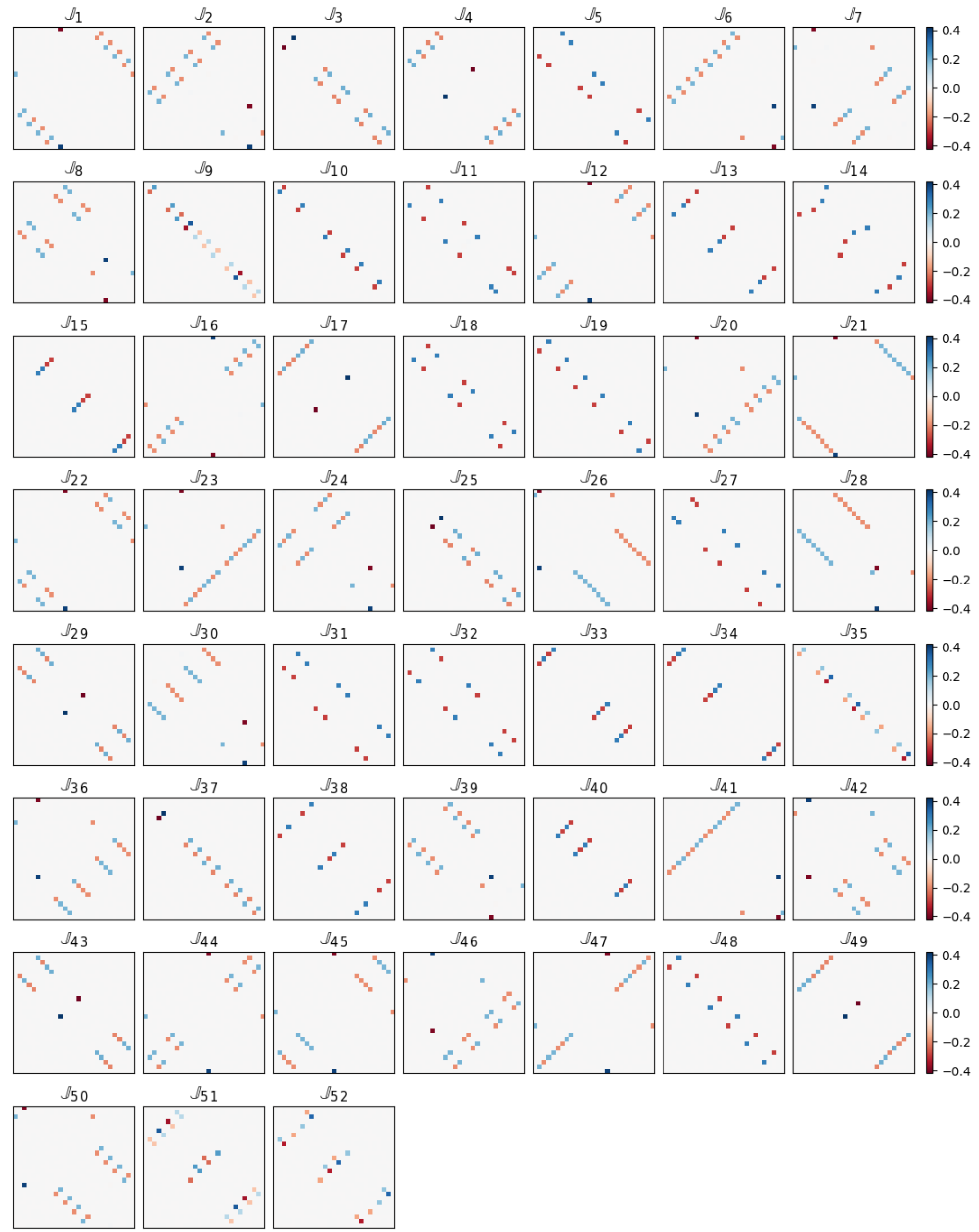
The F_4 group

- There are three oracles
- n=27 dimensional real feature space
- F_4 has 52 generators
- Our result matches previous results in the literature obtained with Mathematica

$$\varphi^{(1)}_{F_4}(\mathbf{x}) = \mathrm{Tr}\,\mathfrak{h}_3 = \sum_{a=1}^{3} r_a$$

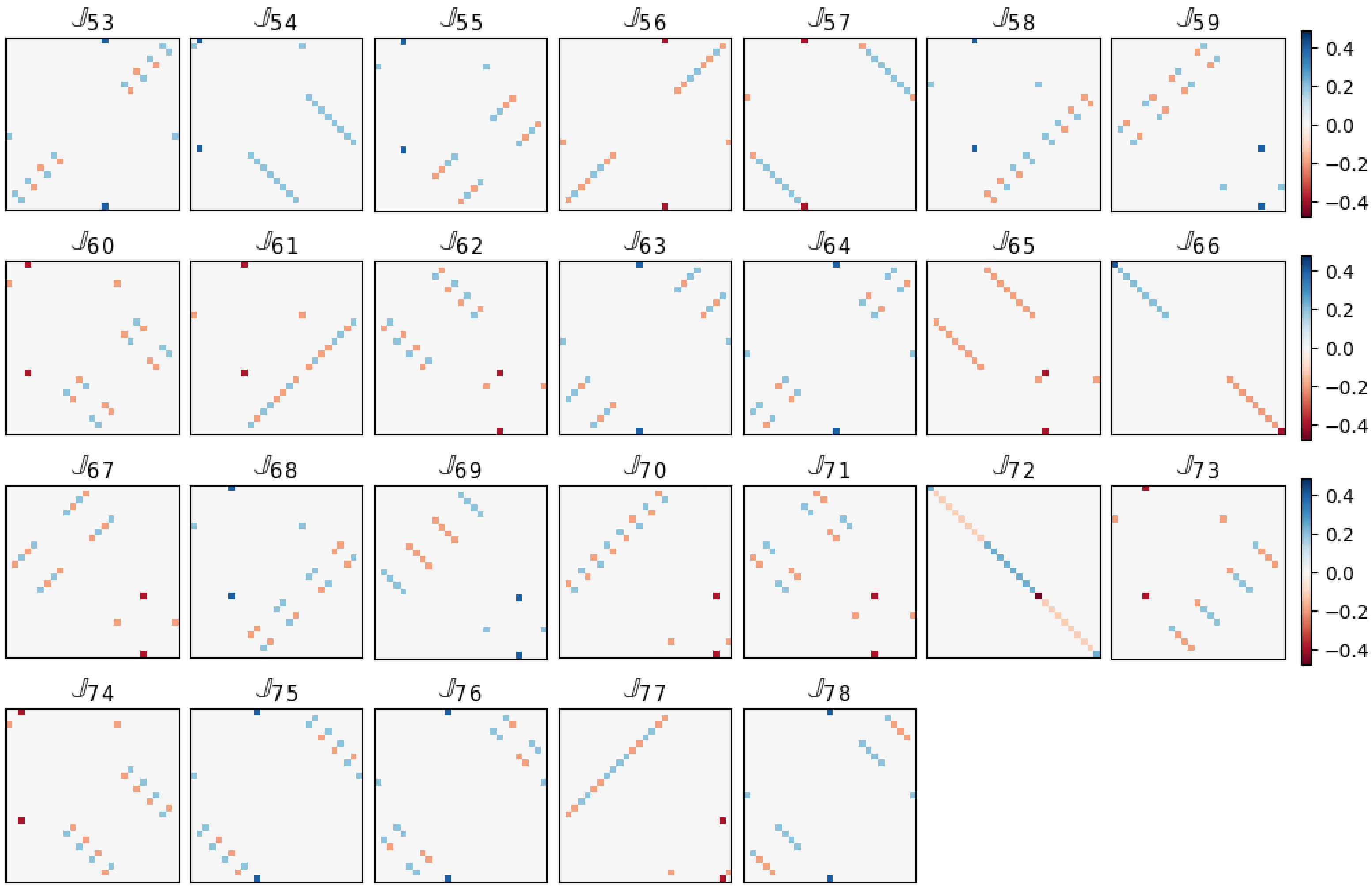
$$\varphi^{(2)}_{F_4}(\mathbf{x}) = \mathrm{Tr}\,\mathfrak{h}_3^2 = \sum_{a=1}^{3} \left(r_a^2 + 2\,|\mathbf{o}_a|^2 \right)$$

$$\varphi^{(3)}_{F_4}(\mathbf{x}) = \det \mathfrak{h}_3 = r_1 r_2 r_3 - \sum_{a=1}^{3} r_a\,|\mathbf{o}_{4-a}|^2 + 2\,\mathrm{Re}\left(\mathbf{o}_3 \mathbf{o}_2^* \mathbf{o}_1\right)$$



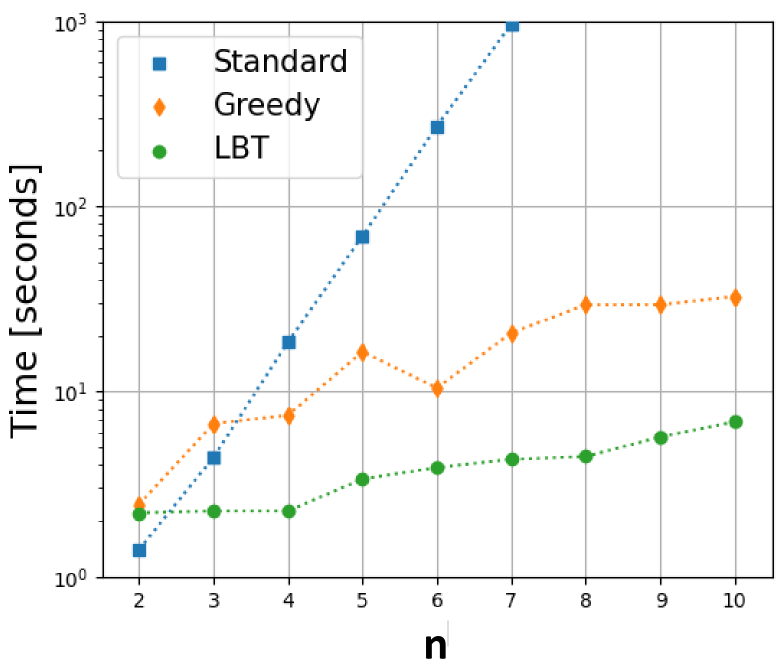
The E_6 group

- For E_6 only the last oracle applies => E_6 contains F_4 as a subgroup
  - The additional 26 generators beyond those of F_4 are shown below
- Our results match those previously derived with Mathematica



Speeding up the learning process

- A: learning all symmetry generators at once
  - pro: we can ensure closure
  - con: slow in high dimensions or for many symmetries
- B: learn one symmetry generator at a time
  - pro: much faster
  - con: delay the study of group properties to a post-processing stage
- Timing tests for SU(n):



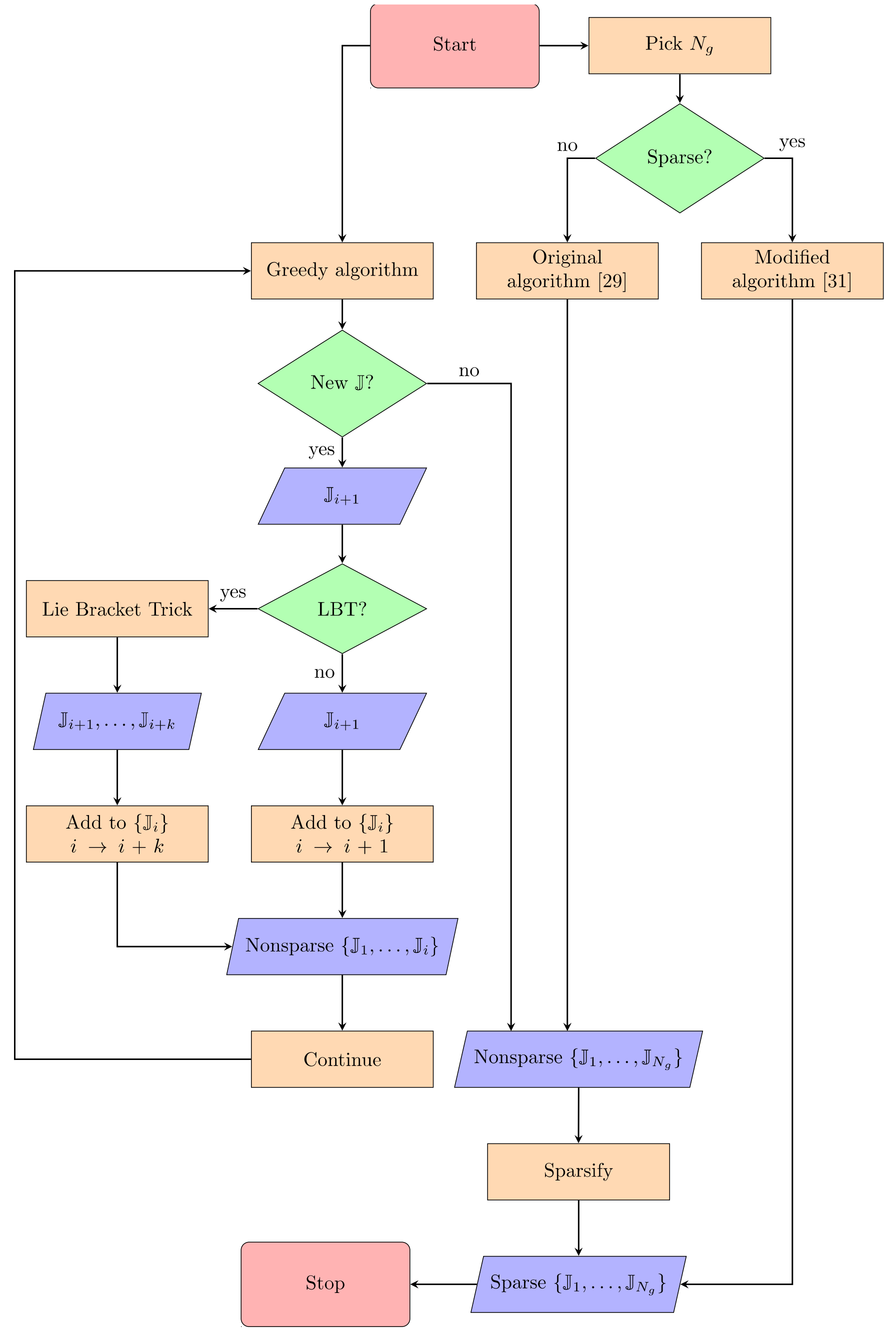
Algorithm 1: The greedy algorithm.

```
Parameters: λ, L_min, N_epochs;
{J} ← [];
W ← W_initial ~ N;
for i from 1 to N_epochs do
    L ← L_greedy(G(W), {J}, x) ;
    if L < L_min then
        append G(W) to {J};
        goto 3;
    end
    W ← W − λ∇_W L_greedy;
end
stop
```

Algorithm 2: The Lie bracket trick (LBT) algorithm.

```
Input: {J_1, ..., J_i}: known algebra; J_{i+1}: new
  generator;
append J_{i+1} to G;
repeat
    k    |G|;
    i    |J|;
    append G to J;
    clear G;
    for p from 1 to i do
        for q from i+1 to i+k do
            C    J_p J_q − J_q J_p;
            C    C − g2J g/||g|| →(C · g);
            if ||C|| , 0 then
                C    C/||C||;
                if L_inv(C, x) < L_min then
                    append C to G;
                end
            end
        end
    end
until |G| = 0;
```

Different Flavors of Symmetry Learning Algorithms



Summary and Future Directions

- We have a method (and a public python code) to derive the **symmetry** algebra of a **labelled dataset**
- Many known results from **group theory** can be rederived and verified (useful teaching tool)
  - Orthogonal groups
  - Lorentz group
  - Unitary groups
  - Exceptional groups
- It would be interesting to apply this to real datasets and discover **unexpected or unknown symmetries**
  - The MNIST example sets the blueprint
- **Experimental mathematics** is a lot of fun!

Main References

- The papers

- The python code

BACKUP SLIDES

Symbolic Learning

THE ASTROPHYSICAL JOURNAL, 930:33 (13pp), 2022 May 1 https://doi.org/10.3847/1538-4357/ac610c

CrossMark

Analytical Modeling of Exoplanet Transit Spectroscopy with Dimensional Analysis and Symbolic Regression

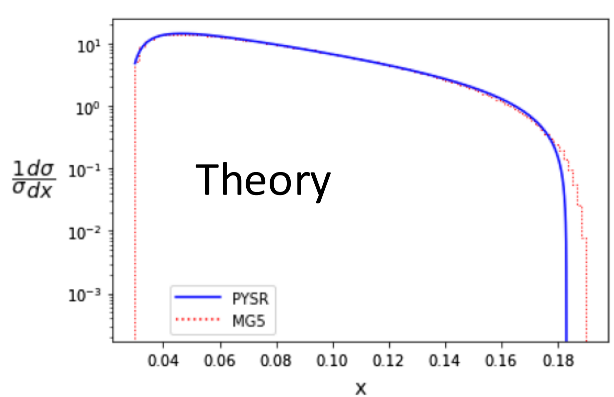
Konstantin T. Matchev, Katia Matcheva, and Alexander Roman
Physics Department, University of Florida, Gainesville, FL 32611, USA
Received 2021 December 23; revised 2022 March 20; accepted 2022 March 22; published 2022 May 2

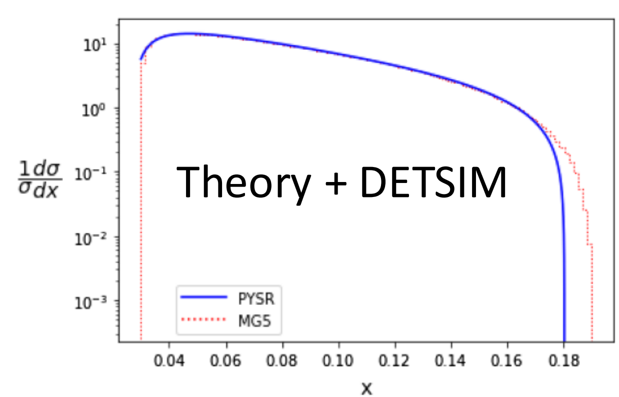
PHYSICAL REVIEW D **107,** 055018 (2023)

Is the machine smarter than the theorist: Deriving formulas for particle kinematics with symbolic regression

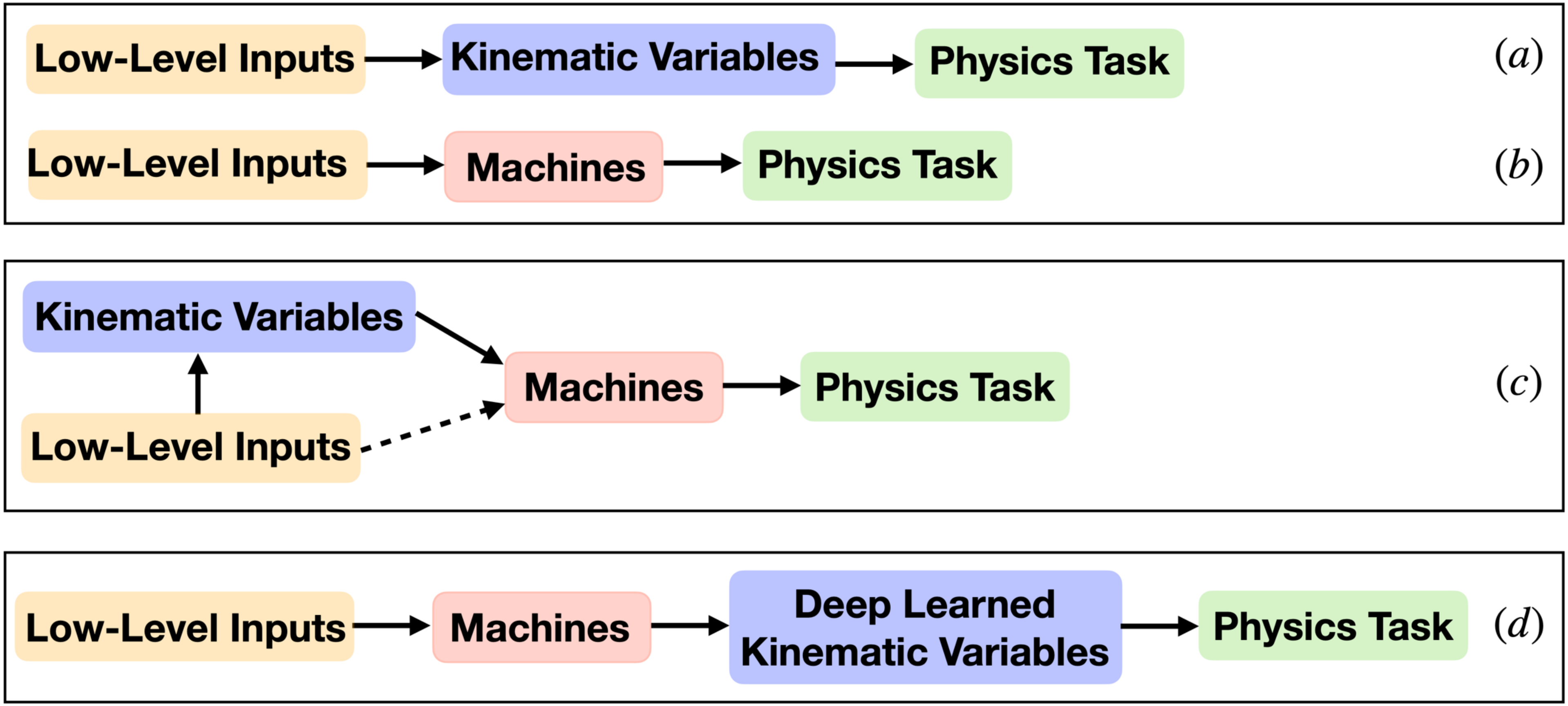
Zhongtian Dong,[1,*] Kyoungchul Kong,[1,†] Konstantin T. Matchev,[2,‡] and Katia Matcheva[2,§]
[1]*Department of Physics and Astronomy, University of Kansas, Lawrence, Kansas 66045, USA*
[2]*Institute for Fundamental Theory, Physics Department, University of Florida, Gainesville, Florida 32611, USA*
(Received 18 November 2022; accepted 27 February 2023; published 13 March 2023)

$$e^+e^- \to \chi\chi + \gamma$$





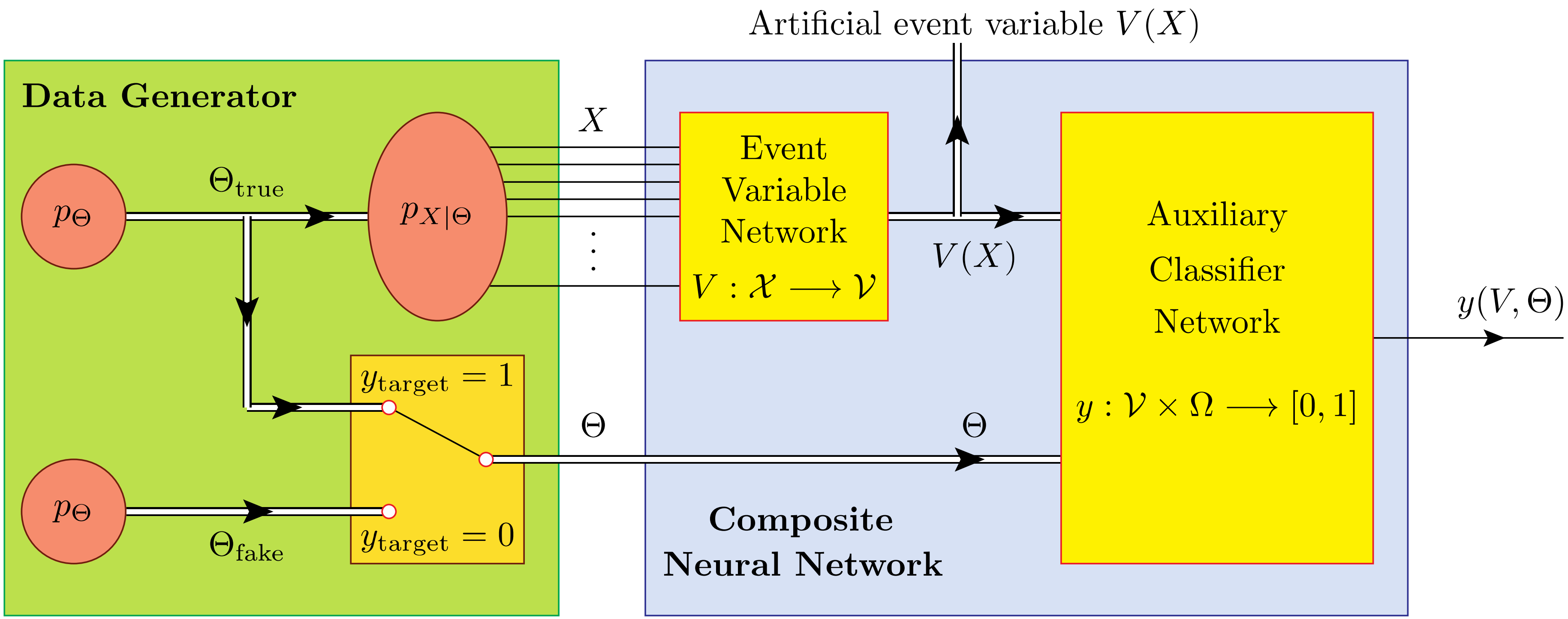
Feature Engineering with Machine Learning



Franceschini,Kim,Kong,KM,Park,Shyamsundar RMP 2023

Machine-Learned Event Variables

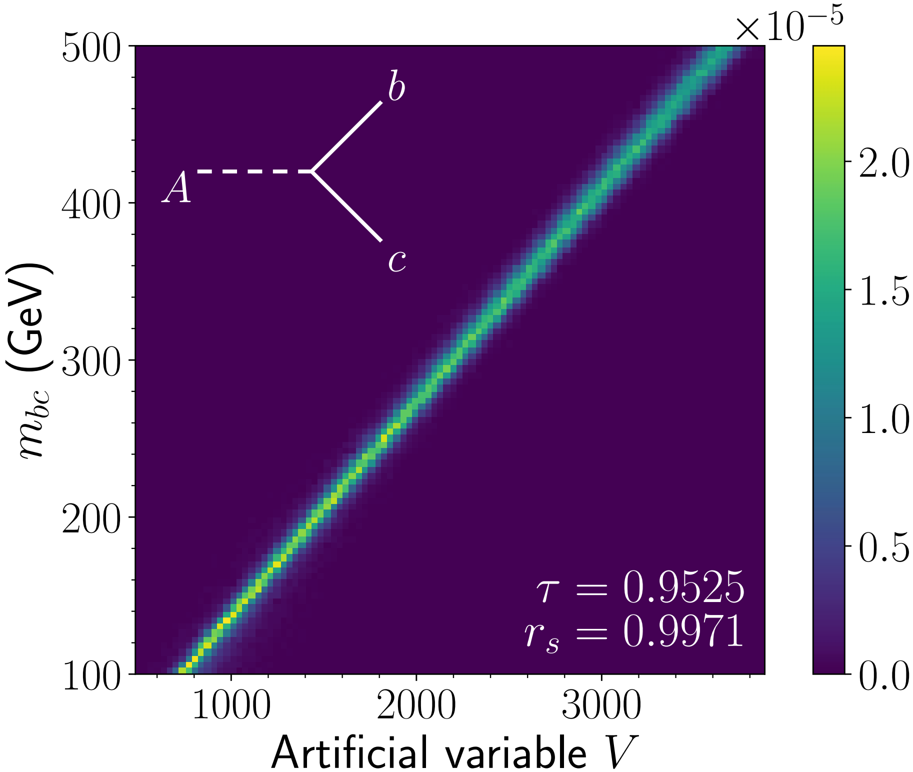
- Let's ask the machine to learn a good event variable

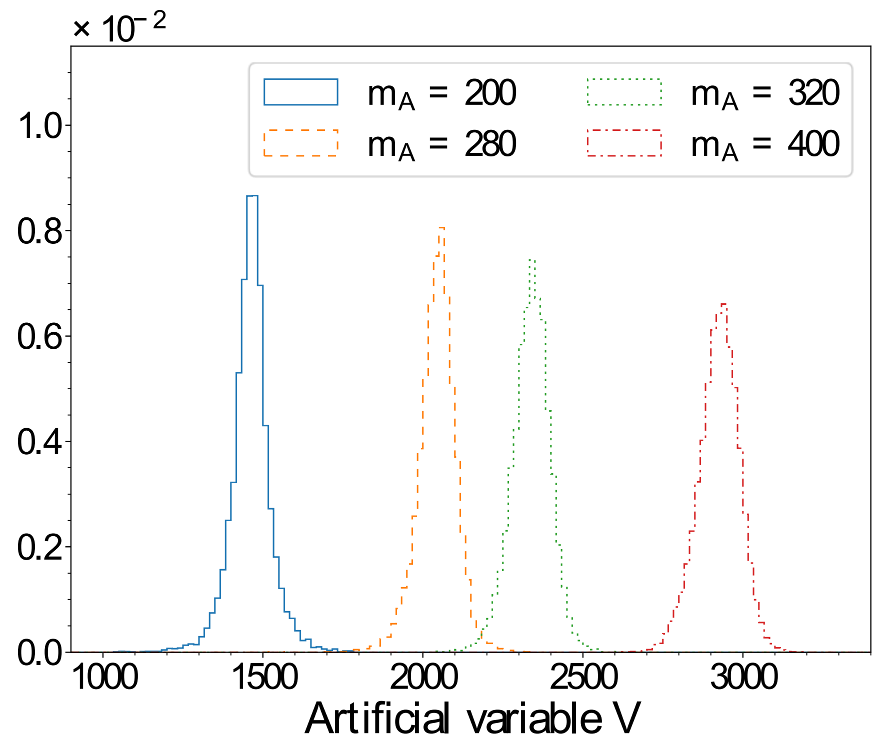


Kim, Kong, KM, Park, Shyamsundar (2021)

Example 1: Drell-Yan events

- Feed the momenta of the two leptons, ask the machine to guess if they came from a resonance A with a mass M_A

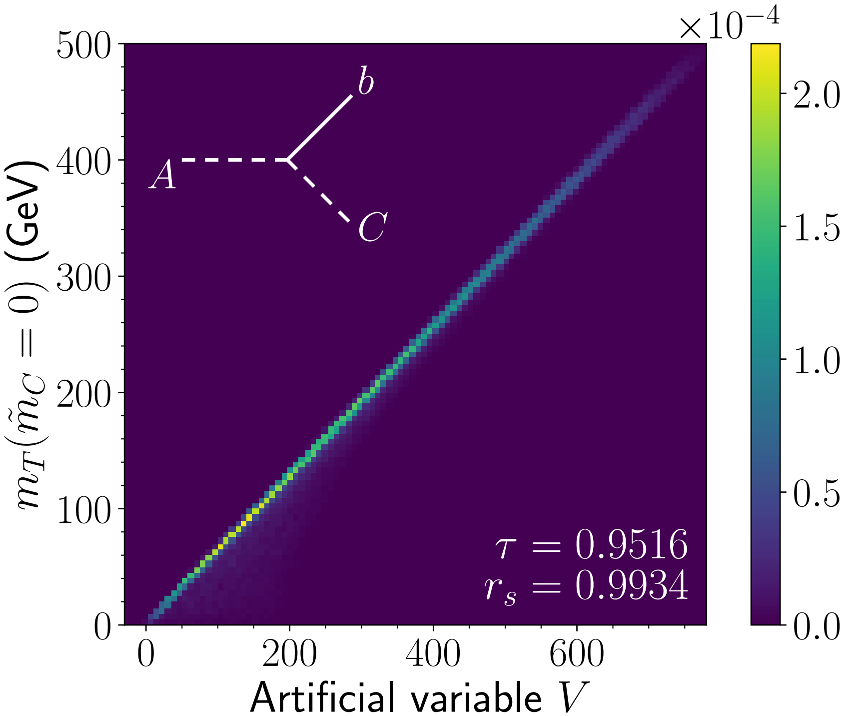


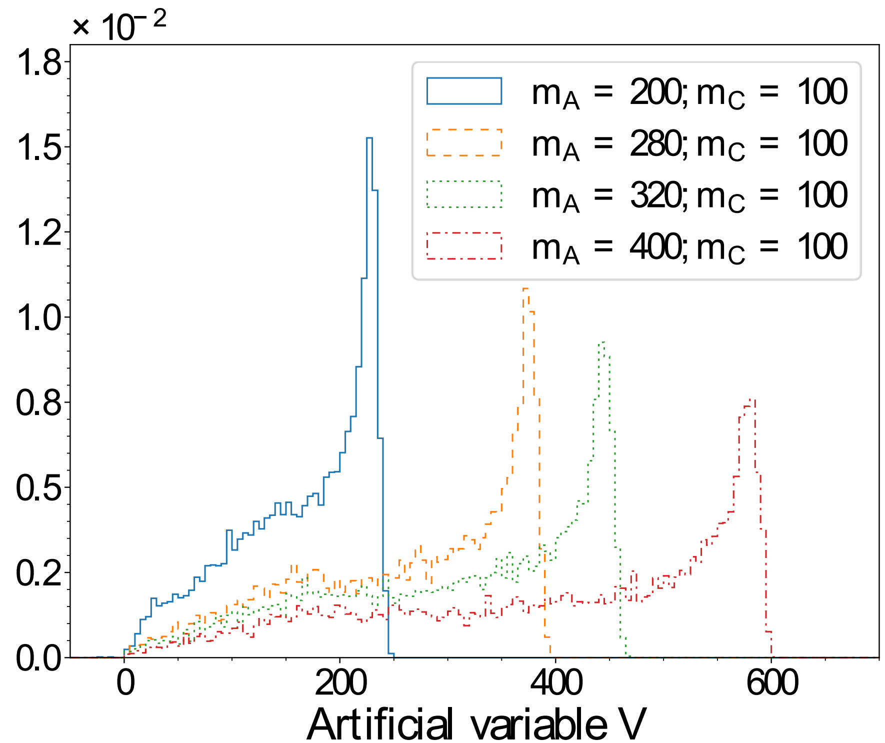


Kim, Kong, KM, Park, Shyamsundar (2021)

Example 2: leptonic W events

- Feed the MET vector (transverse momentum of C) and the momentum of the lepton (b), ask the machine to guess if the event came from a resonance A with a mass M_A

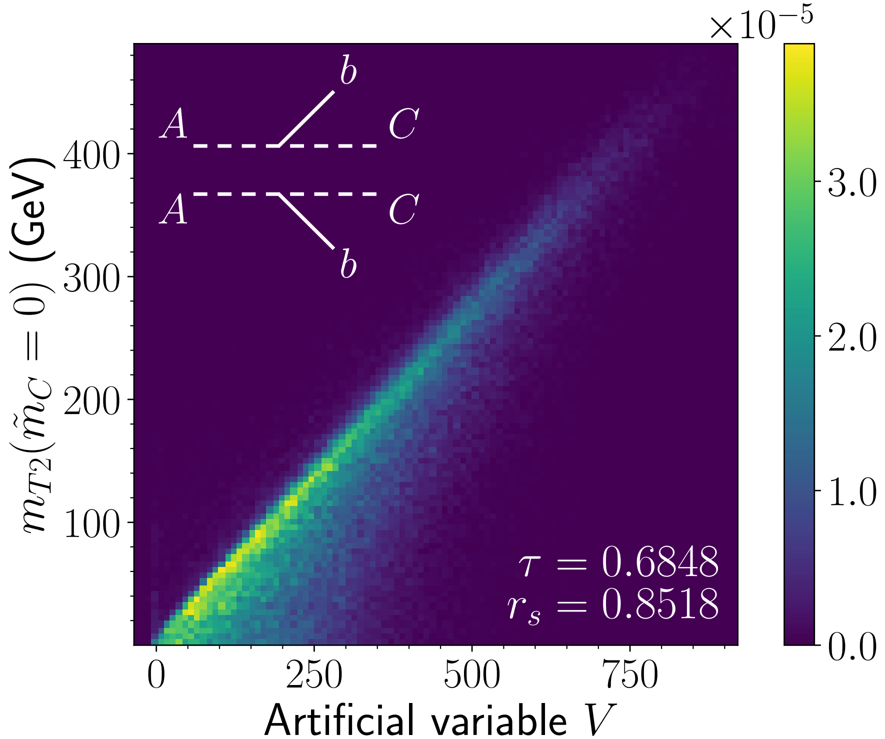


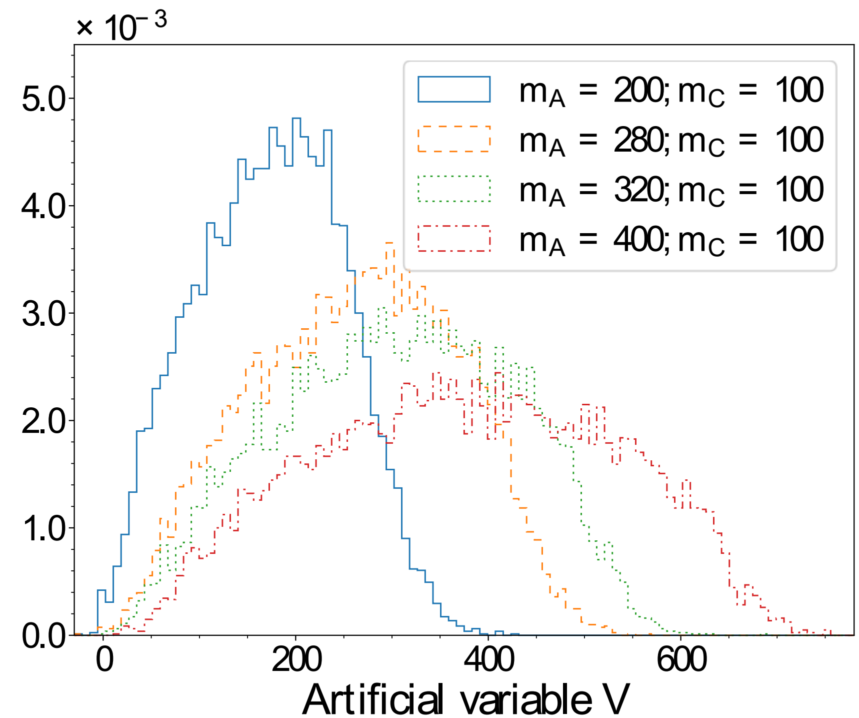


Kim, Kong, KM, Park, Shyamsundar (2021)

Example 3: leptonic W-pair events

- Feed the MET vector (net transverse momentum of the two C's) and the momentum of the two leptons (b's), ask the machine to guess if the event came from the pair production of resonances A with a mass M_A

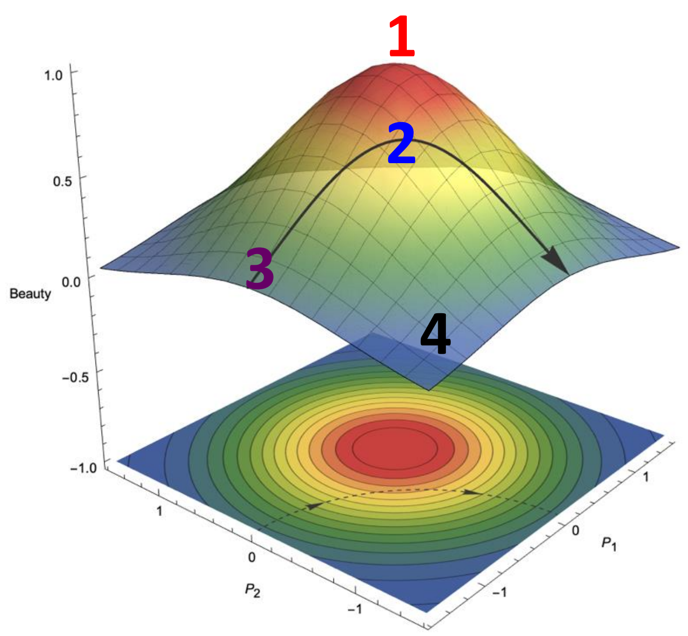




Kim, Kong, KM, Park, Shyamsundar (2021)

Truth and Beauty in Particle Theory

- **Truth:** the model fits the existing experimental data
  - A measurement of an observable O places a constraint on the model parameters p_i (dashed line)
- **Beauty:** a subjective criterion influenced by personal preferences, community views, etc.
  - We consider examples where beauty can be quantified (z-axis)



- Examples of theory models:
  - **1**: beautiful and wrong
  - **2**: beautiful and true
  - **3**: ugly and true
  - **4**: ugly and wrong

***Matchev, Matcheva, Ramond, Verner 2023**
Accepted at NeurIPS 2023 (poster)

Beauty = Uniformity

- In each run, inputs are re-sampled within the experimental errors
- The training results in low values of the total loss (10 runs, right plot)
- Pictorial representation of the result from a typical run: