

# AI and ML for Particle Physics

Konstantin Matchev  
University of Alabama



**Φαινó 2026**  
**The 2026 Phenomenology Symposium**  
*An odyssey through particle physics and related encounters in astrophysics and cosmology*

**University of Pittsburgh**  
**May 11-13, 2026**  
[indico.global/e/phenoz6](http://indico.global/e/phenoz6)

**Organizers:**  
Brian Batell  
Kun Cheng  
Christopher Condon  
Arnab Dasgupta  
Ayres Freitas  
Jon George  
Gracie Gollinger  
Tao Han (chair)  
Sam Hostler  
Adam Leibovich  
Zahra Tabrizi

**Program Advisors:**  
Vernon Barger  
Lisa Everett  
JoAnne Hewett  
Tae Min Hong  
Arthur Kotzevsky  
James Mueller  
Vittorio Palomonte  
Yitang Piao  
Laura Reina  
Vladimir Savinov  
Liam T. Wang  
Andrew Zentner

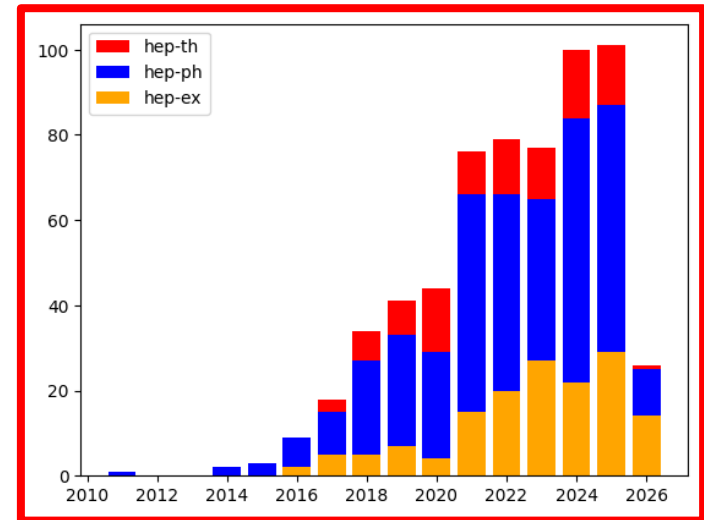
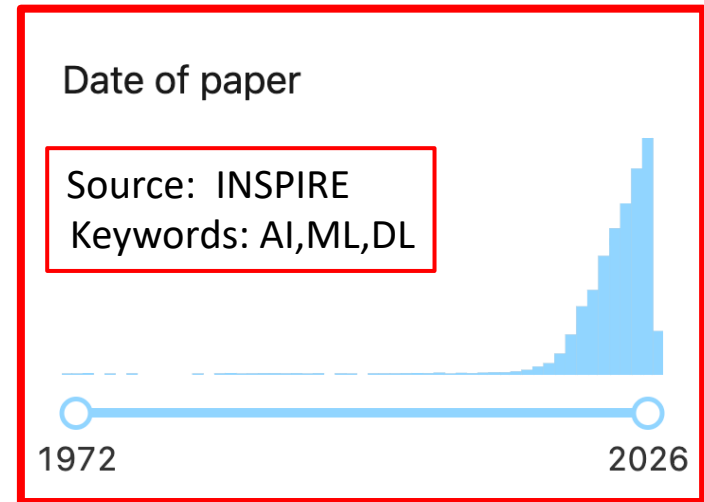
With financial support from the National Science Foundation and Department of Energy

PHENO Symposium  
University of Pittsburgh  
May 13, 2026



# Outline

- AI interlude
- The particle physics enterprise
- “Narrow AI” applications in theory and phenomenology
  - Optimal kinematic variables
  - Symbolic learning
  - Symmetry discovery (in math)
  - Symmetry discovery (in data)
  - Model building
- Agentic scientific workflows
  - **HEPTAPOD** (HEP Toolkit for Agentic Planning, Orchestration & Deployment)
  - Agentic **Diagrammatica**
- Related resources:

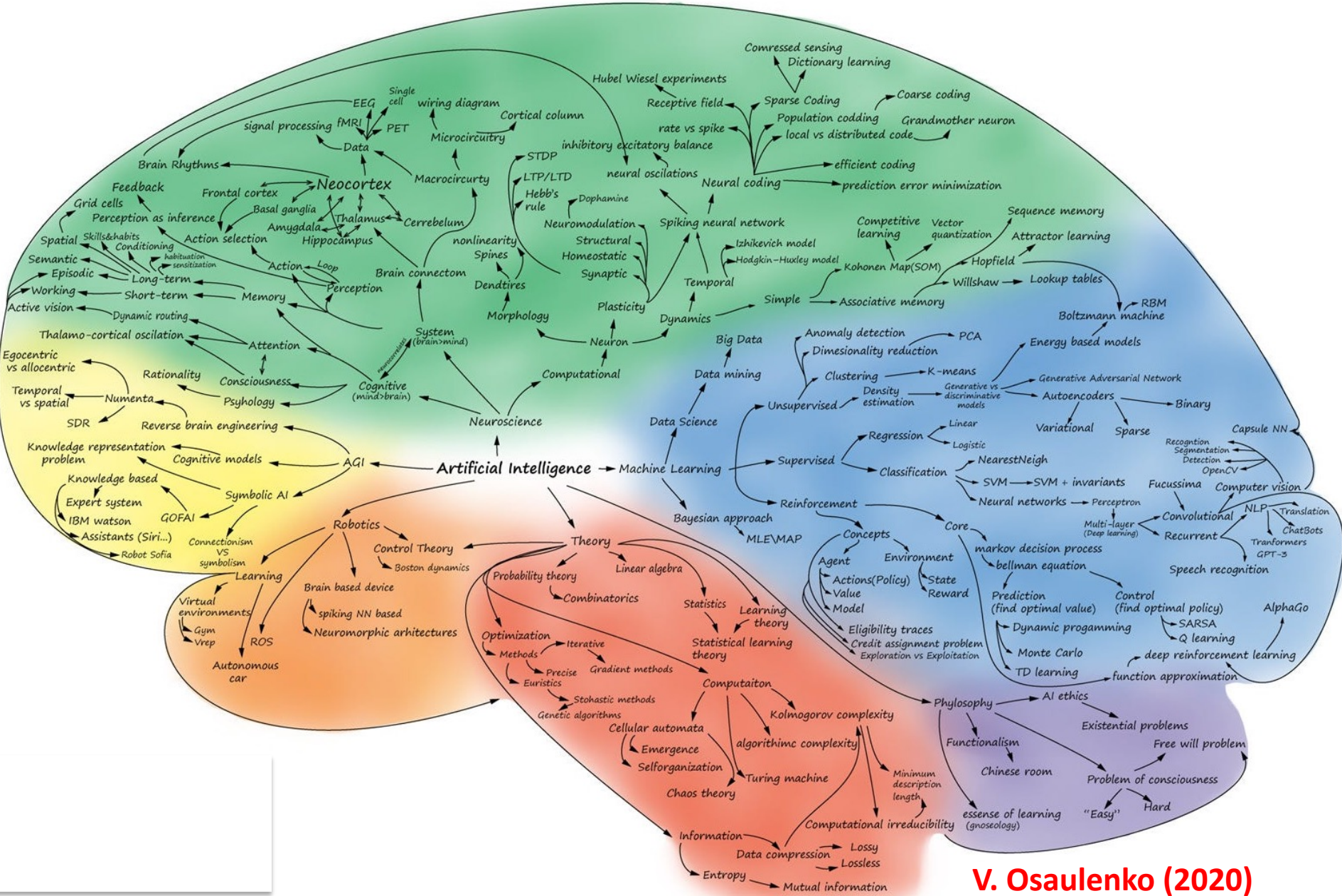


APS March Meeting: Navigating the AI revolution: Future-proofing your science career

Any session that starts with “How to graduate before your advisor retires”

PHENO: mini-review “Deep Learning in HEP” by Sergei Gleyzer on Tuesday 2 pm.

# The AI Map

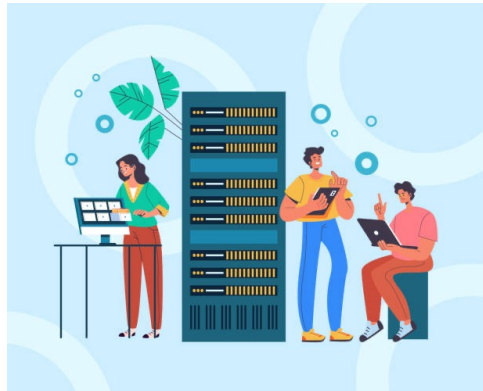


V. Osaulenko (2020)

# Data Science in STEM

<http://drewconway.com/>

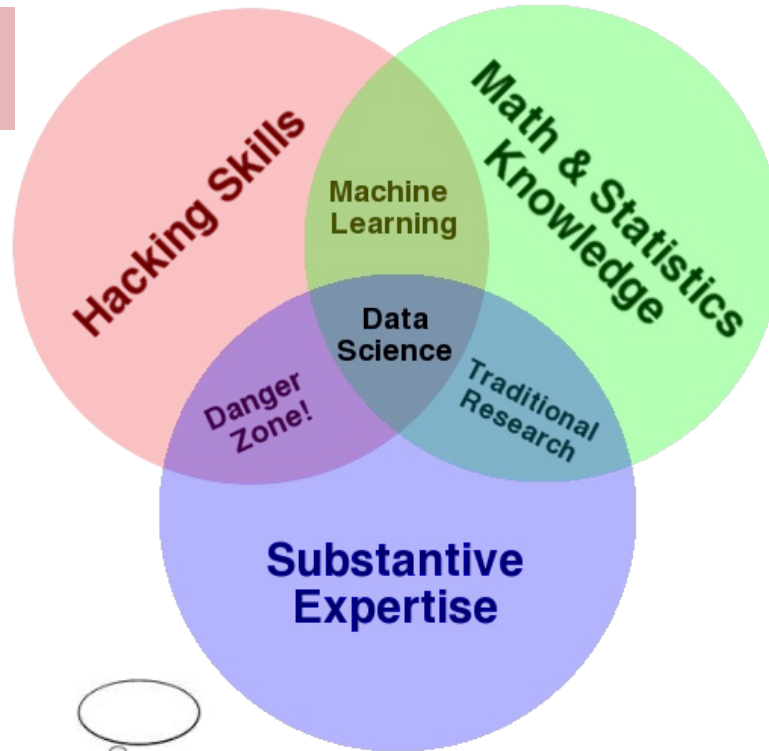
Automation



Validation

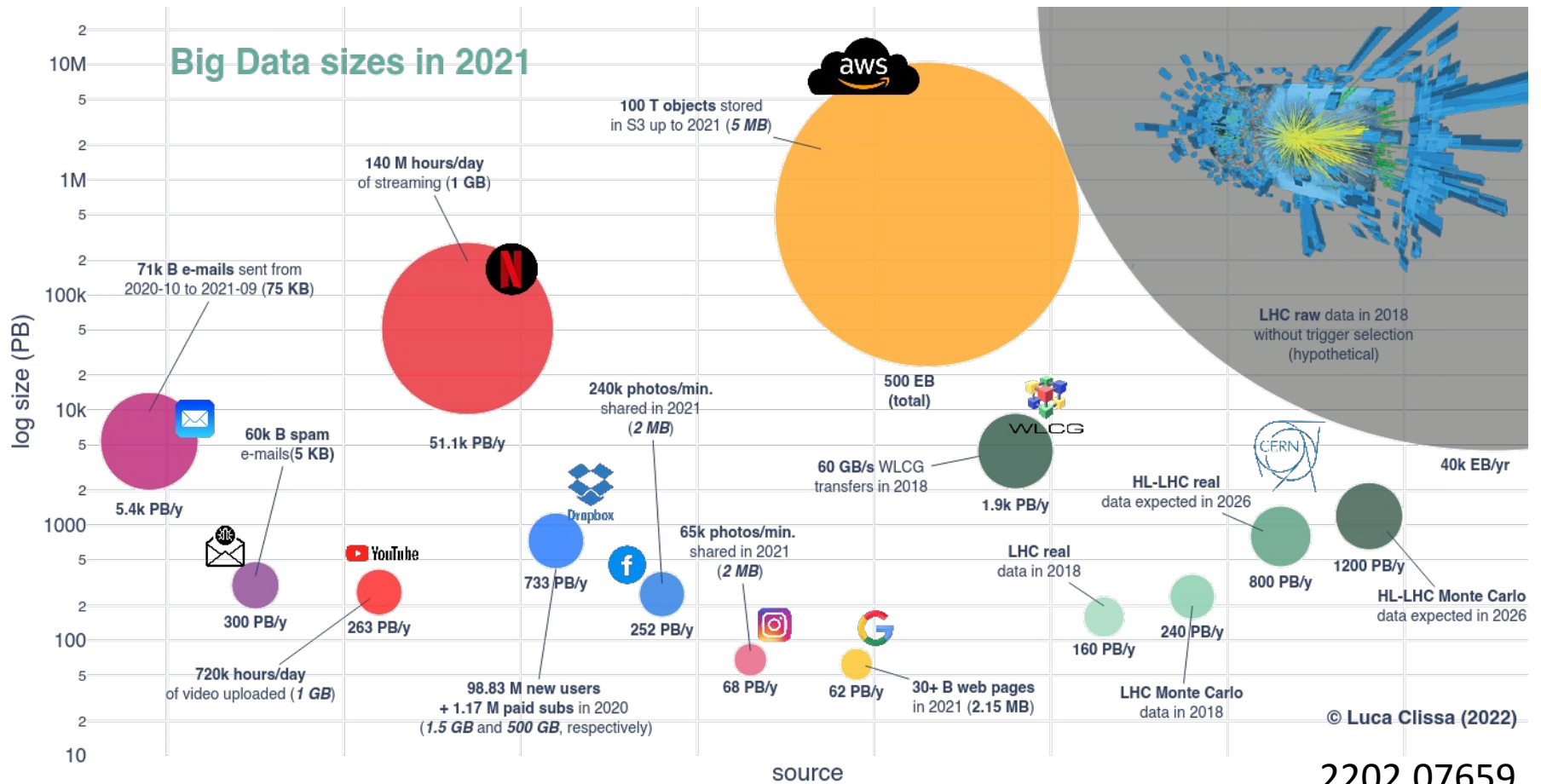


designed by freepik



Intuition

# High Energy Physics has (Lots of) Data



2202.07659

# AI in HEP theory and experiment



Building an AI-native Research Ecosystem  
for Experimental Particle Physics:  
A Community Vision

February 20, 2026 - Version 1.00

U.S. DEPARTMENT of ENERGY

About ▾ Our Mission ▾ Topics ▾ Funding Opportunities ▾ Consumer Savings ▾ Services & Opportunities ▾

Office of Science > Department of Energy Seeks Input on Advancing AI for Science and Engineering Workforce Development and Genesis Mission Challenges

## Department of Energy Seeks Input on Advancing AI for Science and Engineering Workforce Development and Genesis Mission Challenges



A National Research Collaboration in Particle Physics  
as a Strategic Platform for AI Workforce Development

*A response to the RFI on Mobilizing Talent for the Genesis Mission and Developing an American Workforce to Advance Artificial Intelligence (AI) for Science and Engineering  
Submitted on behalf of the authors of arXiv 2602.17582.*

## AI-Accelerated Discoveries Through Particle Theory\*

George T. Fleming<sup>1</sup>, Daniel C. Hackett<sup>1</sup>, James Halverson<sup>2</sup>, Timothy Hobbs<sup>3</sup>, Stefan Hoeche<sup>1</sup>,  
Konstantin T. Matchev<sup>4†</sup>, Stephen Mrenna<sup>1</sup>, Benjamin Nachman<sup>5</sup>, Prasanth Shyamsundar<sup>1</sup>,  
David Shih<sup>6</sup>, Jesse Thaler<sup>7</sup>, Jure Zupan<sup>8</sup>

<sup>1</sup> Fermi National Accelerator Laboratory, <sup>2</sup> Northeastern University, <sup>3</sup> Argonne National Laboratory,  
<sup>4</sup> University of Alabama, <sup>5</sup> SLAC National Laboratory, <sup>6</sup> Rutgers University,  
<sup>7</sup> Massachusetts Institute of Technology, <sup>8</sup> University of Cincinnati

March 4, 2026

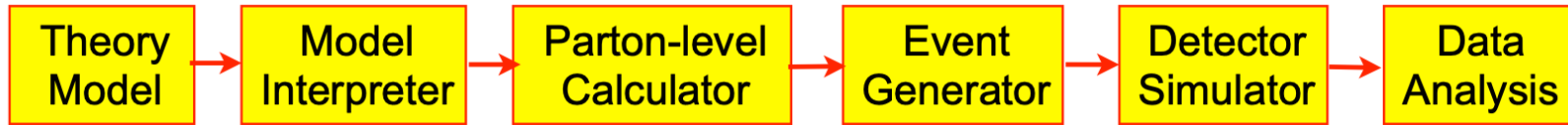
Extended version for the arXiv  
coming soon...

GENESIS MISSION

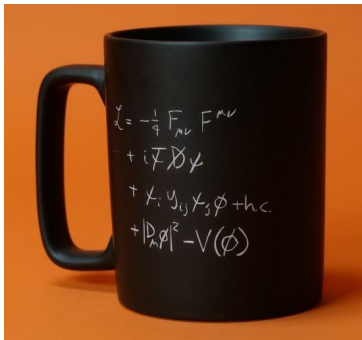
Blueprint workshop - Towards a National-Scale  
AI Collaboration in HEP, May 18-19, DC

timeline

# The simulation pipeline in HEP

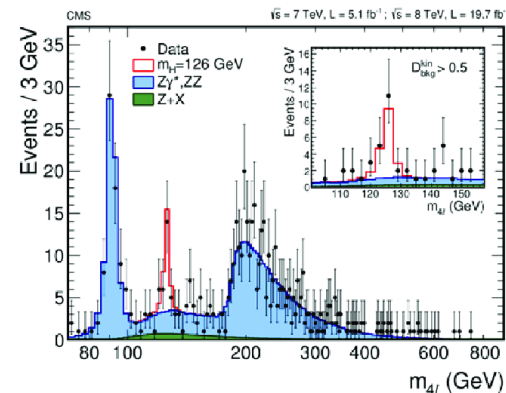


- Symmetries
- Representations
- Fields (basis)
- Lagrangian
- Aesthetics

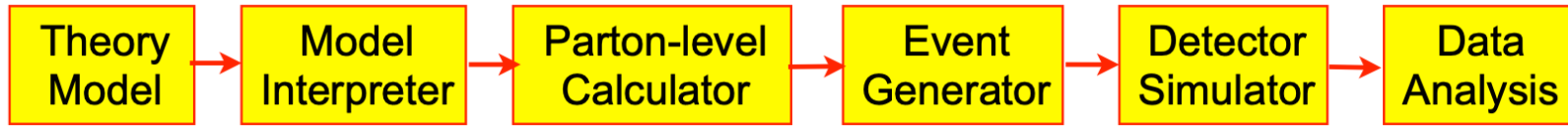


See talk by  
Federico Buccioni

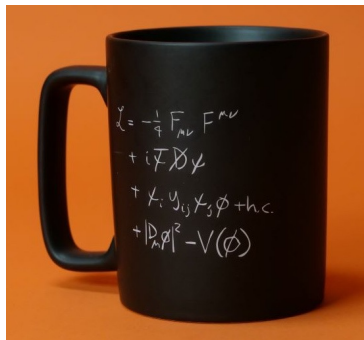
- Resolution
- RecoObjects
- Histograms
- Backgrounds
- Selection cuts
- ROC curve
- Error analysis



# The simulation pipeline in HEP



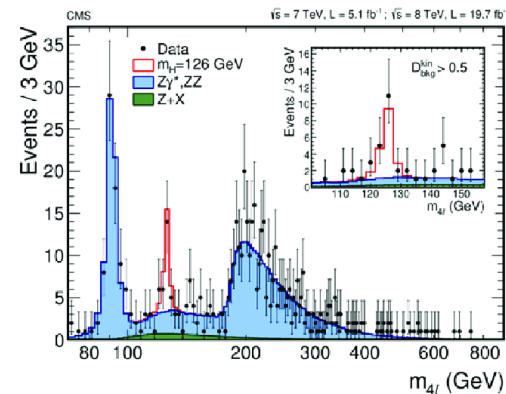
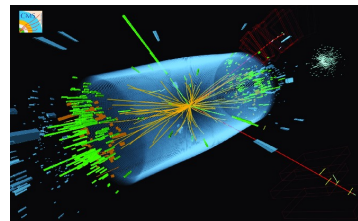
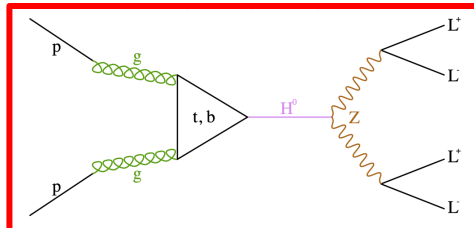
- Symmetries
- Representations
- Fields (basis)
- Lagrangian
- Aesthetics



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

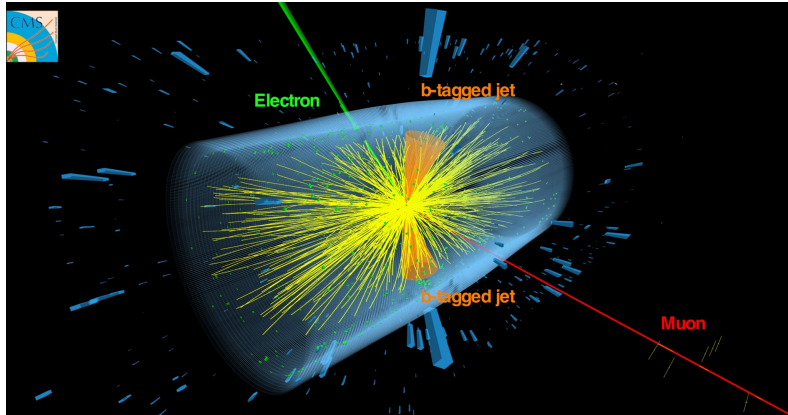
- Resolution
- RecoObjects
- Histograms
- Backgrounds
- Selection cuts
- ROC curve
- Error analysis

See talk by  
Federico Buccioni

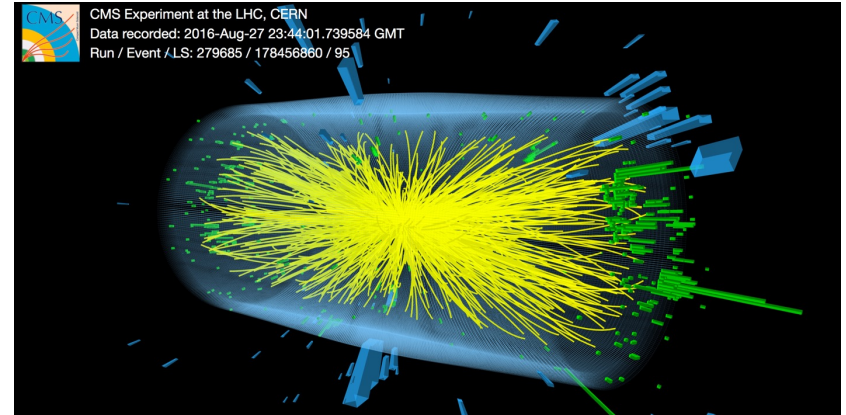


# Typical LHC Events

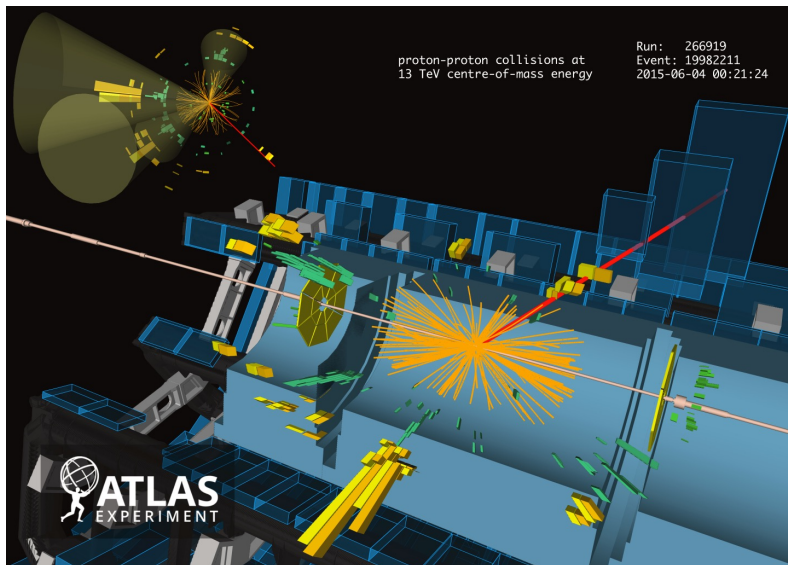
## Two top quarks (dilepton channel)



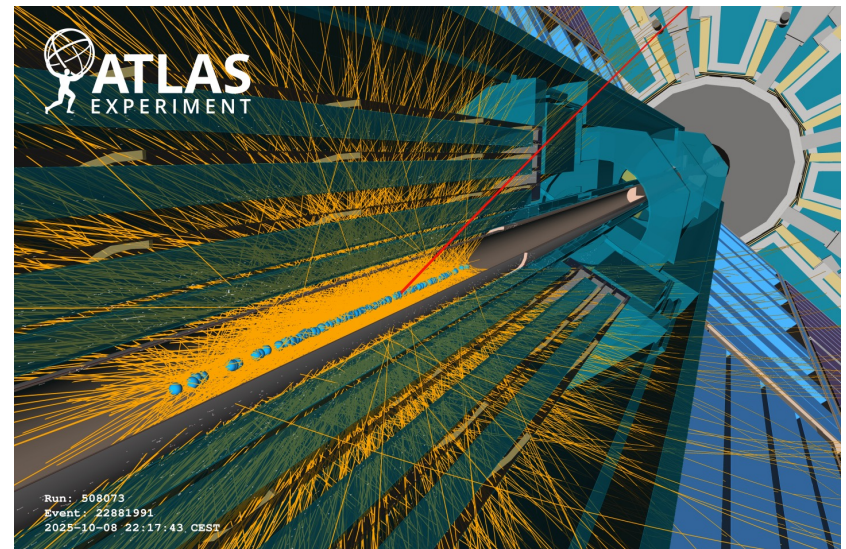
## High pile-up event



## Two top quarks (single lepton channel)

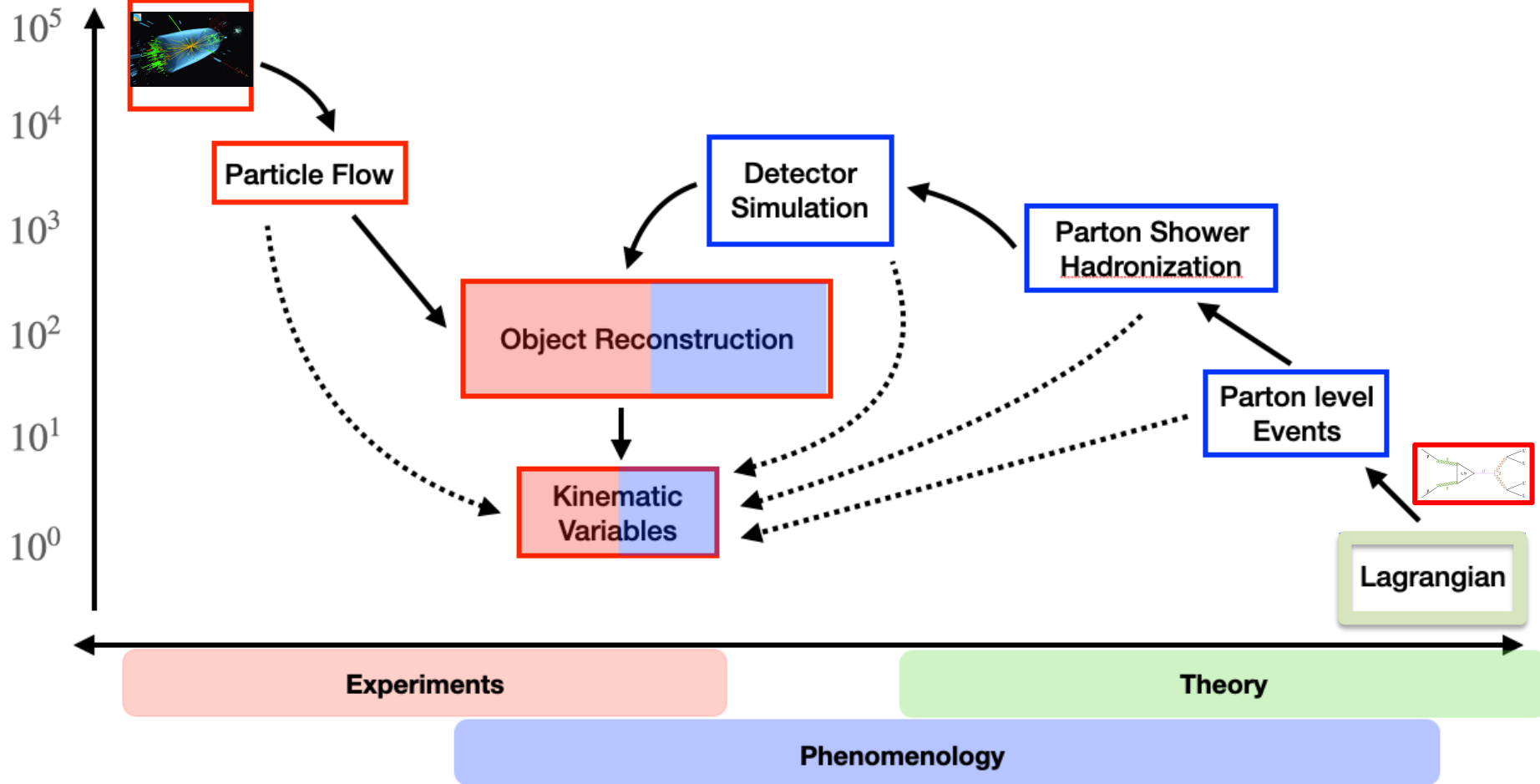


## High pile-up event



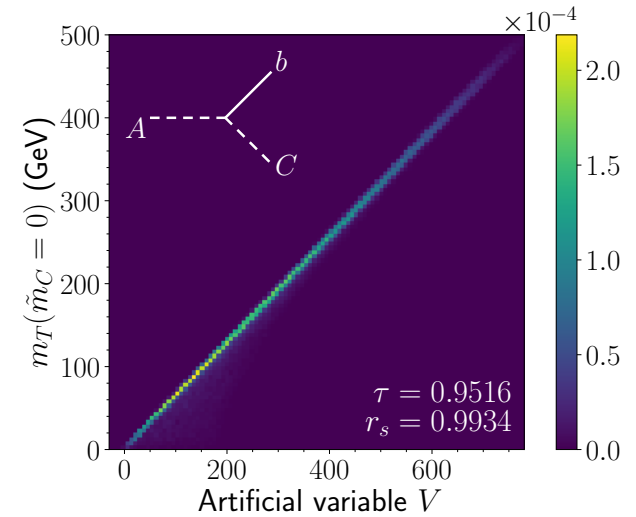
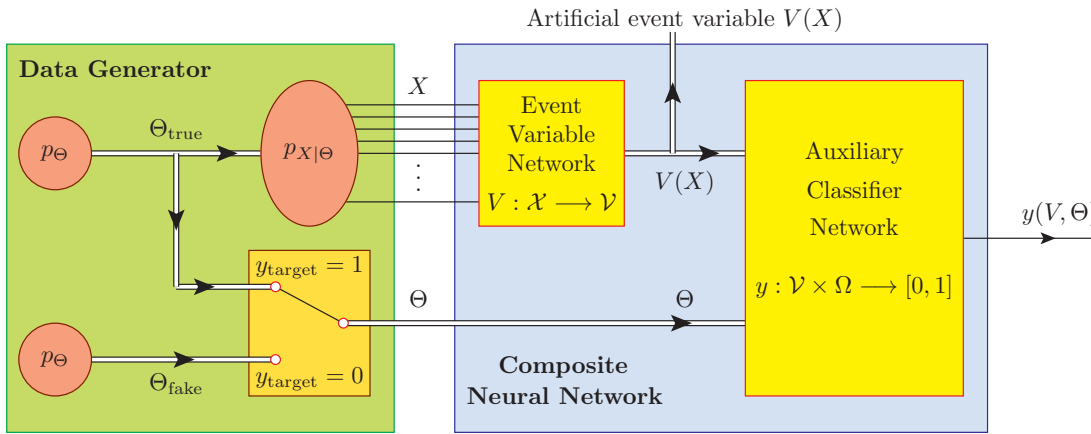
# The curse of dimensionality

Dimensionality per event



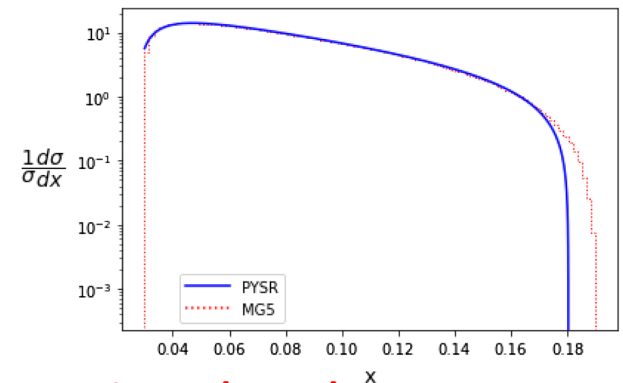
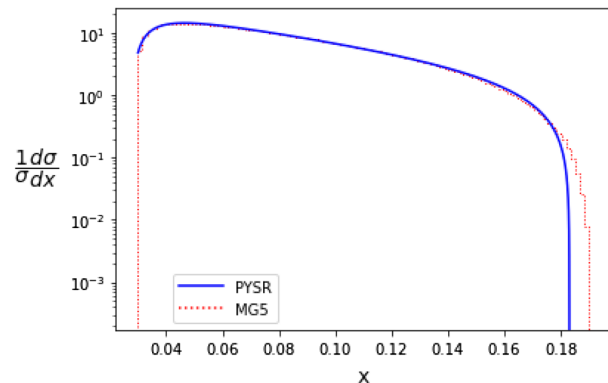
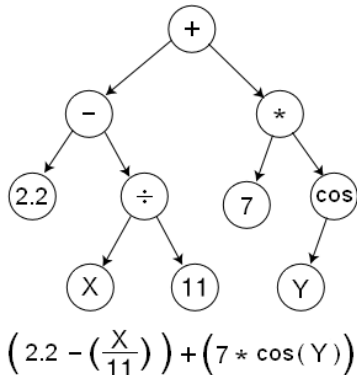
# AI applications in particle theory

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



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

- Let's ask the machine to derive a good symbolic formula



Dong, Kong, KM, Matcheva (2022)

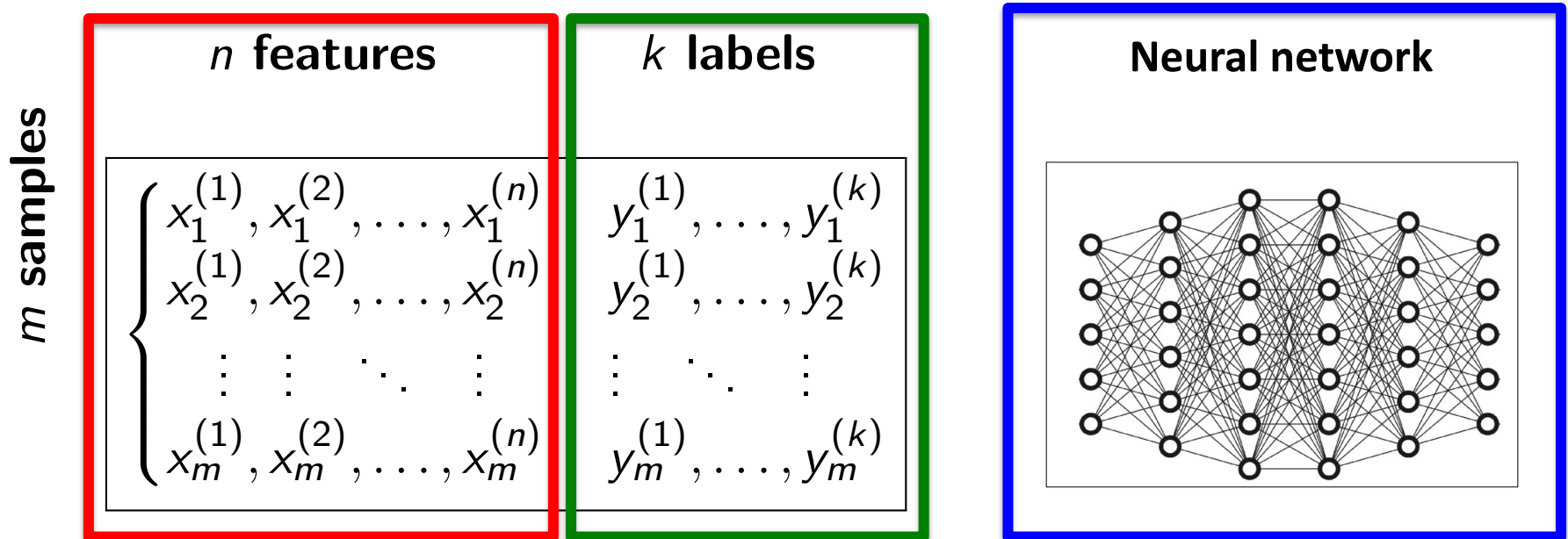
# Finding symmetries with AI

Forestano, KM, Matcheva, Roman, Unlu, Verner (2023)

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



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



This part (the data) will be given to us

We will be building this



# Contributions to the Loss Function

Invariance

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

Normalization

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

Sparsity

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

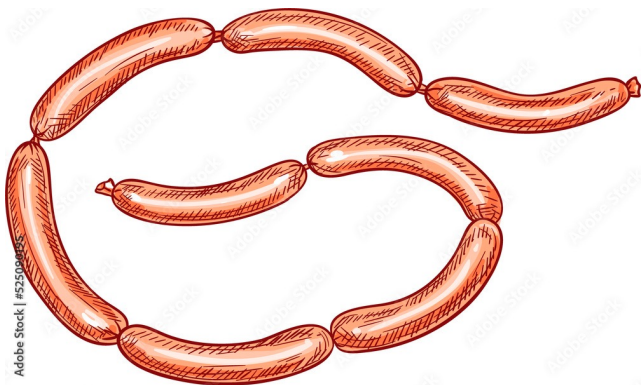
Orthogonality

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

Closure

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

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

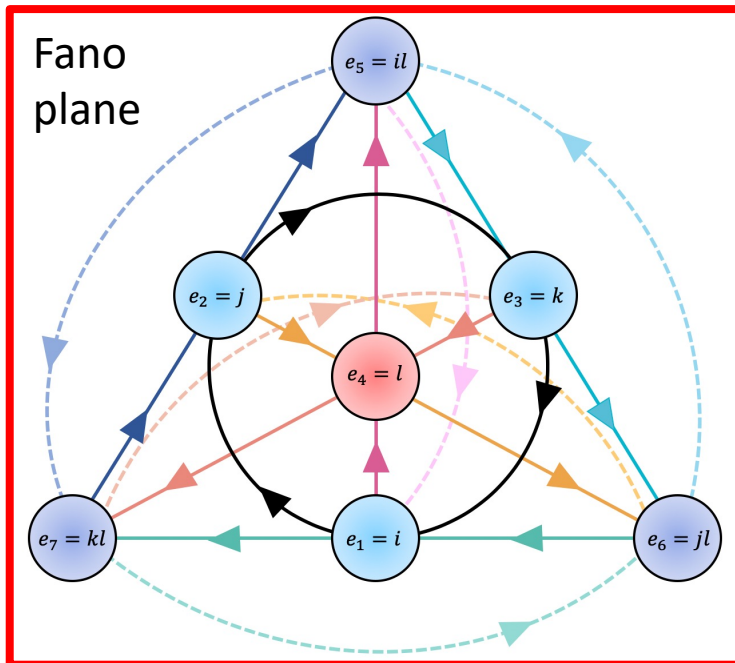


# Refresher on Octonions

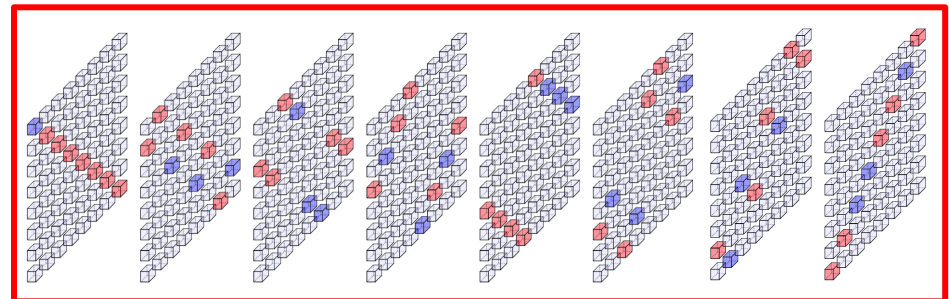
- Generalizations of the complex numbers
  - 7 imaginary units:  $\{i, j, k, l, il, jl, kl\}$ .
- Inherit the familiar operations:
  - multiplication, conjugation, norm, inverse
- Related to exceptional structures in mathematics
  - E.g.: Jordan algebra  $h_3 \rightarrow F_4$  and  $E_6$  exceptional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



imagine.art



$$h_3 = \begin{pmatrix} r_1 & \mathbf{0}_1 & \mathbf{0}_2 \\ \mathbf{0}_1^* & r_2 & \mathbf{0}_3 \\ \mathbf{0}_2^* & \mathbf{0}_3^* & r_3 \end{pmatrix}$$



Real component of a triple octonion product

# The $E_6$ group

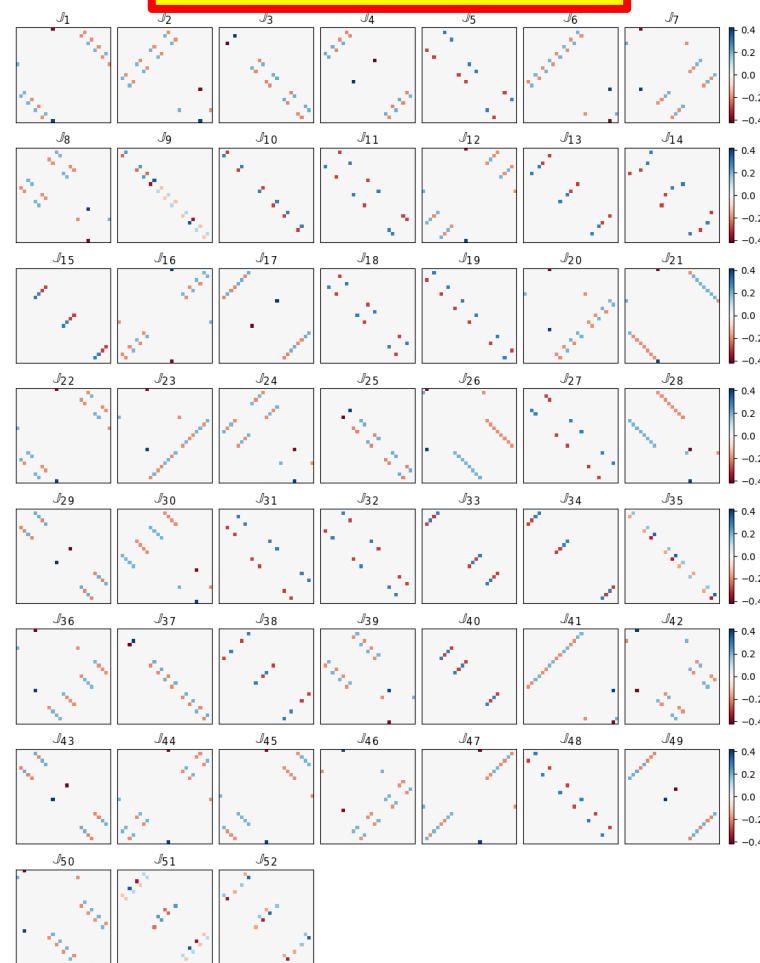
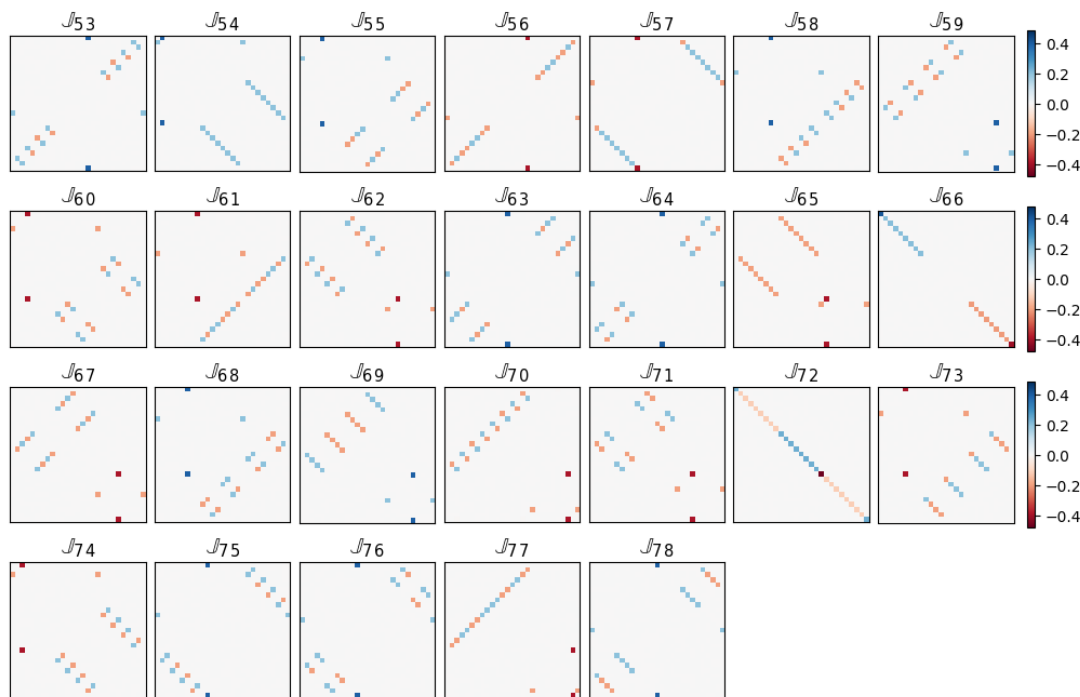
Forestano, KM, Matcheva, Roman, Unlu, Verner (2023)

- $E_6$  has 78 generators
  - 27x27 matrices
- $F_4$  subgroup (52 generators)

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

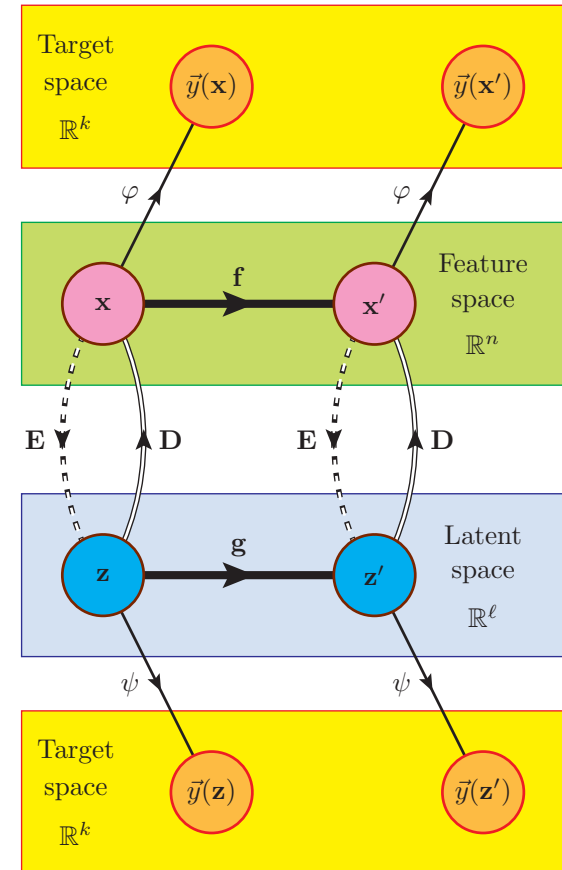
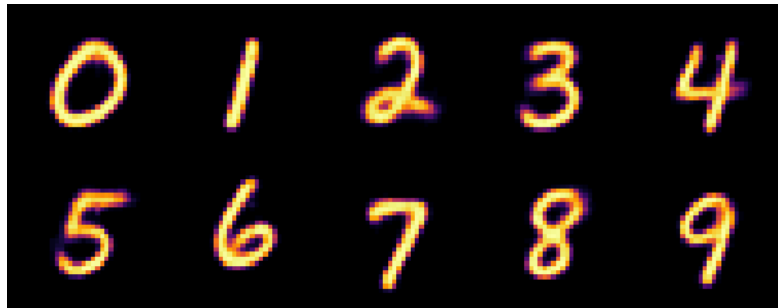
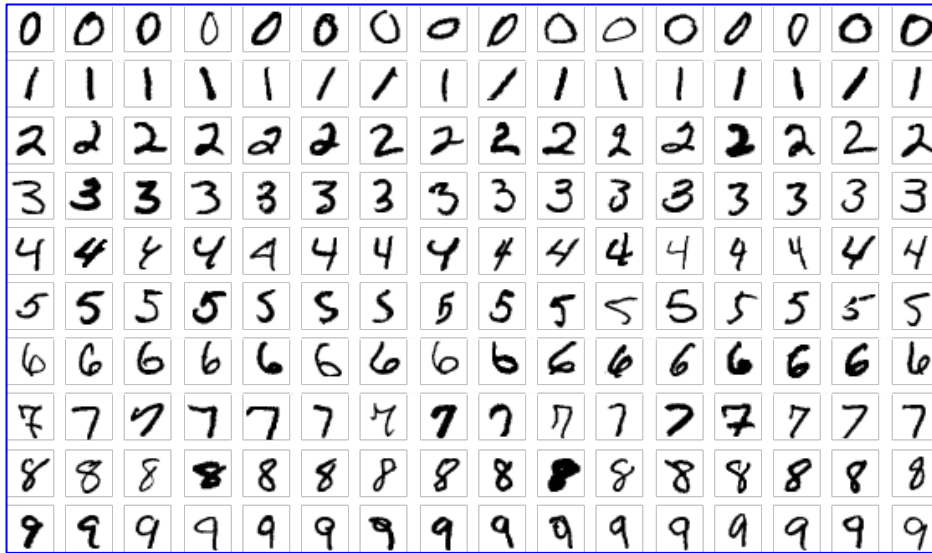
$$\varphi_{F_4}^{(1)}(\mathbf{x}) = \operatorname{Tr} \mathfrak{h}_3 = \sum_{a=1}^3 r_a$$

$$\varphi_{F_4}^{(2)}(\mathbf{x}) = \operatorname{Tr} \mathfrak{h}_3^2 = \sum_{a=1}^3 (r_a^2 + 2 |\mathbf{o}_a|^2)$$



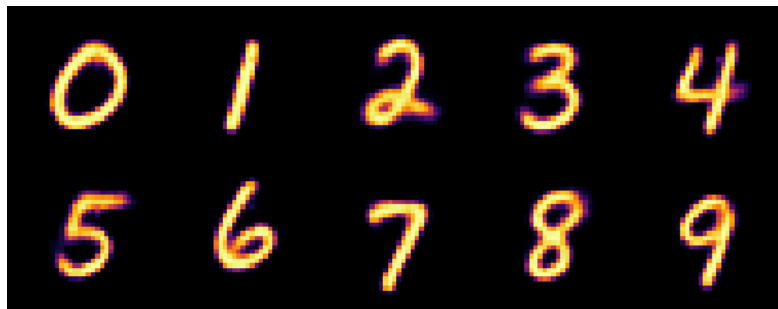
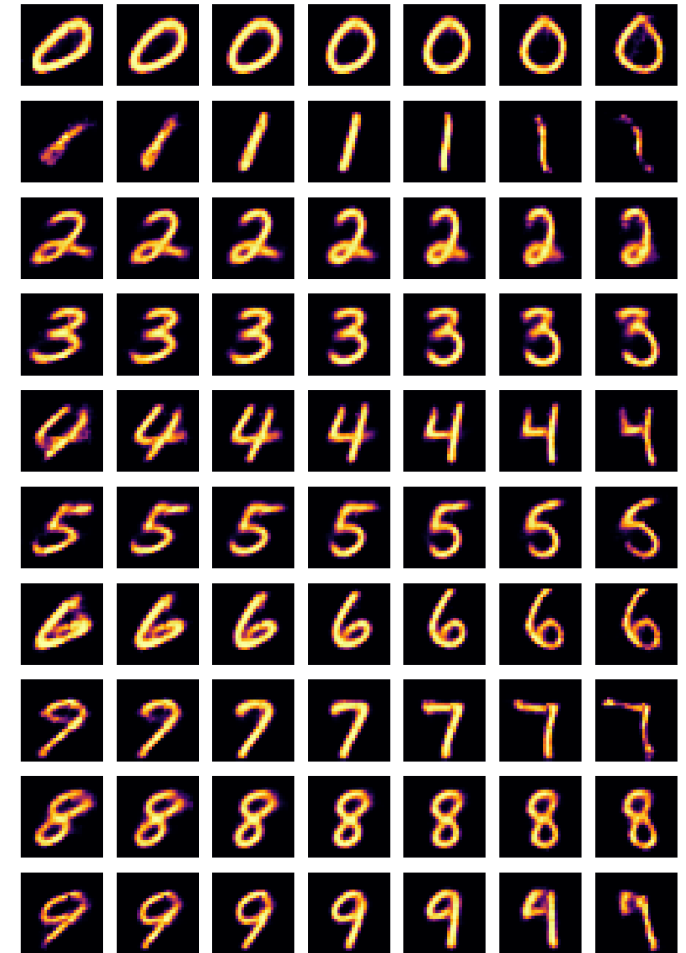
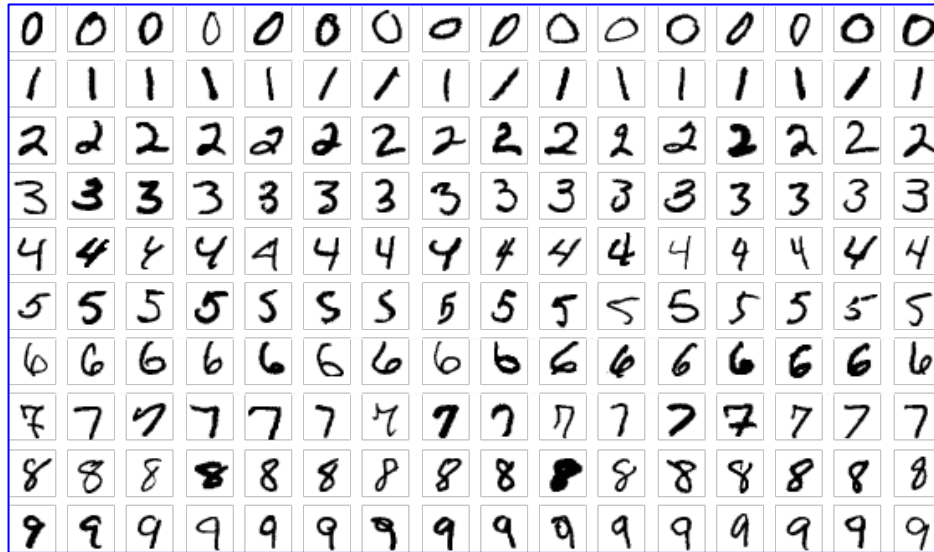
# AI applications in data science

- Let's ask the machine to find symmetries in real data



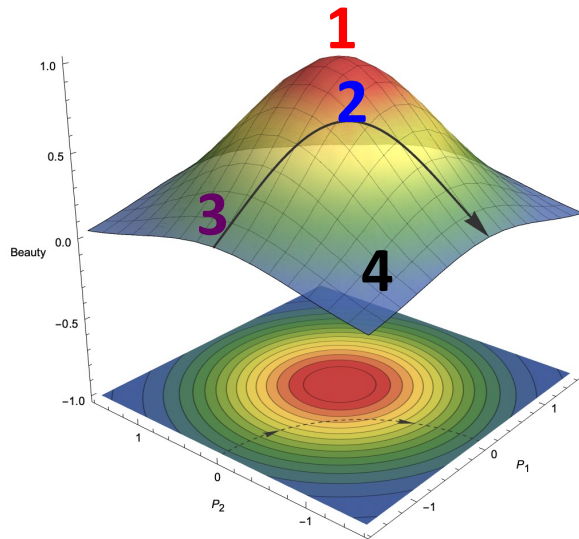
# AI applications in data science

- Let's ask the machine to find symmetries in real data



# AI applications in particle theory

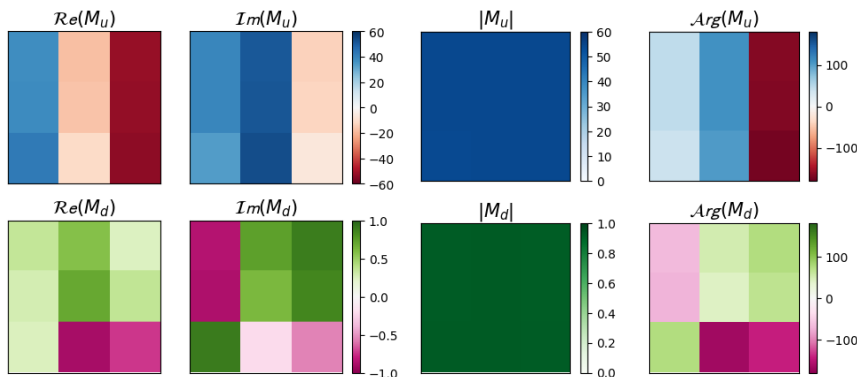
- Let's ask the machine for a **true** and **beautiful** model



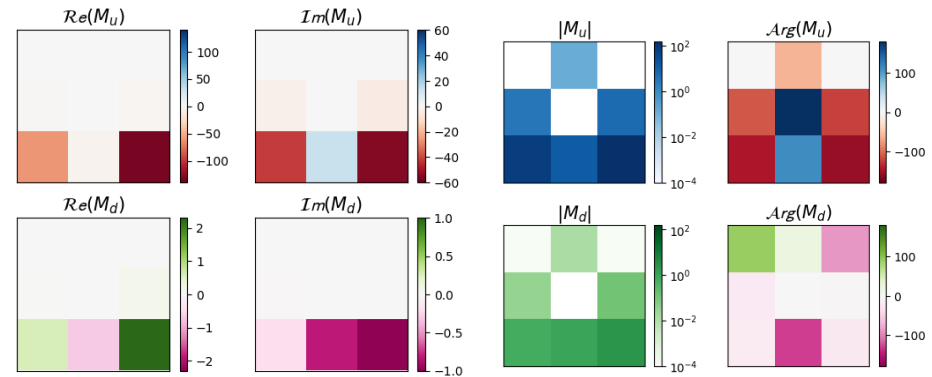
- Examples of theory models:
  - **1**: beautiful and wrong
  - **2**: beautiful and true
  - **3**: ugly and true
  - **4**: ugly and wrong

**KM, Matcheva, Ramond, Verner (2023)**

Beauty = Uniformity

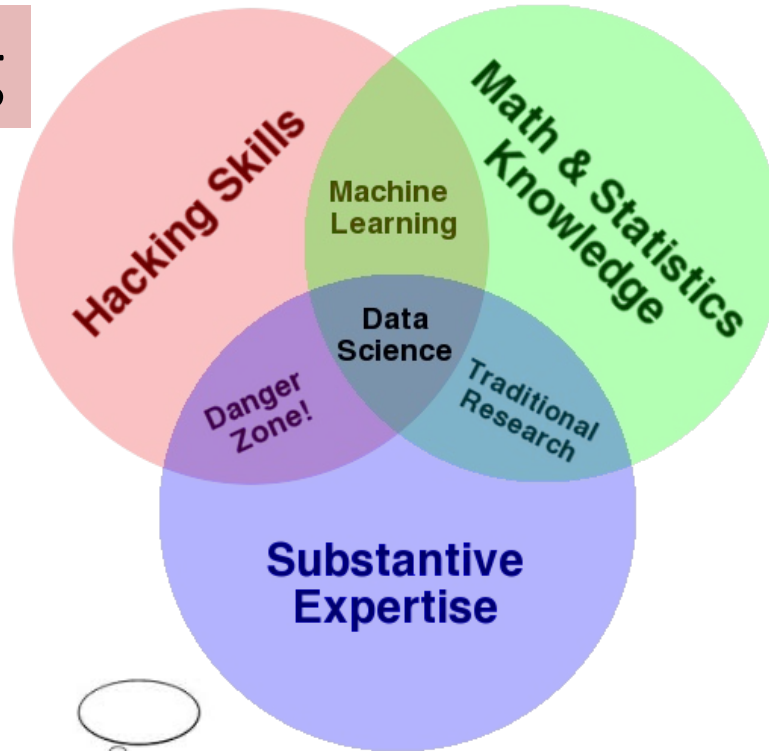
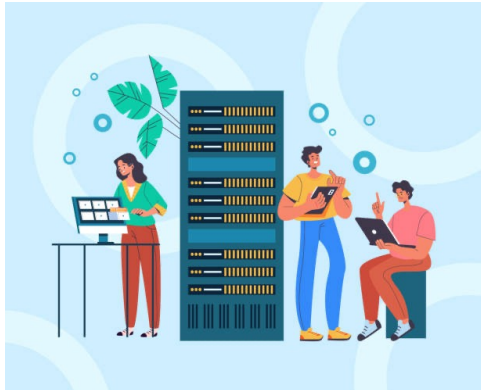


Beauty = Sparsity



# The Tasks of a High-Energy Theorist

Programming



Calculating



designed by freepik

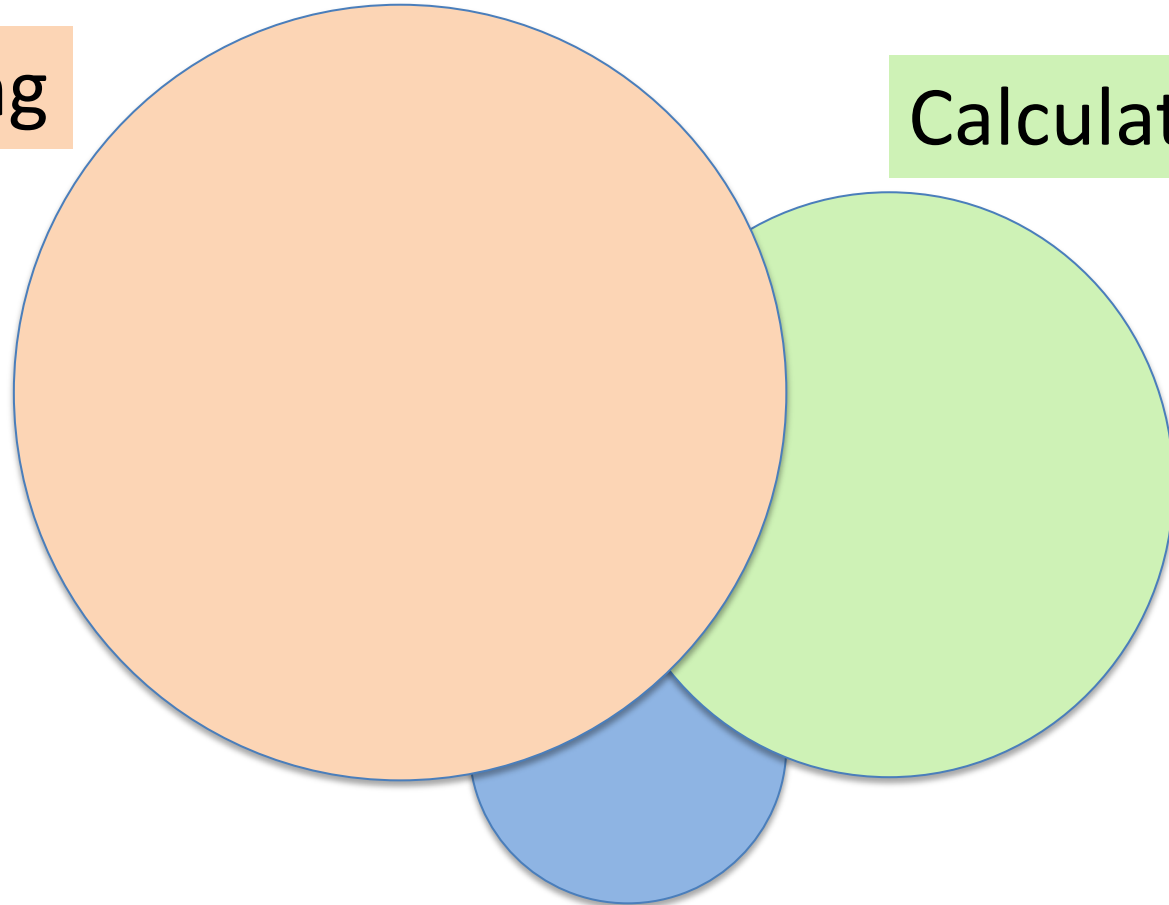


Model Building

# Time Spent by a High-Energy Theorist

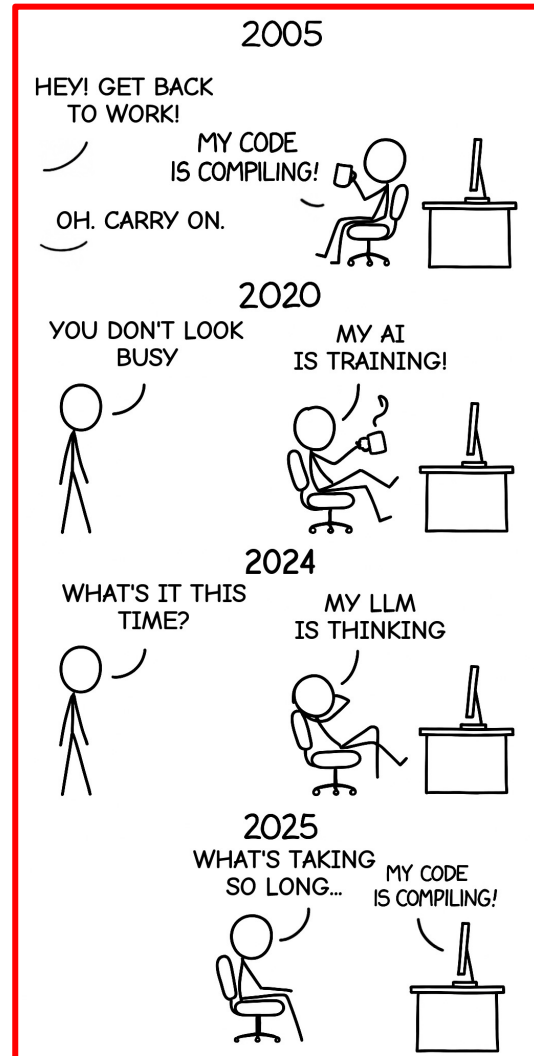
Programming

Calculating



Model Building

# Enter LLMs



# Agentic programming

See talk by Tony Menzo in AI session on Tuesday 2pm

- Modern LLMs are
  - Pretty **reliable** compressions of human knowledge
  - Becoming very **powerful**
  - **Evolving** rapidly
- Agent: Given **context**  $C_t$  at a decision step  $t$ , and an **action**  $A_t$  selected from an action space, an agent is a system that implements a **conditional distribution** over actions,  $p(A_t | C_t)$  and is embedded in a **feedback loop** such that executed actions influence future context.

## Traditional programming

- Specify how to do something
- Scripting
- Control flow is explicit and tractable.

## Agentic programming

- Specify what to do
- Prompt engineering
- Control flow is non-deterministic

<https://github.com/tonymenzo/heptapod>

# HEPTAPOD

HEP Toolkit for Agentic Programming, Orchestration, and Deployment

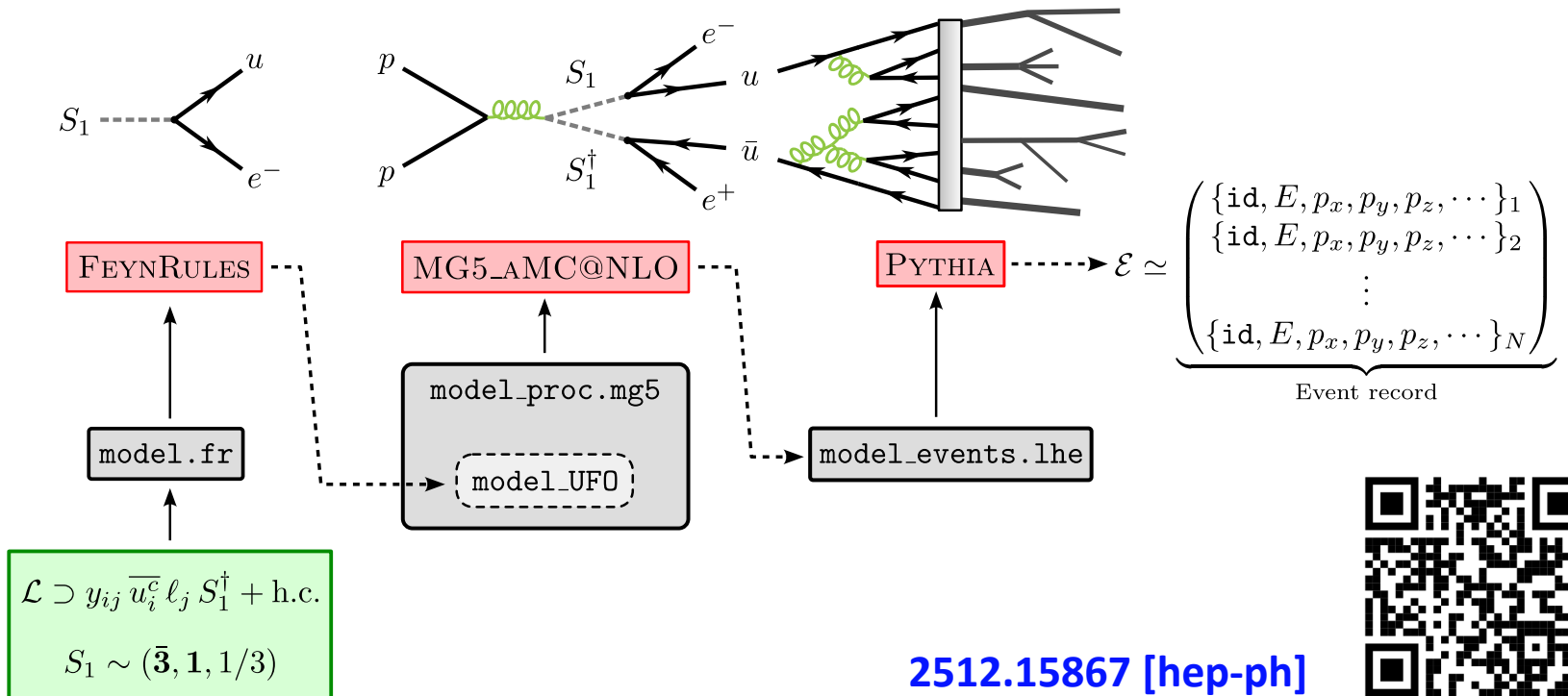
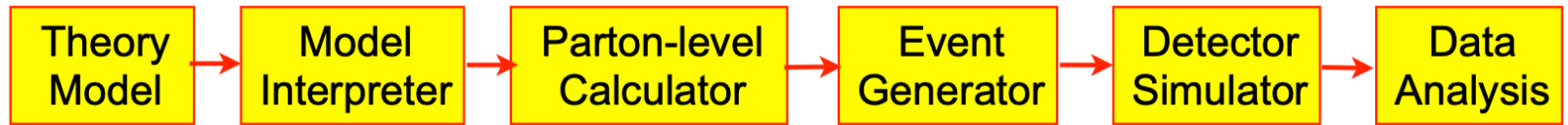
**Emphasis on toolkit: 3+ harnesses, 60+ tools, prompts, skills**

#	Tool	Reliability	Auditable	Object	External	Repeated	Diagrammatica / exact symbolic tools							
<b>Event generation and simulation</b>							20	ComputeSymbolicAmplitudeTool	*	✓	✓		✓	
1	FeynRulesToUFOTool	✓		✓	*	✓	21	RunWolframTool	✓			*	✓	
2	MadGraphFromRunCardTool	✓	✓	✓	*	✓	22	RunWolframScriptBatch	✓			*	✓	
3	PythiaFromRunCardTool	✓		✓	*	✓	23	SimplifyResultTool	*	✓	✓		✓	
4	SherpaFromRunCardTool	✓		✓	*	✓	24	ConvertToPythonTool	✓	✓	✓		*	
5	JetClusterSlowJetTool	✓	✓	*	✓	✓	<b>NDA and FeynGraph</b>							
<b>Event-format conversion and array bridges</b>							25	EstimateDecayWidthNDATool		✓		*	✓	
6	LHETOJSONLTool	*	✓	✓		✓	26	EstimateBranchingRatioNDATool	✓	✓	*		✓	
7	EventJSONLToNumpyTool	✓		✓		*	27	EnumerateDiagramsTool	✓	✓	*		✓	
8	JetsJSONLToNumpyTool	✓	✓	✓		*	28	DiagramVisualizationTool		*	✓		✓	
<b>Kinematic analysis and event selection</b>							<b>PDG and INSPIRE tools</b>							
9	CalculateInvariantMassTool	*	✓	✓		✓	29	PDGDatabaseTool		✓		✓	*	✓
10	CalculateTransverseMomentumTool	✓		✓		*	30	PDGSearchTool		✓		✓	*	✓
11	CalculateDeltaRTool	*	✓	✓		✓	31	PDGPropertyTool		✓		✓	*	✓
12	ApplyCutsTool	✓	*	✓		✓	32	InspireSearchTool		✓		✓	*	✓
13	GetHardestNTool	✓	*	✓		✓	33	InspirePaperTool		✓		✓	*	✓
14	GetHardestNJetsTool	✓	*	✓		✓	34	InspireCitationTool	✓			✓	*	✓
15	FilterByPDGIDTool	✓	*	✓		✓	35	InspireBibTeXTool	✓			✓	✓	*
16	SortByPtTool	✓	*	✓		✓	36	InspireAuthorTool				✓	*	✓
17	MergeObjectCollectionsTool	✓	✓	✓	*	✓	37	InspireInstitutionTool				✓	*	✓
18	FilterByDeltaRTool	*	✓	✓		✓	38	InspireConferenceTool				✓	*	✓
<b>Resonance reconstruction</b>							39	InspireJournalTool				✓	*	✓
19	ResonanceReconstructionTool	✓	✓	*		✓	40	InspireExperimentTool				✓	*	✓
							41	InspireReadingListTool		✓		✓	✓	*
							42	InspireNotesTool		✓		✓	✓	*

# Agents to the rescue

- **HEPTAPOD** (HEP Toolkit for Agentic Planning, Orchestration & Deployment)

Menzo, Roman, KM, Gleyzer, Fleming, Hoche, Mrenna, Shyamsundar 2025



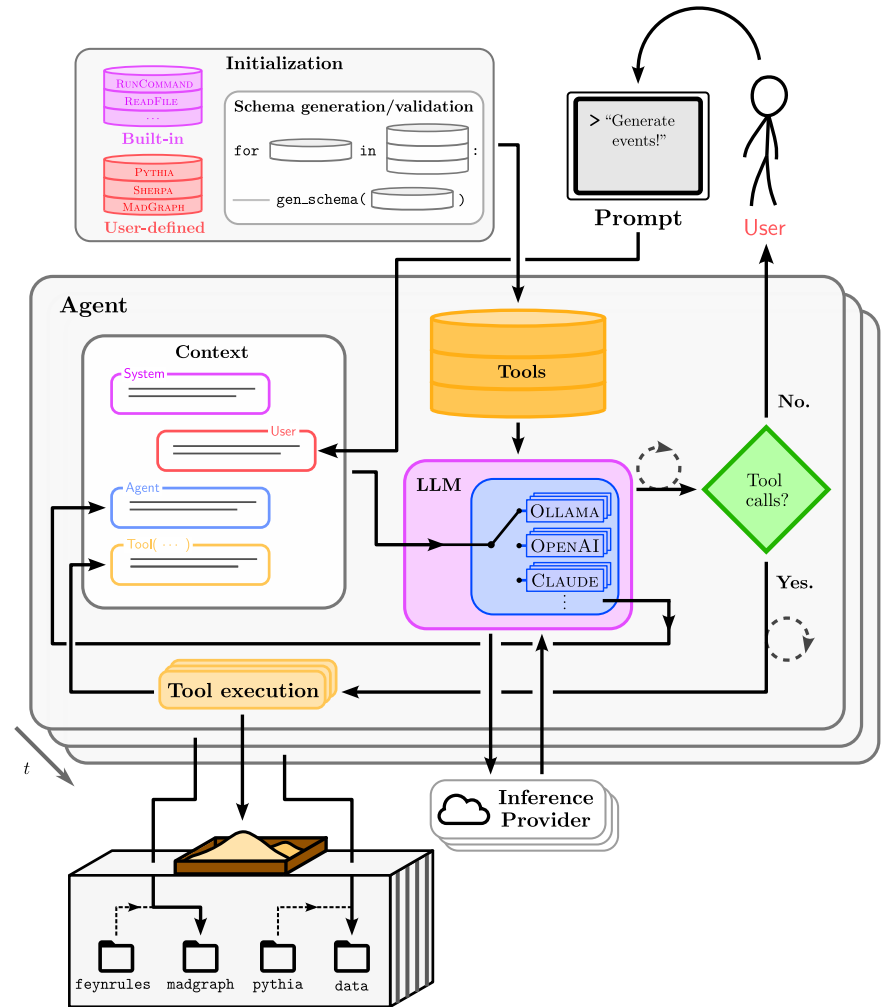
2512.15867 [hep-ph]



# How does the agent work?

HEPTAPOD: A HEP-focused toolkit for agentic programming that integrates LLMs into HEP workflows spanning theoretical calculations, simulation, and data analysis.

- Define workflows at the level of **physics intent**, not scripts
- Query particle properties, literature, and unit conversions
- Execute multi-stage pipelines using run-cards and filesystem tools
- Equipped with tools like to-do lists and terminal access
- Approval hooks preventing misbehavior
- Fully reproducible, auditable logs
- Built on the **Orchestral AI** engine [Roman&Roman 2601.02577]



# Pictures from the agent in action

## User

Plan a Monte Carlo signal-validation analysis of the BSM scalar leptoquark defined by the supplied run cards. Generate 10,000 events, select objects, and create lepton-jet invariant-mass distributions for each parameter point. Attempt to isolate and reconstruct the leptoquark resonance by applying reasonable lepton-jet separation and experimental cuts. Compute and plot the resulting  $m_{LQ}^{\min}$  distribution. Write a todo list first and then execute the todo list until completed.

```

#####
## Monte Carlo Signal-Validation Analysis for S1 Leptoquark

- [*] Generate UFO model from FeynRules file
    `feynrules/models/S1_LQ_RR.fr`
- [] Generate parton-level events with MadGraph using card
    `madgraph/cards/S1_LQ_RR_pp_l1jq_scan.mg5`
    - set `nevents = 10000`
    - keep all parameter points
- [] Shower and hadronize events with Pythia using card
    `pythia/cards/S1_LQ_RR_pp_l1lj.cmd`
    - `shower_lhe = True`
- [] Convert Pythia JSDEL events to NumPy array
- [] Apply kinematic cuts to leptons and jets
    - leptons: pt>20 GeV, ljetal<2.5, PDG IDs 11,13 (and charge
      conjugates)
    - jets: pt>30 GeV, ljetal<2.5
- [] Select hardest 2 leptons and hardest 2 jets per event
- [] Compute lepton-jet invariant masses for all pairings and retain
    the minimum per event (`m_min_LQ`)
- [] Plot the `m_min_LQ` distribution (histogram, 50 bins,
    appropriate range) and save as `mmin_LQ.png`
- [] Summarize results (number of events, cross-section, plot
    location)
    
```

```

GetHardestNJets(...)
Same as GetHardestNJets above with run_01 → run_02
    
```

```

ResonanceReconstruction(min_delta_r=0.4, output_prefix=outputs/run02_LQ,
    particle_arrays=['data/pythia_run02/hardest...',
    template=two_body_symmetric
    
```

```

Error: Processing Error
- Reason: Array 1 not found: data/pythia_run02/hardest2_jets.jsonl
    
```

```

GetHardestNJets(input_path=data/pythia_run03/jets.jsonl,
    n_hardest=2,
    output_path=data/pythia_run03/hardest2...
    
```

```

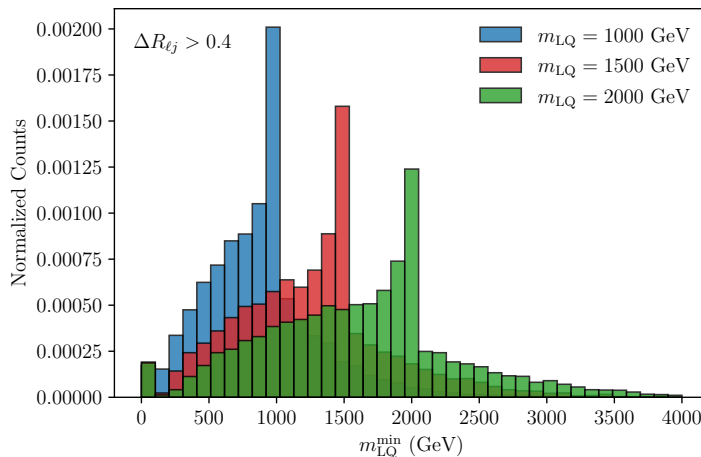
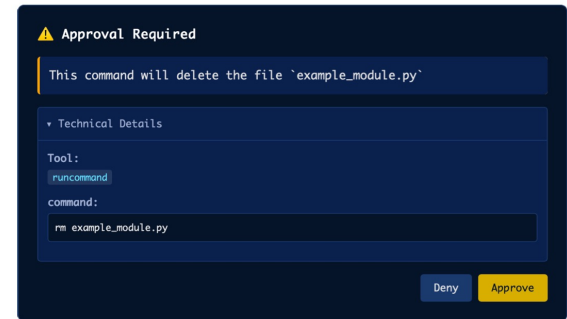
Error: File Not Found
- Reason: Input file not found: data/pythia_run03/jets.jsonl
    
```

```

JetClusterSlowJet(...)
Same as above with run_02 → run_03
    
```


```

GetHardestN(...)
Same as GetHardestN above with run_02 → run_03
    
```



Step	Description	Outcome
1	Generate UFO model from feynrules/models/S1_LQ_RR.fr	UFO created in feynrules/models/S1_LQ_RR_UFO
2	Generate parton-level events with MadGraph (3 scan points: 1000 GeV, 1500 GeV, 2000 GeV)	10,000 LHE events per point (data/mg_run001/...)
3	Shower & hadronize with Pythia8 (pythia/cards/S1_LQ_RR_pp_l1lj.cmd)	10,000 final-state events per point (data/pythia_run0X/events.jsonl)
4	Jet clustering (anti- $k_t$ , $R = 0.4$ )	Jets stored in data/pythia_run0X/jets.jsonl
5	Lepton selection (electrons $\pm 11$ , muons $\pm 13$ )	leptons_all.jsonl for each run
6	Keep hardest 2 leptons & hardest 2 jets per event	hardest2_leptons.jsonl, hardest2_jets.jsonl
7	Reconstruct leptoquark candidates (two-body symmetric template, $\Delta R > 0.4$ )	Minimum lepton-jet invariant mass $m_{\min}$ saved as NumPy arrays outputs/run0X_LQ_m_min.npy
8	Plot the distribution of $m_{LQ}^{\min}$ for the three mass points	Histogram saved as mmin_LQ.png
9	Summarize results	See below

# Summary and Outlook

- AI (LLMs) is now ubiquitous in science and in everyday life.
- There is still (lots of) room for narrow AI applications in HEP
- Where do we go next? Three questions:
  - Can we teach AI routine research tasks? **Community tool repository?**
  - Can we teach AI high-level thinking? What is the  for HEP theory?
  - How do we integrate AI and quantum? **Knipfer,Roman,Gleyzer,KM,Matcheva (2026)**
- Mark your calendars:

**ML4Jets2028 @ The University of Alabama**



**BACKUP SLIDES**

# 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  $3 \times 3$  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.



# 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)

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

SO(1,3)

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

Squeeze map

$$\varphi(\mathbf{x}) = x^{(1)} x^{(2)}$$

G<sub>2</sub>

$$\varphi_{G_2}^{(1)}(\mathbf{x}) = \sum_{i=1}^7 (x^{(i)})^2$$



$$\varphi_{G_2}^{(2)}(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) = \sum_{i,j,k=1}^7 \mathcal{D}_{ijk} x_1^{(i)} x_2^{(j)} x_3^{(k)}$$

F<sub>4</sub>

$$\varphi_{F_4}^{(1)}(\mathbf{x}) = \text{Tr } \mathfrak{h}_3 = \sum_{a=1}^3 r_a$$



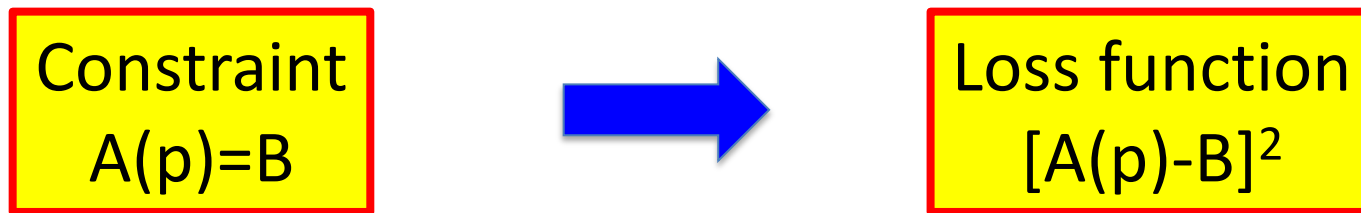
$$\varphi_{F_4}^{(2)}(\mathbf{x}) = \text{Tr } \mathfrak{h}_3^2 = \sum_{a=1}^3 (r_a^2 + 2|\mathbf{o}_a|^2)$$



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

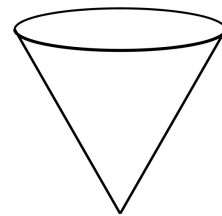
# Loss Functions

- **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

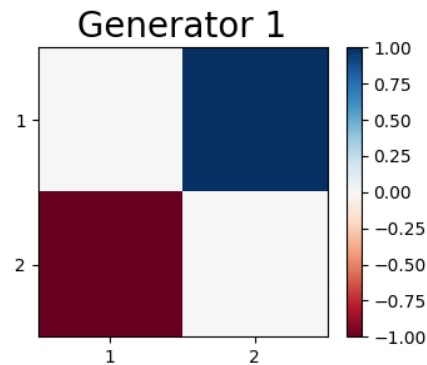


- Our loss function will contain several contributions, one for each desired symmetry property
  - **Invariance** (the transformation preserves the oracle)
  - **Normalization** (the transformation is non-trivial)
  - **Orthogonality** (the found symmetries are different from each other)
  - **Closure**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

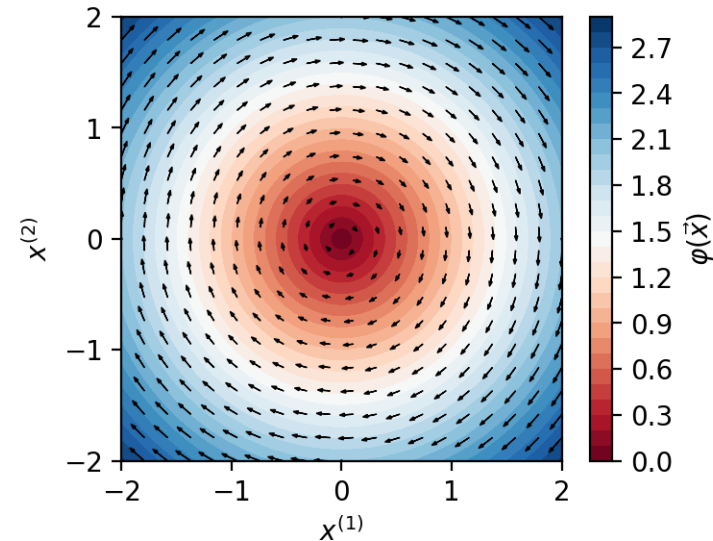
# First Toy Example: 2D Rotations



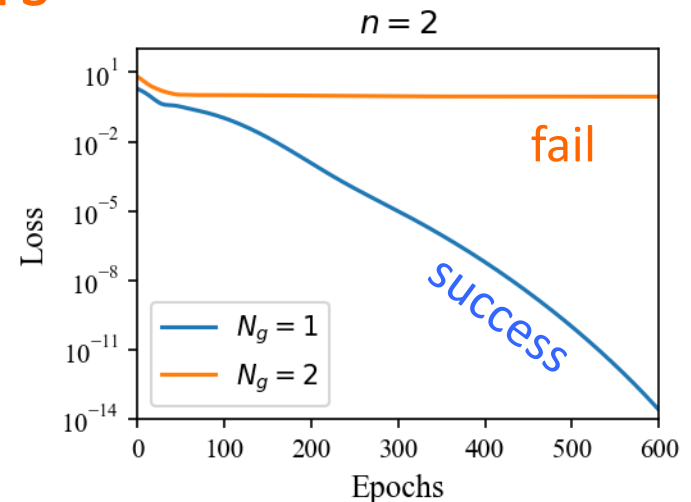
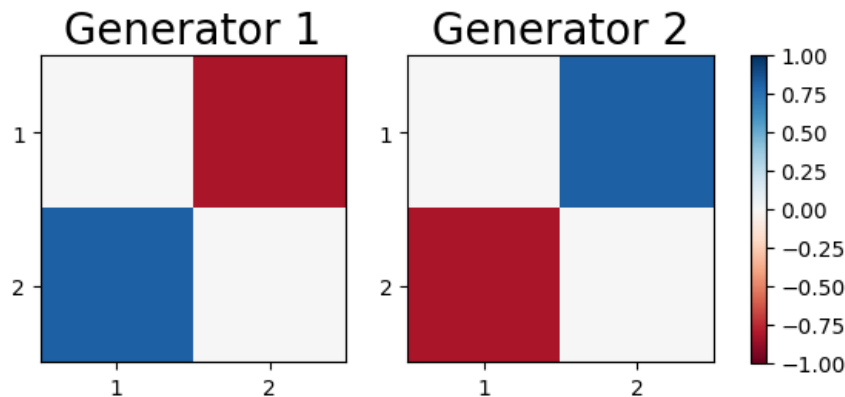
- Oracle: L2 norm of a vector
- $n=2$  dimensions;  $N_g=1$  generator



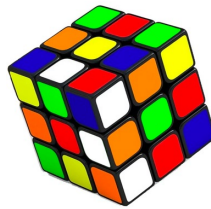
Orthogonal matrices  
are anti-symmetric!



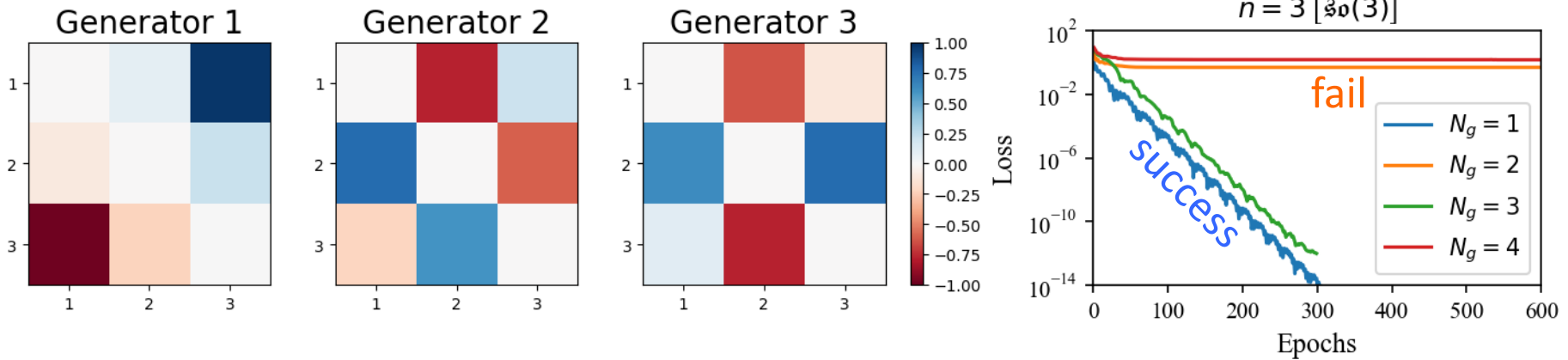
- $n=2$  dimensions;  $N_g=2$  generators



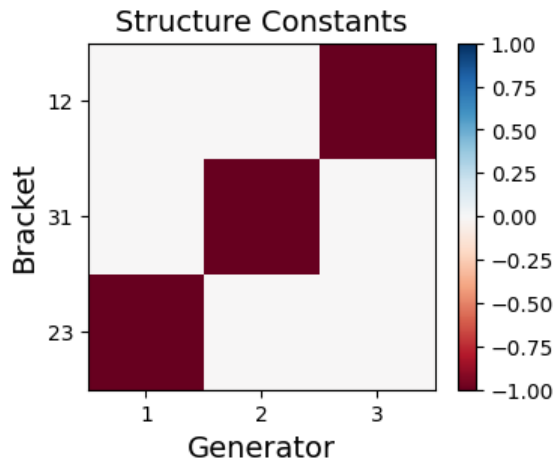
# Rotations in 3D



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

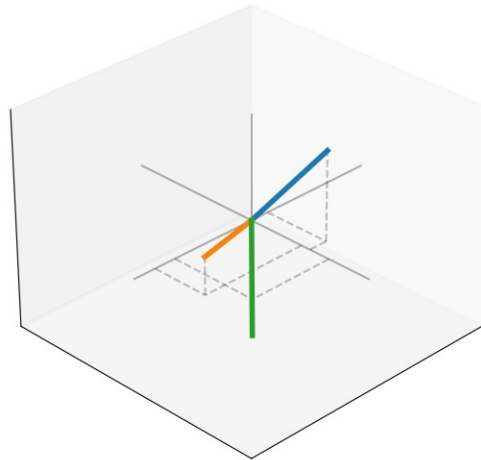


$$[\mathbf{J}_\alpha, \mathbf{J}_\beta] = \sum_{\gamma=1}^{N_g} a_{[\alpha\beta]\gamma} \mathbf{J}_\gamma$$



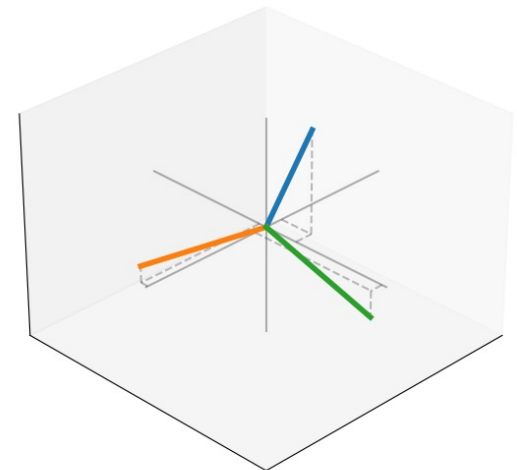
Beginning of training

Epoch: 0 | Angles = 66.16°, 92.56°, 44.16°

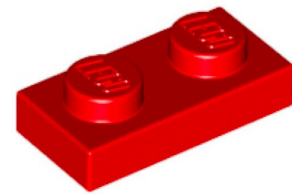


End of training

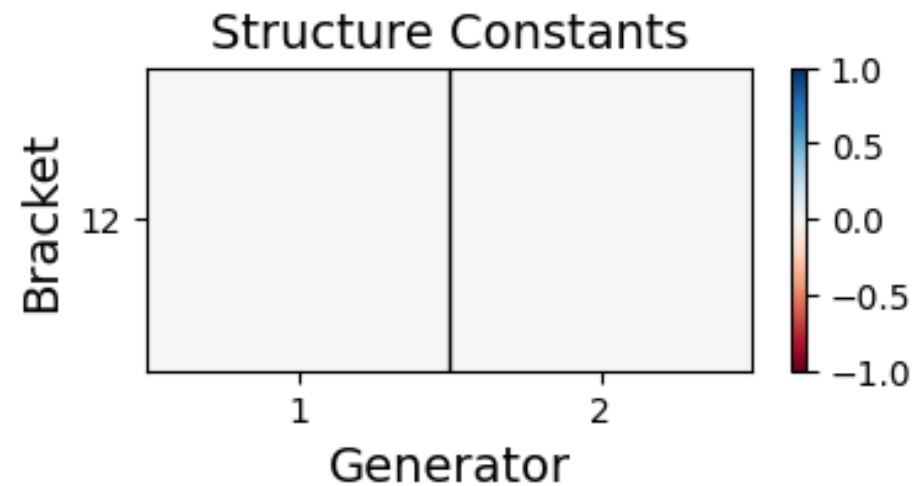
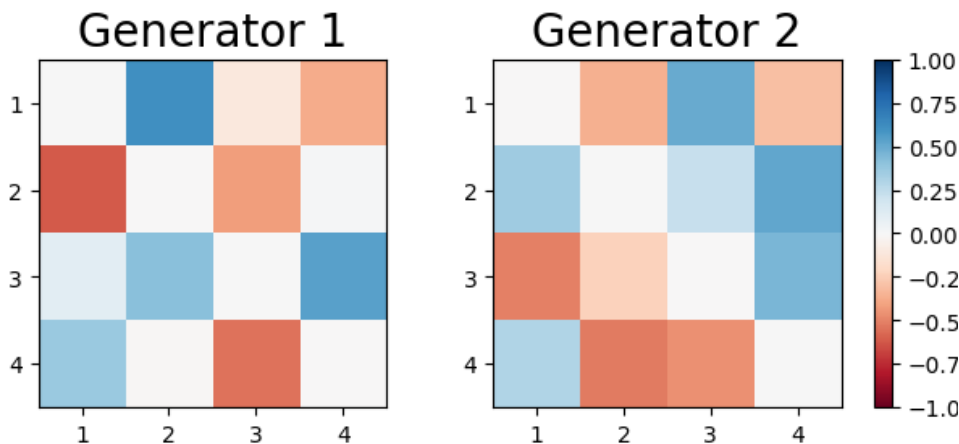
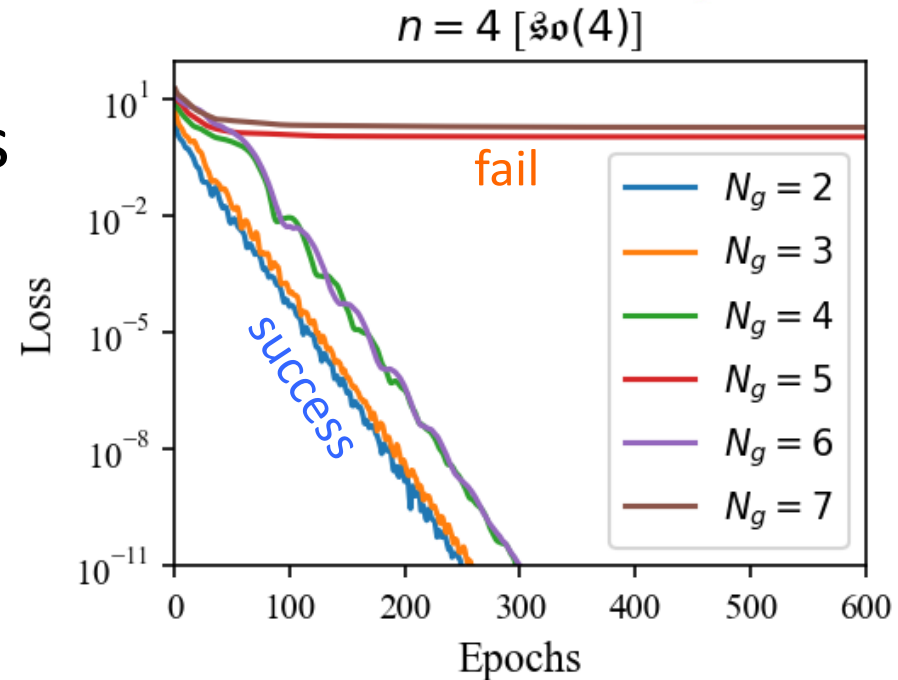
Epoch: 300 | Angles = 90.0°, 90.0°, 90.0°



# Rotations in 4D

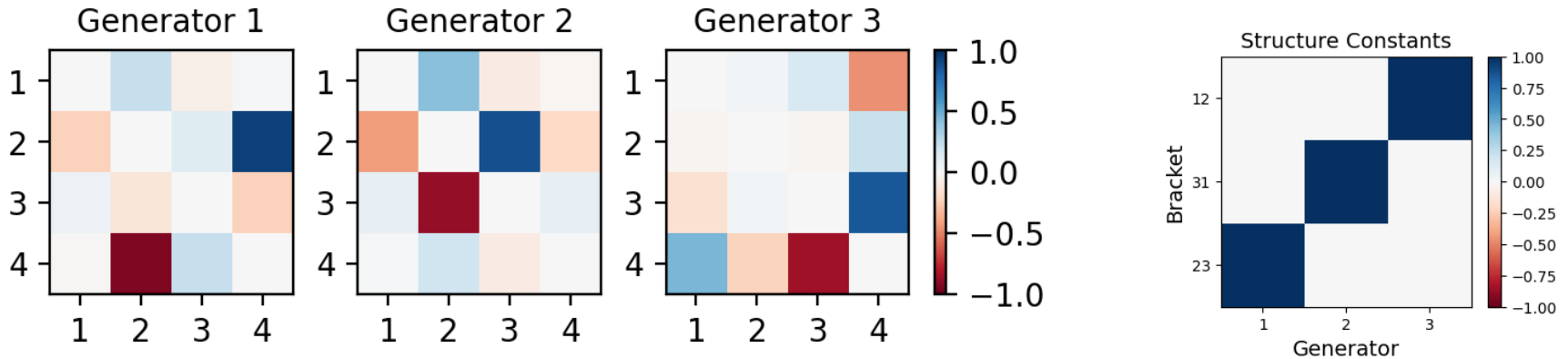


- The training is successful for  $N_g=1,2,3,4,6$  generators
- The full symmetry group  $SO(4)$  is rank 2
  - There exists an Abelian (Cartan) subalgebra with  $N_g=2$  generators

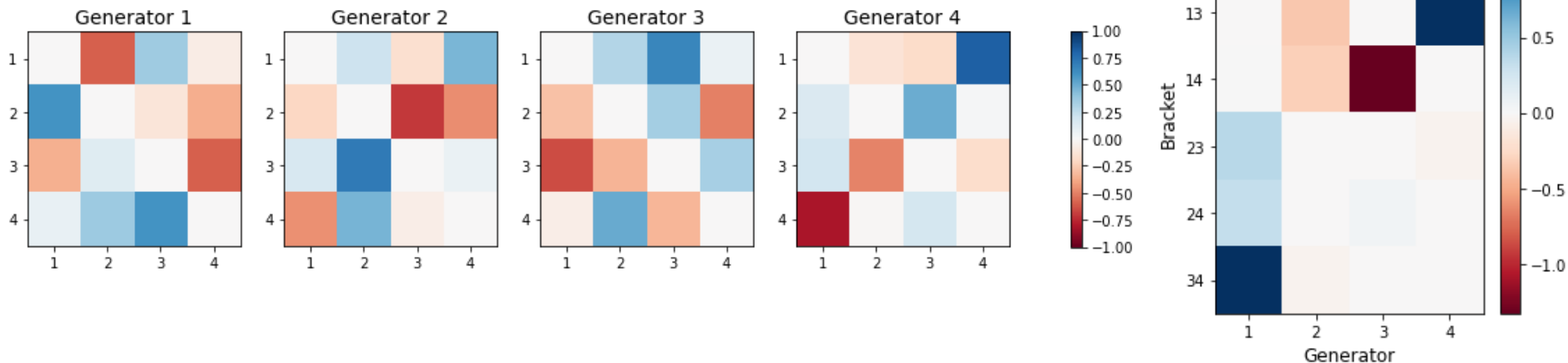


# Rotations in 4D: other subalgebras

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

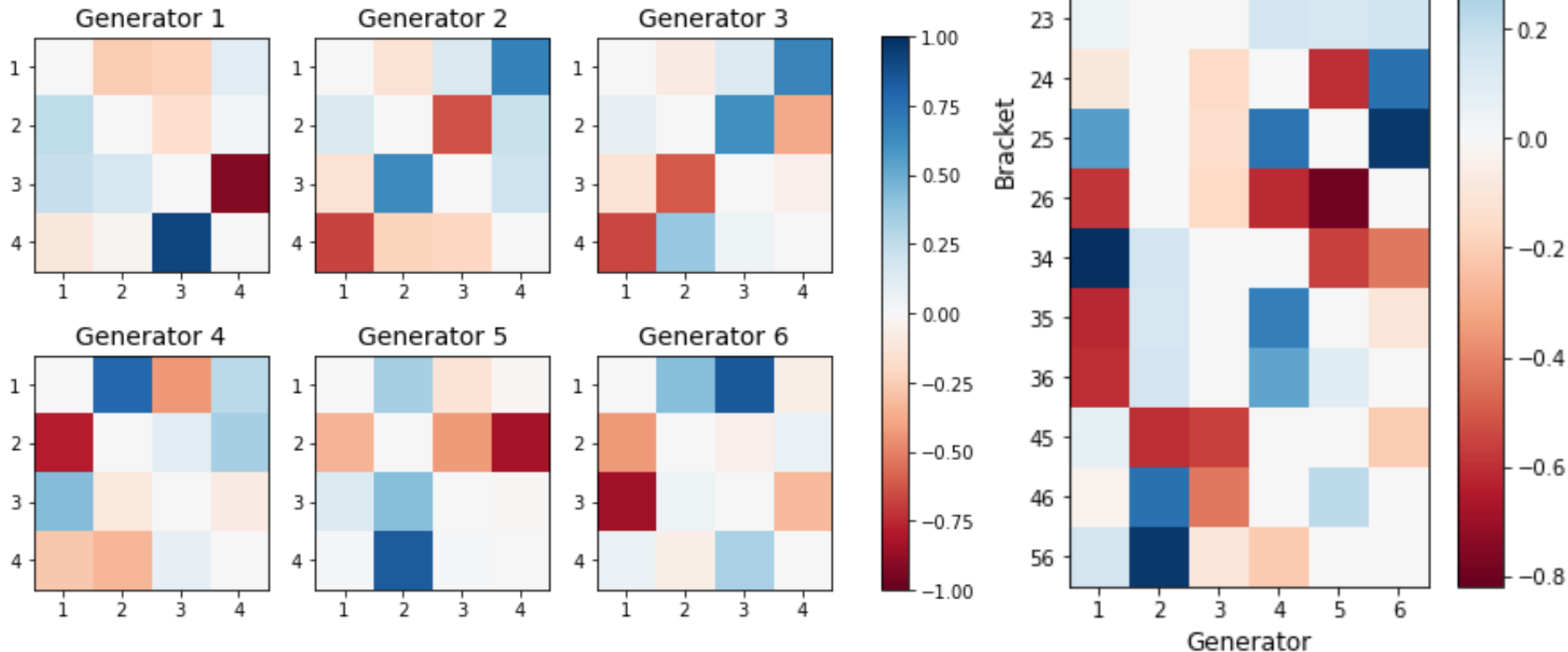


- The training is successful for  $N_g=4$  generators
  - $SO(3) \times SO(2)$  is a subgroup of  $SO(4)$



# Rotations in 4D: full so(4) algebra

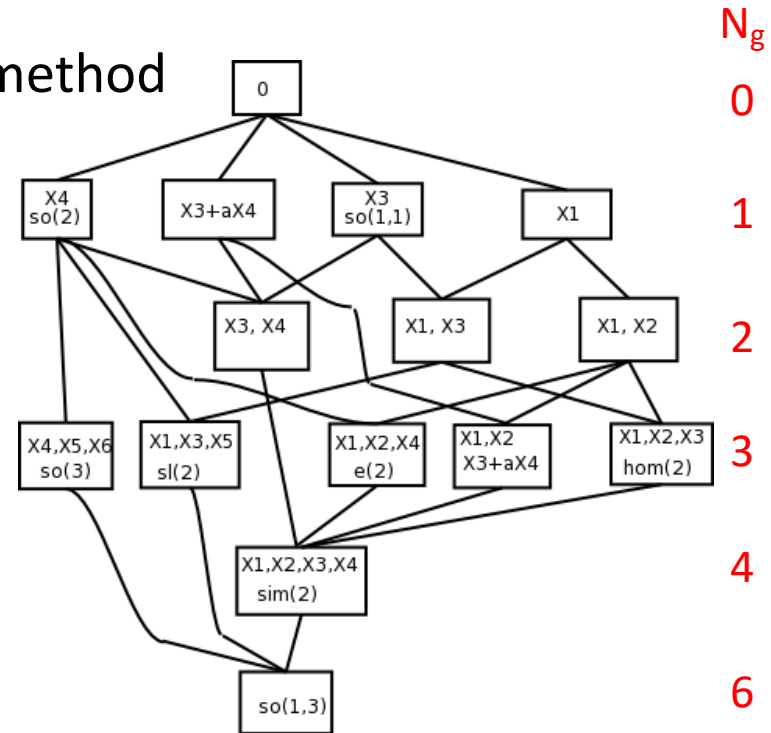
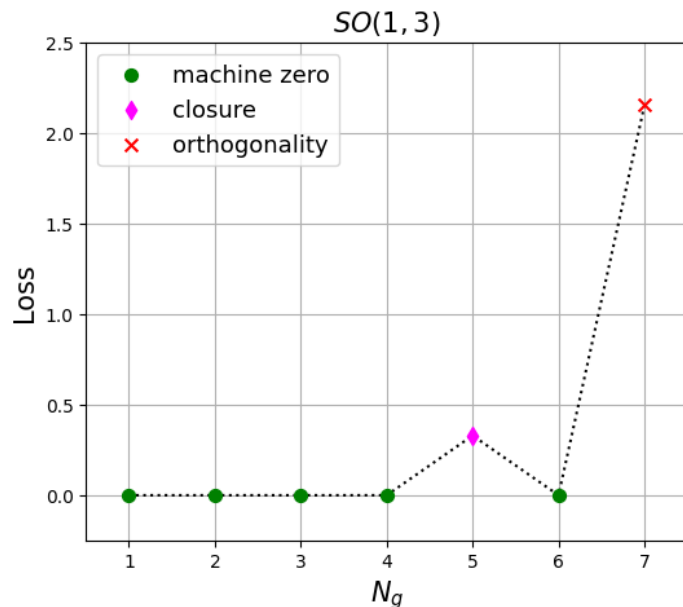
- $N_g=6$  is the maximum number of generators which
  - are symmetries
  - are orthonormal
  - form a closed algebra





# Lorentz Group

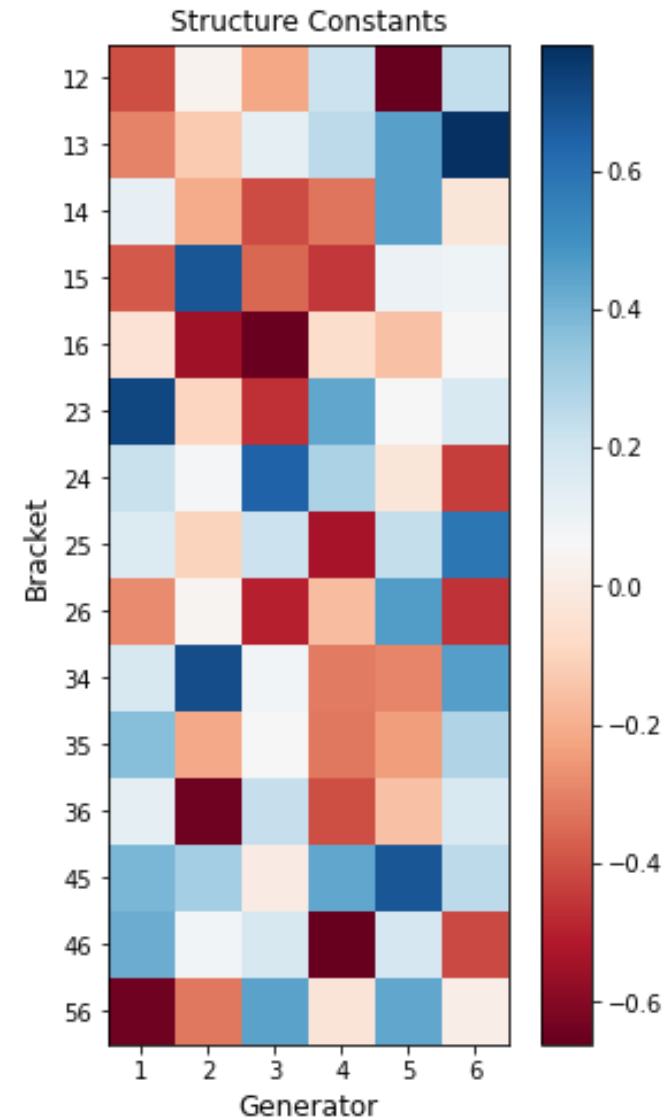
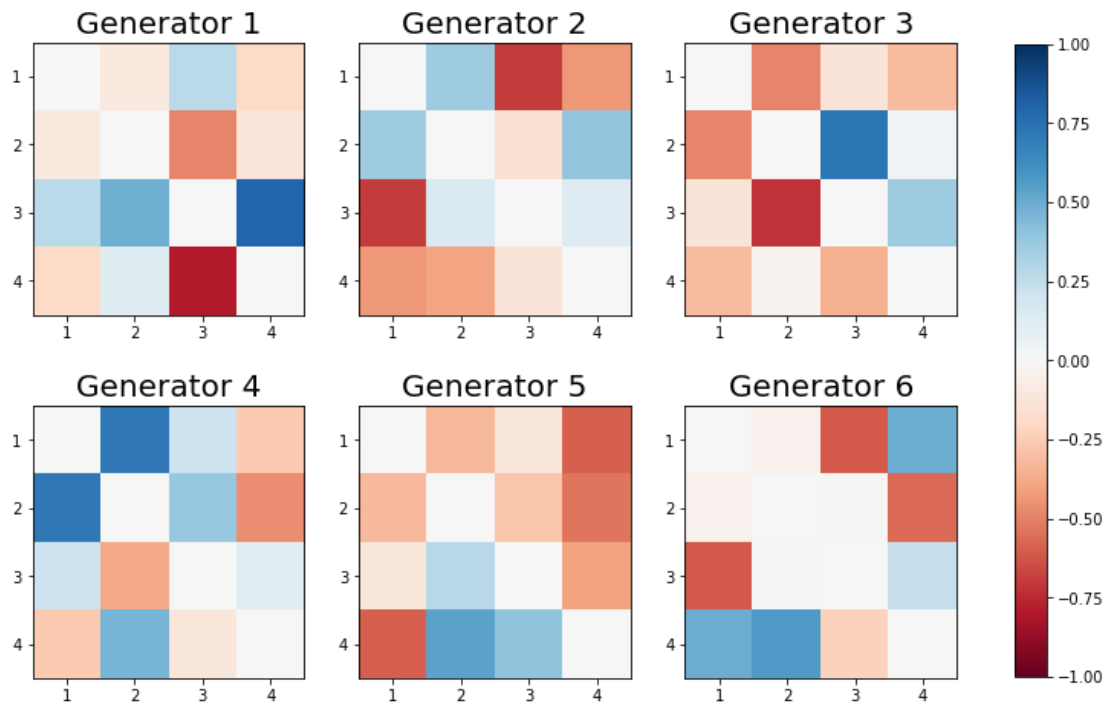
- **Oracle:** the Minkowski metric  $t^2-x^2-y^2-z^2$
- The Lorentz group is the cornerstone of relativity
- Being **SO(1,3)**, it resembles SO(4), but has richer subalgebra structure
  - **$N_g=2$ :** both an Abelian and a non-Abelian subalgebra
  - **$N_g=3$ :** five different types of non-Abelian 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



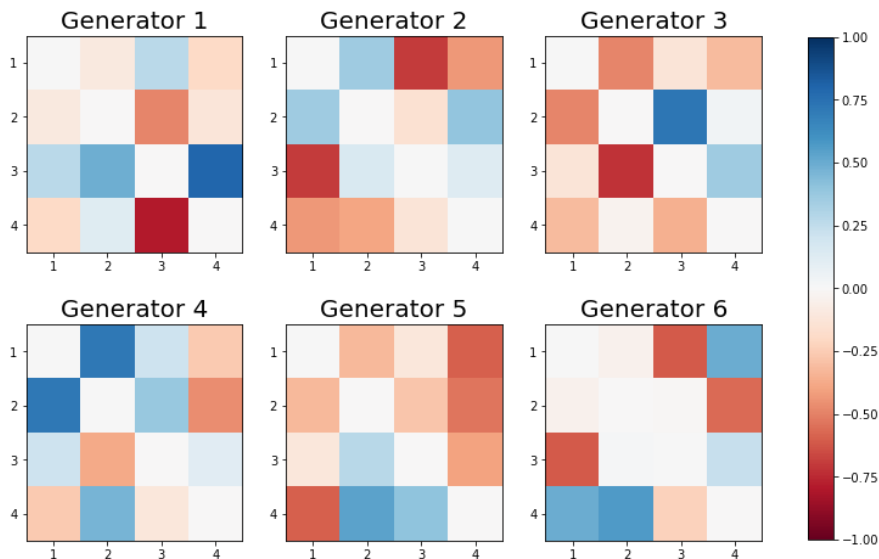
# Learning Sparse Representations

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

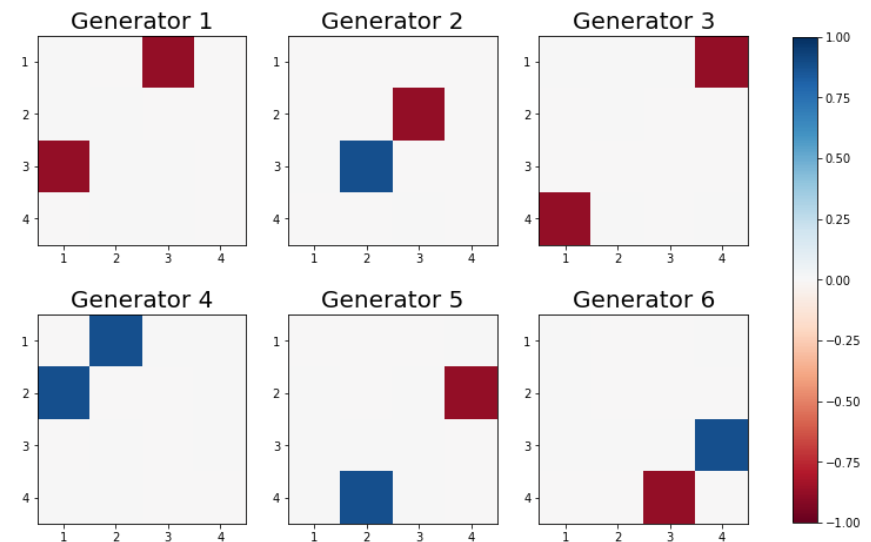
$$L_{\text{sp}}(\mathbb{G}) = \sum_{j,k=1}^n \sum_{j',k'=1}^n |\mathbb{G}^{(jk)} \mathbb{G}^{(j'k')}| (1 - \delta_{jj'} \delta_{kk'})$$

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

Before adding the sparsity loss



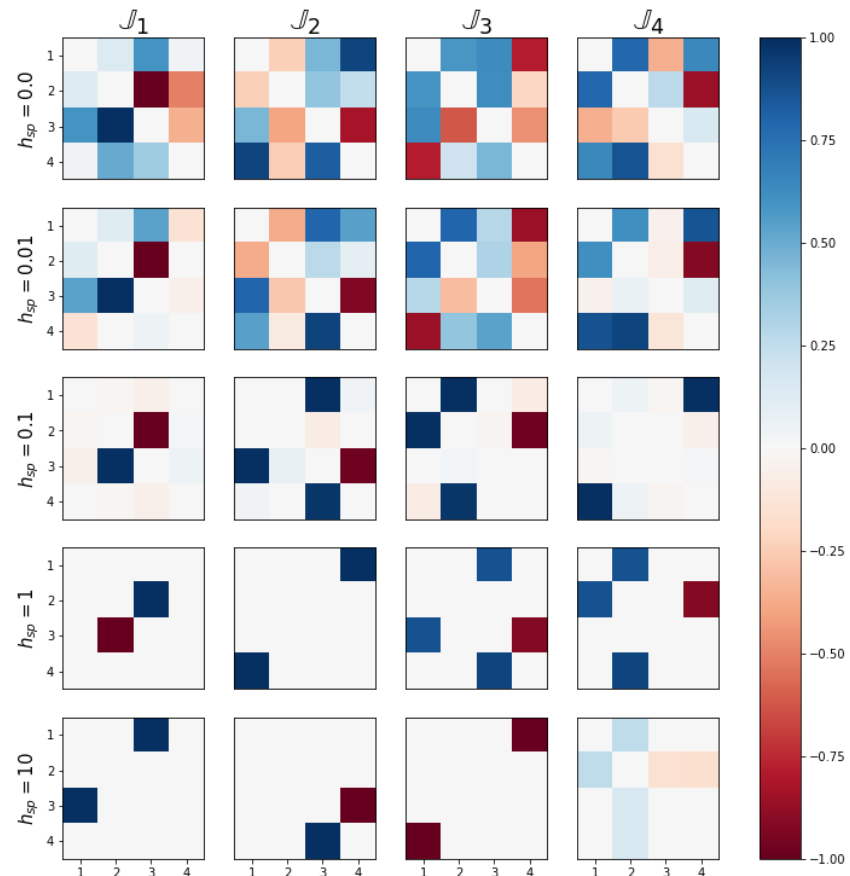
After adding the sparsity loss



# 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

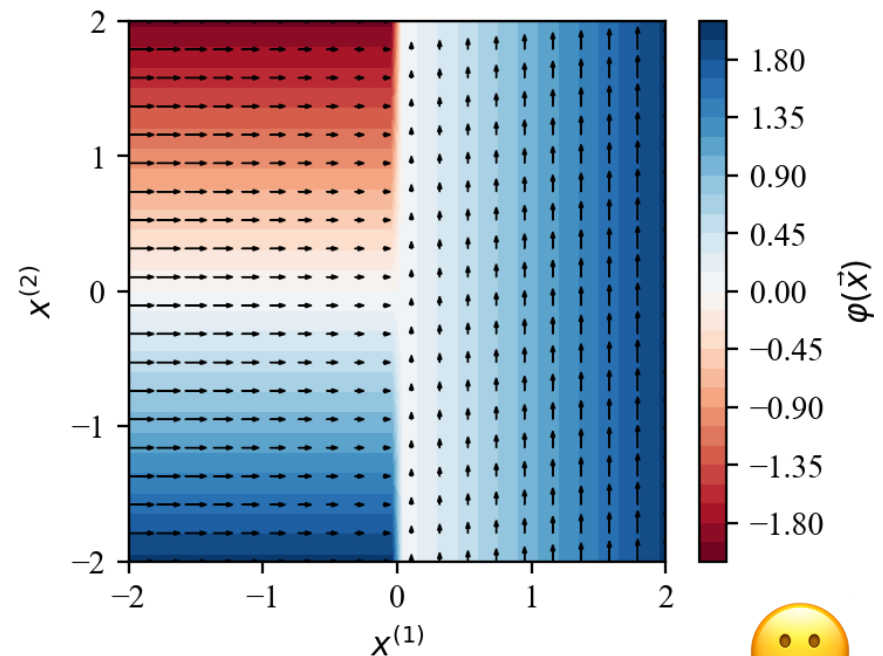
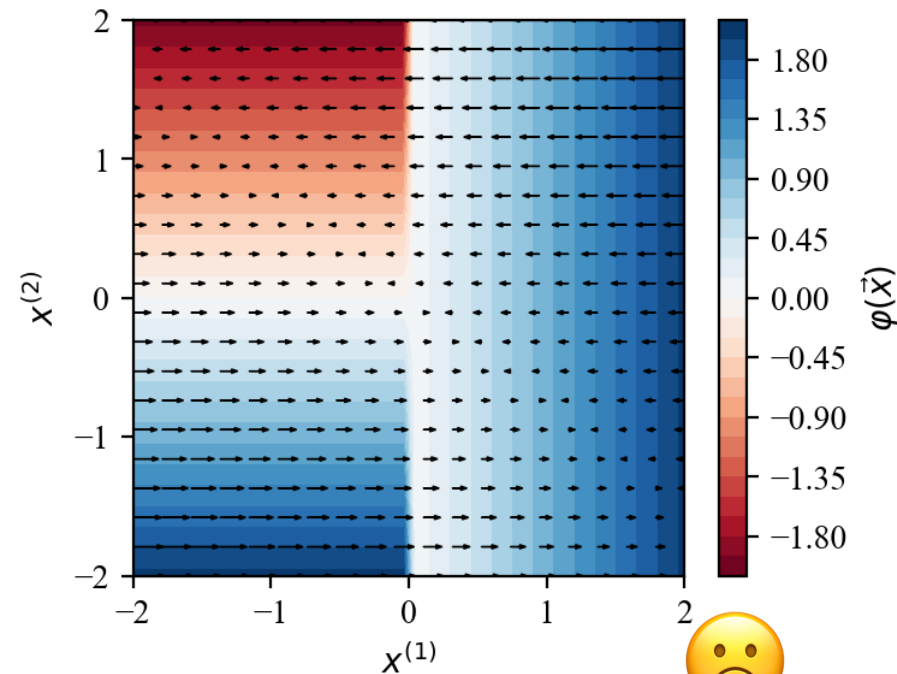
# Discontinuous oracles – example I

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

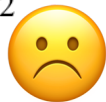
$$\varphi(\mathbf{x}) = \begin{cases} -x^{(2)}, & \text{for } x^{(1)} < 0, \\ +x^{(1)}, & \text{for } x^{(1)} \geq 0. \end{cases}$$

Shallow network  
(no hidden layers)

Deep network  
(3 hidden layers)



Arrows cross contours



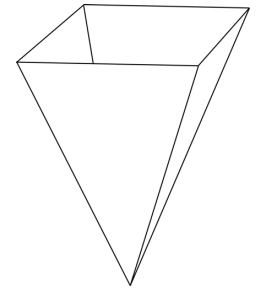
Arrows aligned with contours



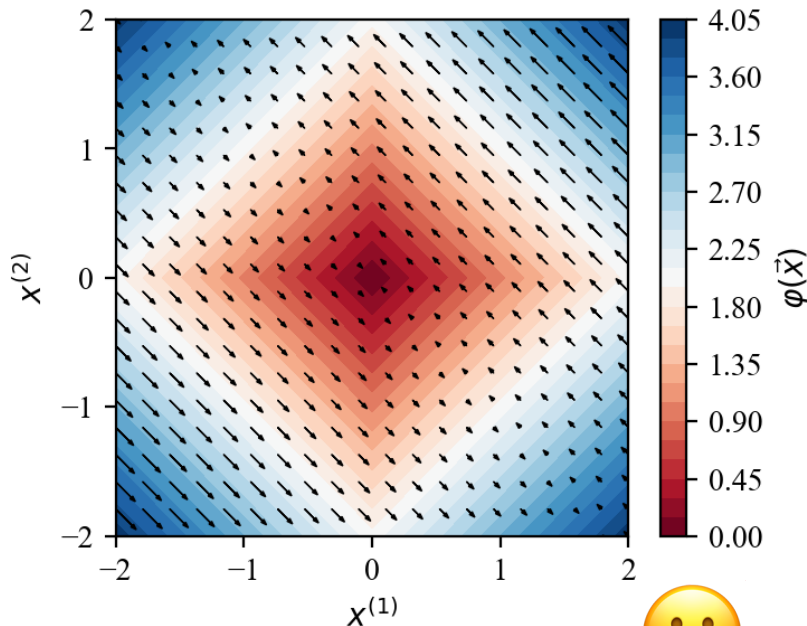
# Discontinuous oracles – example II

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

$$\varphi(\mathbf{x}) = |x^{(1)}| + |x^{(2)}|$$



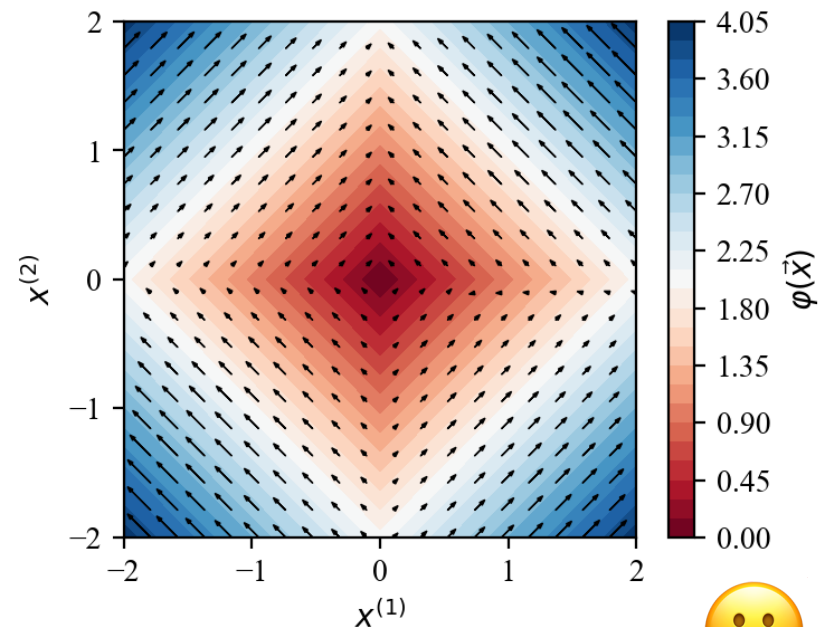
Shallow network  
(no hidden layers)



Arrows cross contours



Deep network  
(3 hidden layers)

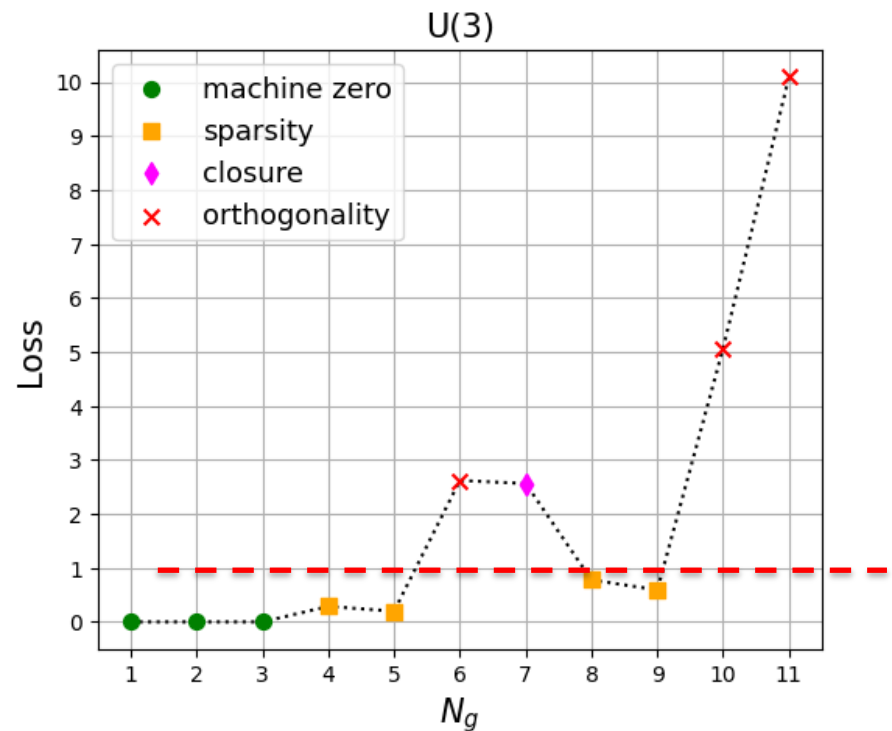
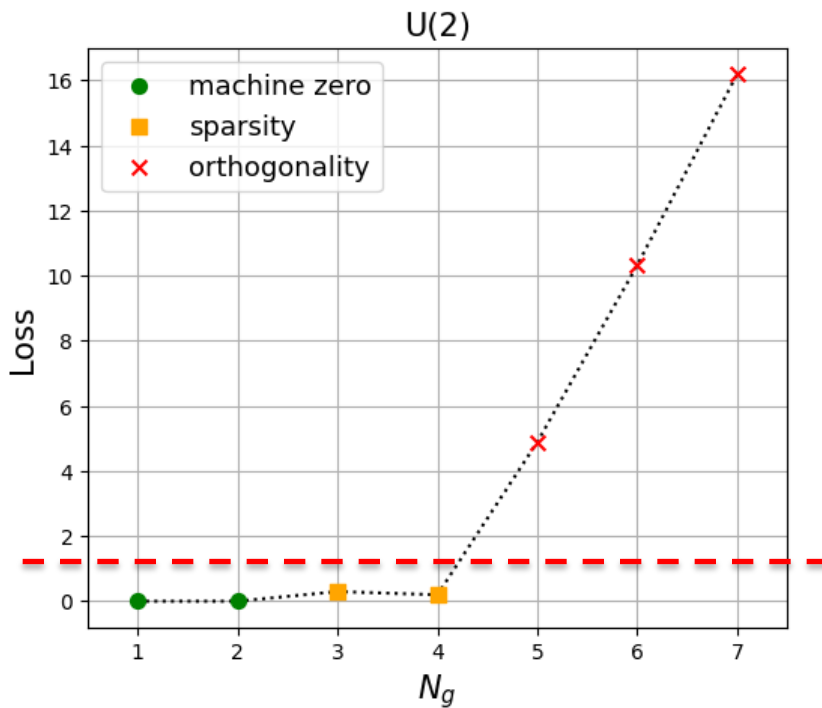


Arrows aligned with contours



# Unitary groups: subgroup structure

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



# The subalgebra structure of $\mathfrak{u}(4)$

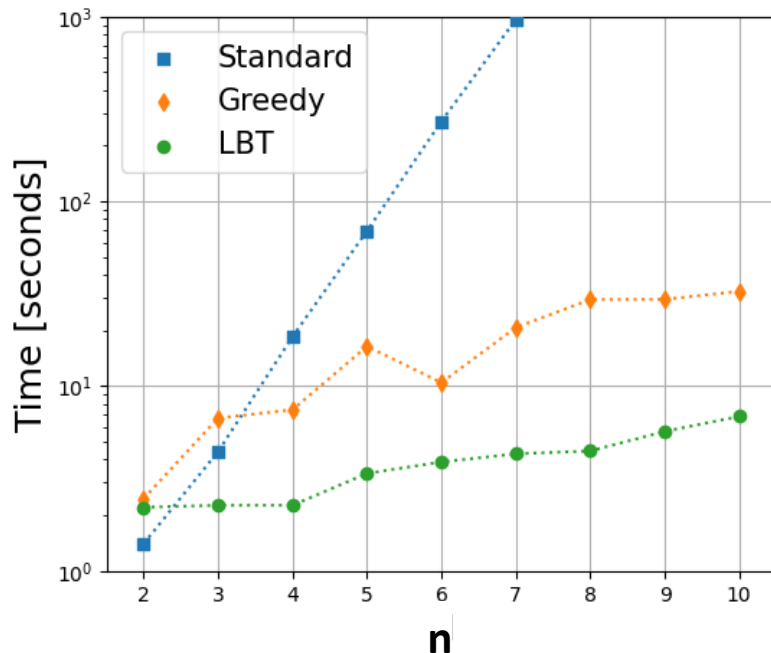
$$\mathfrak{h} = \mathfrak{h}_1 \oplus \mathfrak{h}_2 \oplus \cdots \oplus \mathfrak{h}_h$$

- 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

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

# 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.

---

```

1 Parameters:  $\lambda, L_{min}, N_{epochs}$ ;
2  $\{\mathbb{J}\} \leftarrow []$ ;
3  $\mathcal{W} \leftarrow \mathcal{W}_{initial} \sim \mathcal{N}$ ;
4 for  $i$  from 1 to  $N_{epochs}$  do
5    $L \leftarrow L_{greedy}(\mathbb{G}(\mathcal{W}), \{\mathbb{J}\}, \mathbf{x})$ ;
6   if  $L < L_{min}$  then
7     append  $\mathbb{G}(\mathcal{W})$  to  $\{\mathbb{J}\}$ ;
8     goto 3;
9   end
10   $\mathcal{W} \leftarrow \mathcal{W} - \lambda \nabla_{\mathcal{W}} L_{greedy}$ ;
11 end
12 stop

```

---

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

---

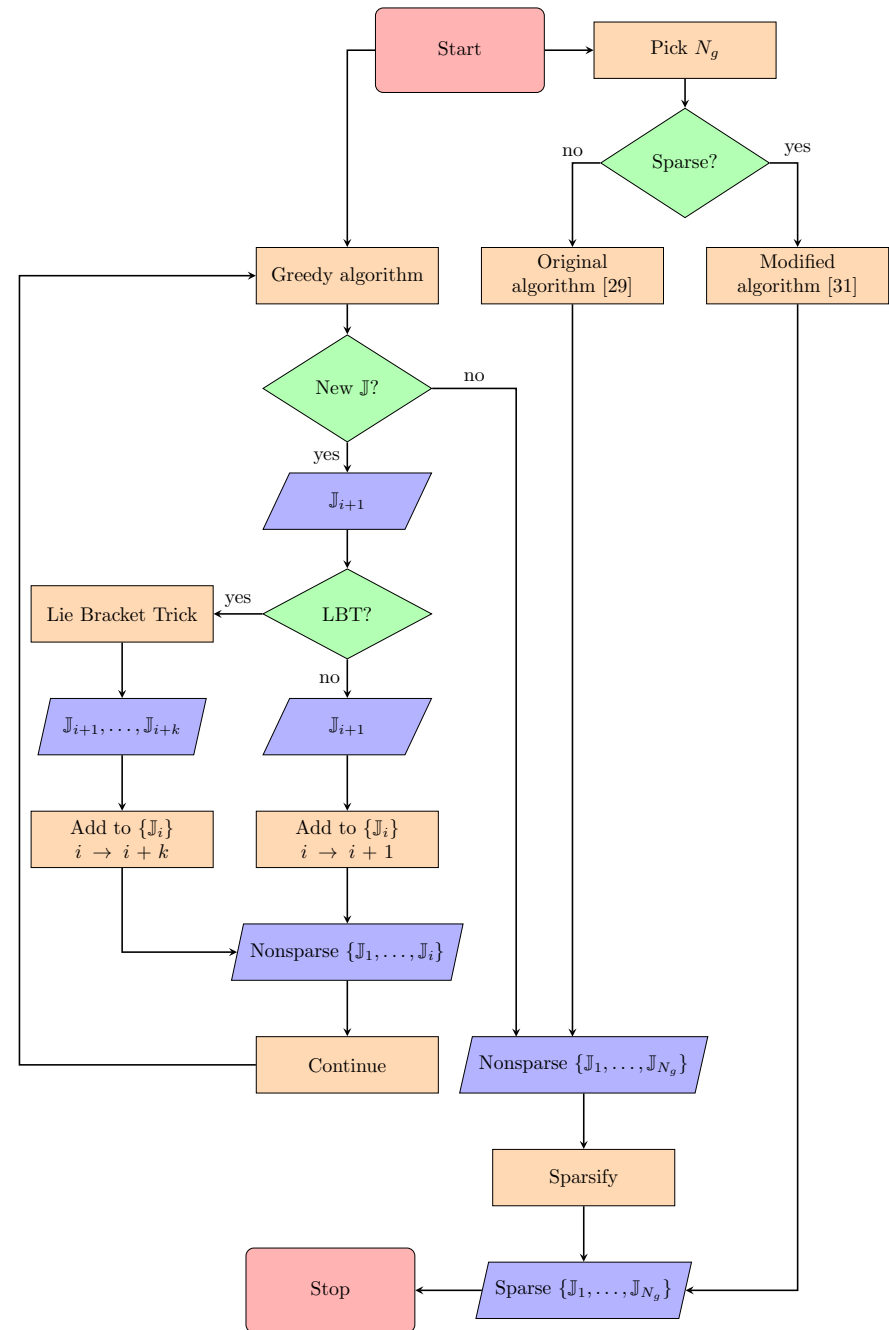
```

1 Input:  $\{J_1, \dots, J_i\}$ : known algebra;  $J_{i+1}$ : new
   generator;
2 append  $J_{i+1}$  to  $G$ ;
3 repeat
4    $k \leftarrow |G|$ ;
5    $i \leftarrow |J|$ ;
6   append  $G$  to  $J$ ;
7   clear  $G$ ;
8   for  $p$  from 1 to  $i$  do
9     for  $q$  from  $i+1$  to  $i+k$  do
10       $C \leftarrow J_p J_q - J_q J_p$ ;
11       $C \leftarrow C - \sum_{g \in J} \frac{g}{\|g\|} \times (C \cdot g)$ ;
12      if  $\|C\| \neq 0$  then
13         $C \leftarrow \frac{C}{\|C\|}$ ;
14        if  $L_{inv}(C, \mathbf{x}) < L_{min}$  then
15          append  $C$  to  $G$ ;
16        end
17      end
18    end
19  end
20 until  $|G| = 0$ ;

```

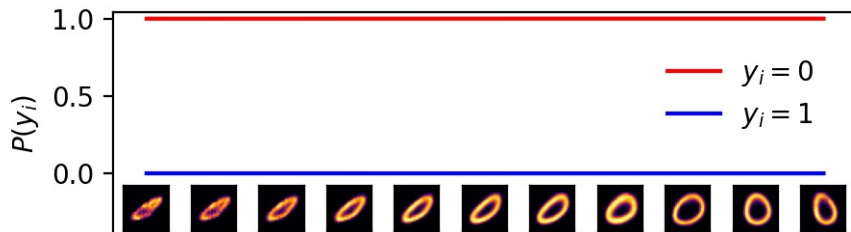
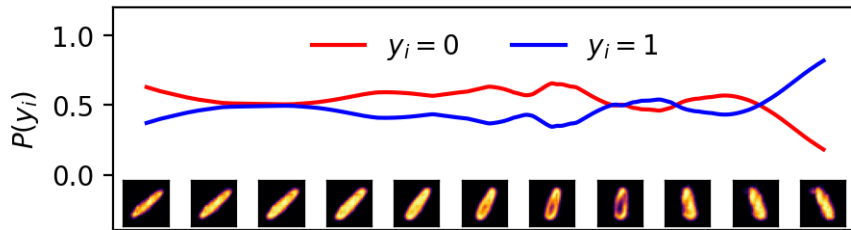
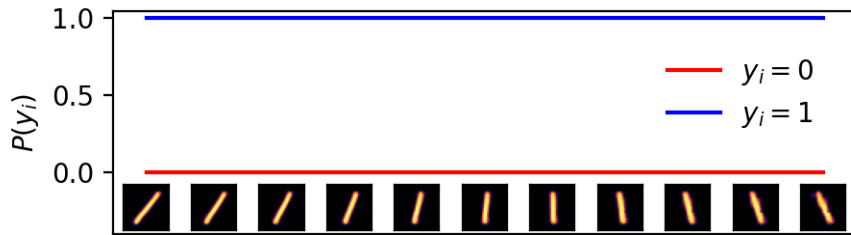
---

# Different Flavors of Symmetry Learning Algorithms

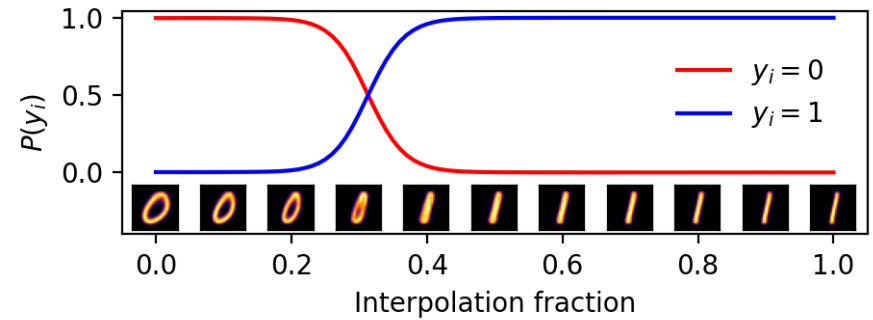


# Symmetries of the digits 0 and 1

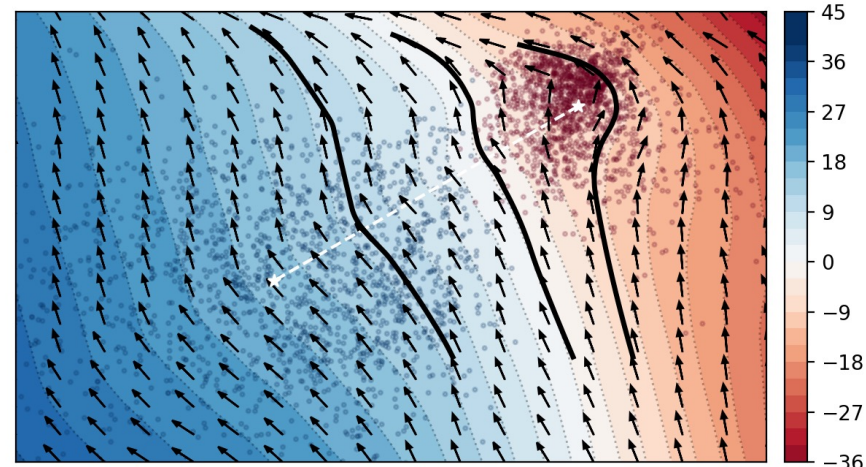
- Symmetry direction



- Orthogonal direction



**Latent space z**



# 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  $M_A$

