Uncovering Singularities in Feynman Integrals via Machine Learning

The 5th Symposium on Quantum Field Theory and its Applications

Yuanche Liu, Dept. of Mod. Phys., **USTC**

2025.10.31

Based on **arXiv:2510.10099**, in collaboration with **Yingxuan Xu** & **Yang Zhang**

01

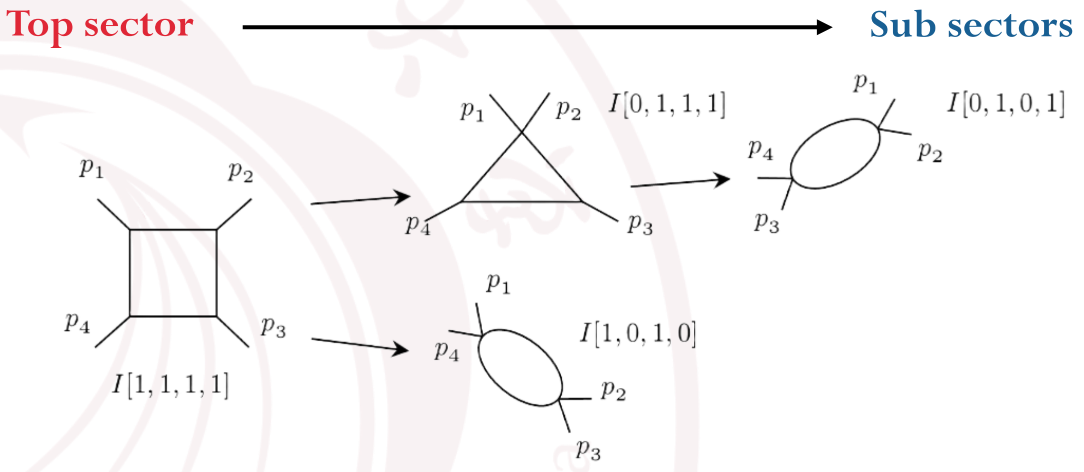
Introduction on Symbology

About Feynman integrals, canonical differential equations and their symbol letters.

Feynman Integrals and their Family Structure

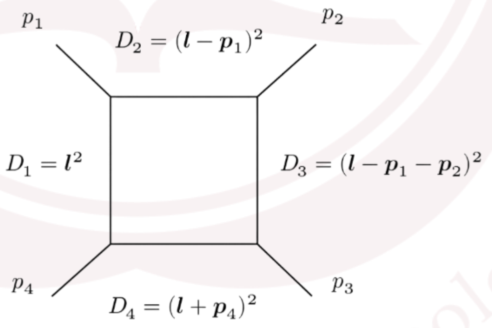
Feynman Integrals can be **recursively built** from top sector to sub sectors

- Form a **family structure**, in which integrals are determined by combination of propagators.



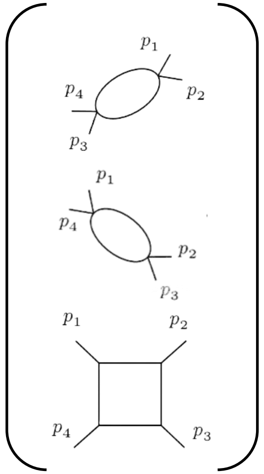
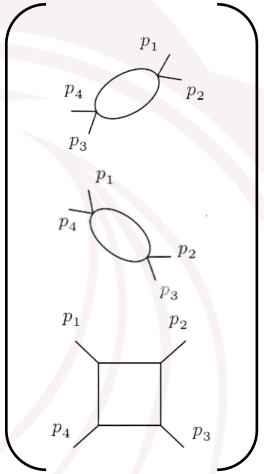
The well-known **IBP relations**[F. Tkachov, 1981] uncover the **linear relations** between different integrals

- One can determine a **basis** in respect to linear relations – the **Master Integrals**



$$\begin{aligned}&-\alpha_4 I_{\text{box}}[-1+\alpha_1,\alpha_2,\alpha_3,1+\alpha_4] - \alpha_3 I_{\text{box}}[-1+\alpha_1,\alpha_2,1+\alpha_3,\alpha_4]\\ &- \alpha_2 I_{\text{box}}[-1+\alpha_1,1+\alpha_2,\alpha_3,\alpha_4] + \alpha_3 s I_{\text{box}}[\alpha_1,\alpha_2,1+\alpha_3,\alpha_4]\\ &+ (-2\alpha_1-\alpha_2-\alpha_3-\alpha_4+D) I_{\text{box}}[\alpha_1,\alpha_2,\alpha_3,\alpha_4] = 0\end{aligned}$$

Canonical Differential Equations

∂_s  $=\begin{pmatrix} 0 & 0 & 0 \\ 0 & -\frac{\varepsilon}{s} & 0 \\ \frac{4\varepsilon-2}{st(s+t)} & \frac{-4\varepsilon+2}{s^2(s+t)} & -\frac{\varepsilon t+s+t}{s(s+t)} \end{pmatrix}$ 

$$\boldsymbol{I}(s,t,\varepsilon)=\{I_{\text{box}}[0,1,0,1],I_{\text{box}}[1,0,1,0],I_{\text{box}}[1,1,1,1]\}$$

$$\boldsymbol{I}=\varepsilon^2\exp(\varepsilon\gamma_E)(-s)^{\varepsilon}\begin{pmatrix} \frac{1-2\varepsilon}{\varepsilon}I_{\text{box}}[0,1,0,1] \\ \frac{1-2\varepsilon}{\varepsilon}I_{\text{box}}[1,0,1,0] \\ stI_{\text{box}}[1,1,1,1] \end{pmatrix}$$

$$\frac{\partial}{\partial x_i}\boldsymbol{I}(\boldsymbol{x},\varepsilon)=\varepsilon\boldsymbol{M}_i(\boldsymbol{x})\boldsymbol{I}(\boldsymbol{x},\varepsilon)$$

$$M(x)=\begin{pmatrix} -\frac{1}{x} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{2}{x(x+1)} & -\frac{2}{x+1} & -\frac{1}{x(x+1)} \end{pmatrix}$$

Taking **derivatives** for master integrals in respect to **kinematic variables** lead to a **differential equation** system.

Change to **Canonical Basis**, one obtain ε-factorized **Canonical Differential Equations**[J. Henn, 2013]

- Sometimes this can be very complicated because of **complicated function space**, see [ε-collaboration, 2025]

Symbol Letters

$$\frac{\partial}{\partial x_i}\boldsymbol{I}(\boldsymbol{x},\varepsilon) = \varepsilon \boldsymbol{M}_i(\boldsymbol{x})\boldsymbol{I}(\boldsymbol{x},\varepsilon)$$

$$\boldsymbol{M}(x) = \begin{pmatrix} -\frac{1}{x} & 0 & 0 \\ 0 & 0 & 0 \\ \frac{2}{x(x+1)} & -\frac{2}{x+1} & -\frac{1}{x(x+1)} \end{pmatrix}$$

$$M(x) = \sum_i a_i \,\mathrm{d}\log \boxed{W_i}, \qquad a_i \in \mathbb{Q}^{n\times n}$$

$$W_i = P \in \mathbb{Q}[x_i]$$

parity **even letter**

$$W_i = \frac{\sqrt{P}-Q}{\sqrt{P}+Q}, \qquad P, Q \in \mathbb{Q}[x_i]$$

parity **odd letter**

Symbol letters

- **Symbol Alphabet** = {All **Symbol Letters**}
- Very **limited** for a certain family!

CDE Matrix contains different **Symbol letters**, revealing the **analyticity**(poles, regions) of Feynman Integrals.

- Ingredients for **analytical bootstrap** on Feynman Integrals or Amplitudes[Carrolo, Chicherin, Henn, Yang, Zhang, 2025]
- **Cluster Algebra** structures[Pokraka, Spradlin, Anastasia, Weng, 2025] from Mathematical Physics

Some previous algorithms to generate **Symbol letters**:

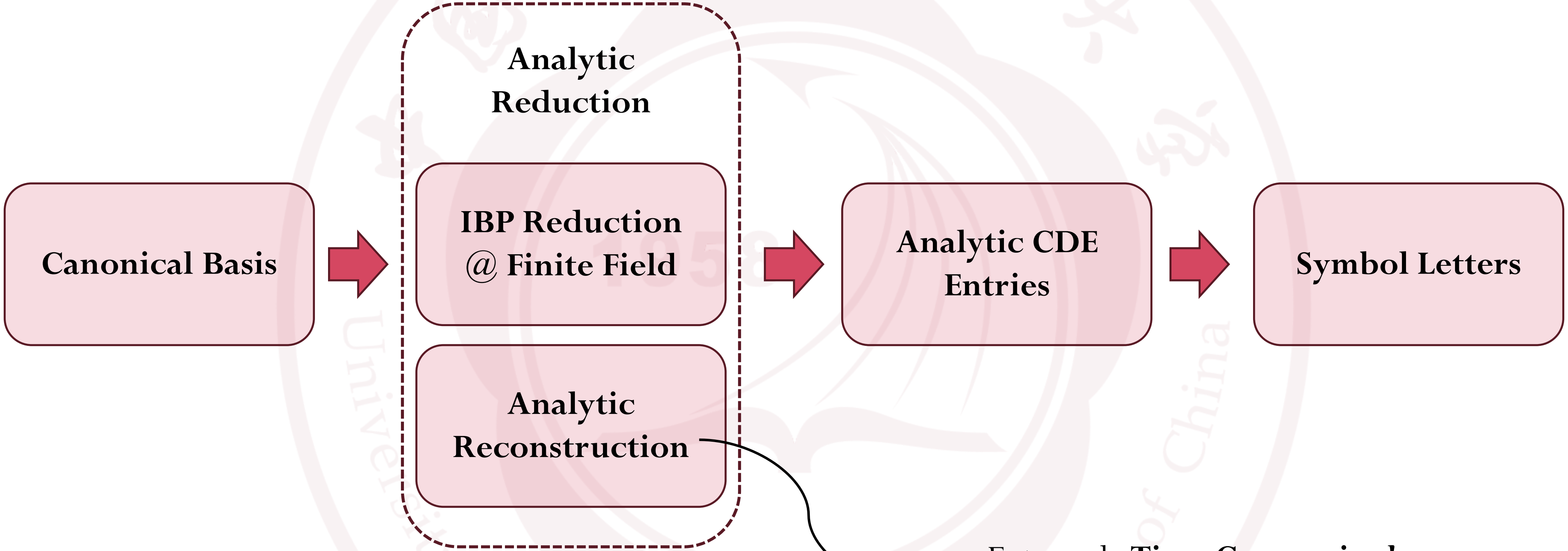
- `Effortless`[Antonela, 2024] and `SOFIA`[Correia, Giroux, Mizera, 2025], using **Landau analysis** and some conjectures.
- `BaikovLetter`[Jiang, Lian, Yang, 2024] using **Baikov Rep.+ Maximal Cut**.

02

Machine Learning And Symbolic Regression

An introduction about Machine Learning and Symbolic Regression.

Why Symbolic Regression?



Extremely **Time Consuming!**

- 2l6p planar[Henn, Matijasic, Miczajka, Peraro, Xu, Yang, 2025]
- **2 hrs/pt** for **numerical** CDE matrix
- **Unable to reconstruct** all letters!

Why Symbolic Regression?

Canonical Basis → **Black Box** → **Analytic CDE Entries** → **Symbol Letters**

Black Box

Input:
Master Integrals

Output:
Analytic Coefficients

Hint:
Easy for **numeric**
Hard for **analytic**

This is a typical Symbolic Regression task!

Machine Learning: Symbolic Regression

Numerical Data Set

$x = 0.3333, f(x) = 2.1111$
$x = 0.0000, f(x) = 2.0000$
$x = 1.0000, f(x) = 3.0000$
$\dots$

Search & Verification

No	Expr	Comp	Error
1	0.4exp(x)	3	0.132
2	x^2+2.000	4	0.000
3	x+ln(x)	4	0.143
4	x^3+x^2	5	0.475

Symbolic Expressions

$$f(x) = x^2 + 2.000$$

Symbolic Regression: Search interpretable **symbolic expressions** from **numerical samplings**.

Various Routings:

- **PySR**[M.Cranmer, 2023]—Genetic Programming,
- **SymbolicGPT**[Valipour, Panju, You, Ghodsi, 2021]—LLM based,
- **IdeaSearch-Fit**[Song, Cai, Zhang, Wei, Pan et.al, 2025]—GP + LLM based mixing strategy…

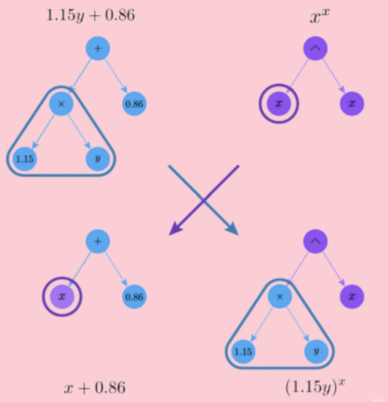
Machine Learning: Symbolic Regression

Input Data
`X, y`

Symbolic Regression
PySR

Selected Equation
“X+ln(y)”

Searching Tree



Pareto Frontier

No	Expr	Comp	Error
1	x+y	3	0.132
2	x^2+0.1y	5	0.329
3	x+ln(y)	4	0.003
4	x^3+y^2	5	0.475

PySR is a powerful **Symbolic Regression** toolkit.

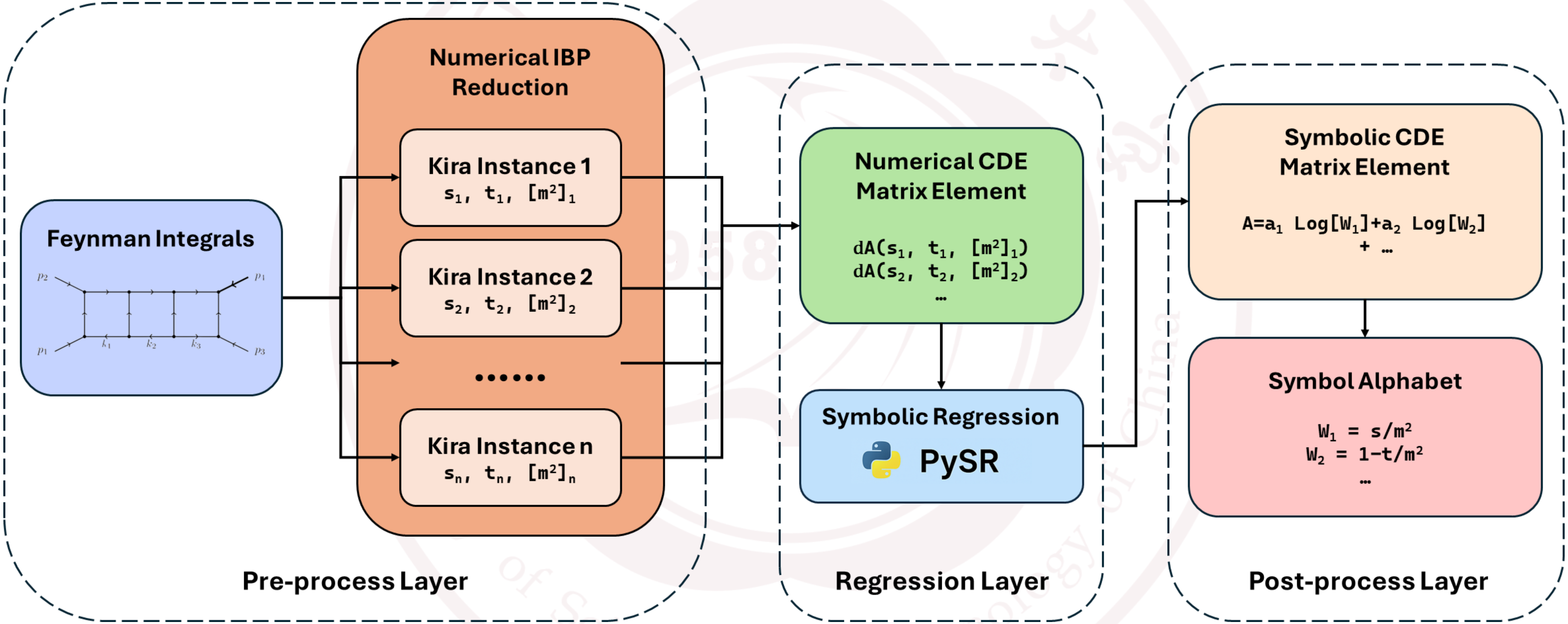
- **No pre-train**: Based on Genetic Programming. Easy to use, no GPU requirement.
- **Efficiency** & **Accuracy**: Pareto Frontier from complexity and accuracy.
- **Sufficient Extension**: Support Julia & Python user-defined functions, and many advanced math operations…

03

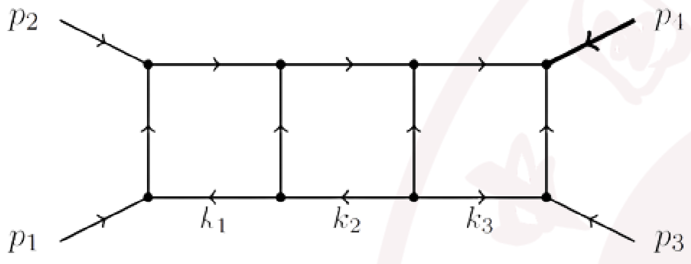
Apply Machine Learning to Symbol Alphabet

Detailed workflows and examples for Symbol Alphabet building.

Workflow Introduction



Examples: 3l4p1m & 2l3p1m-np

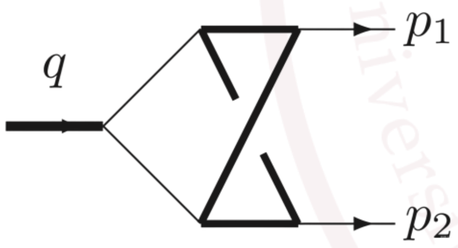


p_4 has mass m^2, others are massless.

3l4p1m[Vita, Matrolia, Schubert, Yundin, 2014]

- **83MIs**, **2**(+1) **scales**: $x=s/m^2$, $y=t/m^2$
- 6 different Symbol letters—**all successfully built**!
- KIRA reduction:
  - **Num**:~20mins 12 threads of i9-13950HX CPU @ 5.30GHz
  - **Ana**: ~100mins 12 threads

$$W_i = \{x, 1-x, y, 1-y, x+y, 1-x-y\}$$



q has mass s, others are massless.
4 propagators with coincide mass.

2l3p1m-np[Manteuffel, Tancredi, 2017]

- **9MIs**(sub-topology), **1**(+1) **scales**: $x=s/m^2$
- 5 different Symbol letters—**all successfully built**!
- Letters have square roots!
- KIRA reduction:
  - **Num**:~10mins 12 threads of i9-13950HX CPU @ 5.30GHz
  - **Ana**: ~60mins 12 threads

$$W_i = \{\sqrt{x}, \sqrt{x+4}, \sqrt{x-4}, \frac{1}{2}(\sqrt{x}+\sqrt{x+4}), \frac{1}{2}(\sqrt{x}+\sqrt{x-4})\}$$

Trials on more Diagrams…

#Loop \ Family	1 Scale 3p1m,4p0m	2 Scales 4p1m	3 Scales 4p2m	5 Scales 5p0m	5+ Scales 5p1m, 6p0m
1	✓✓	✓✓	✓✓	✓	✓
2	✓✓	✓✓	✓✓	✓	✗
3	✓✓	✓✓	✓✓	✓	—
4	✓✓	—	—	—	—

✓✓ : **Complete Symbol Alphabet** reconstructed;
✓ : **All Even Letters** obtained, remaining ones (**Odd Letters**) can be **constructed manually**;
✗ : Some letters not found;
— : Not tested because of incomplete reference.

3l5p-PBB family
[**Liu**, Matijasic, Miczjka,
YX. Xu, YQ. Xu, Zhang, 2024]

316MIs, 5 scales
20+1 **Even Letters Found.**
10 Odd Letters manually built.

Discussions and Future…

Unlike traditional Machine Learning project…

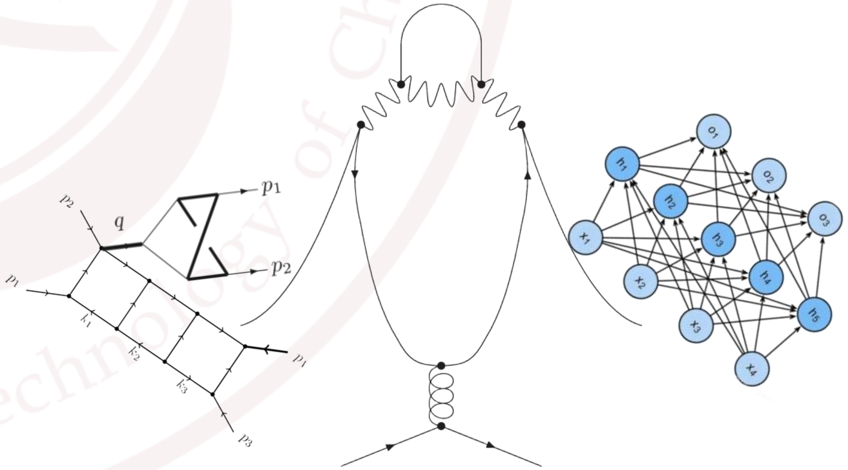
- We pursue a **completely exact solution**, instead of **approximate computation**.

To achieve this…

- We **“weirdly” tune** Hyper-parameter to **“force” the model over-fitting**;
- We also perform an **early-terminate** as soon as **exact solution comes up**.

In the future…

- Improve **accuracy and efficiency**?
- New areas for **AI-ML** to obtain **exact & closed form solution**?
- ……



Thank you!

Yuanche Liu, Dept. of Mod. Phys., **USTC**

2025.10.31