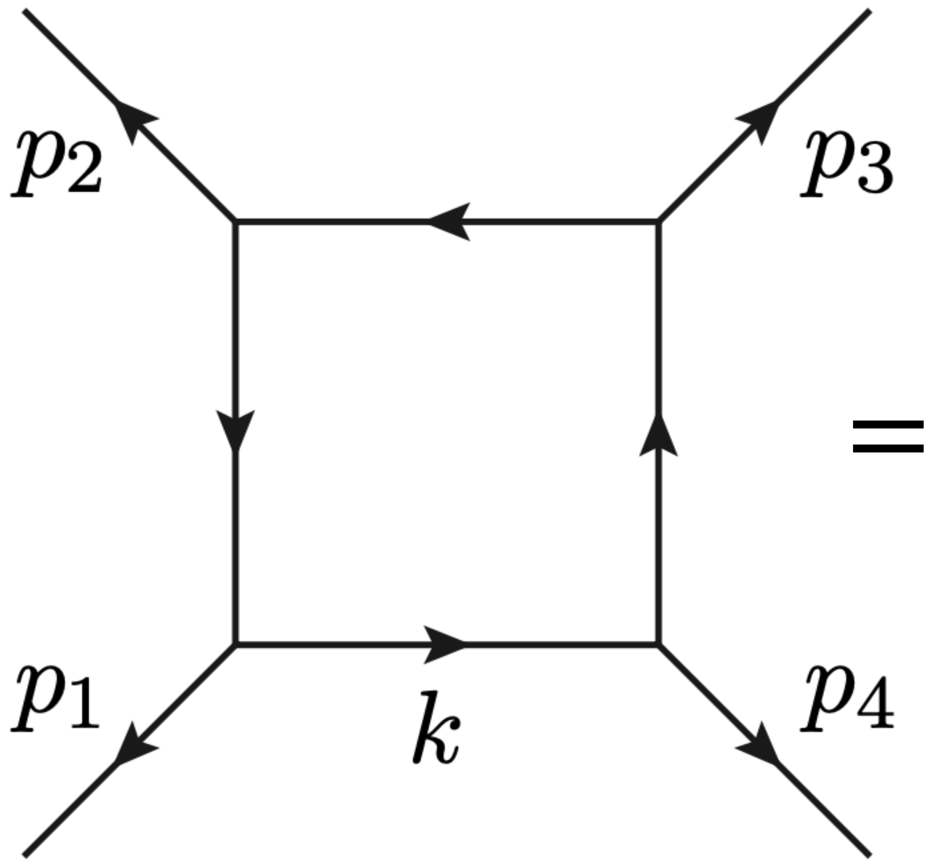
Evaluating Feynman integrals through differential equations: two opposite approaches

Simone Zoia

CERN, 27th November 2023

We need to evaluate Feynman integrals



$$= \int \frac{\mathrm{d}^D k}{\mathrm{i}\pi^{D/2}} \frac{1}{k^2 \, (k+p_1)^2 \, (k+p_1+p_2)^2 \, (k+p_1+p_2+p_3)^2}$$

Essential ingredients of perturbative computations → particle phenomenology

Also: gravitational waves, cosmology, statistical mechanics, mathematics…

Many techniques developed over many years, yet they remain a bottleneck

Integrating by differentiating

[Barucchi, Ponzano ’73; Kotikov ’91; Bern, Dixon, Kosower ‘94; Gehrmann, Remiddi 2000; Henn 2013]

View Feynman integrals as solutions to PDEs

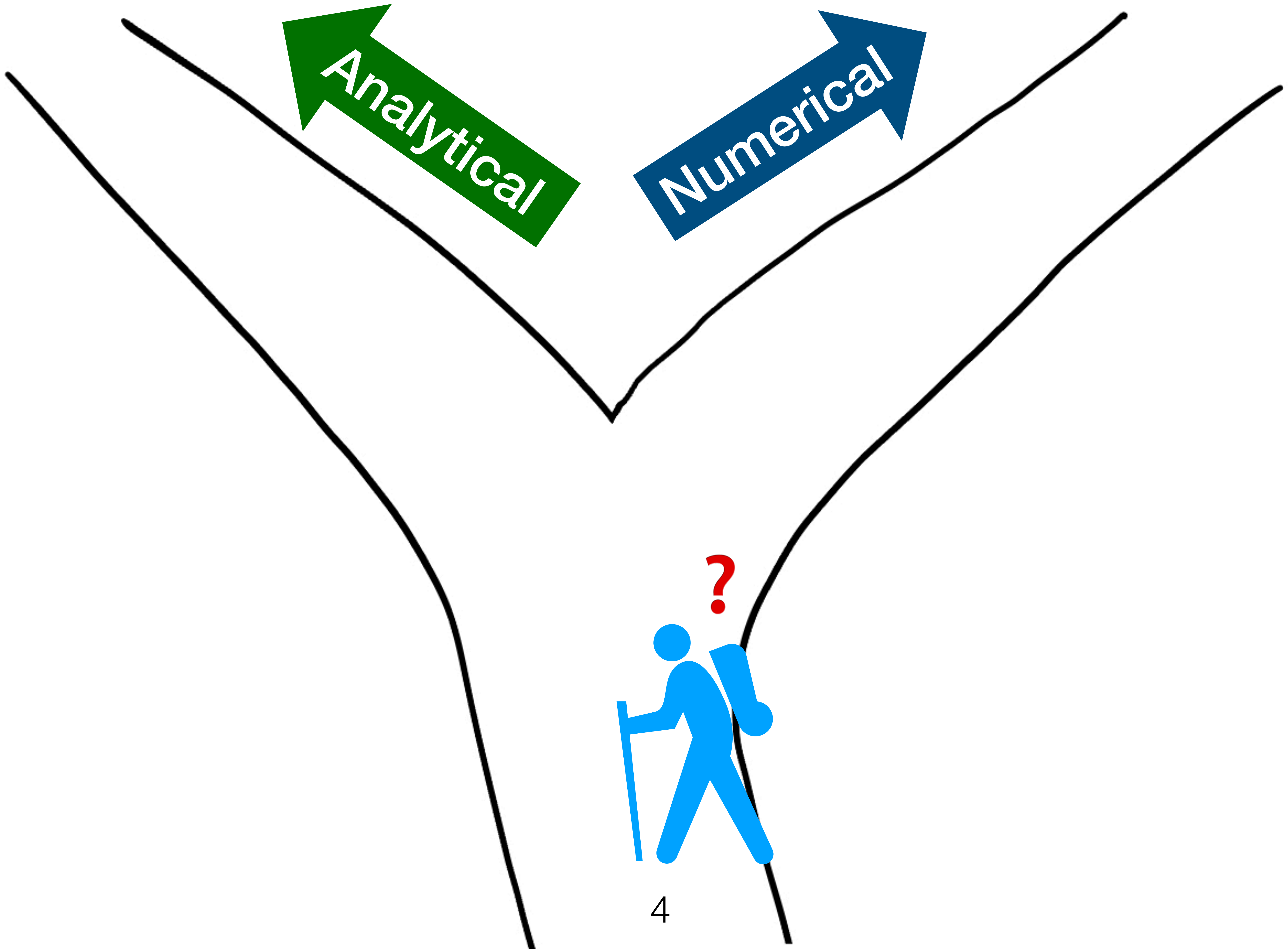
$$\frac{\partial}{\partial s_{12}} \overrightarrow{\mathrm{F}}(s;\epsilon) = A_{s_{12}}(s;\epsilon) \cdot \overrightarrow{\mathrm{F}}(s;\epsilon)$$

Most powerful tool for analytic computation of Feynman integrals

Neat connection with study of special functions

Growing interest for semi-numerical solution with generalised power series

How do we solve the DEs?



Outline

- Quick review of the method of DEs

- Analytic method: “pentagon functions” = basis of special functions

  Chicherin, Sotnikov, SZ (2110.10111)

  Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, SZ (2306.15431)

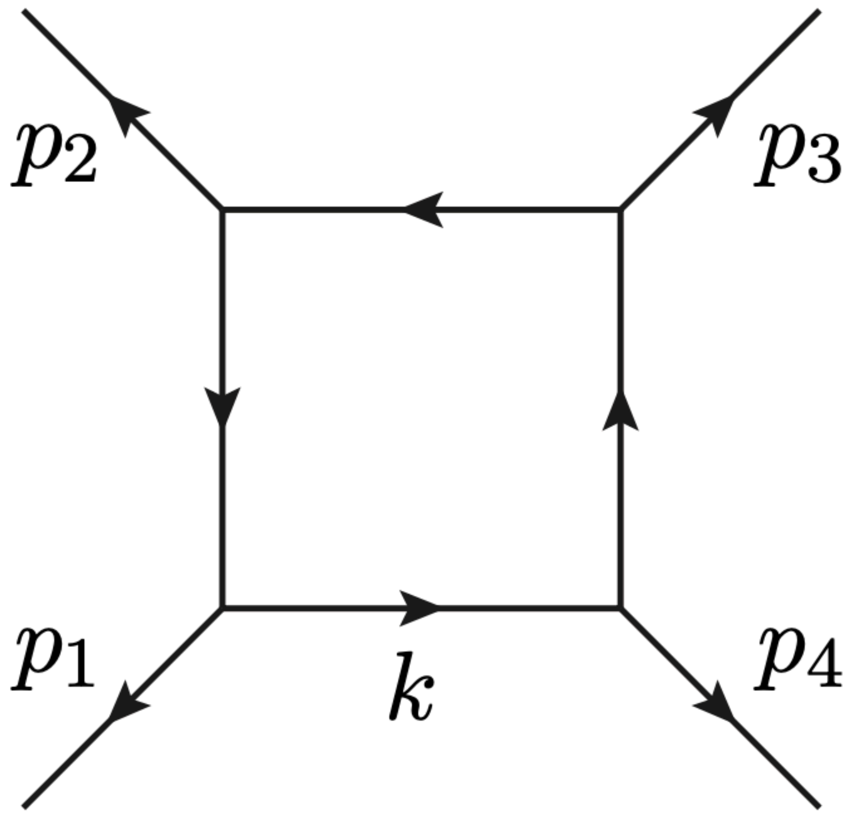
- Numerical method: “physics-informed deep learning”

  Calisto, Moodie, SZ (23XX.XXXXX)

Method of differential equations

Integral families and master integrals

Scalar Feynman integrals with the same propagator structure = **integral family**



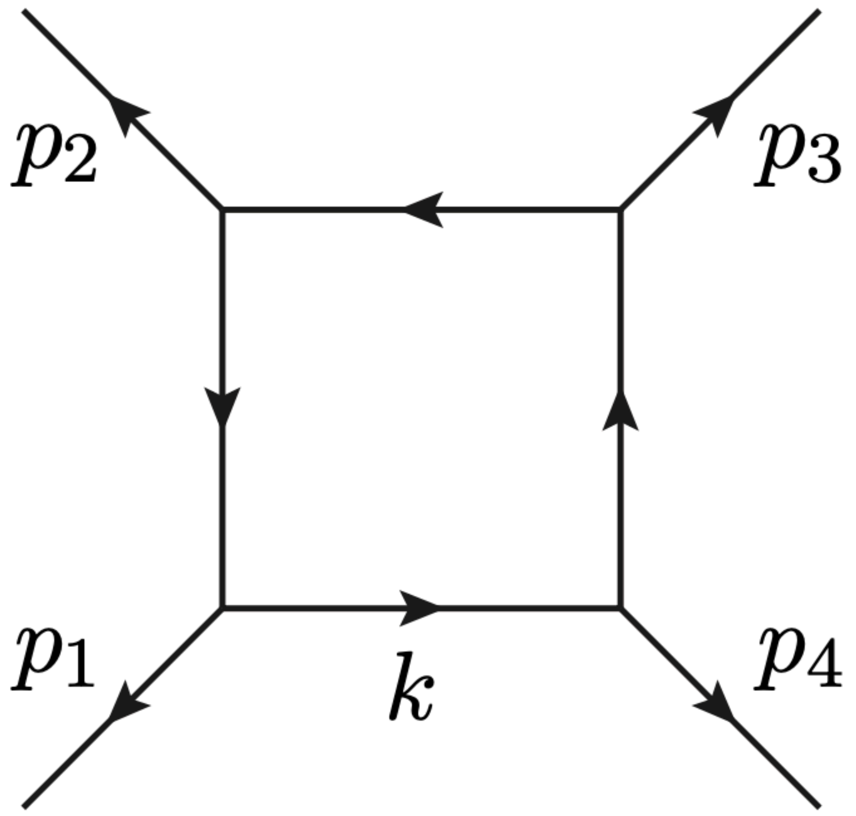
$$I_{\vec{a}}(s,t;\epsilon) = \int \frac{\mathrm{d}^D k}{\mathrm{i}\pi^{D/2}} \frac{1}{D_1^{a_1}\ldots D_4^{a_4}}$$

$$\left\{ I_{\vec{a}}(s,t;\epsilon) \,|\, \forall\, \vec{a} \in \mathbb{Z}^4 \right\}$$

$$D_1 = -\,k^2$$
$$D_2 = -\,(k+p_1)^2$$
$$D_3 = -\,(k+p_1+p_2)^2$$
$$D_4 = -\,(k-p_4)^2$$

Integral families and master integrals

Scalar Feynman integrals with the same propagator structure = **integral family**

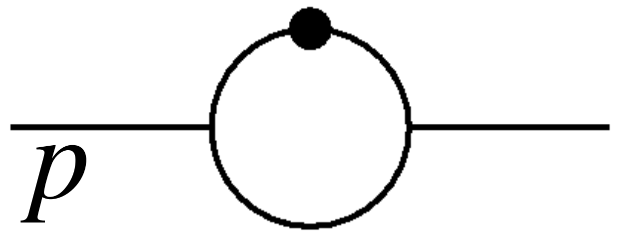


$$\mathrm{I}_{\vec{a}}(s,t;\epsilon) = \int \frac{\mathrm{d}^D k}{\mathrm{i}\pi^{D/2}} \frac{1}{D_1^{a_1}\ldots D_4^{a_4}}$$

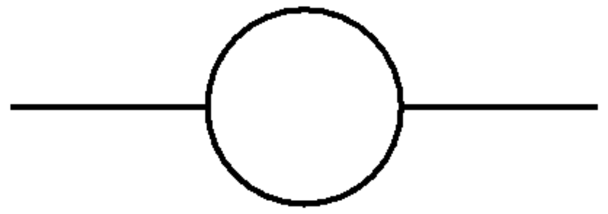
$$D_1 = -\,k^2$$
$$D_2 = -\,(k+p_1)^2$$
$$D_3 = -\,(k+p_1+p_2)^2$$
$$D_4 = -\,(k-p_4)^2$$

$$\left\{\mathrm{I}_{\vec{a}}(s,t;\epsilon)\,|\,\forall\,\vec{a}\in\mathbb{Z}^4\right\}$$

Identities among the $\mathrm{I}_{\vec{a}}$'s



$$= \frac{3-D}{p^2} \times$$

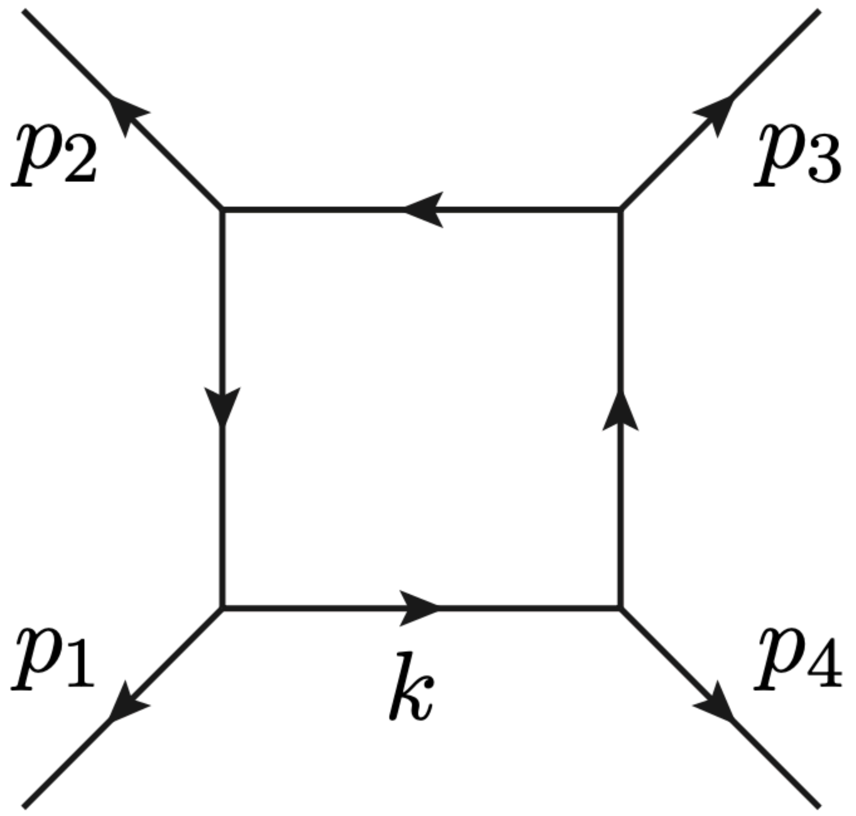


e.g. Integration-By-Parts relations

[Chetyrkin, Tkachov '81; Laporta 2000]

Integral families and master integrals

Scalar Feynman integrals with the same propagator structure = **integral family**

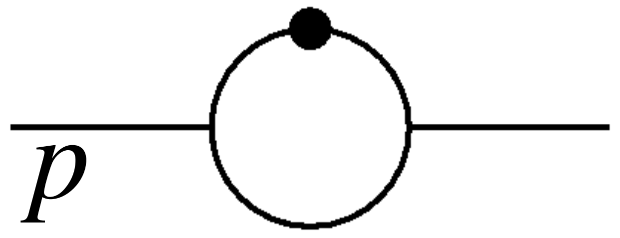


$$\mathrm{I}_{\vec{a}}(s,t;\epsilon) = \int \frac{\mathrm{d}^D k}{\mathrm{i}\pi^{D/2}} \frac{1}{D_1^{a_1}\ldots D_4^{a_4}}$$

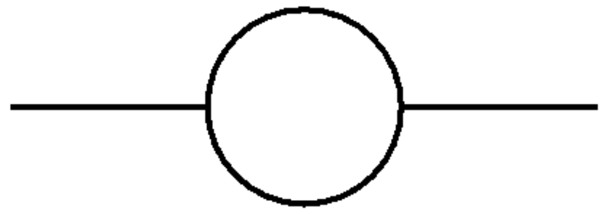
$$\left\{\mathrm{I}_{\vec{a}}(s,t;\epsilon)\,|\,\forall\,\vec{a}\in\mathbb{Z}^4\right\}$$

$$D_1 = -\,k^2$$
$$D_2 = -\,(k+p_1)^2$$
$$D_3 = -\,(k+p_1+p_2)^2$$
$$D_4 = -\,(k-p_4)^2$$

Identities among the $\mathrm{I}_{\vec{a}}$'s



$$= \frac{3-D}{p^2} \times$$



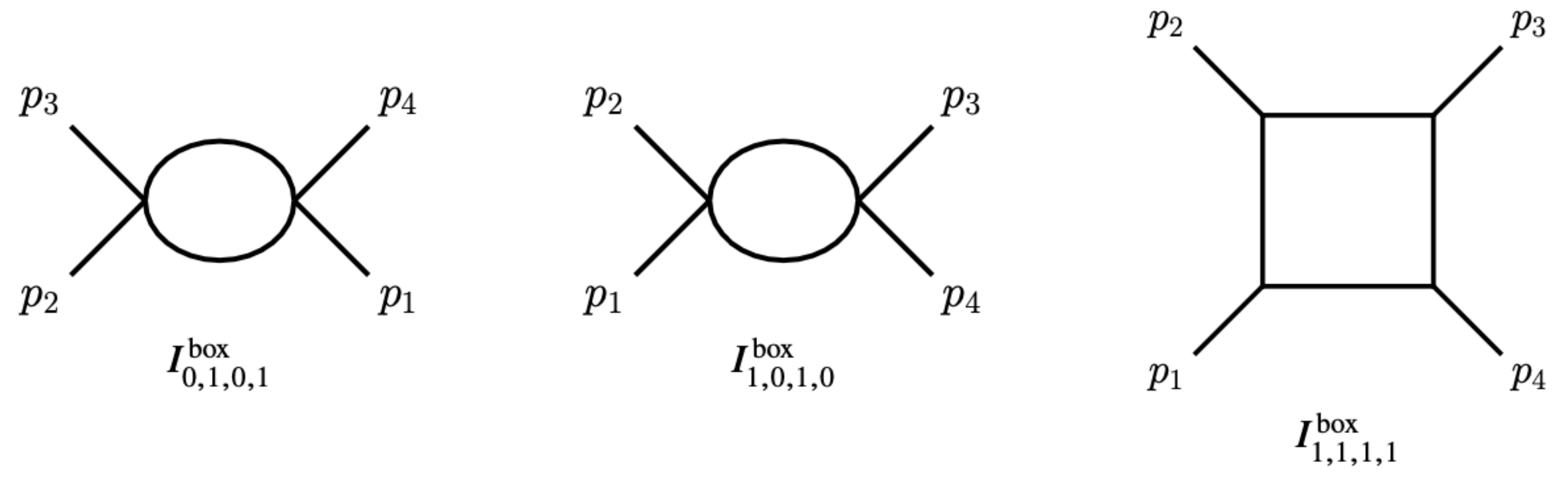
e.g. Integration-By-Parts relations

[Chetyrkin, Tkachov '81; Laporta 2000]

$\Rightarrow$

Finite-dimensional basis:

master integrals $\overrightarrow{\mathrm{F}}(s,t;\epsilon)$



Integrating by differentiating

[Barucchi, Ponzano ’73; Kotikov ’91; Bern, Dixon, Kosower ‘94; Gehrmann, Remiddi 2000]

$$\frac{\partial}{\partial s_{12}} \vec{\mathrm{F}}(s;\epsilon) = \sum_{\vec{a}} c_{\vec{a}}\, \mathrm{I}_{\vec{a}}$$

IBP reduction

$$D = 4 - 2\epsilon$$

$$= A_{s_{12}}(s;\epsilon) \cdot \vec{\mathrm{F}}(s;\epsilon)$$

⇒ System of 1st order linear PDEs for the MIs $\vec{\mathrm{F}}$

- How do we solve it? $\vec{\mathrm{F}}(s;\epsilon) = \sum_{w} \epsilon^{w}\, \vec{\mathrm{F}}^{(w)}(s)$

- What is a “good” choice of MIs?

Solution made simple by the canonical form

[Henn 2013]

Choose MIs such that the DEs take the **canonical form**

$$\mathrm{d}\vec{\mathrm{F}}(s;\epsilon) = \epsilon \; \mathrm{d}\tilde{A}\,(s) \cdot \vec{\mathrm{F}}\,(s;\epsilon)$$

Solution made simple by the canonical form

[Henn 2013]

Choose MIs such that the DEs take the **canonical form**

$$\mathrm{d}\vec{\mathrm{F}}(s;\epsilon) = \epsilon\ \mathrm{d}\tilde{A}(s) \cdot \vec{\mathrm{F}}(s;\epsilon)$$

In the best understood cases (= most of the integrals computed so far):

$$\tilde{A}(s) = \sum_i a_i \log W_i(s)$$

Constant matrices

Letters: algebraic functions of kinematics
e.g. $\{s, t, s+t\}$ for the box

Solution made simple by the canonical form

[Henn 2013]

Choose MIs such that the DEs take the **canonical form**

$$\mathrm{d}\overrightarrow{\mathrm{F}}(s;\epsilon) = \epsilon\ \mathrm{d}\tilde{A}(s) \cdot \overrightarrow{\mathrm{F}}(s;\epsilon)$$

In the best understood cases (= most of the integrals computed so far):

$$\tilde{A}(s) = \sum_i a_i \log W_i(s)$$

Constant matrices

Letters: algebraic functions of kinematics
e.g. $\{s, t, s+t\}$ for the box

Best-case scenario! 🥳

Solution made simple by the canonical form

[Henn 2013]

Choose MIs such that the DEs take the **canonical form**

$$\mathrm{d}\vec{\mathrm{F}}(s;\epsilon) = \epsilon\ \mathrm{d}\tilde{A}(s) \cdot \vec{\mathrm{F}}(s;\epsilon)$$

Many "strategies", but no general algorithm!

In the best understood cases (= most of the integrals computed so far):

$$\tilde{A}(s) = \sum_i a_i \log W_i(s)$$

Constant matrices

Letters: algebraic functions of kinematics
e.g. $\{s, t, s+t\}$ for the box

Best-case scenario! 🥳

Even the best-case scenario is challenging

If the letters $W_i(s)$ are rational ⇒ solution in terms of **multiple polylogarithms**

$$G\left(z_1,\ldots,z_n;x\right) := \int_0^x \frac{\mathrm{d}t_1}{t_1-z_1}\int_0^{t_1}\frac{\mathrm{d}t_2}{t_2-z_2}\ldots\int_0^{t_{n-1}}\frac{\mathrm{d}t_n}{t_n-z_n}$$

1. **Square roots** ruin the party: solution may be MPL, but difficult and not algorithmic

2. MPLs satisfy functional identities ⇒ **Redundant representation**

$$\mathrm{Li}_2\left(z\right)+\frac{1}{2}\log^2(-z)+\mathrm{Li}_2\left(\frac{1}{z}\right)+\frac{\pi^2}{6}=0$$

Simplifications hidden, inefficient evaluation, complicated expressions…

The best-case scenario is not enough

More complicated classes of functions can appear (e.g. elliptic MPLs)

- Obtaining the canonical form is <u>very</u> challenging
- Mathematical technology much less mature

Growing interest for semi-numerical solution based on series expansions

[Moriello 2019]

DiffExp *[Hidding 2020]*, SeaSyde *[Armadillo et al. 2022]*, AMFlow *[Ma, Liu 2022]*

😄 Very flexible (canonical form not required)

🙁 Long evaluation times

Two opposite methods

1. Write the solution in terms of a **basis** of special functions ("pentagon functions")

→ Make the most out of the canonical DEs

2. Train a **neural network** to approximate solution to the DEs

→ Does not rely on a canonical form at all

1. Pentagon functions

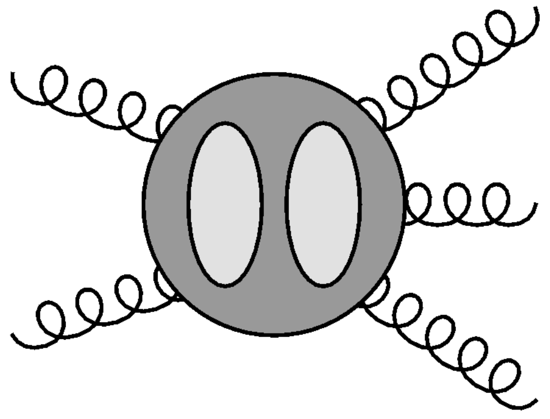
Chicherin, Sotnikov, **SZ** (2110.10111)

Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** (2306.15431)

The “pentagon functions” approach

Very successful for 2-loop 5-point amplitudes

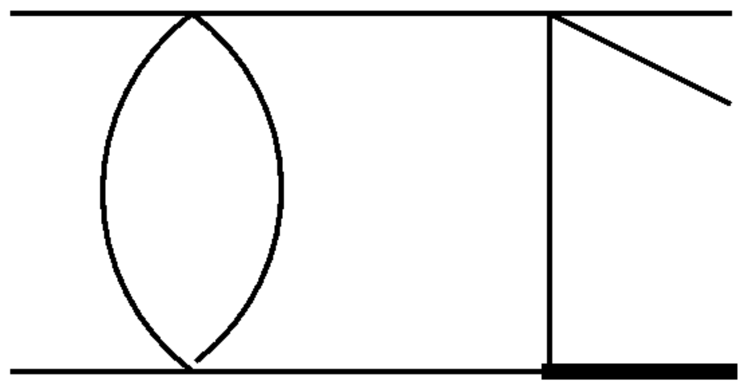
*[Gehrmann, Henn, Lo Presti 2018; Chicherin, Sotnikov 2020; Chicherin, Sotnikov, **SZ** 2021; Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]*



Express MIs in terms of a basis of algebraically **independent** special functions

Algorithmically!

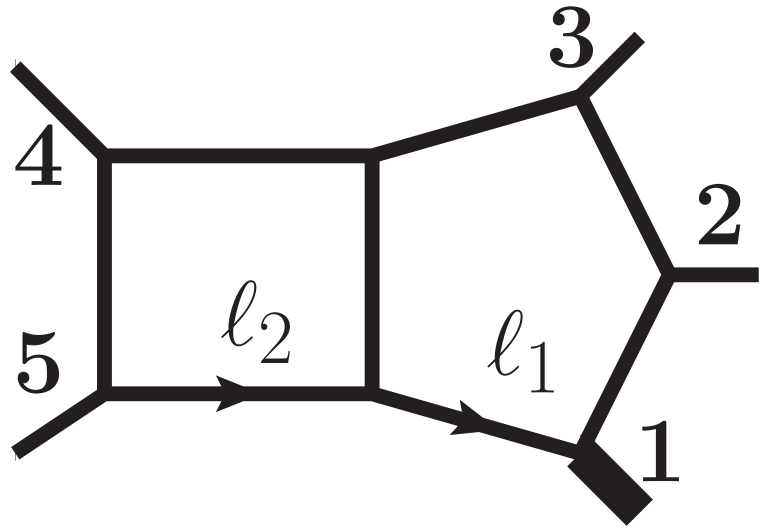
$$\epsilon^3(1-2\epsilon)\sqrt{\Delta_3^{(1)}}\times$$



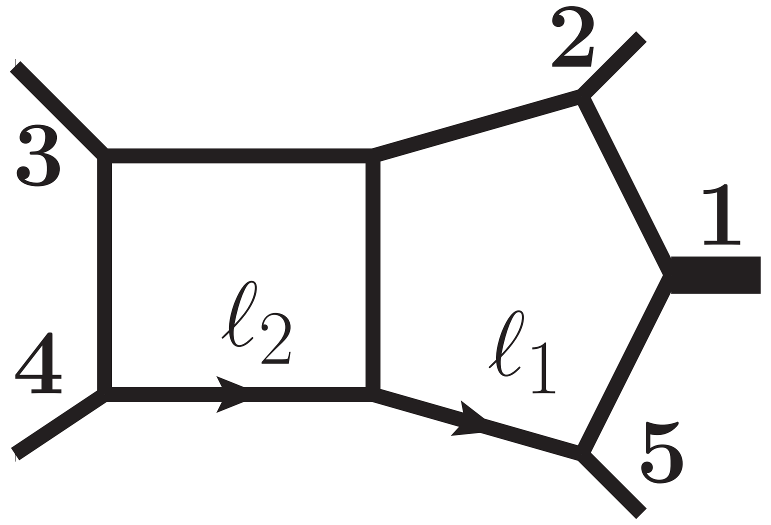
$$= \epsilon^2 f_{23}^{(2)} + \epsilon^3\left[\frac{1}{4}\left(f_1^{(1)} - f_6^{(1)}\right) f_{23}^{(2)} + \frac{1}{2} f_3^{(3)} - \frac{1}{2} f_{29}^{(3)}\right] + \epsilon^4 f_{47}^{(4)} + \mathcal{O}\left(\epsilon^5\right)$$

Efficient numerical evaluation through one-fold integrals

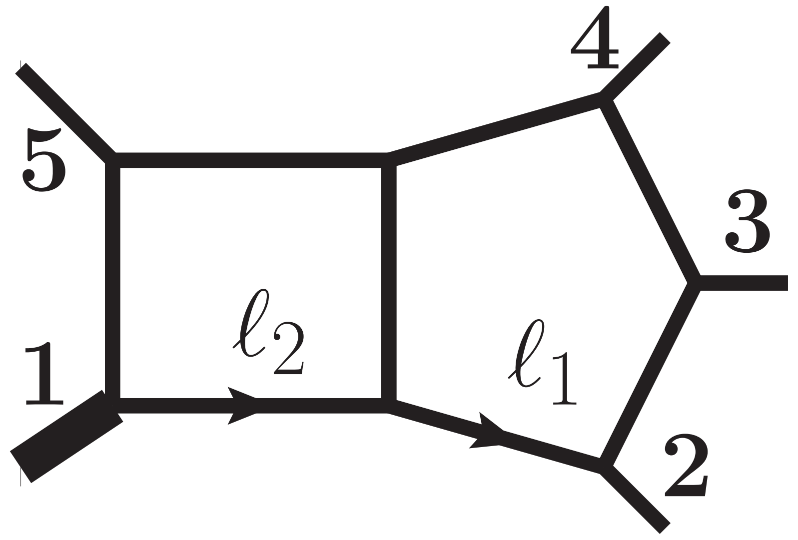
2-loop 5-pt 1-mass master integrals



74

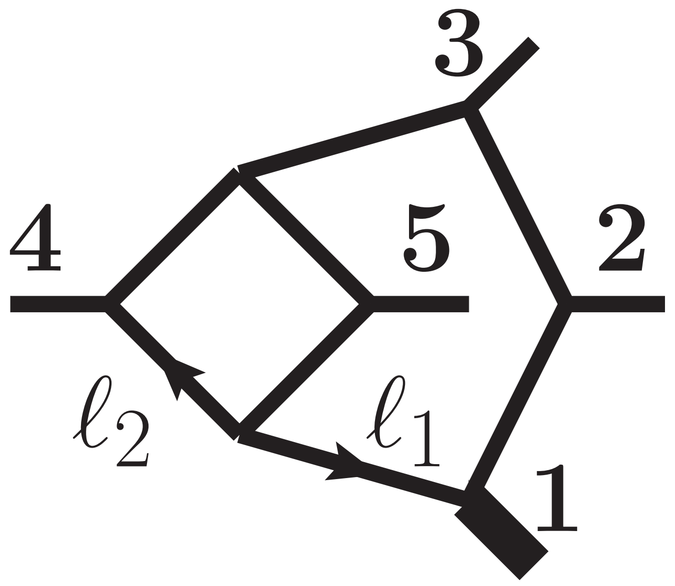


75

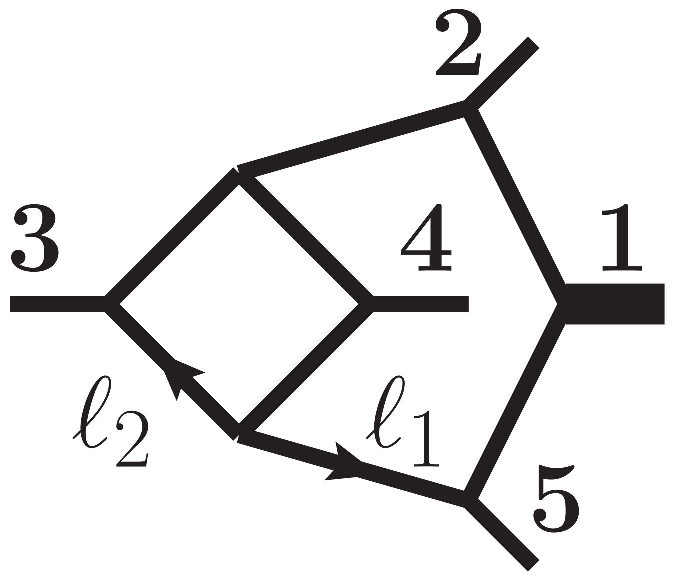


86

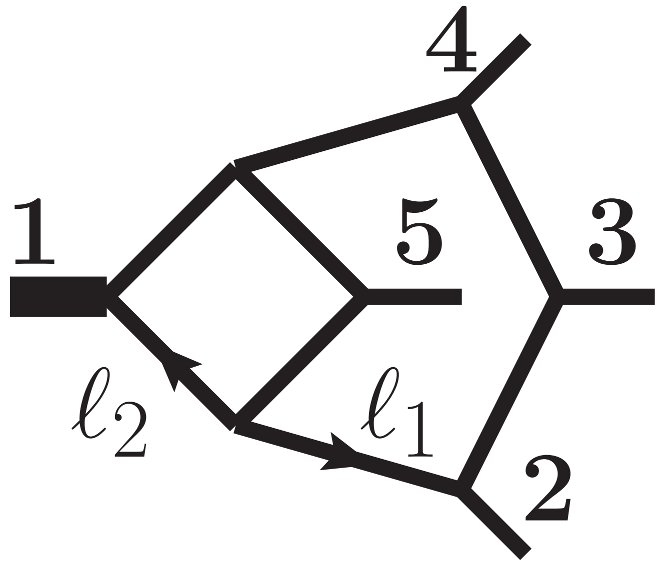
[Abreu, Ita, Moriello, Page, Tschernow, Zeng 2020; Canko, Papadopoulos, Syrrakos 2020; Syrrakos 2020; Chicherin, Sotnikov, **SZ** 2021]



86



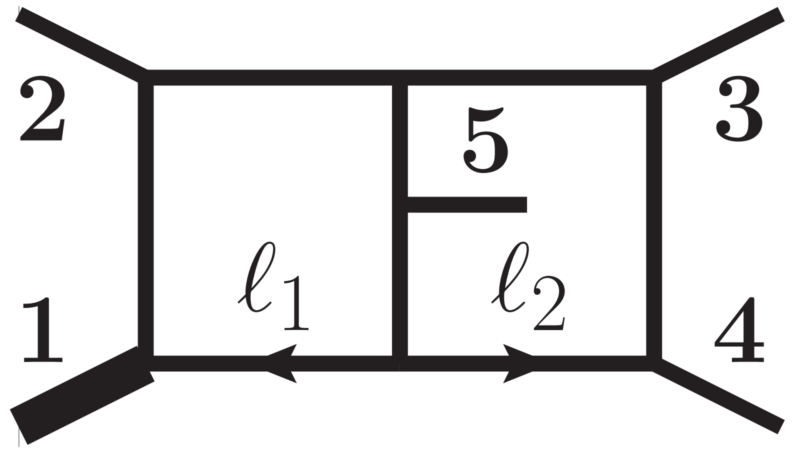
86



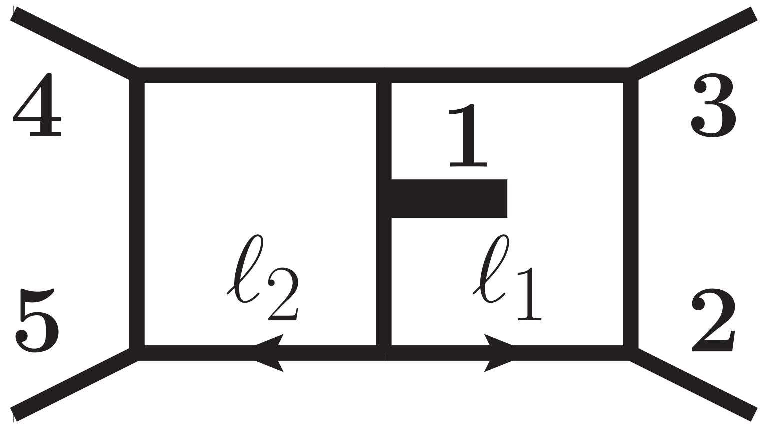
135

[Abreu, Ita, Moriello, Page, Tschernow 2021; Kardos, Papadopoulos, Smirnov, Syrrakos, Wever 2022]

NNLO QCD corrections for
$pp \to H/V + 2\,\text{jets}/\gamma$,
$e^+e^- \to 4\,\text{jets}$



142



179

[Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]

1-mass pentagon alphabet: 204 letters

[Abreu, Ita, Moriello, Page, Tschernow 2021]

127 rational

$$W_1 = p_1^2\,,$$
$$\{W_2,\ldots,W_5\} = \{\sigma(s_{12}) : \sigma \in S_4/S_3[3,4,5]\}\,,$$
$$\{W_6,\ldots,W_{11}\} = \{\sigma(s_{23}) : \sigma \in S_4/(S_2[2,3]\times S_2[4,5])\}\,,$$
$$\{W_{12},\ldots,W_{15}\} = \{\sigma(2\,p_1\cdot p_2) : \sigma \in S_4/S_3[3,4,5]\}\,,$$
$$\{W_{16},\ldots,W_{27}\} = \{\sigma\big(2\,p_2\cdot(p_3+p_4)\big) : \sigma \in S_4/S_2[3,4]\}\,,$$

77 algebraic

$$\{W_{186},\ldots,W_{188}\} = \left\{\sigma\left(\frac{\Omega^{--}\Omega^{++}}{\Omega^{-+}\Omega^{+-}}\right) : \sigma \in S_4/(S_2[2,3]\times S_2[4,5]\times S_2[s_{23},s_{45}])\right\}\,,$$
$$\{W_{189},\ldots,W_{194}\} = \left\{\sigma\left(\frac{\tilde\Omega^{--}\tilde\Omega^{++}}{\tilde\Omega^{-+}\tilde\Omega^{+-}}\right) : \sigma \in S_4/(S_2[3,4]\times S_2[2,5])\right\}\,,$$

where

$$\Omega^{\pm\pm} = s_{12}s_{15} - s_{12}s_{23} - s_{15}s_{45} \pm s_{34}\sqrt{\Delta_3^{(1)}} \pm \sqrt{\Delta_5}\,,$$
$$\tilde\Omega^{\pm\pm} = p_1^2 s_{34} \pm \sqrt{\Delta_5} \pm \sqrt{\Sigma_5^{(1)}}\,,$$

6 variables

$$\Sigma_5 = (s_{12}s_{15} - s_{12}s_{23} - s_{15}s_{45} + s_{34}s_{45} + s_{23}s_{34})^2 - 4s_{23}s_{34}s_{45}(s_{34} - s_{12} - s_{15})$$

10 square roots:

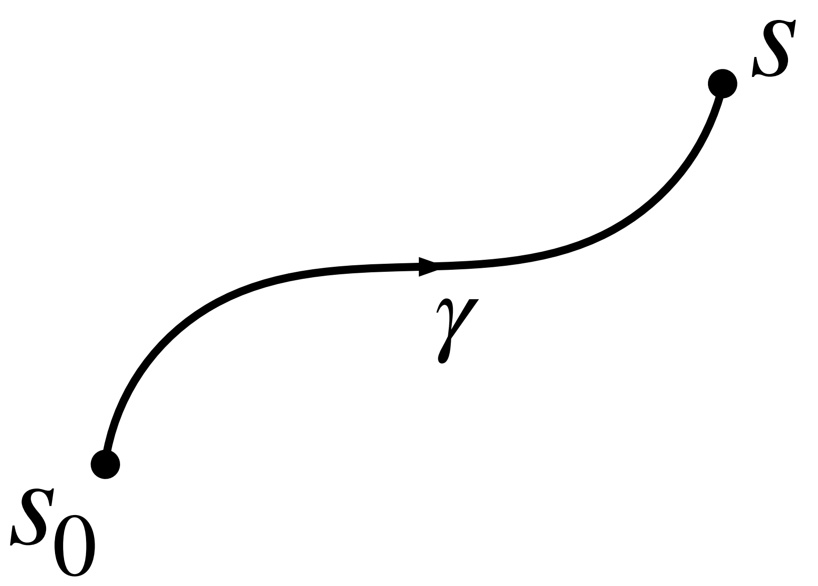
$$\Delta_5 = \det G(p_1,p_2,p_3,p_4)$$
$$= (s_{12}s_{15} - s_{12}s_{23} - p_1^2 s_{34} - s_{15}s_{45} + s_{34}s_{45} + s_{23}s_{34})^2 - 4s_{23}s_{34}s_{45}(p_1^2 - s_{12} - s_{15} + s_{34})\,.$$

Closed under permutations of the massless momenta

Algorithmic approach made necessary by the scale of the problem

Algorithmic construction of the function basis

Chen iterated integrals



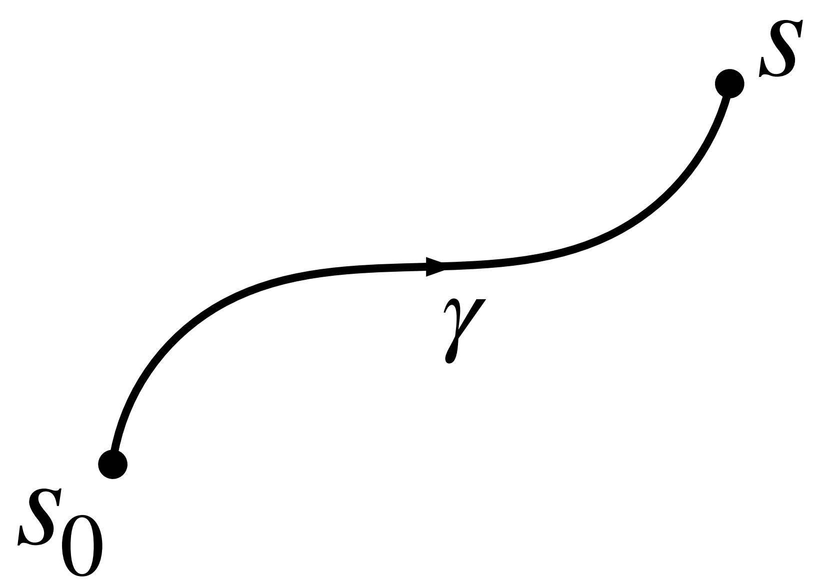
$$\left[w_{i_1}, \ldots, w_{i_n}\right]_{s_0}(s) = \int_\gamma \mathrm{d}\log w_{i_n}(s') \left[w_{i_1}, \ldots, w_{i_{n-1}}\right]_{s_0}(s')$$

n = transcendental weight

All functional relations become manifest in terms of iterated integrals

$$\mathrm{Li}_2(z) + \frac{1}{2}\log^2(-z) + \mathrm{Li}_2\left(\frac{1}{z}\right) + \frac{\pi^2}{6} = 0$$

Chen iterated integrals



$$\left[w_{i_1},\ldots,w_{i_n}\right]_{s_0}(s) = \int_\gamma \mathrm{d}\log w_{i_n}(s')\left[w_{i_1},\ldots,w_{i_{n-1}}\right]_{s_0}(s')$$

n = transcendental weight

All functional relations become manifest in terms of iterated integrals

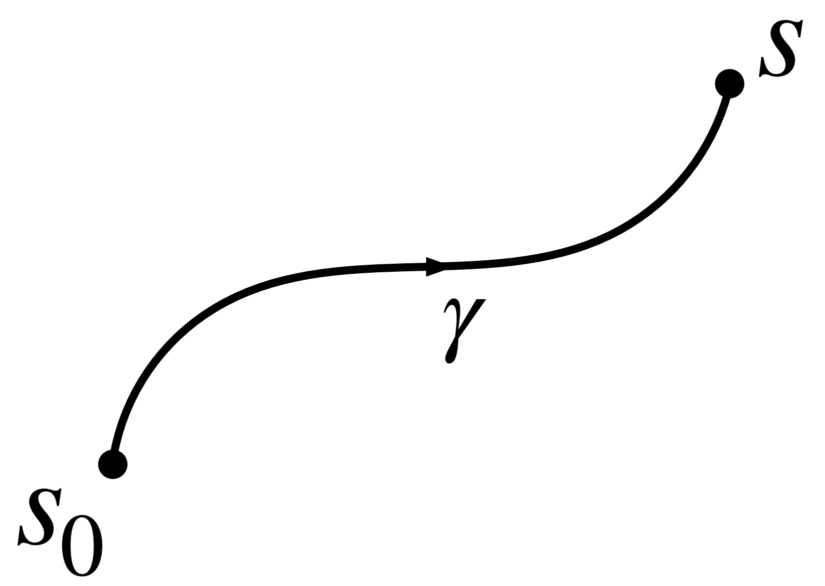
$$\mathrm{Li}_2(z) + \frac{1}{2}\log^2(-z) + \mathrm{Li}_2\left(\frac{1}{z}\right) + \frac{\pi^2}{6} = 0$$

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\frac{1}{2}\log^2(-z) = \left[z, z\right]_{-1}$$

$$\mathrm{Li}_2\left(\frac{1}{z}\right) = \left[1-z, z\right]_{-1} - \left[z, z\right]_{-1} + \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

Chen iterated integrals



$$\left[w_{i_1}, \ldots, w_{i_n}\right]_{s_0}(s) = \int_\gamma \mathrm{d}\log w_{i_n}(s')\, \left[w_{i_1}, \ldots, w_{i_{n-1}}\right]_{s_0}(s')$$

n = transcendental weight

All functional relations become manifest in terms of iterated integrals

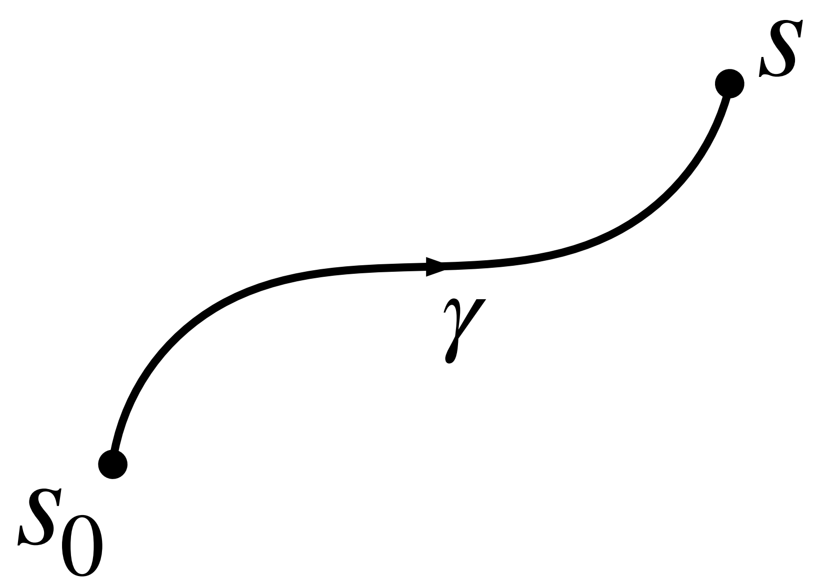
$$\mathrm{Li}_2(z) + \frac{1}{2}\log^2(-z) + \mathrm{Li}_2\left(\frac{1}{z}\right) + \frac{\pi^2}{6} = 0$$

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\,\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\mathrm{Li}_2\left(\frac{1}{z}\right) = \left[1-z, z\right]_{-1} - \left[z, z\right]_{-1} + \log 2\,\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\frac{1}{2}\log^2(-z) = \left[z, z\right]_{-1}$$

Chen iterated integrals



$$\left[w_{i_1}, \ldots, w_{i_n}\right]_{s_0}(s) = \int_\gamma \mathrm{d}\log w_{i_n}(s')\left[w_{i_1}, \ldots, w_{i_{n-1}}\right]_{s_0}(s')$$

n = transcendental weight

All functional relations become manifest in terms of iterated integrals

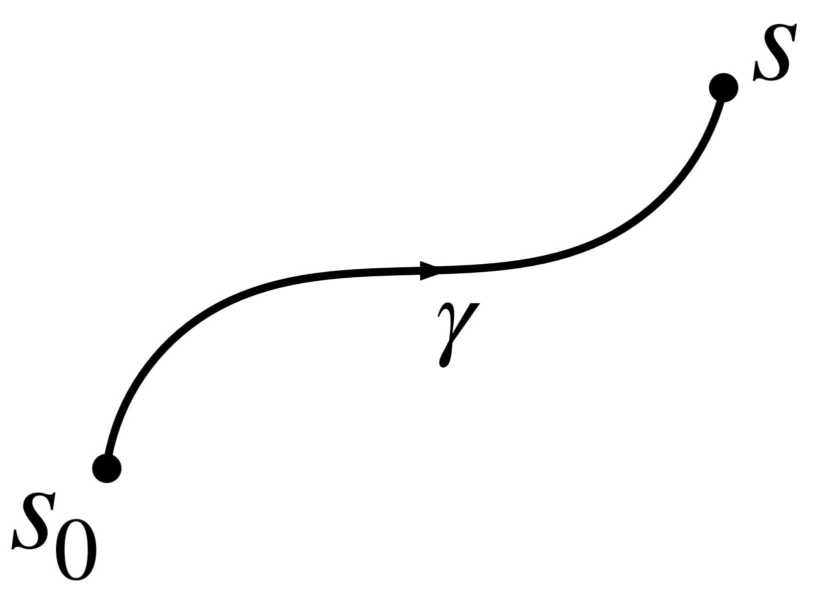
$$\mathrm{Li}_2(z) + \frac{1}{2}\log^2(-z) + \mathrm{Li}_2\left(\frac{1}{z}\right) + \frac{\pi^2}{6} = 0$$

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\mathrm{Li}_2\left(\frac{1}{z}\right) = \left[1-z, z\right]_{-1} - \left[z, z\right]_{-1} + \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\frac{1}{2}\log^2(-z) = \left[z, z\right]_{-1}$$

Chen iterated integrals



$$\left[w_{i_1},\ldots,w_{i_n}\right]_{s_0}(s) = \int_\gamma \mathrm{d}\log w_{i_n}(s')\left[w_{i_1},\ldots,w_{i_{n-1}}\right]_{s_0}(s')$$

n = transcendental weight

All functional relations become manifest in terms of iterated integrals

$$\mathrm{Li}_2(z) + \frac{1}{2}\log^2(-z) + \mathrm{Li}_2\left(\frac{1}{z}\right) + \frac{\pi^2}{6} = 0$$

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\mathrm{Li}_2\left(\frac{1}{z}\right) = \left[1-z, z\right]_{-1} - \left[z, z\right]_{-1} + \log 2\left[z\right]_{-1} - \frac{\pi^2}{12}$$

$$\frac{1}{2}\log^2(-z) = \left[z, z\right]_{-1}$$

The solution to canonical DEs is uniform in the transcendental weight

Solution in terms of iterated integrals can be simply read off from the canonical DEs

$$\mathrm{d}\vec{\mathrm{F}}(s;\epsilon) = \epsilon \; \mathrm{d}\tilde{A}\,(s) \cdot \vec{\mathrm{F}}\,(s;\epsilon)$$

$$\tilde{A}\,(s) = \sum_i a_i \log W_i\,(s)$$

$$\vec{\mathrm{F}}(s;\epsilon) = \frac{1}{\epsilon^{2\ell}} \sum_{w \geq 0} \epsilon^w \, \vec{\mathrm{F}}^{(w)}(s)$$

$\mathbb{Q}$-linear combination of weight-w iterated integrals

Uniform transcendentality

Important properties of iterated integrals

- $\{W_i(s)\}$ multiplicatively independent $\Rightarrow [W_1, \ldots, W_n]$ $\mathbb{Q}$-linearly independent
- No $\mathbb{Q}$-linear relations among iterated integrals with different weight
- Shuffle product:

$$[W_1, W_2]_{s_0} \times [W_3]_{s_0} = [W_1, W_2, W_3]_{s_0} + [W_1, W_3, W_2]_{s_0} + [W_3, W_1, W_2]_{s_0}$$

(weight w_1) $\times$ (weight w_2) $=$ $\mathbb{Q}$-linear combination of weight $(w_1 + w_2)$

Extract function basis from MI coefficients

$$\overrightarrow{\mathrm{MI}}(s,\epsilon) = \sum_{w\geq 0} \epsilon^w\, \overrightarrow{\mathrm{MI}}^{(w)}(s)$$

Written in terms of Chen iterated integrals

Up to required order (here, $w = 4$)

$$\left\{\mathrm{MI}_i^{(1)}\right\} \longrightarrow \left\{f_k^{(1)}\right\}$$

$$\left\{\mathrm{MI}_i^{(2)}\right\} \cup \left\{f_i^{(1)} \times f_j^{(1)}\right\} \longrightarrow \left\{f_k^{(2)}\right\}$$

$$\left\{\mathrm{MI}_i^{(3)}\right\} \cup \left\{f_i^{(2)} \times f_j^{(1)}\right\} \cup \left\{f_i^{(1)} \times f_j^{(1)} \times f_k^{(1)}\right\} \longrightarrow \left\{f_k^{(3)}\right\}$$

Solution of a linear system of equations

$$\left\{f_i^{(w)}\right\}_{i,w=1,\ldots,4}$$

Algebraically independent

Irreducible

Linear algebra only 👍

Need to know relations among boundary values

We only know $\overrightarrow{\mathrm{MI}}^{(w)}(s_0)$ numerically

Previous approach: high-precision evaluation of MPLs + PSLQ algorithm

[Ferguson, Bailey '92]

$$\mathrm{MI}_1^{(2)}(s_0) = -\,1.644934067\ldots$$

$$\mathrm{MI}_2^{(2)}(s_0) = 0.4060916335\ldots$$

$$\mathrm{MI}_3^{(2)}(s_0) = 1.436746367\ldots$$

→

$$3\,\mathrm{MI}_1^{(2)}(s_0) + 4\,\mathrm{MI}_2^{(2)}(s_0) - 2\,\mathrm{MI}_3^{(2)}(s_0) = 0$$

- Very heavy from computational point of view (e.g. ~3000-digit precision in *[Chicherin, Sotnikov, **SZ** 2021]*) 😕
- Relies on MPL representation ☹️

The new algorithm

*[Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]*

1. Select MI coefficients for the basis at **symbol** level $\left\{f_i^{(w)}\right\}$ *[Goncharov, Spradlin, Vergu, Volovich 2010]*

Symbol = iterated integral stripped of boundary information

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12} \longrightarrow \mathcal{S}\left[\mathrm{Li}_2(z)\right] = -\left[1-z, z\right]$$

The new algorithm

*[Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]*

1. Select MI coefficients for the basis at **symbol** level $\left\{f_i^{(w)}\right\}$ *[Goncharov, Spradlin, Vergu, Volovich 2010]*

Symbol = iterated integral stripped of boundary information

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12} \longrightarrow \mathcal{S}\left[\mathrm{Li}_2(z)\right] = -\left[1-z, z\right]$$

2. **Ansatz**: all MI coefficients are polynomials in $\left\{f_i^{(w)}\right\}$ + ζ_2 and ζ_3 (up to weight 4)

$$\mathrm{MI}^{(2)} = \sum_i \alpha_i f_i^{(2)} + \sum_{i\leq j} \beta_{ij} f_i^{(1)} f_i^{(1)} + \gamma\,\zeta_2 \qquad \alpha_i, \beta_{ij}, \gamma \in \mathbb{Q}$$

The new algorithm

*[Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]*

1. Select MI coefficients for the basis at **symbol** level $\left\{f_i^{(w)}\right\}$ *[Goncharov, Spradlin, Vergu, Volovich 2010]*

Symbol = iterated integral stripped of boundary information

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2\left[z\right]_{-1} - \frac{\pi^2}{12} \longrightarrow \mathcal{S}\left[\mathrm{Li}_2(z)\right] = -\left[1-z, z\right]$$

2. **Ansatz**: all MI coefficients are polynomials in $\left\{f_i^{(w)}\right\}$ + ζ_2 and ζ_3 (up to weight 4)

$$\mathrm{MI}^{(2)} = \sum_i \alpha_i f_i^{(2)} + \sum_{i\leq j} \beta_{ij} f_i^{(1)} f_i^{(1)} + \gamma\,\zeta_2 \qquad \alpha_i, \beta_{ij}, \gamma \in \mathbb{Q}$$

Fixed by symbol-level analysis

The new algorithm

*[Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** 2023]*

1. Select MI coefficients for the basis at **symbol** level $\left\{f_i^{(w)}\right\}$ *[Goncharov, Spradlin, Vergu, Volovich 2010]*

Symbol = iterated integral stripped of boundary information

$$\mathrm{Li}_2(z) = -\left[1-z, z\right]_{-1} - \log 2 \left[z\right]_{-1} - \frac{\pi^2}{12} \longrightarrow \mathcal{S}\left[\mathrm{Li}_2(z)\right] = -\left[1-z, z\right]$$

2. **Ansatz**: all MI coefficients are polynomials in $\left\{f_i^{(w)}\right\} + \zeta_2$ and ζ_3 (up to weight 4)

$$\mathrm{MI}^{(2)} = \sum_i \alpha_i f_i^{(2)} + \sum_{i \leq j} \beta_{ij} f_i^{(1)} f_i^{(1)} + \gamma \zeta_2 \qquad \alpha_i, \beta_{ij}, \gamma \in \mathbb{Q}$$

Fixed by symbol-level analysis (α_i, β_{ij})

Fixed by evaluation at s_0 + rationalisation (γ)

Summary of the algorithm

Input:

- canonical DEs
- numerical boundary values $\{\mathrm{MI}_i^{(w)}(s_0)\}$

Only needed at the accuracy required for the evaluation (~70 digits)

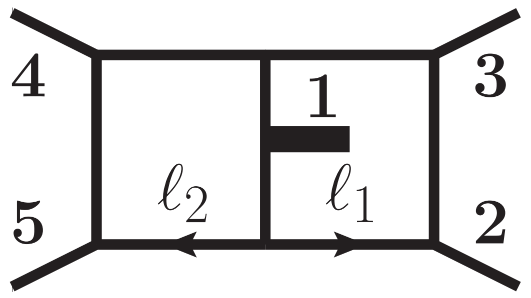
Easy to obtain using **AMFlow**

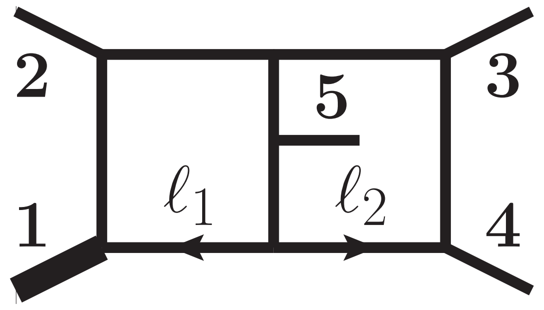
[Liu, Ma 2022]

Output:

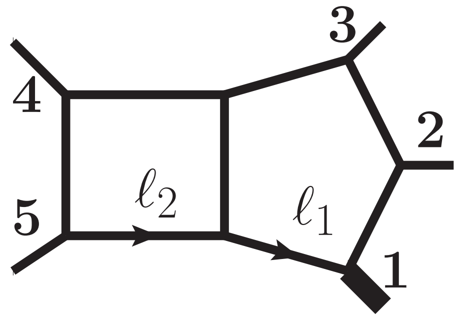
- function basis $\{f_i^{(w)}\}$ (written in terms of iterated integrals)
- relations among the boundary values
- expression of all MI coefficients as polynomials in $\{f_i^{(w)}\}$ and ζ values

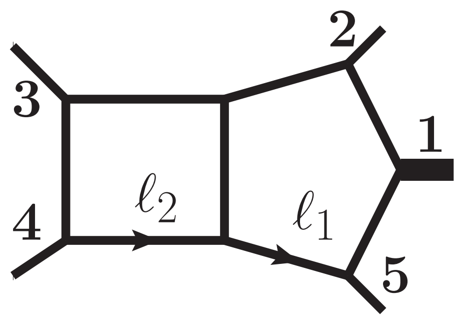
One-mass pentagon functions

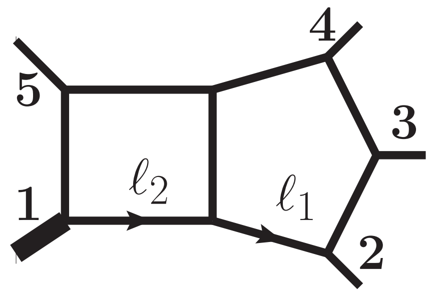




All 1-mass 2-loop 5-pt integrals (2304 independent MIs)



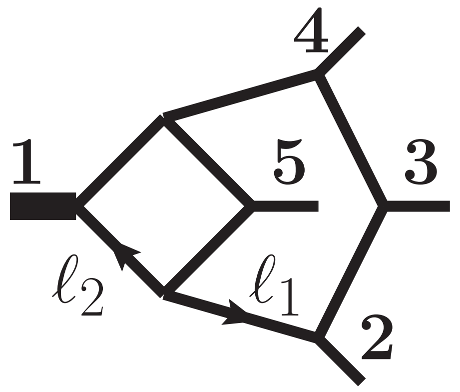


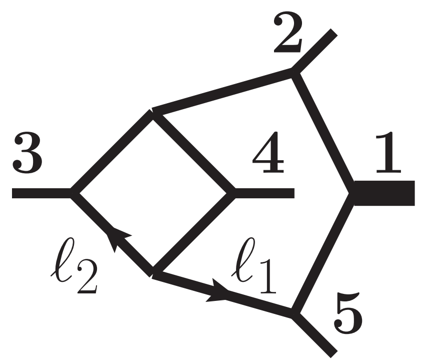


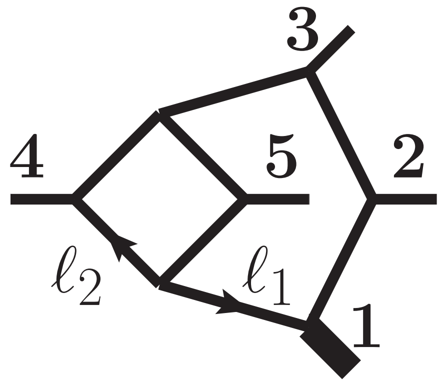
weight	P ∪ PB	HB	DP	Total
1	11	0	0	11
2	25	10	0	35
3	145	72	0	217
4	675	305	48	1028

Functions chosen to highlight analytic properties

E.g. letters expected to drop out, singularities… are isolated in the minimal number of functions







All 4! permutations of external massless legs → everything that is needed for any amplitude of this kind

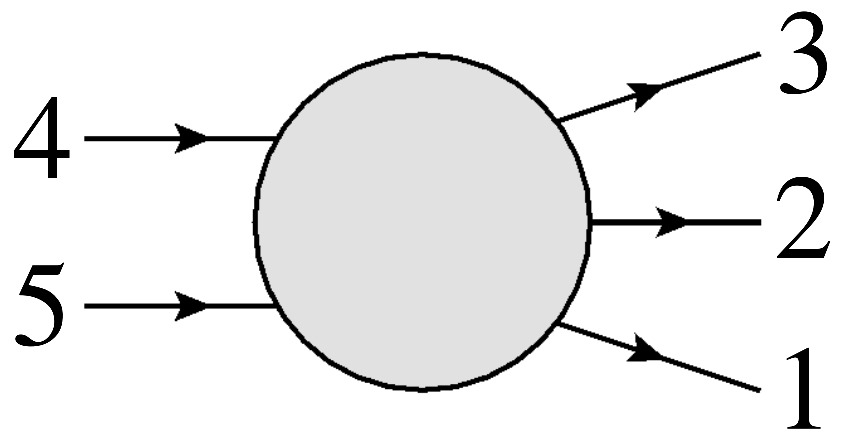
Efficient numerical evaluation

Logs and dilogarithms up to weight 2

Explicit expressions by fitting ansätze

$$f^{(1)} \sim \log + \tau^{(1)}$$
$$f^{(2)} \sim \mathrm{Li}_2 + \log^2 + \tau^{(1)} \log + \tau^{(2)}$$

Arguments guessed *[Duhr, Gangl, Rhodes 2011]* and chosen s.t. functions are well defined in a physical scattering region (s_{45} channel)



$$f_2^{(1)} = \log\left(-s_{34}\right)$$
$$f_2^{(2)} = \mathrm{Li}_2\left(\frac{s_{14}}{p_1^2}\right) + \log\left(-\frac{s_{14}}{p_1^2}\right)\log\left(1-\frac{s_{14}}{p_1^2}\right) + i\pi\log\left(s_{15}-s_{23}+s_{45}\right) - i\pi\log\left(p_1^2\right)$$

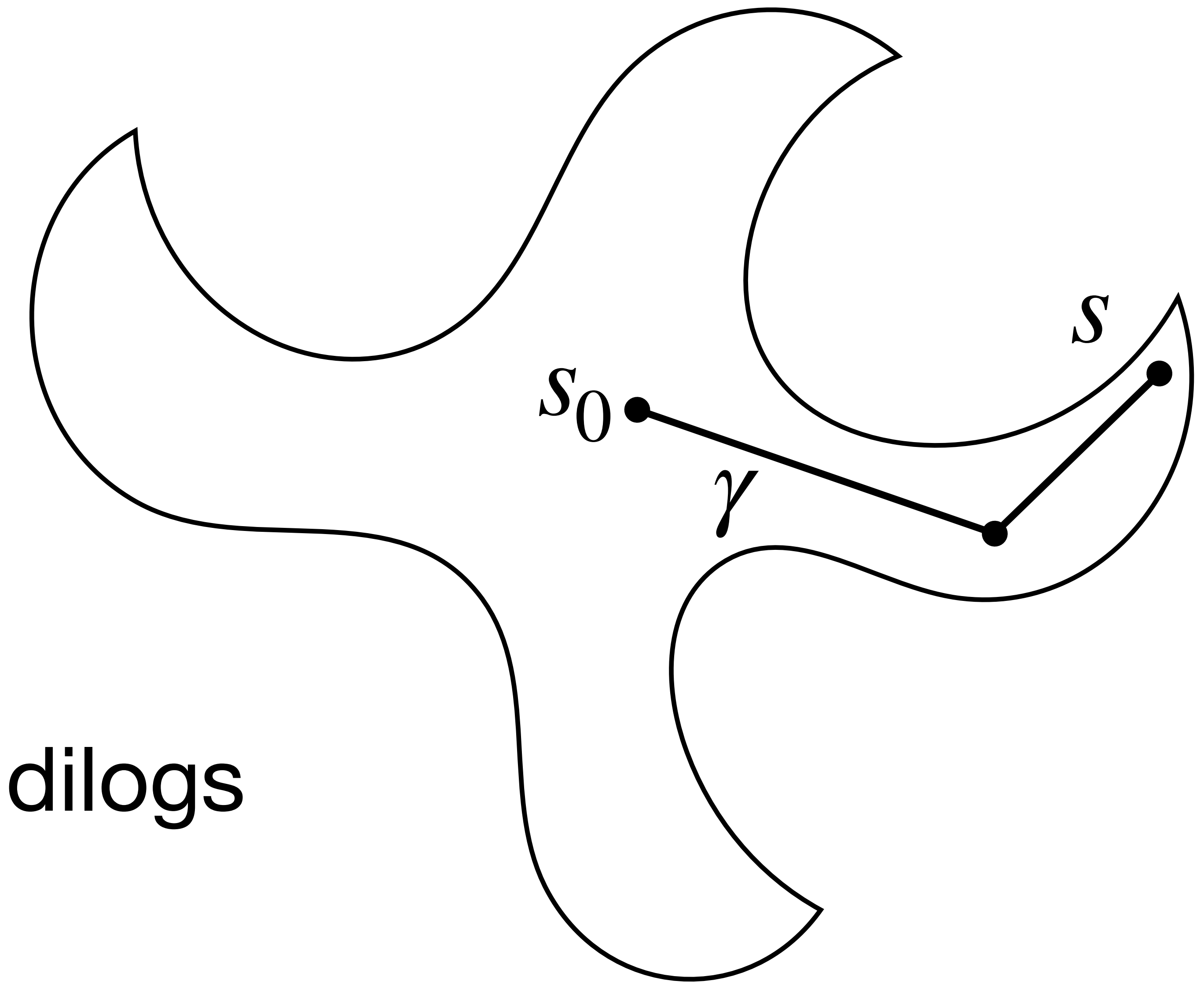
Can be evaluated numerically straightforwardly

One-fold integrals at weight 3 and 4

Path $\gamma : [0,1] \to s$ entirely within the physical region

$$[W_{i_1}, W_{i_2}, W_{i_3}]_{s_0}(s) = \int_0^1 \mathrm{d}t \, \frac{\mathrm{d}\log W_{i_3}\left(s'(t)\right)}{\mathrm{d}t} \left[W_{i_1}, W_{i_2}\right]_{s_0}\left(s'(t)\right)$$

Logs and dilogs



s_{45} channel

Through integration by parts *[Caron-Huot, Henn 2014]*

$$f^{(4)} \sim \int_0^1 \mathrm{d}t \, \log \times \frac{\mathrm{d}\log}{\mathrm{d}t} \times f^{(2)}$$

No analytic continuation required!

Numerical integration implemented in C++ library **PentagonFunctions++**

⇒ **All** functions evaluated in ~3 s (double precision, single core)!

Pentagon functions allowed for efficient amplitude computation

Massless pentagon functions
[Chicherin, Sotnikov 2020]

- 3γ *[Abreu, Page, Pascual, Sotnikov 2020; Chawdhry, Czakon, Mitov, Poncelet 2021; Abreu, De Laurentis, Ita, Klinkert, Page, Sotnikov, 2023]*
- $2\gamma + \mathrm{j}$ *[Agarwal, Buccioni, von Manteuffel, Tancredi 2021; Chawdhry, Czakon, Mitov, Poncelet 2021; Badger, Brönnum-Hansen, Chicherin, Gehrmann, Hartanto, Henn, Marcoli, Moodie, Peraro, **SZ** 2021]*
- $3\mathrm{j}$ *[Abreu, Febres-Cordero, Ita, Page, Sotnikov 2021; De Laurentis, Ita, Klinkert, Sotnikov 2023; Agarwal, Buccioni, Devoto, Gambuti, von Manteuffel, Tancredi 2023]*
- $\gamma + 2\mathrm{j}$ *[Badger, Czakon, Bayu Hartanto, Moodie, Peraro, Poncelet, **SZ** 2023]*

1-mass pentagon functions (planar)
*[Chicherin, Sotnikov, **SZ** 2021]*

- $\mathrm{W} + \mathrm{b}\bar{\mathrm{b}}$ (planar) *[Badger, Bayu Hartanto, **SZ** 2021; Bayu Hartanto, Poncelet, Popescu, **SZ** 2022]*
- $\mathrm{W} + 2\mathrm{j}$ (planar) *[Abreu, Febres Cordero, Ita, Klinkert, Page, Sotnikov 2022]*
- $\mathrm{H} + \mathrm{b}\bar{\mathrm{b}}$ (planar) *[Badger, Bayu Hartanto, Kryś, **SZ** 2021]*
- $\mathrm{W} + \gamma + \mathrm{j}$ (planar) *[Badger, Bayu Hartanto, Kryś, **SZ** 2022]*

Ready for deployment in NNLO QCD phenomenology

Leading colour @ 2 loops

$pp \to 3\gamma$ *[Kallweit, Sotnikov, Wiesemann 2020; Chawdhry, Czakon, Mitov, Poncelet 2020]*

$pp \to 2\gamma + j$ *[Chawdhry, Czakon, Mitov, Poncelet 2021; Badger, Gehrmann, Marcoli, Moodie 2021]*

$pp \to 3j$ *[Czakon, Mitov, Poncelet 2021; Chen, Gehrmann, Glover, Huss, Marcoli 2022]*

$pp \to W + b\bar{b}$ *[Bayu Hartanto, Poncelet, Popescu, **SZ** 2022; Buonocore, Devoto, Kallweit, Mazzitelli, Rottoli, Savoini 2023]*

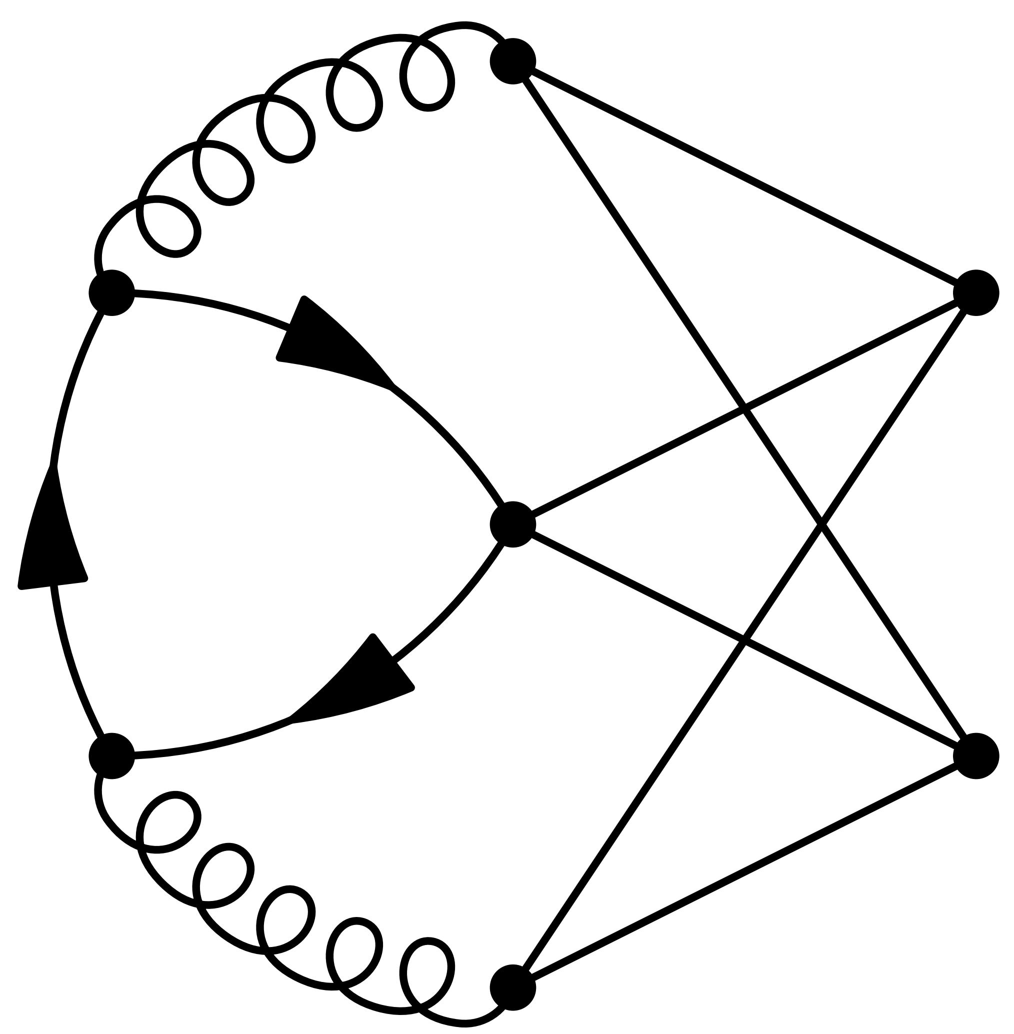
$pp \to \gamma + 2j$ *[Badger, Czakon, Bayu Hartanto, Moodie, Peraro, Poncelet, **SZ** 2023]*

The pentagon functions meet the demands of phenomenological applications

What if we don’t have canonical DEs?

2. Physics-informed deep learning

Calisto, Moodie, **SZ** (23XX.XXXXX)



Logo by Ryan Moodie

Neural networks are universal function approximators

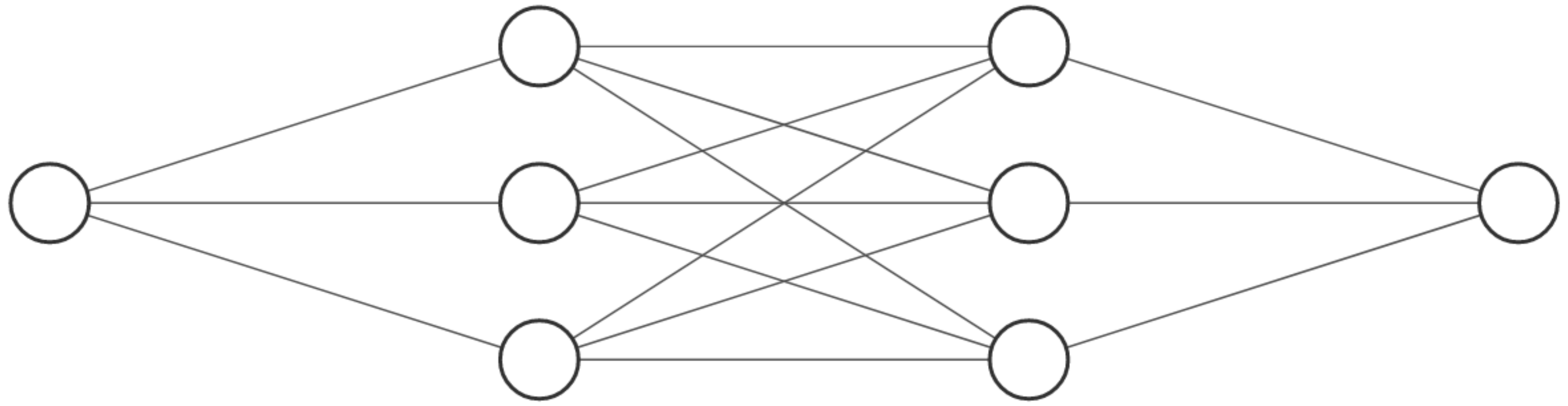
[Hornik, Stinchcombe, White '89]

Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$

Neural networks are universal function approximators

[Hornik, Stinchcombe, White ’89]

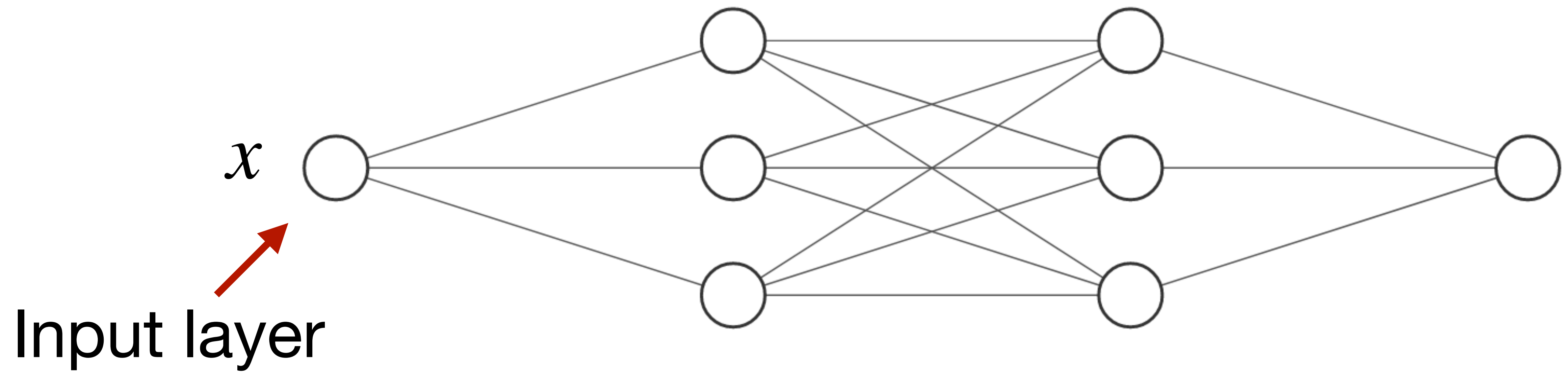
Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Neural networks are universal function approximators

[Hornik, Stinchcombe, White ’89]

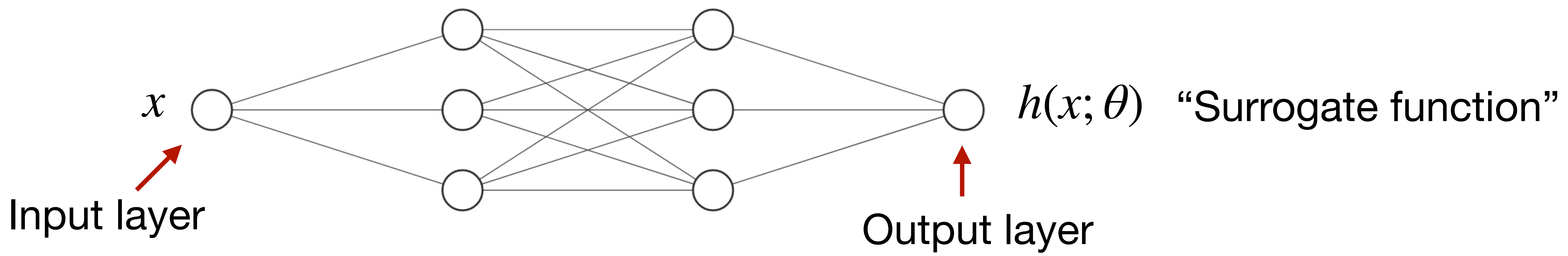
Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Neural networks are universal function approximators

[Hornik, Stinchcombe, White '89]

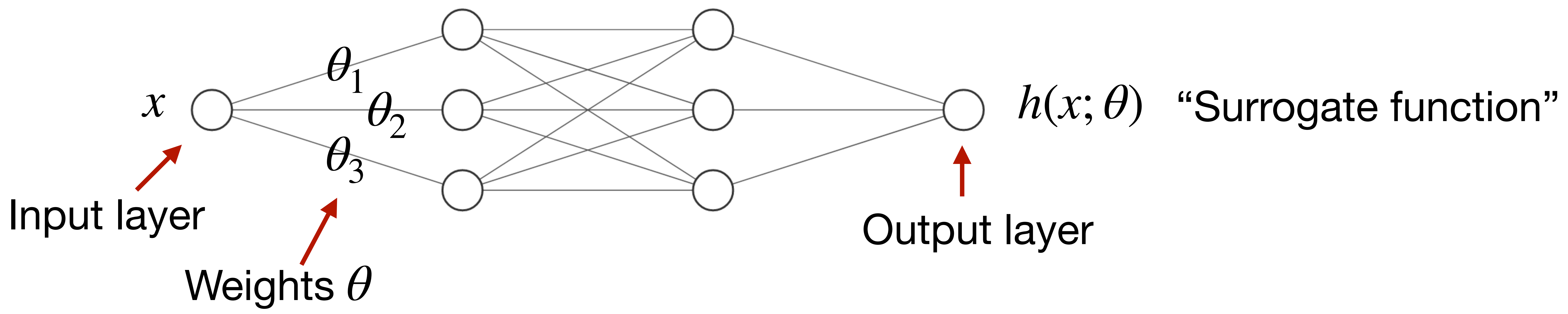
Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Neural networks are universal function approximators

[Hornik, Stinchcombe, White '89]

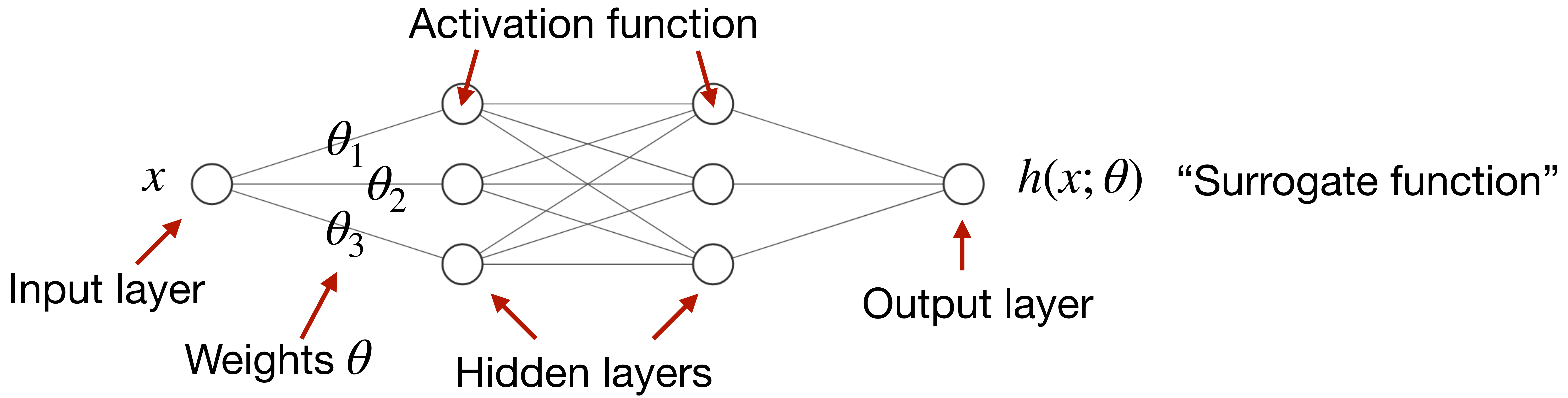
Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Neural networks are universal function approximators

[Hornik, Stinchcombe, White '89]

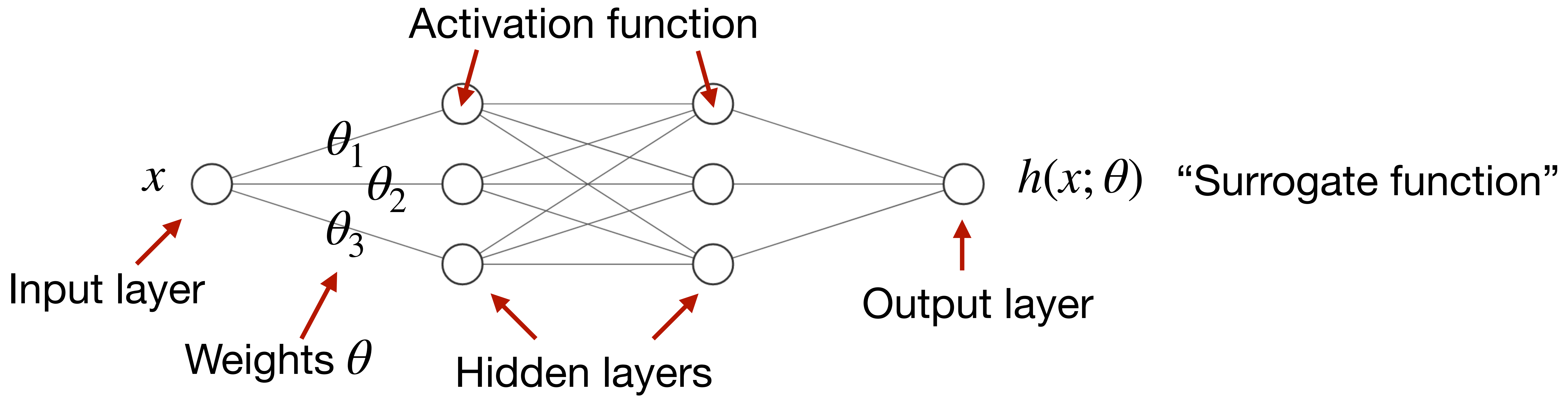
Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Neural networks are universal function approximators

[Hornik, Stinchcombe, White '89]

Typical problem: approximate function $f(x)$ from large dataset of values $f(x_i)$



Optimisation problem: find weights θ such that a **loss function** is minimised

$$L(\mathrm{D};\theta) = \frac{1}{N}\sum_{i=1}^{N}\left[f(x_i) - h(x_i;\theta)\right]^2$$

We don't have a large dataset…

What we have:

- Small dataset of values (at least 1), obtained numerically in other ways

  E.g. AMFlow *[Liu, Ma 2022]* → Expensive evaluation, but very flexible

- Differential equations: $$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = A(x)\,f(x)$$

Physics-informed deep learning

[Raissi, Perdikaris, Karniadakis 2017]

💡 Idea: include the DEs in the loss function

$$L(\mathrm{D};\theta) = \sum_i \left[h(x_i;\theta) - f(x_i)\right]^2 + \sum_j \left[\left.\frac{\mathrm{d}h(x;\theta)}{\mathrm{d}x}\right|_{x=x_j} - A(x_j)\,h(x_j;\theta)\right]^2$$

Small "boundary" dataset

Infinite dimensional "DE" dataset

Derivatives of the NN computed with automatic differentiation [Griewank, Walther 2008]

Input: few boundary values + the analytic DEs

The canonical form of the DEs is not needed

We make mild assumptions to simplify the problem:

$$\frac{\partial}{\partial v_i} \overrightarrow{F}(\vec{v}; \epsilon) = A_{v_i}(\vec{v}; \epsilon) \cdot \overrightarrow{F}(\vec{v}; \epsilon) \quad \forall\, i = 1, \ldots, n_v$$

$\vec{v}$: kinematic variables

The canonical form of the DEs is not needed

We make mild assumptions to simplify the problem:

$$\frac{\partial}{\partial v_i}\overrightarrow{F}(\vec{v};\epsilon) = A_{v_i}(\vec{v};\epsilon)\cdot\overrightarrow{F}(\vec{v};\epsilon) \quad \forall\, i = 1,\ldots,n_v$$

$\vec{v}$: kinematic variables

1. The matrices $A_{v_i}(\vec{v};\epsilon)$ are rational functions ⇒ Separate Re/Im parts, only deal with real numbers

The canonical form of the DEs is not needed

We make mild assumptions to simplify the problem:

$$\frac{\partial}{\partial v_i}\overrightarrow{F}(\vec{v};\epsilon) = A_{v_i}(\vec{v};\epsilon)\cdot\overrightarrow{F}(\vec{v};\epsilon) \quad \forall\, i = 1,\ldots,n_v$$

$\vec{v}$: kinematic variables

1. The matrices $A_{v_i}(\vec{v};\epsilon)$ are rational functions ⇒ Separate Re/Im parts, only deal with real numbers

2. The matrices $A_{v_i}(\vec{v};\epsilon)$ are finite at $\epsilon = 0$,

$$A_{v_i}(\vec{v};\epsilon) = \sum_{k=0}^{k_{\max}} \epsilon^k\, A_{v_i}^{(k)}(\vec{v})$$

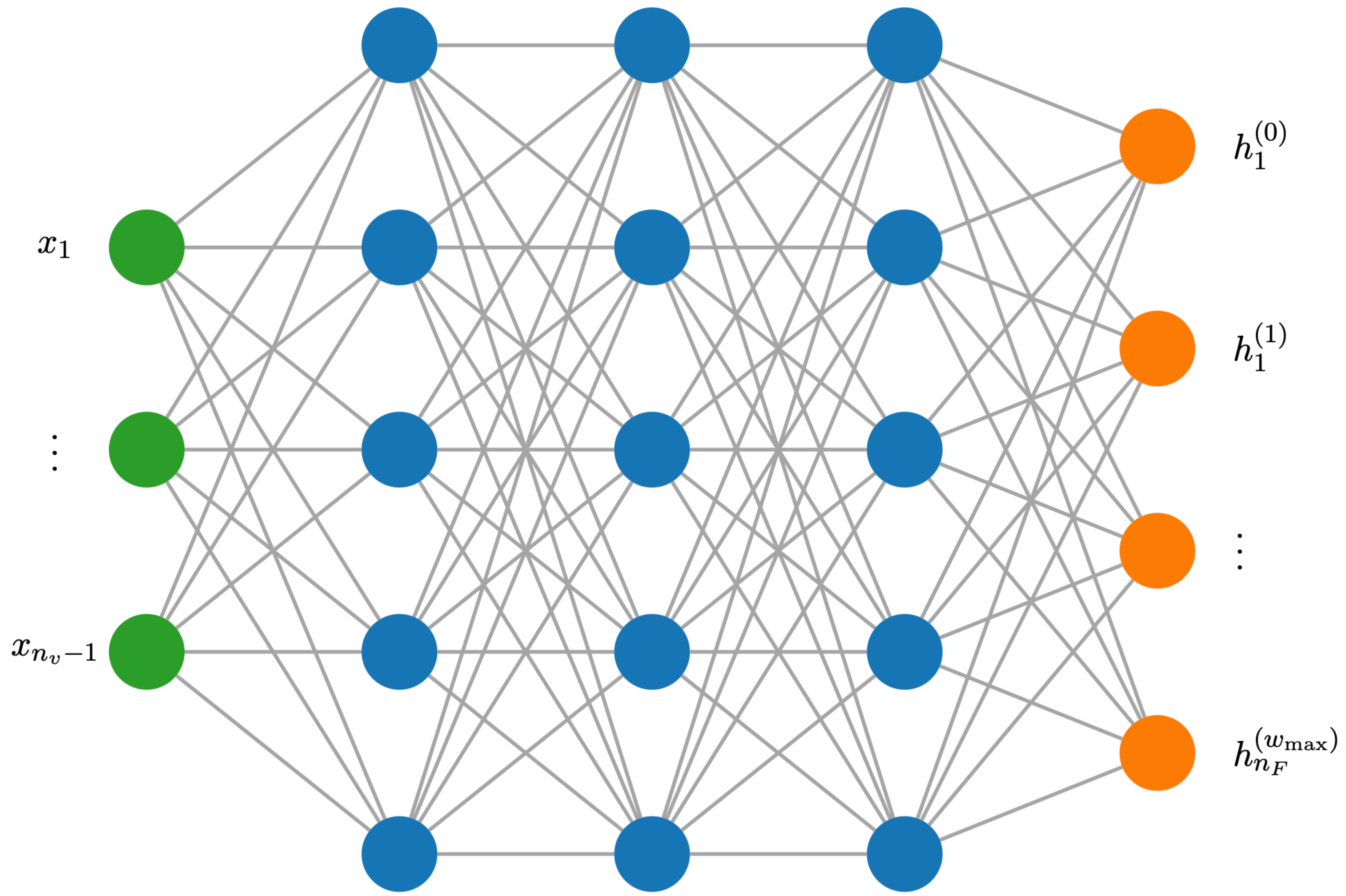
⇒ Simplifies the ϵ expansion of the solution

$$\overrightarrow{F}(\vec{v};\epsilon) = \epsilon^{w*}\sum_{w=0}^{w_{\max}} \epsilon^w\, \overrightarrow{F}^{(w)}(\vec{v})$$

Architecture

PyTorch

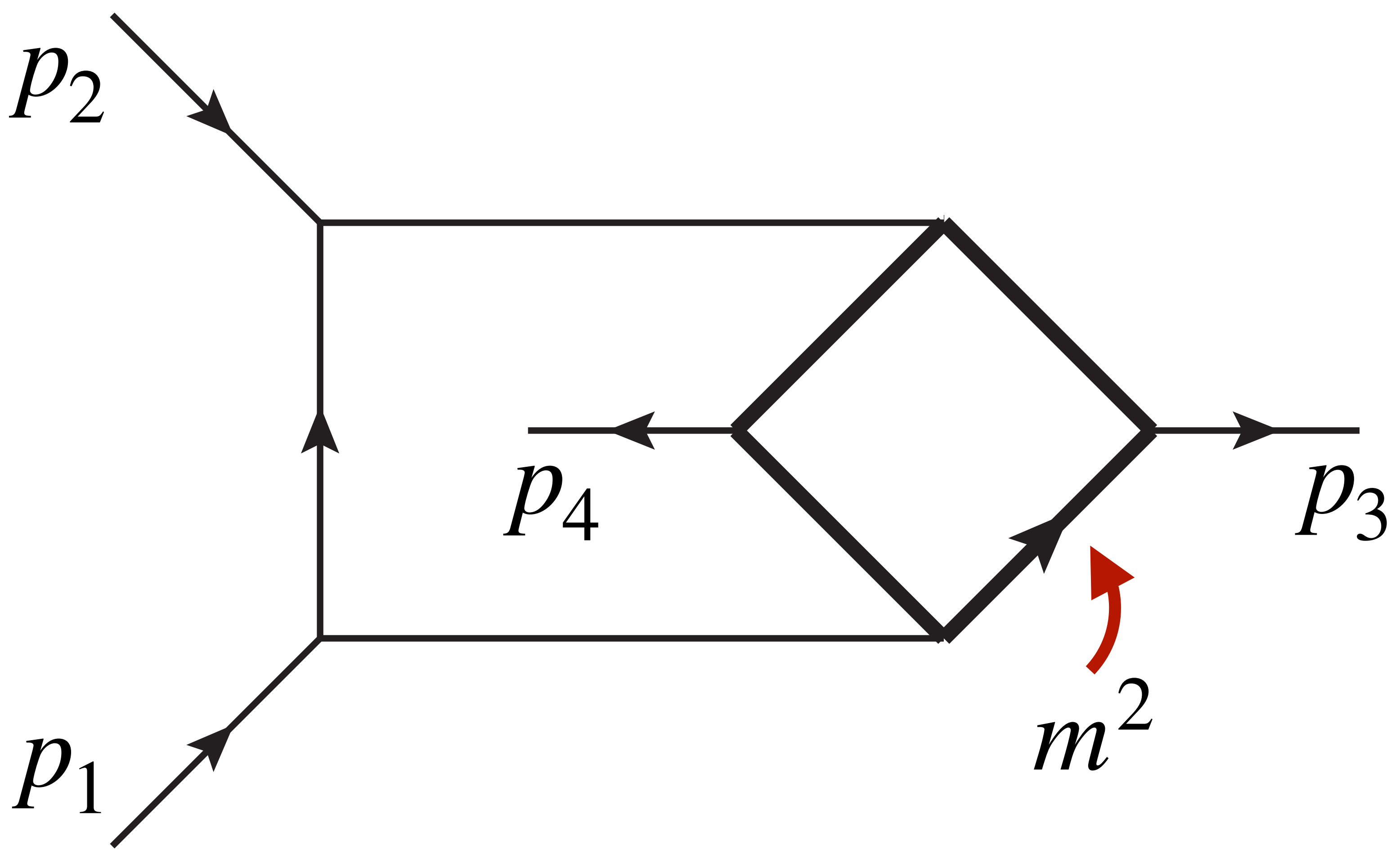
Dimensionless kinematic variables



Re or Im part of $\vec{F}^{(w)}$ up to a certain order in ϵ

In the examples we considered: 3/4 hidden layers, 32—256 nodes per layer

Heavy crossed box



3 kinematic variables, 36 MIs

$$\vec{v} = \left\{ s = (p_1 + p_2)^2 \,,\; t = (p_1 - p_3)^2 \,,\; m^2 \right\}$$

Canonical DEs / analytic solution unavailable

Subsectors involve elliptic functions

[von Manteuffel, Tancredi 2017]

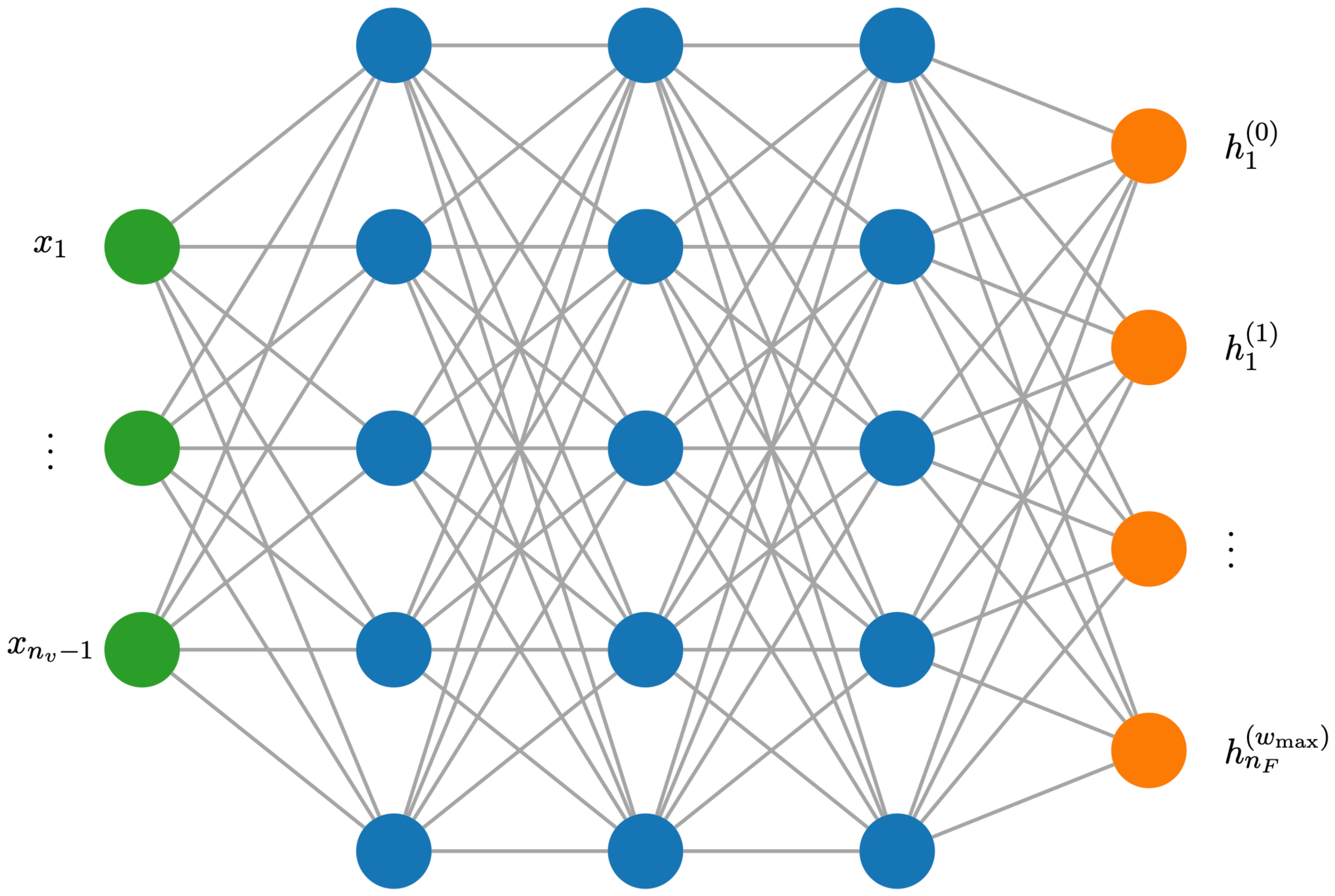
Full computation only recently, using generalised power series expansions (DiffExp)

[Becchetti, Bonciani, Cieri, Coro, Ripani 2023] *[Hidding 2020]*

MIs stripped of square roots → $A_{v_i}(\vec{v};\epsilon) = \sum_{k=0}^{2} \epsilon^k \, A_{v_i}^{(k)}(\vec{v})$

Heavy crossed box: architecture

2 input variables
(fix $m^2 = 1$)



3 hidden layers, 256 neurons each

MIs (Re or Im)

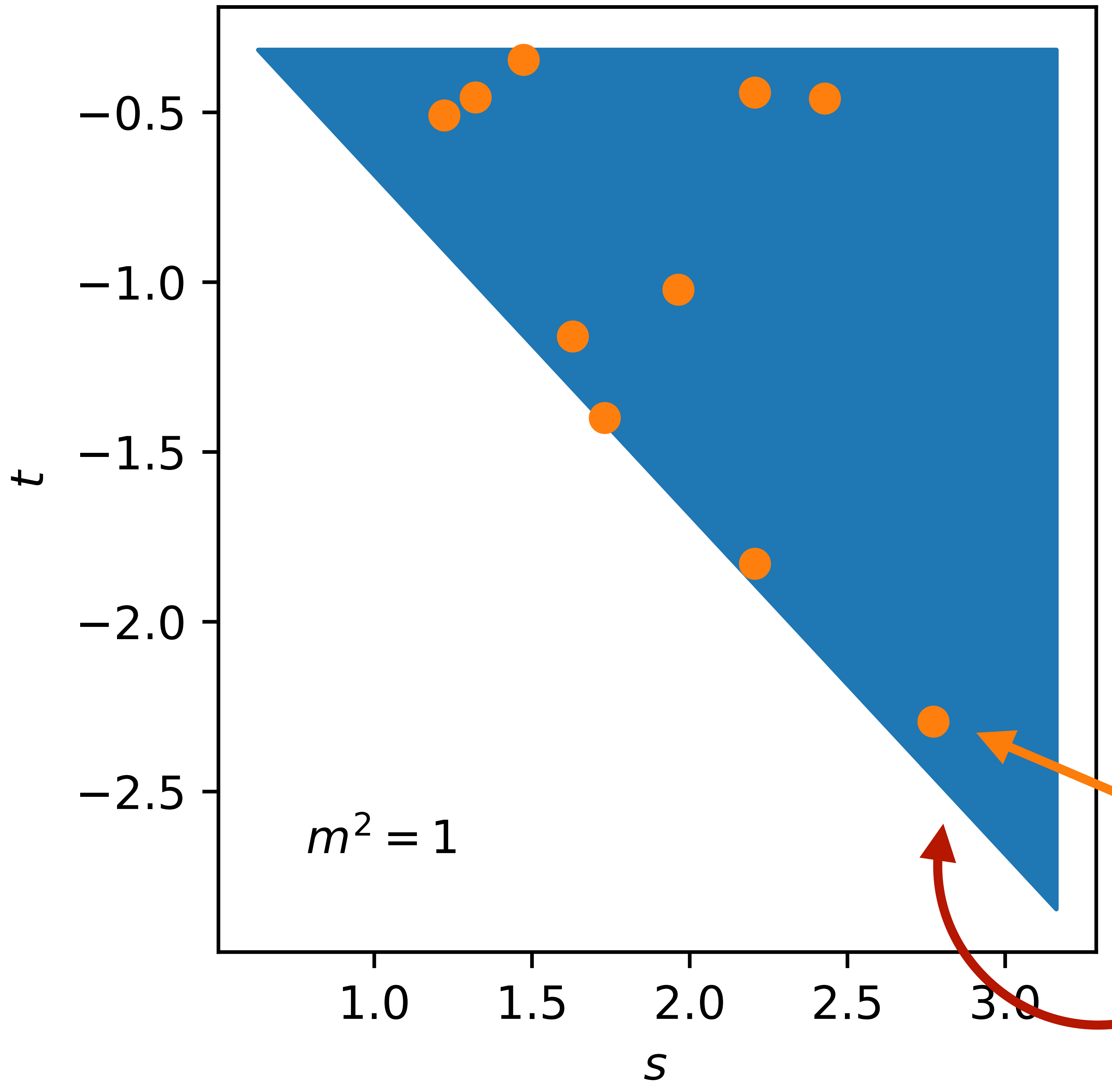
36 x 5 = 180 outputs

ϵ orders

$$\vec{F}(\vec{v};\epsilon) = \frac{1}{\epsilon^4}\sum_{w=0}^{4}\epsilon^w\,\vec{F}^{(w)}(\vec{v})$$

Heavy crossed box: kinematic region

s channel: $s > -t > 0 \wedge m^2 > 0$ → Never leave the chosen domain of analyticity domain, so analytic continuation is not required



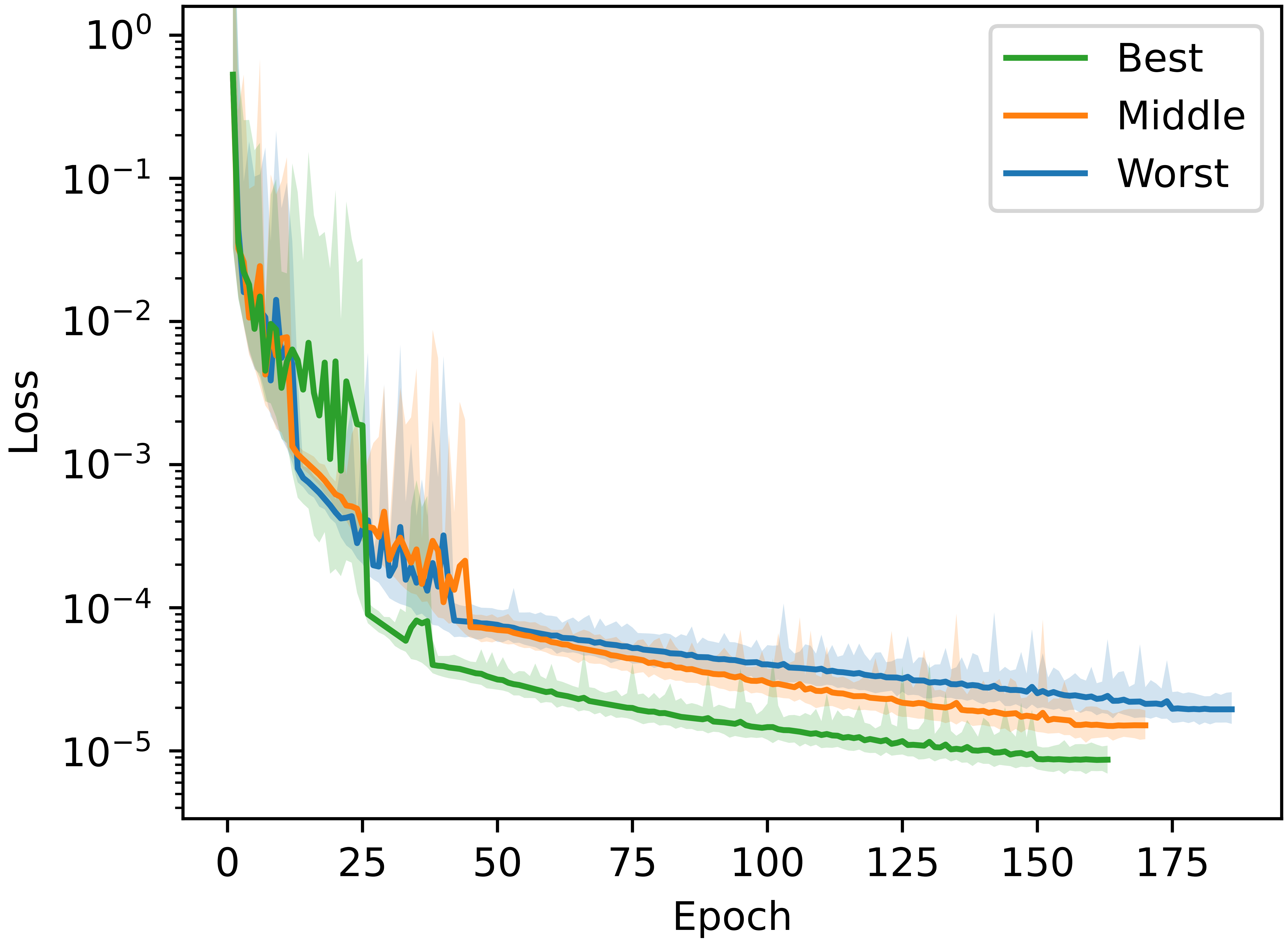
We choose $s < \sqrt{10}$

Singularities of the solution

Cut near boundaries:
$10\,\%$ of largest value ($\sqrt{10}$)

Boundary values at 10 random points, obtained with AMFlow *[Liu, Ma 2022]*

Heavy crossed box: training



Ensemble of 10 NNs

Iterations: 7.9×10^4

Time to train 1 NN: 75 min (on a good laptop, GPU)

Use training metric for validation, as inputs for DE loss function are dynamically random sampled

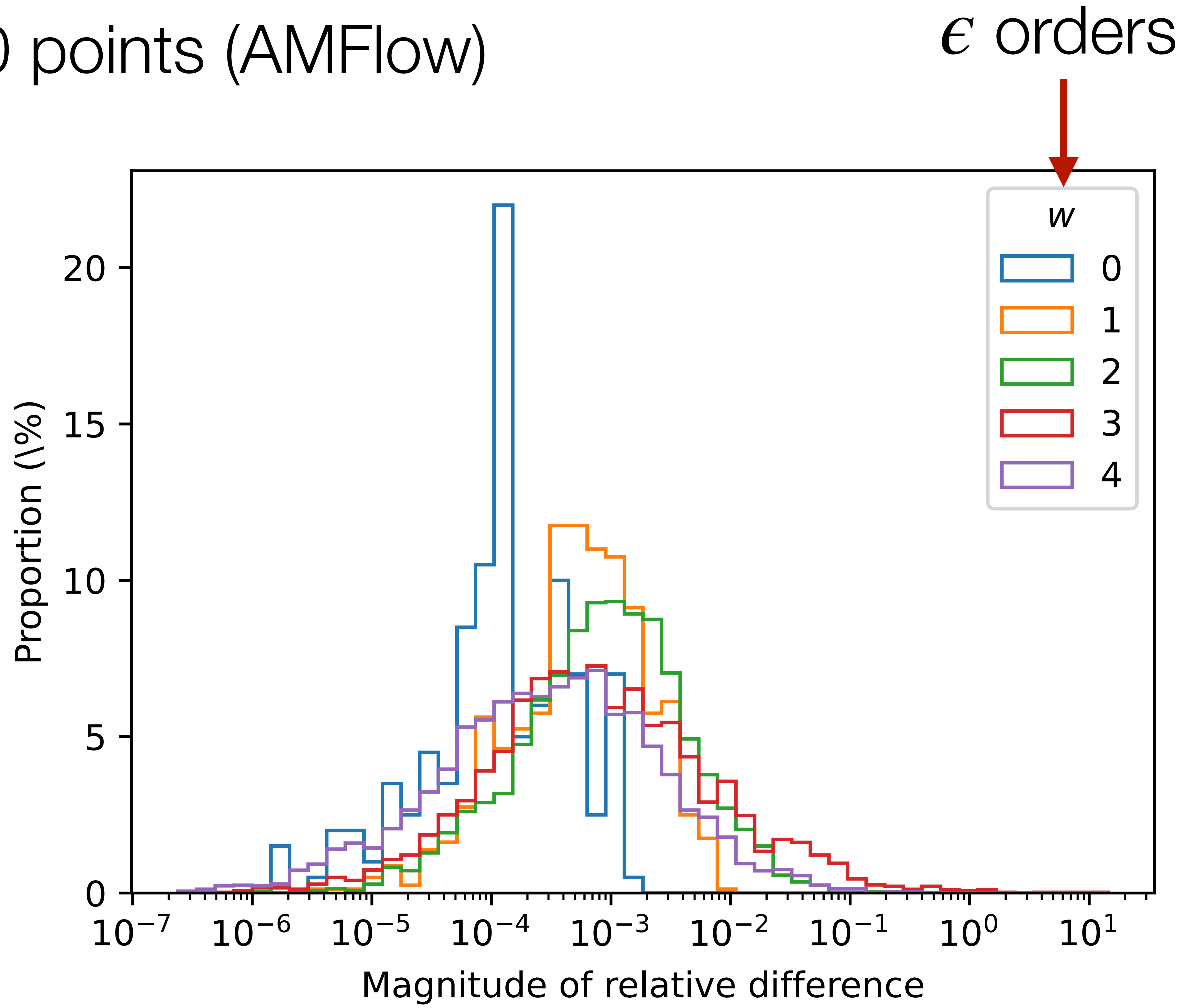
Heavy crossed box: model performance

Comparison against testing dataset of 100 points (AMFlow)

Mean absolute difference: 1.6×10^{-3}

Mean magnitude of rel. diff.: 7.3×10^{-3}

Evaluation time $\sim 1 - 10\ \mu s$

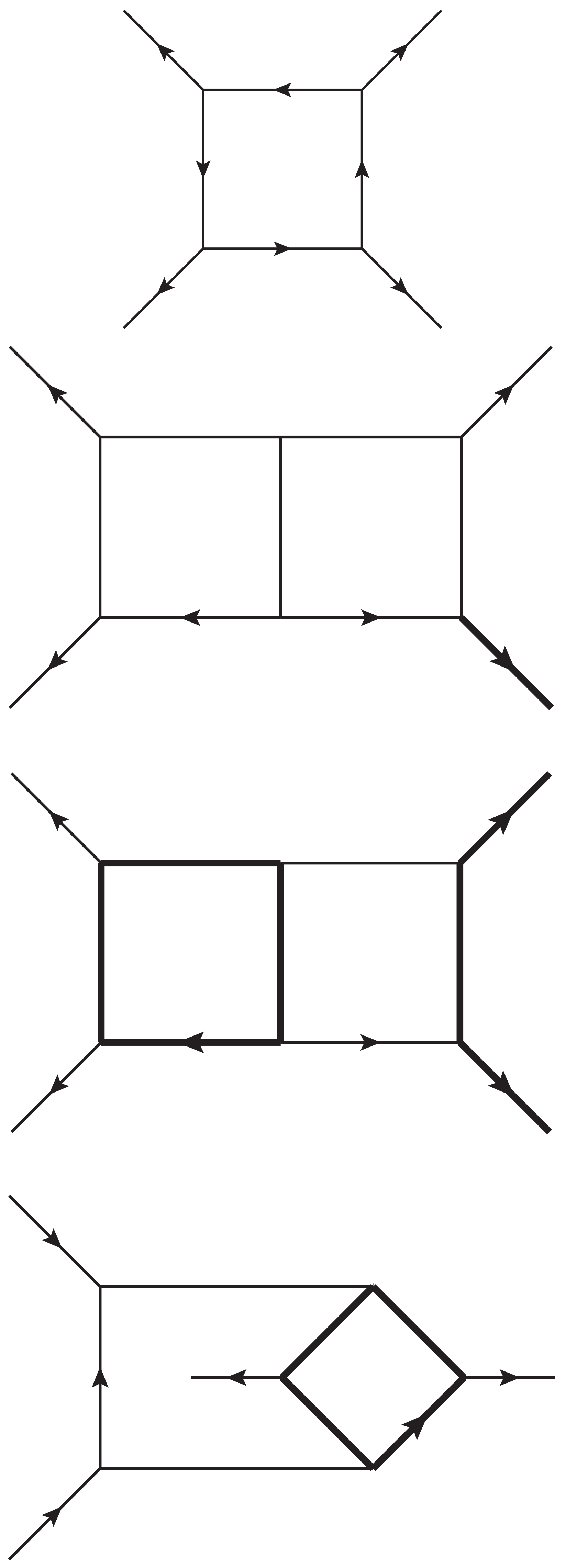


General comments

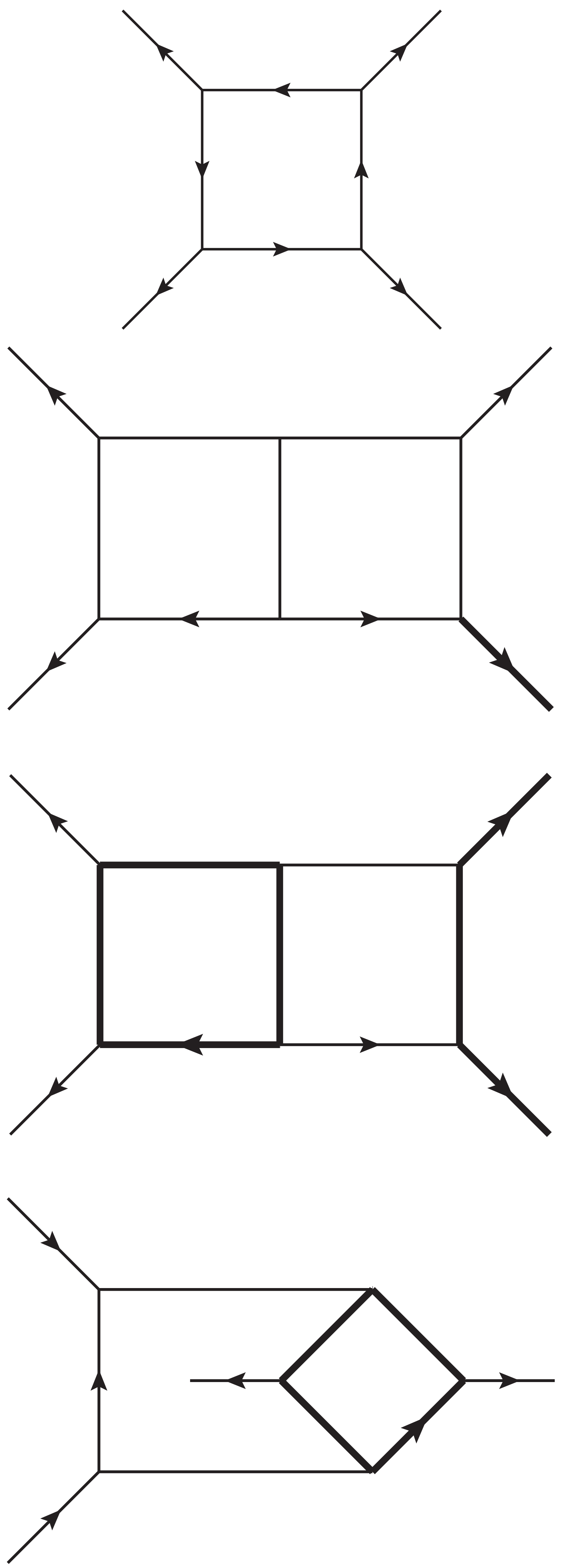
Flatness of the performance with respect to

- Analytic complexity (ϵ orders, MI) within the same family
- Across different families

Instantaneous evaluation times 🥳



General comments



Flatness of the performance with respect to

- Analytic complexity (ϵ orders, MI) within the same family
- Across different families

Instantaneous evaluation times 🥳

As of now, low control over accuracy 🙁

We can estimate it (ensemble uncertainty, differential error…), but unclear how to increase it arbitrarily

General comments

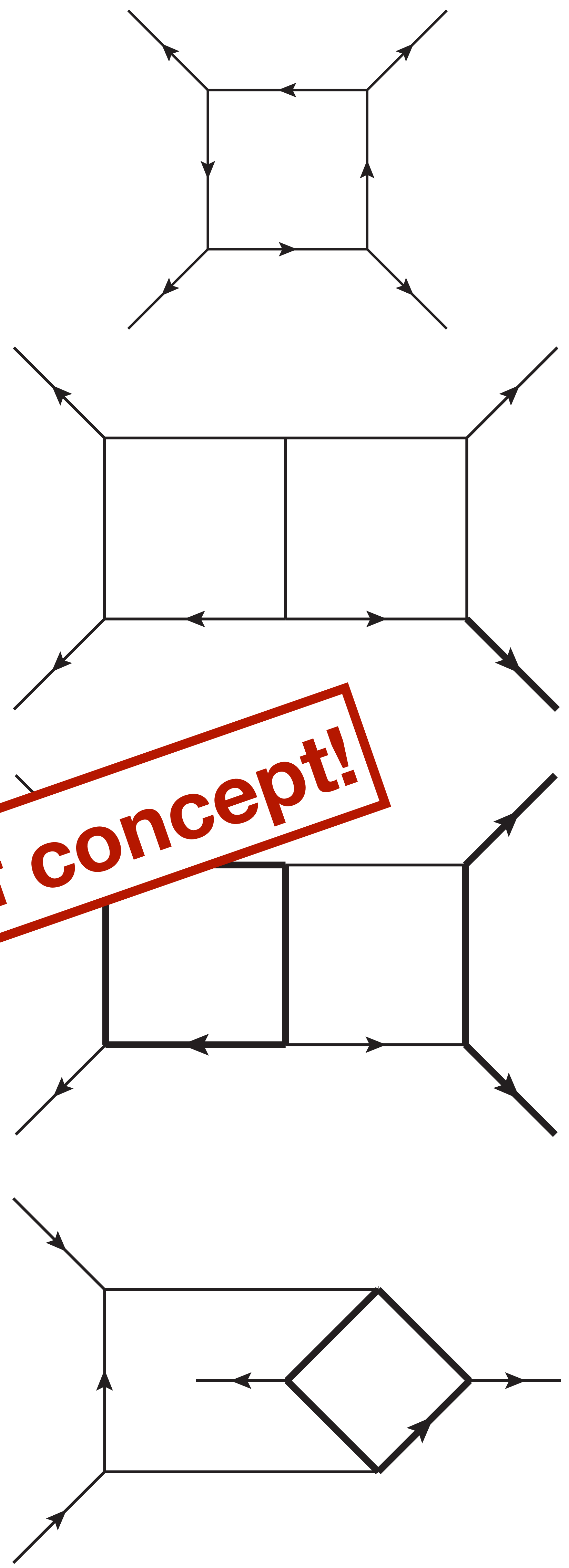
Flatness of the performance with respect to

- Analytic complexity (ϵ orders, MI) within the same family
- Across different families

Instantaneous evaluation times 🥳

As of now, low control over accuracy 🙁

We can estimate it (ensemble uncertainty, differential error…), but unclear how to increase it arbitrarily



Only proof of concept!

General comments

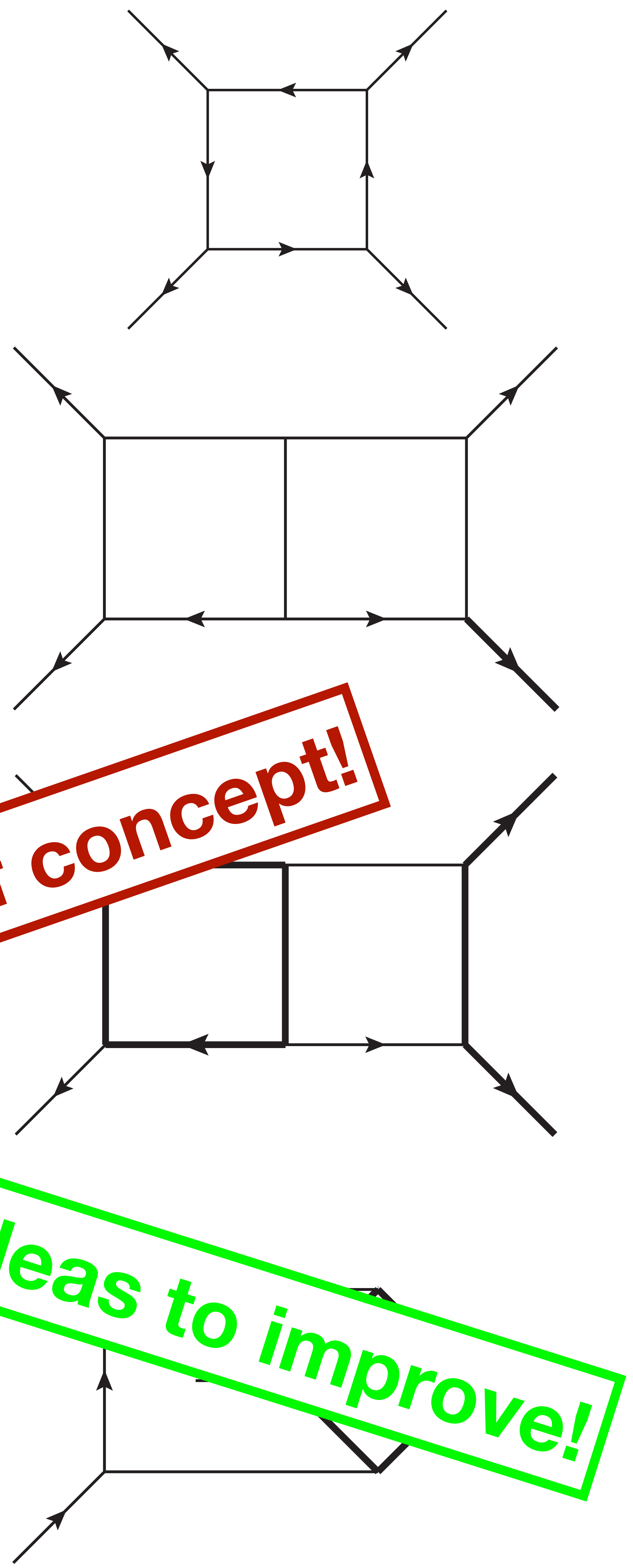
Flatness of the performance with respect to

- Analytic complexity (ϵ orders, MI) within the same family
- Across different families

Instantaneous evaluation times 🥳

As of now, low control over accuracy 🙁

We can estimate it (ensemble uncertainty, differential error…), but unclear how to increase it arbitrarily



Only proof of concept!

Many ideas to improve!

Conclusion

New algorithm to construct solutions to canonical DEs

- Efficient computation of amplitudes
- Efficient numerical evaluation

→ All 2-loop 5-pt integrals with 1 external massive leg now available!

*Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** (2306.15431)*

New method to evaluate numerically Feynman integrals satisfying generic DEs using physics informed deep learning

*Calisto, Moodie, **SZ** (23XX.XXXXX)*

Conclusion

New algorithm to construct solutions to canonical DEs

- Efficient computation of amplitudes
- Efficient numerical evaluation

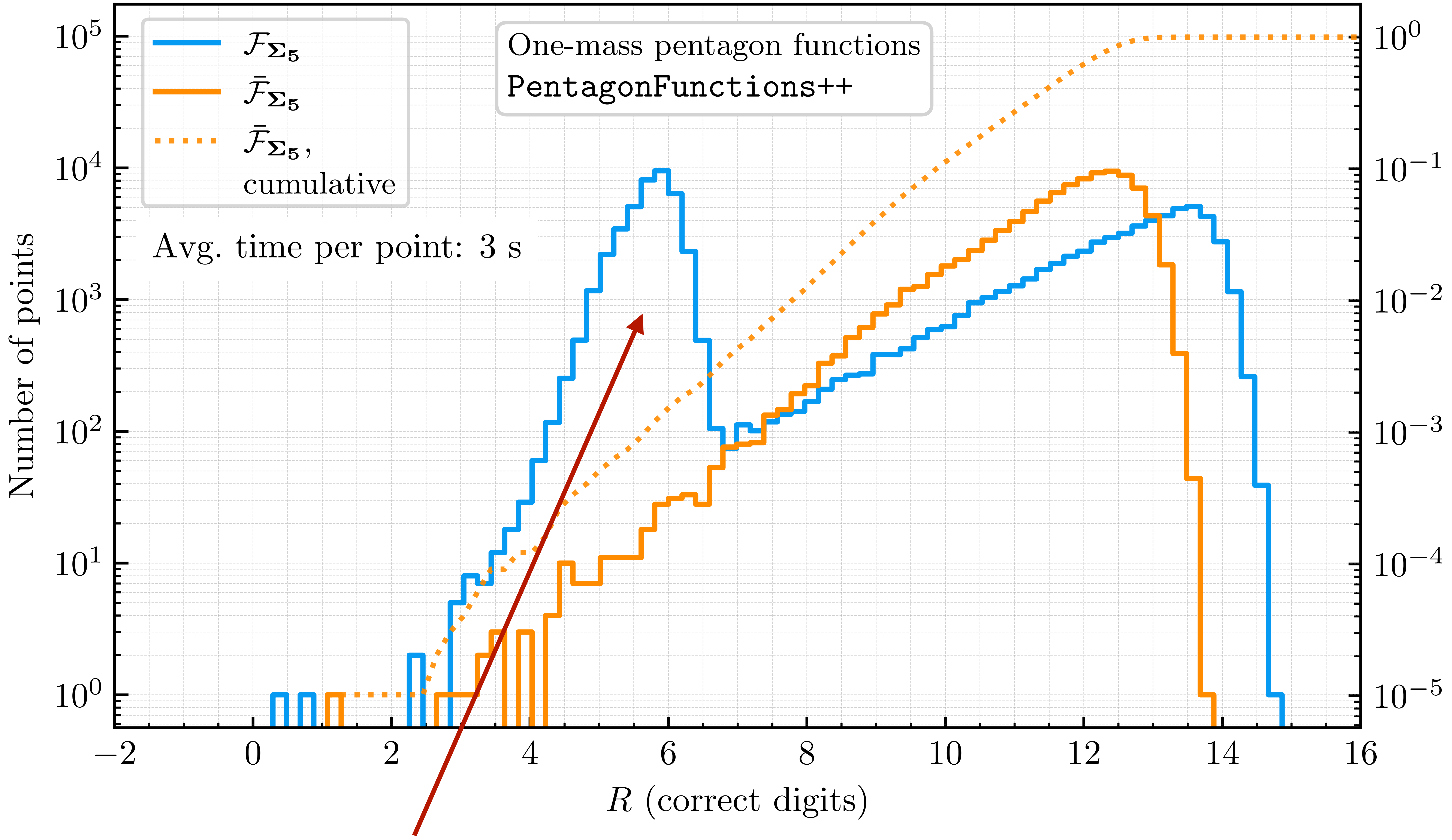
→ All 2-loop 5-pt integrals with 1 external massive leg now available!

*Abreu, Chicherin, Ita, Page, Sotnikov, Tschernow, **SZ** (2306.15431)*

New method to evaluate numerically Feynman integrals satisfying generic DEs using physics informed deep learning

*Calisto, Moodie, **SZ** (23XX.XXXXX)*

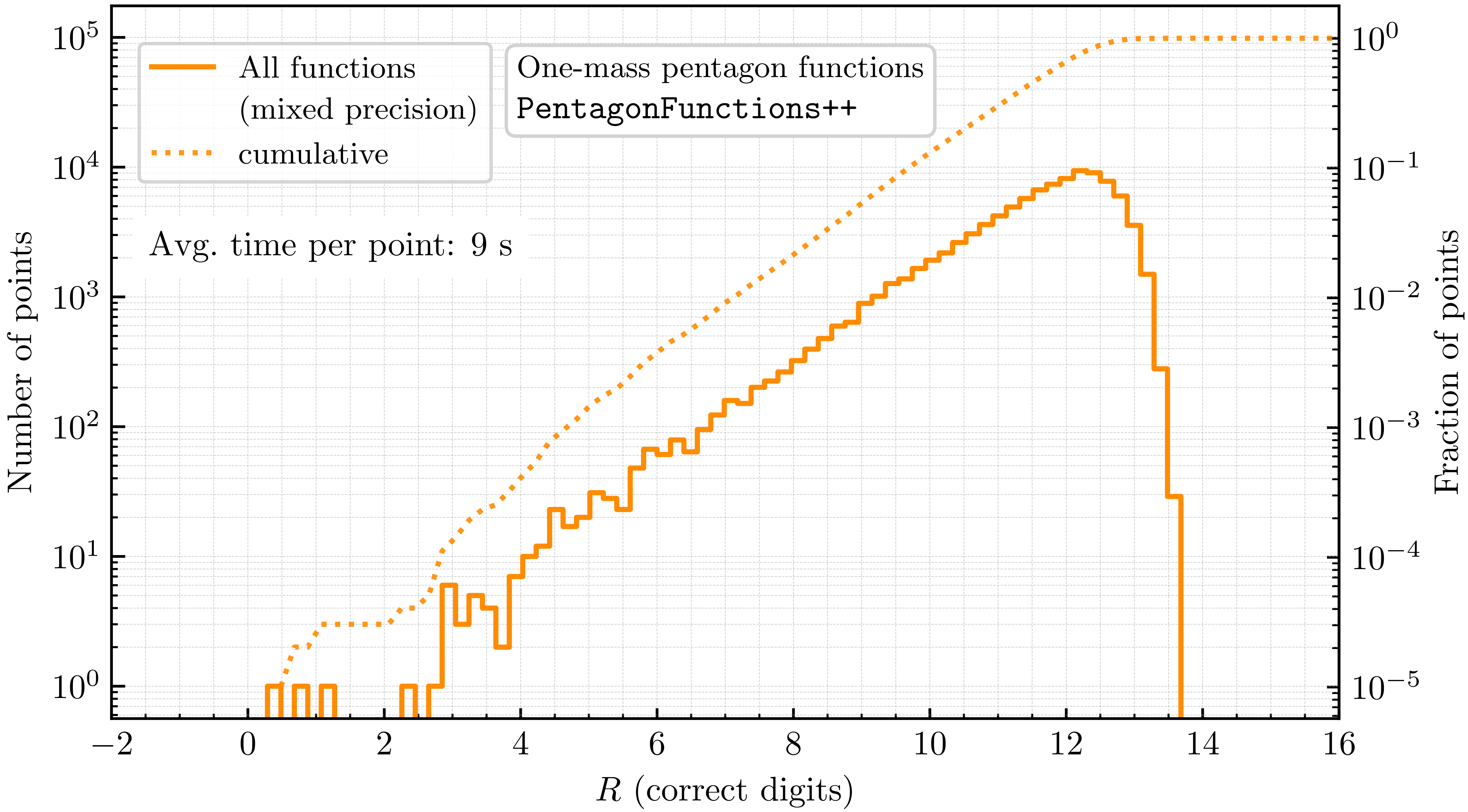
Thank you!



Double accuracy (quadruple and octuple also available)

New non-planar feature: integrable (and genuine) singularities in the s_{45} channel at $\Sigma_5^{(i)} = 0$

Planar subset ~10 times better



All functions
(mixed precision)
cumulative
One-mass pentagon functions
PentagonFunctions++
Avg. time per point: 9 s
Number of points
Fraction of points
R (correct digits)

Solving the canonical DEs in terms of iterated integrals is straightforward

$$\begin{cases} \mathrm{d}\left[w_{i_1}, \ldots, w_{i_n}\right]_{s_0}(s) = \mathrm{d}\log w_{i_n}(s) \left[w_{i_1}, \ldots, w_{i_{n-1}}\right]_{s_0}(s) \\ \left[w_{i_1}, \ldots, w_{i_n}\right]_{s_0}(s_0) = 0 \end{cases}$$

Chen's iterated integrals

Canonical DEs

$$\begin{cases} \mathrm{d}\,\vec{\mathrm{F}}^{(w)}(s) = \sum_i a_i \,\mathrm{d}\log w_i(s)\; \vec{\mathrm{F}}^{(w-1)}(s) \\ \vec{\mathrm{F}}^{(w)}(s_0) = \vec{\mathrm{F}}_0^{(w)} \end{cases}$$

Insensitive to square roots!

Proof-of-concept implementation

PyTorch

GELU activation function (nonzero and continuous 2nd-order derivatives)

Train with stochastic gradient descent (Adam optimiser)

Mini-batch training: iterations organised into epochs composed of small batches, taking a dynamic random sample of the inputs for each batch

- No need for regularisation to avoid overfitting
- Validation can be done on the training dataset

Loss function

$$L_{\mathrm{DE}}(\mathrm{D}_{\mathrm{DE}}, \theta) = \overline{\sum_{\vec{x}^{(i)} \in \mathrm{D}_{\mathrm{DE}}}} \; \overline{\sum_{j=1}^{n_F}} \overline{\sum_{l=1}^{n_v - 1}} \overline{\sum_{w=0}^{w_{\max}}} \left[\partial_{x_l} h_j^{(w)}\left(\vec{x}^{(i)}; \theta\right) - \sum_{k=0}^{\min(w, k_{\max})} \sum_{r=1}^{n_F} A_{x_l, jr}^{(k)}\left(\vec{x}^{(i)}\right) h_r^{(w-k)}\left(\vec{x}^{(i)}; \theta\right) \right]^2$$

$$L_{\mathrm{b}}(\mathrm{D}_{\mathrm{b}}, \theta) = \overline{\sum_{\vec{x}^{(i)} \in \mathrm{D}_{\mathrm{b}}}} \; \overline{\sum_{j=1}^{n_F}} \overline{\sum_{w=0}^{w_{\max}}} \left[h_j^{(w)}\left(\vec{x}^{(i)}; \theta\right) - g_j^{(w)}\left(\vec{x}^{(i)}\right) \right]^2$$

Integral family	box	one-mass double box	heavy crossed box	top double box
Inputs	1	2	2	2
Hidden layers	3×32	3×256	3×256	4×128
Outputs	15	90	180	99
Learning rate	10^{-2}	10^{-3}	10^{-3}	10^{-3}
Batch size	64	256	256	256
Boundary points	2	6	10	20
c_{n_v}	$s = 10$	$s_{12} = 2.5$	$m^2 = 1$	$m_{\mathrm{t}}^2 = 1$
Scale bound	—	—	$s \leq \sqrt{10}$	$s_{12} \leq 5$
Physical cut (%)	10	10	10	10
Spurious cut (%)	0	0	0	1

Summary of hyperparameters

Integral family	Final loss	Iterations	Time (minutes)
box	2.7×10^{-7}	2.5×10^{5}	16
one-mass double box	3.4×10^{-4}	1.1×10^{5}	53
heavy crossed box	1.4×10^{-5}	7.9×10^{4}	75
top double box	7.1×10^{-4}	5.2×10^{4}	32

Training statistics

Integral family	MEU	MDE	MAD	MMRD	MLR	Size
box	2.8×10^{-5}	3.6×10^{-4}	2.9×10^{-5}	2.2×10^{-5}	3.9×10^{-7}	10^5
one-mass DB	8.1×10^{-4}	1.1×10^{-2}	2.0×10^{-3}	1.1×10^{-2}	-2.8×10^{-4}	10^5
heavy CB	2.8×10^{-4}	2.8×10^{-3}	1.6×10^{-3}	7.3×10^{-3}	-4.5×10^{-4}	10^2
top DB	1.9×10^{-4}	1.7×10^{-3}	9.0×10^{-4}	3.9×10^{-3}	1.8×10^{-4}	10^2

Uncertainty and testing errors