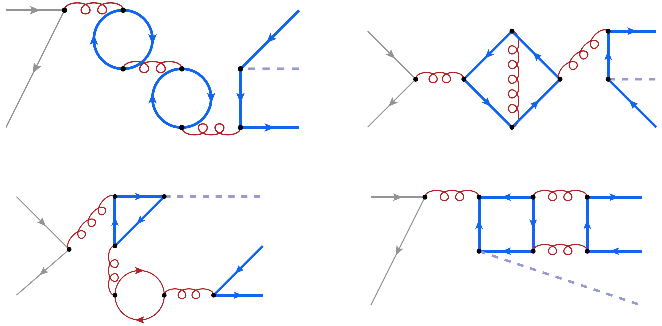
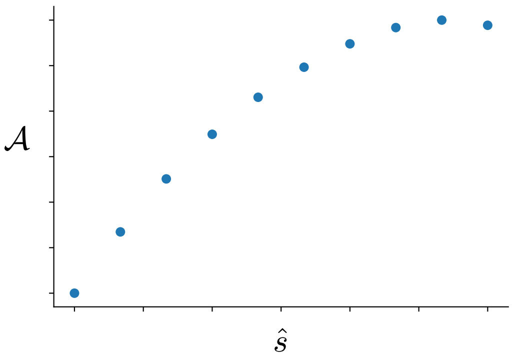
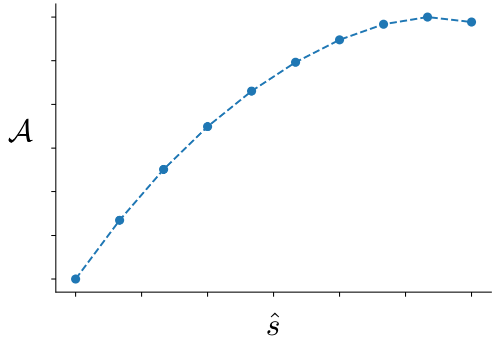
Numerical Scattering Amplitudes with pySecDec: towards $q\bar{q} \rightarrow t\bar{t}H$ at two loops



→



→



Based on 2305.19768, 2402.03301, 2409.xxxxx:

Bakul Agarwal, Víctor Bresó-Pla, Gudrun Heinrich, Stephen Jones, Matthias Kerner, Sven Yannick Klein, Jannis Lang, Vitaly Magerya, AO, Johannes Schlenk

High Precision for Hard Processes – September 2024

Anton Olsson

Outline

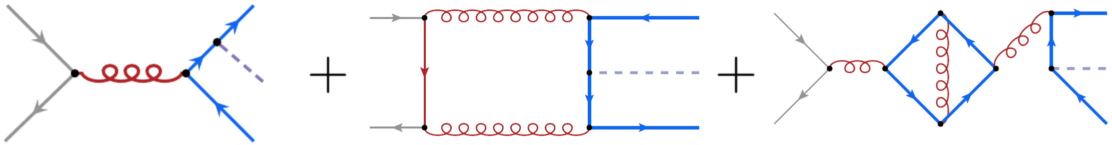
1. Numerical scattering amplitudes with pySecDec
2. Application to $q\bar{q} \to t\bar{t}H$ production
3. Constructing an amplitude grid

1. Numerical Scattering Amplitudes with pySecDec

Calculating an amplitude

- **Generation:** Write amplitude as sum of Feynman diagrams

$$\mathcal{A}_{\mathcal{V}} = \mathcal{A}_{\mathcal{V}}^{\text{LO}} + \mathcal{A}_{\mathcal{V}}^{\text{NLO}} + \mathcal{A}_{\mathcal{V}}^{\text{NNLO}} + \cdots \sim$$



  - This results in a very large number of *linearly dependent* Feynman integrals

- **Reduction:** Find basis of *linearly independent* **master integrals** through IBP relations

- **Evaluation:** Evaluate the **master integrals**

  - With many loops, scales and legs this is very hard analytically
  - Resort to numerical solutions (**pySecDec)**

Evaluating with pySecDec

- Targets dimensionally regulated Feynman integrals

$$I \sim \int_0^\infty \prod_{j=1}^{N} \mathrm{d}x_j x_j^{\nu_j - 1} \delta(1 - \sum_{i=1}^{N} x_i) \frac{\mathcal{U}(\bar{x})^{N_\nu - (L+1)D/2}}{\mathcal{F}(\bar{x}, p^2, m^2)^{N_\nu - LD/2}}$$

L loops

$$D = 4 - 2\varepsilon$$

$$N_\nu = \sum_{j=1}^{N} \nu_j$$

- Two-step process:
  1. Sector decomposition: Isolate and extract singularities as expansion in the regulator, **do once**
  2. Quasi-Monte Carlo (QMC) integration, **do once for each phase space point**

Sector Decomposition

- Transforms integral into sums of expansions in the regulator

$$I \to \sum_{\text{Sectors}} \sum_{n=-r}^{2L} C_n(p^2, m^2) \frac{1}{\varepsilon^n} + \mathcal{O}(\varepsilon^{r+1})$$

Parameter integrals

- Singularities are extracted as poles in regulator with simple subtraction terms

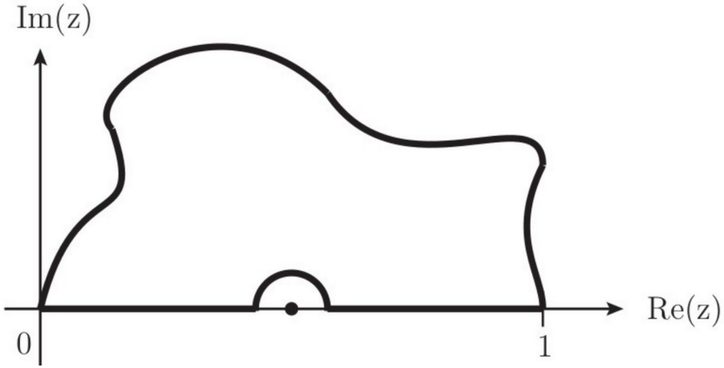
$$\int_0^1 \mathrm{d}x\, x^{-1-\varepsilon} \mathcal{I}(x,\varepsilon) = \frac{-1}{\varepsilon} \mathcal{I}(0,\varepsilon) + \underbrace{\int_0^1 \mathrm{d}x\, x^{-1-\varepsilon} [\mathcal{I}(x,\varepsilon) - \mathcal{I}(0,\varepsilon)]}_{\text{Finite piece, integrate numerically!}} = \frac{C_1}{\varepsilon} + C_0$$

Contour Deformation

- Sector decomposition extracts endpoint singularities only
  - Kinematic (bulk) singularities are avoided with contour deformation
- Contour deformation works in most cases, but is computationally very expensive
  - New ideas to avoid having to use contour deformation [2407.06973], (see talk by Stephen Jones)

$$0 = \oint_c \prod_{j=1}^{N} \mathrm{d}z_j \mathcal{I}(\vec{z}) = \int_0^1 \prod_{j=1}^{N} \mathrm{d}x_j \, \mathcal{I}(\vec{x}) + \int_\gamma \prod_{j=1}^{N} \mathrm{d}z_j \, \mathcal{I}(\vec{z}(\vec{x}))$$

$$\mathcal{F}(\vec{z}(\vec{x})) = \mathcal{F}(\vec{x}) - i\,\lambda \underbrace{\sum_j x_j(1-x_j) \left(\frac{\partial \mathcal{F}}{\partial x_j}\right)^2}_{\text{Complicates integrand massively}} + \mathcal{O}(\lambda^2)$$

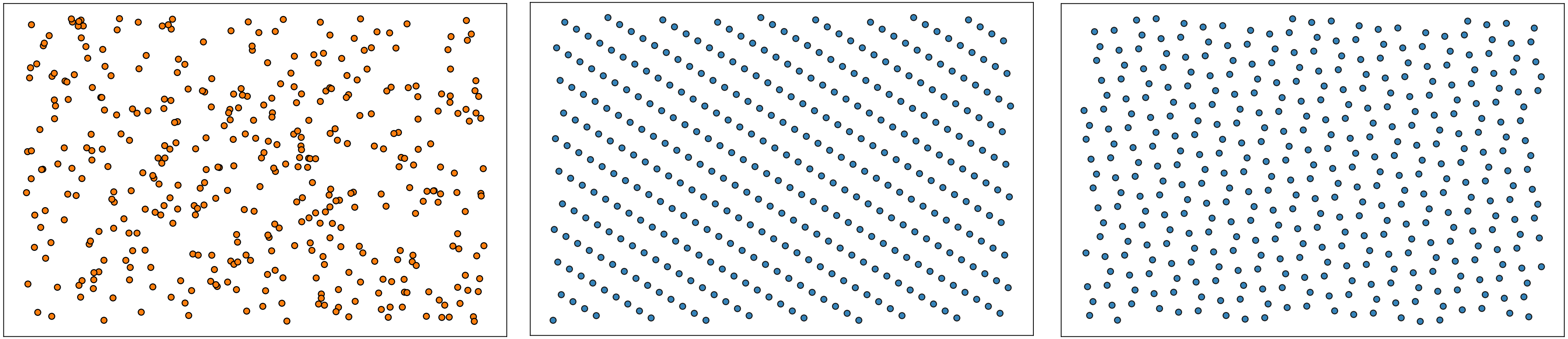


QMC Integration

- Observation: $\varepsilon \leq \text{Discrepancy}\{x_i\} \cdot V[\mathcal{I}]$, take $\{x_i\}$ from a low discrepancy sequence (R1SL-rule)
- Estimate of integral is achieved through random shifts
- Error convergence: $\varepsilon_{MC} \sim \mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$ and $\varepsilon_{RQMC} \sim \mathcal{O}\left(\frac{1}{n}\right)$

$$I[\mathcal{I}] \approx \bar{Q}_{n,m}[\mathcal{I}] \equiv \frac{1}{m}\sum_{k=0}^{m-1} Q_n^{(k)}[\mathcal{I}], \quad Q_n^{(k)}[\mathcal{I}] \equiv \frac{1}{n}\sum_{i=0}^{n-1} \mathcal{I}\left(\left(\frac{i\mathbf{z}}{n} + \Delta_k\right) \mathbf{mod}\, 1\right)$$

Random samples (MC) Quasi-random samples (QMC) Random shifts (RQMC)



2. Application to $t\bar{t}H$ production

Motivation

- Why is this an interesting process?
  - Direct sensitivity to the top-quark Yukawa coupling y_t
  - Probe CP properties of y_t [2208.02686 (CMS) 2303.05974 (ATLAS)]
- Projected statistical uncertainties for HL-LHC ~2-3% [Snowmass '22]
  - Systematics will dominate
- At NLO QCD scale uncertainties ~10-15%
  - NNLO QCD amplitude necessary to match precision
  - Approximations exist [2210.07846, 2402.00431] (see talks by Chiara Savoini and Guoxing Wang)

Generation

- Generate Feynman diagrams and insert Feynman rules (Qgraf [1])
- Project amplitude onto the Born

$$\langle \mathcal{A}^0 | \mathcal{A} \rangle = \langle \mathcal{A}^0 | \mathcal{A}^0 \rangle + \left(\frac{\alpha_s}{2\pi}\right) \langle \mathcal{A}^0 | \mathcal{A}^1 \rangle + \left(\frac{\alpha_s}{2\pi}\right)^2 \langle \mathcal{A}^0 | \mathcal{A}^2 \rangle + \mathcal{O}(\alpha_s^3)$$

  - Also: $+ \left(\frac{\alpha_s}{2\pi}\right)^2 \langle \mathcal{A}^1 | \mathcal{A}^1 \rangle$

- Spinor and colour summations (Form [2], COLOR.H [3], Alibrary [4])
- For the N_f – part [2402.03301] of the amplitude, this yielded ~20 000 scalar integrals
  - For the full $q\bar{q}$ channel ~90 000 integrals

Reduction and Evaluation

- Generation produces many *linearly dependent* integrals
  - IBP reduction reveals linear relations and allows construction of a basis of master integrals
- Ideally: solve the system of IBP equations symbolically once and for all
- In reality:
  - we have $s_{12}, s_{23}, s_{34}, s_{45}, s_{51},\ m_H^2, m_t^2,\ \varepsilon$
  - Symbolic reductions are at the moment not possible
- Solution: plug in numbers for each variable and solve system numerically [5-7]
  - Need to solve the IBP system repeatedly for each phase space point ~ O(minutes)
- Evaluate linear combination of reduced integrals with pySecDec ~ O(minutes)
  - New in 1.6: Optimized for sums of integrals [2305.19768]

3. Constructing an Amplitude Grid

Amplitude Grids

- Amplitudes are used to compute physical observables

$$\langle \mathcal{O} \rangle_\Phi = \int_{\bar{x} \in \Phi} \mathcal{O}(\bar{x}) \mathrm{d}\sigma(\bar{x})$$

Amplitude

PDFs

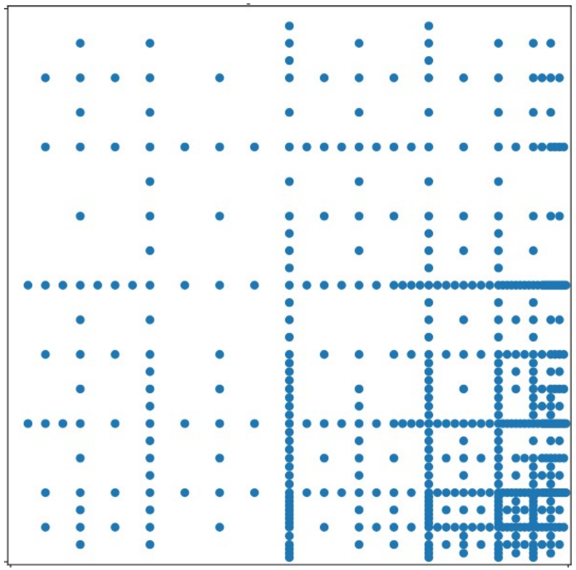
$$\mathrm{d}\sigma = \frac{1}{2\hat{s}} |\mathcal{A}|^2 \mathrm{d}\Phi \, \mathrm{d}\rho_{a,b}(\hat{s})$$

Phase space differential

- These integrals require millions of MC samples
  - Evaluation time of amplitude at 1 point ~ O(minutes)
  - Evaluation time of 1 observable ~ O(years)
- The solution is to evaluate the amplitude at a few points and interpolate for values in between
  - This implies there will be interpolation/grid uncertainties
  - How do these uncertainties propagate to observables?

The Interpolation Problem

- Our goal: approximate $a = |\mathcal{A}|^2$ with some $\tilde{a}$ defined on the whole phase space, based on the knowledge of a at some data points
  - When the approximation error of $\tilde{a}$ is “small enough”, the calculation is finished
- Two main questions:
  1. What should $\tilde{a}$ be? (polynomial, spline, neural network etc.)
  2. Where to evaluate a? (selecting interpolation nodes)
- Promising approximation methods we have investigated:
  - Chebyshev polynomials (global)
  - Sparse grids
  - Spline interpolation
  - Neural networks (GATr) [2405.14806]



Spatially adaptive sparse grid

Why is this difficult?

- $2 \rightarrow 2$: Usually 2 phase space variables
- $2 \rightarrow 3$: Usually 5 phase space variables

- Points with increasing dimension:
  - Curse of dimensionality

	$q\bar{q} \rightarrow t\bar{t}$	$q\bar{q} \rightarrow t\bar{t}H$
Points per dim \ Dimension	$\mathbf{d} = 2$	$\mathbf{d} = 5$
10	100	100000
20	400	3200000
30	900	24300000

Error Estimation

- Naive estimate: $\varepsilon = \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} |a(x_i) - \tilde{a}(x_i)|$, this often *overestimates* the error
  - A flat MC over-samples physically irrelevant regions of phase space

- Weight samples toward "relevant" phase space regions

- Define the "error" as the *variational distance* of the probability distributions

$$\varepsilon = \frac{||f - \tilde{f}||_1}{||f||_1} \quad , \quad f(x) = \frac{a(x)}{2\hat{s}} \left| \frac{\mathrm{d}(\Phi, \rho)}{\mathrm{d}x} \right|$$

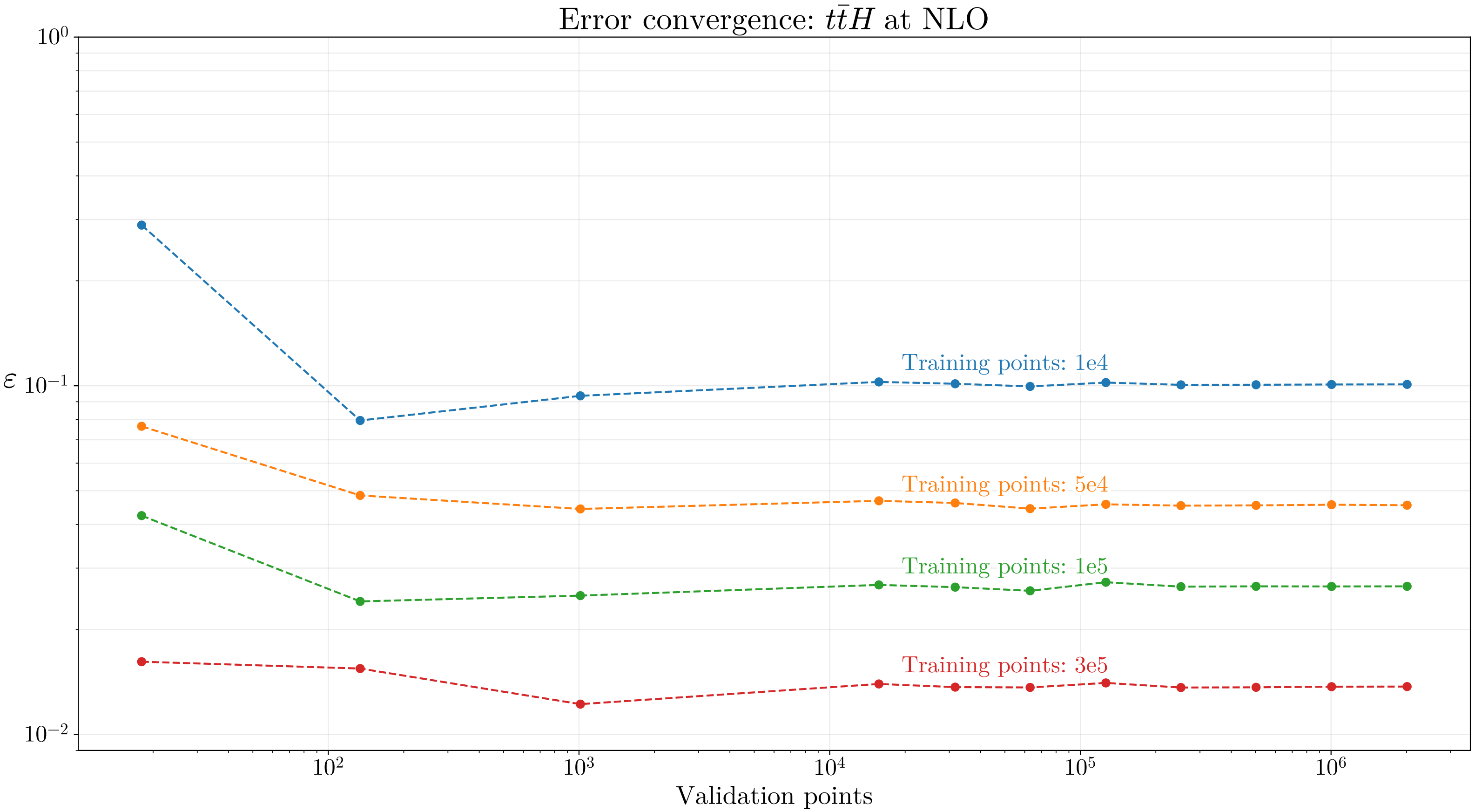
; target probability density for σ^{tot}

- Ensures the error on $\sigma^{\mathrm{tot}} \leq \varepsilon \left(\frac{\alpha_s}{2\pi} \right)^k ||f||_1$

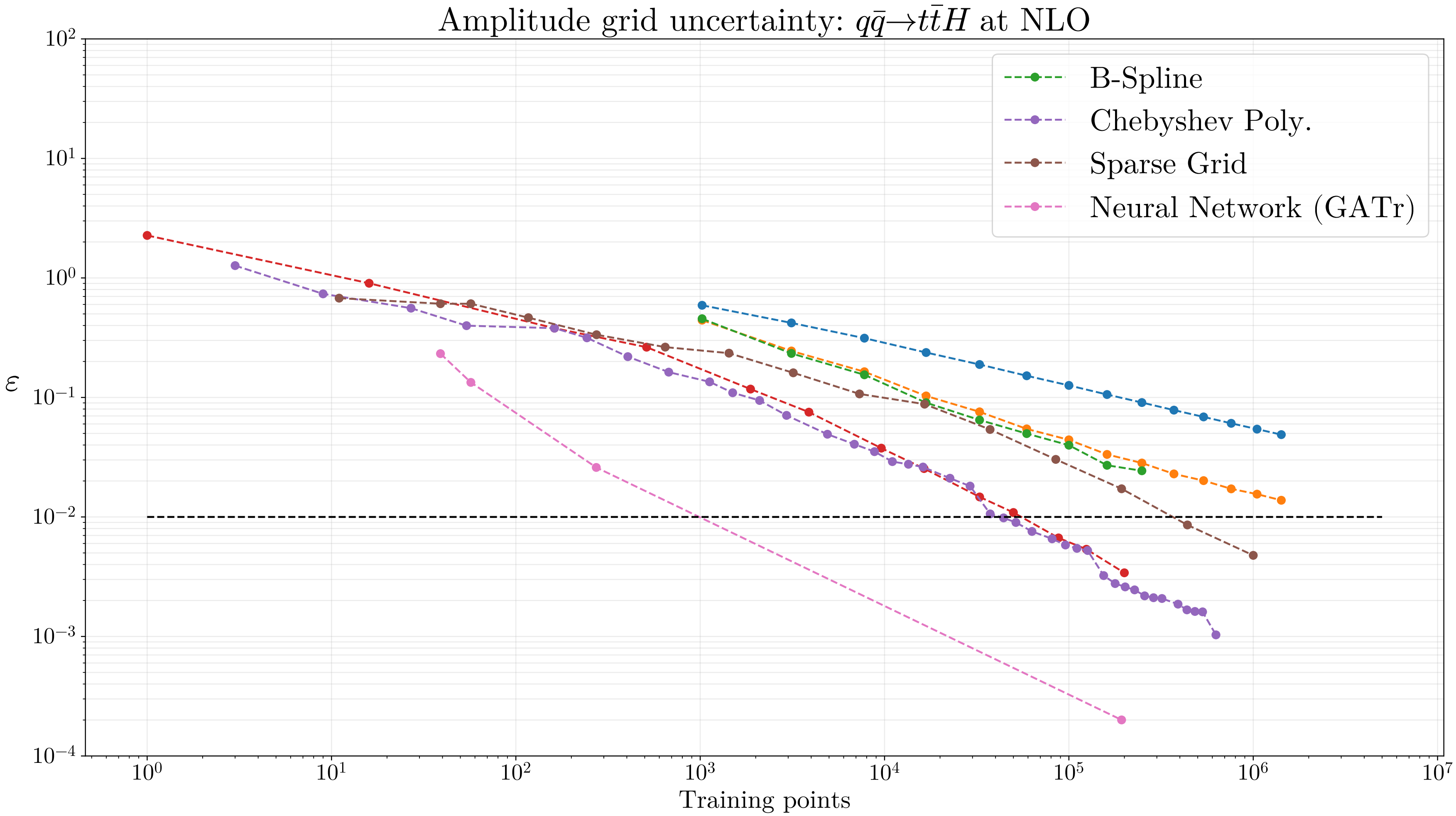
- If $\langle \mathcal{O} \rangle_\Phi = \sigma^{tot}$ then take $x \sim \frac{1}{2\hat{s}} \mathrm{d}\Phi \, \mathrm{d}\rho_{a,b}(\hat{s})$ and MC integrate: $\varepsilon = \frac{\sum_i |\widetilde{a}_i - a_i|}{\sum_i |a_i|}$

How many validation points are required?

- “Waste” as few points as possible on validation : errors defined as 1-norms stabilize quickly



Comparison of Interpolation Methods



Summary

- pySecDec is used to compute scattering amplitudes numerically
  - Current goal is the virtual amplitude to $q\bar{q} \to t\bar{t}H$ at NNLO
- Amplitude grids are required for combination with MC event generators
  - Especially for $2 \to 3$ processes with slow numerical evaluation times
  - This requires careful consideration of grid uncertainties
  - Grids can be optimized towards specific observables through physical arguments
- Our goal is to have an interpolation framework that can handle $2 \to 3$ scattering problems
- Next: apply to $t\bar{t}H$ production and other processes

Backup: QMC Integration

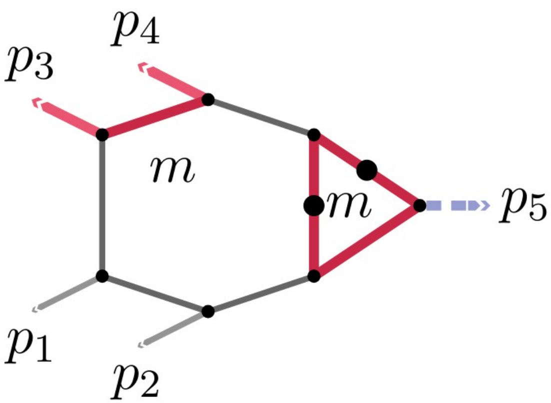
- Classical QMC error bound: $\varepsilon \leq \text{Discrepancy}\{x_i\} \cdot V[\mathcal{I}]$ (Star disc. ; Hardy and Krause variation)
- Smooth integrands have: $\mathcal{O}((\log n)^d/n)$ (dimension dependent?!)
- In certain *weighted function spaces* convergence becomes independent of dimension
- Example: Korobov space of periodic smooth functions
  - Our integrands are usually smooth but not periodic: Apply Korobov transformation
  - Differentiable integrands after Korobov transformation have: $\mathcal{O}\left(\frac{1}{n}\right)$
  - Convergence is independent of dimension

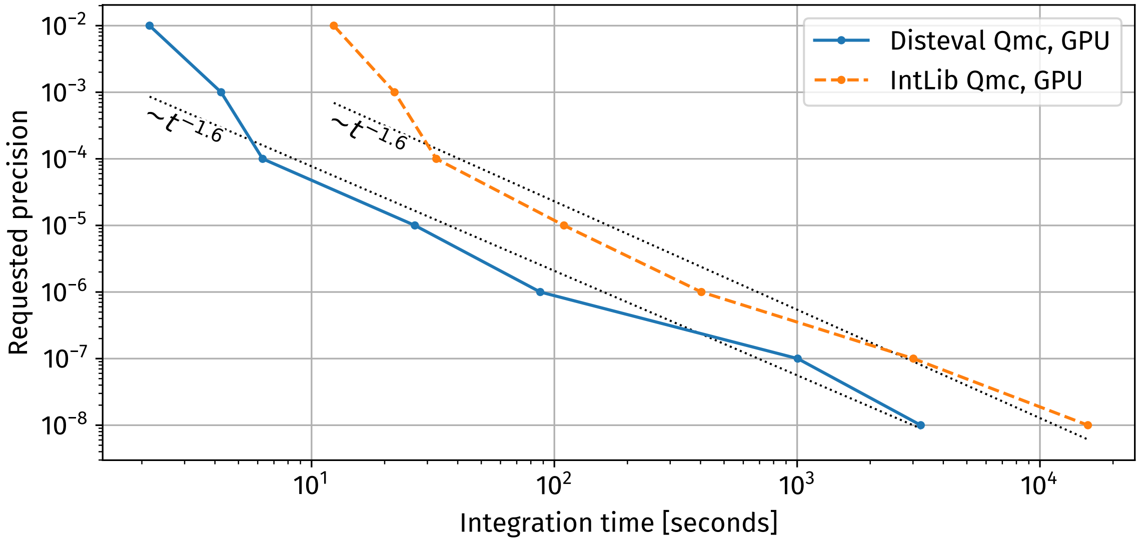
Backup: Latest pySecDec Release

- New QMC integrator “Disteval” (distributed evaluation)
  - Code improvements on CPU and GPU side
  - Up to an order of magnitude performance increase compared to previous versions
- Median generating vectors (new QMC lattice rule)
  - Avoids unlucky lattices
- Automated detection and insertion of extra regulators in EBR
  - Resolves spurious singularities introduced by EBR
- Increased practicality of amplitude calculations (sums of integrals)

Backup: Disteval speed-up sources

- Reasons for speedup:
  - Separated GPU and CPU code generation enable hardware specific optimizations
  - Integrand samples are summed up on the GPU using the Nvidia CUB library
  - Support for the *Single Instruction, Multiple Data* (SIMD) instruction set
  - Reduced overhead (very significant for small integrands)





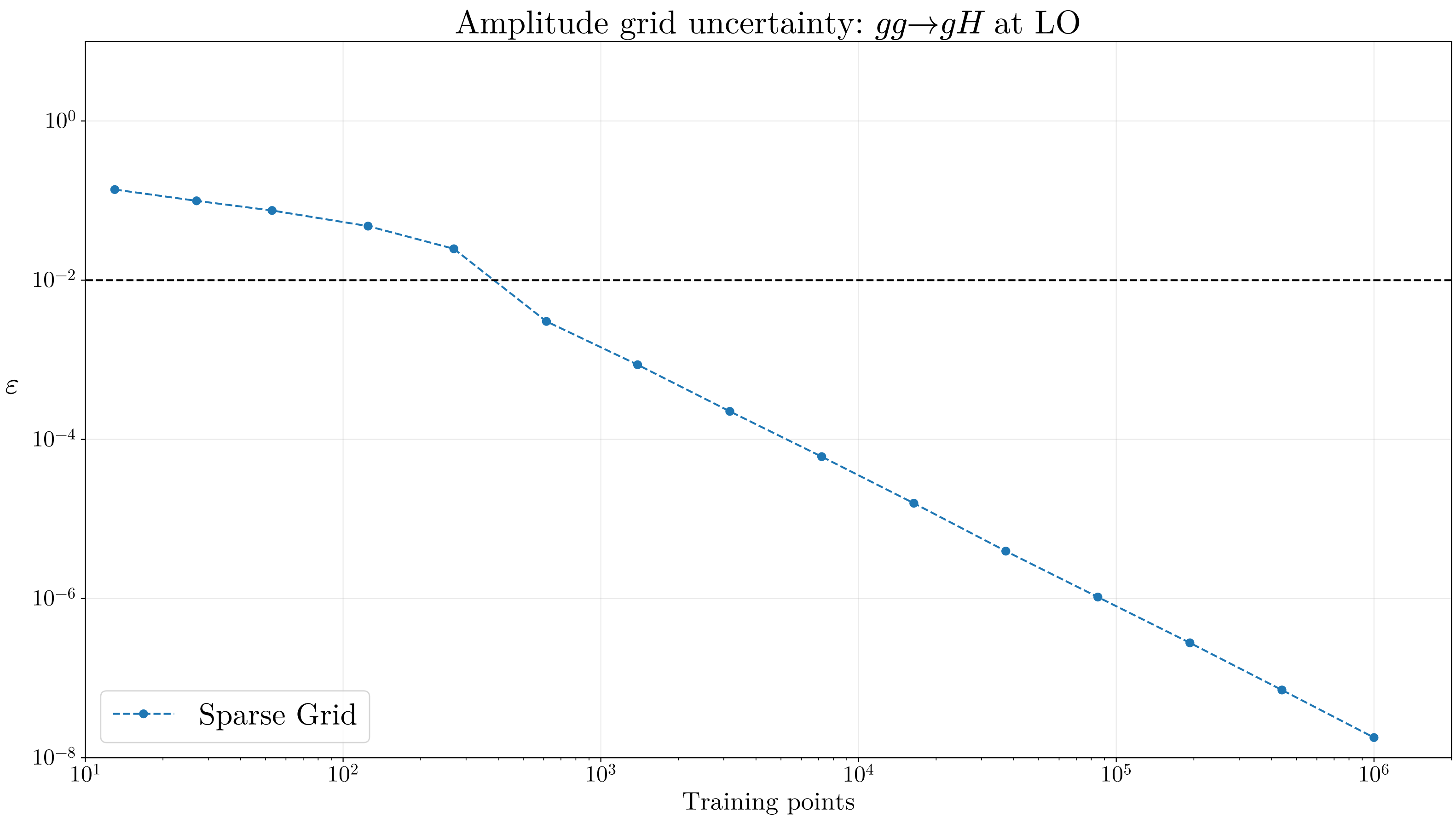
Backup: Selecting an IBP basis

- An IBP basis is not unique, which master integrals should be selected?
- Four criteria: **1.** finiteness, **2.** D-factorising, **3.** fast to evaluate with pySecDec, **4.** simple denominators in IBP coefficients
- Finiteness and D-factorisation is achieved by dimension shifts and dotted propagators
  - d=4,6,8 for most integrals (d=2 for some easy ones with 4 propagators)
  - 2 dots in most sectors, in some lower sectors there are more dots (up to 6 for a three propagator integral)
- Fast evaluation and simple denominators is done through trial and error
  - Generate a basis that fulfills finiteness and D-factorisation
  - Perform reductions while neglecting sub-sectors for sets of master integrals, select the set with smallest denominator factors
  - Benchmark which master integrals in this set are fast to evaluate with pySecDec
  - Repeat the process while restricting the basis to include the fast to evaluate masters

Backup: Training Data

- Interpolation nodes (“data points”) can be selected freely
  - Depends on the interpolation method
- Also: target function can be freely modified, try to get as flat function as possible!
  - Include/exclude phase space factors, PDF weights, flux factor
  - Other flattening techniques such as Korobov transformations?
  - Use symmetries of amplitude to get rid of some peaks
- We are creating *amplitude* grids: Undo all modifications to target function at evaluation
  - This can be difficult if certain integral transformations have been applied

Backup: Low dimensional case



References for software used in ttH

- [1] P. Nogueira, *Automatic Feynman Graph Generation*, J. Comput. Phys. 105 (1993) 279
- [2] B. Ruijl, T. Ueda and J. Vermaseren, FORM version 4.2
- [3] T. van Ritbergen, A.N. Schellekens and J.A.M. Vermaseren, *Group theory factors for Feynman diagrams*,
- [4] V. Magerya, "*Amplitude library (Alibrary): gluing all the tools needed for computing multi-loop amplitudes in QCD and beyond*."
- [5] V. Magerya, *Semi- and Fully-Inclusive Phase-Space Integrals at Four Loops*, Ph.D. thesis, Hamburg U., 7, 2022.
- [6] P. Maierhofer, J. Usovitsch and P. Uwer, Kira—*A Feynman integral reduction program*, Comput. Phys. Commun. 230 (2018) 99
- [7] V. Magerya, *Rational Tracer: a Tool for Faster Rational Function Reconstruction*
- [8] G. Heinrich, S.P. Jones, M. Kerner, V. Magerya, A. Olsson and J. Schlenk, *Numerical scattering amplitudes with pySecDec*, Comput. Phys. Commun. 295 (2024) 108956