A fully numerical approach to perturbative computations in QFT

Valentin Hirschi

CERN

For the position of Group Leader in Numerics

at the

Max Planck Institute for Physics

January 26, 2021

Self.me

2009 **Ph.D.** on the automation of NLO computations

@EPFL, Lausanne supervisor : S. Frixione

2013 **Post-doc #1**

@SLAC, Stanford SNF mobility grants

2017 **Post-doc #2**

@ETHZ, Zurich in the group of C. Anastasiou

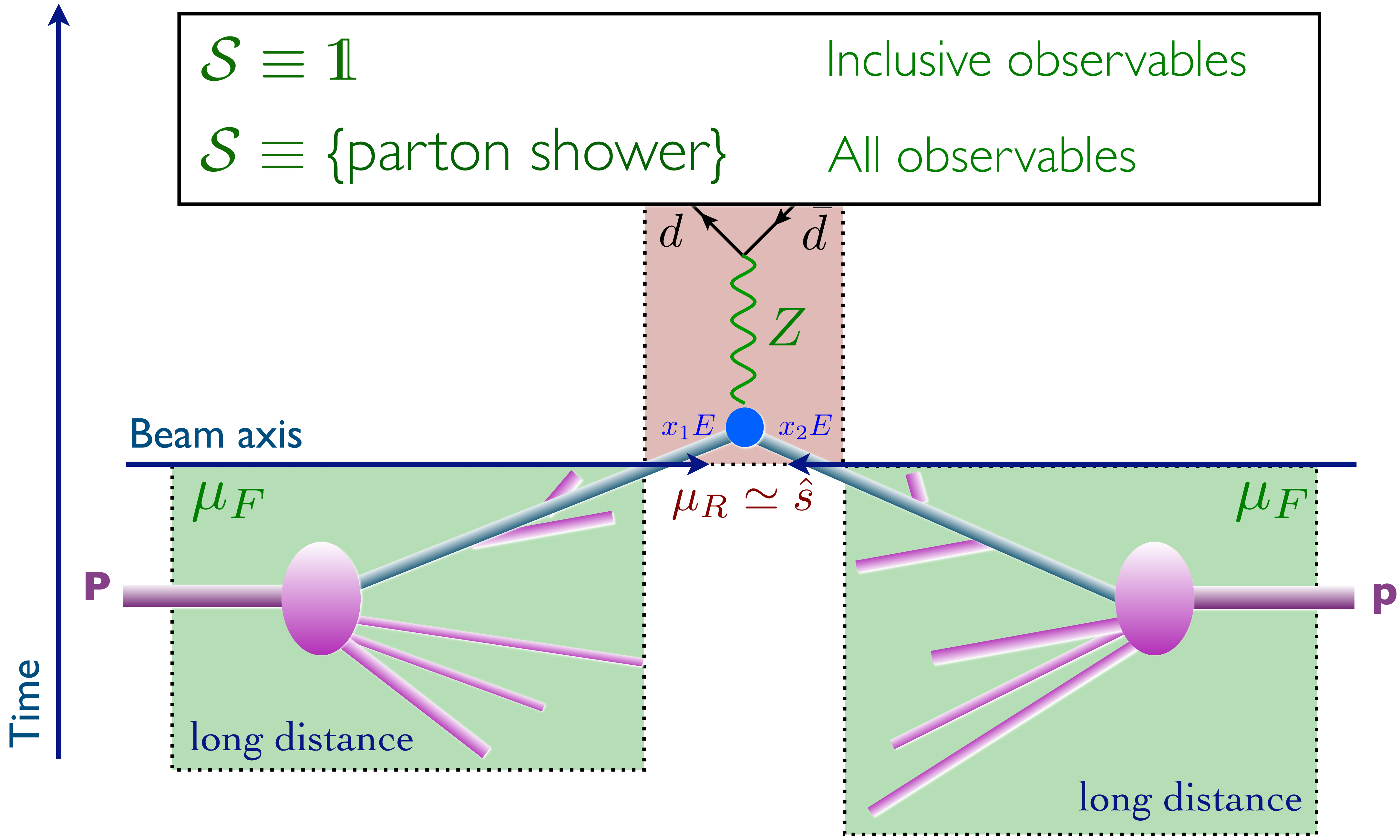
Oct. 2020 **CERN Fellow**

Broad context of my research

$$\sum_{a,b} \int dx_1 dx_2 d\Phi_{\rm FS}\, f_a(x_1,\mu_F) f_b(x_2,\mu_F)\, \hat{\sigma}_{pp\rightarrow\{P\}}(\mu_F,\mu_R,\mu_S)\, \mathcal{S}_{\{P\}\rightarrow\{H\}}(\mu_S)$$

Phase-space integral — Parton density functions — Parton-level differential cross section — Parton evolution operator

$\mathcal{S} \equiv \mathbb{1}$	Inclusive observables
$\mathcal{S} \equiv$ {parton shower}	All observables



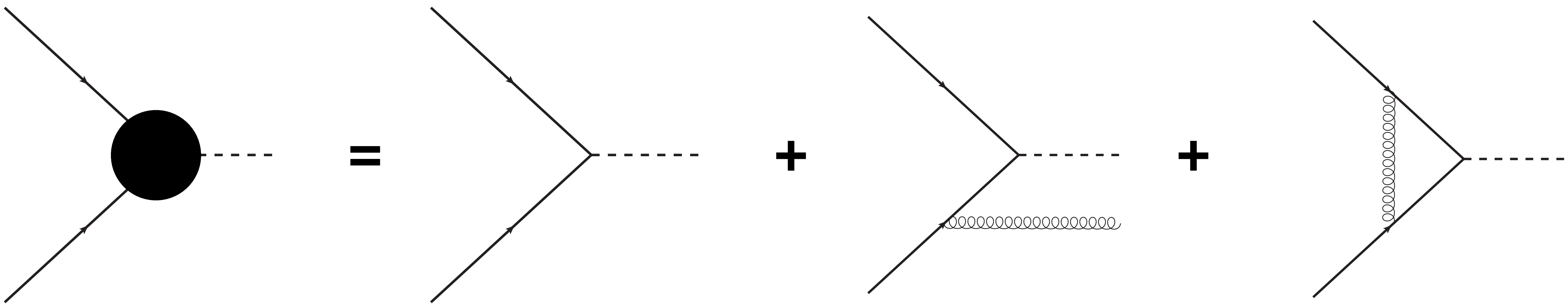
PERTURBATIVE EXPANSIONS

The differential cross section can be written as a perturbation series, using the coupling constant as an expansion parameter :

$$\hat{\sigma} = \sigma^{\text{Born}}\left(1 + \frac{\alpha_s}{2\pi}\sigma^{(1)} + \left(\frac{\alpha_s}{2\pi}\right)^2\sigma^{(2)} + \left(\frac{\alpha_s}{2\pi}\right)^3\sigma^{(3)} + \dots\right)$$

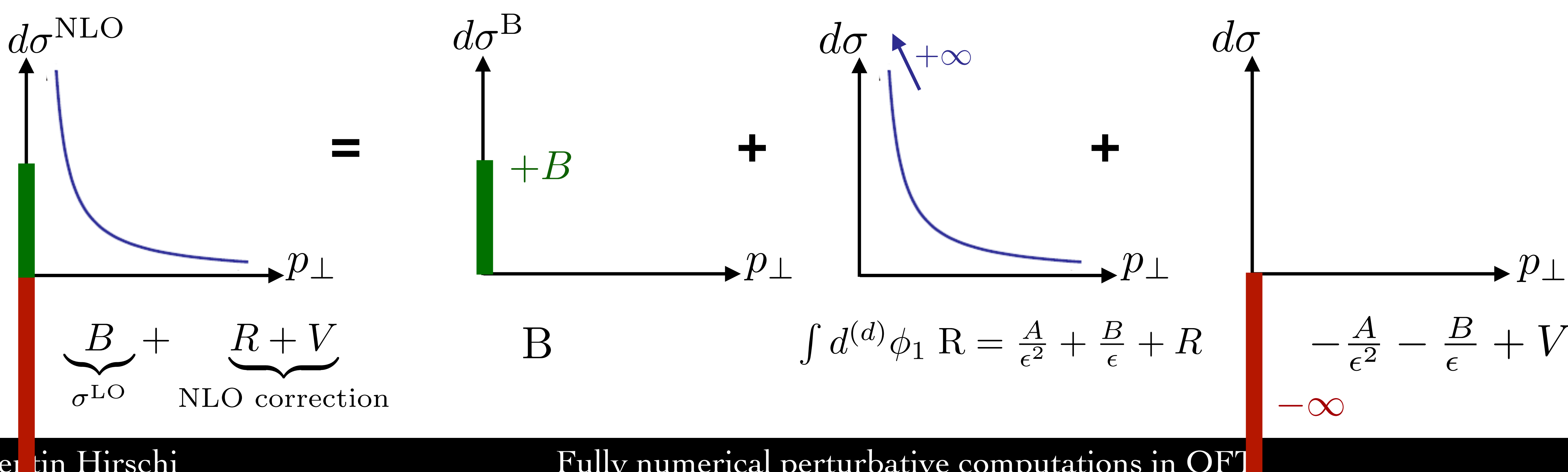
LO predictions	NLO corrections	NNLO corrections	NNNLO corrections
Easy	Difficult but **automated.** My past research MadGraph5	Case-by-case only Automation at this order and beyond is the goal of my current research	pp → V/H only

NLO Anatomy



$$\sigma^{\rm NLO} = \int_m d^{(4)}\sigma^B + \int_{m+1} d^{(d)}\sigma^R + \int_m d^{(4)} \int_1 d^{(d)}\sigma^V$$

NLO inclusive — Born (B) — Real-emission (R) — Virtual (V)



$\underbrace{B}_{\sigma^{\rm LO}} + \underbrace{R+V}_{\text{NLO correction}}$

B

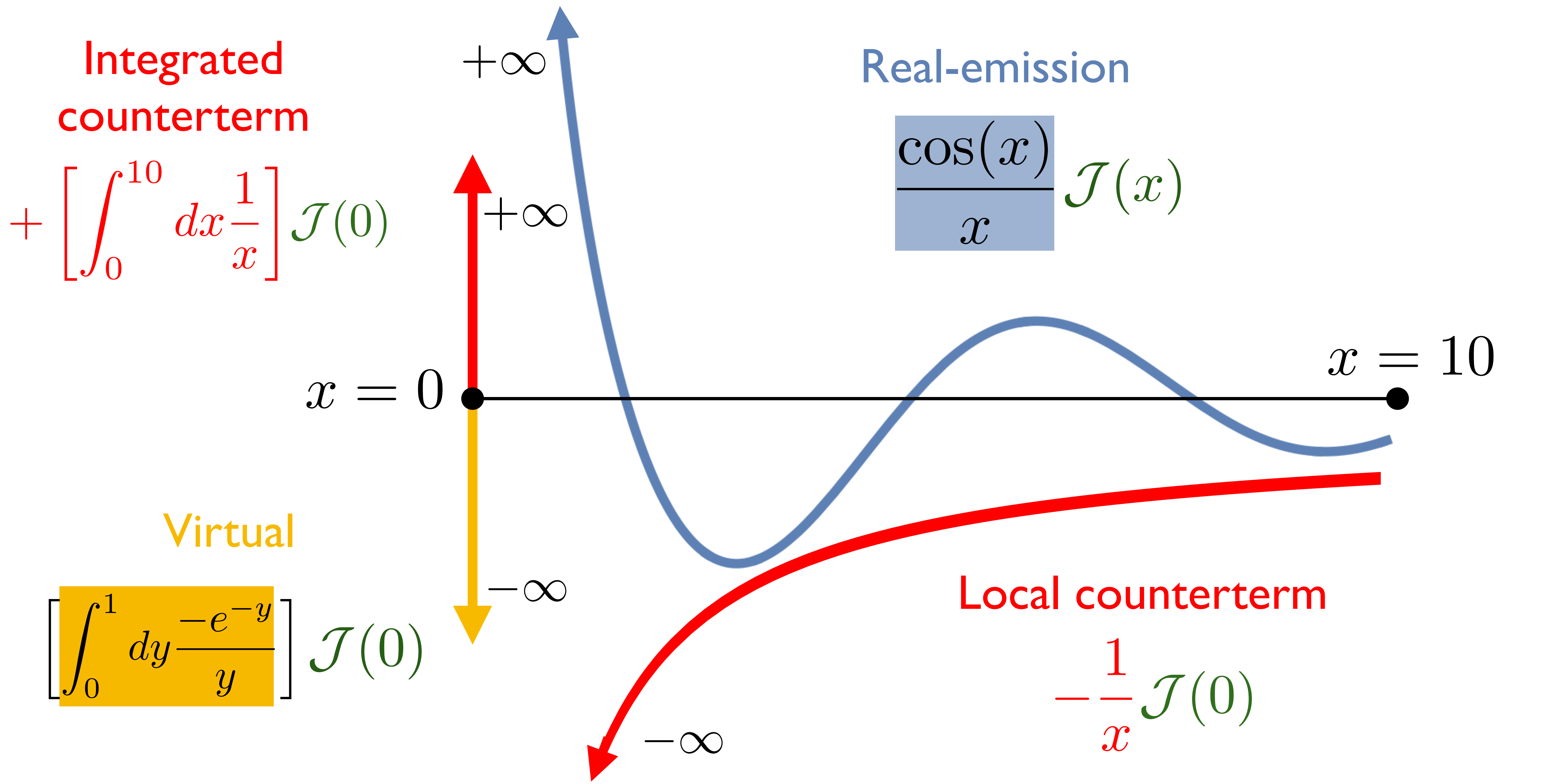
$\int d^{(d)}\phi_1 \, \mathrm{R} = \frac{A}{\epsilon^2} + \frac{B}{\epsilon} + R$

$-\frac{A}{\epsilon^2} - \frac{B}{\epsilon} + V$

One-dimensional toy example

- Toy expression with $\mathcal{J}$ a measurement function, over $x \in [0, 10]$

$$\sigma^{(R+V)}(\mathcal{J}) = \int_0^{10} dx \left[\frac{\cos(x)}{x} \mathcal{J}(x) - \frac{1}{x} \mathcal{J}(0) \right] + \left[\int_0^1 dy \frac{-e^{-y}}{y} \right] \mathcal{J}(0) + \left[\int_0^{10} dx \frac{1}{x} \right] \mathcal{J}(0)$$



Subtraction vs Local Unitarity

- One can introduce the following local (in x) **counterterm**:

$$\sigma^{(R+V)}(\mathcal{J}) = \int_0^{10} dx \left[\frac{\cos(x)}{x}\mathcal{J}(x) - \frac{1}{x}\mathcal{J}(0)\right] + \left(\left[\int_0^1 dy \frac{-e^{-y}}{y}\right] + \left[\int_0^{10} dx \frac{1}{x}\right]\right)\mathcal{J}(0)$$

- And a **regulator** to evaluate the divergent integrals

$$\sigma^{(R+V)}(\mathcal{J}) = \int_0^{10} dx \left[\frac{\cos(x)}{x}\mathcal{J}(x) - \frac{1}{x}\mathcal{J}(0)\right] + \lim_{\epsilon\to 0}\left(\left[\int_\epsilon^1 dy \frac{-e^{-y}}{y}\right] + \left[\int_\epsilon^{10} dx \frac{1}{x}\right]\right)\mathcal{J}(0)$$

- To finally arrive at a **finite** result, differential in $x \in [0, 10]$

$$= \int_0^{10} dx \left[\frac{\cos(x)}{x}\mathcal{J}(x) - \frac{1}{x}\mathcal{J}(0)\right] + \lim_{\epsilon\to 0}\left(\log(\epsilon) + \gamma - \text{Ei}(-1) + \log(10) - \log(\epsilon)\right)\mathcal{J}(0)$$

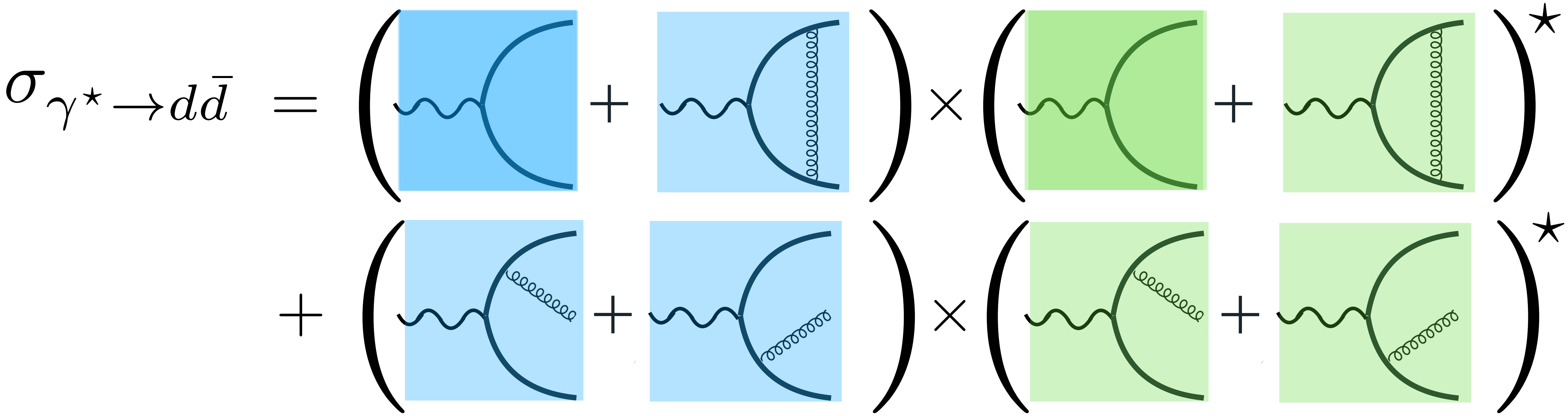
NUMERICAL | ANALYTIC

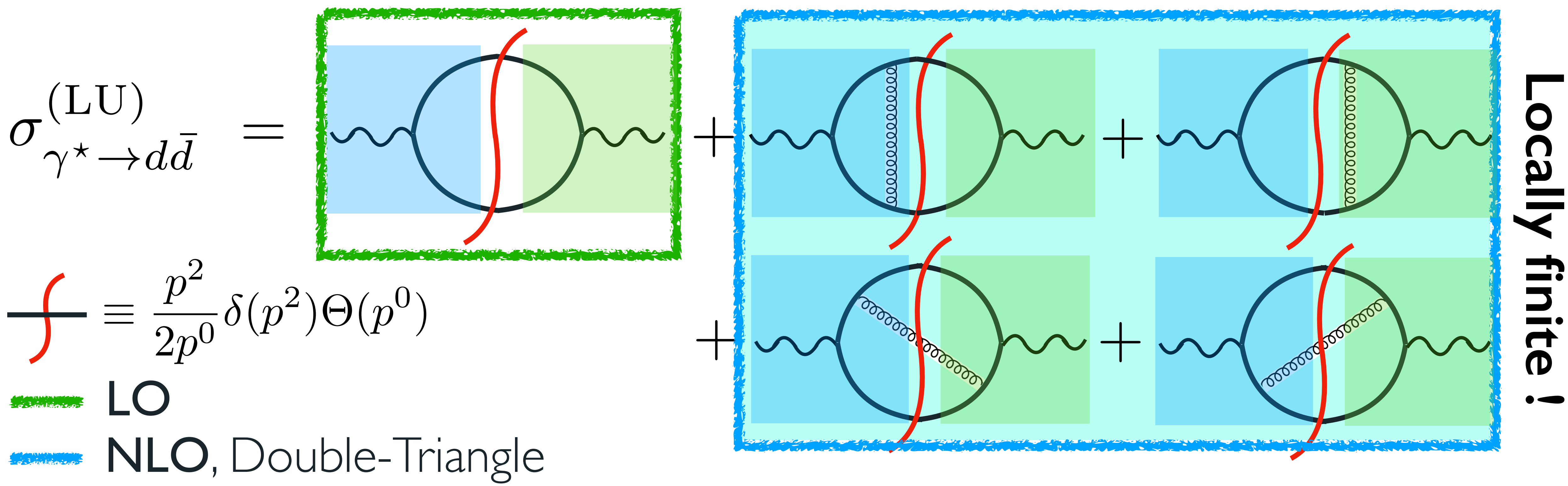
- My research introduces a new paradigm, **Local Unitarity**, whereby :

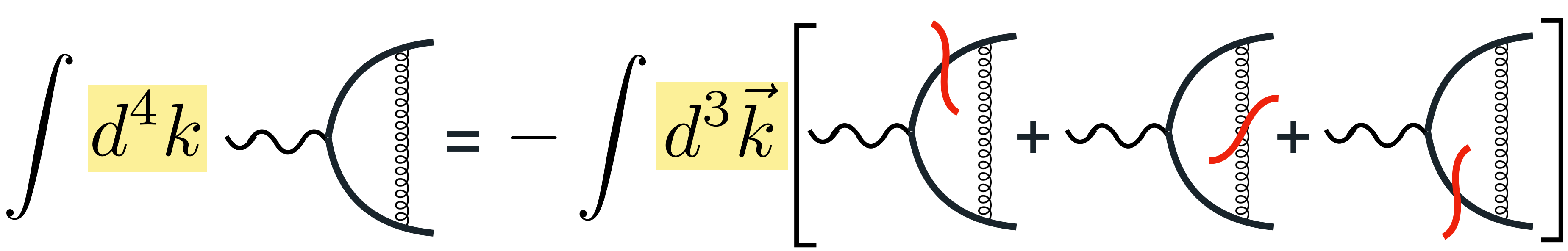
$$\sigma^{(R+V)}(\mathcal{J}) = \int_0^{10} dx \left[\frac{\cos(x)}{x}\mathcal{J}(x) + \frac{-e^{-x}}{x}\mathcal{J}(0)\Theta(1-x)\right]$$

Local Unitarity: supergraphs

[Z. Capatti, VH, A. Pelloni, B. Ruijl, arXiv : 2010.01068]   [Also more details in my recent CERN Seminar talk]







Loop-Tree Duality

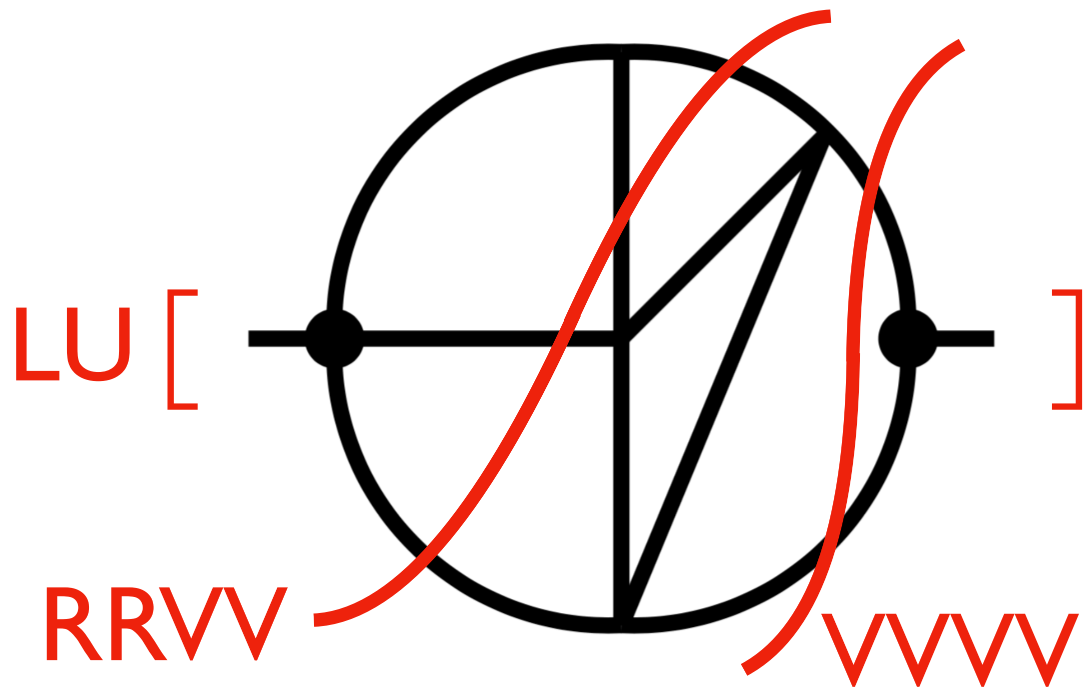
[arXiv: 1906.06138]
[arXiv: 1912.09291]
[arXiv: 2009.05509]

Numerical results from αLoop

[Capatti, VH, Pelloni, Ruijl, arxiv: 2010.01068]

Piece of the "N⁴LO" correction to $\phi^\star \to \phi\phi$ within $\{\phi^3, \phi^4\}$-theory

One individual **5-loop supergraph** :



$$= \frac{5\pi}{(16\pi)^5}\frac{441}{40}\zeta(7)$$

[**FORCER** B. Ruijl, T. Ueda, J. Vermaseren arxiv: 1704.06650]

[**R*** F. Herzog, B. Ruijl arxiv: 1703.03776]

without **contour deformation** (doing it differentially would have required one)

Γ	N_p [10^6]	t/p [μs] min	t/p [μs] avg	N_{ch}	FORCER [GeV²]	αLoop [GeV²]	exp.	Δ [σ]	Δ [%]
c.3	1	1600	69000	130	1.77832	1.7797(33)	-9	0.42	0.00077

SOFTWARE AND COMPUTING ACHIEVEMENTS I

MadGraph5_aMC@NLO

[arXiv: 1103.0621 first NLO QCD automaton]

[arXiv: 1405.0301 each year in top-ten hep cite]

C++ / Python / Fortran

Core contributor: ~100k lines

[2009 - 2017]

- Experience with development in large collaboration and iterative design (Agile coding)
- Numerous users with different technical skills and needs:
  - Automated code with a high-level of abstraction with user-friendly interface
  - Frequent releases with robust testing suite (unit, acceptance, parallel, I/O tests,...)
  - Accommodate multiple computing infrastructures (slurm, LSF, condor, multicore, MPI,...)
  - Automation of NLO EW corrections and MSSM (CMS, photon isolation, multi-couplings,...)
- Widely used by experiments, so that efficiency matters:
  - Advanced parallelisation paradigms (multi-channels, work-stealing, async,...)
  - Optimised algorithms (MadLoop, diagram generation, phase-space generation,...)
  - Variance minimisation in Monte-Carlo (adaptive sampling, dynamic approximations,...)
- Expertise in interfacing with a vast ecosystem of HEP softwares:
  - Loop computations (GoSam, Recola, OpenLoops, COLLIER, Ninja, QCDLoop, LoopTools, OneLOop,...)
  - Event generation (FastJet, Pythia8, LHAPDF, Vegas3, Cuba, reweighting, MadSpin, MadOS, biasing,...)
  - Collider analysis (HepMC, Rivet, MadAnalysis, HwU,...)
- Organised tutorials at many schools and workshops (TASI, CERN, MG schools,...)

Software and computing achievements II

MadN^{k}LO

[arXiv: 2010.09451, 1902.10167 private code]

Python / Rust

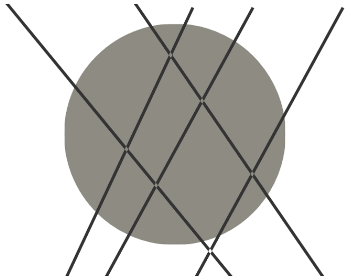
MG5aMC for beyond-NLO predictions
Lead developer

[2016 - now]

- Higher level abstraction suited to beyond-NLO subtraction formalisms:
  - Flexible customisation of each fixed-order piece (NLO loop-induced,…)
  - Generalised PS supports and process generation (RRR, RVV, ISR convolutions,…)
  - Multi-loop through effective vertices in UFO models (WAMP, gg>Hg 2-loop EW,…)
  - Containerisation of dependencies in Docker images for ease of deployment
  - Integrator abstracted away, opening up to ML-based approach (iFlow, Zünis,…)
  - Generalised color and spin correlators to arbitrary perturbative orders
  - Interfaced multi-loop computation tools (Chaplin, GINAC, pySecDec,…)

While very **useful**, this work is not addressing the root cause of **complexity growth** of fixed-order predictions: IR singularities

Software and computing achievements III



α-Loop

[arXiv: 1906.06138, 1912.09291, 2009.05509, 2010.01068 private code]

Python / Rust / C / FORM / Mathematica

Fully numerical computation of cross-sections

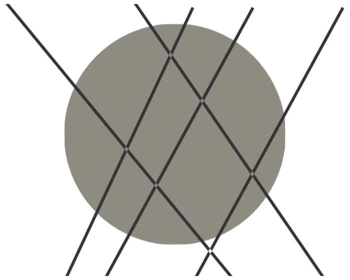
Co-leader of a team of junior scientists

[2019 - now]

- High loop count algorithms (up to six loops):
  - Heavy graph manipulations (QGRAF, isomorphisms, spanning trees, BPHZ forest, topology reduction,…)
  - Efficient code meta-generation (Fast Lorentz and colour algebra, polynomial optimisation in FORM,…)
  - Generalised Loop Tree Duality algorithms in Python standalone codes

- Contour deformation:
  - Solving NP hard problem with custom domain-specific heuristics
  - Conic solvers for numerical optimisiation under convex constraints (ECOS, SCS,…)
  - Automatic differentiation using dual numbers

Software and computing achievements III



α-Loop

[arXiv: 1906.06138, 1912.09291, 2009.05509, 2010.01068 private code]

`Python / Rust / C / FORM / Mathematica`

Fully numerical computation of cross-sections

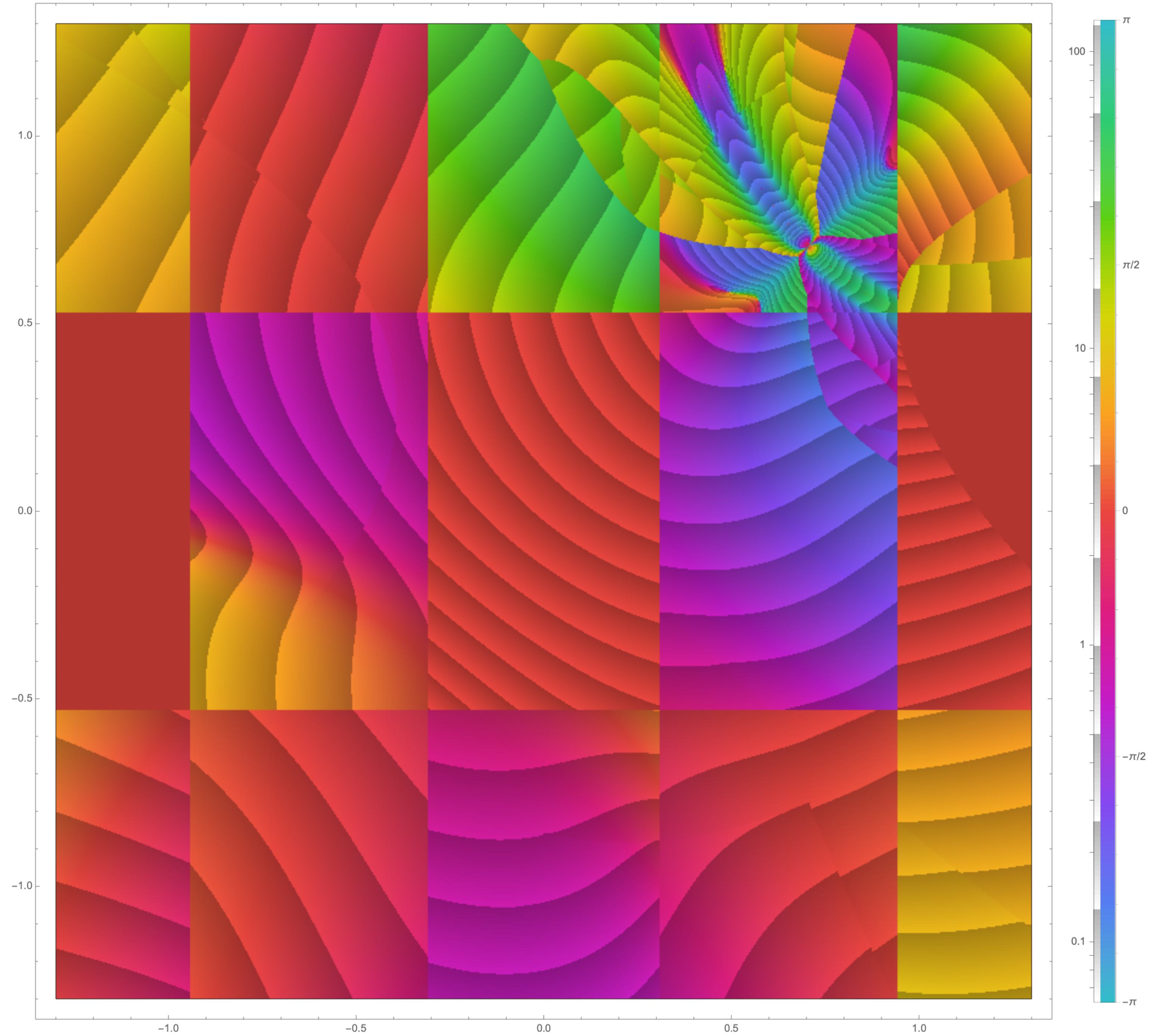
Co-leader of a team of junior scientists

[2019 - now]

- Numerical stability:
  - Rescue pipeline involving f128 arithmetics and a new more stable cLTD representation
  - Automated Python sampler of all UV limits using Rust-Python bindings
- Advanced Monte-Carlo and parallelisation techniques
  - New nested adaptive sampling involving multiple discrete and continuous dimensions
  - Semi-autonomous search for optimal hyperparameters distributed over clusters
  - Custom parameterisations and multi-channeling for taming integrable singularities
- Analytic multi-(up to three)loop vacuum integrals
  - Integration by part identities with high rank numerator reduction
  - Consistent combination with $\overline{\mathrm{MS}}$ NNLO renormalisation

Differential Local Unitary Integrand

$$e^+e^- \rightarrow \gamma^\star \rightarrow d\bar{d}$$



" Cut what you cannot compute "