technische universität
dortmund

IMAPP Master Thesis Project

Enhancing the efficiency of event generation with MCMC and machine learning techniques

Cornelius Grunwald

cornelius.grunwald@tu-dortmund.de

TU Dortmund University, Germany

Motivation - Improving Sampling Efficiency

- high-energy physics heavily relies on simulated events ⇒ Monte Carlo simulations
- we need high-statistic samples to precisely investigate tails of distributions and more complex final states
- MC simulation efficiency and speed need to improve for precision era, e.g. for HL-LHC (factor ~25 more simulated data required)
- multiple efforts made such as ML, nested sampling, MCMC sampling

[Yallup et al. 2205.02030]
[Danziger et al. 2109.11964]
[Kröninger et al. 1404.4328]

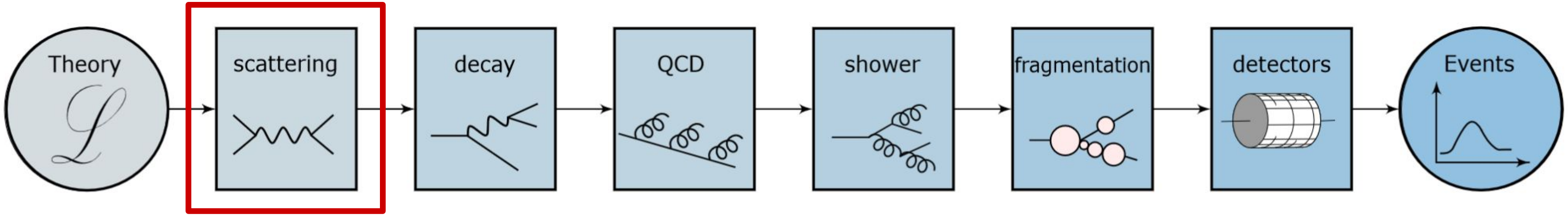
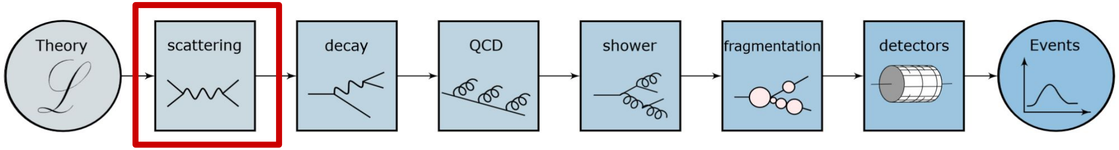


Image: **Machine learning and LHC event generation,** Butter et al., **10.21468/SciPostPhys.14.4.079**, SciPost Physics 14 (2021)

The Challenge - Expensive event generation



Computational bottleneck: the hard scattering component

$$\sigma_{pp\to X_n} = \sum_{ab} \int \mathrm{d}x_a \mathrm{d}x_b \, \mathrm{d}\Phi_n \; f_a(x_a, \mu_F^2) f_b(x_b, \mu_F^2) \, |\mathcal{M}_{ab\to X_n}|^2 \, \Theta_n(p_1, \ldots, p_n)$$

Difficulty:

- $|\mathcal{M}|^2$ is typically multi-modal, wildly fluctuating & computationally expensive

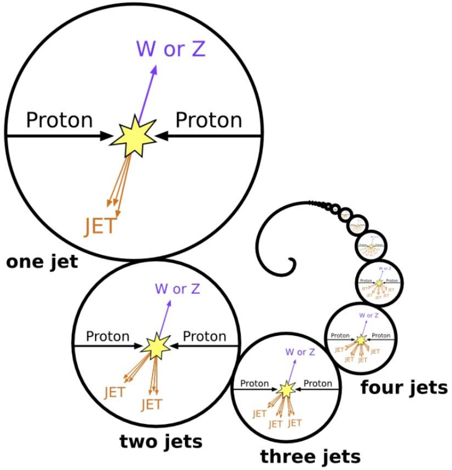
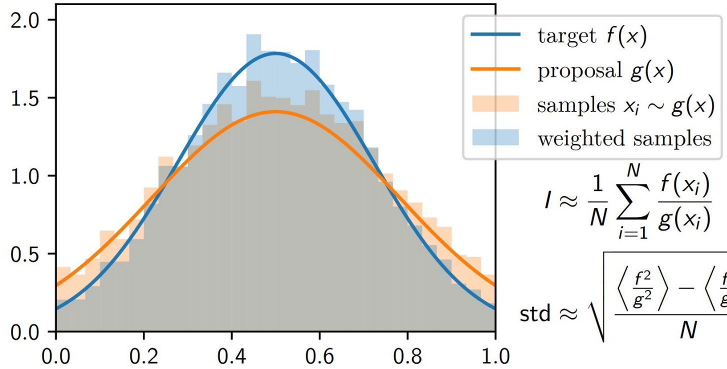


Image by Jim Pivarski
https://www.fnal.gov/pub/today/images/images12/figure.jpg

Sherpa

- MC event generator for collision events
- user-friendly configuration files for selecting processes and setting cuts
- main sampling method: importance sampling within physics-informed channel mappings



$$I \approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{g(x_i)}$$

$$\text{std} \approx \sqrt{\frac{\left\langle \frac{f^2}{g^2} \right\rangle - \left\langle \frac{f}{g} \right\rangle^2}{N}}$$

SHERPA

https://sherpa-team.gitlab.io
[Bothmann et al., SciPost Phys.7 (2019)]

Sherpa.yaml

```
TAGS: {
  MCUT:  66.0,
  NJETS: 3,
  PTMIN: 20.0
}

BEAMS: 2212
BEAM_ENERGIES: 6500.

EVENTS: 100000

PROCESSES:
- 21 21 -> 11 -11 1 -1 21:
    ME_Generator: Amegic
    Order: {QCD: Any, EW: 2}

SELECTORS:
- [Mass, 11, -11, $(MCUT), E_CMS]
- NJetFinder:
    N: $(NJETS)
    PTMin: $(PTMIN)
    R: 0.4
    Exp: -1

```

Rambo & Multichannel Mappings

- task: generate four-momenta of incoming & outgoing particles from random numbers
- need to fulfill constraints like energy conservation & on shell conditions
- RAMBO mapping: [1308.2922]

$$\mathrm{d}\Phi_n(P, p_1, \ldots, p_n) = \prod_{i=1}^{n} \frac{\mathrm{d}^3 p_i}{(2\pi)^3 2E_i}(2\pi)^4 \delta^4\left(P - \sum_{i=1}^{n} p_i\right) \qquad d = 3n - 4$$

- Multichannel interface:

$$\boxed{g(x) = \sum_{i}^{N_c} \alpha_i g_i(x)\,, \quad \sum_{i}^{N_c} \alpha_i = 1}$$

  - use mixture distribution for multimodal targets
  - construct channels based on physics knowledge
  - automatic channel weight optimization

The Bayesian Analysis Toolkit - BAT.jl

- collection of state-of-the art algorithms for Bayesian data analysis in Julia
- focusing on **efficiently sampling distributions** (particularly via MCMC)
- not relying on a specific modelling language / domain specific language
- provides modern sampling approaches & new algorithms

BAT.jl
Bayesian Analysis Toolkit
https://github.com/bat/BAT.jl

user-specified:

- target (likelihood & data)
- parameters & prior

provided by BAT.jl:

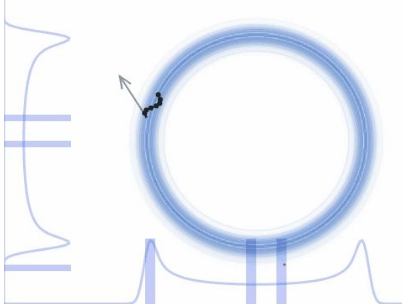
- sampling algorithms
  - **MCMC sampling**
  - Nested Sampling
- integration algorithms
- optimization algorithms

automated posterior exploration

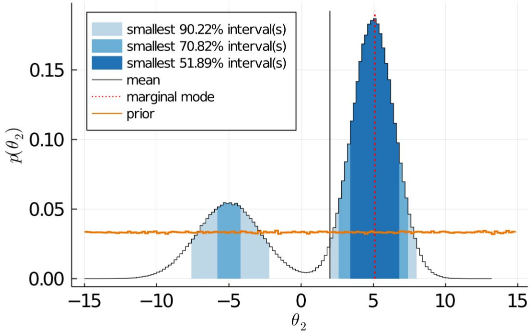
(tuning, parameter space transformations, parallelization, ...)

outputs

- samples
- plots
- modes, mean values, intervals

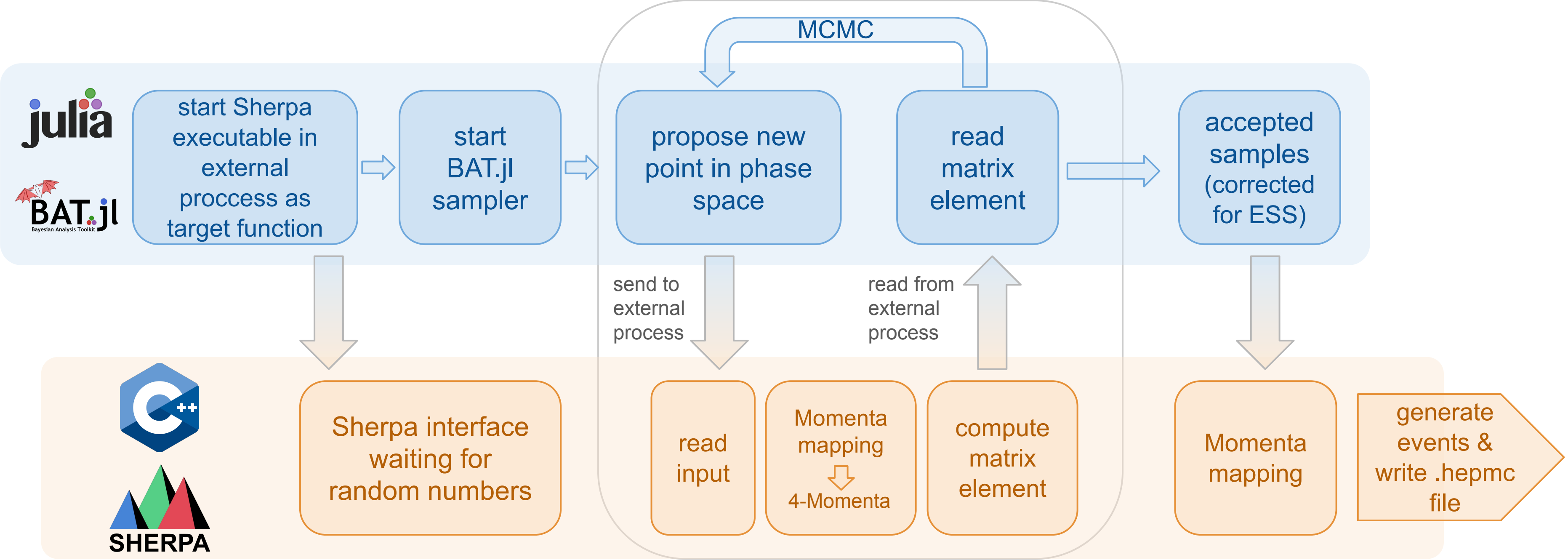


https://github.com/chi-feng/mcmc-demo



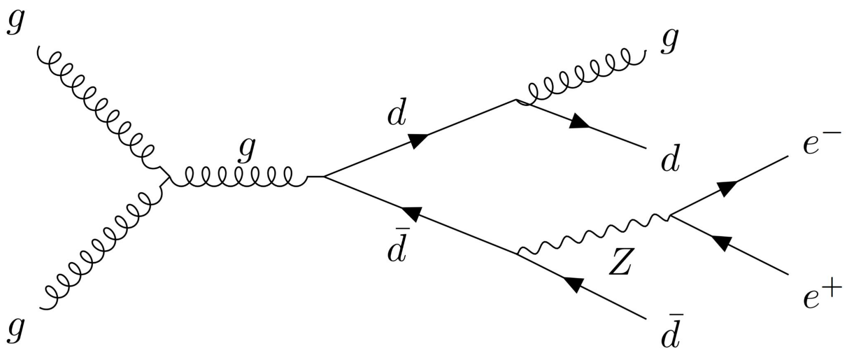
The BAT.jl - Sherpa Interface

Current interface: Run BAT.jl and call Sherpa as the target distribution



Example Process: Z + 3 Jets

Z+3jets : **$g\ g \rightarrow d\ d\ e^+\ e^-\ g$** @ *13GeV* pp collisions



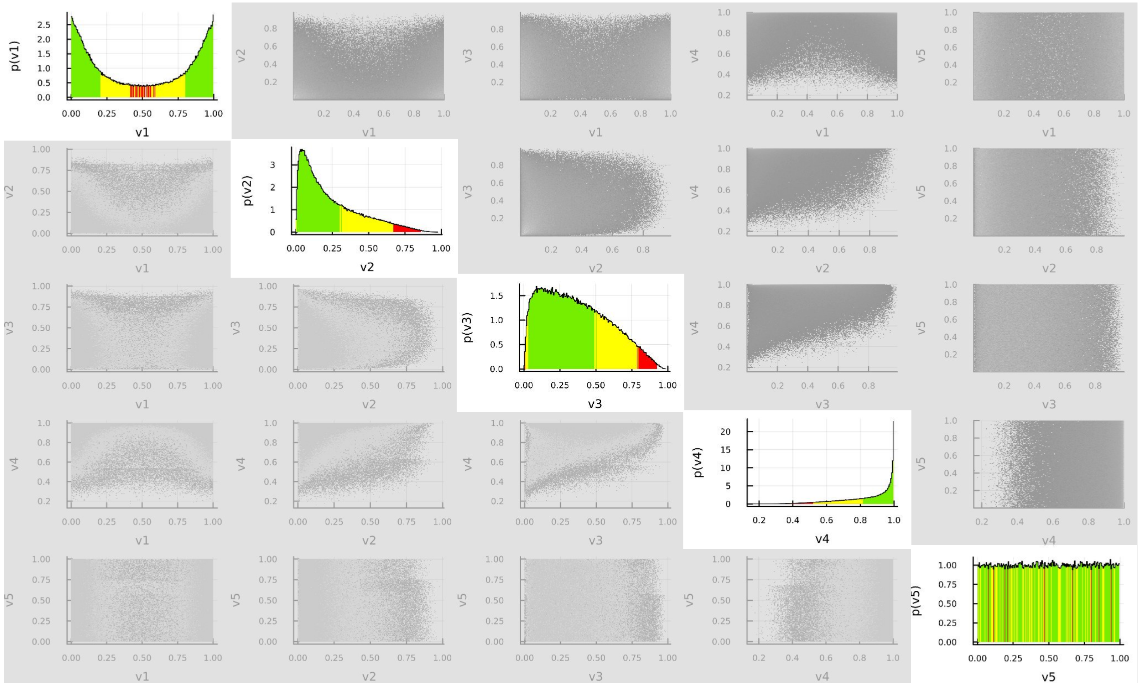
2 parameters for the **incoming** momenta fractions

11 parameters for the momenta of the **5 outgoing particles**

⇨ 13 dimensional sampling space

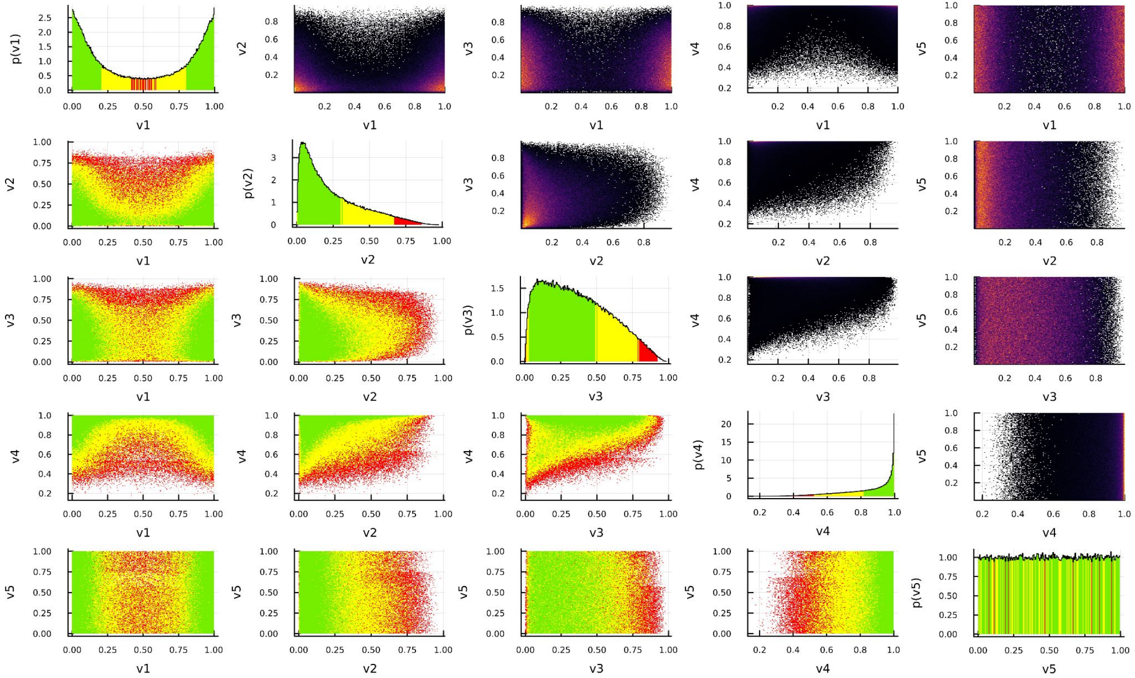
Phase space when sampling in a selected channel (1D)

- one dimensional marginalized distributions of samples
- shown first five parameters of phase space
- abstract parameter space
- wide variety of shapes



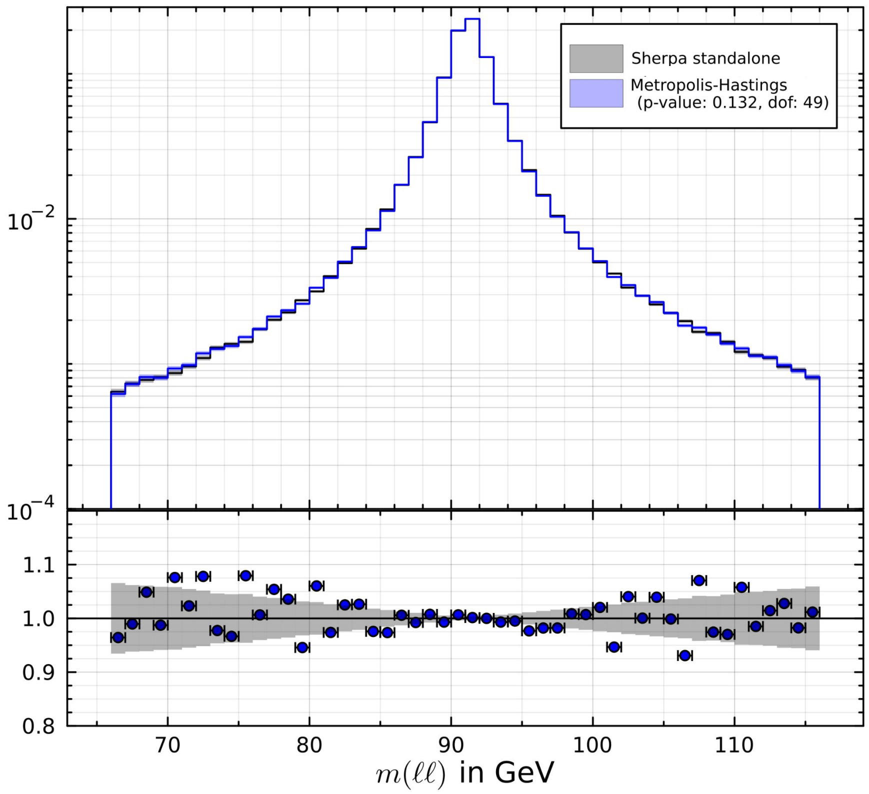
Phase space when sampling in a selected channel (2D)

- one and two dimensional marginalized distributions of samples
- shown first five parameters of phase space
- abstract parameter space
- wide variety of shapes

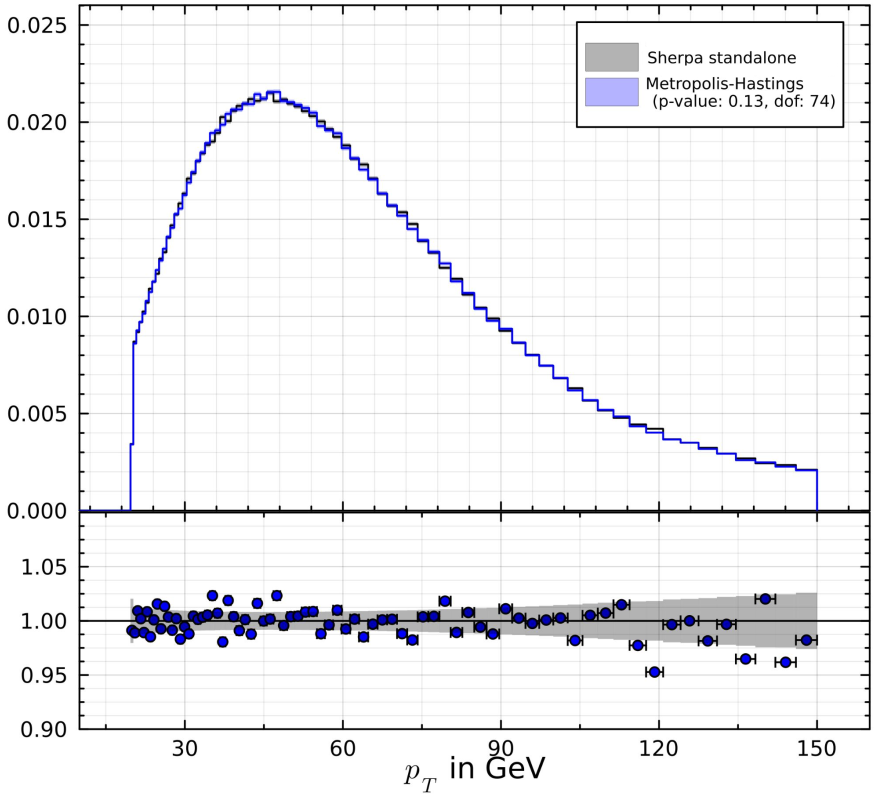


Physical Observables

dilepton mass

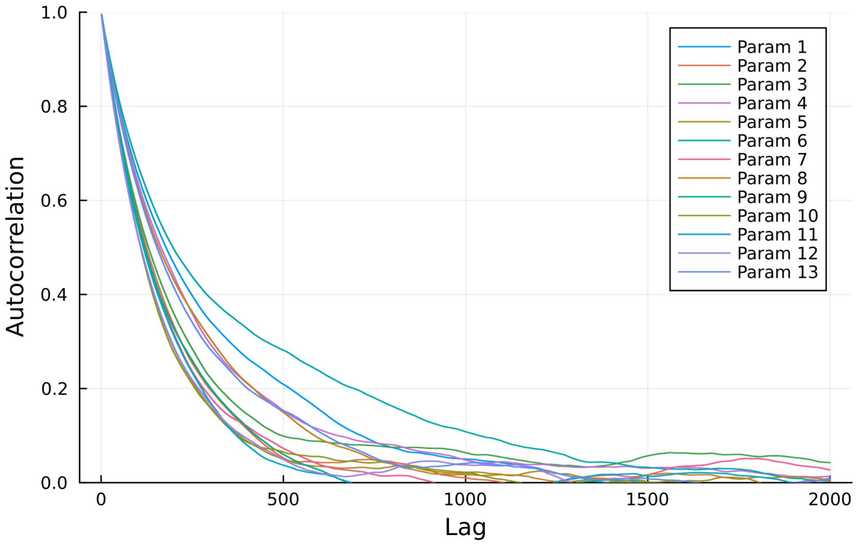


Lepton pT



MCMC - Autocorrelation

- events generated by MCMC methods are not independent
- **autocorrelation** plots allow to visualize this effect
- effective sample size (ESS) can be used to account for correlated samples

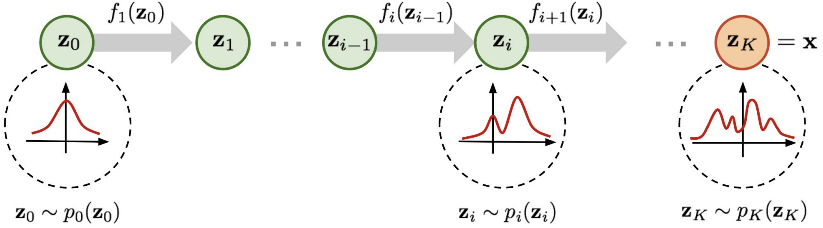


open problems:

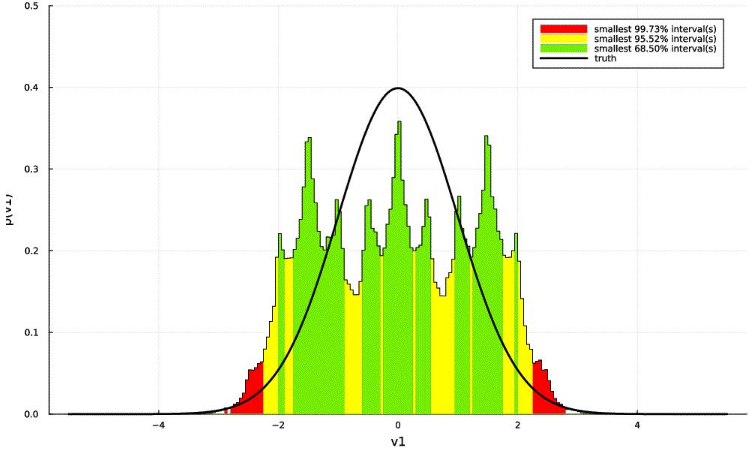
- need to reduce autocorrelation to improve sampling efficiency
- test interface on more complex final state

ML enhanced MCMC sampling

- improving the performance of high-dimensional sampling by combinin MCMC & ML methods ⇨ normalizing flow enhanced MCMC



- learn a normalzing transformation from MCMC samples by training a NN



- test this sampling approach for sampling Sherpa processes

Master project - Possible Roadmap

- learn to use the BAT.jl-Sherpa interface
- investigate more complex final state examples
- trest new sampling algorithms / strategies
- test normalizing flow enhanced MCMC sampling on example processes

- computing & coding heavy project -> working on the computer
- basic knowledge of statistics & MC/ML methods would be helpful
- programming languages: C++ & Julia (& python)

If you are interested, feel free to contact me: cornelius.grunwald@tu-dortmund.de