Phenomenology of nucleon tomography

Paweł Sznajder
National Centre for Nuclear Research, Poland

November 30th, 2022

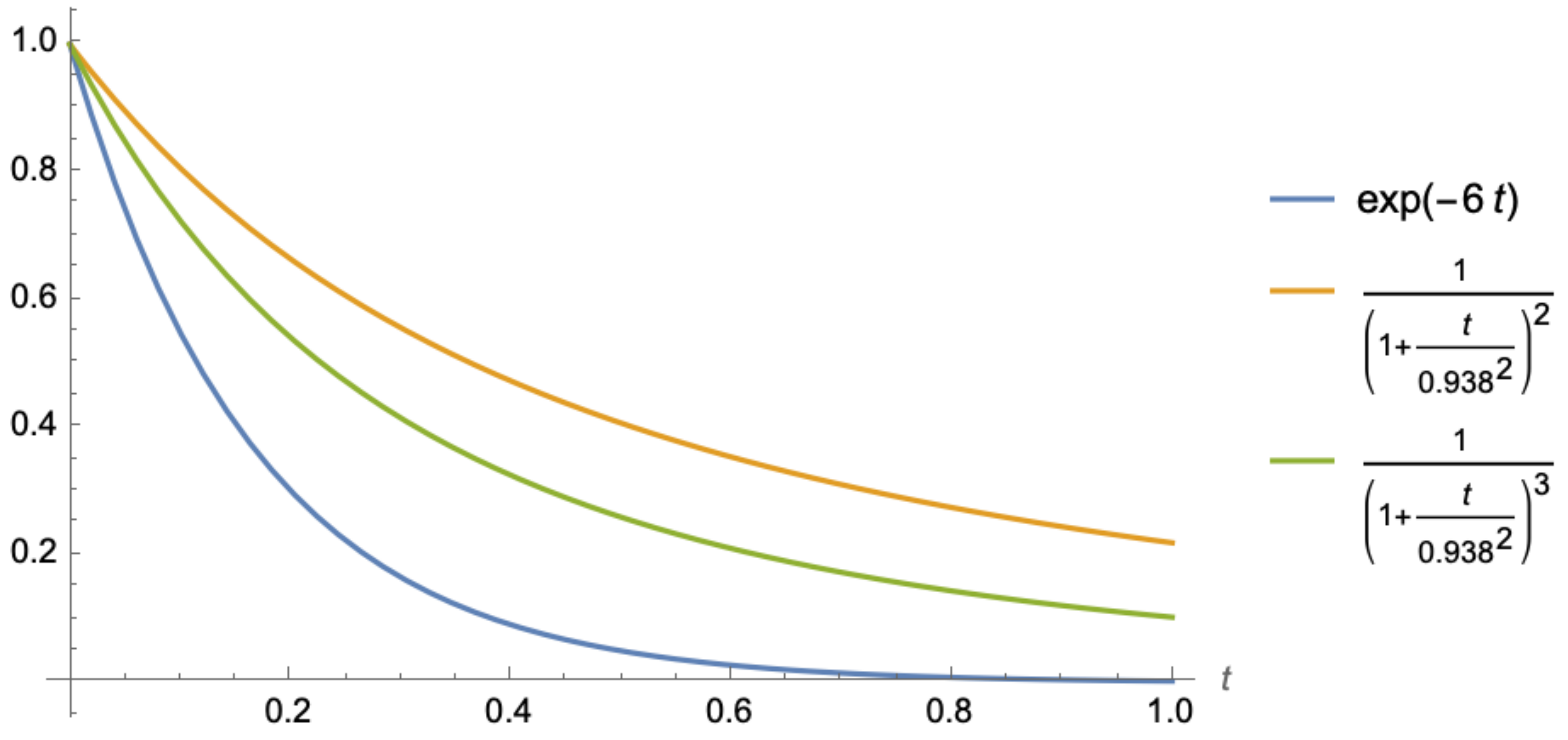
Motivation

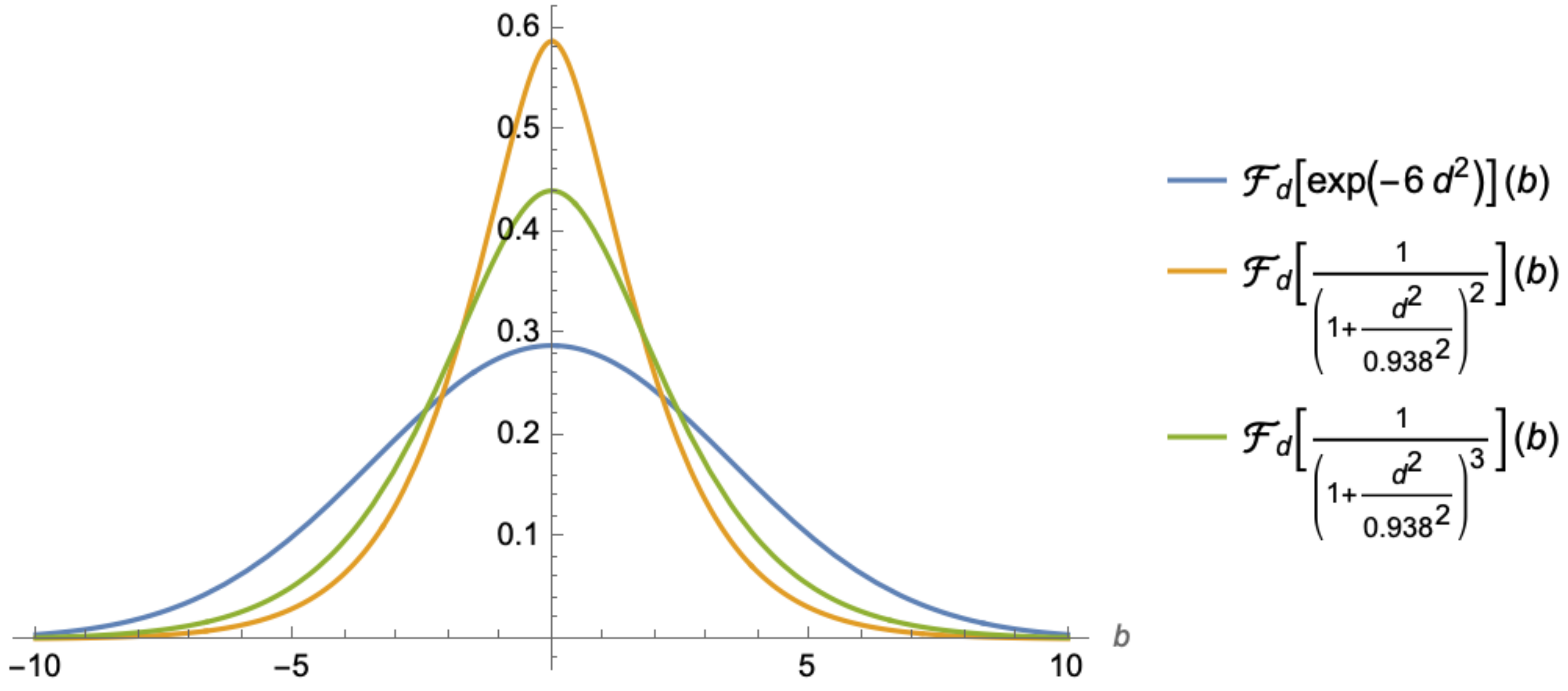
- At least two prescription of building t-parameterisation exist

$$f_{\text{exp}}(t) = \exp(bt)$$

$$f_{\text{ipole}}(t) = (1 - t/\Lambda^2)^{-\alpha}$$

- Both prescriptions give different FTs

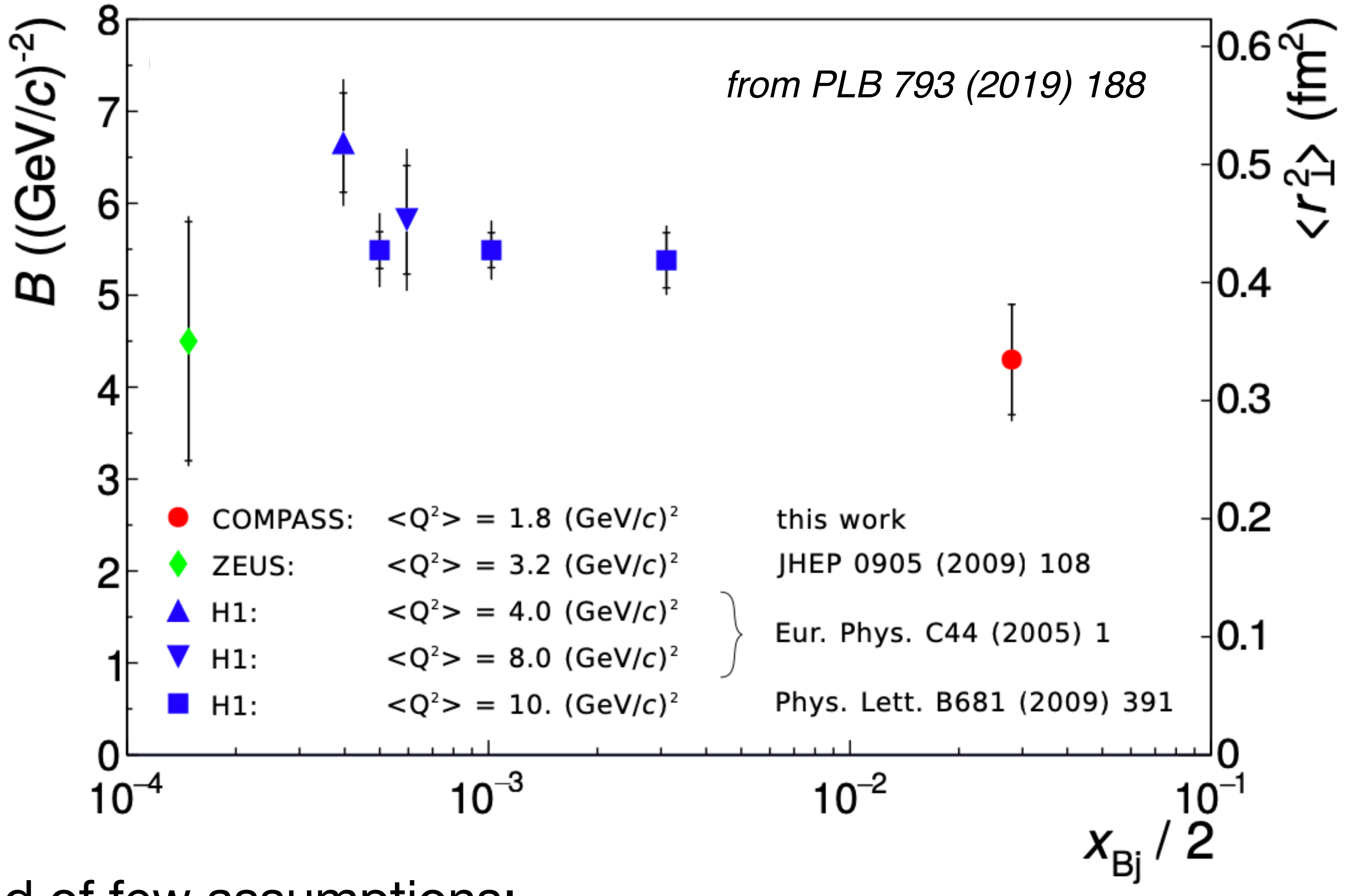




- In real applications these prescriptions are used to build parameterisations involving various kinematic dependent factors (e.g. b-slope)
- Nature can give us quite different shapes

Motivation

- HERA and COMPASS measurement of "transverse extension of partons"



- This method depend of few assumptions:
  - dominance of GPD H
  - negligible skewness effect: $\sigma \propto H(x, 0, t)^2$
  - $f_{exp}(t)$ prescription applicable → quarks and gluons (suppressed at LO) have the same slope, no HT corrections
- We will work on the last assumption
- Note practical problem: if Nature has decided on e.g. $f_{ipole}(t)$ prescription how to describe it with single slope parameter?

Goal

- The goal is to create the following framework:
  you give me experimental t-distribution, I give you proper FT
- Useful for description of data, but also for estimation of model uncertainties
- Useful for EIC impact studies!

Implementation

- We need a flexible prescription for modelling of t-dependance

- Let us use exponential polynomial function, used e.g. :

$$f_{\mathrm{flex}}(t) = \frac{\sum_i (w_{0,i} + w_{1,i}t)\exp(b_i t)}{\sum_i (b_i w_{0,i} + w_{1,i})/b_i^2} \qquad b_i > 0$$

- When used on histogram of events similar to real Fourier discrete decomposition

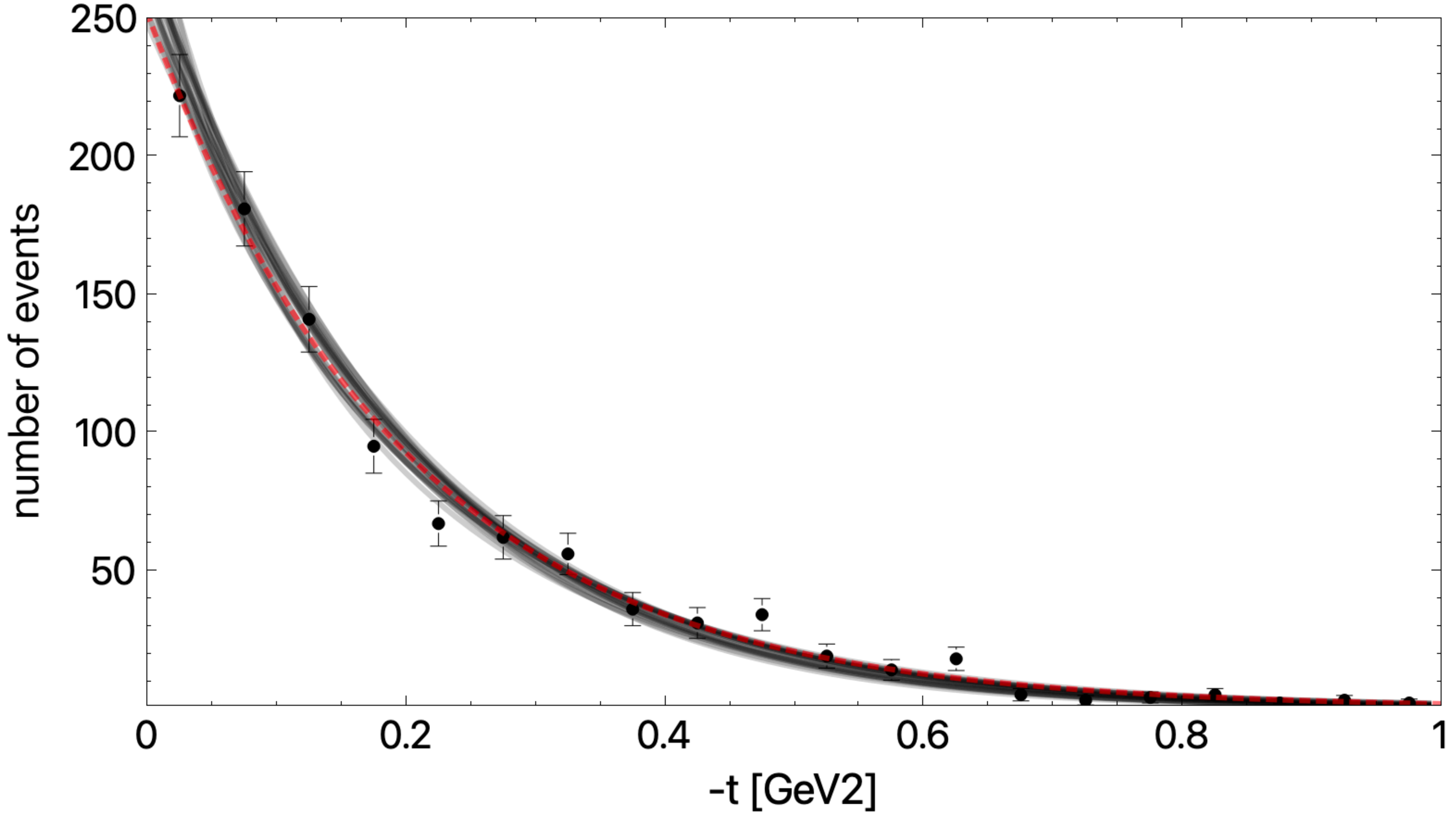
Implementation

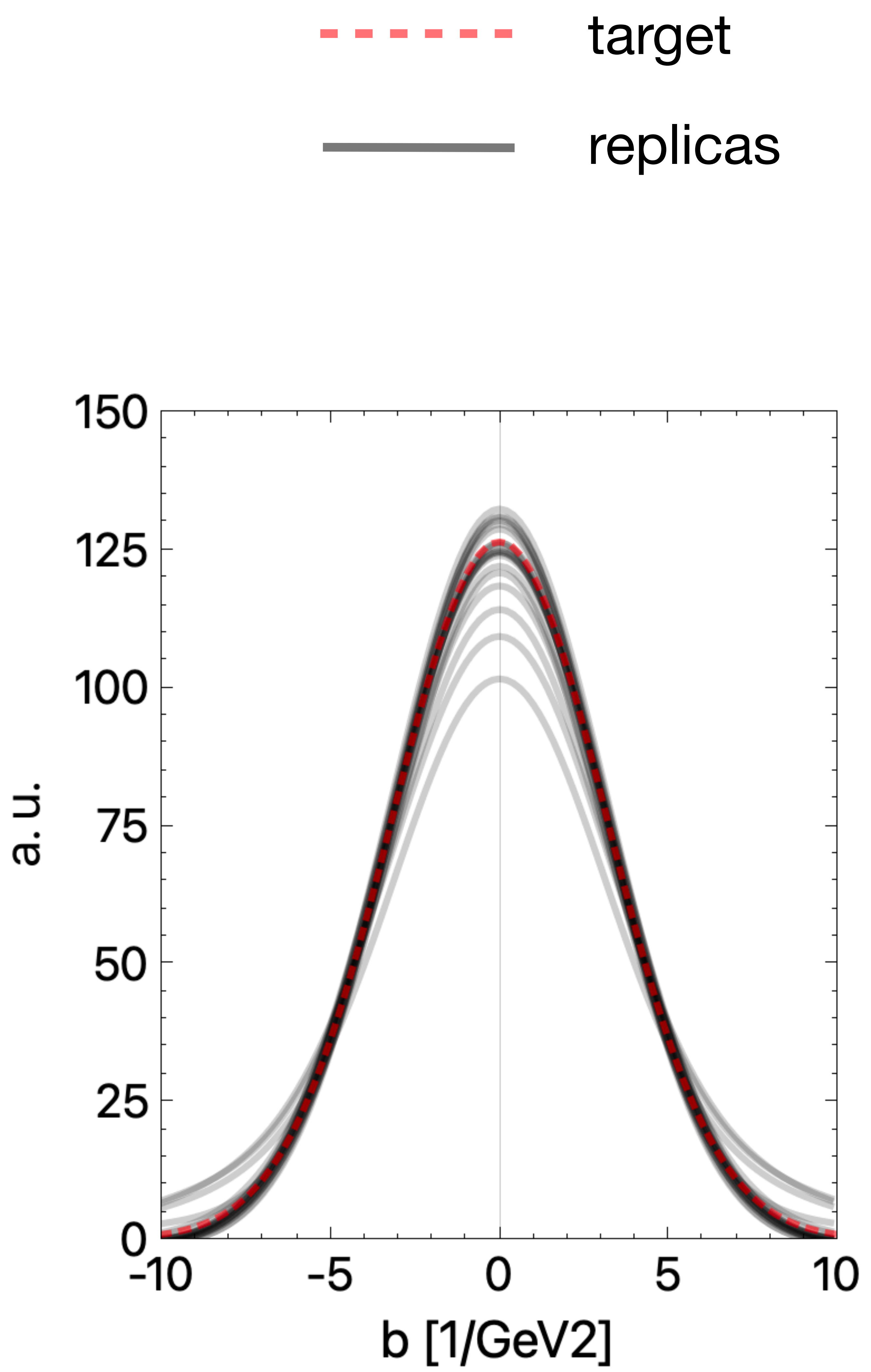
- Tune truncation parameter (i.e. number of elements in the sums) so that $\chi^2/\mathrm{ndf} \approx 1$
- Eventually, since we deal with many free parameters one can use techniques known from ML, e.g. dropout regularisation:
  - take large truncation parameter
  - randomly disable elements in each iteration of the minimisation with probability p
  - tune p so that $\chi^2/\mathrm{ndf} \approx 1$
- Both methods are not equivalent: the second one gives more flexible parameterisation (due to more free parameters) which influences e.g. propagation of uncertainties to unfitted regions

Demonstration

- Generate pseudo-data according to given probability distribution
- Fit histogram with $f_{flex}(t)$ prescription
- 10 elements in the sums, $p = 0.2$ (dropout probability - tuning required)
- χ^2 as FCN
- genetic algorithm for minimisation
- replication for uncertainty estimation (20 replicas)

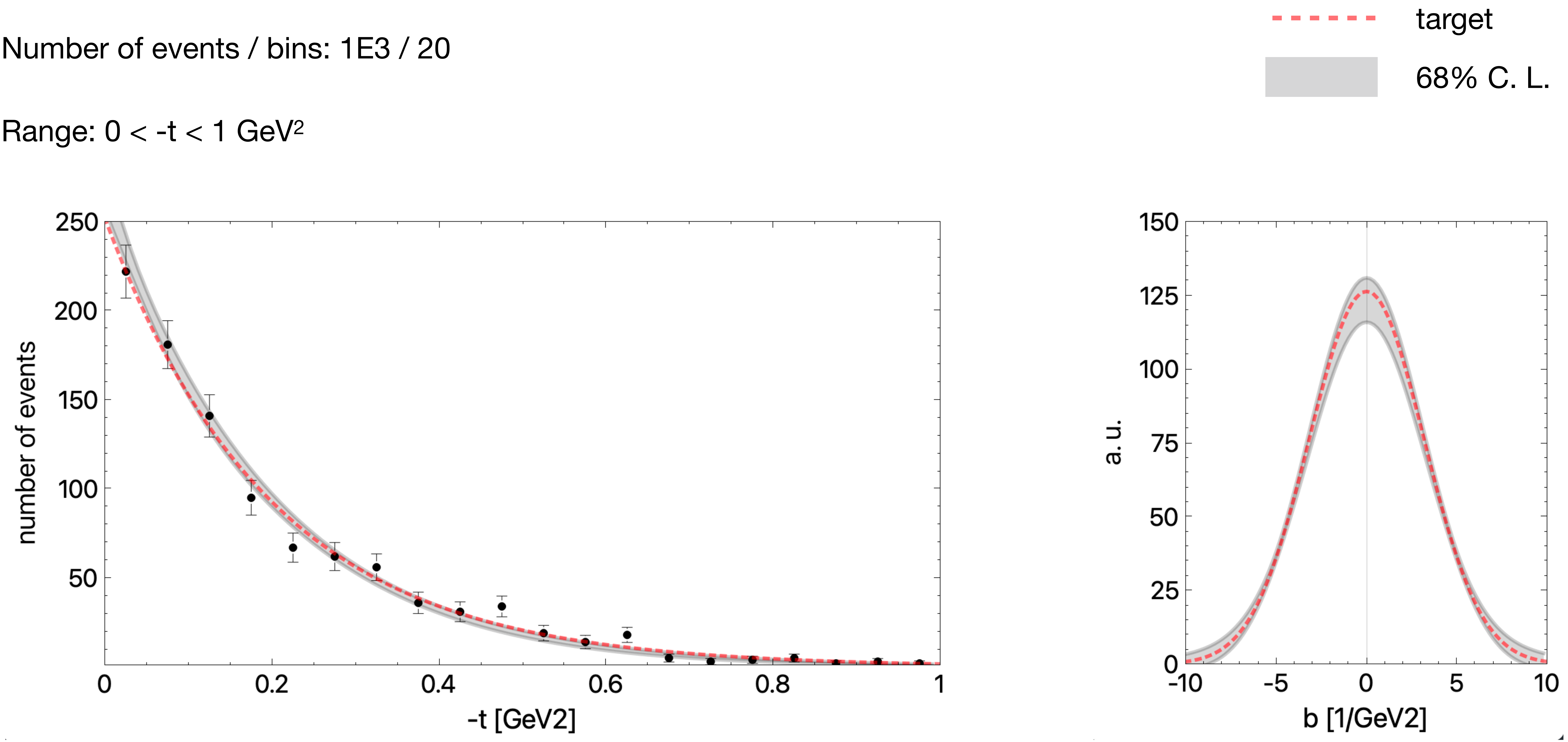
- Target function: $\exp(5t)$
- Number of events / bins: 1E3 / 20
- Range: $0 < -t < 1\ \mathrm{GeV}^2$





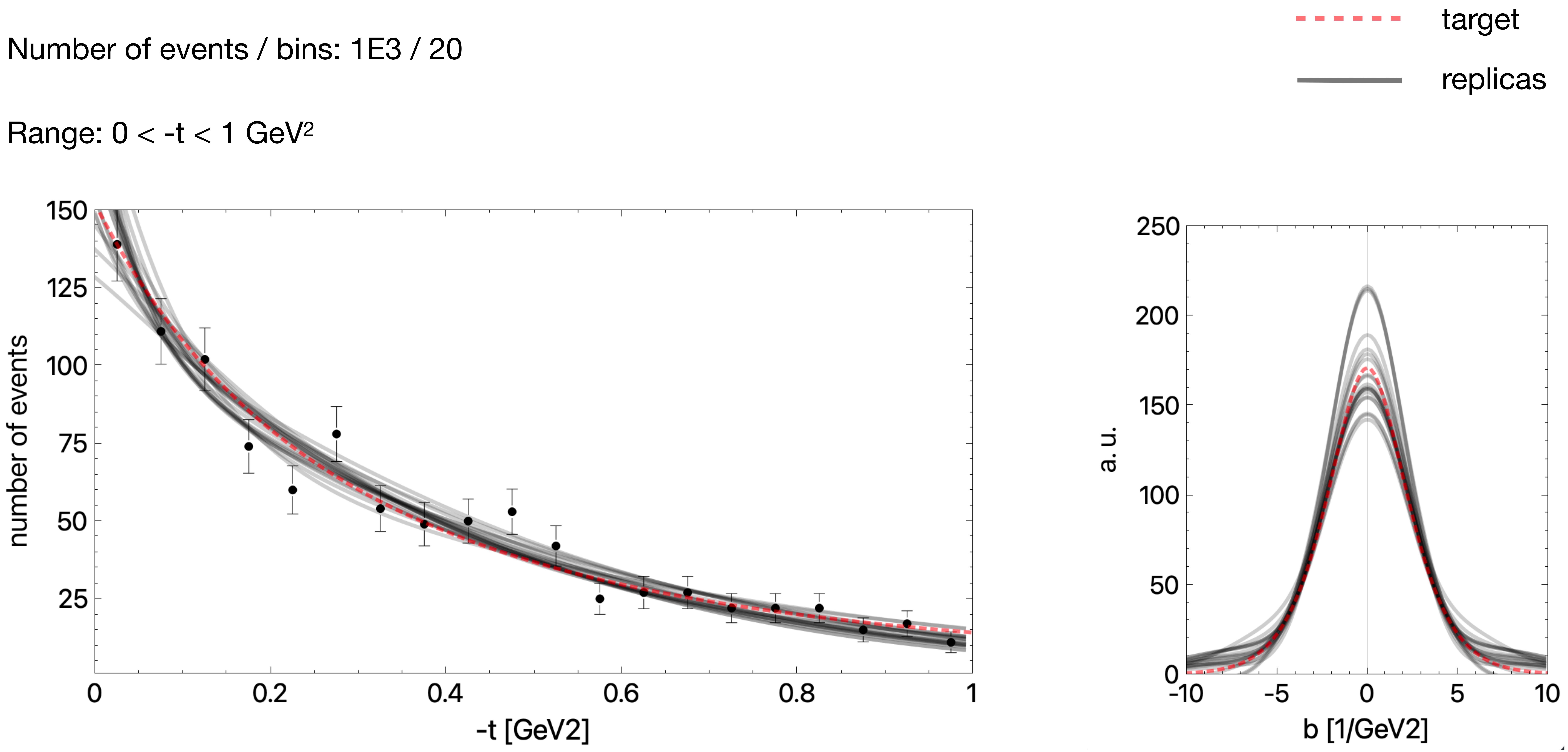
Demonstration

- Target function: $\exp(5t)$
- Number of events / bins: 1E3 / 20
- Range: 0 < -t < 1 GeV^2



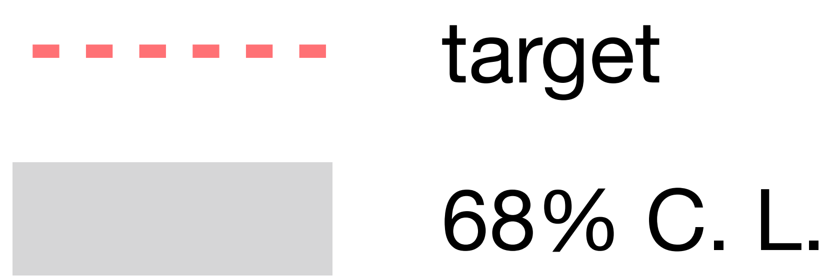
Demonstration

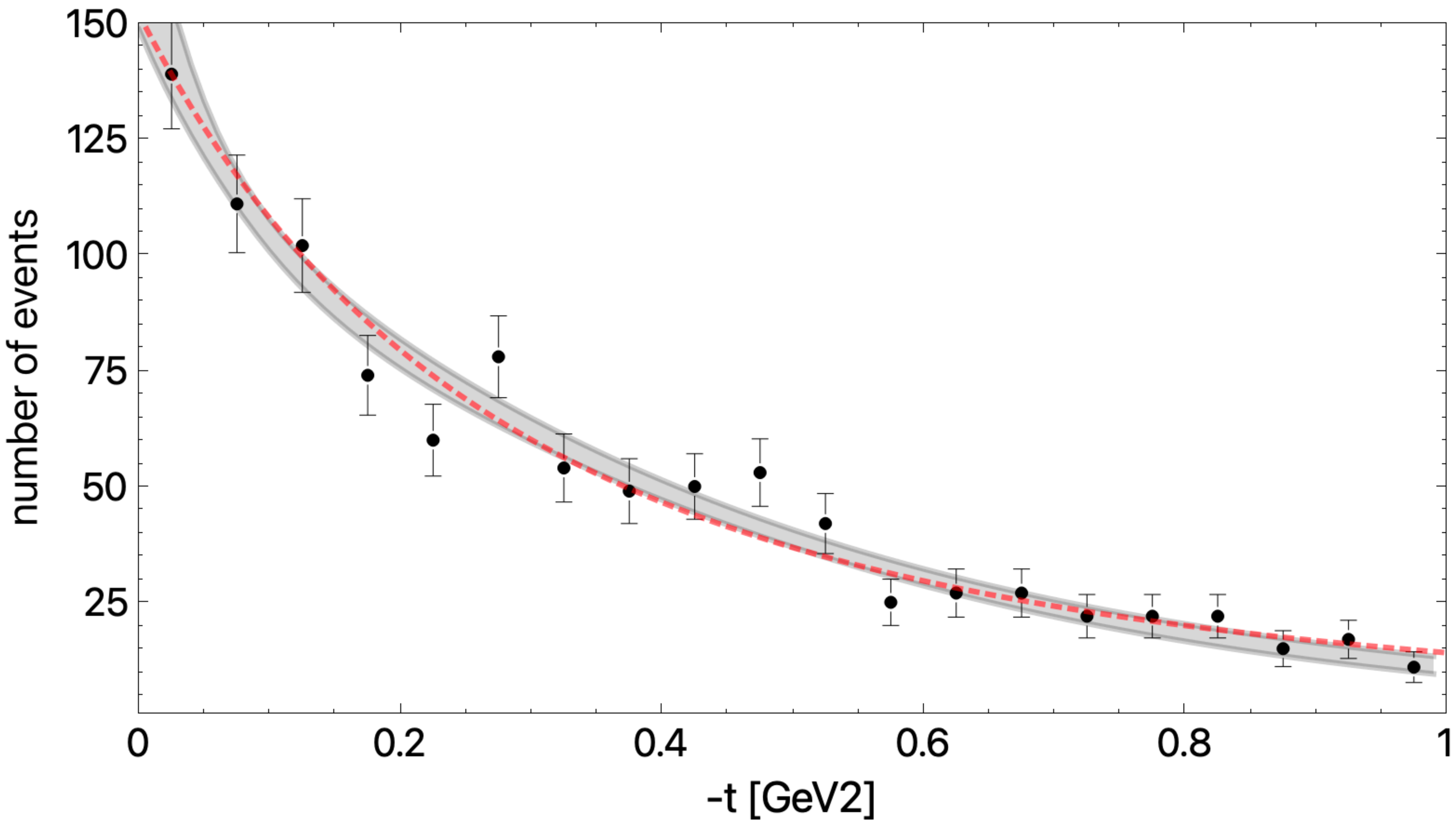
- Target function: $(1 - t/M_\mathrm{p}^2)^{-3}$
- Number of events / bins: 1E3 / 20
- Range: 0 < -t < 1 GeV^2

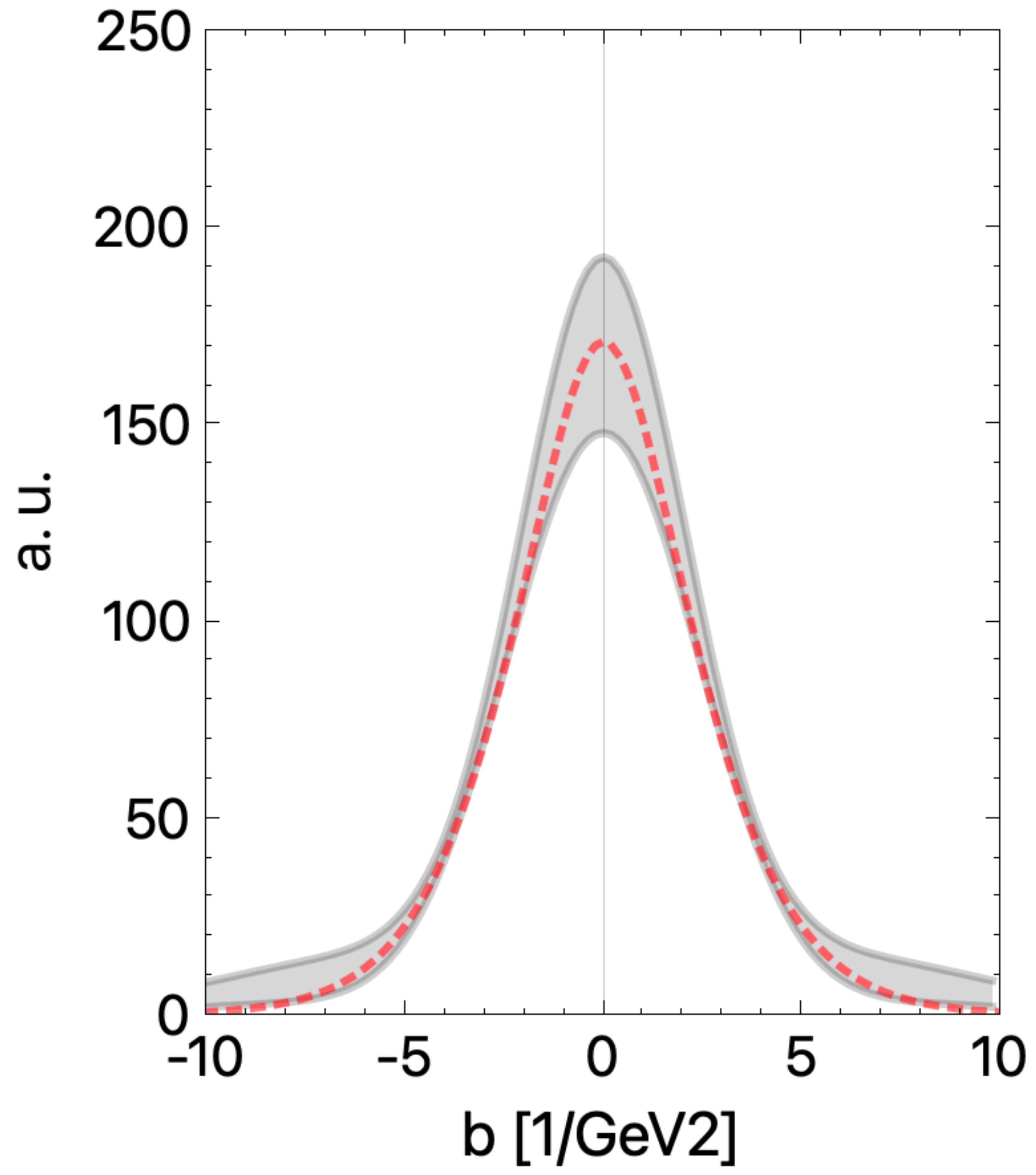


Demonstration

- Target function: $(1 - t/M_{\mathrm{p}}^2)^{-3}$
- Number of events / bins: 1E3 / 20
- Range: 0 < -t < 1 GeV^2

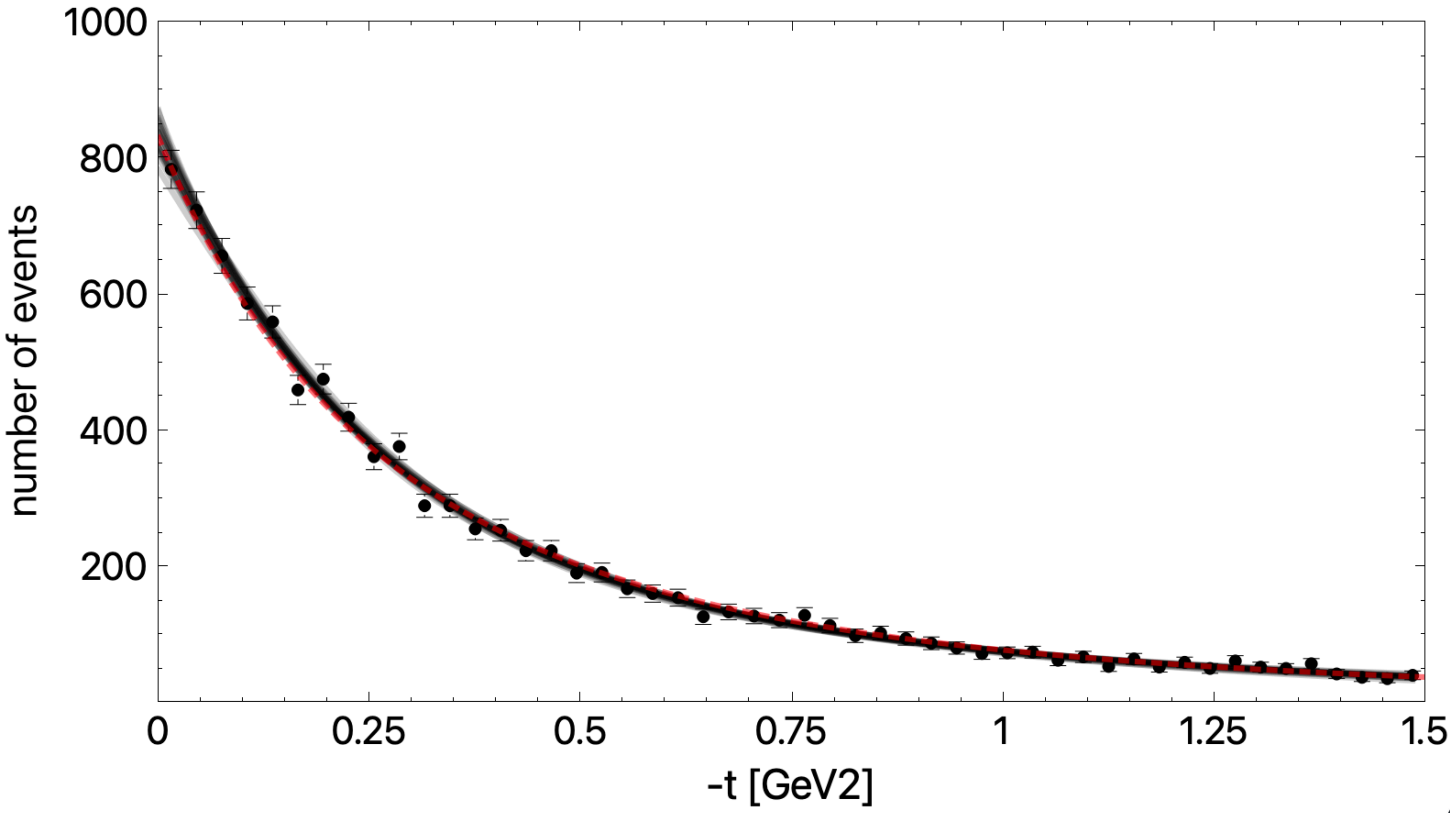


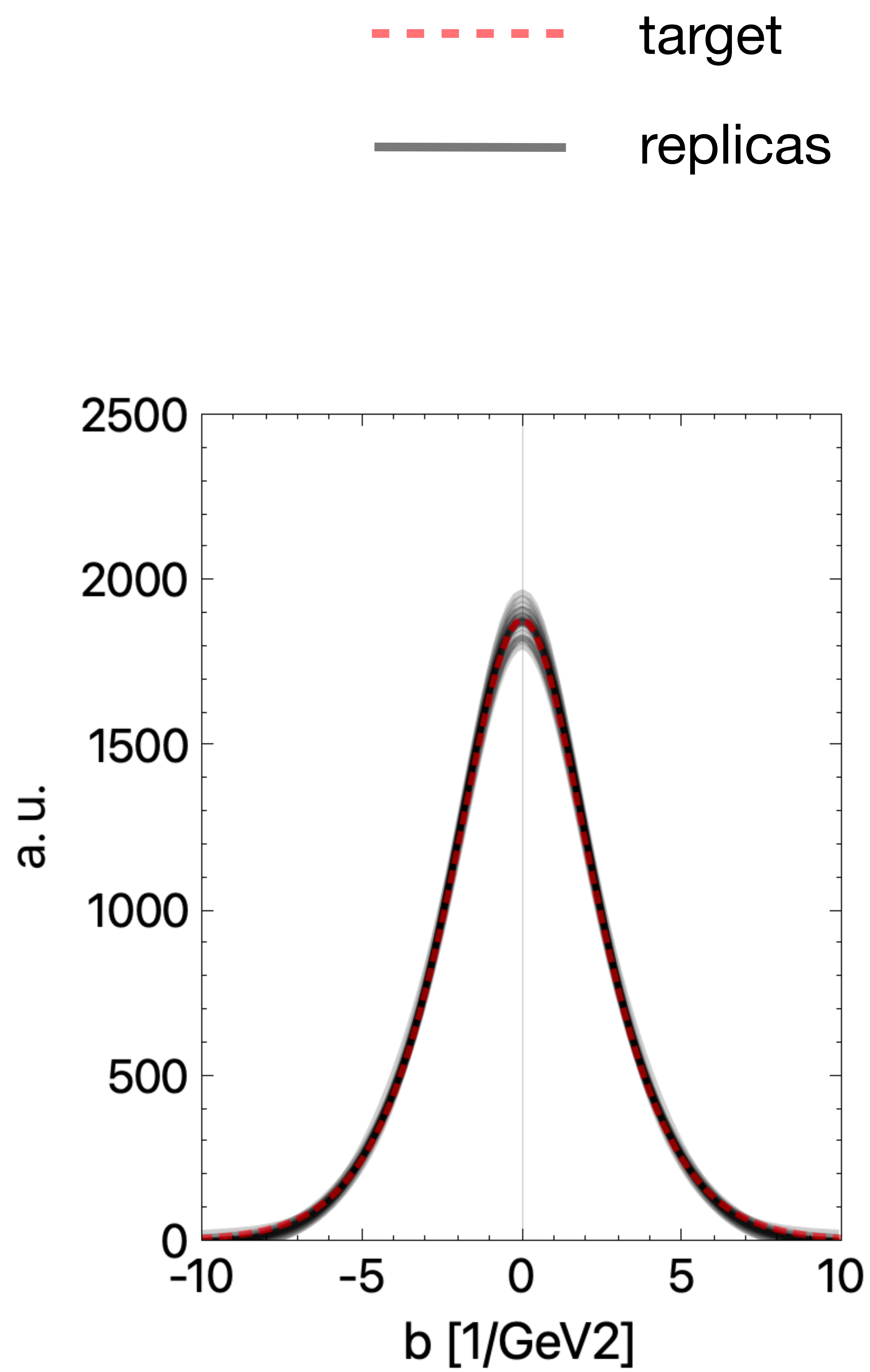




Demonstration

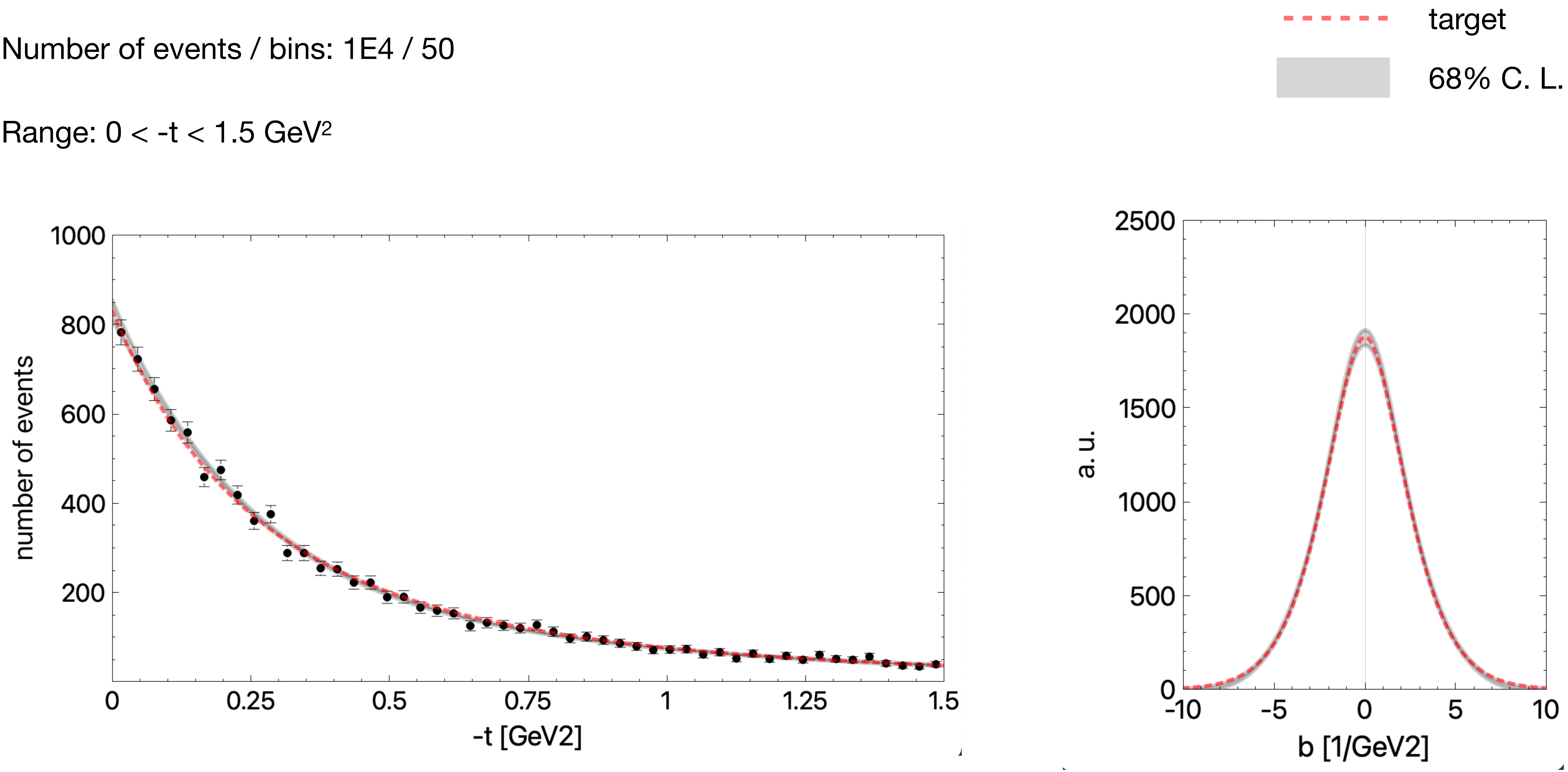
- Target function: $(1 - t/M_{\mathrm{p}}^2)^{-3}$
- Number of events / bins: 1E4 / 50
- Range: 0 < -t < 1.5 GeV[2]





Demonstration

- Target function: $(1 - t/M_{\mathrm{p}}^2)^{-3}$
- Number of events / bins: 1E4 / 50
- Range: $0 < -t < 1.5$ GeV^2



Unbinned likelihood method

- We have N events in the range (x_{min}, x_{max}) obtained from probability distribution $f(x, \mathbf{p})$, which is normalised by:

$$A(\mathbf{p}) = \int_{x_{\min}}^{x_{\max}} f(x, \mathbf{p}) \mathrm{d}x$$

here **p** represents vector of parameters

- Assuming Poisson variation for measuring N the likelihood of obtaining observed set of events is:

$$\mathscr{L}(\mathbf{p}) = \frac{A(\mathbf{p})^N \exp(-A(\mathbf{p}))}{N!} \prod_{i=1}^{N} \frac{f(x_i, \mathbf{p})}{A(\mathbf{p})}$$

- It is more convenient to minimise negative log of this likelihood

$$-\ln \mathscr{L}(\mathbf{p}) = -\sum_{i=1}^{N} \ln \left(\frac{f(x_i, \mathbf{p})}{A(\mathbf{p})} \right) - N \ln A(\mathbf{p}) + A(\mathbf{p}) + \ln N! \approx -\sum_{i=1}^{N} \ln f(x_i, \mathbf{p}) + A(\mathbf{p})$$

Conclusions

- Interesting but not trivial problem
- Work is ongoing
- Analysis should be useful