technische universität dortmund

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

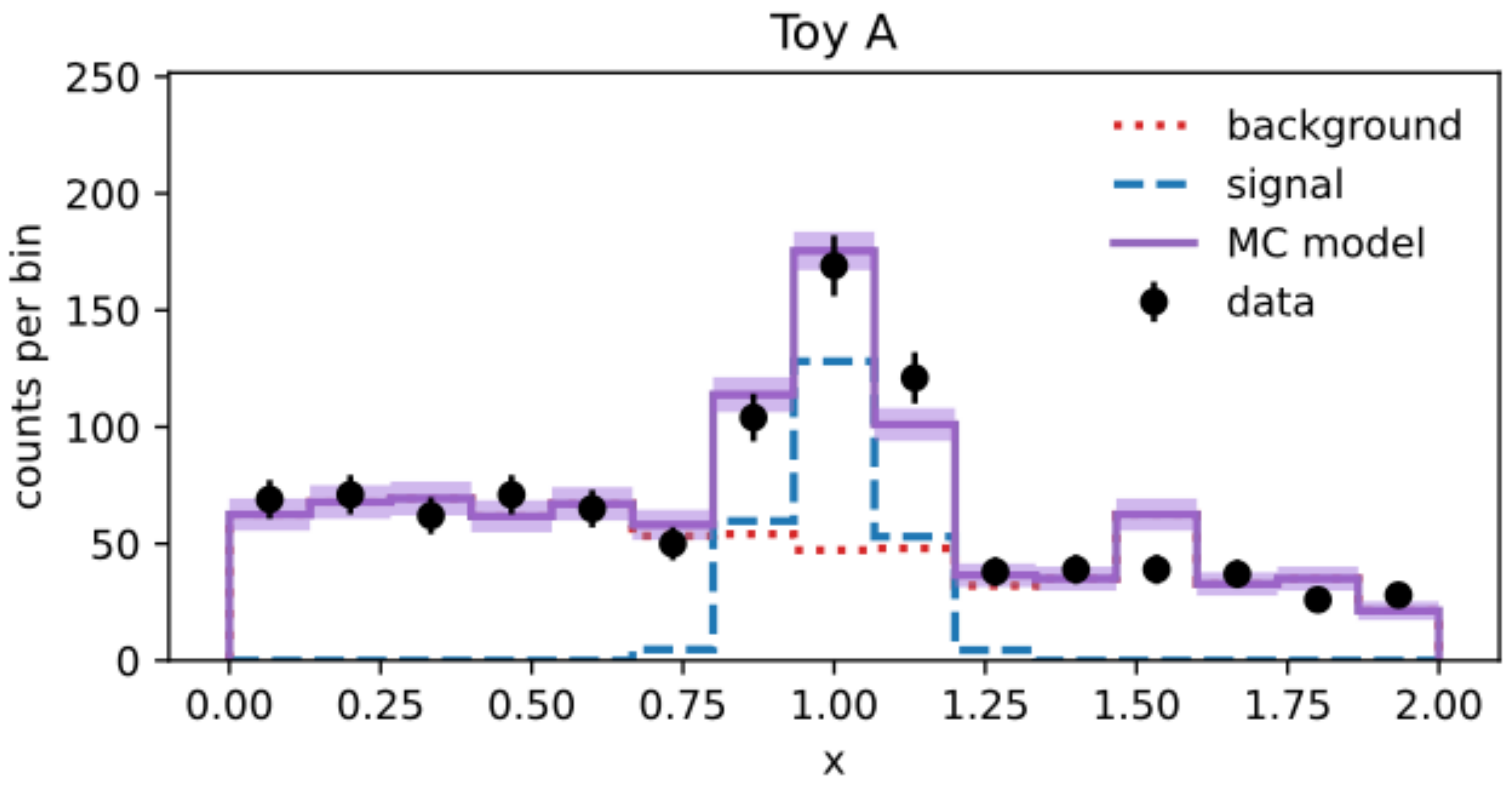
Template fits

Fitting non-parametric density models to data

Hans Dembinski, TU Dortmund
Ahmed Abdelmotteleb, University of Warwick

Overview

- Analysis problem
  - Given: sample **mixture of 2+ components**, e.g. signal and background
  - Variable ***x*** allows to discriminate components
    e.g. invariant mass of decay candidates
  - Want: **component yields**



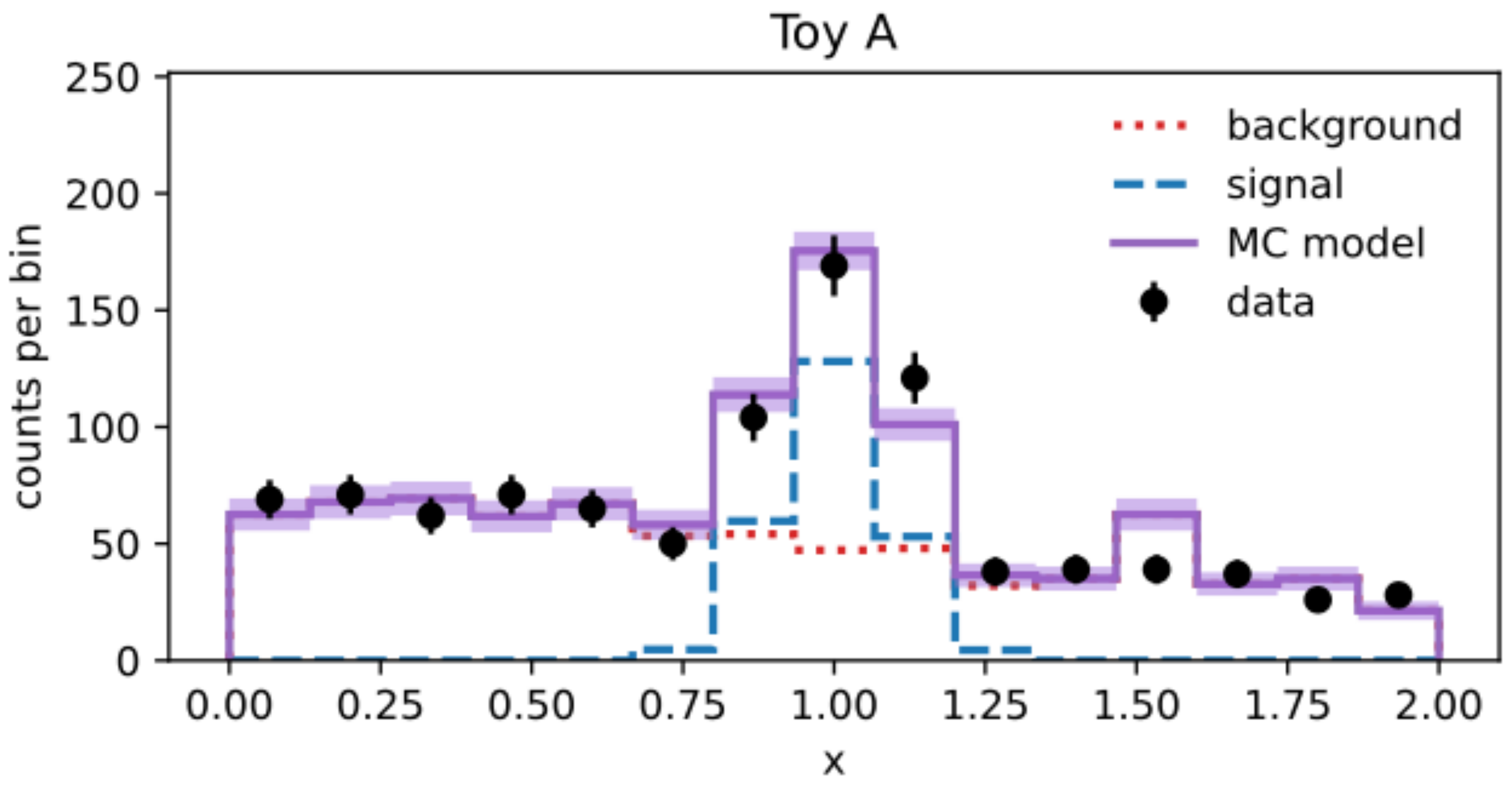
- Template fit
  - Template: component density estimated non-parametrically from independent samples
  - Need to propagate uncertainty of template
  - Elegant solution by **Barlow & Beeston**, 1993

- Templates from weighted samples
  - Barlow & Beeston solution not applicable
  - **Bayesian approach** by Argüelles, Schneider & Yuan, 2019
  - **ML approach** by HD, Abdelmotteleb, 2022

This talk: review and comparison of these methods

Analysis of sample mixture

- Binned maximum-likelihood approach
  - Bin sample over discriminating variable **x**
  - Assumption: Observed count in each bin is Poisson distributed



$$\ln \mathscr{L} = n \ln \mu - \mu - \ln n!$$

n ... observed count

$\mu = \mu_1 + \ldots + \mu_k$

$\mu_k = \dfrac{y_k \xi_k}{M_k}$ with yield y_k

$M_k =$ sum of ξ_k over all bins (normalisation factor)

Notation

- log-likelihood always calculated for single bin
- Total log-likelihood is sum of bin-wise log-likelihood

Templates

- Template: Collection of ξ_k from component *k*
- Parametric template
  - Shape computed from parametric model e.g. normal distribution for signal peak
    $$\xi_k = \int_{x_{\mathrm{low}}}^{x_{\mathrm{high}}} f(x; \vec{p}_k)\,\mathrm{d}x$$
  - Maximise log-likelihood to estimate yields $\hat{y}_k$ and nuisance parameters $\hat{\vec{p}}_k$
- Non-parametric template
  - ξ_k estimated from independent sample e.g. simulation or pure control sample
  - Key insight (B&B): true ξ_k unknown, but constrained by count a_k in independent sample
  - Maximise log-likelihood to estimate yields y_k and nuisance parameters ξ_k

Likelihood for non-parametric template

- μ constrained by n via log-likelihood $\ln \mathscr{L} = n \ln \mu - \mu - \ln n!$
- ξ_k constrained by a_k via log-likelihood $\ln \mathscr{L}_k = a_k \ln \xi_k - \xi_k - \ln a_k!$
- Total log-likelihood $\underbrace{\ln \mathscr{L}}_{\text{data}} + \underbrace{\ln \mathscr{L}_1}_{\text{template 1}} + \ldots + \underbrace{\ln \mathscr{L}_k}_{\text{template k}}$

Interlude: Baker & Cousins transform

- Baker and Cousins, 1984
  - Binned likelihood can be transformed so that minimum is asymptotically chi-square distributed

$$Q(\vec{p}) = -2\ln\left[\frac{\mathscr{L}(n;\mu(\vec{p}))}{\mathscr{L}(n;n)}\right]$$

  - Minimum value $Q_{\min}$ doubles as **goodness-of-fit test statistic**
  - For Poisson-distributed data identical to **Cash statistic** C (Cash, 1979)

$$C(n;\mu) \equiv Q(n;\mu) = 2\big(\mu - n - n(\ln\mu - \ln n)\big)$$

- Further beneficial effects
  - Calculation more numerically stable
  - Avoids expensive calculation of factorials in Poisson likelihood

Apply Baker & Cousins transform

- Before: Maximise

$$\ln \mathscr{L} \;+\; \ln \mathscr{L}_1 \;+\; \dots \;+\; \ln \mathscr{L}_k$$

data template 1 ... template k

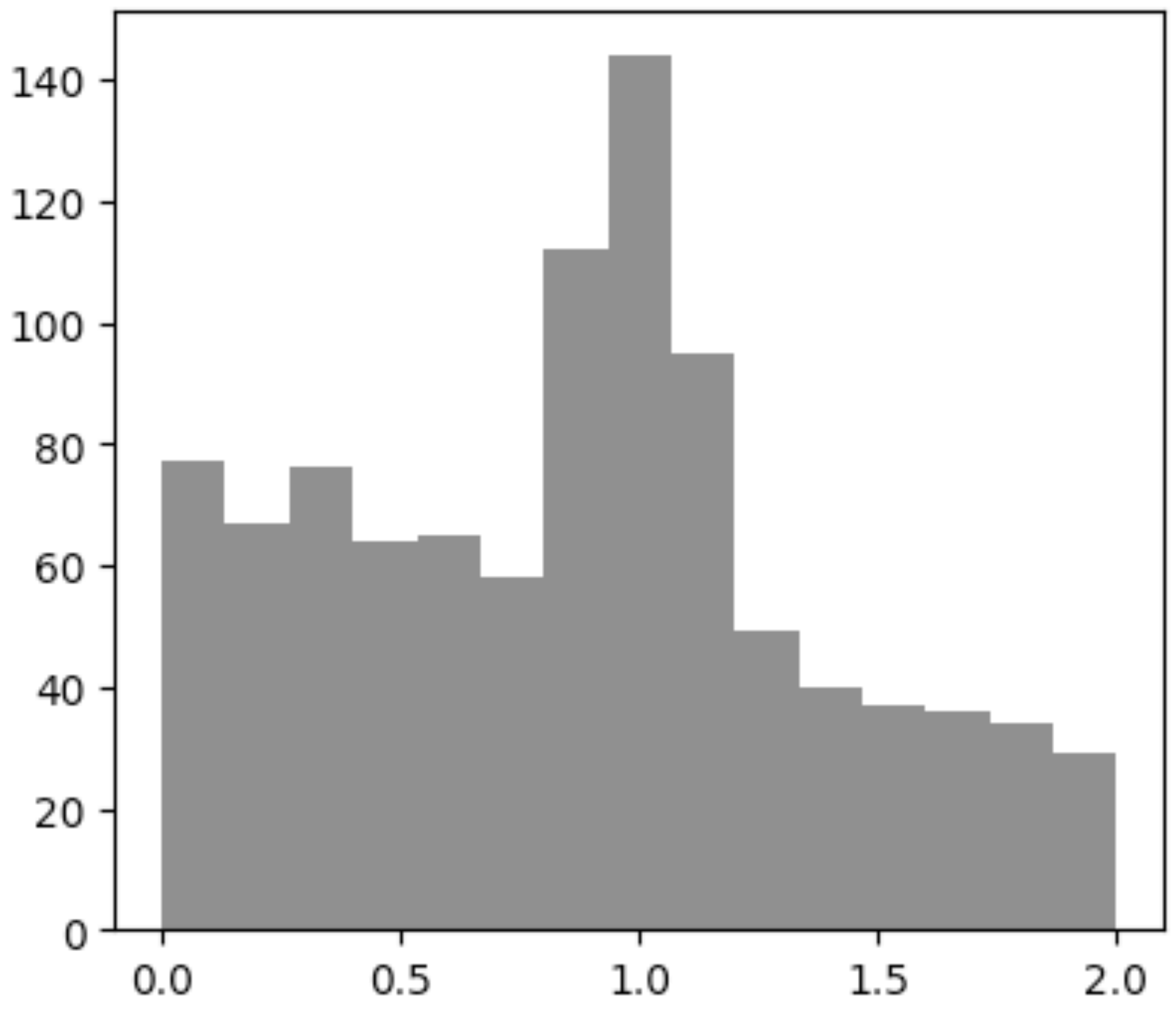
- After: Minimise

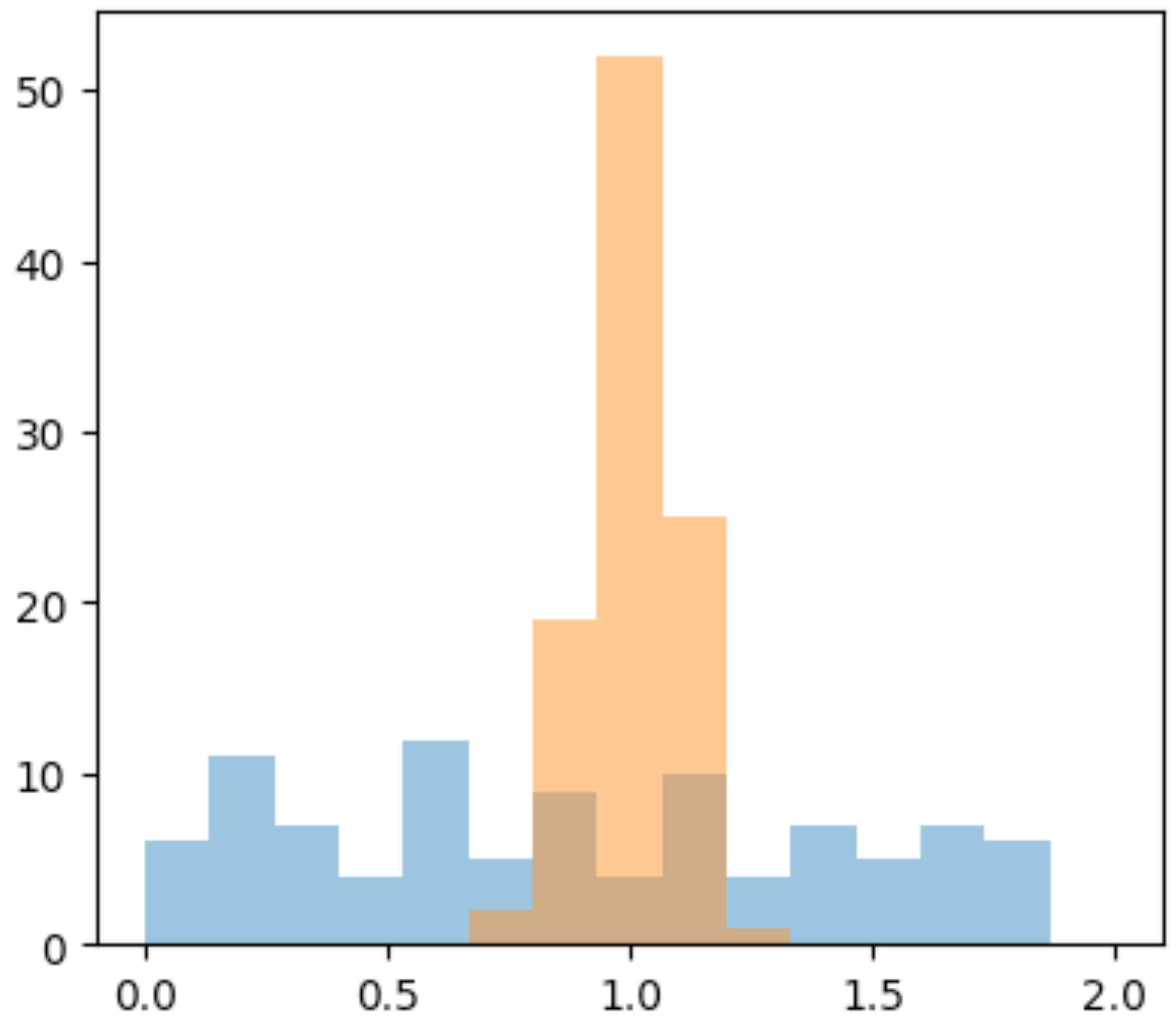
$$Q = C\left(n; \frac{y_1 \xi_1}{M_1} + \dots + \frac{y_k \xi_k}{M_k}\right) \;+\; C(a_1; \xi_1) \;+\; \dots \;+\; C(a_k; \xi_k)$$

- Q can be minimised with standard software, e.g. MIGRAD from MINUIT
- However: $K \times N$ nuisance parameters with *K* number of components, *N* bins

Numerical example

https://scikit-hep.org/iminuit/notebooks/template_fits.html





Naive fit with $\xi_k = a_k$

$$y_{\mathrm{bkg}} = 761 \pm 30$$
$$y_{\mathrm{peak}} = 193 \pm 19$$

Correct fit

$$y_{\mathrm{bkg}} = 800 \pm 50$$
$$y_{\mathrm{peak}} = 190 \pm 40$$

Solution by Barlow & Beeston

- Number of bins N can be very large, if discriminant variable $\boldsymbol{x}$ is multi-dimensional

  Example: 4 dimensions, 10 bins per dimension = 10 000 bins
  $\rightarrow$ **20 000 nuisance parameters**

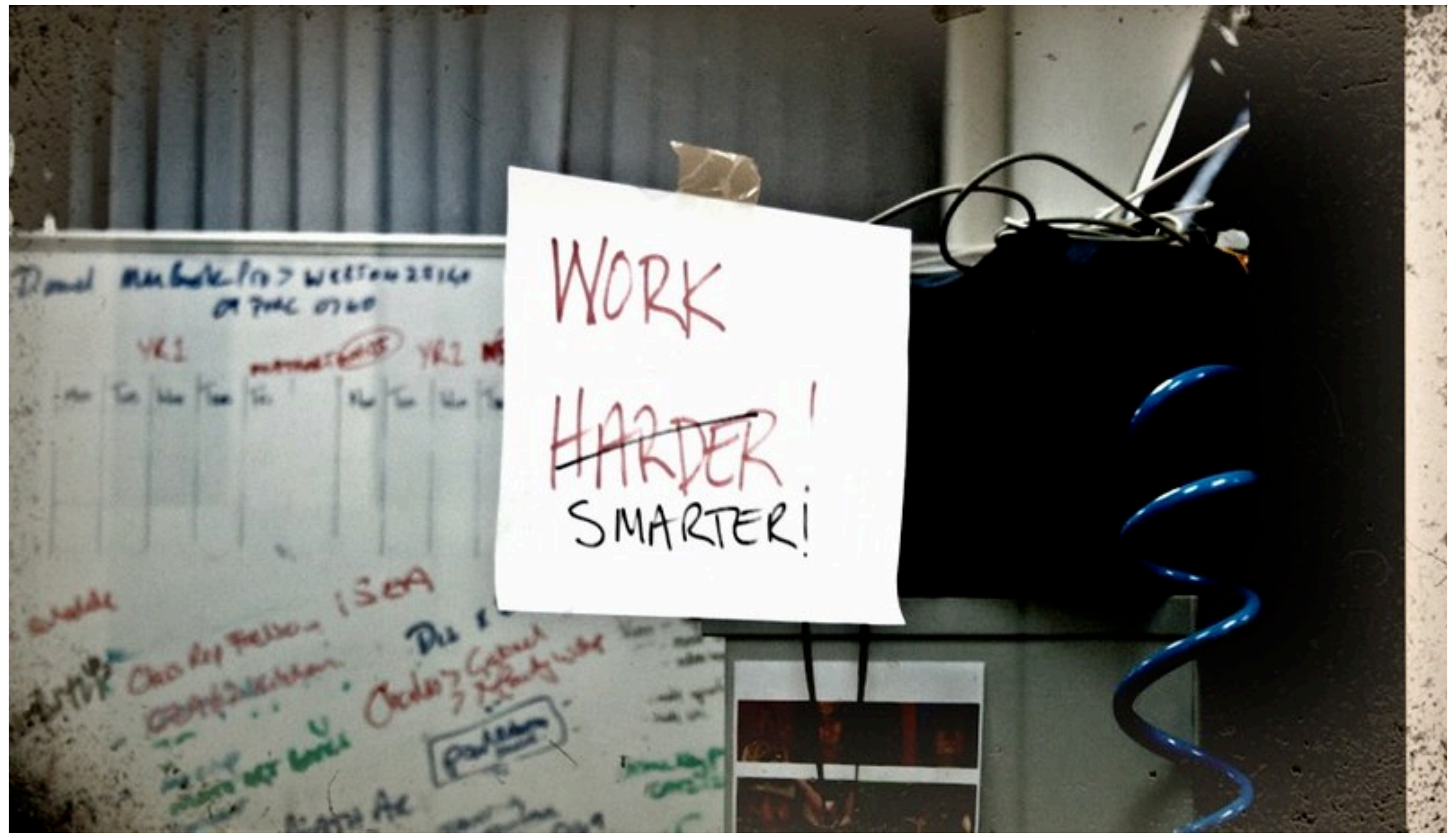
- Problems with large number of parameters solvable with modern methods
  - L-BFGS: quasi-newton method for large problems
  - Stochastic gradient descent methods (e.g. Adam)

- Both not available in 1993, brute-force still expensive today

- Barlow & Beeston ansatz
  - Split problem into nested two-step minimisation

$$Q = C\left(n; \frac{y_1 \xi_1}{M_1} + \ldots + \frac{y_k \xi_k}{M_k}\right) + C(a_1; \xi_1) + \ldots + C(a_k; \xi_k)$$

- Outer step
  - Minimise $C(n; y_1, \ldots, y_k, \hat{\xi}_1(\vec{y}), \ldots, \hat{\xi}_k(\vec{y}))$ using MIGRAD
  - Outer step only sees $\vec{y}$ as floating variables



https://www.flickr.com/photos/37230837@N04/5146762770

- Inner step
  - Compute solution to $\partial_{\xi_k} Q = 0$ analytically for each proposal $\vec{y}$
  - Solve score equations with numerical root-finder to find estimates $\hat{\xi}_k(\vec{y})$
  - Problem can be reduced to **one call to root-finder per bin**

Conway's approximation

- Barlow & Beeston approach **exactly** solves maximum-likelihood problem, but computation still **relatively expensive**

- Conway, 2011, proposed alternative inexpensive approach
  - Also two-step approach, minimise

$$Q_{\mathrm{C}} = C(n; \beta\, \mu_0(\vec{y})) + \frac{(\beta - 1)^2}{V_\beta} \quad \text{with} \quad \mu_0(\vec{y}) = \sum_k \frac{y_k a_k}{M_k}$$

  - For fixed $\vec{y}$, get estimate $\hat{\beta}(\vec{y})$ for each bin by solving quadratic equation
  - No derivation given in original publication
  - **Our derivation** revealed two approximations to get Q_{C} and better alternative

- Approximation 1: setting $\beta \approx \beta_k$

$$\mu = \sum_k \frac{y_k \xi_k}{M_k} = \sum_k \frac{y_k \beta_k a_k}{M_k} \approx \beta \underbrace{\sum_k \frac{y_k a_k}{M_k}}_{\mu_0}$$ yields cost function $Q \approx C(n; \beta\mu_0) + \sum_k C(a_k; \beta a_k)$

- Approximation 2: Taylor expansion around $\beta = 1$

$Q \approx C(n; \beta\mu_0) + a(\beta - 1)^2$ with $a = \sum_k a_k$

  - Second term $a(\beta - 1)^2$ resembles Gaussian penalty term in Q_C
  - Indeed, $V_\beta \to 1/a$ if one component is dominant, but Q_C performs better generally

- Approximations valid if...
  - Templates are constructed from large samples
  - One component dominates in each bin

Our insight

- Approximation 2 not necessary

  Starting from $Q \approx C(n; \beta\mu_0) + \sum_k C(a_k; \beta a_k)$ we compute $\partial_\beta Q = 0$ and get:

  $$\hat{\beta} = \frac{n + a}{\mu_0 + a} \text{ with } a = \sum_k a_k$$

  - Limits $n \to \infty$ and $a \to \infty$ easy to interpret

- Remaining caveat: Assumption 1 that one component is dominant
  - Will partially repair this later

Templates from weighted samples

- In current analyses, templates often build from weighted samples
  - Weights from NLO Monte-Carlo generators
  - Frequency weights applied to simulation to better match observed distributions
  - sWeighted control samples

- Barlow & Beeston solution not applicable to these cases
- Exact likelihood for weighted samples intractable → **approximations mandatory**

Interlude: SPD approximation

- In weighted samples, count in a bin replaced by sum of weights

$$n = \sum_i w_i$$

- Bohm & Zech, 2014
  - Assumption: w_i drawn independently and identically (iid) from discrete distribution
    - **Discreteness** assumed **without loss of generality**
    - **iid** assumption **often slightly violated** in practice
  - Then: n is effectively drawn from **compound Poisson distribution (CPD)**

$$n = n_1 w_1 + \ldots + n_k w_k$$

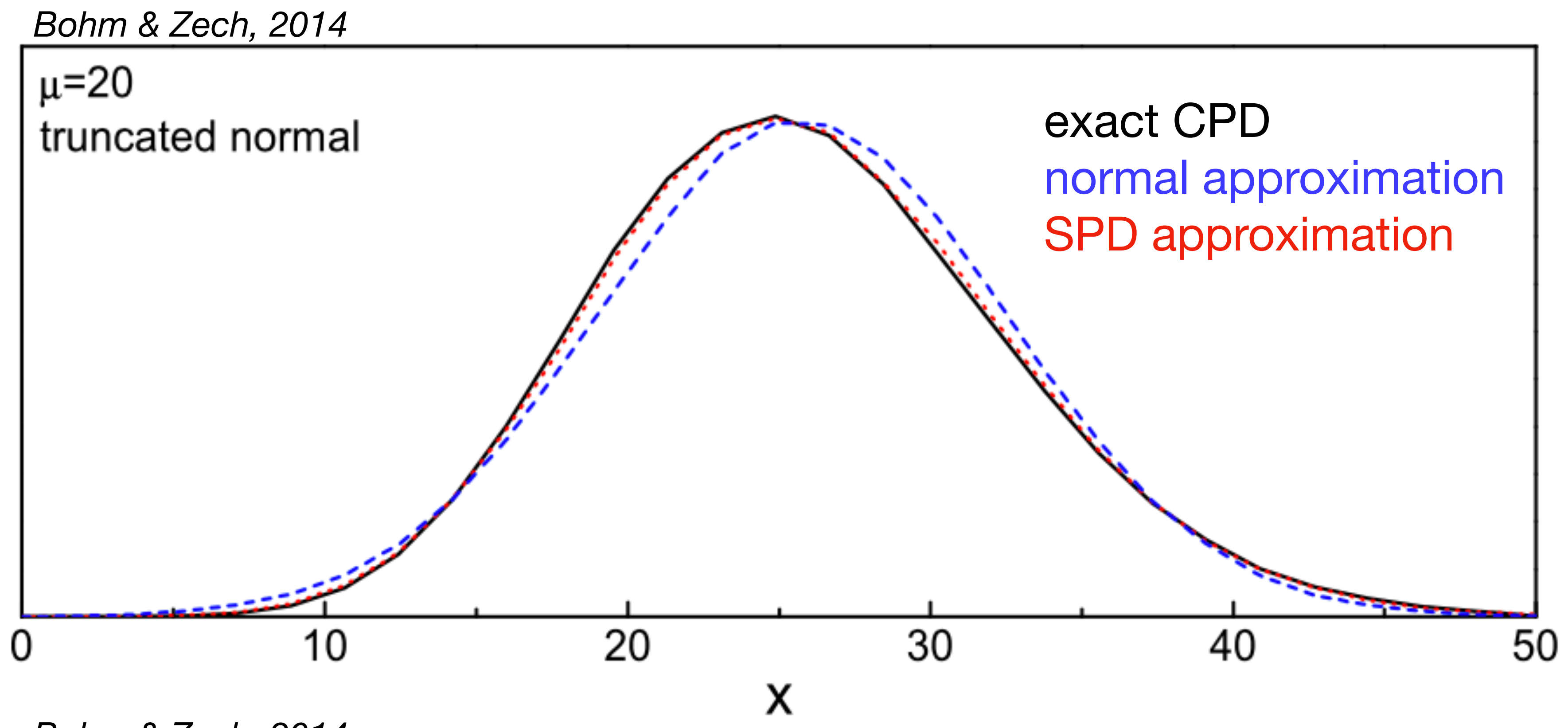
where n_k are Poisson-distributed with unknown expectations λ_k

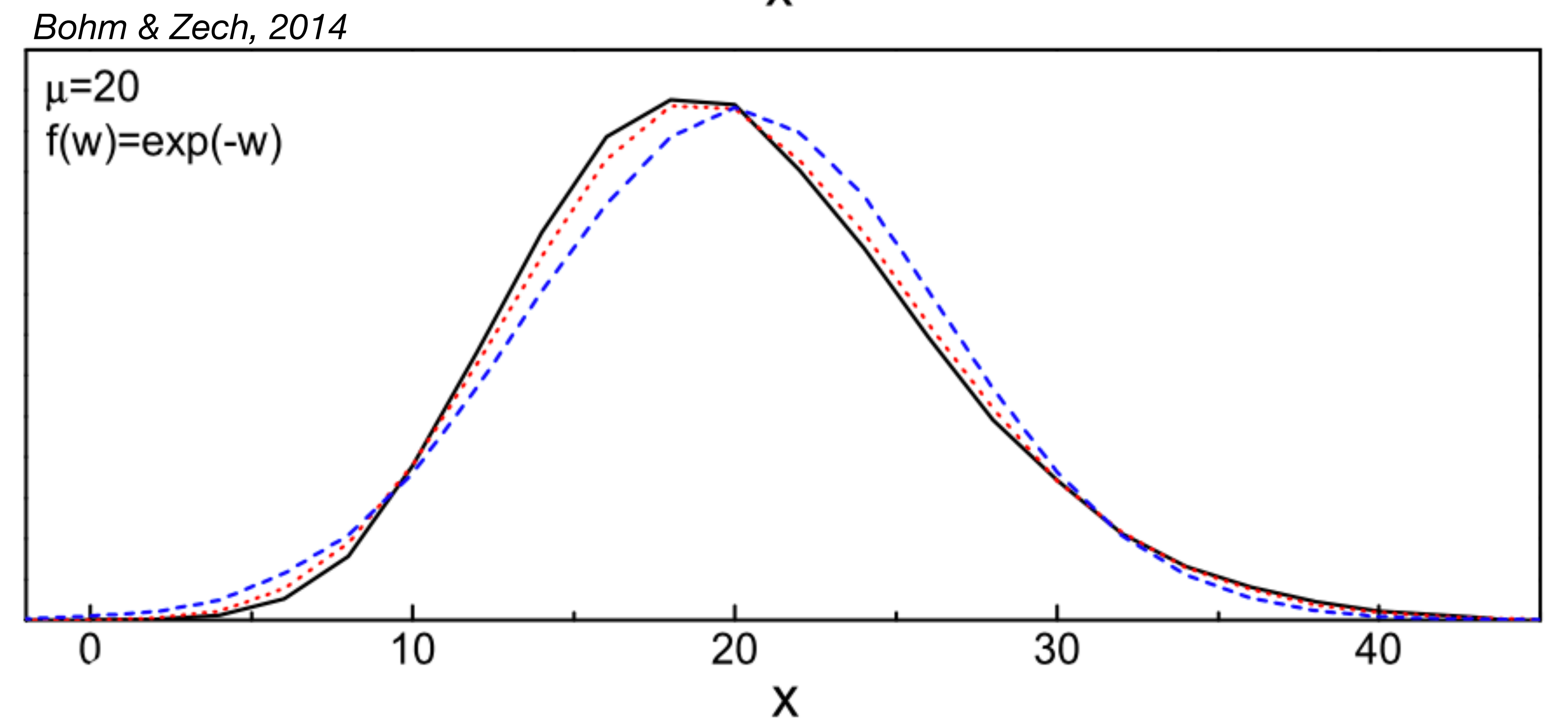
- CPD analytically intractable, approximated by appropriately **scaled Poisson distribution (SPD)**

$$n = \sum_i w_i \qquad V_n = \sum_i w_i^2 \qquad t = \frac{n}{V_n}$$

$n = kt$ with $k \sim$ Poisson with $\lambda = nt$

  - k also known as *effective count*
  - SPD has **same first and second moments** as CPD
  - SPD has similar third and forth moments as CPD
  - SPD has correct limit for $w_i = w$





- SPD is good approximation unless weight distribution has extreme tails
- SPD can be constructed **for any variable** x using $t_x = E[x]/V_x$
- Practical challenge: accurately computing n and V_n requires **sufficiently populated bins**

Bayesian approach

- Argüelles, Schneider & Yuan, 2019, first used the SPD in context of template fitting
- Marginal likelihood $\mathscr{L}_{ASY}$ for n obtained by integrating over probability density $p(\mu)$

$$\mathscr{L}_{ASY} = \int_0^\infty \frac{\mu^n e^{-\mu}}{n!} p(\mu)\, \mathrm{d}\mu$$

- $p(\mu)$ obtained by applying Bayes' theorem

likelihood for observing μ_0 (SPD)

$$p(\mu; \mu_0, V_\mu) \propto \mathscr{L}(\mu_0; \mu, V_\mu)\, q(\mu)$$

prior for μ

$$\mu_0 = \sum_k \frac{y_k \sum_i w_{k,i}}{M_k}$$

$$V_\mu = \sum_k \frac{y_k^2 \sum_i w_i^2}{M_k^2}$$

Flat prior $q(\mu)$ used in main result

- Integral can be solved analytically, one gets

$$\mathscr{L}_{\mathrm{ASY}} = \frac{s^{s\mu_0+1}\,\Gamma(n + s\mu_0 + 1)}{n!\,(s+1)^{n+s\mu_0+1}\,\Gamma(s\mu_0 + 1)} \quad \text{with} \quad s = \frac{\mu_0}{V_\mu}$$

- Authors propose to use $\mathscr{L}_{\mathrm{ASY}}$ in frequentist-style fit
  - Estimate $\vec{y}$ by minimising $-\ln \mathscr{L}_{\mathrm{ASY}}$
  - Compute uncertainties of $\vec{y}$ with standard MINUIT algorithms
  - Point estimates and uncertainties have good frequentist properties

- Minor caveat: $\mathscr{L}_{\mathrm{ASY}}$ does not provide chi-square-distributed test statistic

Our approach

- Use SPD to generalise our previous result

$$Q = C(n; \beta\mu_0) + \sum_k C(a_k; \beta a_k) \rightarrow Q_{\mathrm{DA}} = C(tn; \beta t\mu_0) + C(s\mu_0; \beta s\mu_0)$$

$$n = \sum_i w_i' \qquad V_n = \sum_i w_i'^2 \qquad t = \frac{n}{V_n} \qquad \boxed{\hat{\beta} = \frac{tn + s\mu_0}{t\mu_0 + s\mu_0}}$$

$$\mu_0 = \sum_k \frac{y_k \sum_i w_{k,i}}{M_k} \qquad V_\mu = \sum_k \frac{y_k^2 \sum_i w_{k,i}^2}{M_k^2} \qquad s = \frac{\mu_0}{V_\mu}$$

with data weights w_i' and template weights w_i

Our approach

- Minimise

$$Q_{\mathrm{DA}} = C(tn; \hat{\beta} t\mu_0) + C(s\mu_0; \hat{\beta} s\mu_0) \text{ with } \hat{\beta} = \frac{tn + s\mu_0}{t\mu_0 + s\mu_0}$$

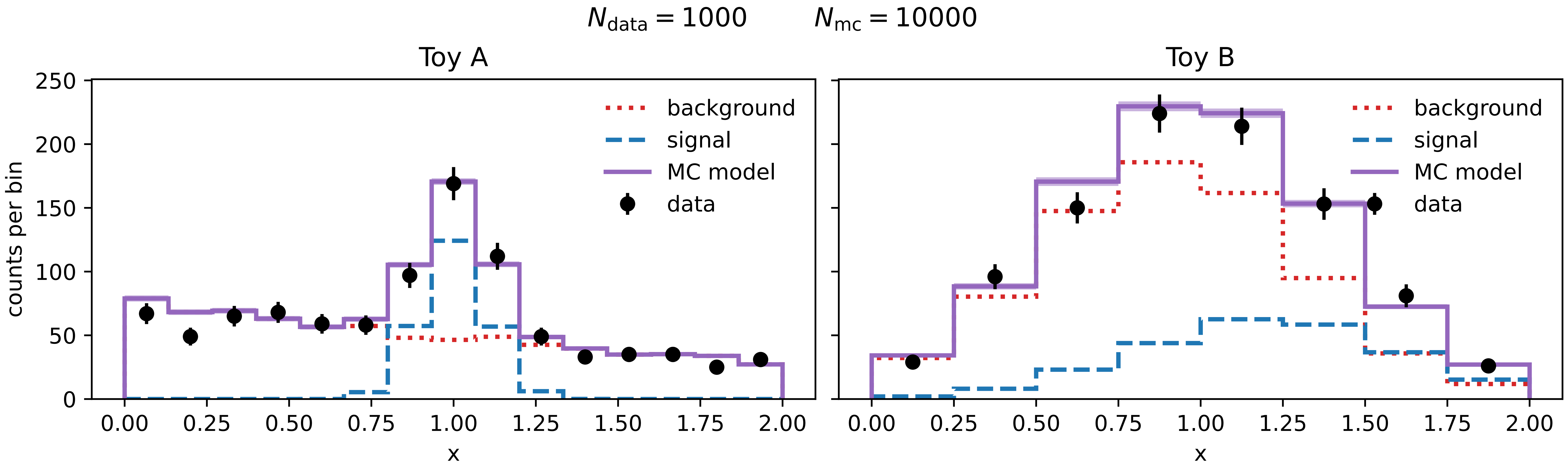
with respect to $\vec{y}$; implicit in $\mu_0 = \mu_0(\vec{y})$ and $s = s(\vec{y})$

- Integration of SPD provides two benefits
  - Approach supports both weighted data and weighted templates
  - Variance of μ_0 correct if more than one component is dominant (analog to Conway)

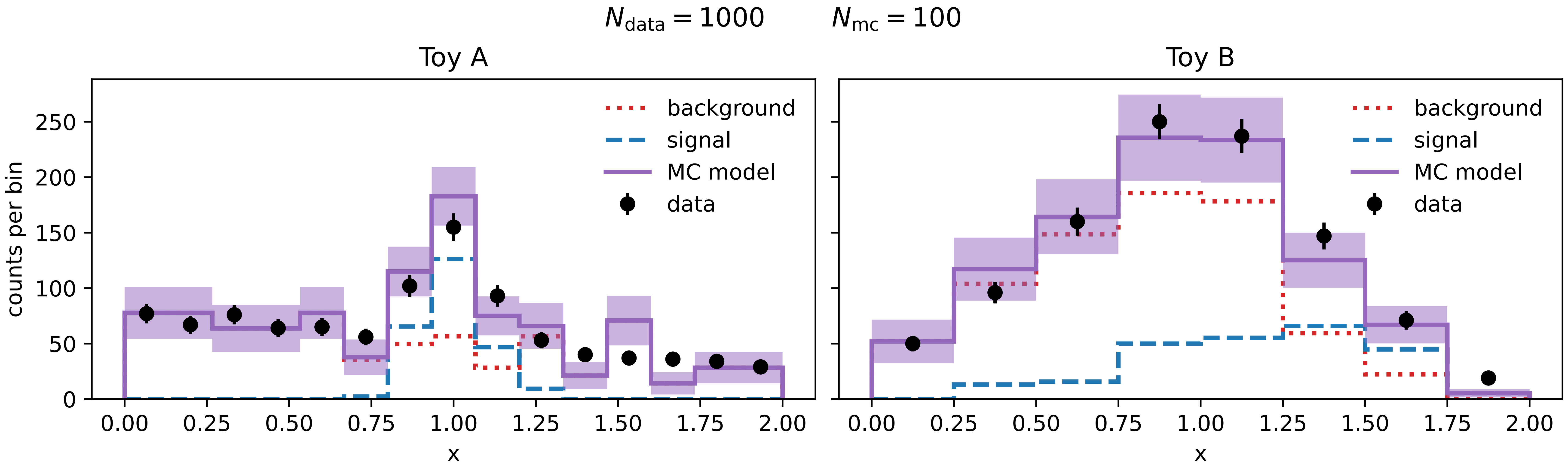
- Approximation 1 analog to SPD approximation
  - Compound Poisson distribution replaced by appropriately scaled Poisson distribution

	Barlow & Beeston	Conway (original)	Argüelles, Schneider & Yuan	Conway (our variant)	Our approach
Theoretical foundation	Frequentist	Frequentist	Bayesian & Frequentist	Frequentist	Frequentist
Approximations / Limitations		A1, A2	SPD, flat prior	SPD, A1, A2	SPD, A1
Supports weighted templates			✓	✓	✓
Supports weighted data				✓	✓
gof test statistic				✓	✓

Toy study



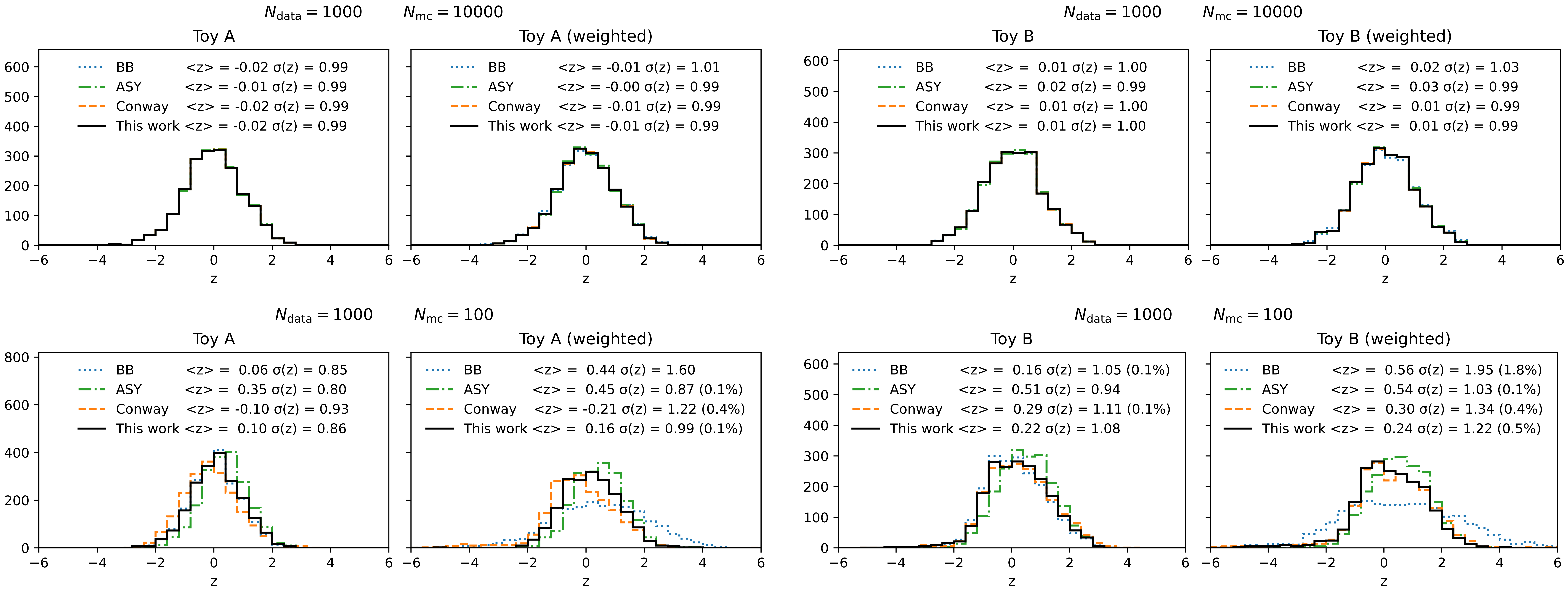
N_data = 1000
N_mc = 10000
Toy A
Toy B
background
signal
MC model
data
counts per bin
x



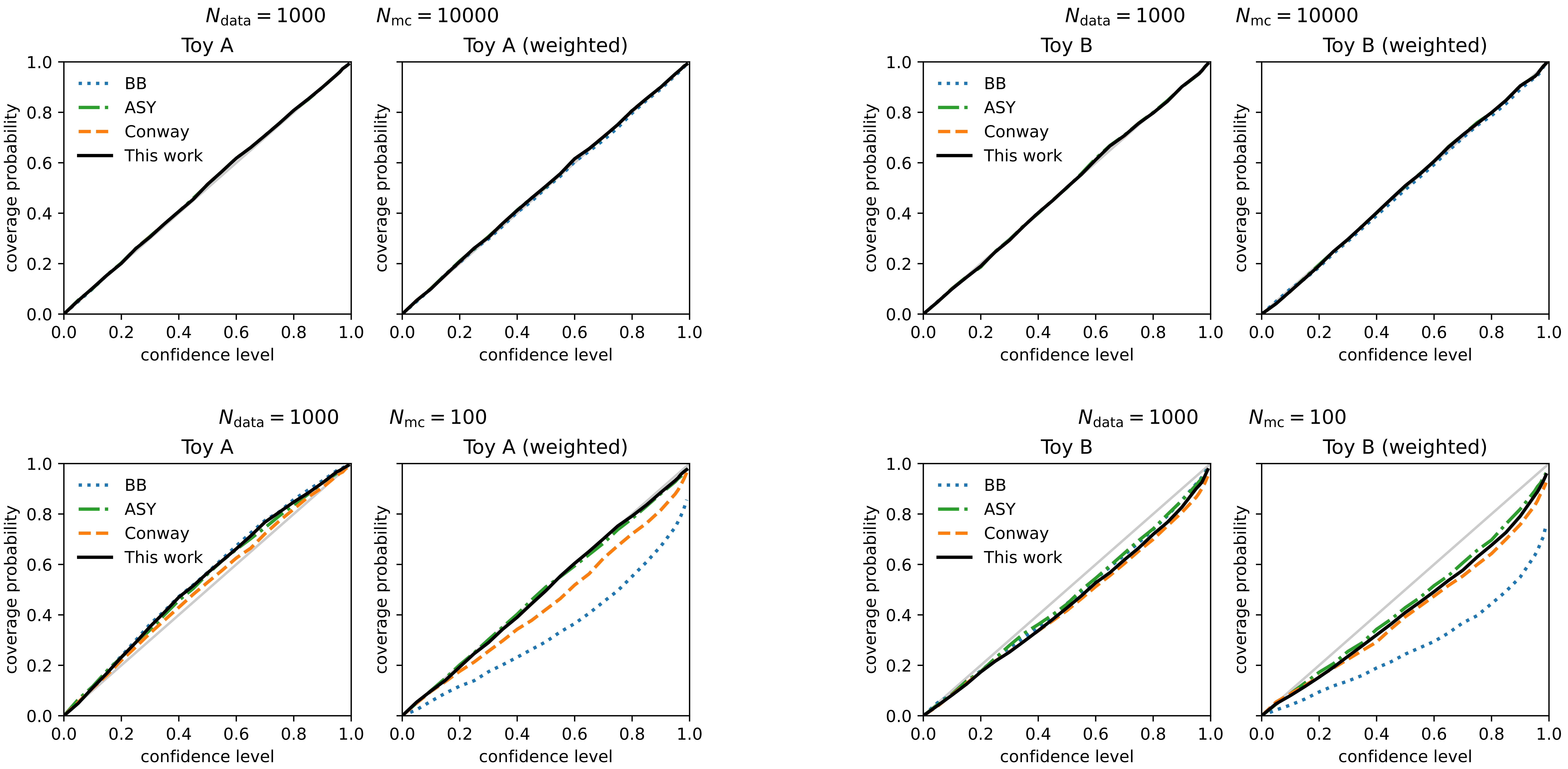
N_data = 1000
N_mc = 100
Toy A
Toy B
background
signal
MC model
data
counts per bin
x

Bias and variance

$$z = \frac{\hat{y}_{\text{signal}} - y_{\text{signal}}}{\sqrt{\hat{V}_{y_{\text{signal}}}}}$$

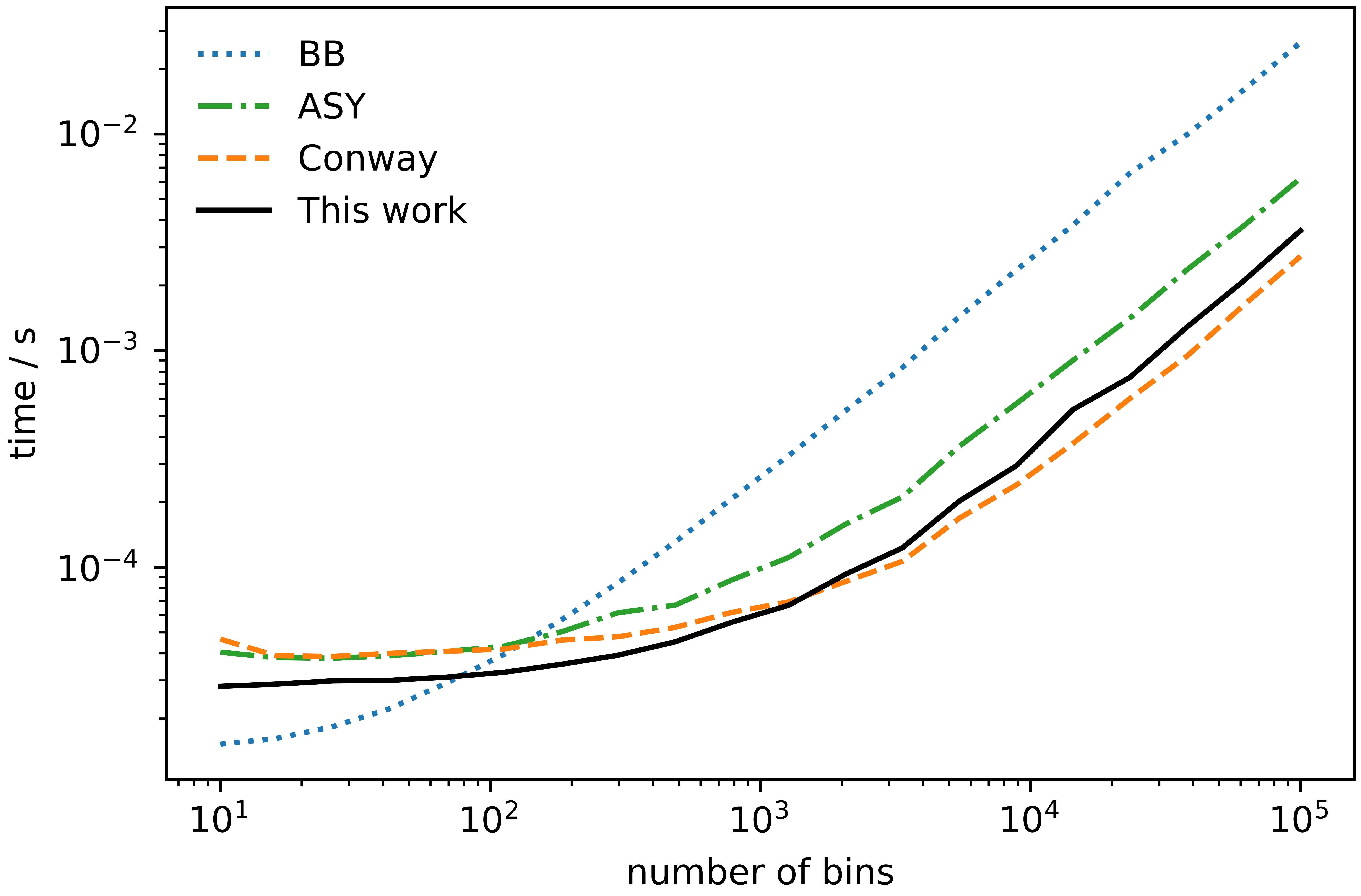


Coverage



$N_{\mathrm{data}} = 1000$
$N_{\mathrm{mc}} = 10000$
Toy A
Toy A (weighted)
Toy B
Toy B (weighted)
$N_{\mathrm{data}} = 1000$
$N_{\mathrm{mc}} = 100$
BB
ASY
Conway
This work
coverage probability
confidence level
0.0
0.2
0.4
0.6
0.8
1.0

Performance



BB: TFractionFitter, **C++**

Others: our implementation in numpy/numba, **Python**

Closing remarks

- Our approach
  - Good performance overall
  - Unique: handles both weighted data and weight templates
  - Provides gof test statistic
  - Reference implementation in *iminuit* package
    - Unique: allows one to mix non-parametric templates with parametric components
  - No implementation in ROOT yet; interest in collaborating?

- Thoughts on other approaches
  - Barlow-Beeston: could integrate SPD and provide gof test statistic
  - Argüelles, Schneider & Yuan: can probably be extended to weighted data

- Further improvements?
  - Templates cannot adjust for data / simulation discrepancies in **template shape**
  - Potential solution: simultaneously fit stiff monotonic transform that distorts simulation sample
    - Similar idea: RooStats::BernsteinCorrection

- For completeness: complementary approach is using bootstrap
  - Perform naive fit of $\vec{y}$ with fixed templates
  - Bootstrap uncertainties of estimates by resampling both data and template samples
  - Can handle situations in which weights are not iid
  - Computationally expensive

Thank you

References

- Baker & Cousins, NIM 221 (1984) 437-442
- Barlow & Beeston, Comput. Phys. Commun. 77 (1993) 219-228
- Conway, proceedings for PHYSTAT 2011 (2011) doi:10.5170/CERN-2011-006.115
- Bohm & Zech, NIM A 748 (2014) 1-6
- Argüelles, Schneider, Yuan, J. High Energy Phys. 2019(6) (2019) 1-18
- Dembinski & Abdelmottelb, Eur. Phys. J. C (2022) 82: 1043