UNIVERSITÀ
DI TORINO

TMD impact on W mass

What we have found and what we want to do

Pontina Sara

Introduction

2018-2019 studies

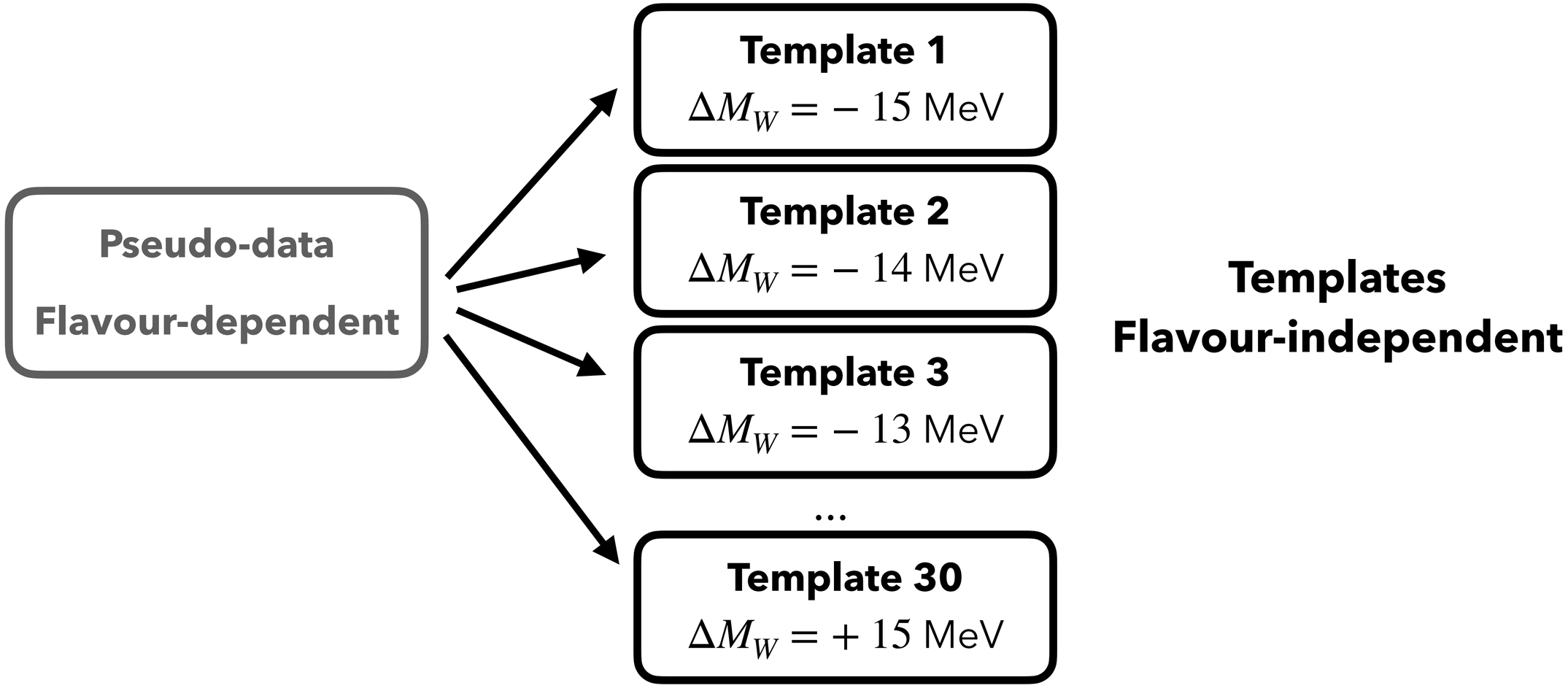
Exploratory studies investigated the **impact of flavour-dependent TMDs** on the **W boson mass**.

A simple **Gaussian ansatz** was used for the non-perturbative TMDs.

The methodology adopted was the **template fitting**.

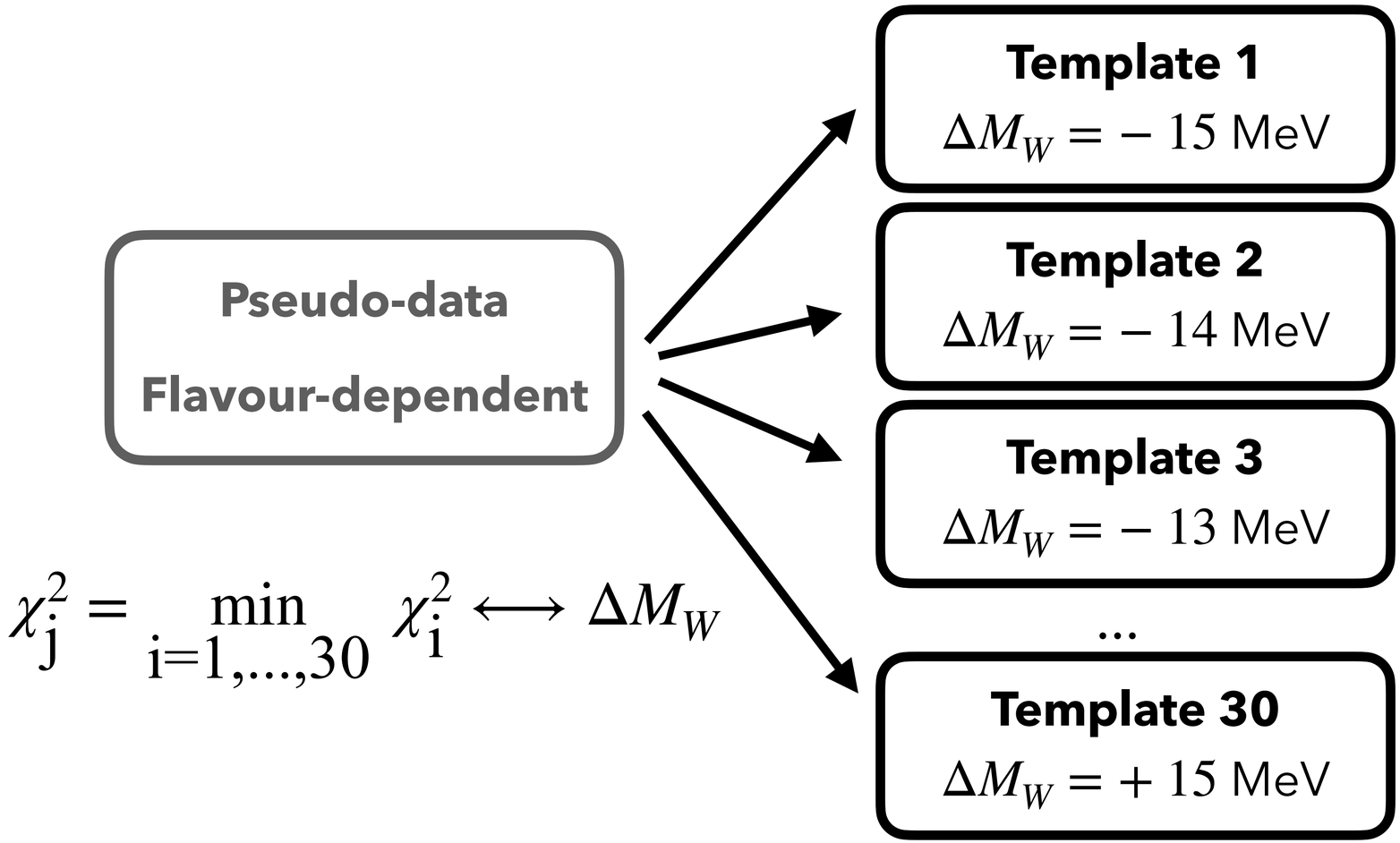
A. Bacchetta et al.; arXiv:1807.02101, G. Bozzi et al.; arXiv:1901.01162

Template fit technique



A. Bacchetta et al.; arXiv:1807.02101, G. Bozzi et al.; arXiv:1901.01162

Template fit technique

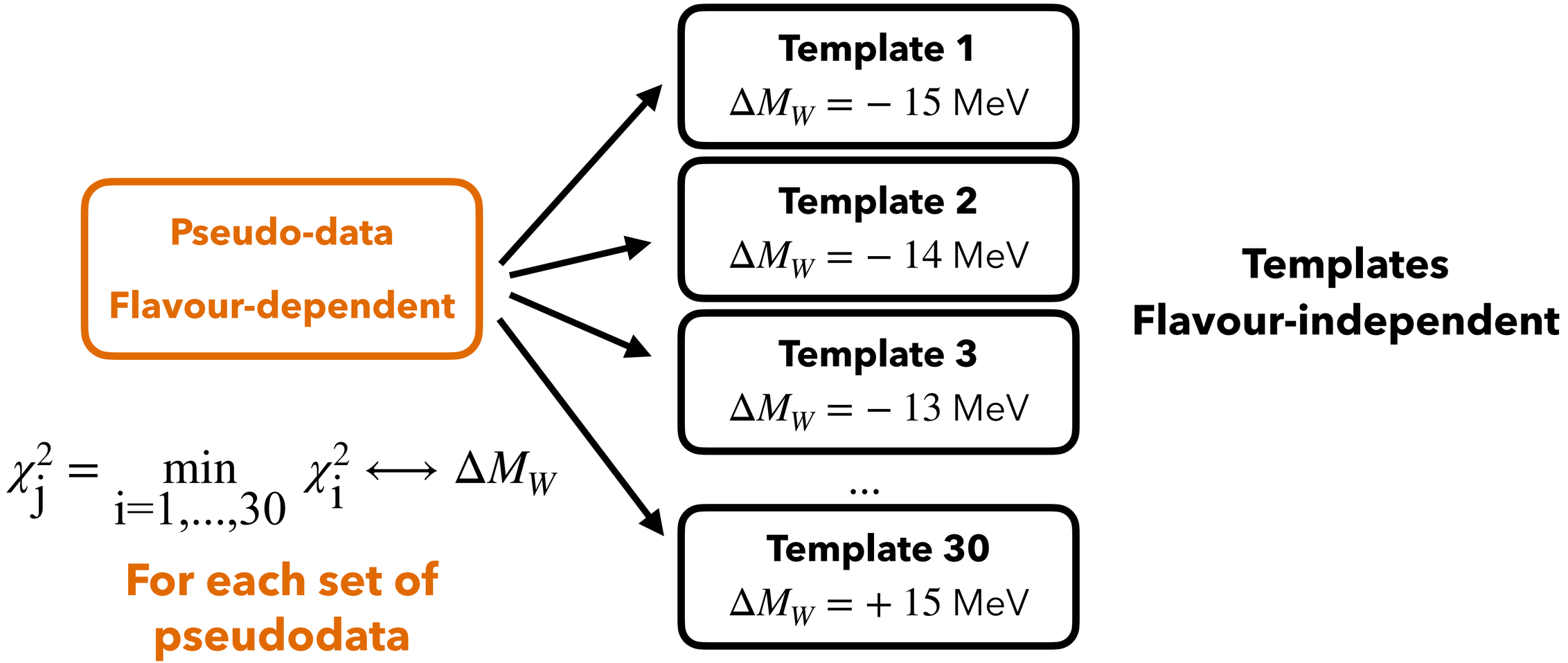


Templates
Flavour-independent

$$\chi^2_j = \min_{i=1,\ldots,30} \chi^2_i \longleftrightarrow \Delta M_W$$

A. Bacchetta et al.; arXiv:1807.02101, G. Bozzi et al.; arXiv:1901.01162

Template fit technique



A. Bacchetta et al.; arXiv:1807.02101, G. Bozzi et al.; arXiv:1901.01162

2018-2019 findings

Observed shifts in M_W from flavour-dependent TMDs:

$$-6 \leq \Delta M_{W^+} \leq 9 \text{ MeV}$$

$$-4 \leq \Delta M_{W^-} \leq 3 \text{ MeV}$$

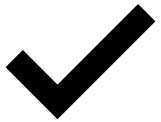
A. Bacchetta et al.; arXiv:1807.02101, G. Bozzi et al.; arXiv:1901.01162

Master thesis project

Inspired by previous works

Master thesis project

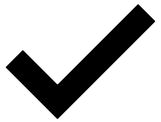
Inspired by previous works



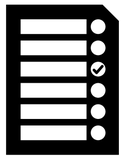
Adopted same methodology as a starting point

Master thesis project

Inspired by previous works



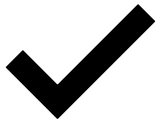
Adopted same methodology as a starting point



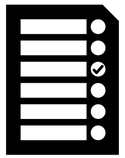
Extended with new modifications and improvements

Master thesis project

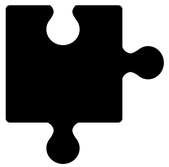
Inspired by previous works



Adopted same methodology as a starting point



Extended with new modifications and improvements



Extended to consider different TMDs extractions

Main modifications

DYqT, DYRes  DYTurbo

Main modifications

DYqT, DYRes → DYTurbo

Gaussian ansatz → **MAP 2022 and 2024 TMDs**

MAP implementation in DYTurbo

In order to use MAP22 and MAP24 parametrizarion in the template fitting, the **functional form** has been **implemented in the code**:

$$f_1(x, b_T; \zeta) = \exp\Big[- g_2^2 \frac{b_T^2}{4} \ln \frac{\zeta}{Q_0^2} \Big] \times \frac{g_1(x)\, e^{-g_1(x)\frac{b_T^2}{4}} + (\lambda)^2 (g_2(x))^2 \left(1 - g_2(x)\frac{b_T^2}{4}\right) e^{-g_2(x)\frac{b_T^2}{4}} + (\lambda_2)^2 g_3(x)\, e^{-g_3(x)\frac{b_T^2}{4}}}{g_1(x) + (\lambda)^2 (g_2(x))^2 + (\lambda_2)^2 g_3(x)}$$

MAP implementation in DYTurbo

In order to use MAP22 and MAP24 parametrizarion in the template fitting, the **functional form** has been **implemented in the code**:

$$f_1(x, b_T; \zeta) = \exp\Big[- g_2^2 \frac{b_T^2}{4} \ln \frac{\zeta}{Q_0^2} \Big] \times \frac{g_1(x)\, e^{-g_1(x)\frac{b_T^2}{4}} + (\lambda)^2 (g_2(x))^2 \left(1 - g_2(x)\frac{b_T^2}{4}\right) e^{-g_2(x)\frac{b_T^2}{4}} + (\lambda_2)^2 g_3(x)\, e^{-g_3(x)\frac{b_T^2}{4}}}{g_1(x) + (\lambda)^2 (g_2(x))^2 + (\lambda_2)^2 g_3(x)}$$

And the corresponding **b * -prescription** has been implemented, too:

$$\bar{b}^*_{MAP}(b_T) = b_{max} \left(\frac{1 - \exp(-b_T^4/b_{max}^4)}{1 - \exp(-b_T^4/b_{\min}^4)} \right)^{1/4}$$

$$b_{max} = 2e^{-\gamma_E} \simeq 1.123\ GeV^{-1}$$

$$b_{\min} = \frac{2e^{-\gamma_E}}{\mu}$$

Main modifications

DYqT, DYRes → DYTurbo

Gaussian ansatz → **MAP 2022 and 2024 TMDs**

Main modifications

DYqT, DYRes → DYTurbo

Gaussian ansatz → **MAP 2022 and 2024 TMDs**

Hadron colliders pseudodata → Focus on kinematics and statistics of **ATLAS**

The ultimate goal

The final goal of this project would be to **collaborate with the experimental community** in order to include these effects in the W boson mass measurements.

Results at NLL

MAP implementation in DYTurbo

In order to use MAP22 and MAP24 parametrizarion in the template fitting, the **functional form** has been **implemented in the code**:

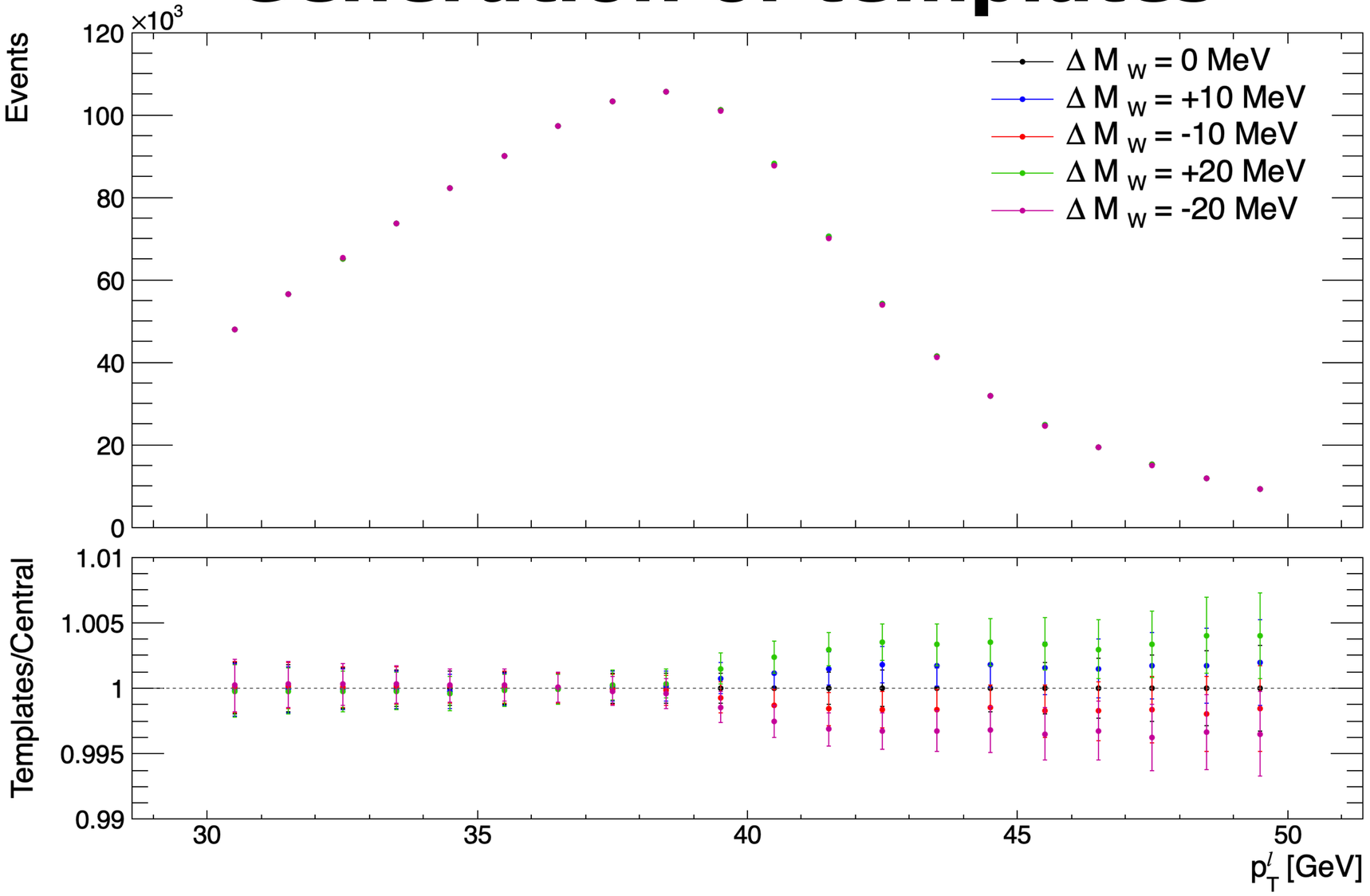
$$f_1(x, b_T; \zeta) = \exp\left[- g_2^2 \frac{b_T^2}{4} \ln \frac{\zeta}{Q_0^2} \right] \times \frac{g_1(x)\, e^{-g_1(x)\frac{b_T^2}{4}} + (\lambda)^2 (g_2(x))^2 \left(1 - g_2(x)\frac{b_T^2}{4}\right) e^{-g_2(x)\frac{b_T^2}{4}} + (\lambda_2)^2 g_3(x)\, e^{-g_3(x)\frac{b_T^2}{4}}}{g_1(x) + (\lambda)^2 (g_2(x))^2 + (\lambda_2)^2 g_3(x)}$$

And the corresponding **b* -prescription** has been implemented, too:

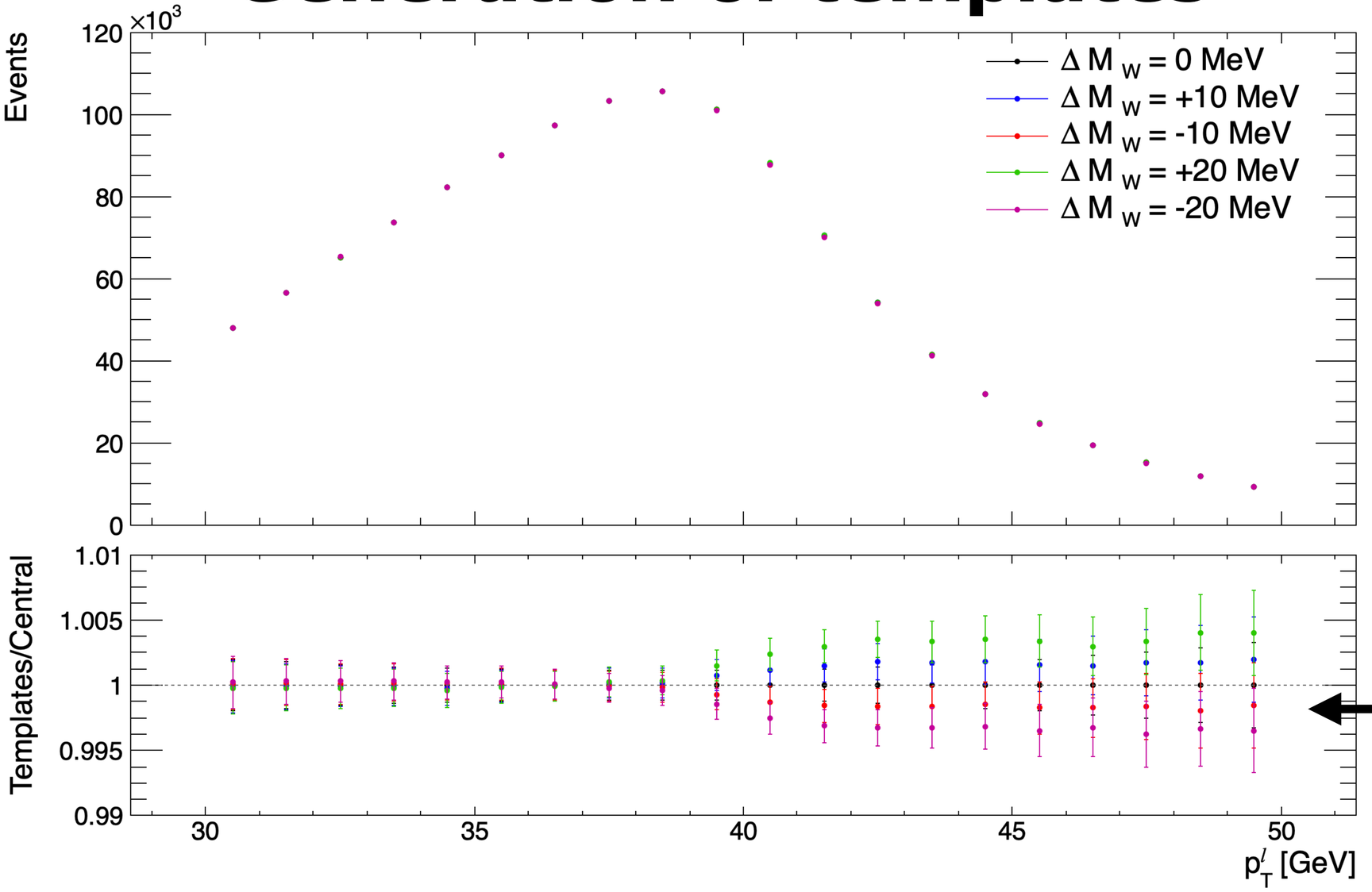
$$\bar{b}^*_{MAP}(b_T) = b_{max} \left(\frac{1 - \exp(-b_T^4/b_{max}^4)}{1 - \exp(-b_T^4/b_{\min}^4)} \right)^{1/4}$$

<u>Important</u>: For simplicity, in the preliminary NLL results we have used a simpler b* -prescription, without the b_{min}

Generation of templates

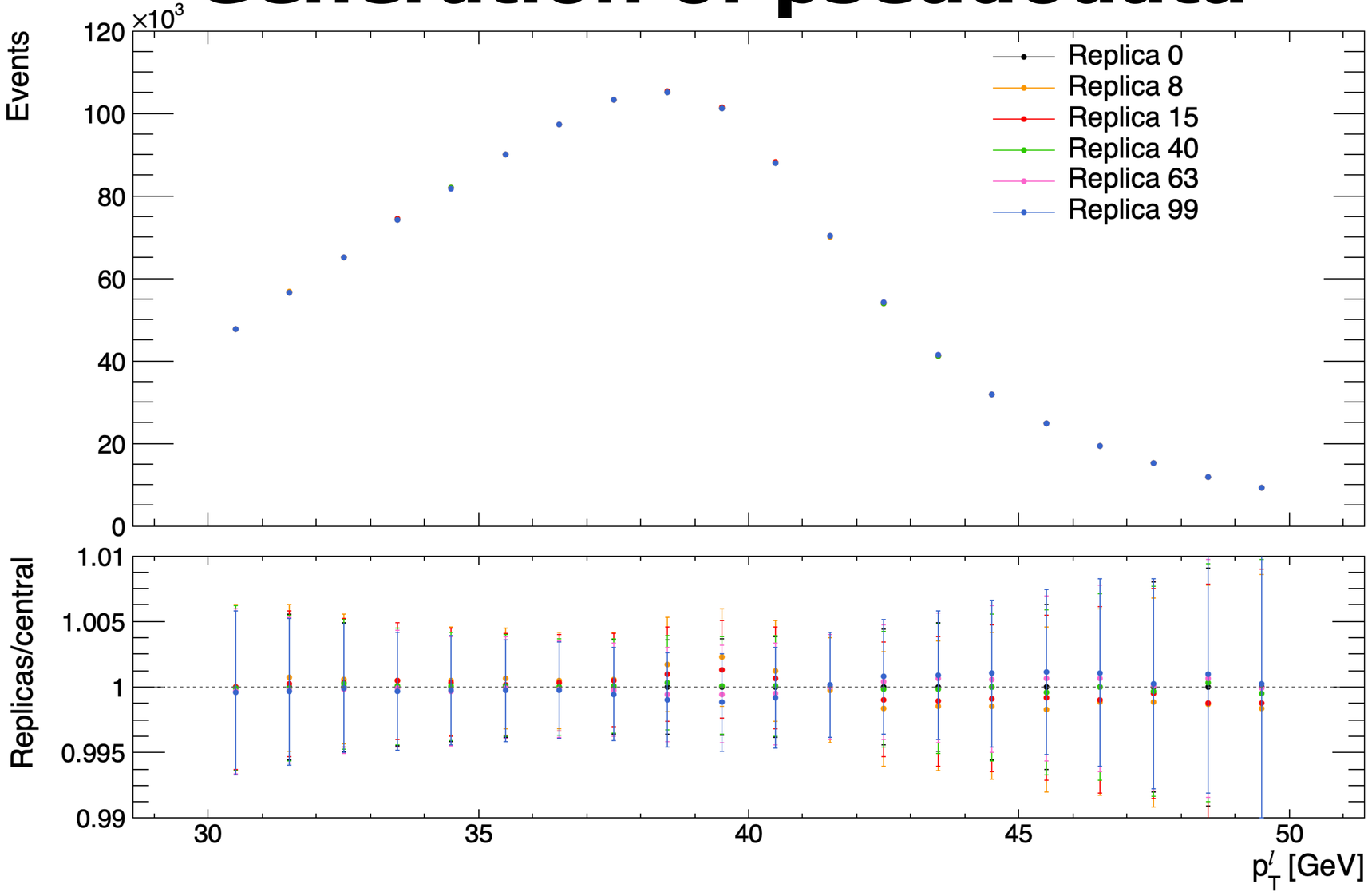


Generation of templates

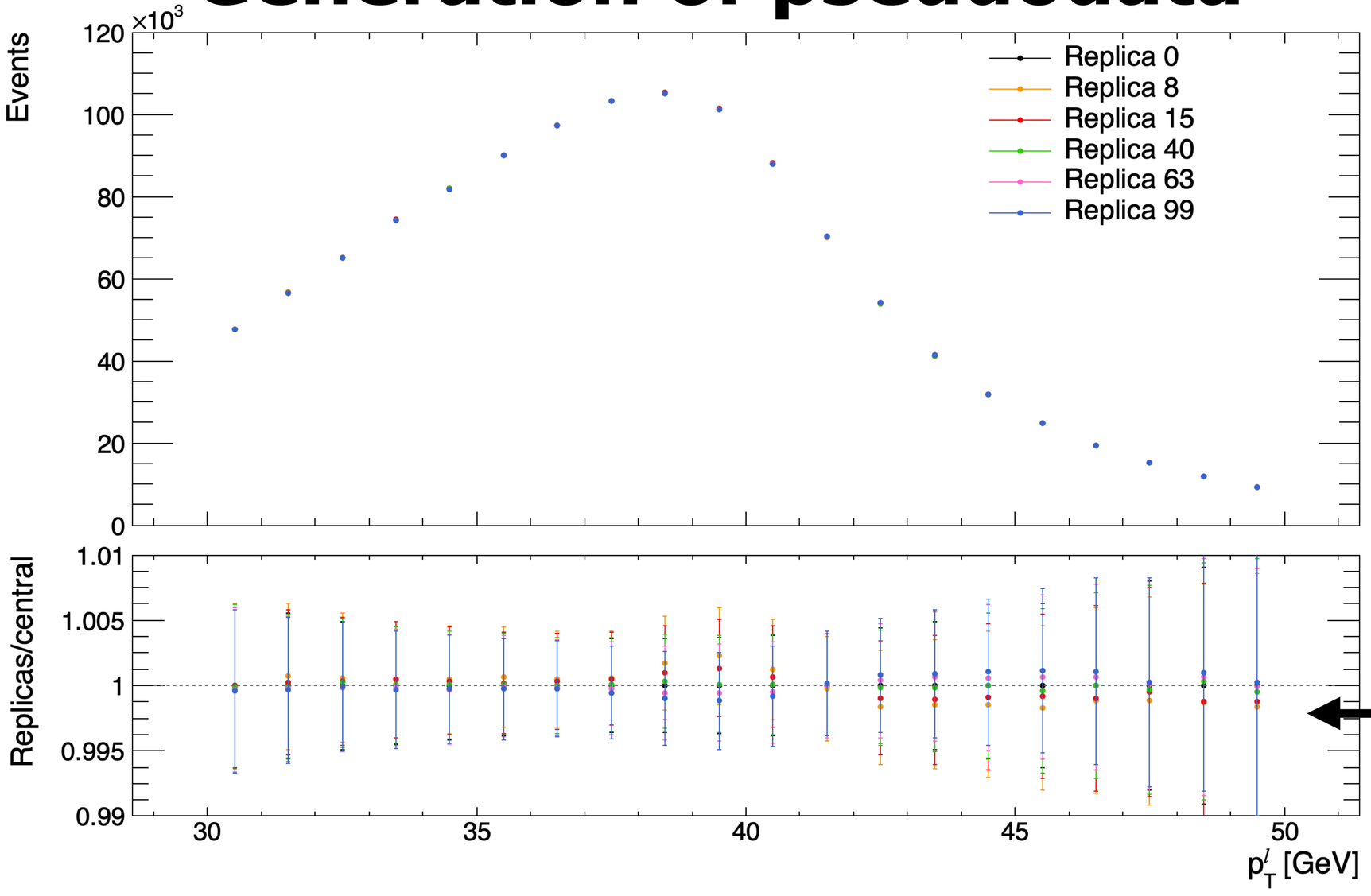


Same TMD replica,
different masses

Generation of pseudodata

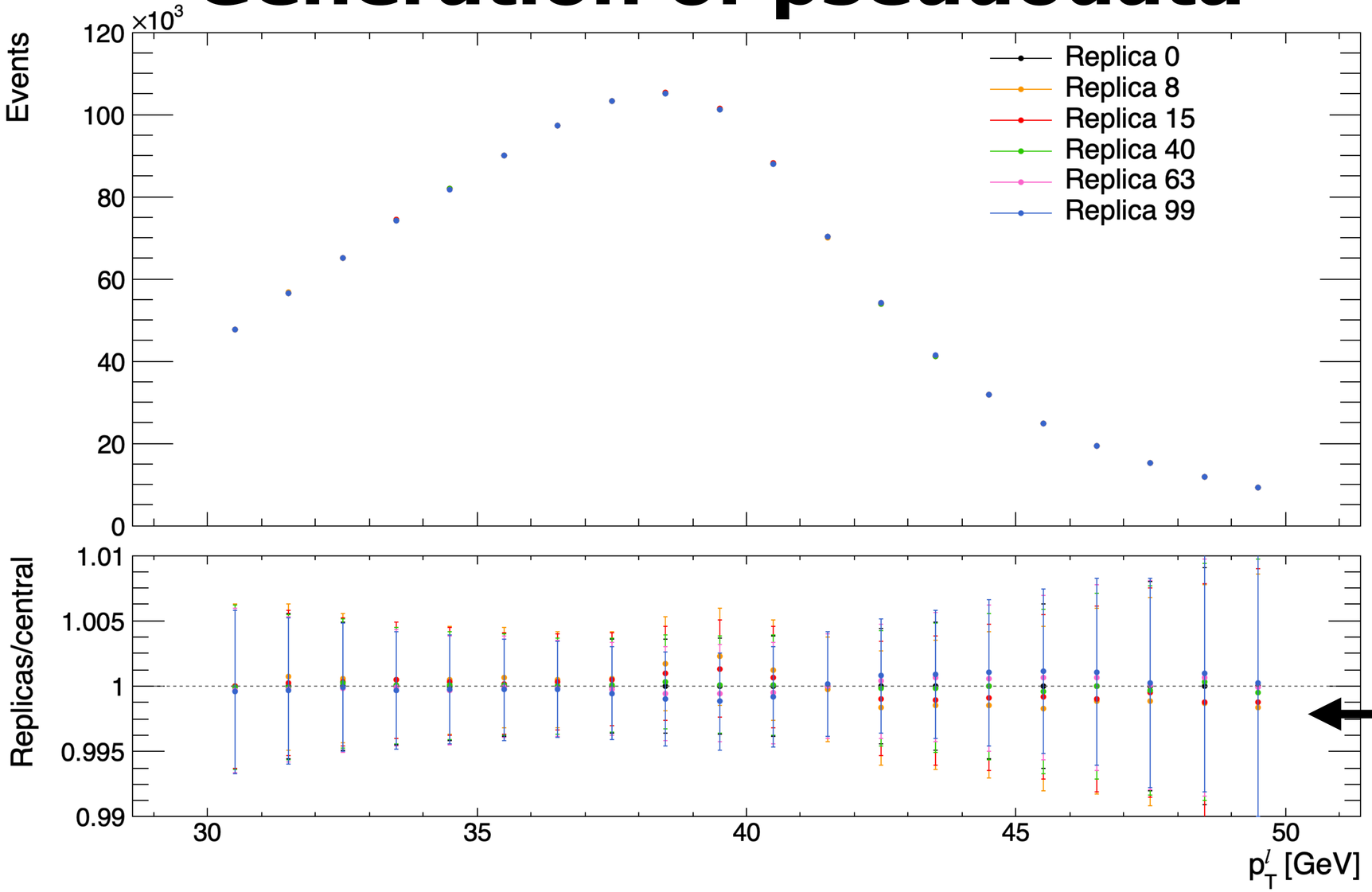


Generation of pseudodata



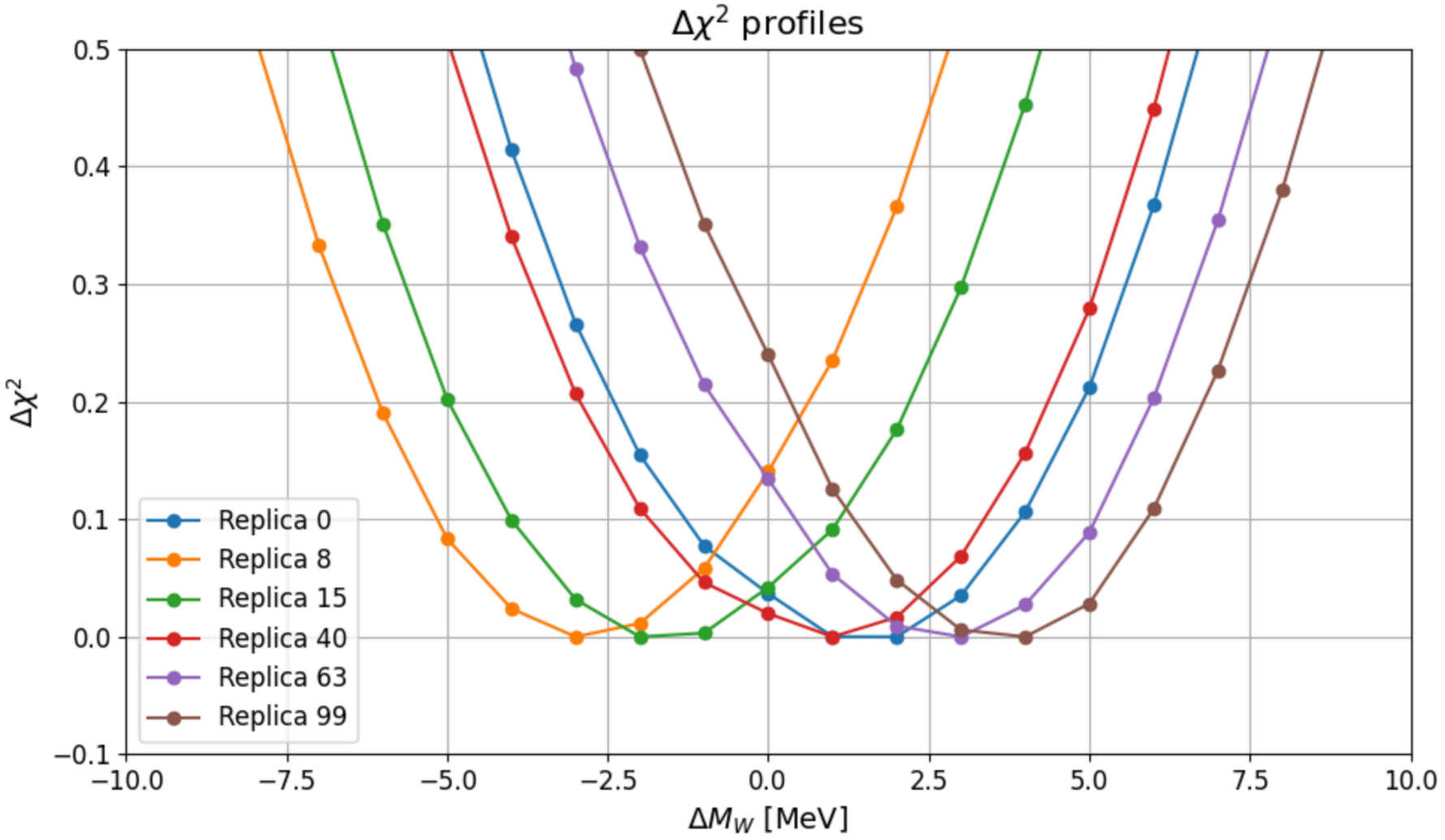
Same mass,
different TMD replicas

Generation of pseudodata

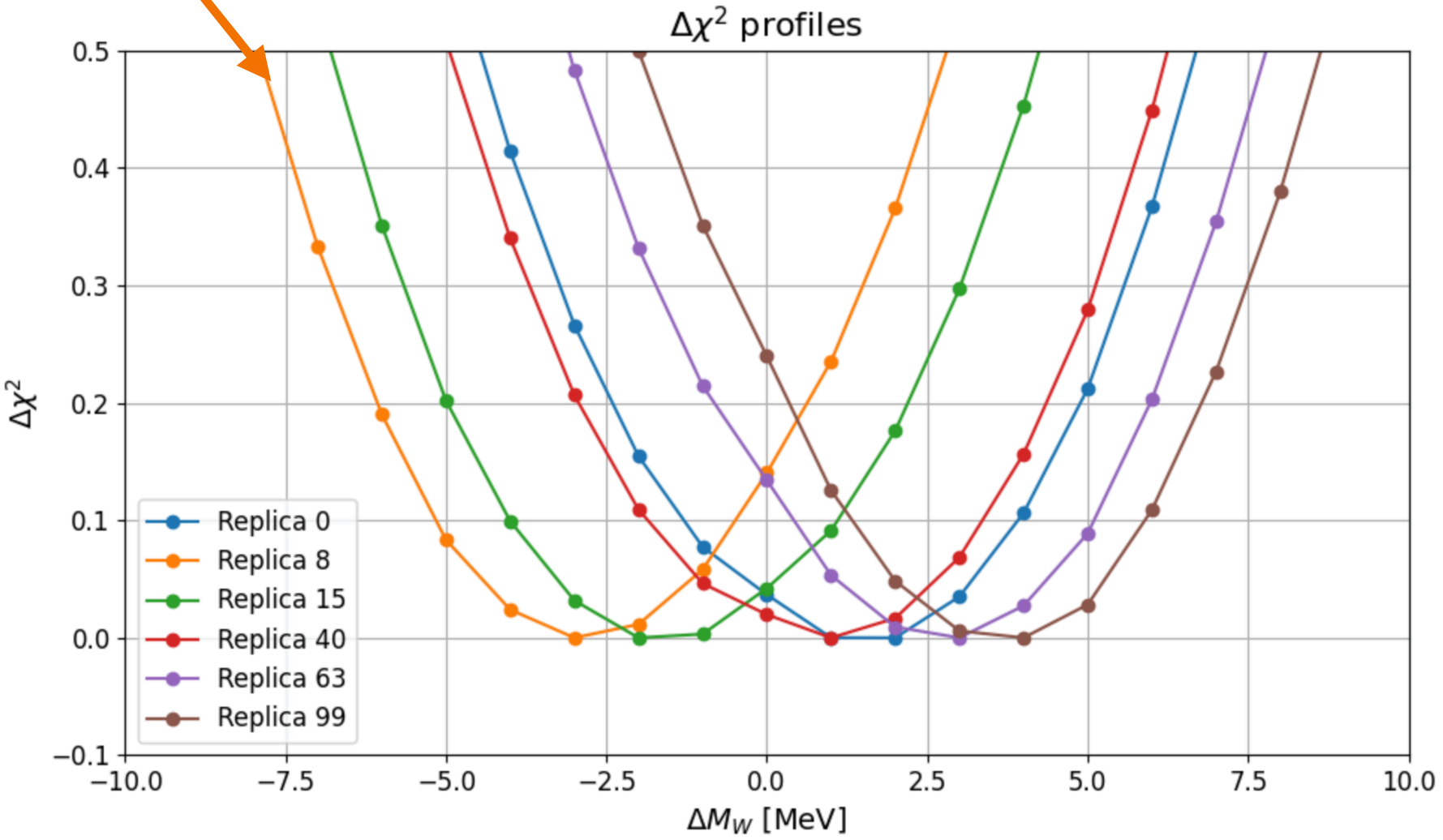


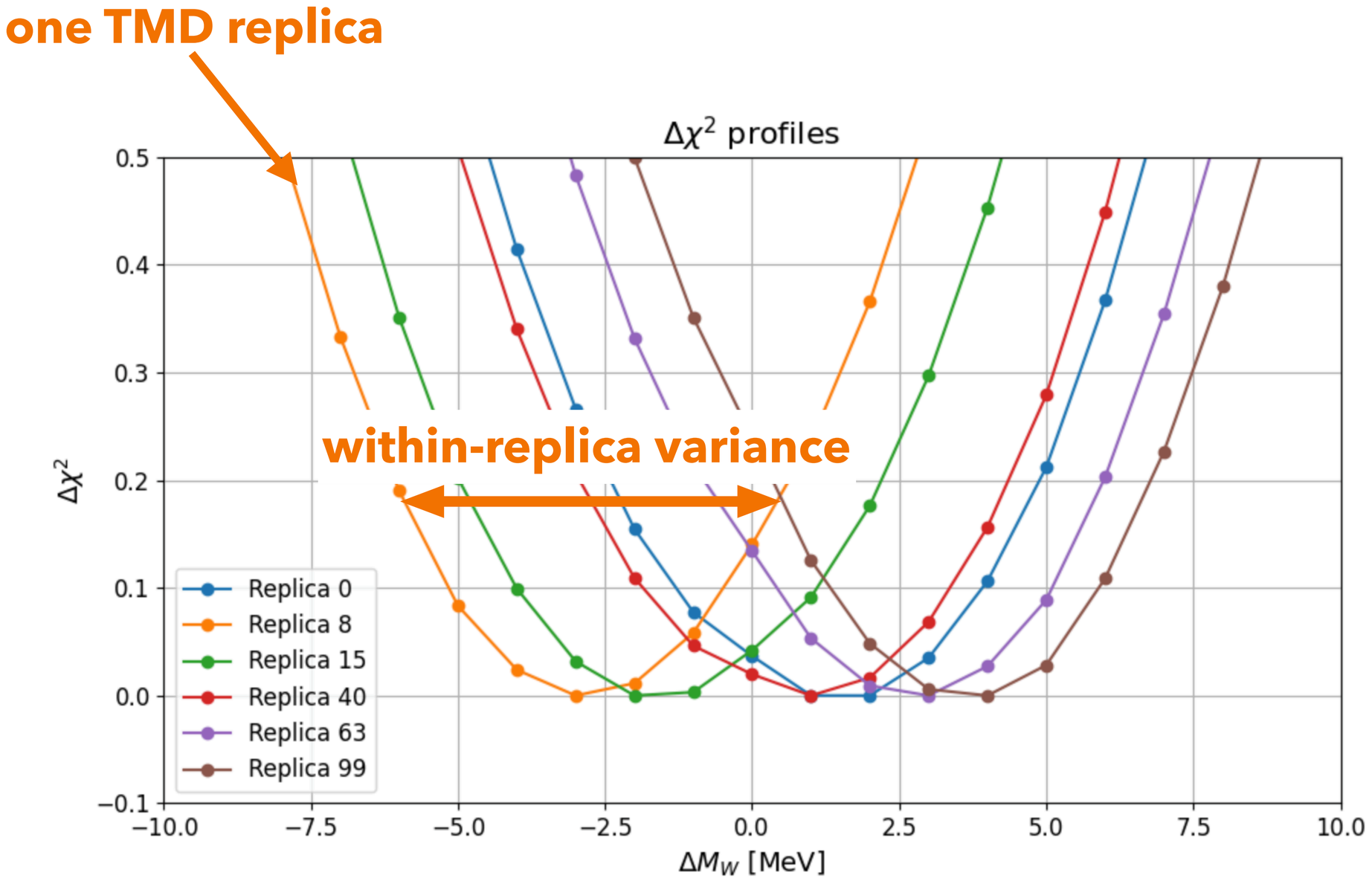
Same mass, **different TMD replicas**

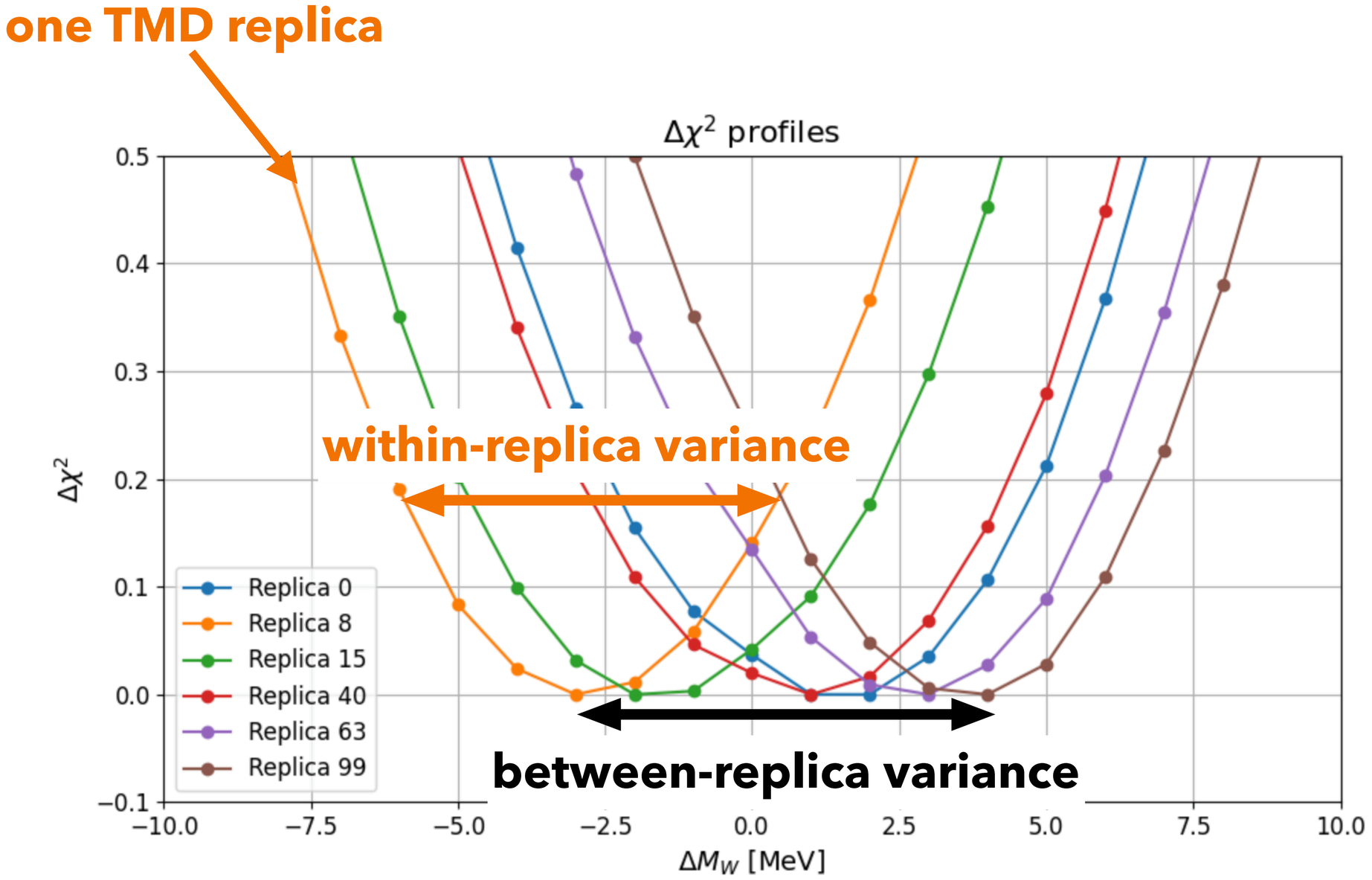
<u>Important</u>: We have varied only TMD parameters (the PDF member has been fixed)

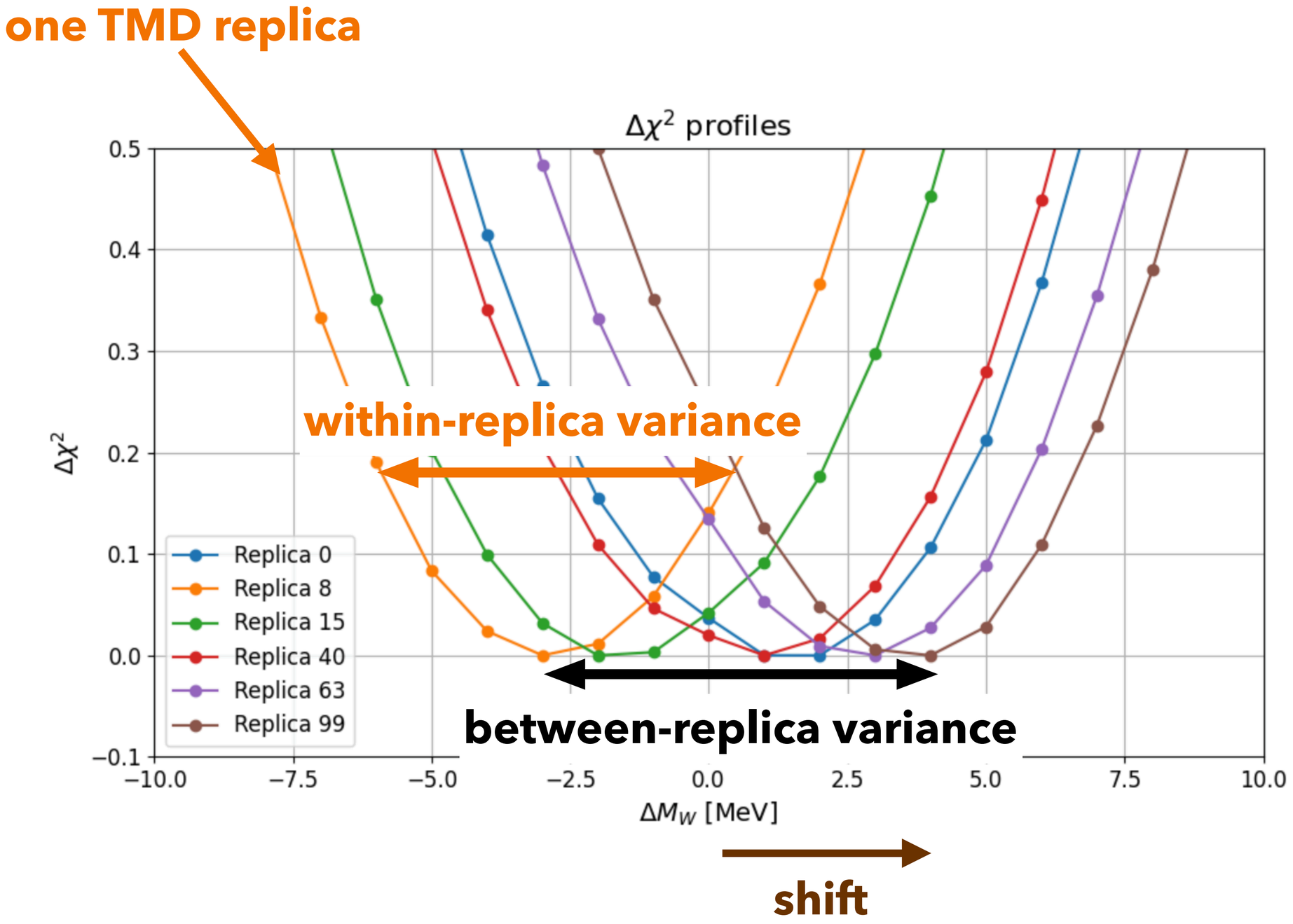

$\Delta\chi^2$ profiles
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
$\Delta\chi^2$
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
ΔM_W [MeV]
Replica 0
Replica 8
Replica 15
Replica 40
Replica 63
Replica 99

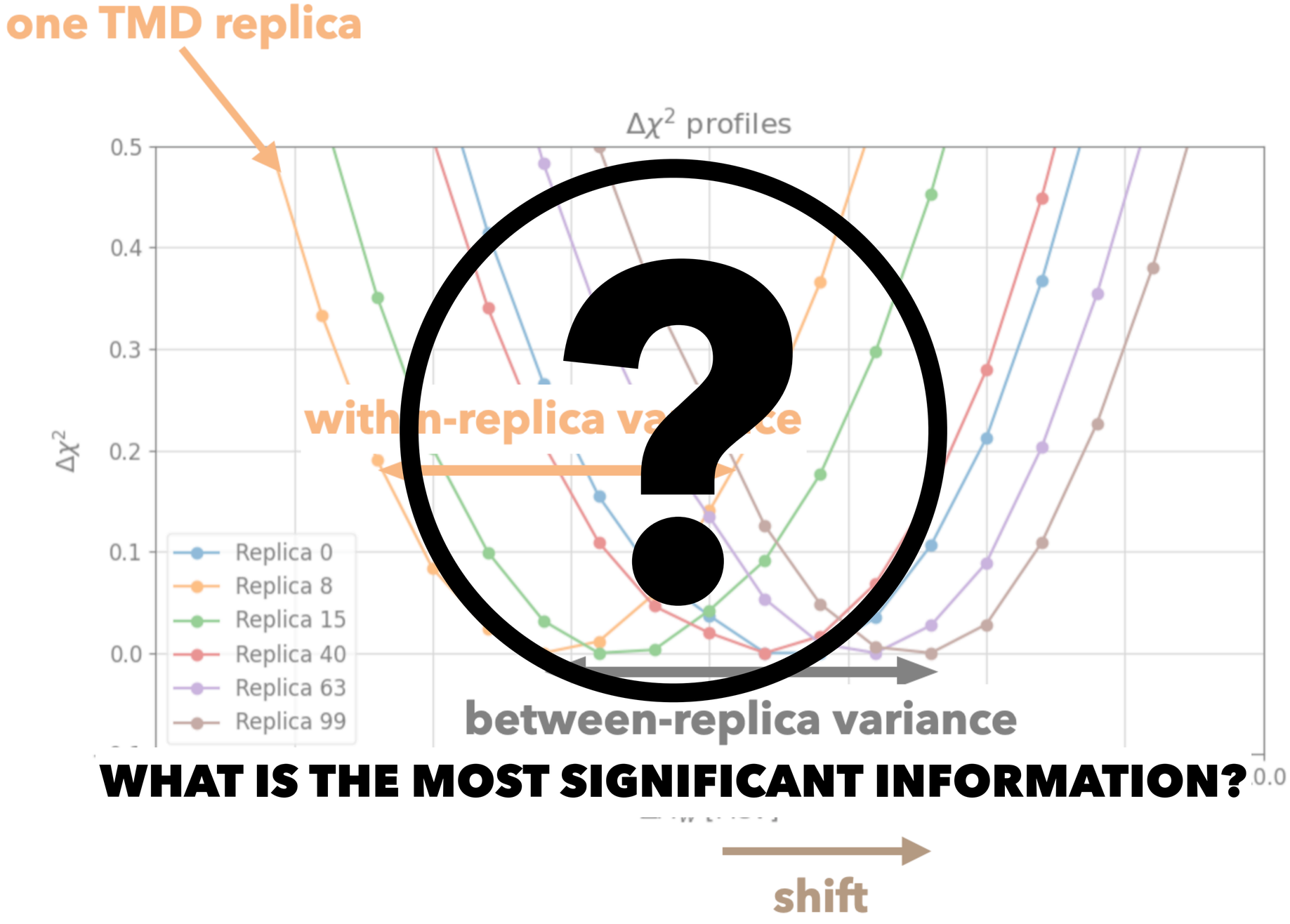
one TMD replica




one TMD replica
$\Delta\chi^2$ profiles
within-replica variance
$\Delta\chi^2$
ΔM_W [MeV]
Replica 0
Replica 8
Replica 15
Replica 40
Replica 63
Replica 99
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


one TMD replica
$\Delta\chi^2$ profiles
within-replica variance
between-replica variance
Replica 0
Replica 8
Replica 15
Replica 40
Replica 63
Replica 99
$\Delta\chi^2$
ΔM_W [MeV]
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0


one TMD replica
$\Delta\chi^2$ profiles
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
$\Delta\chi^2$
within-replica variance
Replica 0
Replica 8
Replica 15
Replica 40
Replica 63
Replica 99
between-replica variance
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
ΔM_W [MeV]
shift


one TMD replica
$\Delta\chi^2$ profiles
0.5
0.4
0.3
0.2
0.1
0.0
$\Delta\chi^2$
Replica 0
Replica 8
Replica 15
Replica 40
Replica 63
Replica 99
within-replica variance
between-replica variance
WHAT IS THE MOST SIGNIFICANT INFORMATION?
shift

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

results obtained with 100 replicas

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1		2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1		1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2.3 MeV	7.5 MeV	0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

flavour-dependence does not seem to have a big impact

Outcome

TMD parametrizations	Range of the shifts	Mean of the shifts	Within-replica error	Between-replica error	Global errors
MAP24 flavour independent	[-4,+7] MeV	1.6 MeV	7.5 MeV	2.2 MeV	7.8 MeV
MAP24 flavour dependent	[-5,+5] MeV	-1.6 MeV	7.5 MeV	1.6 MeV	7.7 MeV
MAP22	[-2,-3] MeV	-2		0.5 MeV	7.5 MeV
MAP22 Drell-Yan-only	[-5,+20] MeV	-1.4 MeV	7.6 MeV	5.7 MeV	9.4 MeV

excluding DIS in the global fit increases the uncertainty

More recent work

The current goals

- Repeat the procedure at **N3LL**
- Consider **both $\mathbf{W^+}$ and $\mathbf{W^-}$ channels**
- Clarify the interplay between PDFs and TMDs
- Produce also MAP24 Drell-Yan-only
- Other suggestions?

The current goals

- Repeat the procedure at **N3LL**
- Consider **both W^+ and W^- channels**
- Clarify the interplay between PDFs and TMDs
- Produce also MAP24 Drell-Yan-only
- Other suggestions?

Backup

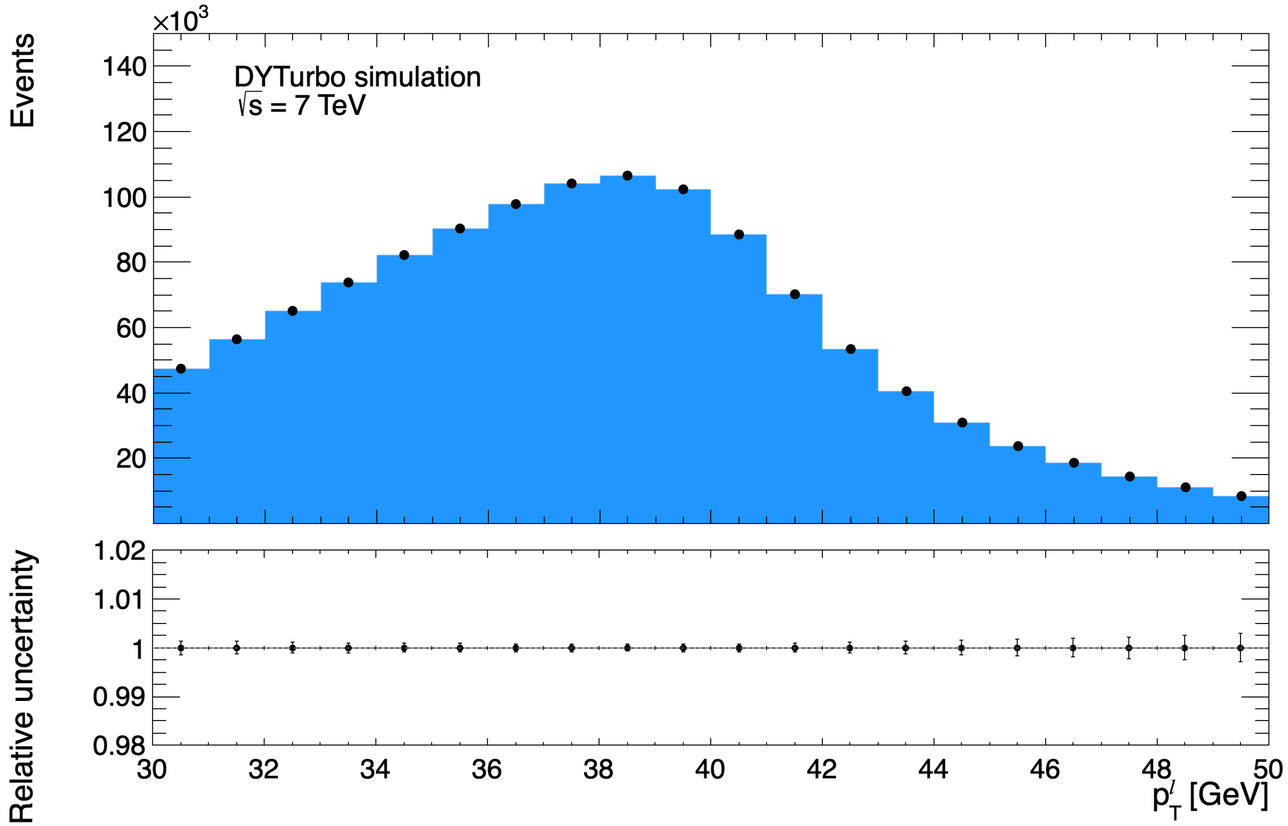
The present situation

The present situation

The present situation

Original choices

Pseudodata at N3LL



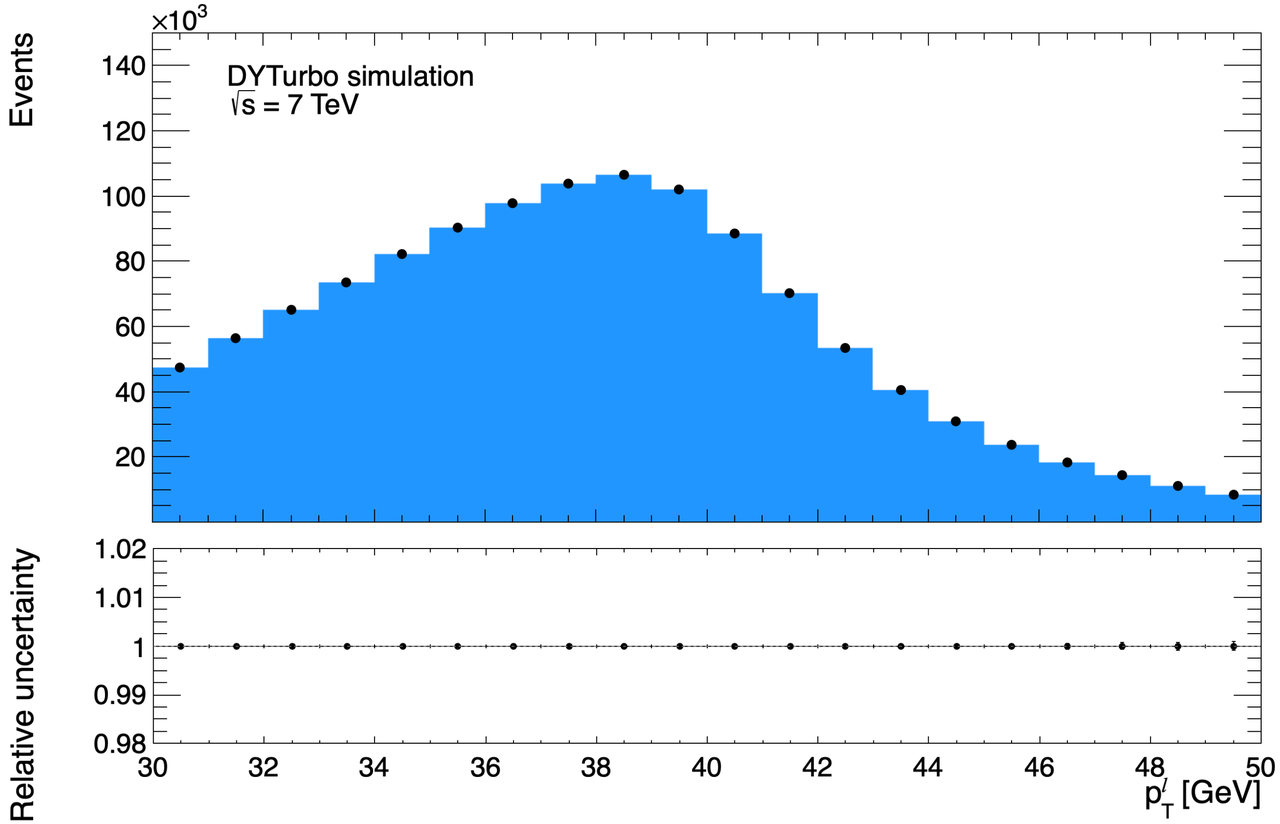
Average relative error: $\sim 0.13\,\%$

Calculation time: **1.5-2 days**

Templates at N3LL

Average relative error: $\sim 0.05\,\%$

Calculation time: **15-20 days**



Pathologies in qT output

Pseudodata output in q_T (there is only one bin, 1-30 GeV):

Resummation	Counter term	V+J Real	V+J Virtual	TOTAL
1.20664 ± 0.00015 ·10⁺⁰⁶ (39.9hr)	−1.7772 ± 0.0019 ·10⁺⁰⁶ (2.49mi)	−2.999 ± 0.058 ·10⁺⁰⁶ (7.5hr)	4.7648 ± 0.009 ·10⁺⁰⁶ (2.36mi)	1.195 ± 0.059 ·10⁺⁰⁶ (47.5hr)
1.20664 ± 0.00015 ·10⁺⁰⁶ (39.9hr)	−1.7772 ± 0.0019 ·10⁺⁰⁶ (2.49mi)	−2.999 ± 0.058 ·10⁺⁰⁶ (7.5hr)	4.7648 ± 0.009 ·10⁺⁰⁶ (2.36mi)	1.195 ± 0.059 ·10⁺⁰⁶ (47.5hr)

Each term seems **reasonable**.

For **templates**:

Resummation	Counter term	V+J Real	V+J Virtual	TOTAL
1.20635 ± 4.7e−05 ·10⁺⁰⁶ (295hr)	280 ± 320 ·10⁺⁰³ (549ms)	−2.974 ± 0.023 ·10⁺⁰⁶ (58.3hr)	24 ± 16 ·10⁺⁰⁶ (479ms)	23 ± 16 ·10⁺⁰⁶ (353hr)
1.20635 ± 4.7e−05 ·10⁺⁰⁶ (295hr)	280 ± 320 ·10⁺⁰³ (549ms)	−2.974 ± 0.023 ·10⁺⁰⁶ (58.3hr)	24 ± 16 ·10⁺⁰⁶ (479ms)	23 ± 16 ·10⁺⁰⁶ (353hr)

We saw very big errors for Counter Term e V+J Virtual; what are the reasons of this behaviour?

Possible explanation

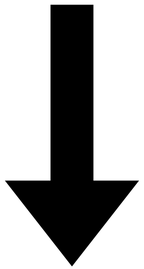
Vegas paramters are defined in src/settings.h

```
int vegasncallsBORN  ;
int vegasncallsCT    ;
int vegasncallsVJLO   ;
long long int vegasncallsVJREAL ;
int vegasncallsVJVIRT ;
```

A parametri defined as int can’t assume values which are larger than 2×10^9.

Vegas calls for this test

```
 Resummation:          vegas       ncalls =    60000000
Counter term:       vegas 8D       ncalls =  2400000000
    V+J Real:          vegas       ncalls = 24000000000
 V+J Virtual:          vegas       ncalls =  2400000000
```



X 10, for the templates

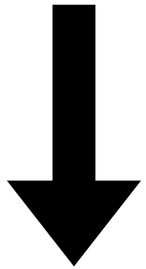
```
 Resummation:          vegas       ncalls =   600000000
Counter term:       vegas 8D       ncalls = 24000000000
    V+J Real:          vegas       ncalls =240000000000
 V+J Virtual:          vegas       ncalls = 24000000000
```

Vegas calls for this test

```
 Resummation:          vegas       ncalls =    60000000
Counter term:       vegas 8D      ncalls =  2400000000
   V+J Real:          vegas       ncalls = 24000000000
 V+J Virtual:         vegas       ncalls =  2400000000
```

2.4×10^9

2.4×10^9



X 10, for the templates

```
 Resummation:          vegas       ncalls =   600000000
Counter term:       vegas 8D      ncalls = 24000000000
   V+J Real:          vegas       ncalls =240000000000
 V+J Virtual:         vegas       ncalls = 24000000000
```

2.4×10^{10}

2.4×10^{10}

Investigation

In order to get further details about this problema, we have decided to **print in the output file** some parameters, related to the Vegas integration algorithm, which are in src/cubacall.C

```
   const int mineval = opts.vegasncallsVJVIRT;
   const int maxeval = opts.vegasncallsVJVIRT;
   const int nstart = max(10, int(opts.vegasncallsVJVIRT/
cubacall::nst));
   const int nincrease = max(10,    int(opts.vegasncallsVJVIRT/10));
   const int nbatch = opts.cubanbatch;
```

Pseudodata output

For pseudodata, we got **negative values**:

```
mineval   = -1894967296
maxeval   = -1894967296
nstart    = 240000000
nincrease = 240000000
nbatch    = 1000
seed      = 123456
```

Modifications of the Mellin parameters

Setting for Mellin inversion

Using the previous settings, we have tried to study the **decrease of the calculation time**, modifying the following settings :

```
# Settings for the inverse Mellin integrations
mellininv = 0        # Strategy for the Mellin inversion (0 Gauss-Legendre, 1 Talbot)
mellinintervals = 1 # Number of intervals
mellinrule = 30      # Number of nodes (should be increased, when increasing ncycle below)

# Options for the Mellin inversion with Gauss-Legendre integration
zmax = 2             # Upper limit of the contour in the imaginary axis
ncycle = 15          # Number of pi-cycles in the contour (use 40 for mellin1d = false at high rapidity)
cpoint = 3           # Point of intersection of the contour with the real axis (scaled by 1/log(z))
cshift = 0.5         # Shift to the right of the point of intersection of the contour with the real axis
phi = 0.5            # Angle between the real axis and the linear contour in units of pi

mellincores = 1      # Number of parallel threads for the Mellin integration (not yet supported)
mellin1d = true      # Use 1d (y-integrated) or 2d (y-dependent) Mellin inversion
melup = 2            # Strategy for the update of the contour for the Mellin inversion (0; do not update; 1:
xspace = false       # Access PDFs in Bjorken-x space, without Mellin inversion (option available only in the
ctmellin = false     # Compute the counterterm in Mellin space
```

For the pseudodata

Changing the parameters as:

```
mellinrule = 30  ⟶  mellinrule = 10
  ncycle = 15    ⟶    ncycle = 6
```

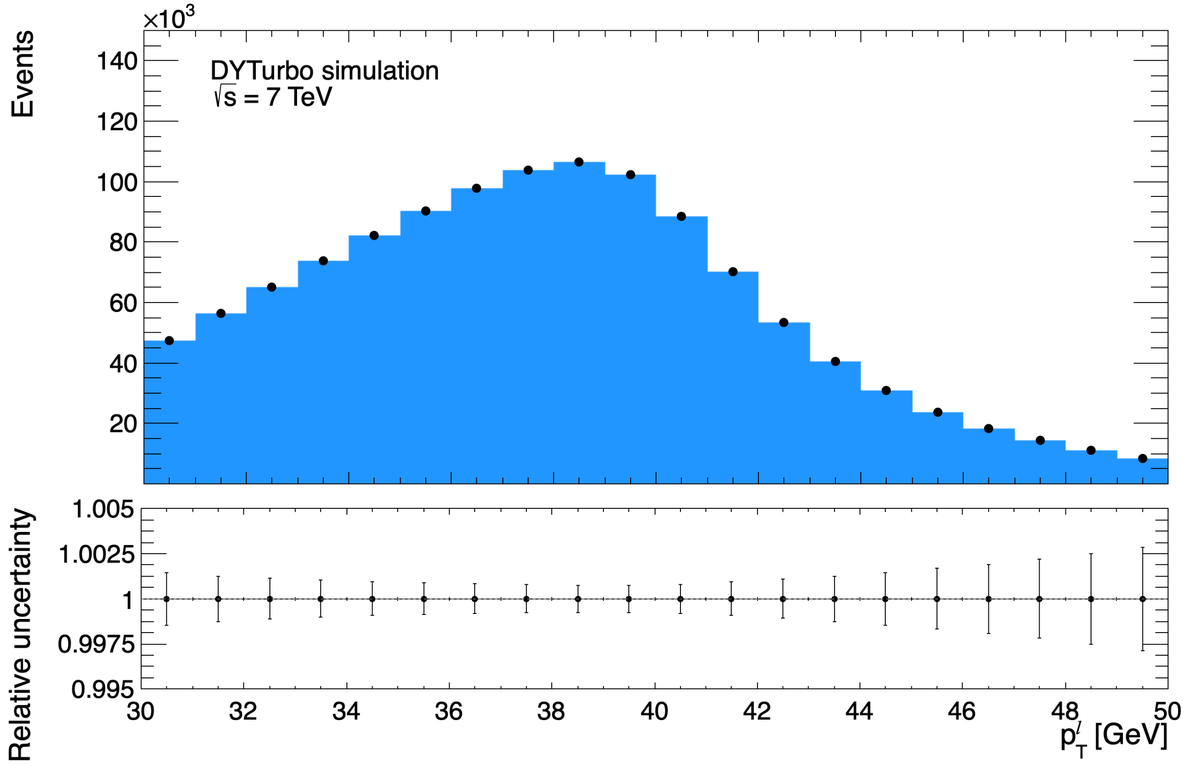
The calculation time of **pseudodata** decreased a lot:

2 days ⟶ **10-12 hours**

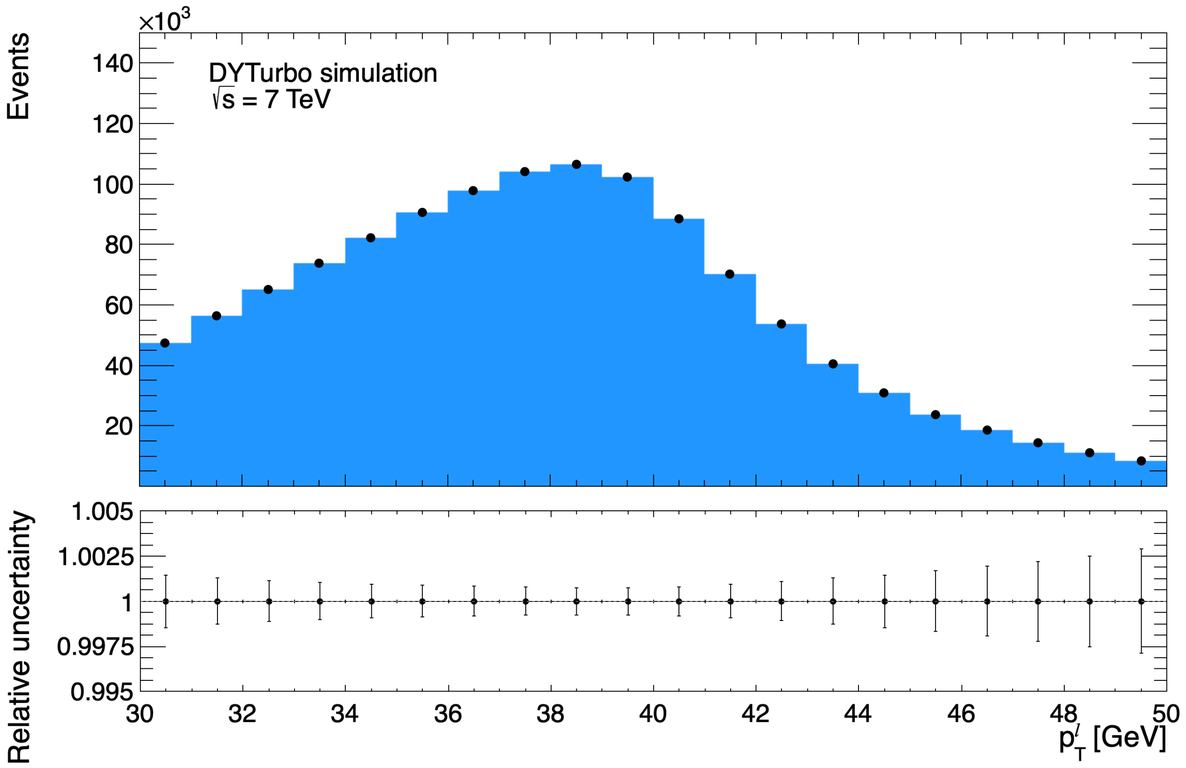
The errors are very similar to the previous case, and the spectrum is still reasonable.

Spectra in pT lepton

Before the modification:



After the modification:



For the templates

Applying the same modifications to the templates:

```
mellinrule = 30  ⟶  mellinrule = 10
  ncycle = 15    ⟶    ncycle = 6
```

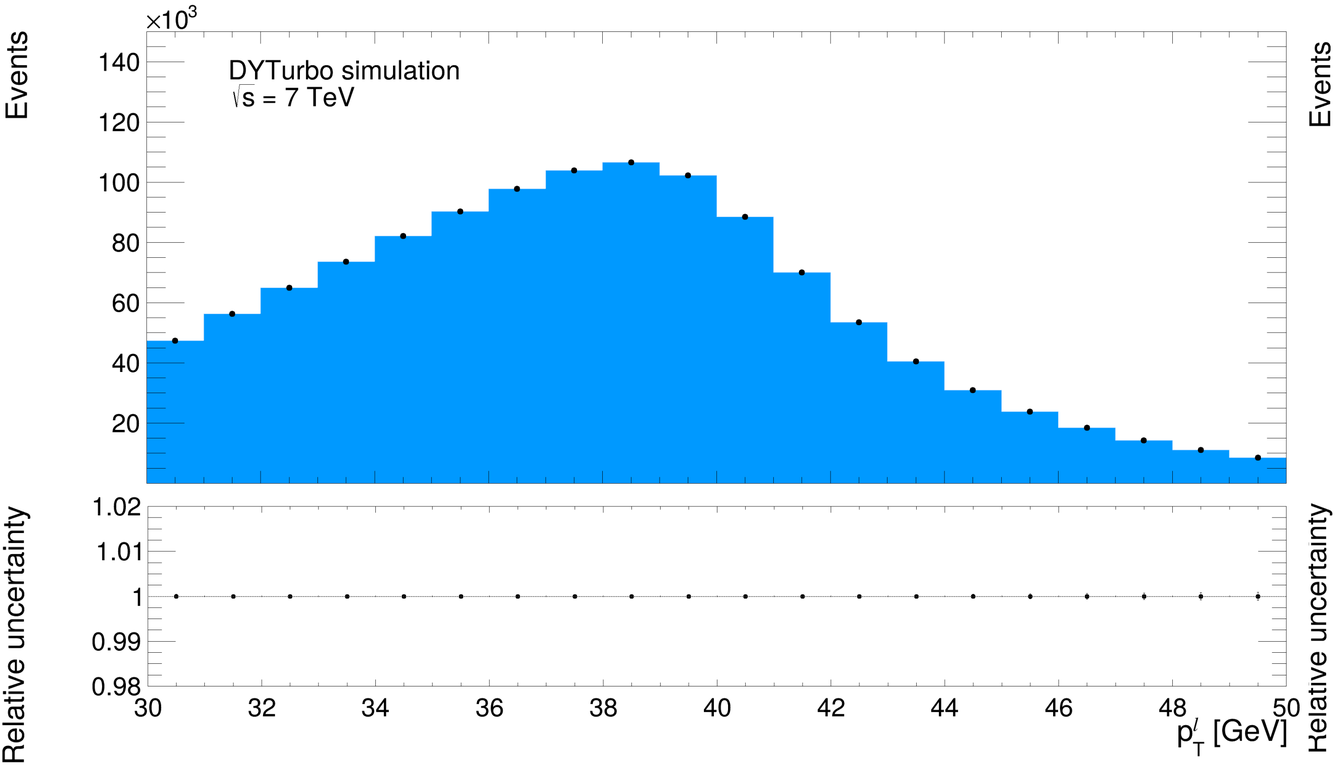
The calculation time of **templates** decreased similarly:

15-20 days ⟶ **6 days**

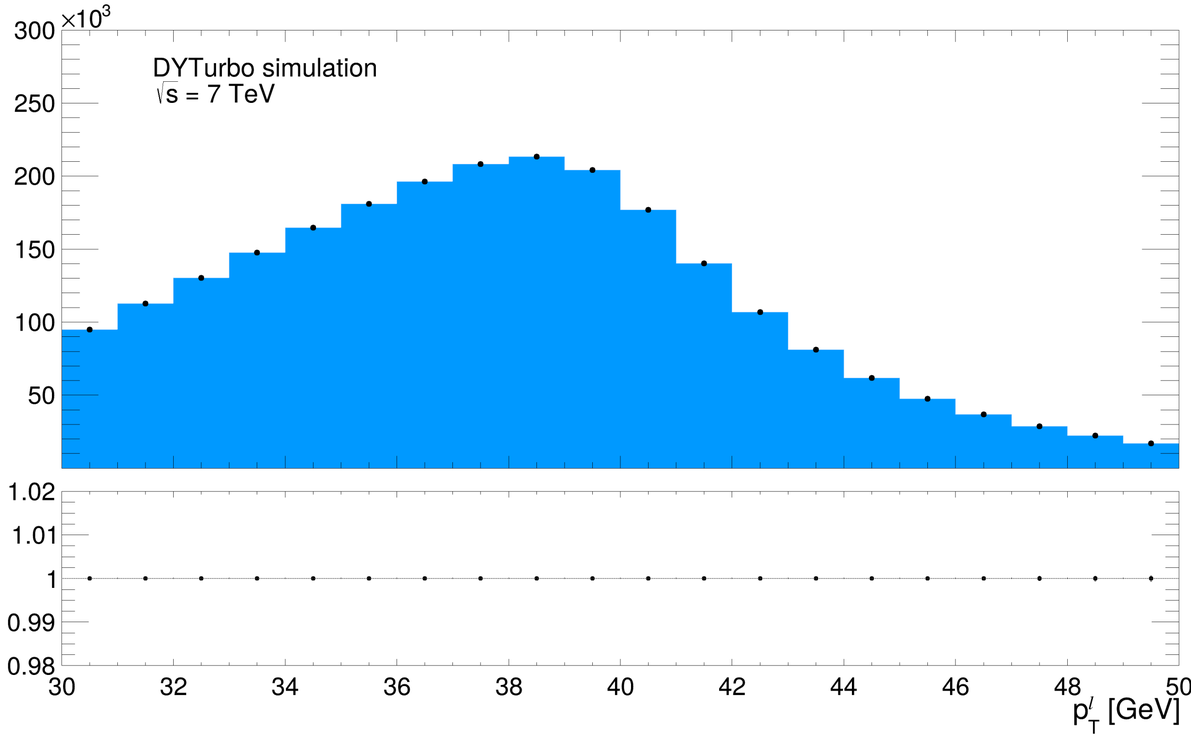
But there are some weirdnesses…

Spectra in pT lepton

Before the modification:



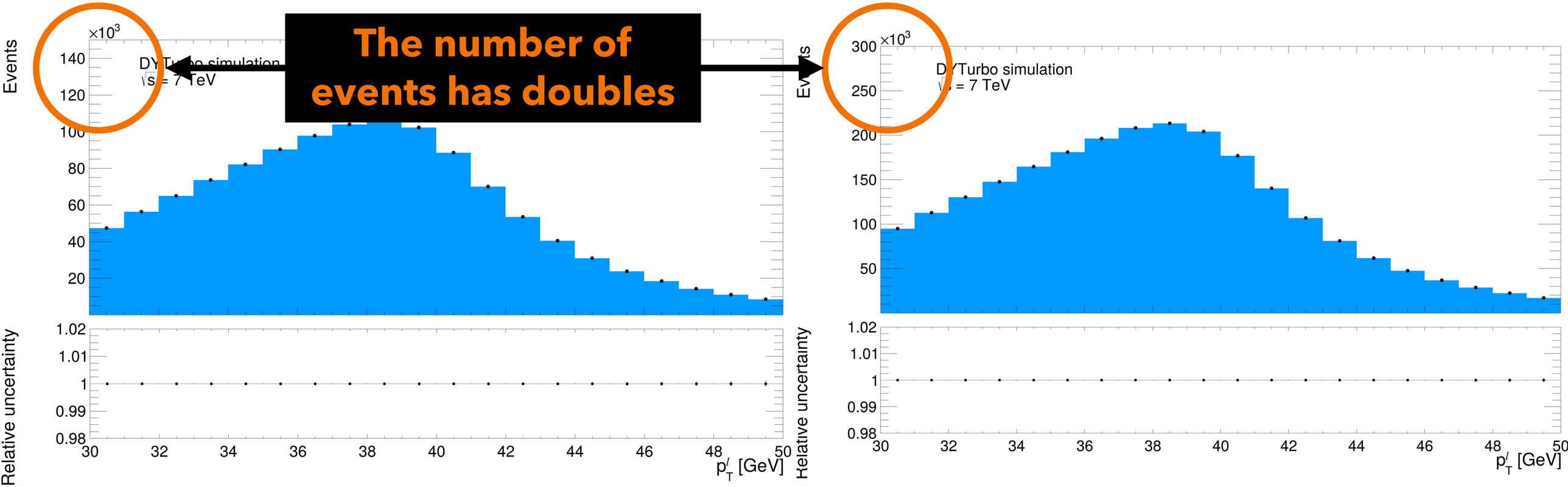
After the modification:



Spectra in pT lepton

Before the modification:

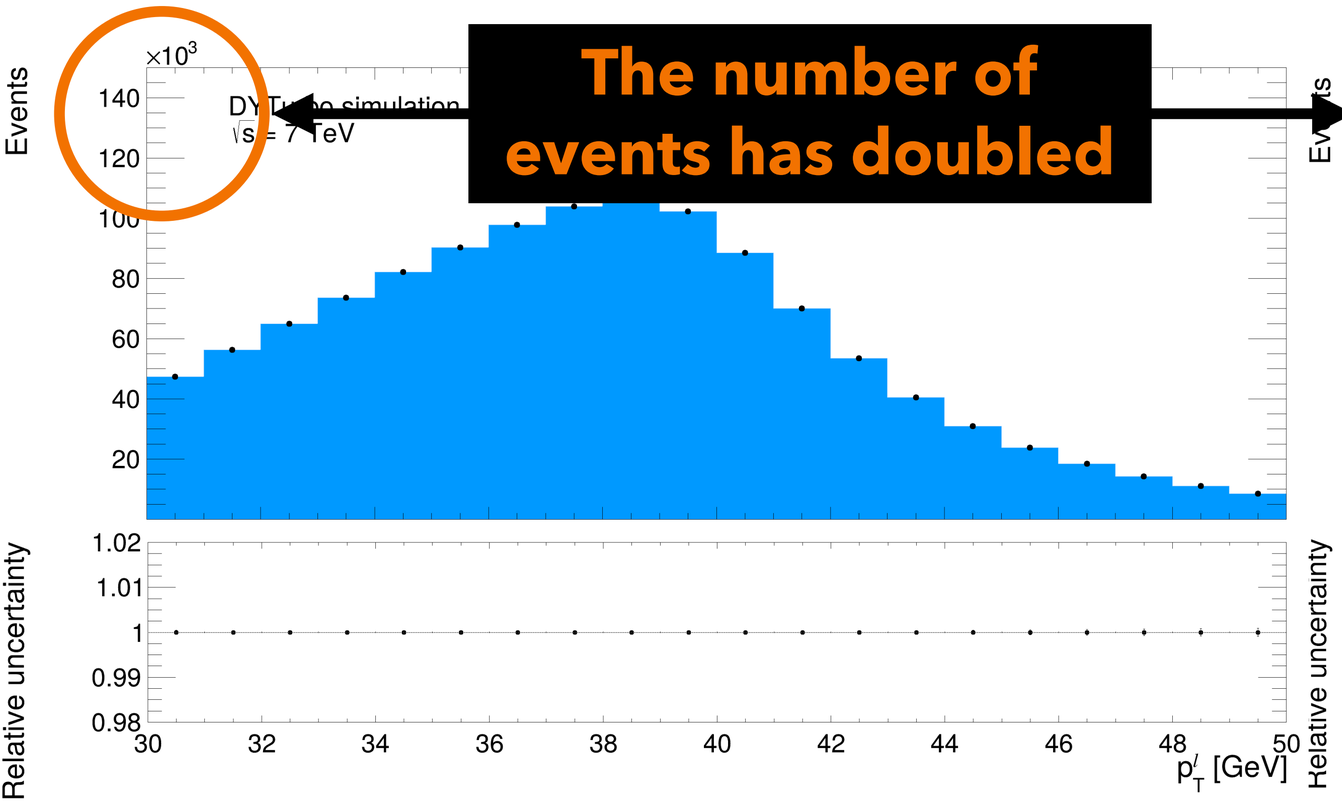
After the modification:



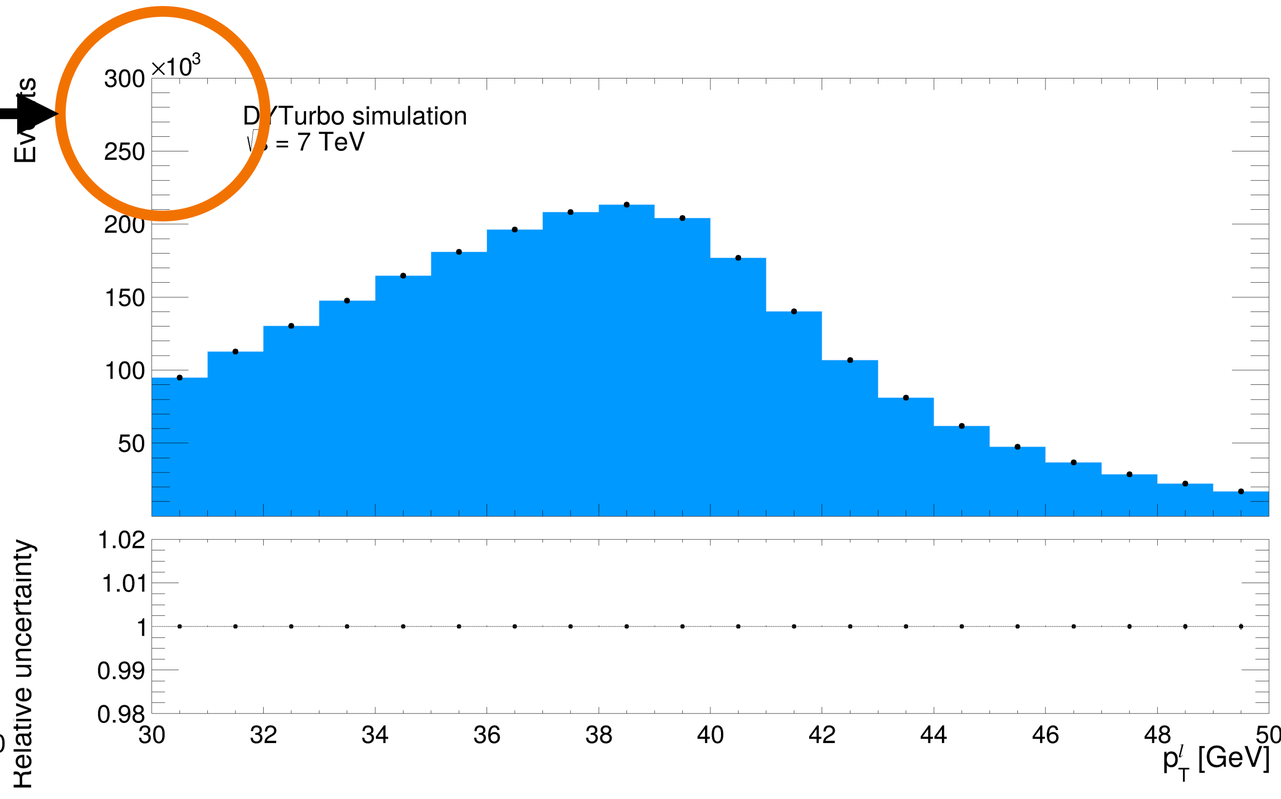
Spectra in pT lepton

Before the modification:

After the modification:



The number of events has doubled



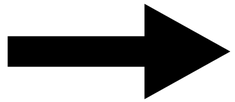
Why is it happening? May it be related to the previous problems?

New definition of the parameters

Modifications in Cuba integration

The Vegas parameters have been redefined in src/settings.h:

```
int vegasncallsBORN  ;
int vegasncallsCT   ;
int vegasncallsVJLO   ;
long long int vegasncallsVJREAL ;
int vegasncallsVJVIRT ;
```



```
int vegasncallsBORN  ;
long long int vegasncallsCT   ;
int vegasncallsVJLO   ;
long long int vegasncallsVJREAL ;
long long int vegasncallsVJVIRT ;
```

And consequently in src/cubacall.C.

If we print mineval and maxeval in the **output**, they are **not negative**.

Output of the first test

The output in q_T does not seem problematic for the pasudodata (with the same statistics as the first slides):

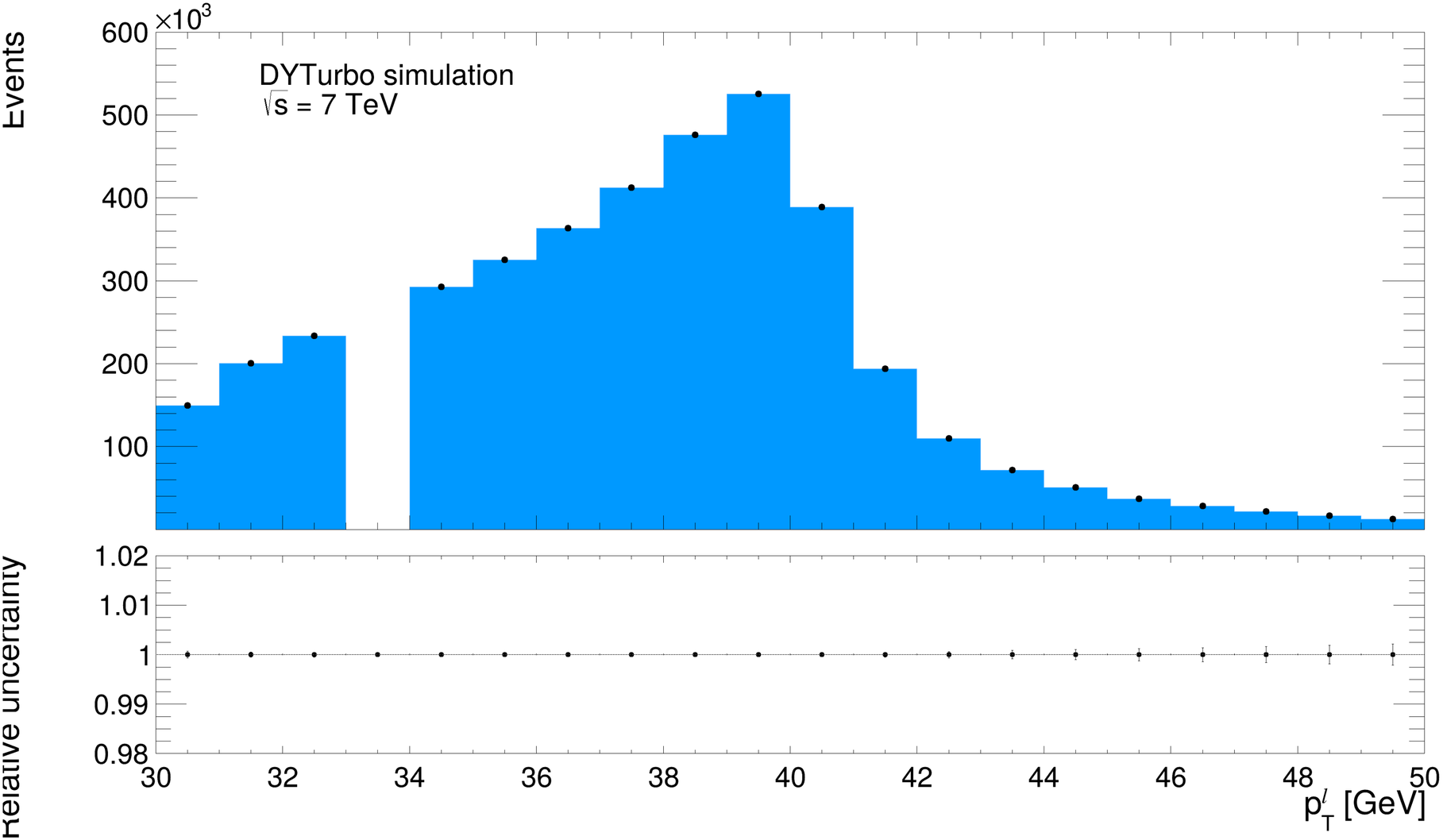
Resummation	Counter term	V+J Real	V+J Virtual	TOTAL
1.20664 ± 0.00015 ·10⁺⁰⁶ (41.7hr)	−1.77466 ± 8.1e−05 ·10⁺⁰⁶ (31.5mi)	−2.957 ± 0.029 ·10⁺⁰⁶ (7.29hr)	4.77543 ± 0.00026 ·10⁺⁰⁶ (28.1mi)	1.25 ± 0.029 ·10⁺⁰⁶ (49.9hr)
1.20664 ± 0.00015 ·10⁺⁰⁶ (41.7hr)	−1.77466 ± 8.1e−05 ·10⁺⁰⁶ (31.5mi)	−2.957 ± 0.029 ·10⁺⁰⁶ (7.29hr)	4.77543 ± 0.00026 ·10⁺⁰⁶ (28.1mi)	1.25 ± 0.029 ·10⁺⁰⁶ (49.9hr)

Summary

Total cross section	1.25 ± 0.029	nb
Calculation time	49.9	hours
Output file	Replica0−MAP24−FI.root	
Result in text file	Replica0−MAP24−FI.txt	

Anyways, we can see that for each column the errors are very different.

Spectrum in pT lepton

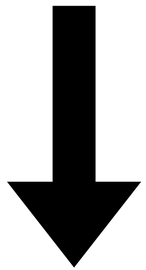


The spectrum is deformed and a bin is -nan.

Second test

Modifying the statistics in order to get more similar errors:

```
 Resummation:          vegas       ncalls =    60000000
Counter term:       vegas 8D      ncalls =  2400000000
    V+J Real:          vegas      ncalls = 24000000000
 V+J Virtual:          vegas      ncalls =  2400000000
```



```
 Resummation:          vegas       ncalls =    10000000
Counter term:       vegas 8D      ncalls =   100000000
    V+J Real:          vegas      ncalls =1000000000000
 V+J Virtual:          vegas      ncalls =   100000000
```

Output of the second test

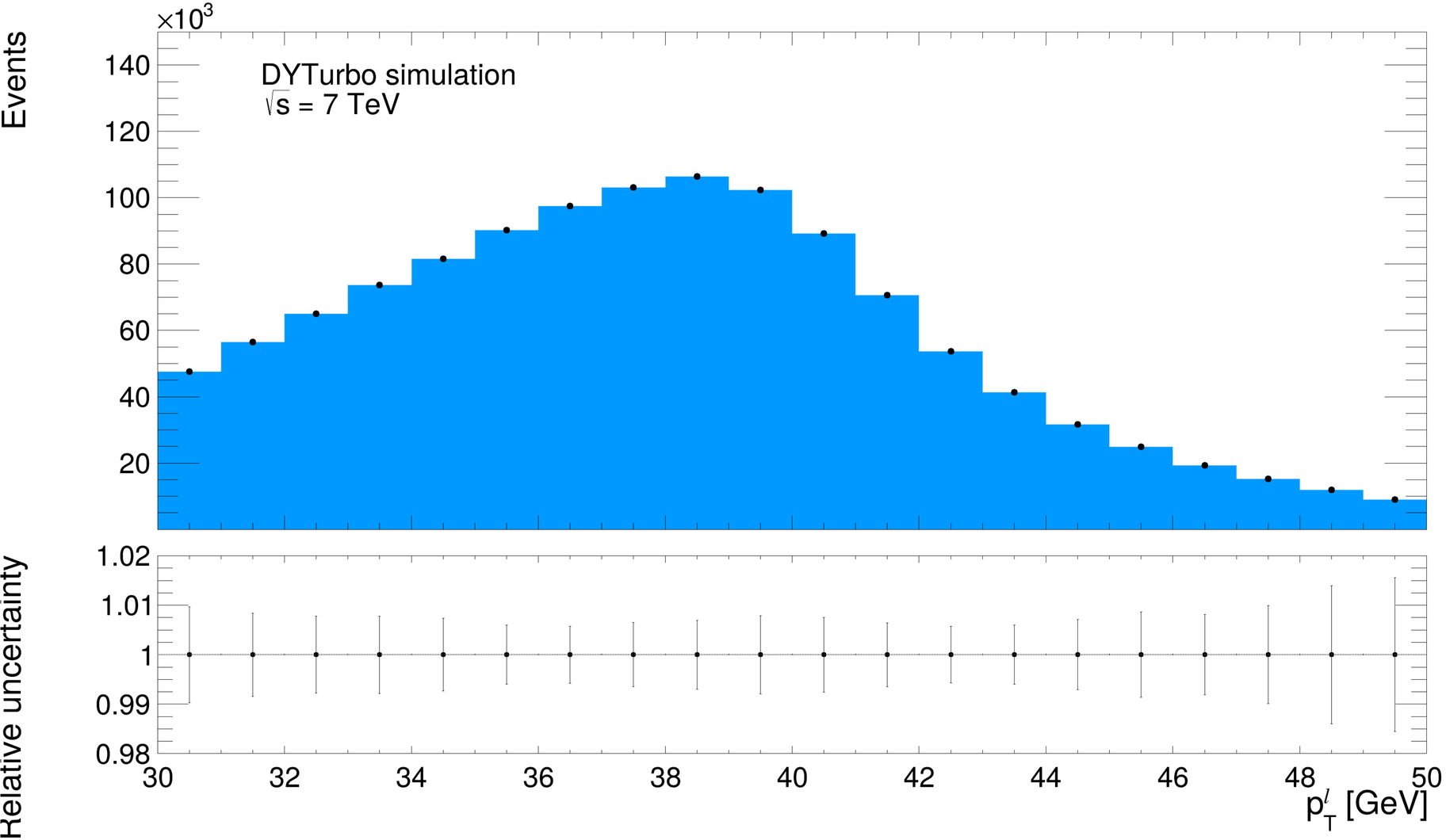
Resummation	Counter term	V+J Real	V+J Virtual	TOTAL
1.20619 ± 0.00036 ·10⁺⁰⁶ (5.38hr)	−1.77488 ± 0.00039 ·10⁺⁰⁶ (3.28mi)	−2.99649 ± 0.00024 ·10⁺⁰⁶ (24.8hr)	4.7779 ± 0.0013 ·10⁺⁰⁶ (1.32mi)	1.2127 ± 0.0014 ·10⁺⁰⁶ (30.2hr)
1.20619 ± 0.00036 ·10⁺⁰⁶ (5.38hr)	−1.77488 ± 0.00039 ·10⁺⁰⁶ (3.28mi)	−2.99649 ± 0.00024 ·10⁺⁰⁶ (24.8hr)	4.7779 ± 0.0013 ·10⁺⁰⁶ (1.32mi)	1.2127 ± 0.0014 ·10⁺⁰⁶ (30.2hr)

Summary

Total cross section	1.2127 ± 0.0014	nb
Calculation time	30.2	hours
Output file	Replica0-goodspectrum.root	
Result in text file	Replica0-goodspectrum.txt	

VJ VIRT error is still larger, but they are definitely more consistent.

Spectrum in pT lepton

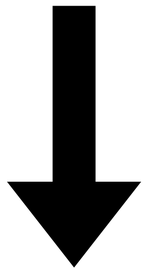


The spectrum is smooth, but the errors are quite large for our purposes.

Third test

We have tried to modify again the statistics in order to get more similar errors and more precise results:

```
Resummation:        vegas       ncalls =    60000000
Counter term:     vegas 8D      ncalls =  2400000000
   V+J Real:        vegas       ncalls = 24000000000
V+J Virtual:        vegas       ncalls =  2400000000
```



```
Resummation:        vegas       ncalls =    20000000
Counter term:     vegas 8D      ncalls =   200000000
   V+J Real:        vegas       ncalls = 50000000000
V+J Virtual:        vegas       ncalls =  2000000000
```

Third test

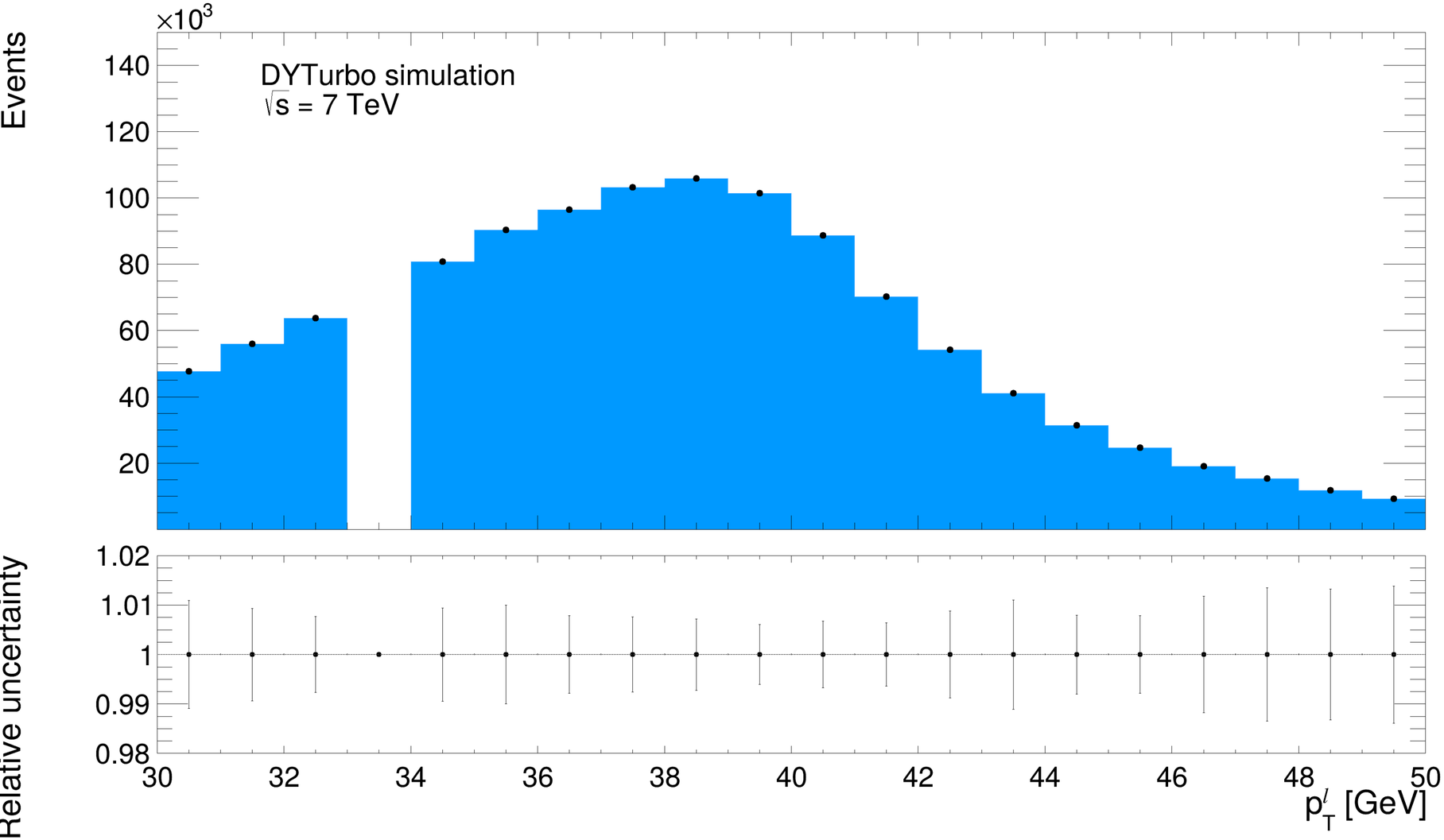
Resummation	Counter term	V+J Real	V+J Virtual	TOTAL
1.20642 ± 0.00026 ·10⁺⁰⁶ (13.5hr)	−1.77536 ± 0.00028 ·10⁺⁰⁶ (3.27mi)	−2.99696 ± 0.00035 ·10⁺⁰⁶ (17.2hr)	4.77572 ± 0.00029 ·10⁺⁰⁶ (24.3mi)	1.20982 ± 0.00059 ·10⁺⁰⁶ (31.2hr)
1.20642 ± 0.00026 ·10⁺⁰⁶ (13.5hr)	−1.77536 ± 0.00028 ·10⁺⁰⁶ (3.27mi)	−2.99696 ± 0.00035 ·10⁺⁰⁶ (17.2hr)	4.77572 ± 0.00029 ·10⁺⁰⁶ (24.3mi)	1.20982 ± 0.00059 ·10⁺⁰⁶ (31.2hr)

Summary

Total cross section	1.20982 ± 0.00059	nb
Calculation time	31.2	hours
Output file	Replica0-laststat.root	
Result in text file	Replica0-laststat.txt	

The errors are small and very similar.

Spectrum in pT lepton



The spectrum has again a problematic bin.

Observations about the printed Vegas parameters

Print of the Vegas parameters

If we print the values for each term, they are positive, mineval and maxeval do not seem problematic anymore but something weird happened for the **VJREAL parameters**:

mineval	**=**	**100000000000**
maxeval	=	100000000000
nstart	=	1410065408
nincrease	=	1410065408
nbatch	=	1000
seed	=	123456

Print of the Vegas parameters

If we print the values for each term, they are positive, mineval and maxeval do not seem problematic anymore but something weird happened for the **VJREAL parameters**:

mineval	=	**100000000000**
maxeval	=	100000000000
nstart	=	1410065408
nincrease	=	1410065408
nbatch	=	1000
seed	=	123456

Definition of these two:

```
const int nstart = max(10, int(opts.vegasncallsVJREAL/cubacall::nst));
const int nincrease = max(10, int(opts.vegasncallsVJREAL/10));
```

Casting of Vegas parameters

They are normally ten times smaller than mineval and maxeval; what is happening here?

```
const int nstart = max(10, int(opts.vegasncallsVJREAL/cubacall::nst));
const int nincrease = max(10, int(opts.vegasncallsVJREAL/10));
```

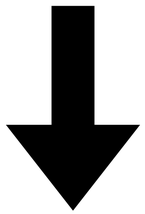
The function int() gives an int in output; so if the result of the ratio is greater than 2x10^9, the number is truncated (in hexadecimal, se we got those weird numbers).

Can this aspect be the responsible of the problematic bin?

Further test

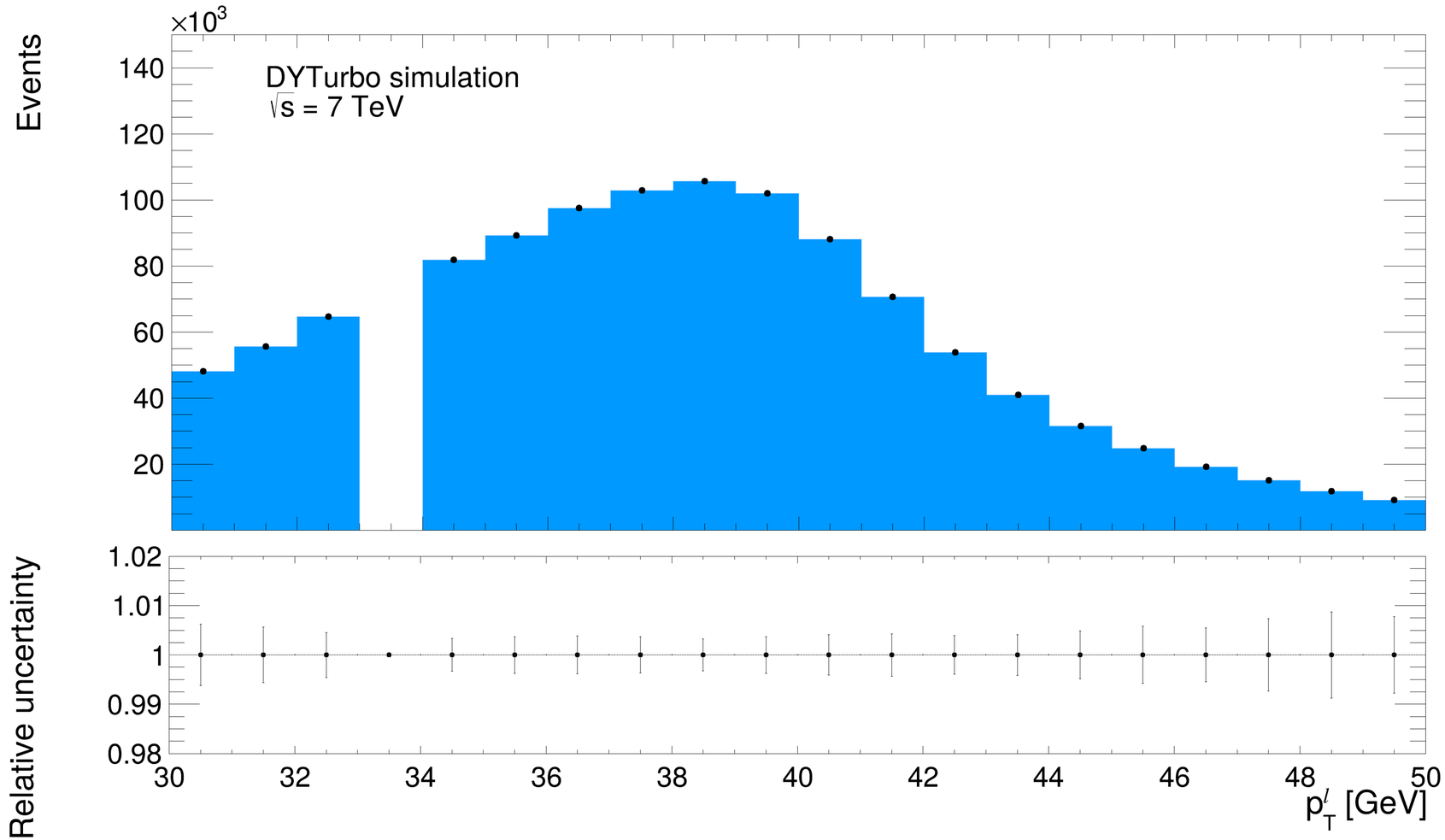
We have done a test choosing a statistics under the problematic values, in order not to encounter the casting of the Vegas variables.

```
 Resummation:          vegas       ncalls =    20000000
Counter term:       vegas 8D       ncalls =   200000000
    V+J Real:          vegas       ncalls = 50000000000
 V+J Virtual:          vegas       ncalls =  2000000000
```



```
 Resummation:          vegas       ncalls =    20000000
Counter term:       vegas 8D       ncalls =  2000000000
    V+J Real:          vegas       ncalls = 20000000000
 V+J Virtual:          vegas       ncalls =  2000000000
```

Spectrum in pT lepton



But the bin is pathological.

Isolation of each term

Responsible of -nan

We have decided to **isolate each term** (resumption, counter term, vj real and vj virtual) in order to **find the responsible of the problematic bin**.

Given that in q_T we did not see some clues, we have extracted the p_T lepton.

Resummed term:

Bin	Center	Width	Value	Error
1	30.50	1.00	4.754004e+04	1.191944e+02
2	31.50	1.00	5.639281e+04	1.232689e+02
3	32.50	1.00	6.501858e+04	1.263705e+02
4	33.50	1.00	7.358935e+04	1.296550e+02
5	34.50	1.00	8.221928e+04	1.335538e+02
6	35.50	1.00	9.022450e+04	1.361567e+02
7	36.50	1.00	9.789100e+04	1.389550e+02
8	37.50	1.00	1.037500e+05	1.398222e+02
9	38.50	1.00	1.064542e+05	1.389628e+02
10	39.50	1.00	1.022644e+05	1.338169e+02
11	40.50	1.00	8.836223e+04	1.250481e+02
12	41.50	1.00	7.006876e+04	1.134841e+02
13	42.50	1.00	5.344708e+04	1.007649e+02
14	43.50	1.00	4.054300e+04	8.904659e+01
15	44.50	1.00	3.079177e+04	7.798286e+01
16	45.50	1.00	2.379600e+04	6.905273e+01
17	46.50	1.00	1.844572e+04	6.126290e+01
18	47.50	1.00	1.429451e+04	5.435181e+01
19	48.50	1.00	1.110844e+04	4.806435e+01
20	49.50	1.00	8.507254e+03	4.231699e+01

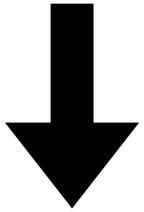
Counter term:

Bin	Center	Width	Value	Error
1	30.50	1.00	-6.342278e+04	2.615665e+01
2	31.50	1.00	-7.846445e+04	2.689057e+01
3	32.50	1.00	-9.385206e+04	2.734053e+01
4	33.50	1.00	-1.081263e+05	2.800397e+01
5	34.50	1.00	-1.220358e+05	2.878834e+01
6	35.50	1.00	-1.364008e+05	2.986658e+01
7	36.50	1.00	-1.514295e+05	3.105303e+01
8	37.50	1.00	-1.668534e+05	3.246603e+01
9	38.50	1.00	-1.792647e+05	3.397535e+01
10	39.50	1.00	-1.735926e+05	3.451078e+01
11	40.50	1.00	-1.454812e+05	3.045961e+01
12	41.50	1.00	-1.042047e+05	2.360907e+01
13	42.50	1.00	-6.919060e+04	1.866915e+01
14	43.50	1.00	-4.781483e+04	1.538597e+01
15	44.50	1.00	-3.435113e+04	1.307734e+01
16	45.50	1.00	-2.535780e+04	1.130587e+01
17	46.50	1.00	-1.909228e+04	9.786171e+00
18	47.50	1.00	-1.450045e+04	8.595429e+00
19	48.50	1.00	-1.107189e+04	7.497442e+00
20	49.50	1.00	-8.435641e+03	6.551726e+00

VJ Real:

Bin	Center	Width	Value	Error
1	30.50	1.00	-1.014099e+05	2.636891e+02
2	31.50	1.00	-1.449103e+05	2.736172e+02
3	32.50	1.00	-1.688184e+05	2.522159e+02
4	33.50	1.00	-1.901624e+05	2.063142e+02
5	34.50	1.00	-2.110571e+05	2.200548e+02
6	35.50	1.00	-2.357733e+05	2.865487e+02
7	36.50	1.00	-2.662259e+05	3.293531e+02
8	37.50	1.00	-3.091876e+05	3.337082e+02
9	38.50	1.00	-3.706764e+05	2.903358e+02
10	39.50	1.00	-4.238629e+05	3.291891e+02
11	40.50	1.00	-3.006147e+05	3.230633e+02
12	41.50	1.00	-1.231144e+05	2.738679e+02
13	42.50	1.00	-5.587504e+04	1.803748e+02
14	43.50	1.00	-3.090803e+04	1.402424e+02
15	44.50	1.00	-1.896401e+04	1.299303e+02
16	45.50	1.00	-1.248954e+04	1.240509e+02
17	46.50	1.00	-8.871217e+03	8.209452e+01
18	47.50	1.00	-6.355819e+03	9.513557e+01
19	48.50	1.00	-4.698550e+03	9.091445e+01
20	49.50	1.00	-3.628709e+03	5.561724e+01

VJ Real:



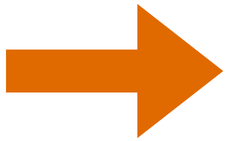
The pathological bin does not seem related to the casting of the VJREAL variables.

Bin	Center	Width	Value	Error
1	30.50	1.00	-1.014099e+05	2.636891e+02
2	31.50	1.00	-1.449103e+05	2.736172e+02
3	32.50	1.00	-1.688184e+05	2.522159e+02
4	33.50	1.00	-1.901624e+05	2.063142e+02
5	34.50	1.00	-2.110571e+05	2.200548e+02
6	35.50	1.00	-2.357733e+05	2.865487e+02
7	36.50	1.00	-2.662259e+05	3.293531e+02
8	37.50	1.00	-3.091876e+05	3.337082e+02
9	38.50	1.00	-3.706764e+05	2.903358e+02
10	39.50	1.00	-4.238629e+05	3.291891e+02
11	40.50	1.00	-3.006147e+05	3.230633e+02
12	41.50	1.00	-1.231144e+05	2.738679e+02
13	42.50	1.00	-5.587504e+04	1.803748e+02
14	43.50	1.00	-3.090803e+04	1.402424e+02
15	44.50	1.00	-1.896401e+04	1.299303e+02
16	45.50	1.00	-1.248954e+04	1.240509e+02
17	46.50	1.00	-8.871217e+03	8.209452e+01
18	47.50	1.00	-6.355819e+03	9.513557e+01
19	48.50	1.00	-4.698550e+03	9.091445e+01
20	49.50	1.00	-3.628709e+03	5.561724e+01

VJ Virt:

Bin	Center	Width	Value	Error
1	30.50	1.00	1.653974e+05	7.481588e+01
2	31.50	1.00	2.225783e+05	8.206659e+01
3	32.50	1.00	2.623818e+05	8.350769e+01
4	33.50	1.00	-nan	-nan
5	34.50	1.00	3.328119e+05	8.883441e+01
6	35.50	1.00	3.712357e+05	9.268194e+01
7	36.50	1.00	4.172593e+05	9.789543e+01
8	37.50	1.00	4.751318e+05	1.049011e+02
9	38.50	1.00	5.491524e+05	1.147289e+02
10	39.50	1.00	5.971733e+05	1.234105e+02
11	40.50	1.00	4.458557e+05	1.029854e+02
12	41.50	1.00	2.278790e+05	6.486472e+01
13	42.50	1.00	1.254988e+05	4.357136e+01
14	43.50	1.00	7.915109e+04	3.243986e+01
15	44.50	1.00	5.407885e+04	2.566630e+01
16	45.50	1.00	3.883892e+04	2.110193e+01
17	46.50	1.00	2.872051e+04	1.772865e+01
18	47.50	1.00	2.169152e+04	1.515503e+01
19	48.50	1.00	1.655309e+04	1.315684e+01
20	49.50	1.00	1.268787e+04	1.143176e+01

VJ Virt:



How to interpret this behaviour and cure it?

Bin	Center	Width	Value	Error
1	30.50	1.00	1.653974e+05	7.481588e+01
2	31.50	1.00	2.225783e+05	8.206659e+01
3	32.50	1.00	2.623818e+05	8.350769e+01
4	33.50	1.00	-nan	-nan
5	34.50	1.00	3.328119e+05	8.883441e+01
6	35.50	1.00	3.712357e+05	9.268194e+01
7	36.50	1.00	4.172593e+05	9.789543e+01
8	37.50	1.00	4.751318e+05	1.049011e+02
9	38.50	1.00	5.491524e+05	1.147289e+02
10	39.50	1.00	5.971733e+05	1.234105e+02
11	40.50	1.00	4.458557e+05	1.029854e+02
12	41.50	1.00	2.278790e+05	6.486472e+01
13	42.50	1.00	1.254988e+05	4.357136e+01
14	43.50	1.00	7.915109e+04	3.243986e+01
15	44.50	1.00	5.407885e+04	2.566630e+01
16	45.50	1.00	3.883892e+04	2.110193e+01
17	46.50	1.00	2.872051e+04	1.772865e+01
18	47.50	1.00	2.169152e+04	1.515503e+01
19	48.50	1.00	1.655309e+04	1.315684e+01
20	49.50	1.00	1.268787e+04	1.143176e+01