UNIVERSITÄT HEIDELBERG
ZUKUNFT
SEIT 1386

Artificial Intelligence and the Uncertainty Challenge in Fundamental Physics
30.11.2023

# Uncertainty-Aware Diffusion Models

## for LHC event generation

SPONSORED BY THE
Federal Ministry
of Education
and Research


Anja Butter, Nathan Huetsch, Sofia Palacios Schweitzer, Tilman Plehn, Peter Sorrenson, Jonas Spinner


**arXiv: 2305.10475**



Anja Butter,Tomas Jezo, Michael Klasen, Mathias Kuschick, Sofia Palacios Schweitzer, Tilman Plehn

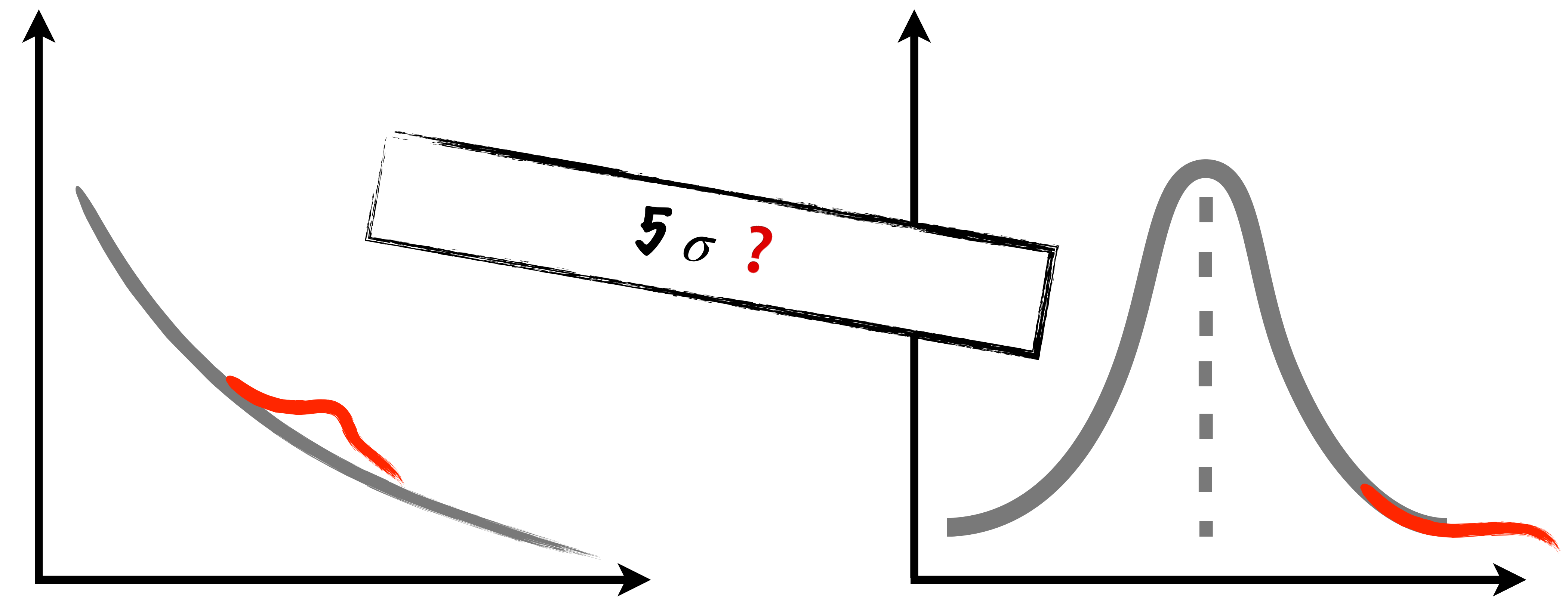

**arXiv: 2311.17175**

# Why Event Generation?

Vast amount of data collected by collider experiments

Standard Model is probed

Theoretical predictions (simulation) need to match experimental statistics



# Why ML Event Generation?

After high luminosity runs → ~ 20 times as much data

Theoretical predictions needs to be even more precise (include higher correction terms)

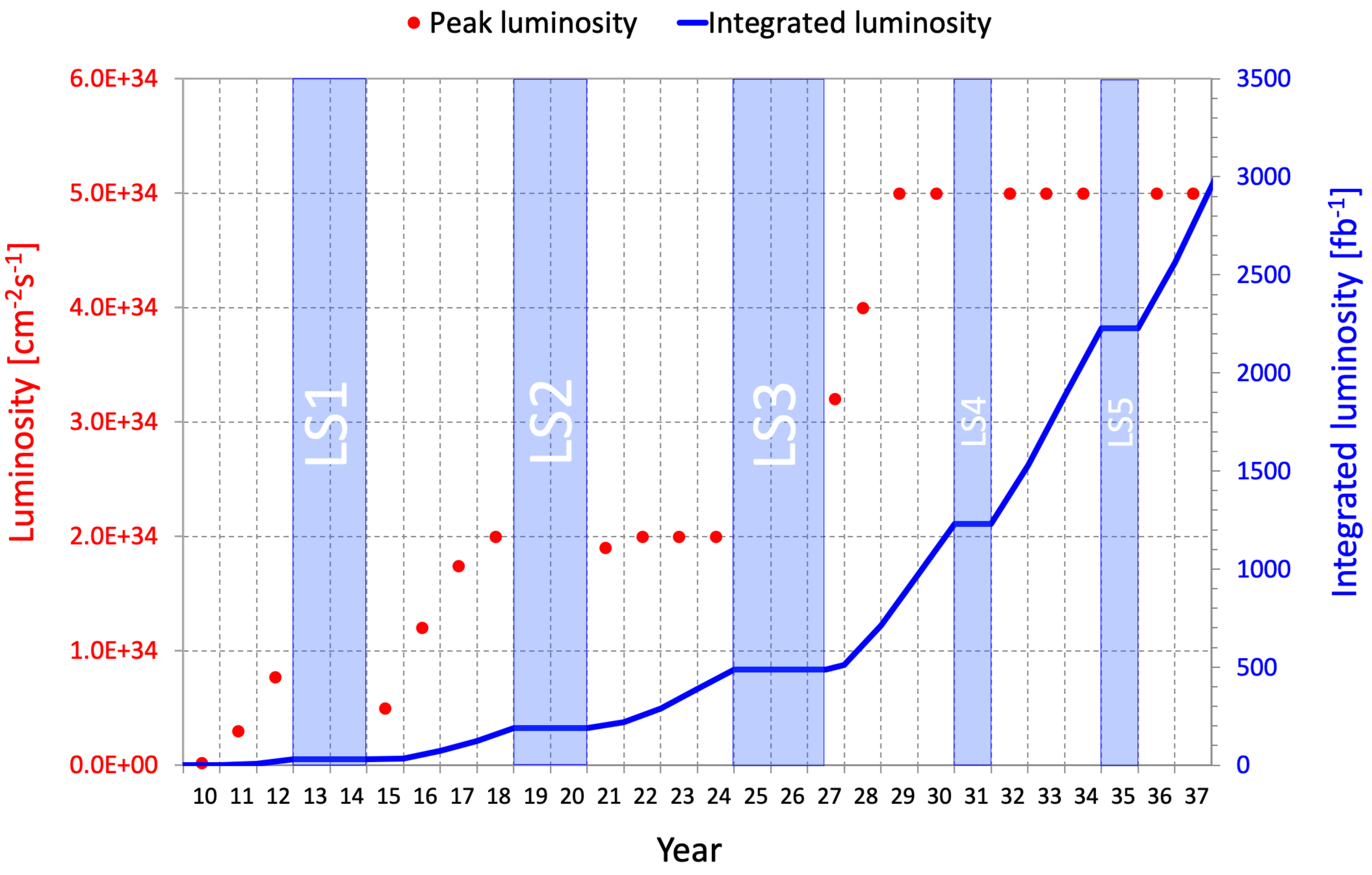



Figure from https://web.archive.org/web/20220706170326/https://lhc-commissioning.web.cern.ch/schedule/images/LHC-nominal-lumi-projection.png

# Why ML Event Generation?

After high luminosity runs → ~ 20 times as much data

Theoretical predictions needs to be even more precise (include higher correction terms)

**But:** Currently computationally expensive

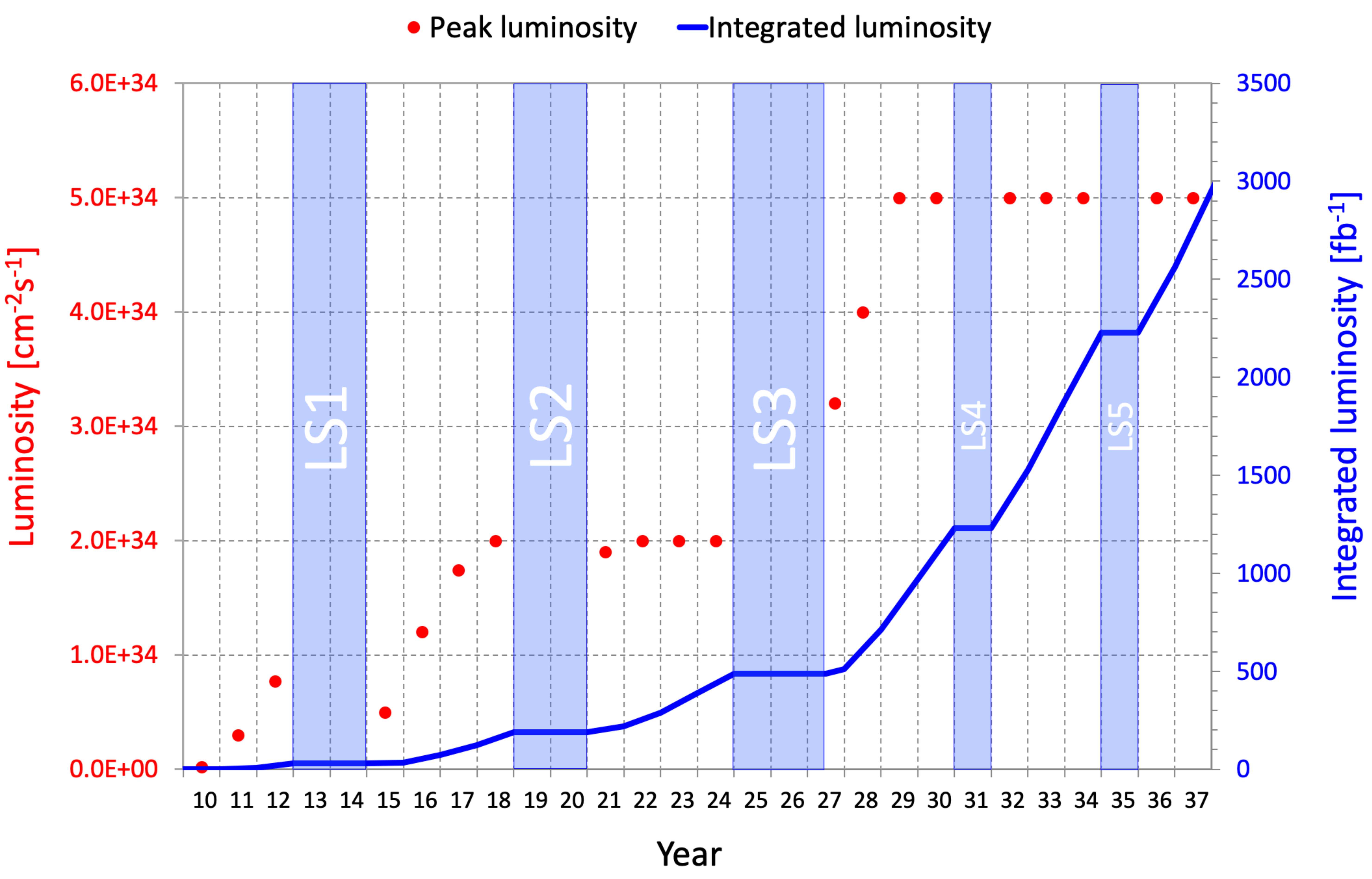



Figure from https://web.archive.org/web/20220706170326/https://lhc-commissioning.web.cern.ch/schedule/images/LHC-nominal-lumi-projection.png

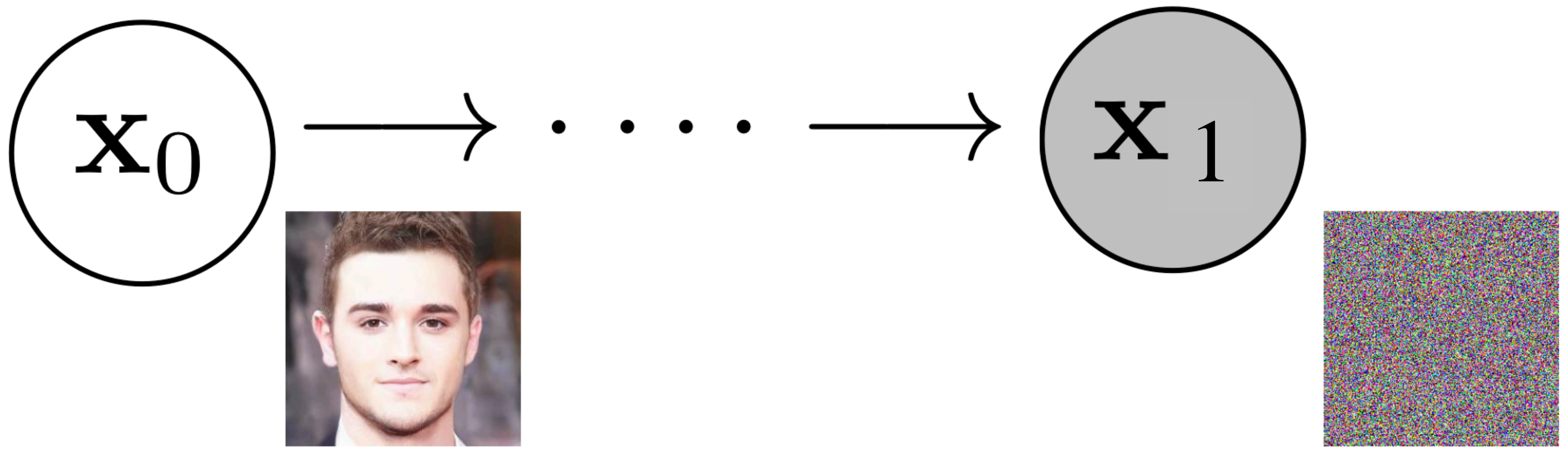
# Diffusion Models (CFM)

**Phase Space**

$x_0 \sim p_0(x_0)$

**Latent Space**

$x_1 \sim p_1(x_1)(= \mathcal{N}(0,1))$

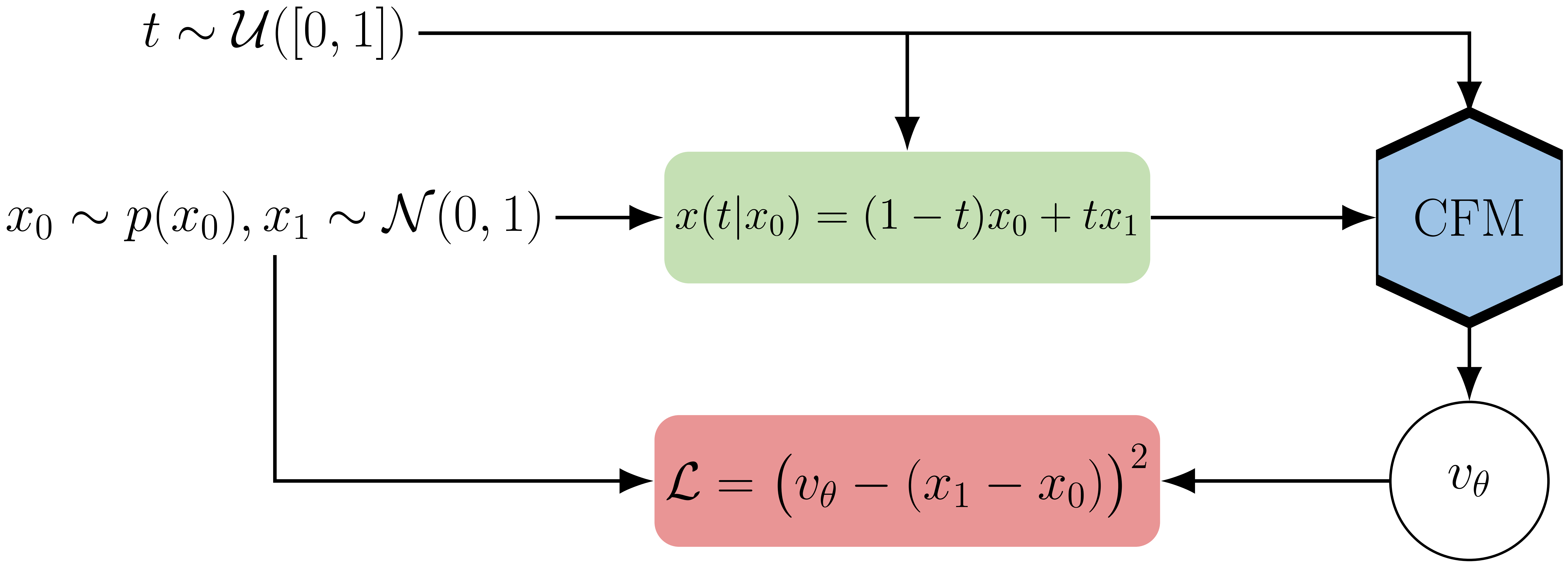



**Continuous time evolution** **t**

Evolution governed by $v \equiv \dfrac{dx}{dt}$

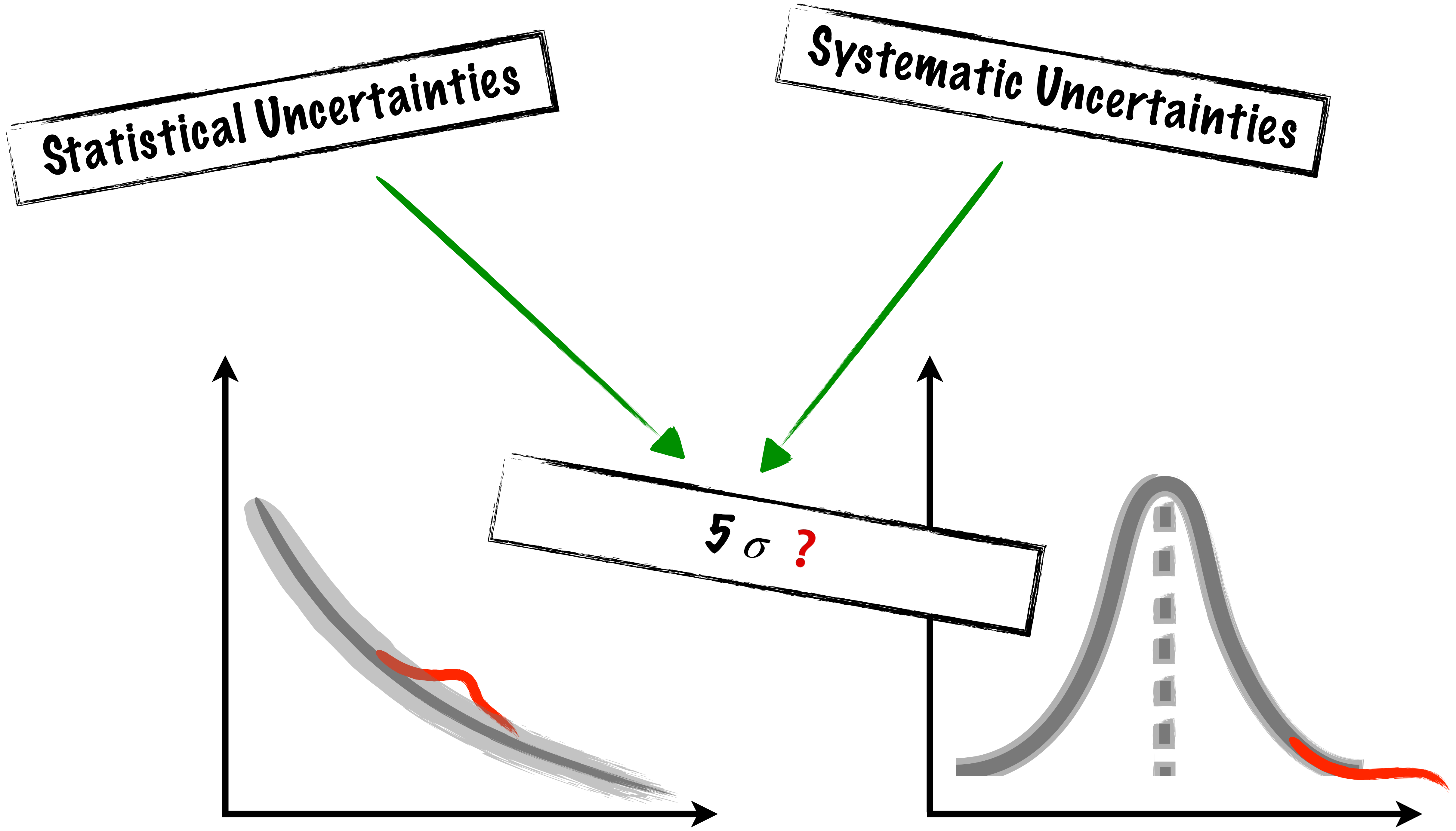
Figure from J.Ho et al.: arXiv:2006.11239

# Diffusion Models (CFM)



# Raising Awareness for Uncertainties

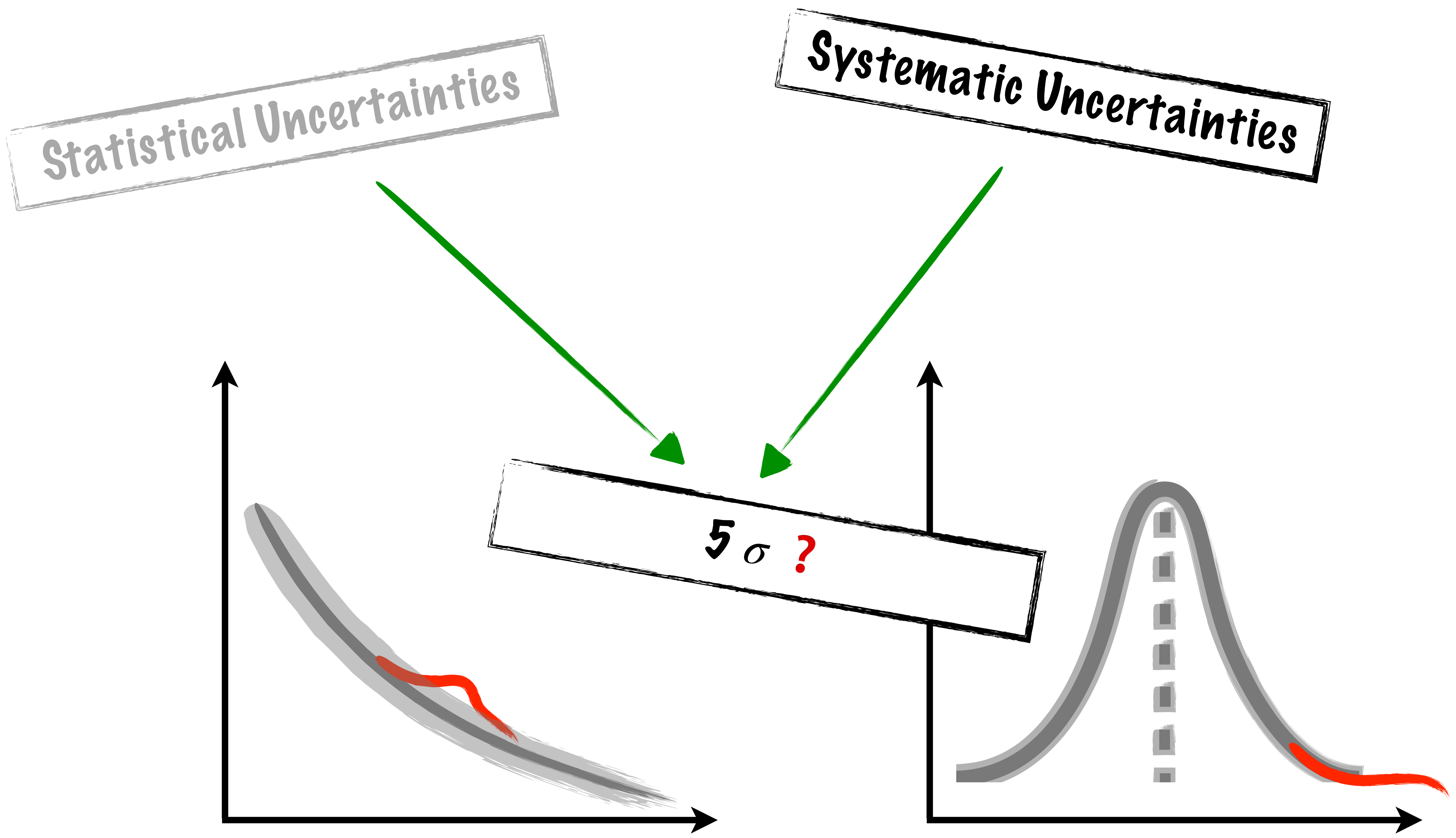
Being precise = estimating uncertainties



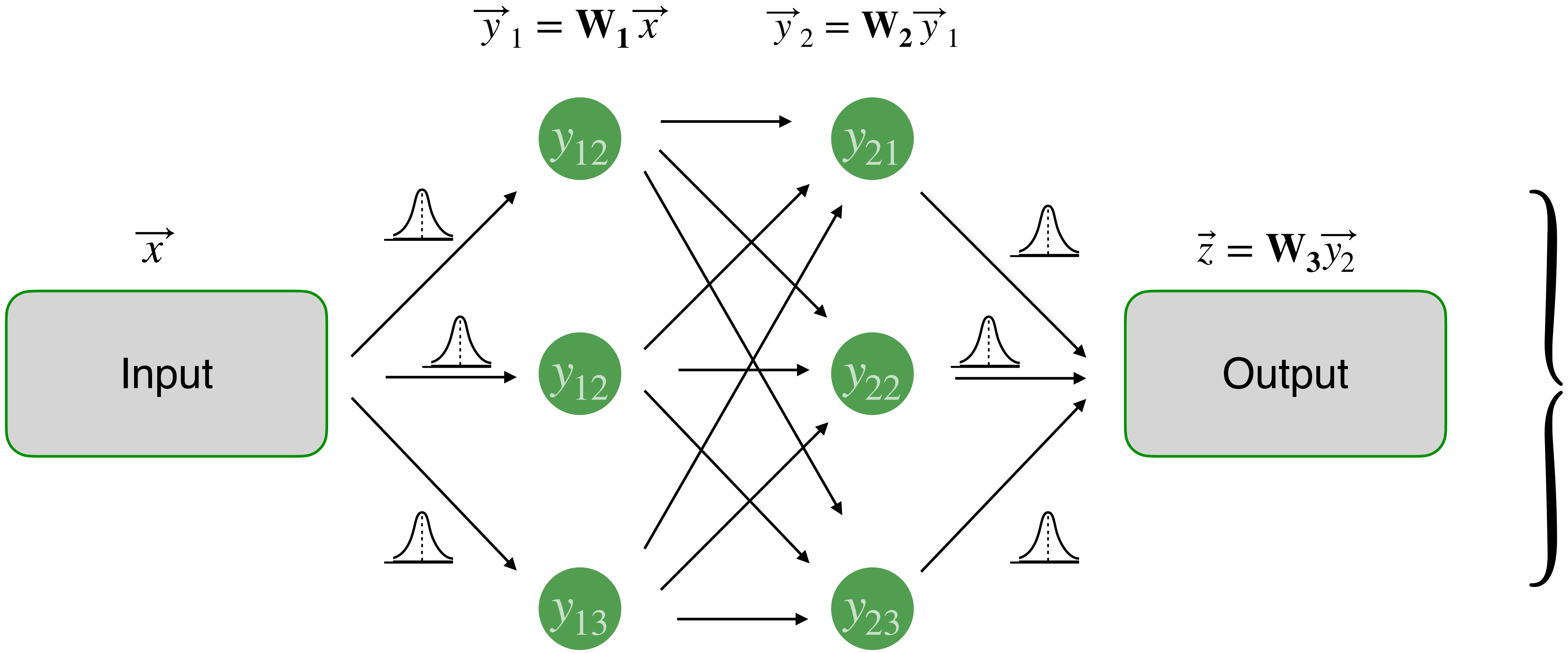
# Raising Awareness for Uncertainties

Being precise = estimating uncertainties

How can we account for network uncertainties?



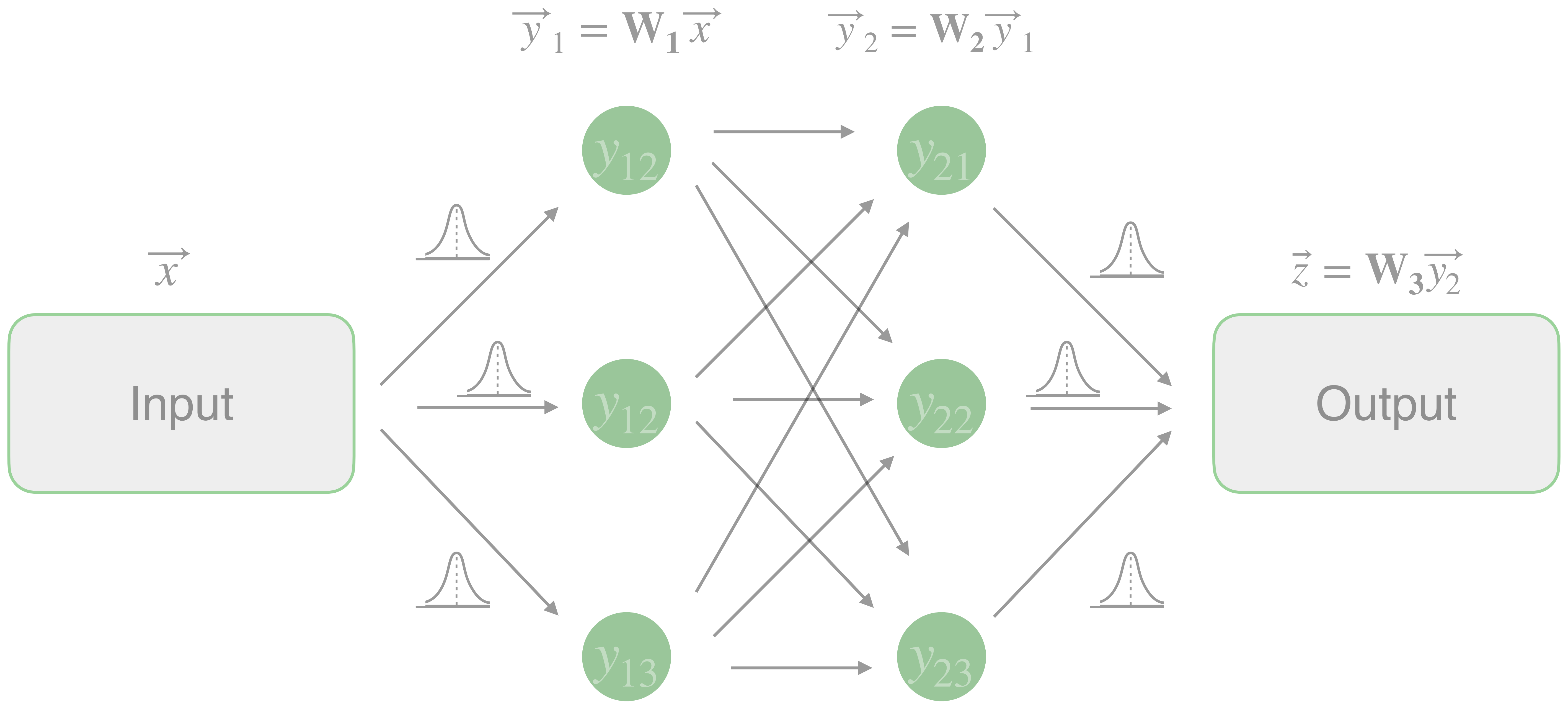
# What about uncertainties?




$$\langle \vec{z} \rangle = \frac{1}{N} \sum_i \vec{z}_i$$

$$\sigma^2_{pred} = \frac{1}{N} \sum_{i=1}^{N} \left( \langle \vec{z} \rangle - \vec{z}_i \right)^2$$

# How to Bayesianize - CFM

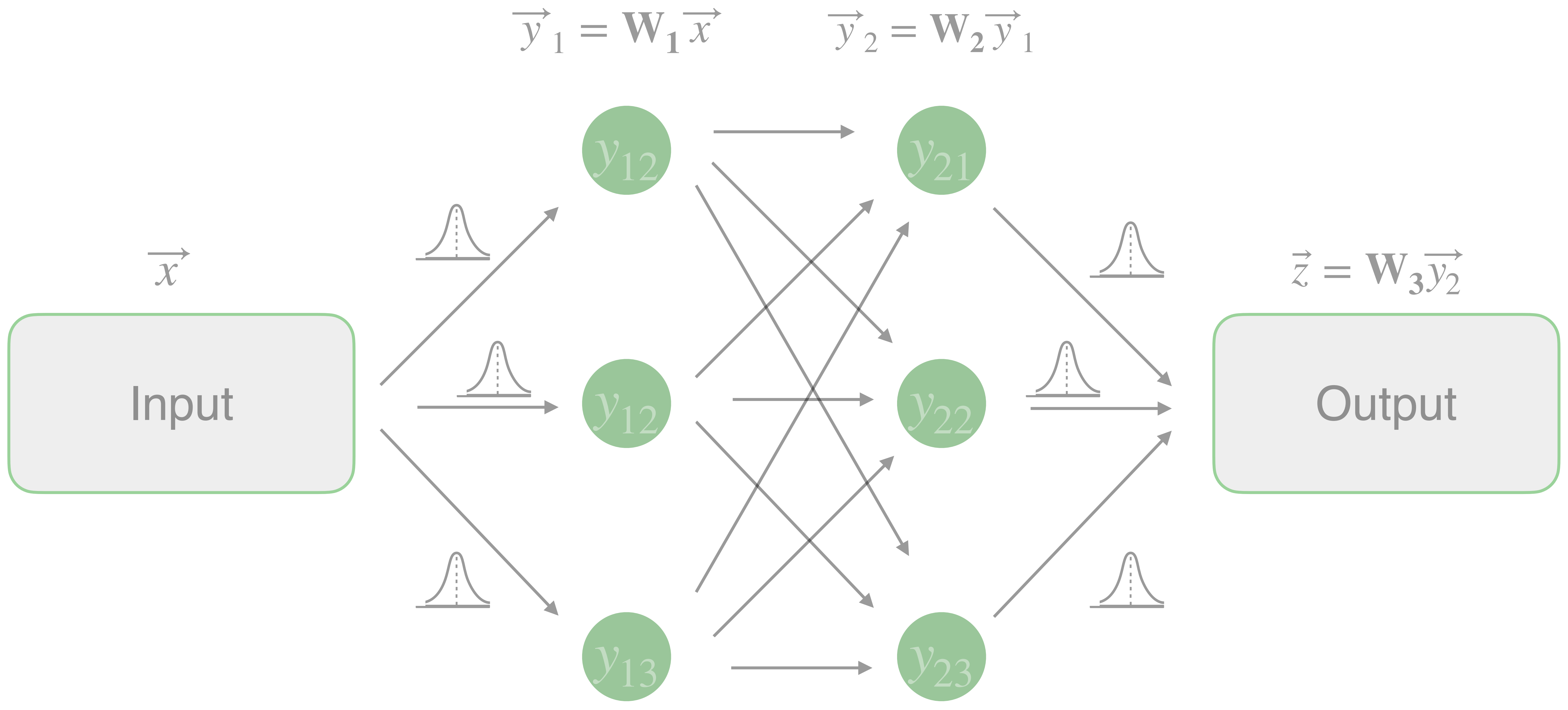



No Likelihood Loss

$$\boxed{\mathscr{L}_{CFM} = \left(v_\theta - (x_1 - x_0)\right)^2}$$

+ ? ? ?

# How to Bayesianize - CFM



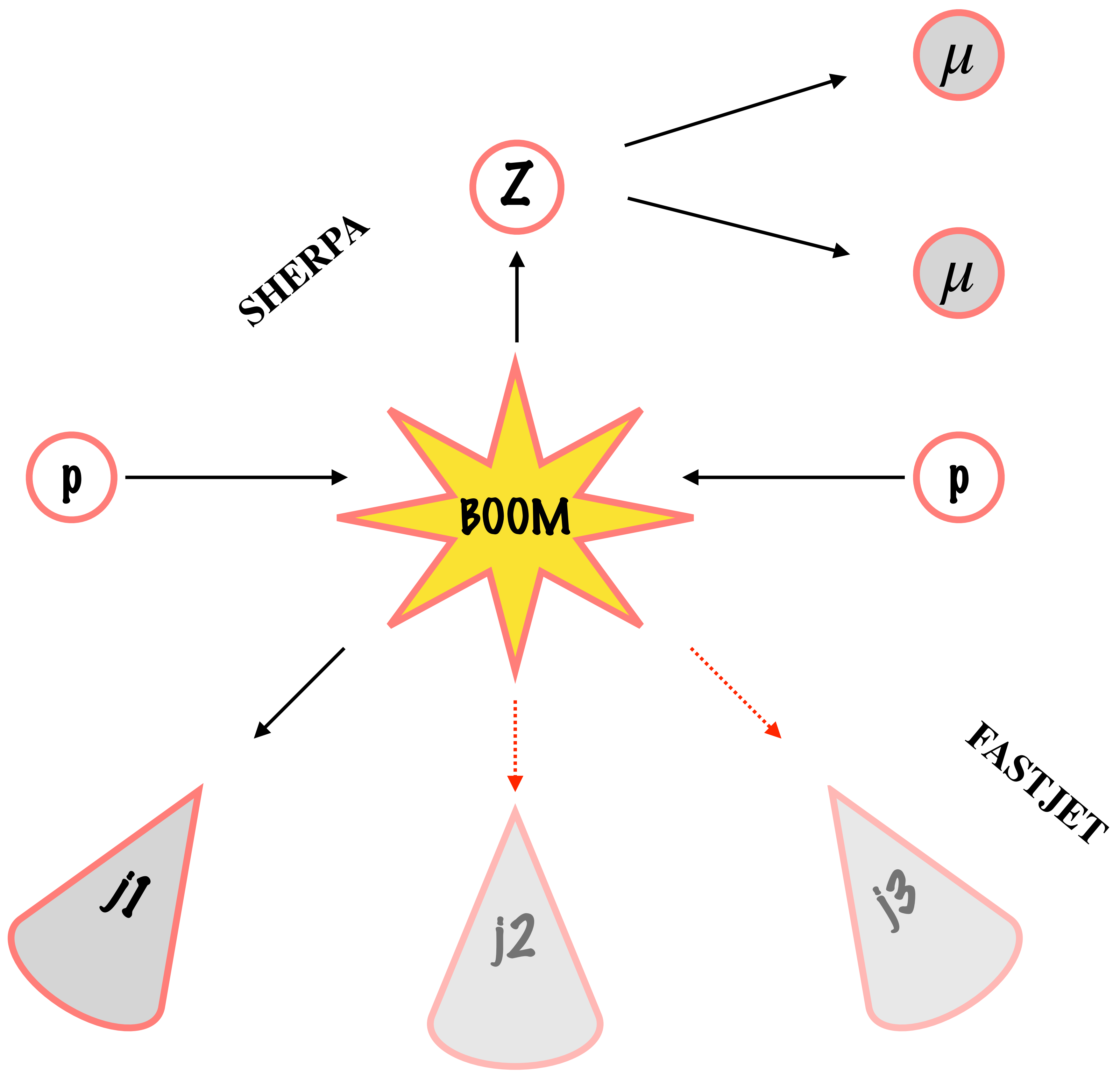

No Likelihood Loss

$$\mathscr{L}_{CFM} = \left\langle \left(v_\theta - (x_1 - x_0)\right)^2 \right\rangle_{\theta \sim q(\theta)}$$

Hyperparameter

$$+ \quad c \cdot \mathscr{D}_{KL}(q(\theta), p(\theta))$$

# Concrete Application — End-to-End LHC




<u>Z ( $\rightarrow \mu\mu$) + jets:</u>

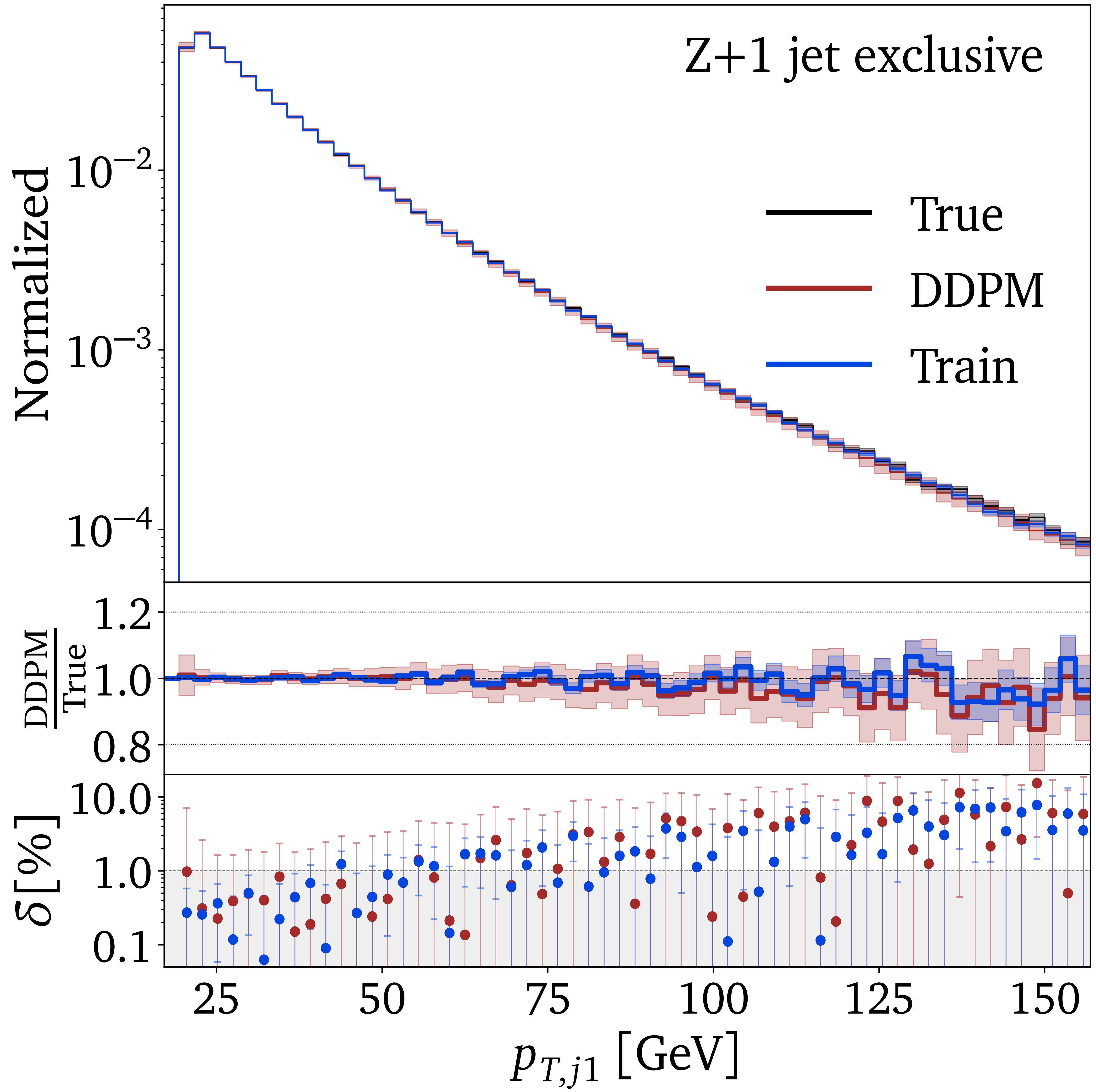
3 - 5 final state particles (including jets)

12 - 20 dimensional phase space

# To be precise

Percent level precision (comparable to statistical uncertainty)

Uncertainty well-defined

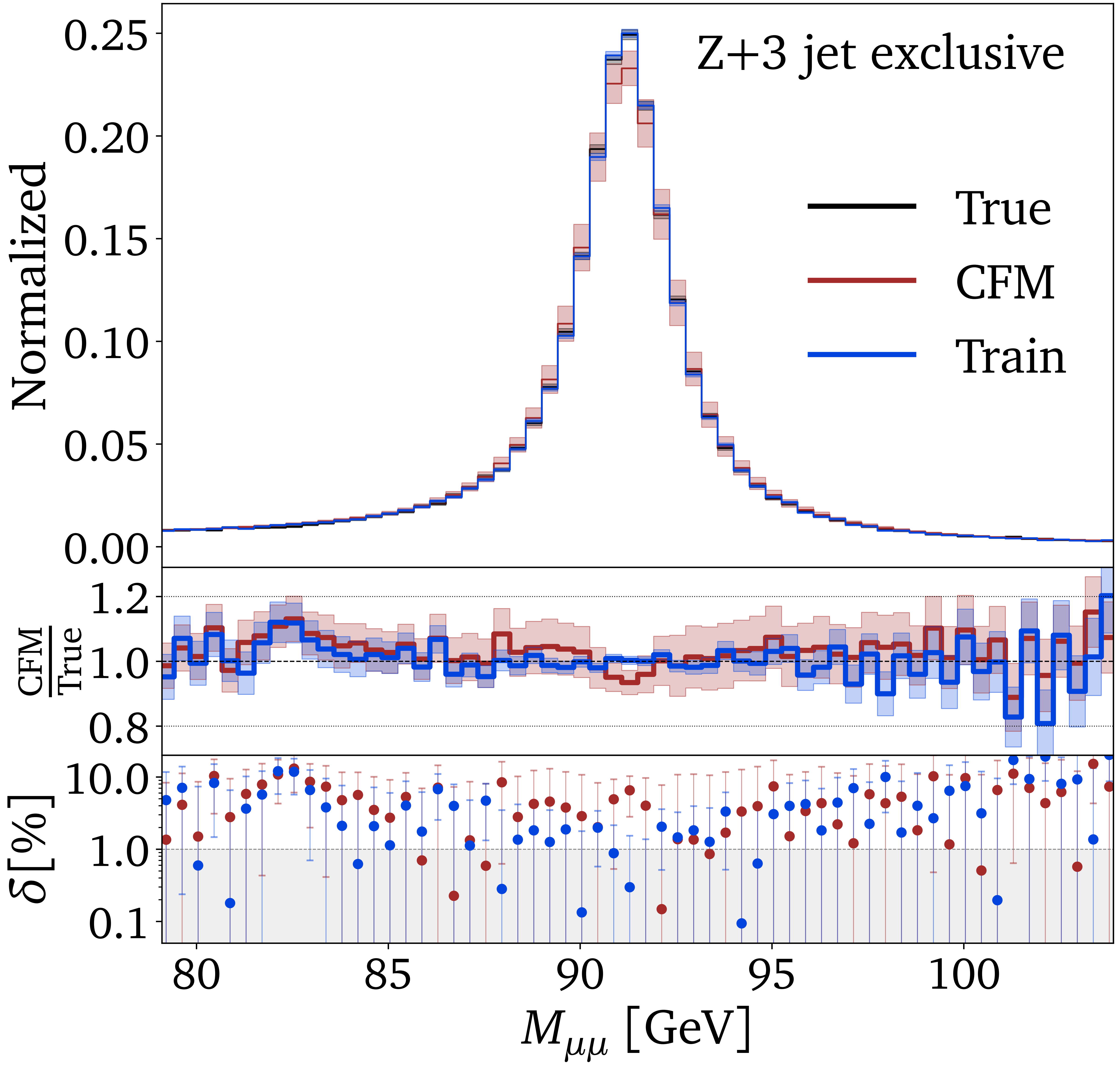


# To be precise

Percent level precision (comparable to statistical uncertainty)

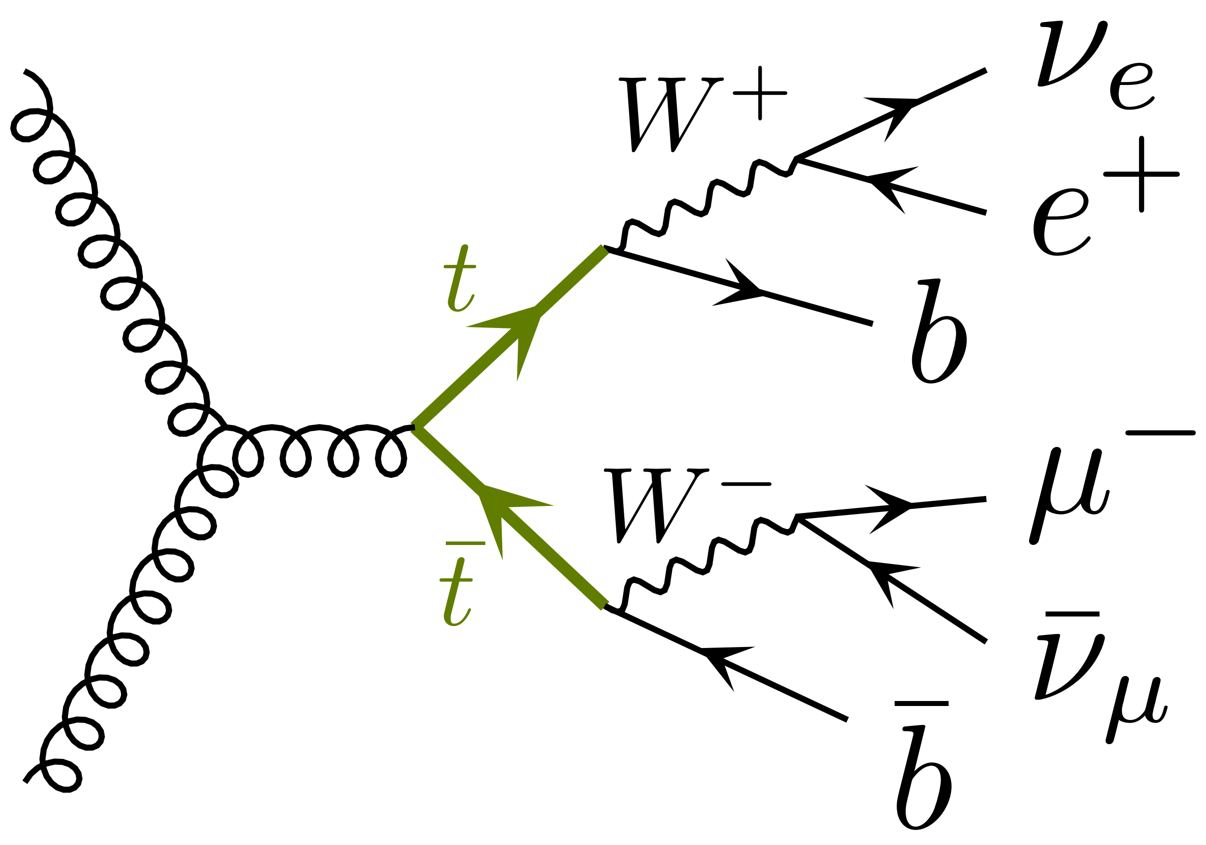
Uncertainty well-defined

Surpasses INN precision (A. Butter et al.: arXiv:2110.13632)



# Concrete Application — Off-Shell processes

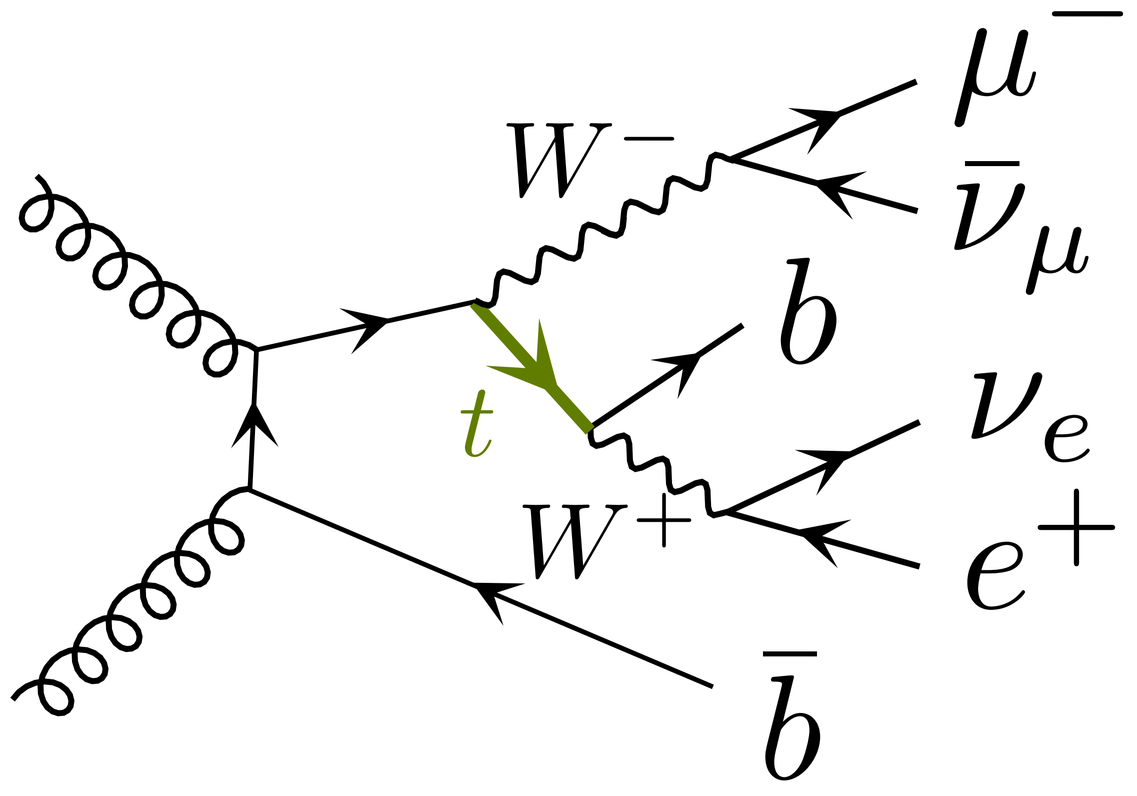
Leptonic ttbar-decay

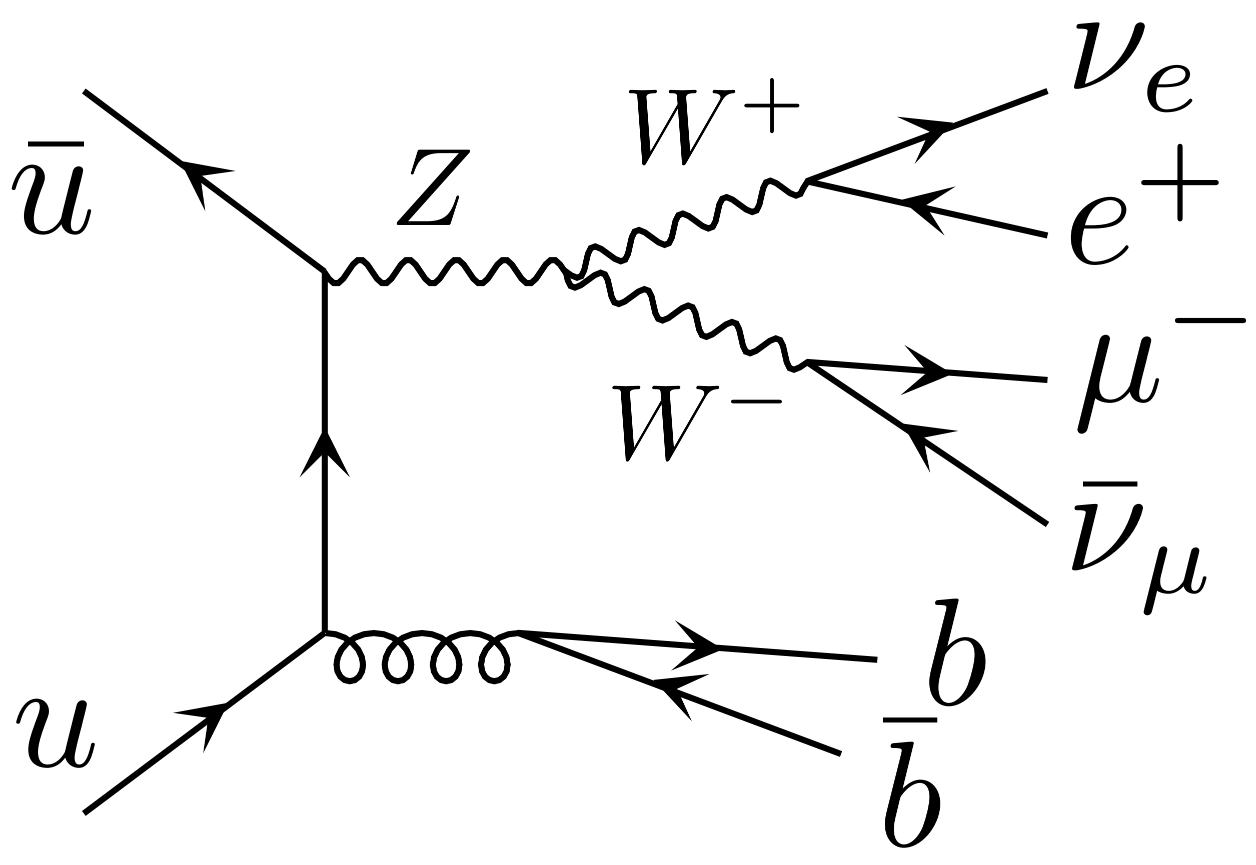



Precise simulation of ttbar decays critical for LHC analyses

# Concrete Application — Off-Shell processes

Off-Shell processes



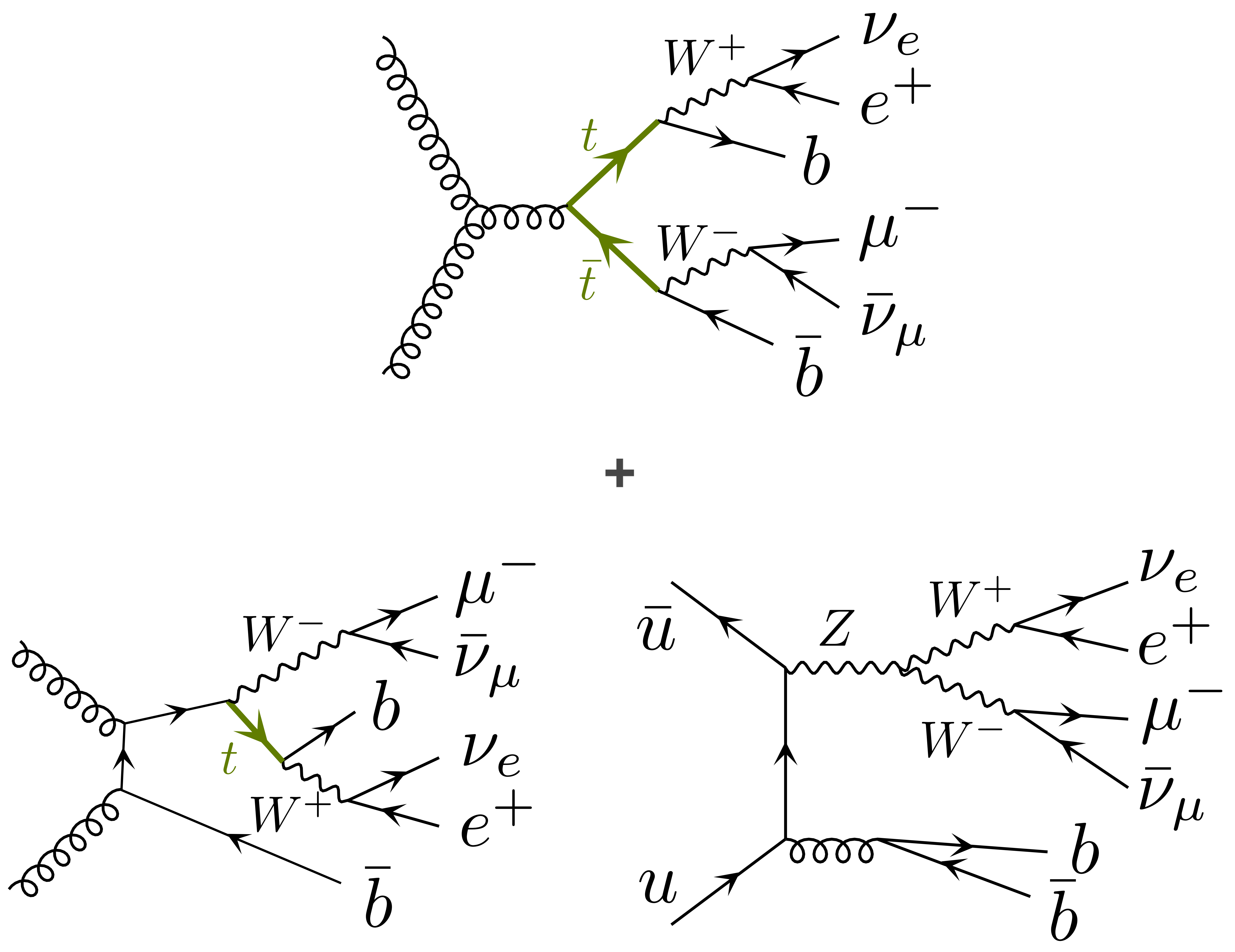




Precise simulation of ttbar decays critical for LHC analyses

Need to account for off shell processes

Including off-shell processes
= extremely costly

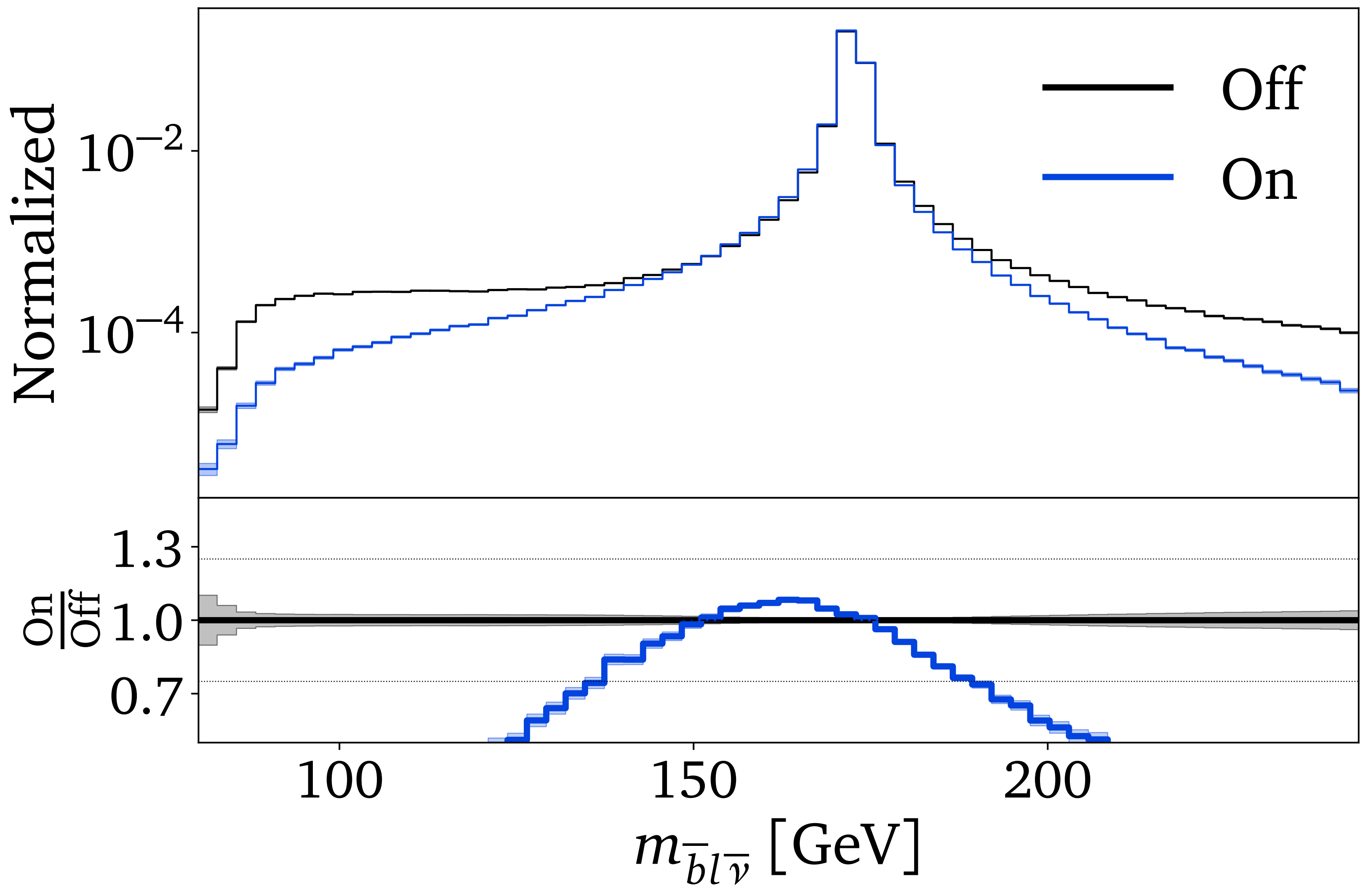
# Concrete Application — Off-Shell processes

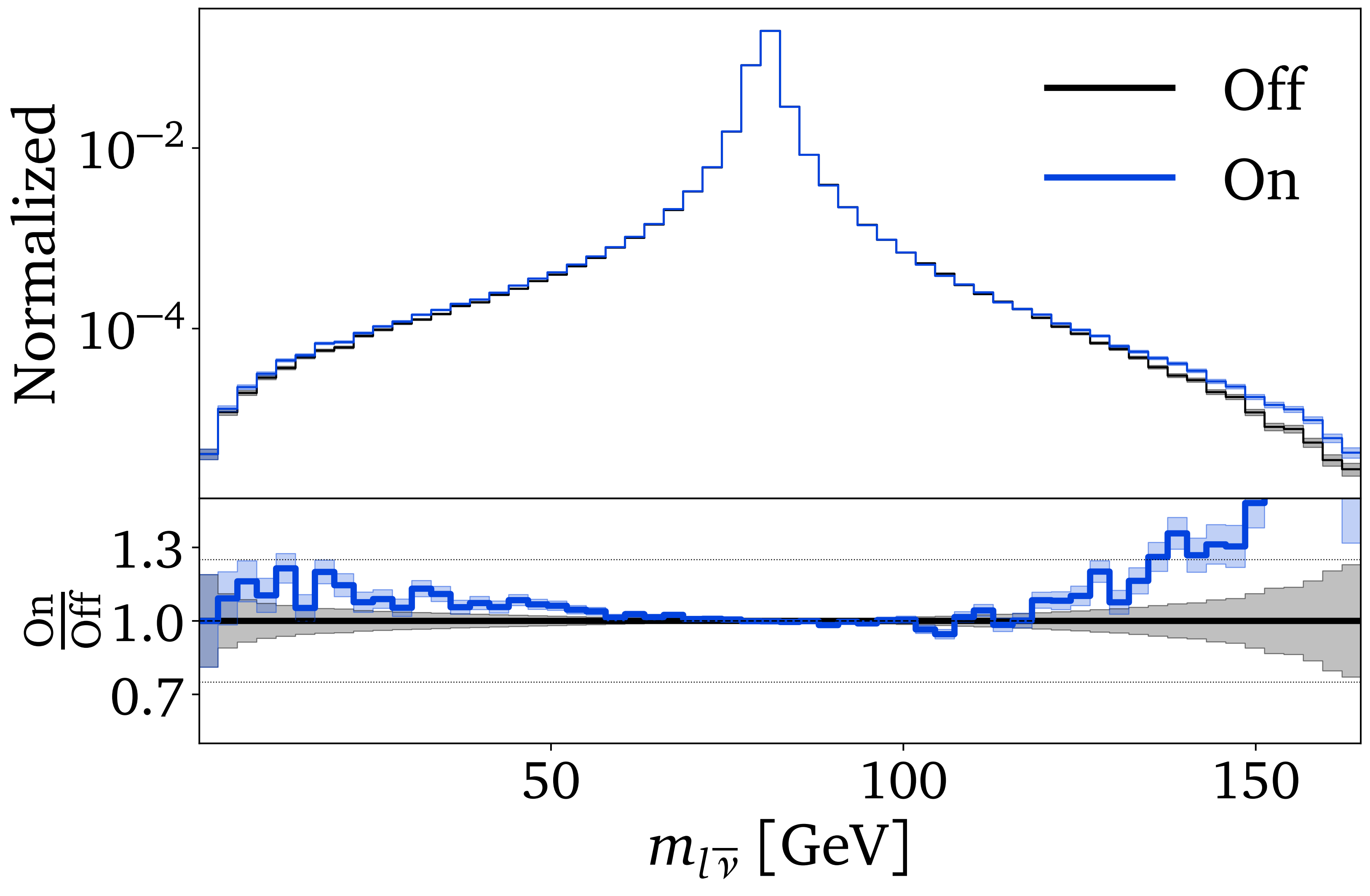


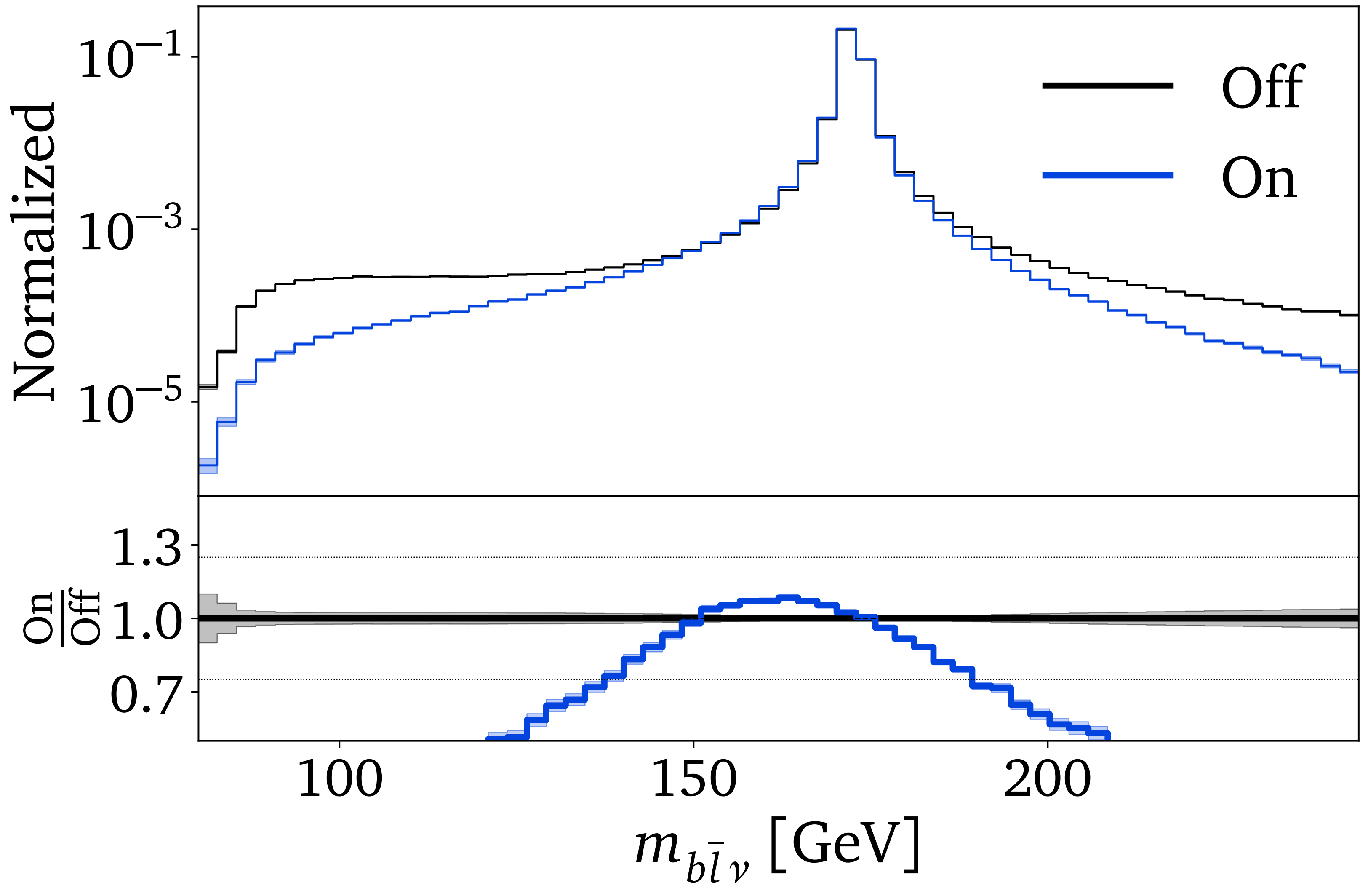

Need: Fast event generator

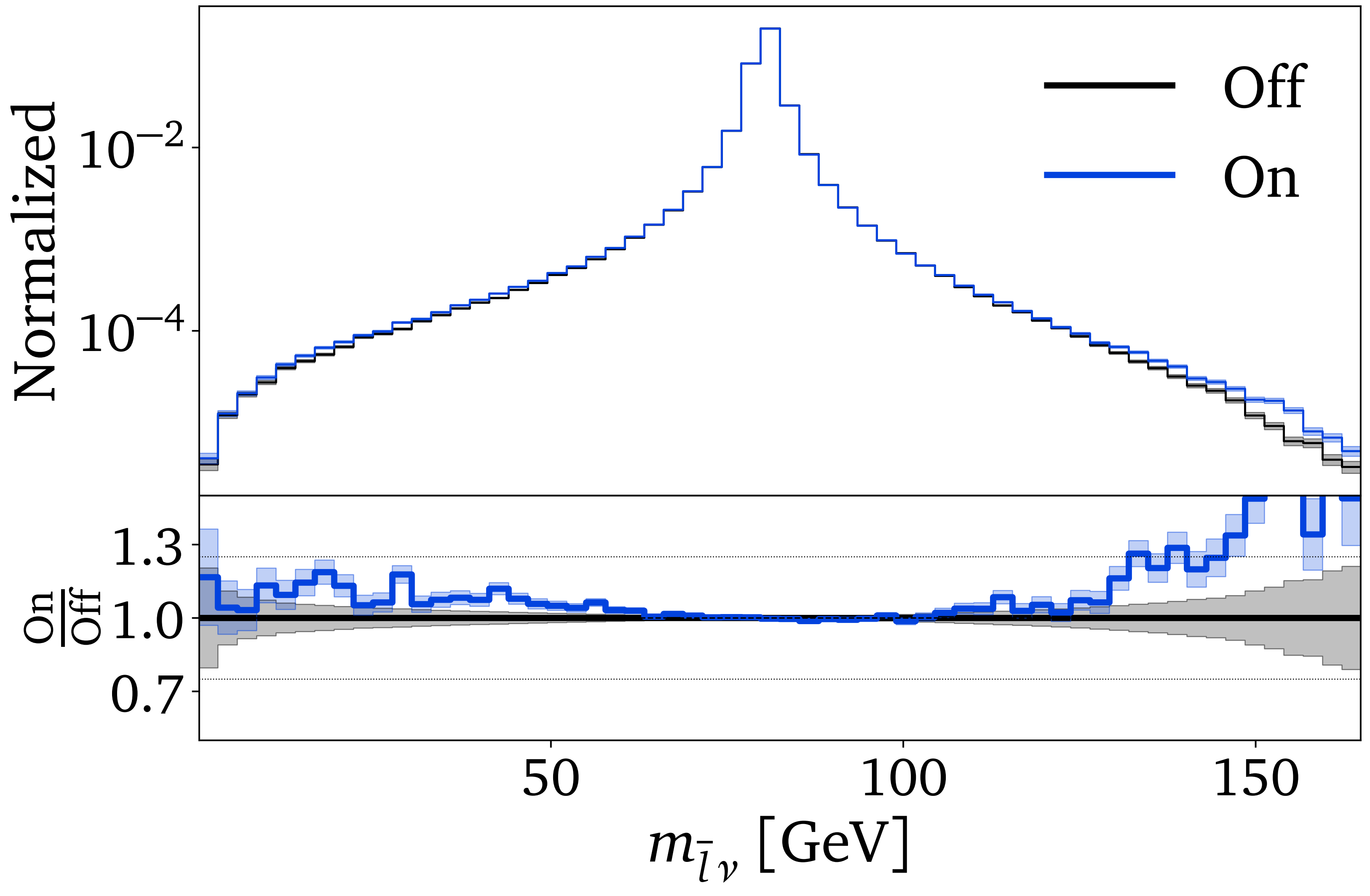
Problem: Multiresonant phase space in 24 dimensions

# What's the problem?

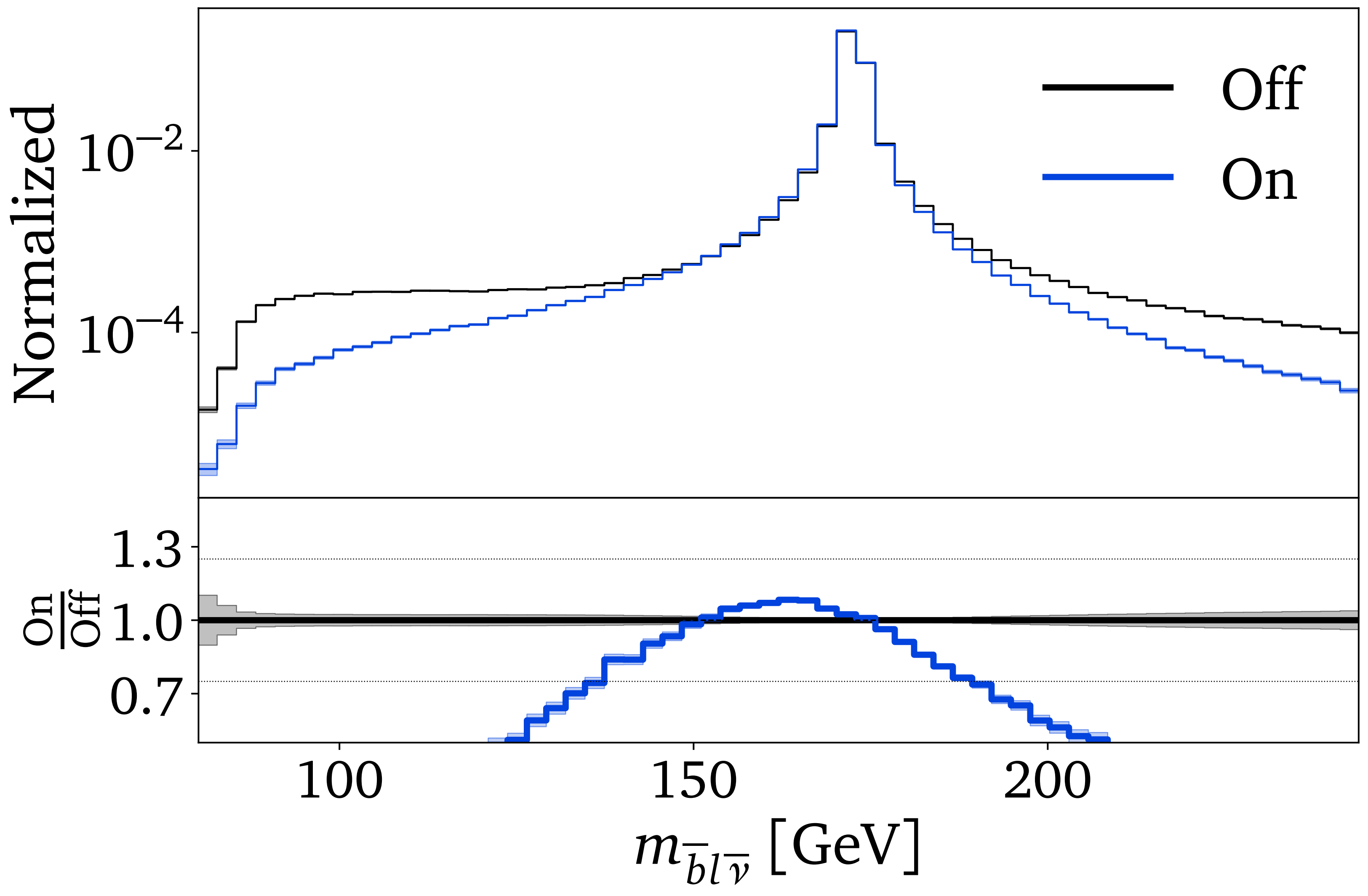


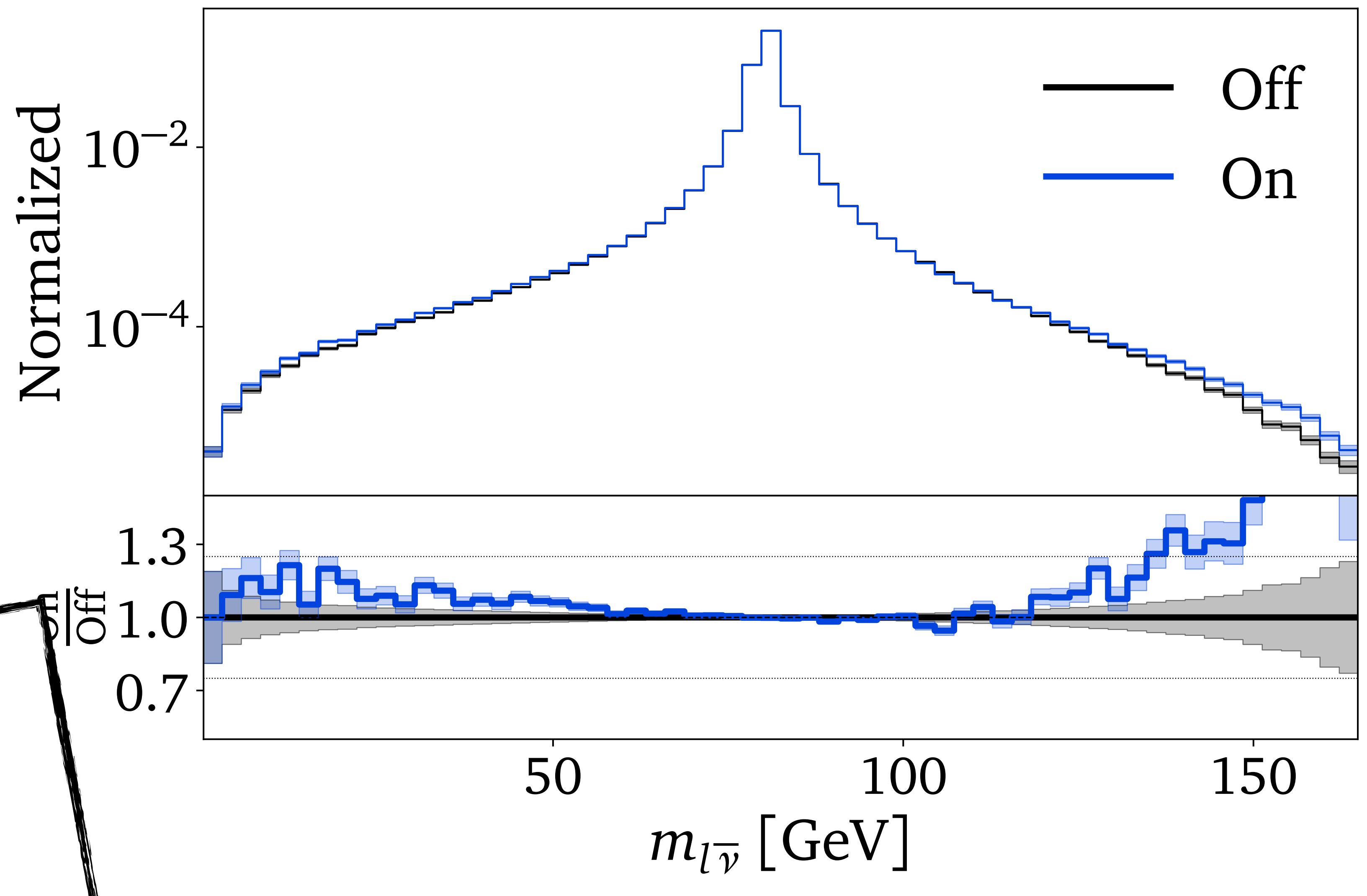



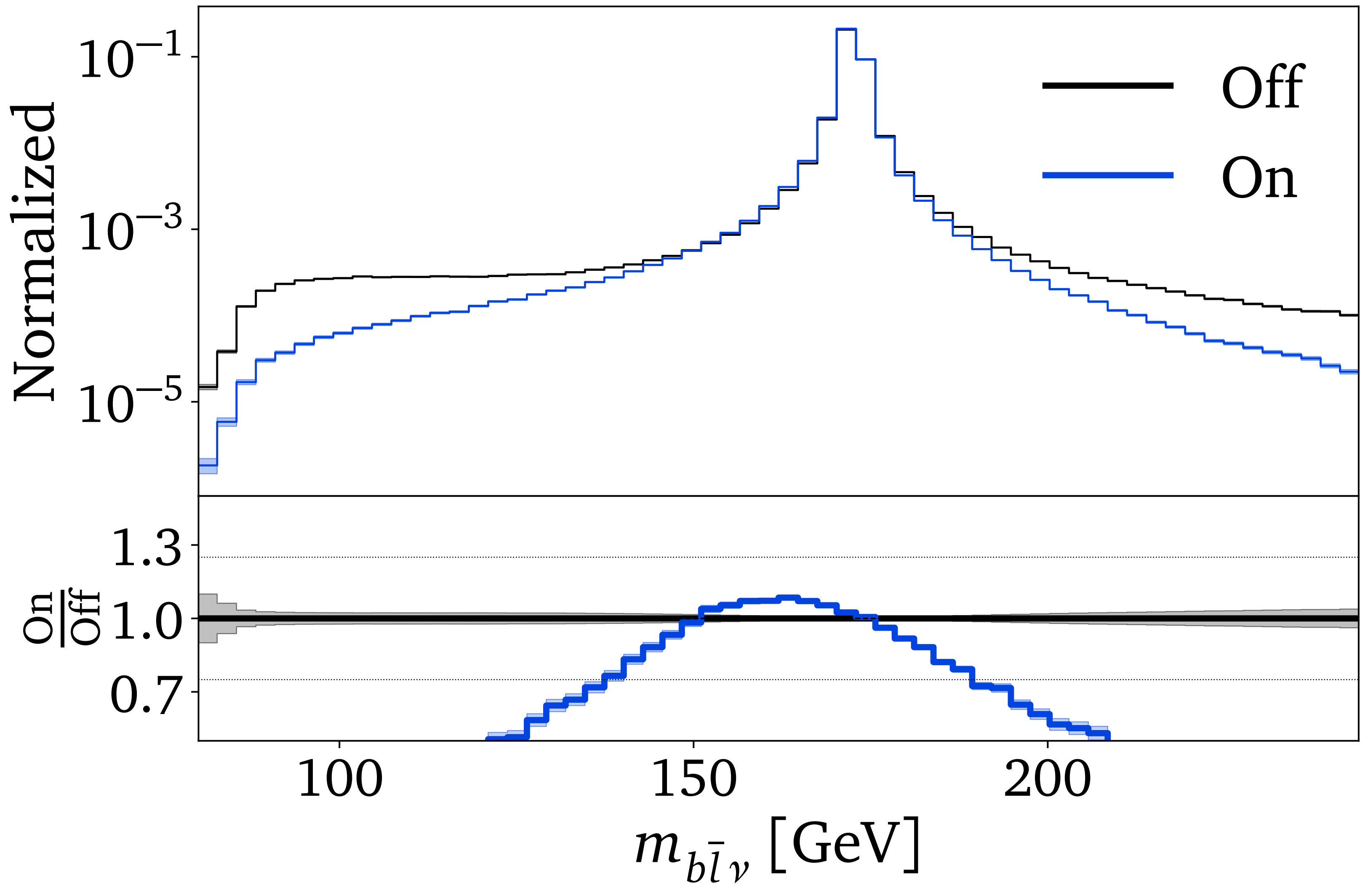



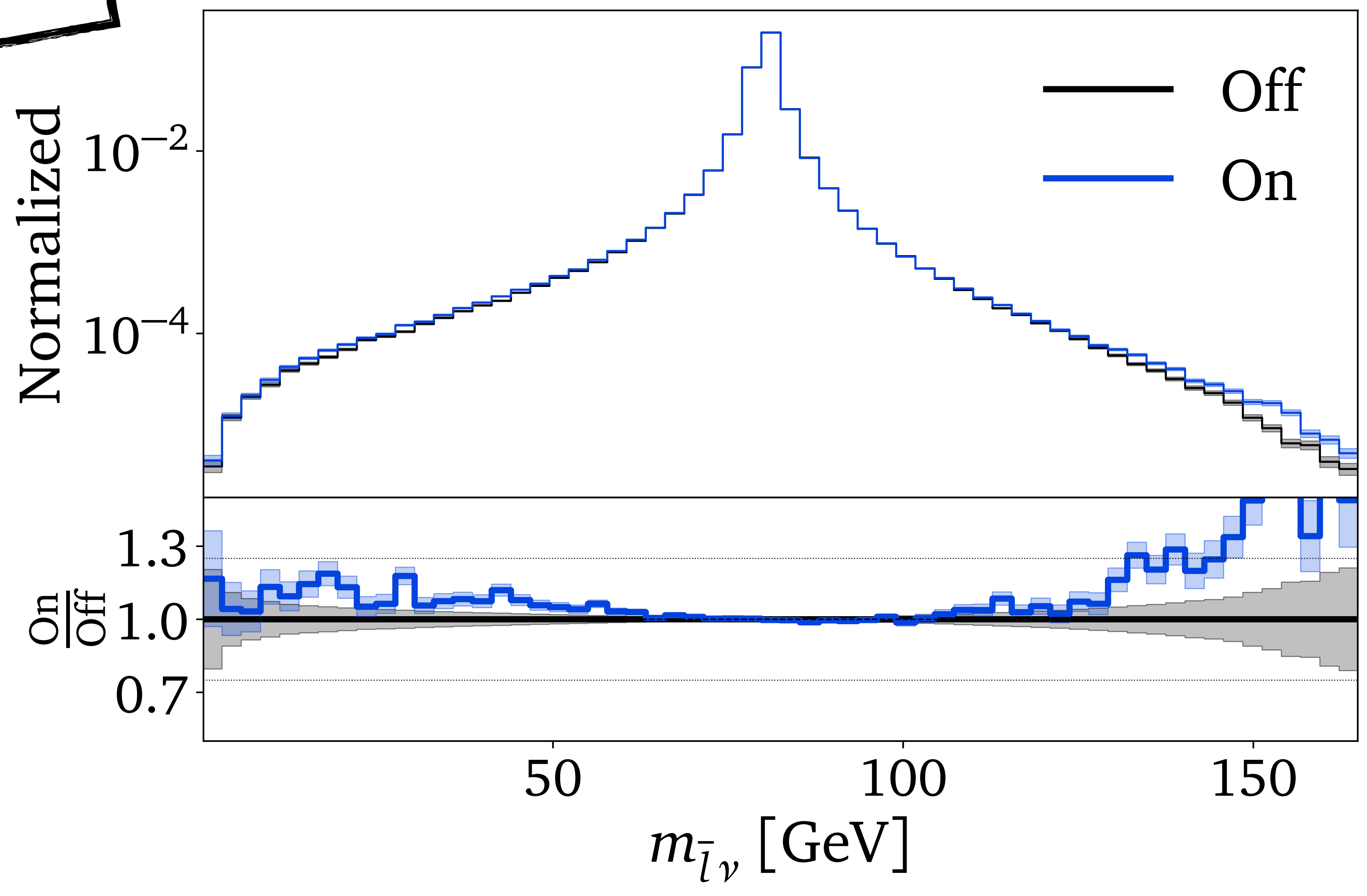
# What's the problem?



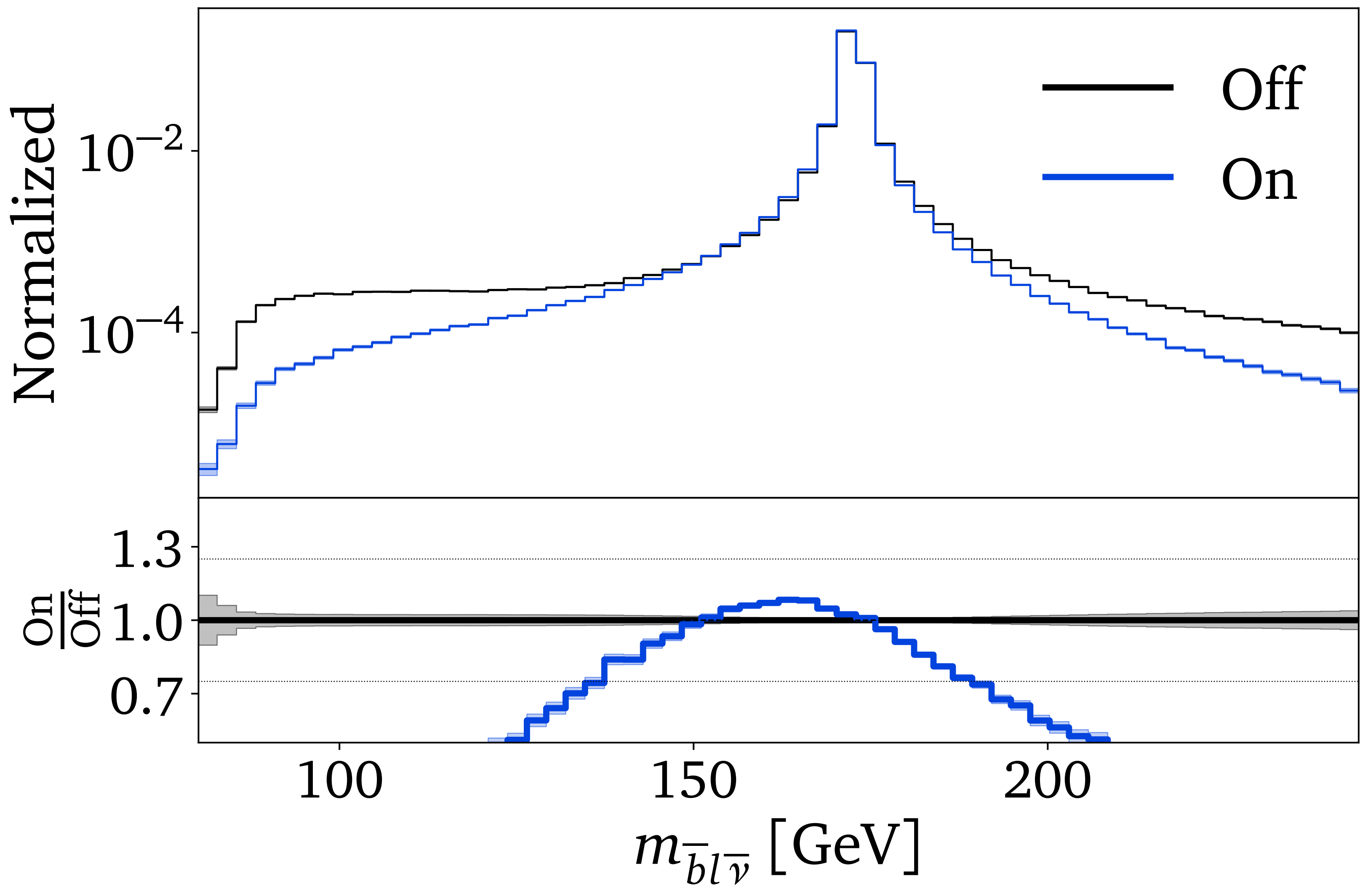



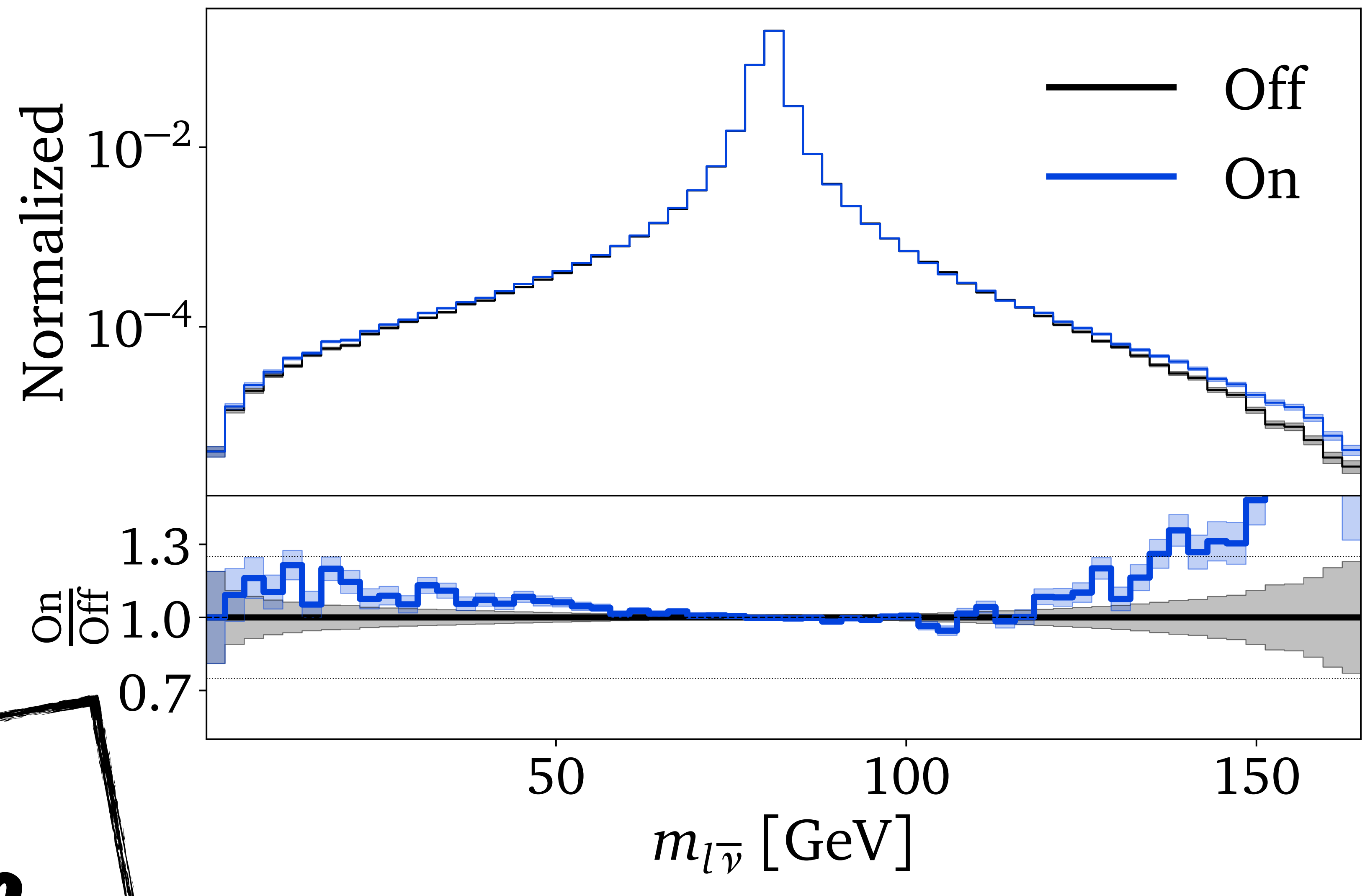

Can we just learn correction?



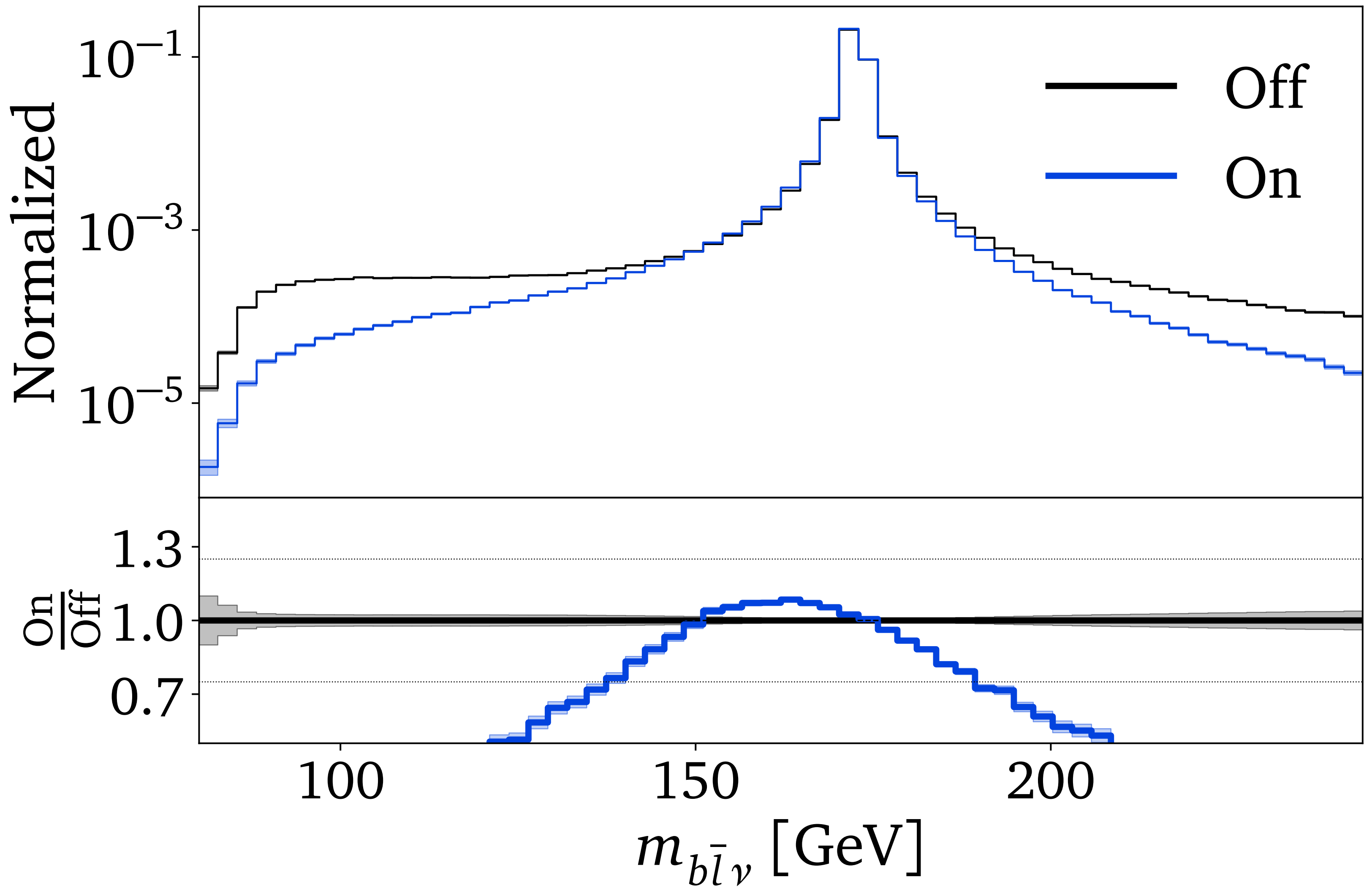



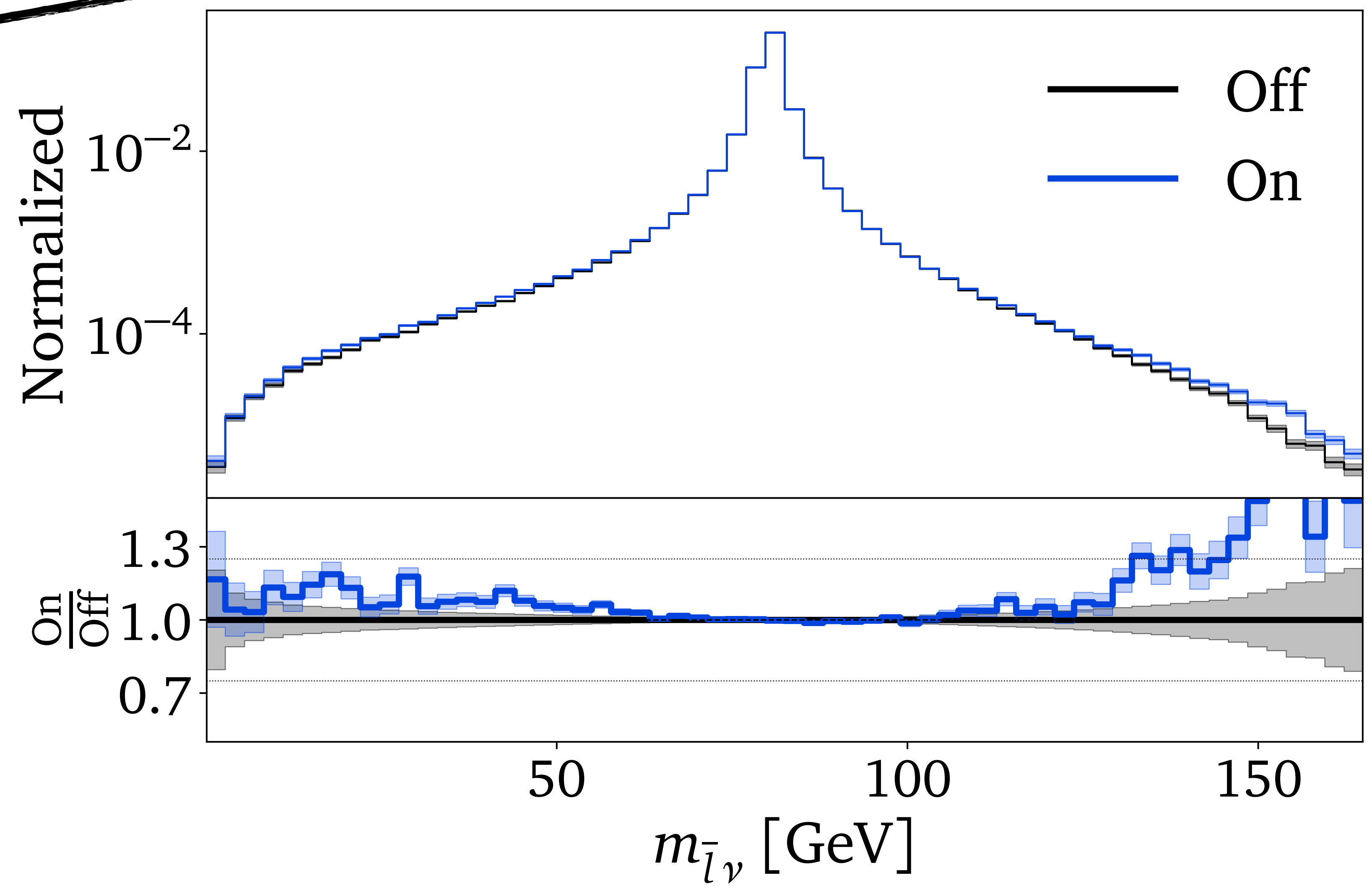
# How to not learn correction



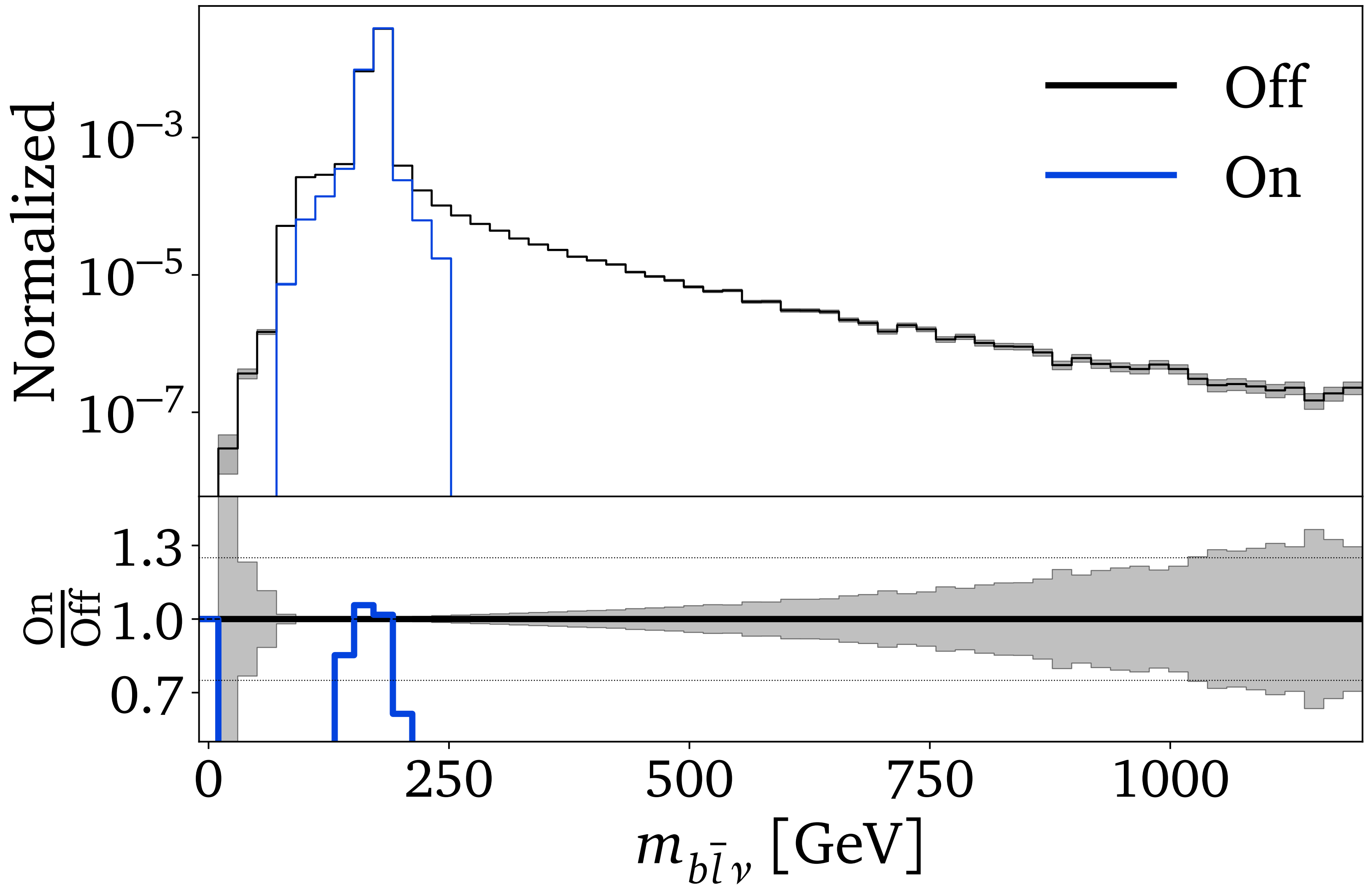



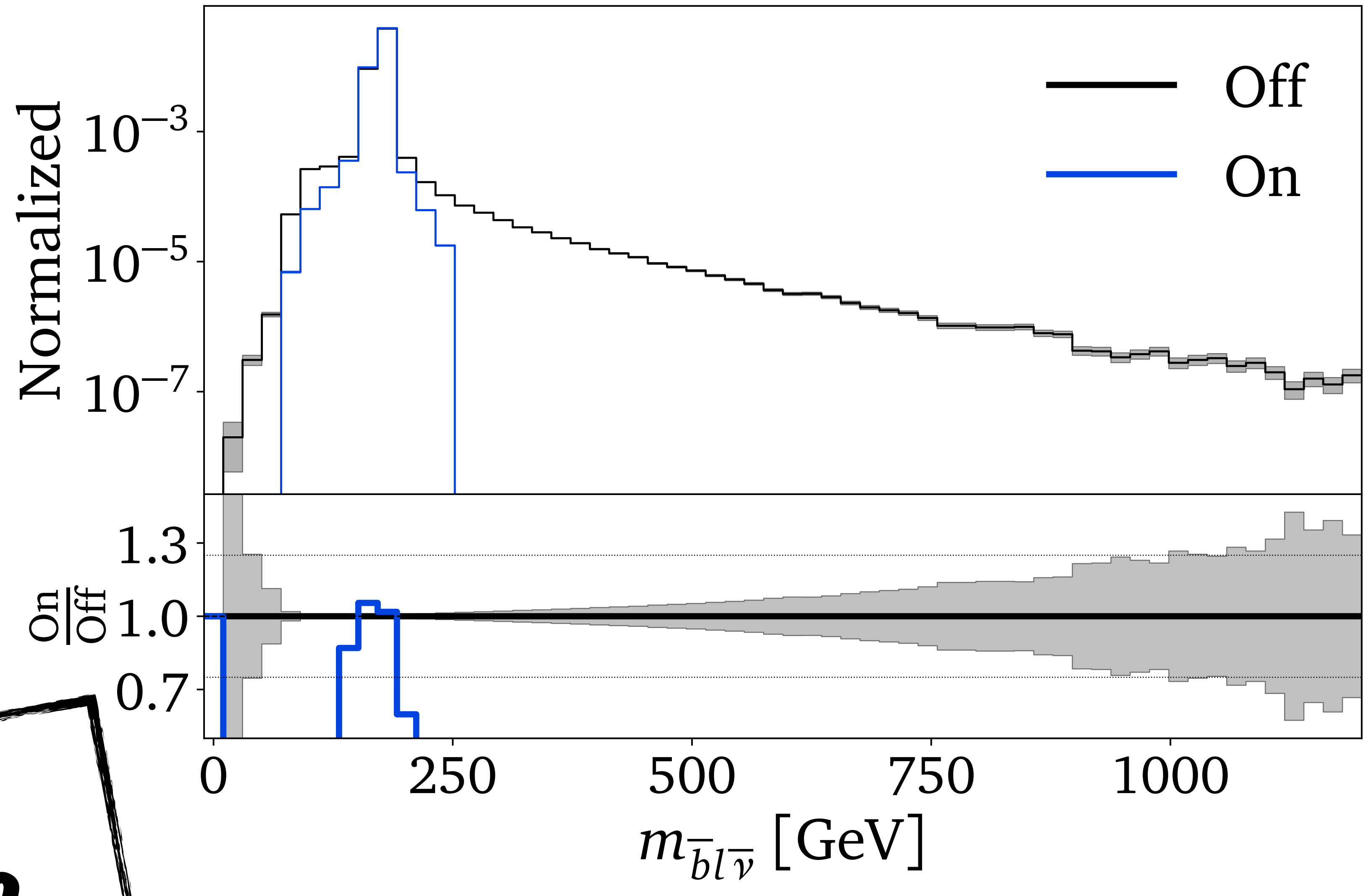

**Classifier reweighting?**



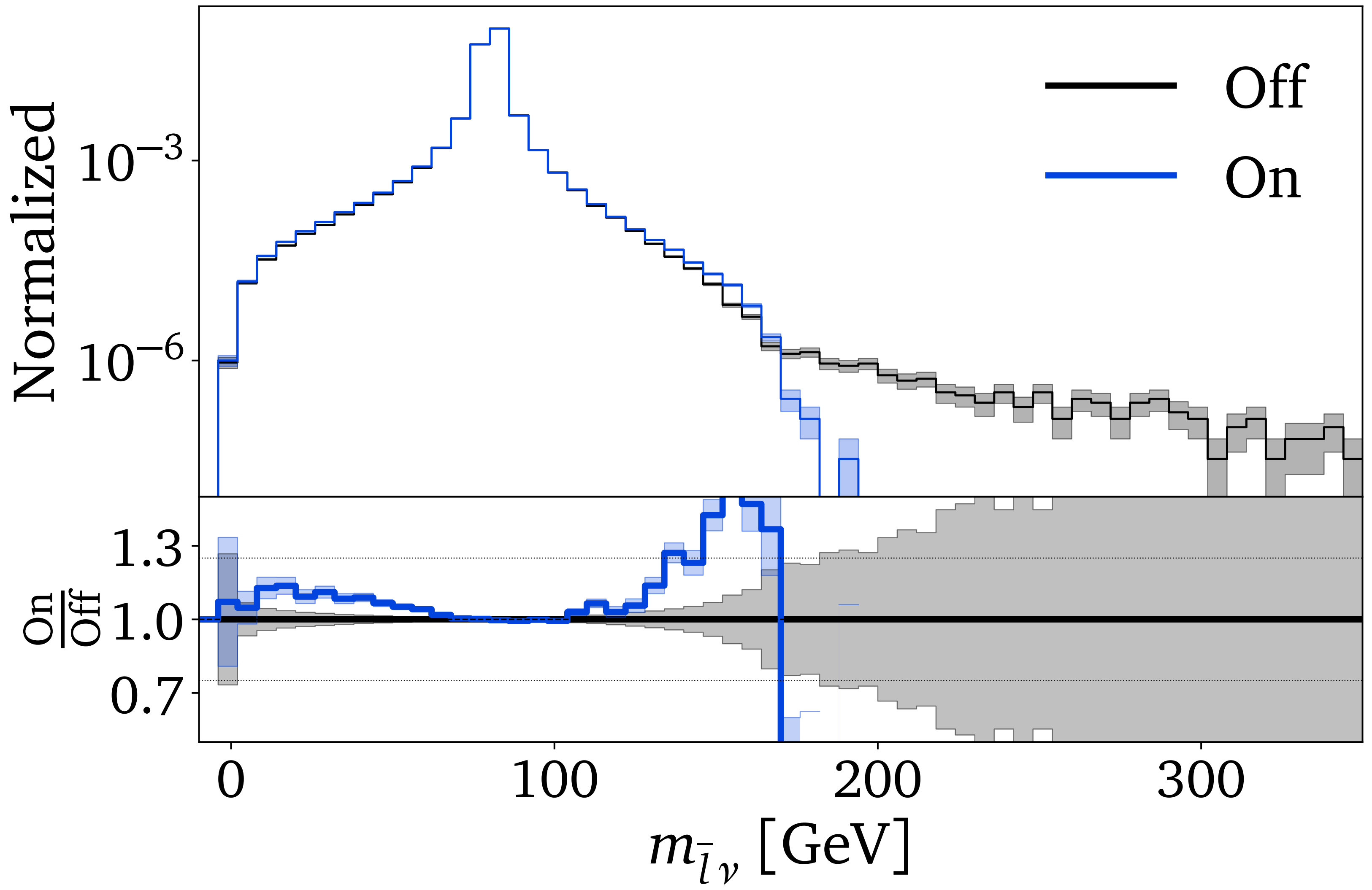



# How to not learn correction

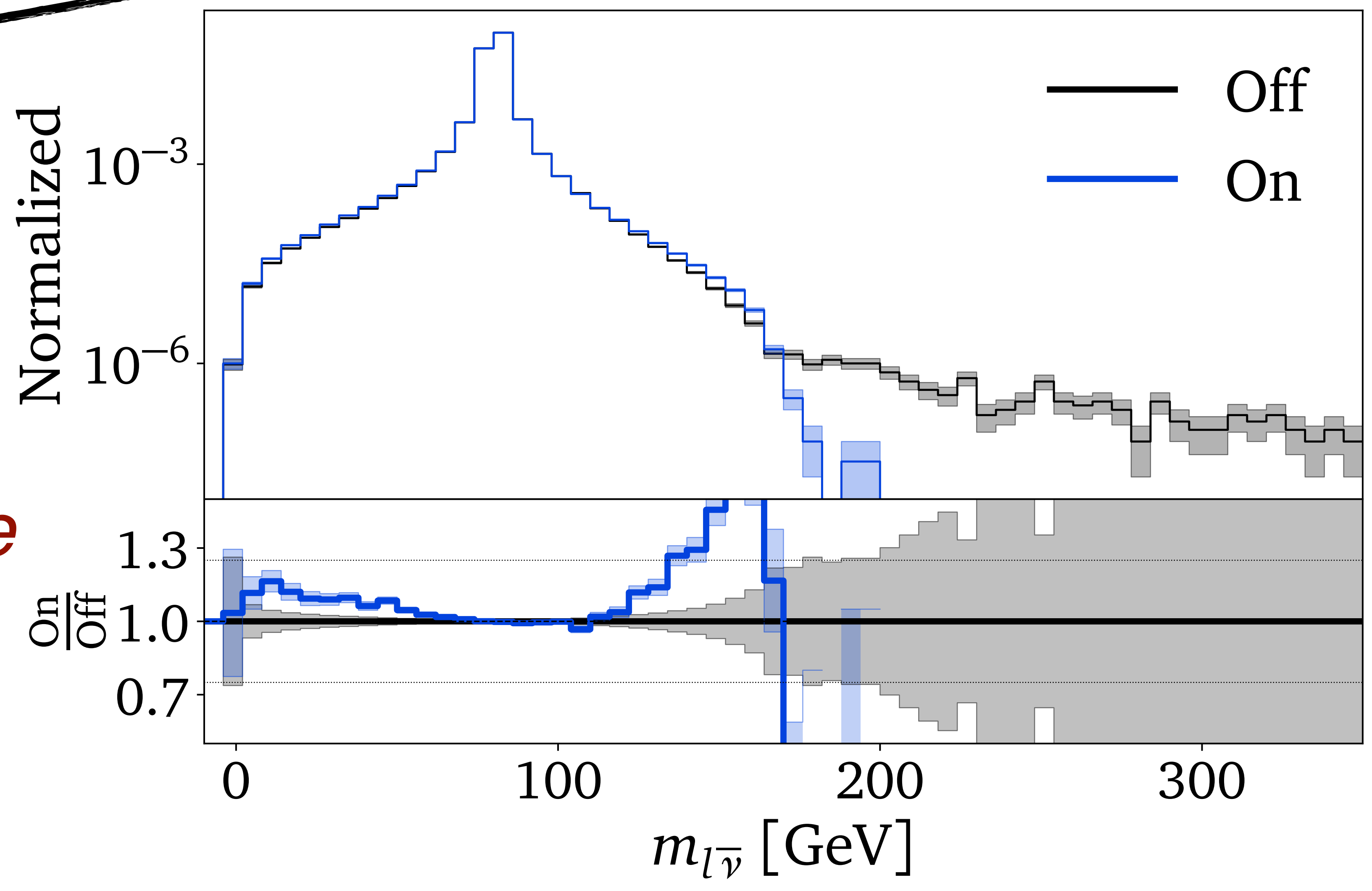


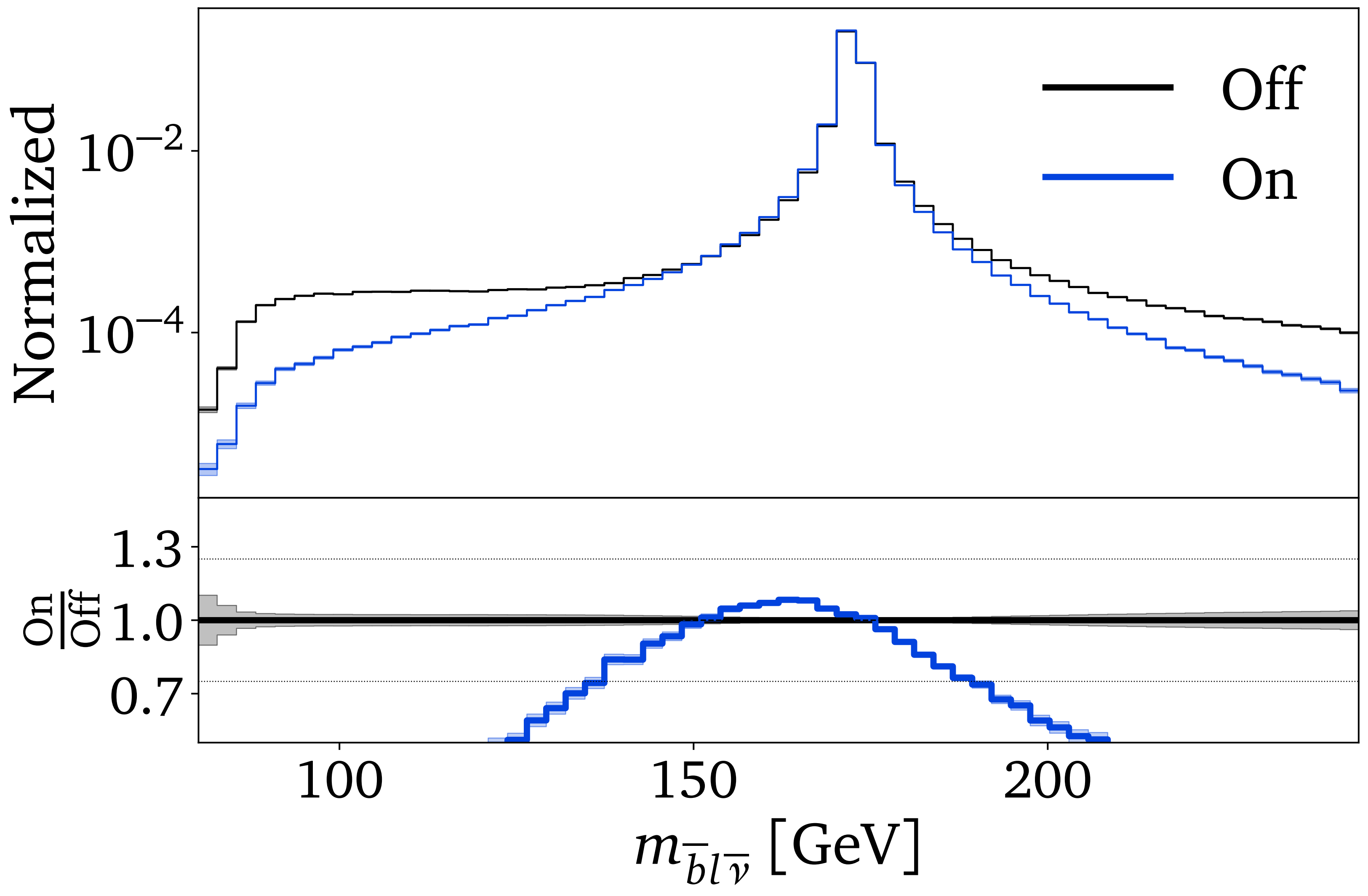

~~Classifier reweighting?~~

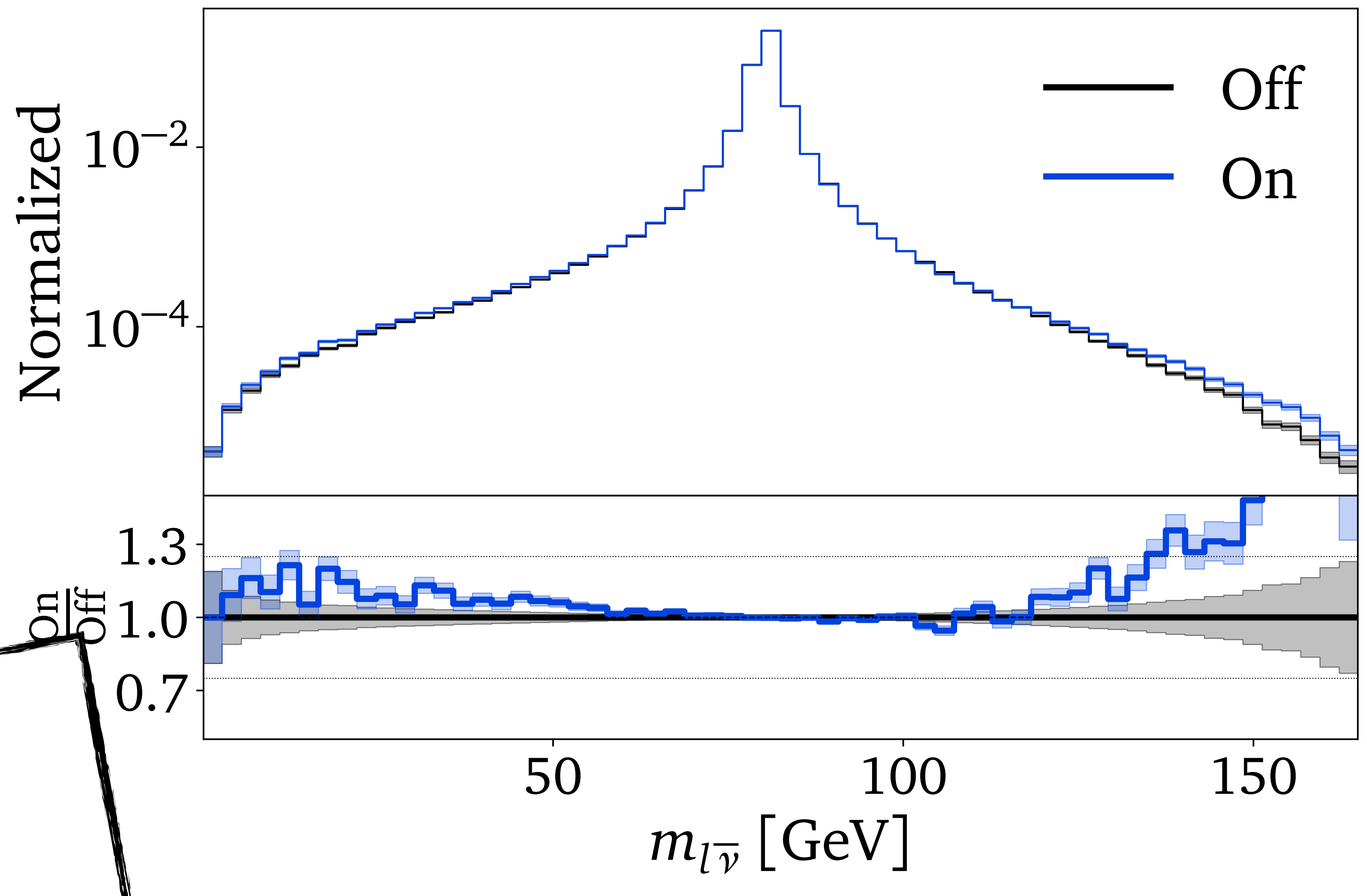



*On- & Off-shell phase space not the same support*

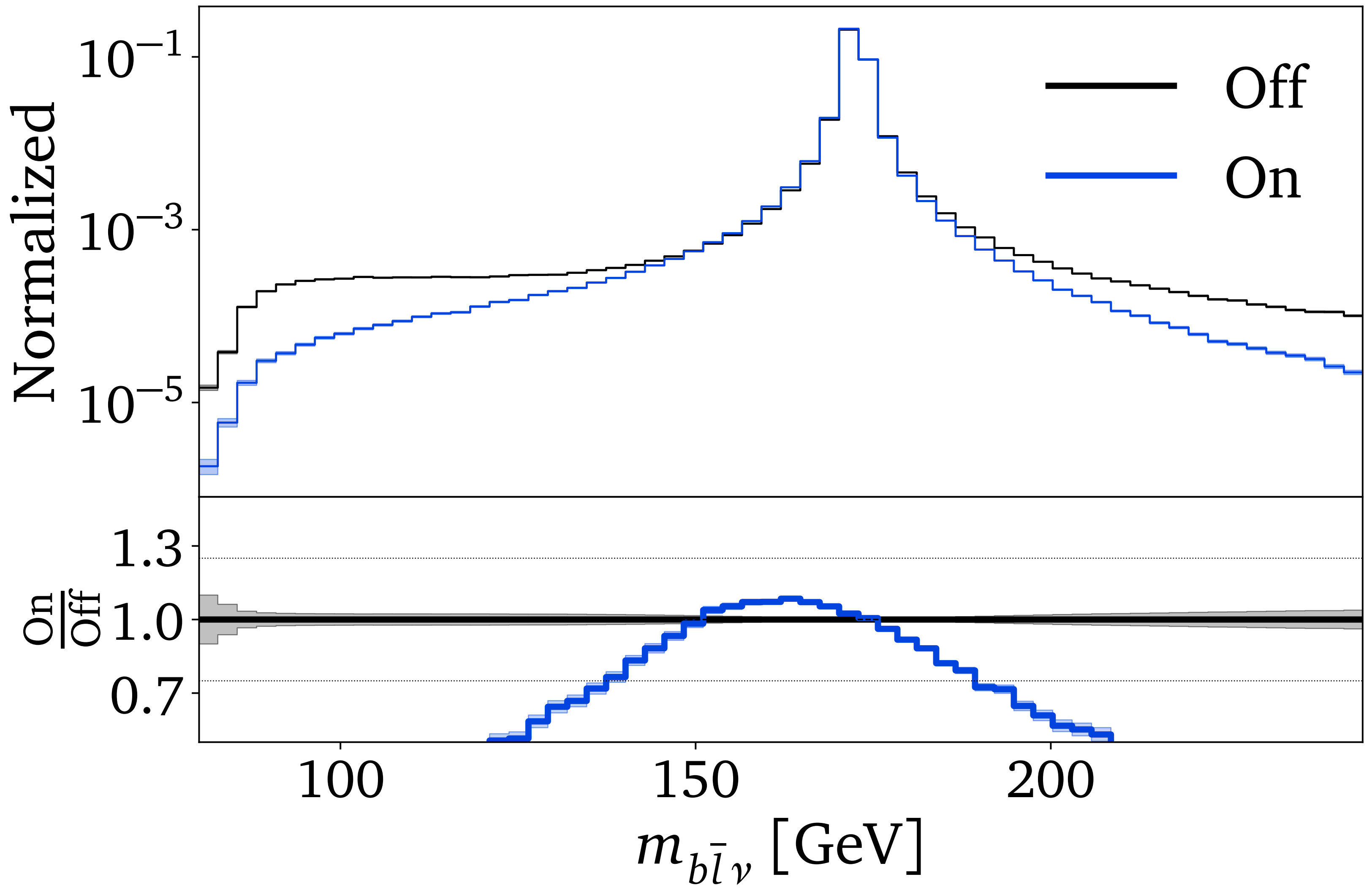


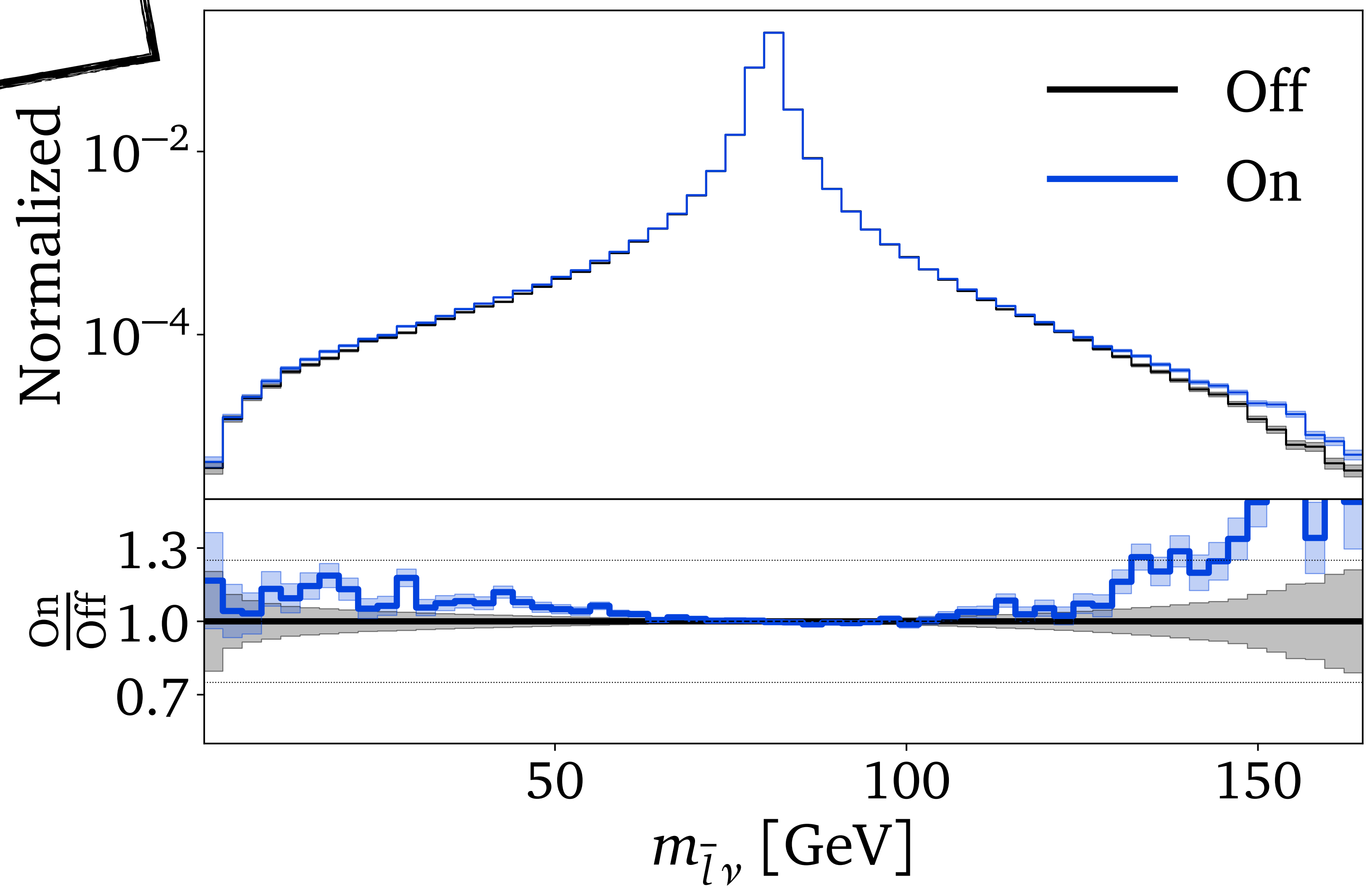
# How to not learn correction



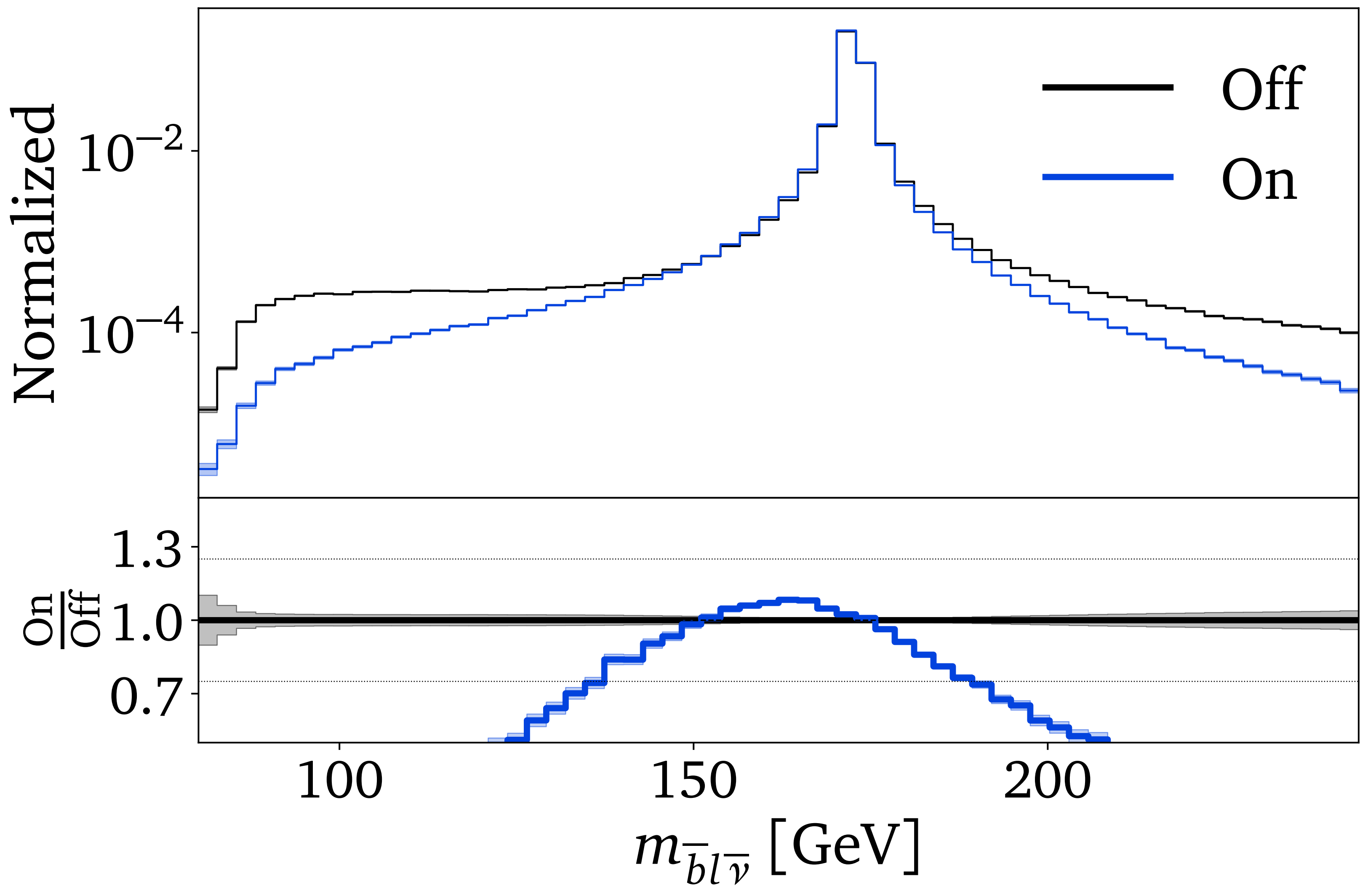



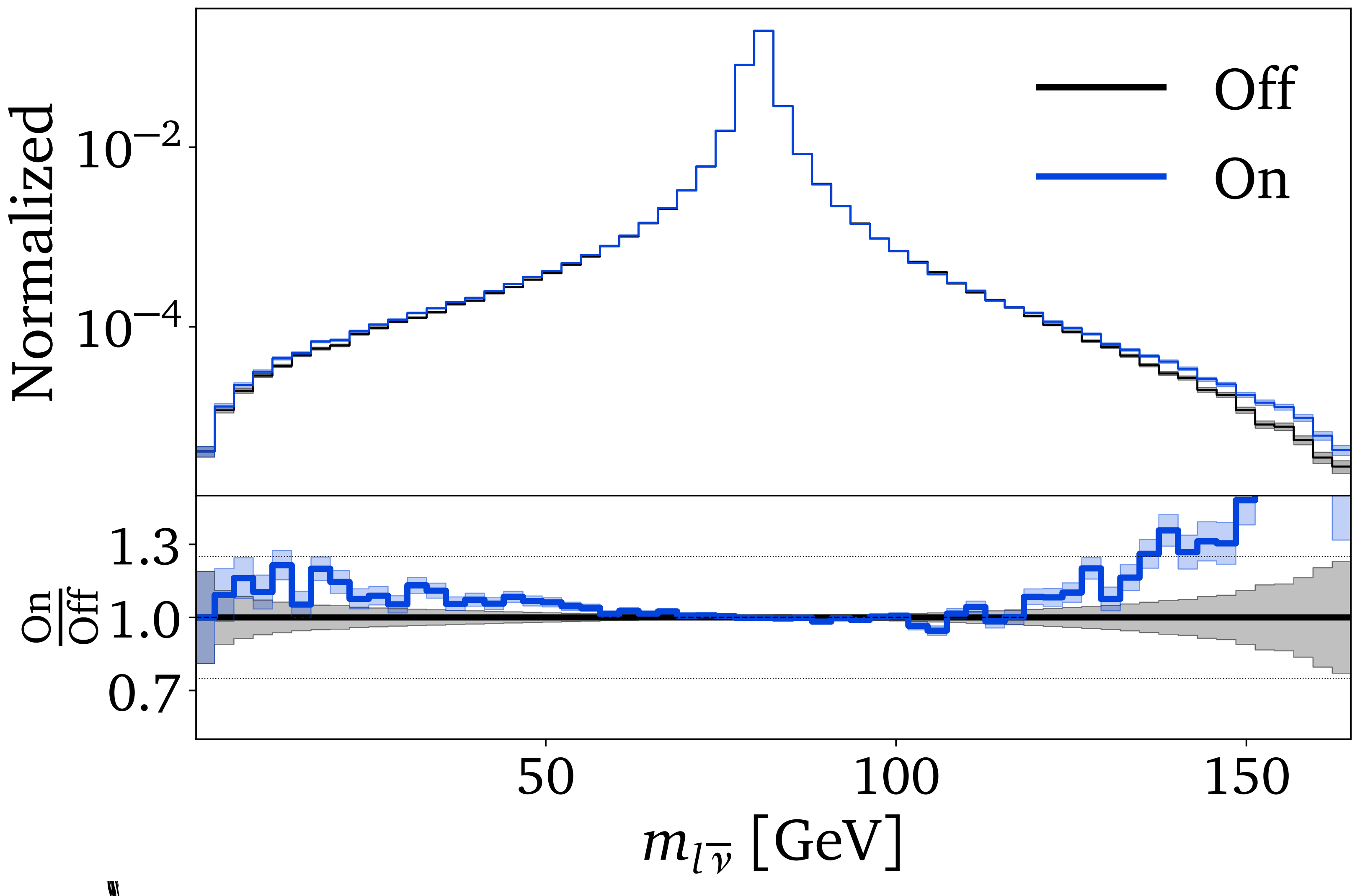

Method à la generative unfolding?



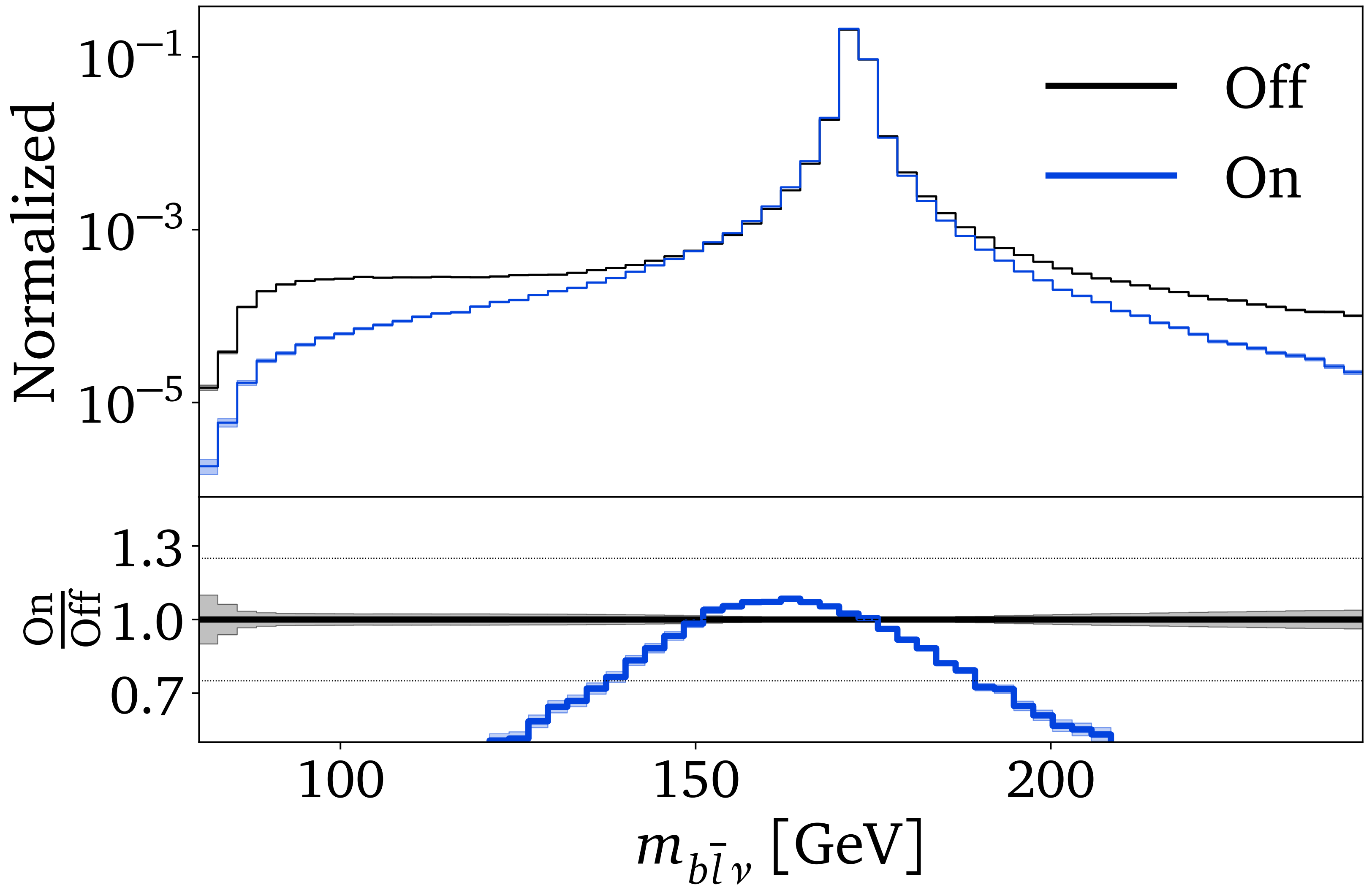



# How to not learn correction

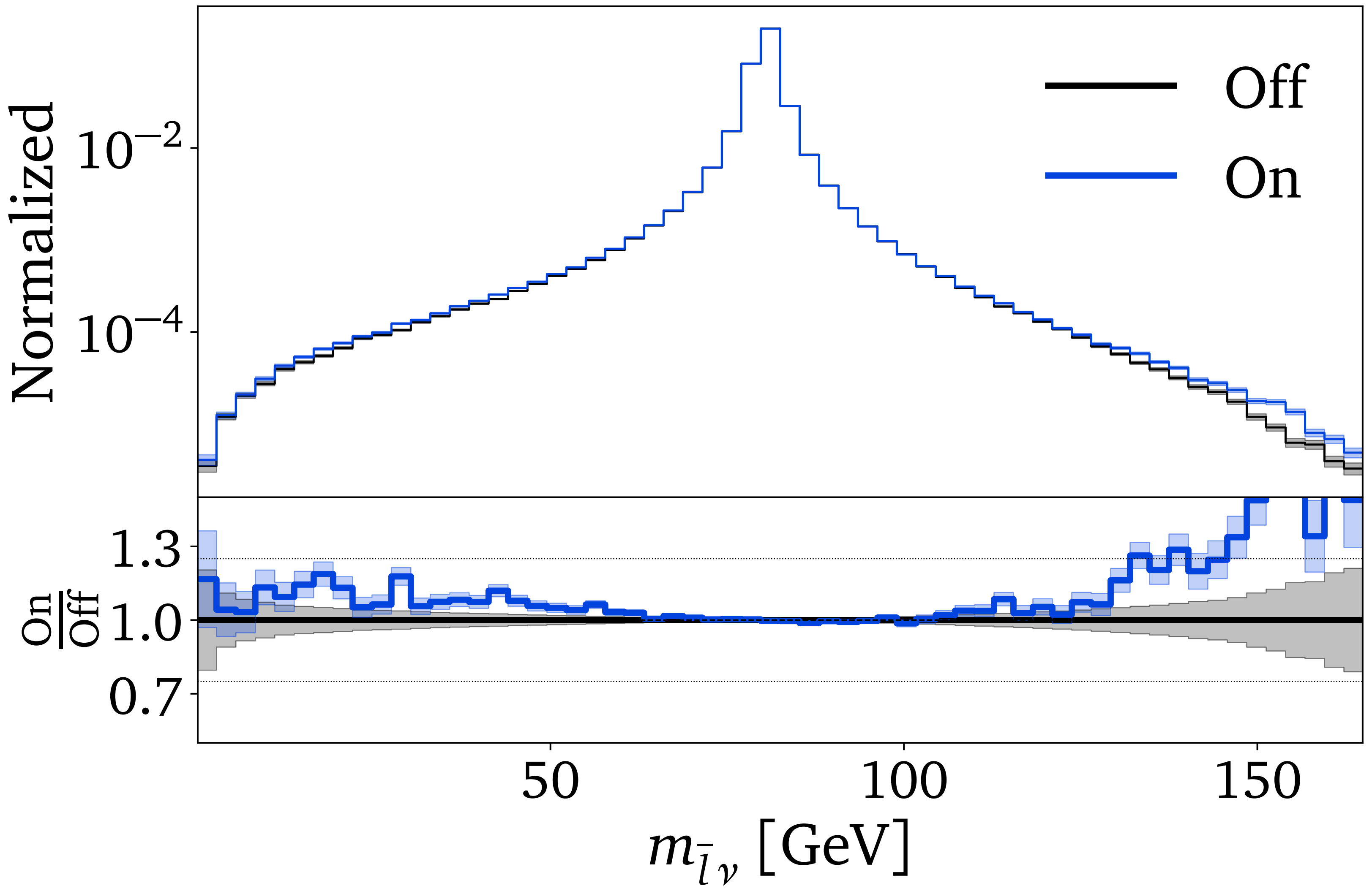



~~Method à la generative unfolding?~~




*No paired data*



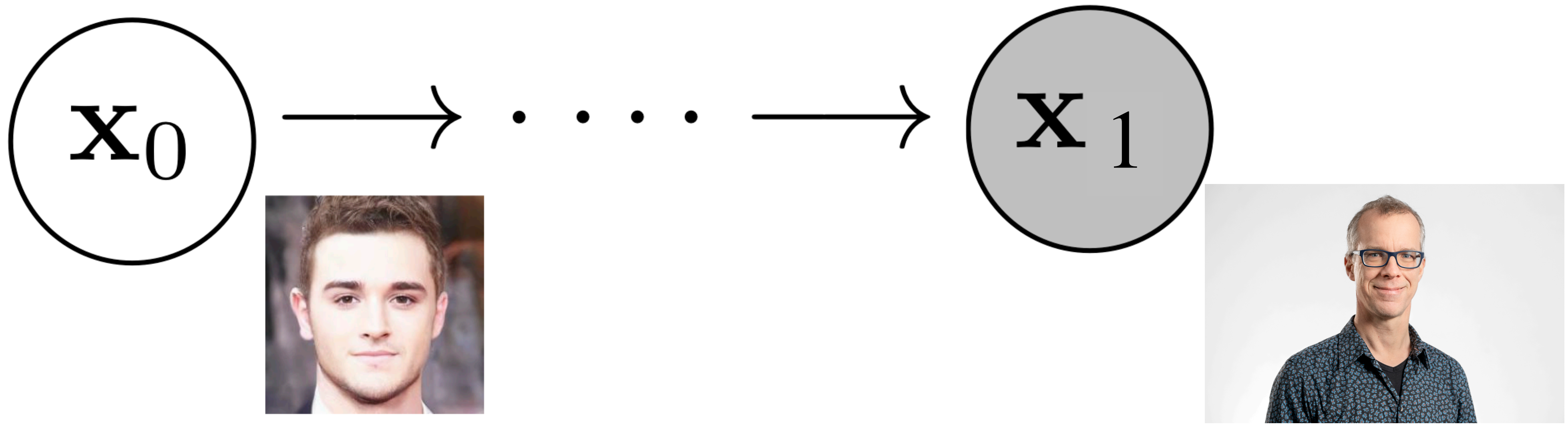
# Diffusion Models (CFM) — revisit

Phase Space

$x_0 \sim p_0(x_0)$

Latent Space

$x_1 \sim p_1(x_1)$

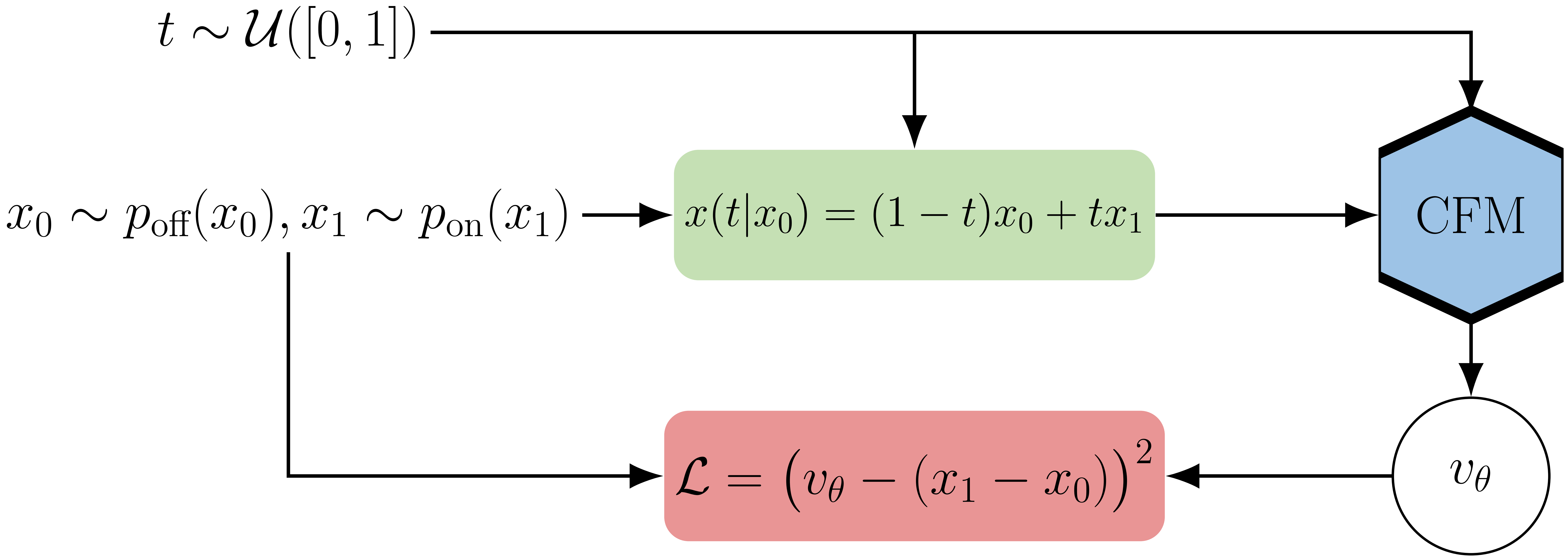



**Continuous time evolution** **t**

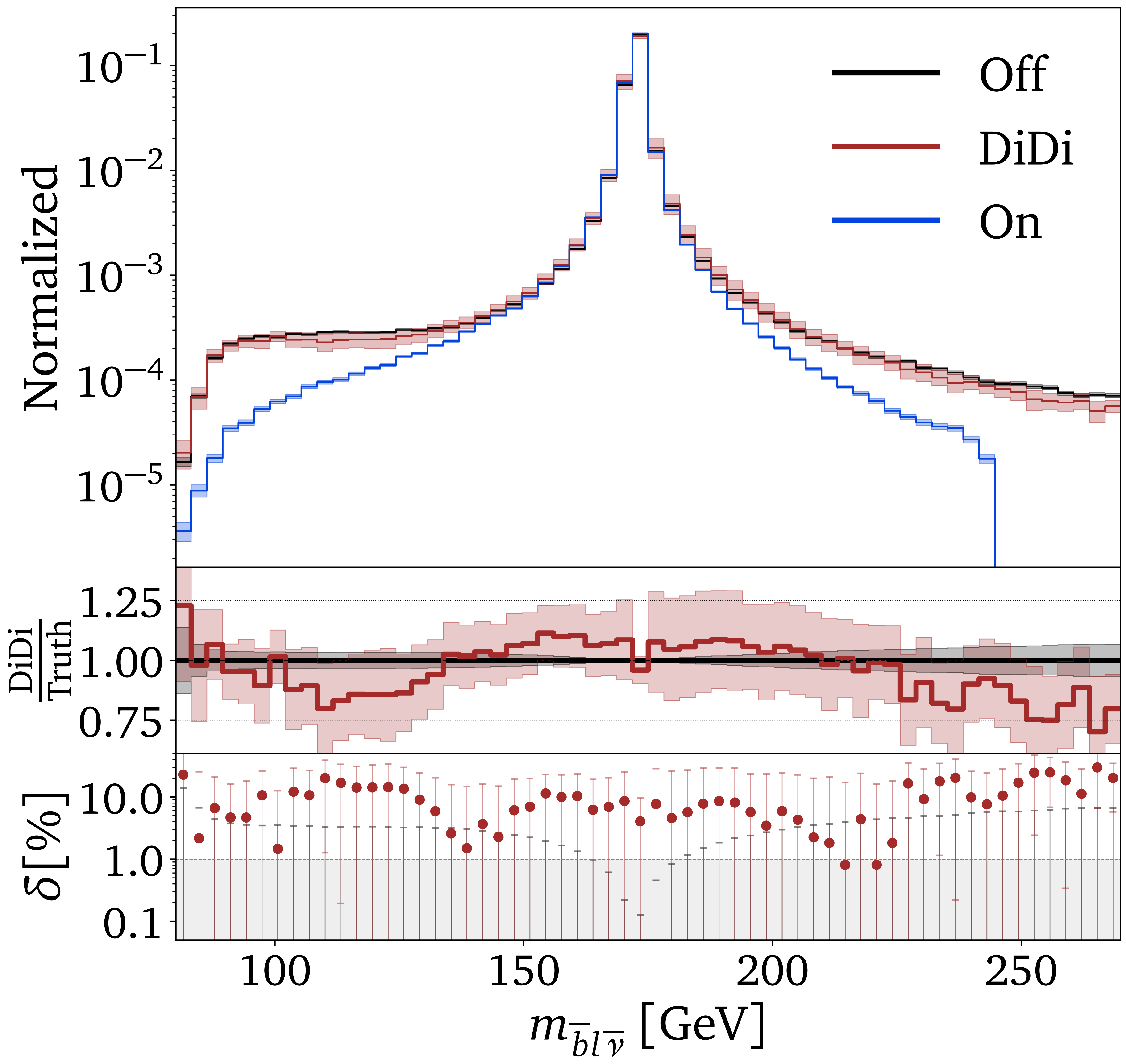
Evolution governed by $v \equiv \dfrac{dx}{dt}$

Figure adapted from J.Ho et al.: arXiv:2006.11239

# Diffusion Models (CFM)



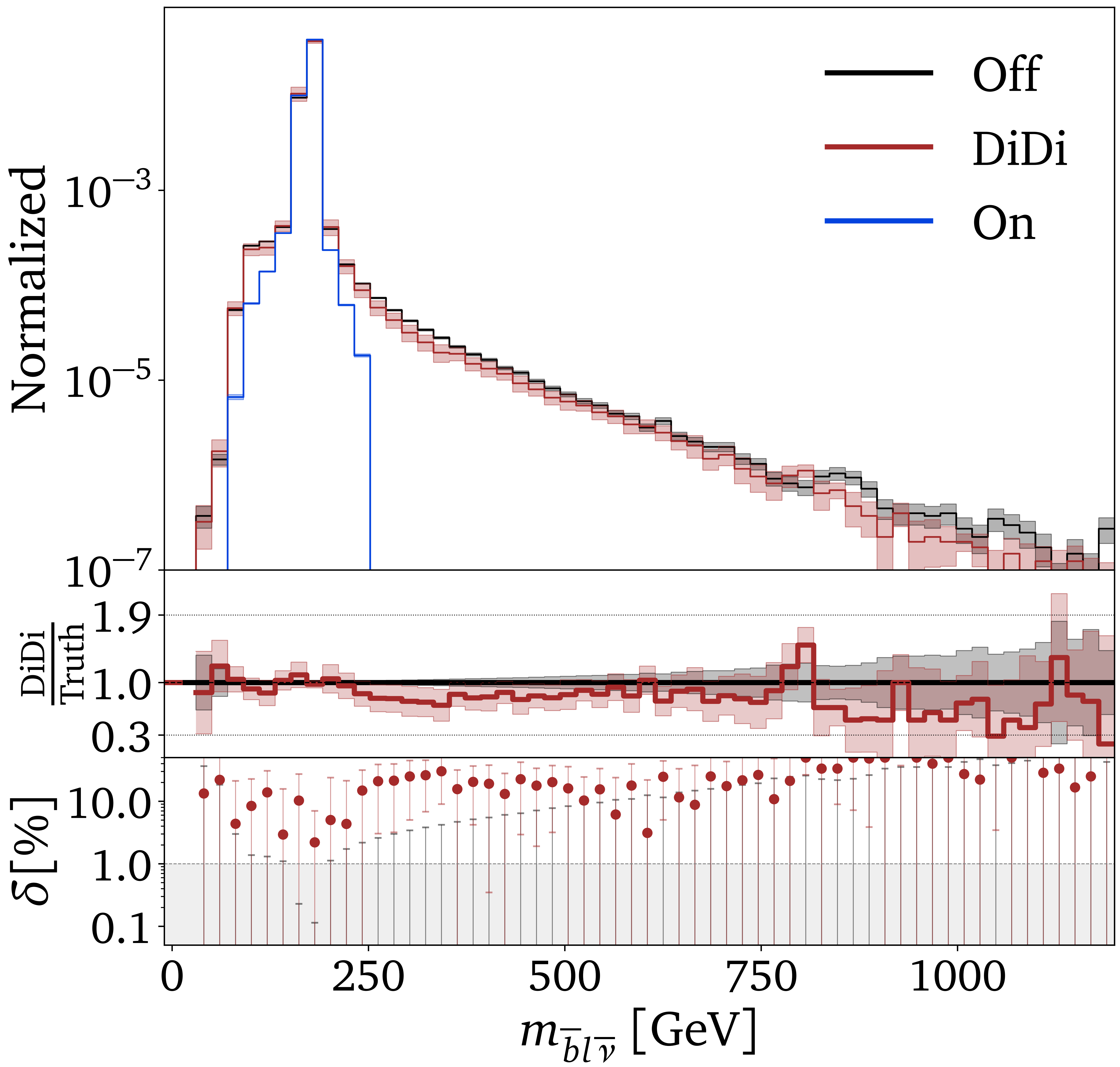
# Direct Diffusion (DiDi)




Reconstructed tbar mass learned to $\mathcal{O}(1\%)$ - $\mathcal{O}(10\%)$ precision

Truth value within our uncertainties

# Direct Diffusion (DiDi) — Tails

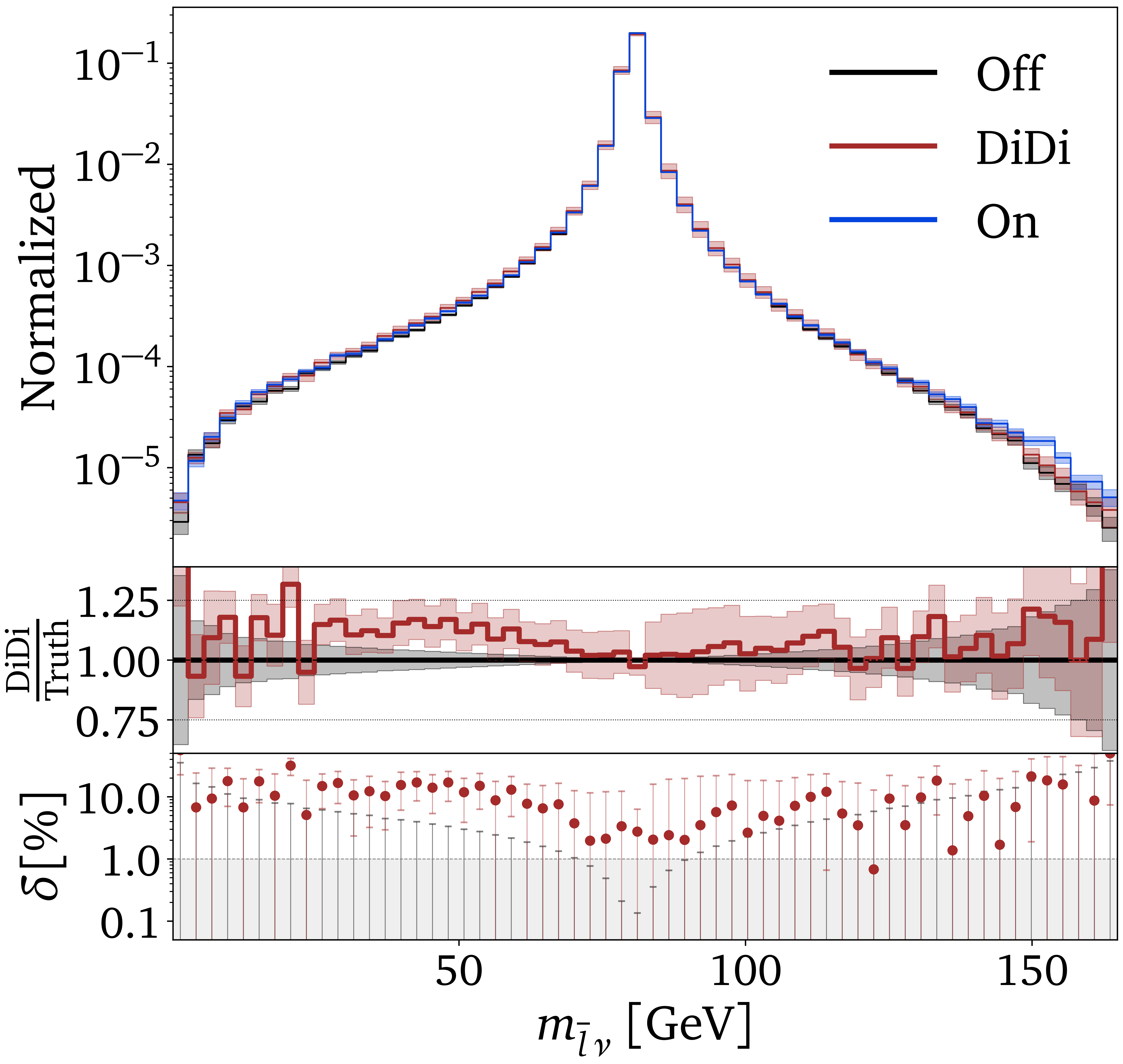



Reconstructed tbar mass learned to $\mathcal{O}(1\%)$ - $\mathcal{O}(10\%)$ precision

Truth value within our uncertainties

Also true for tails, with very little training statistic

# Direct Diffusion (DiDi)



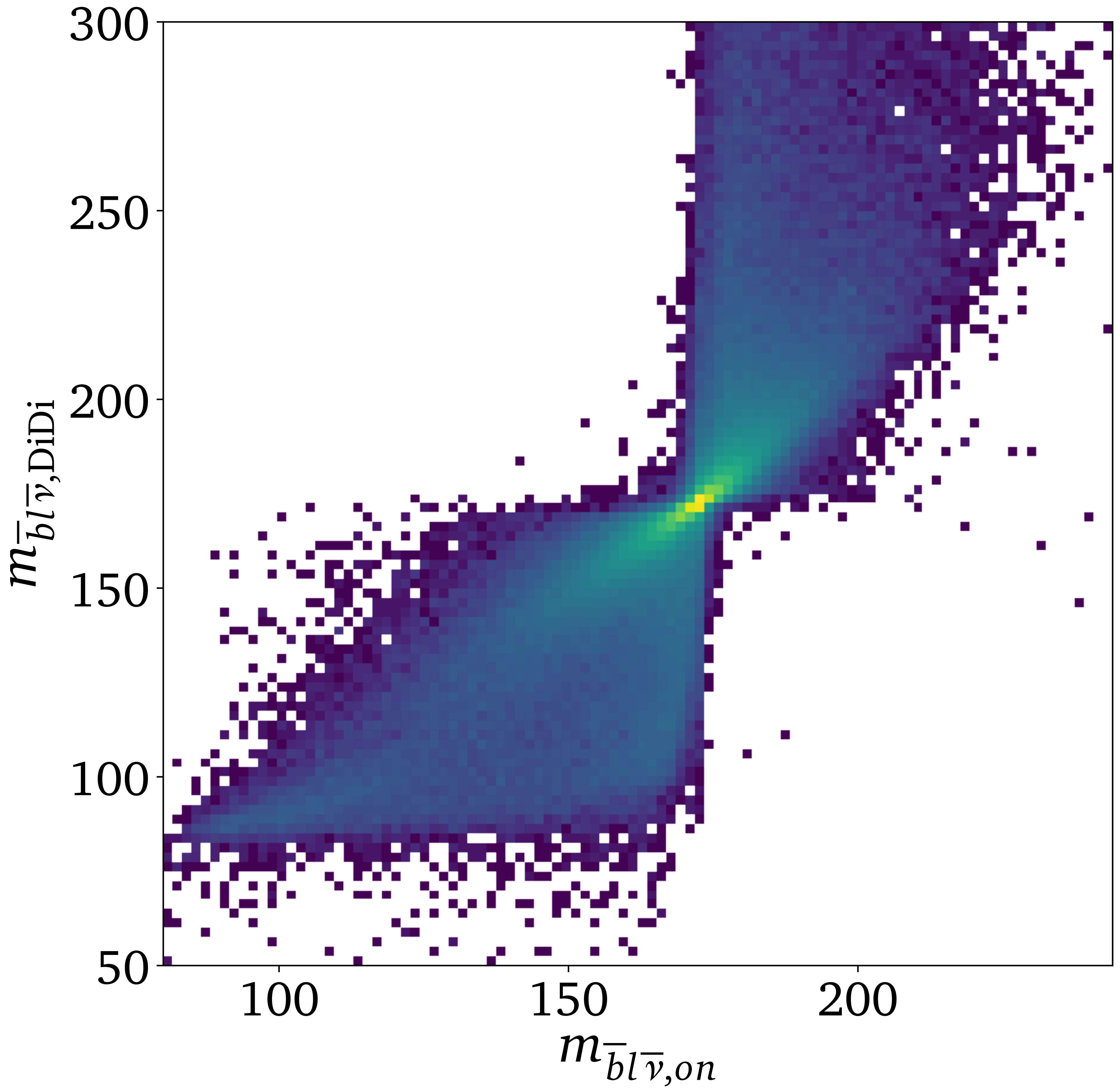

Reconstructed tbar mass learned to $\mathcal{O}(1\%)$ - $\mathcal{O}(10\%)$

Truth value within our uncertainties

Also true for tails, with very little training statistic

Multiresonant structure taken into account

# Direct Diffusion (DiDi) — Migration

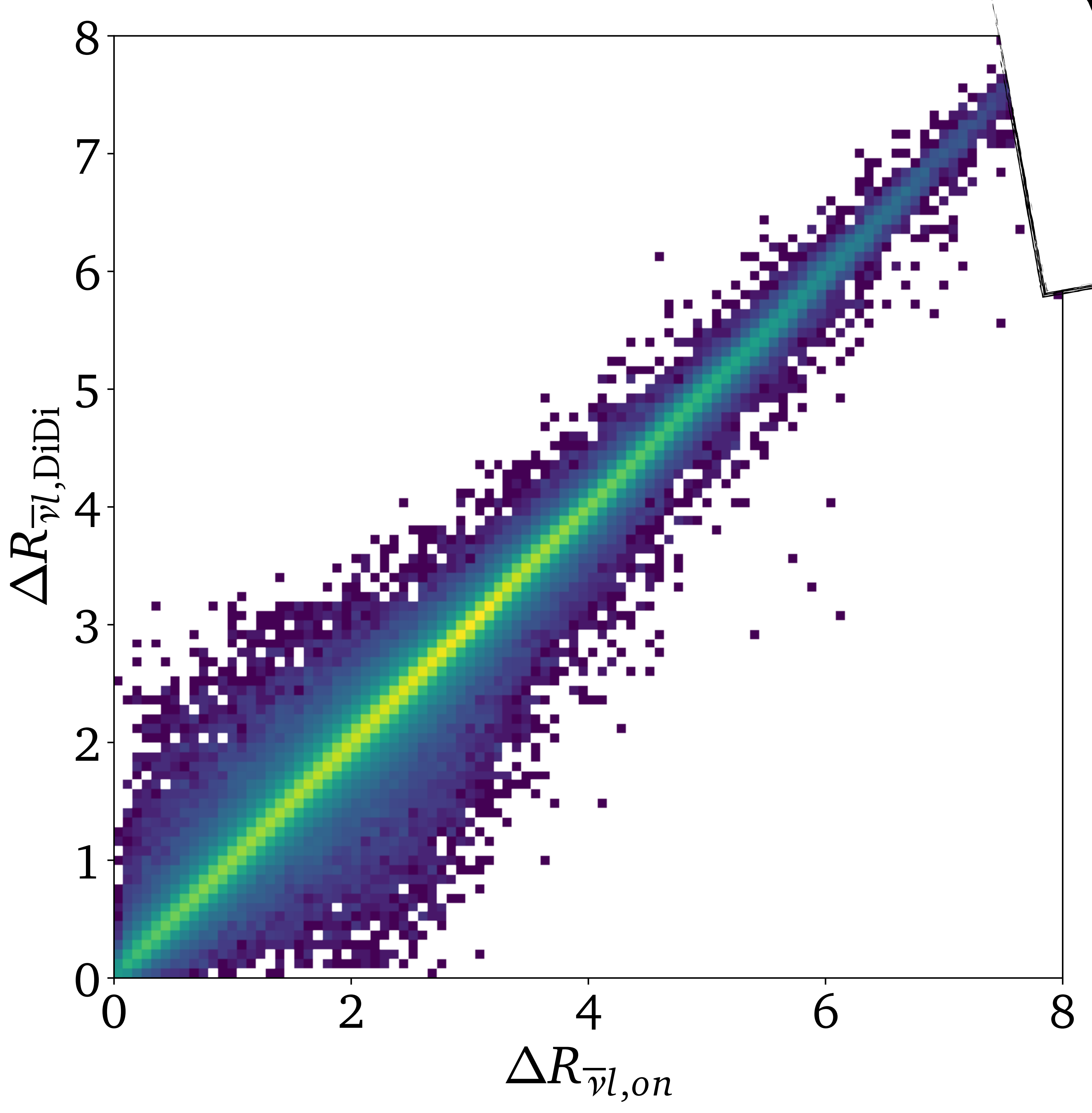



How does our mapping look like?

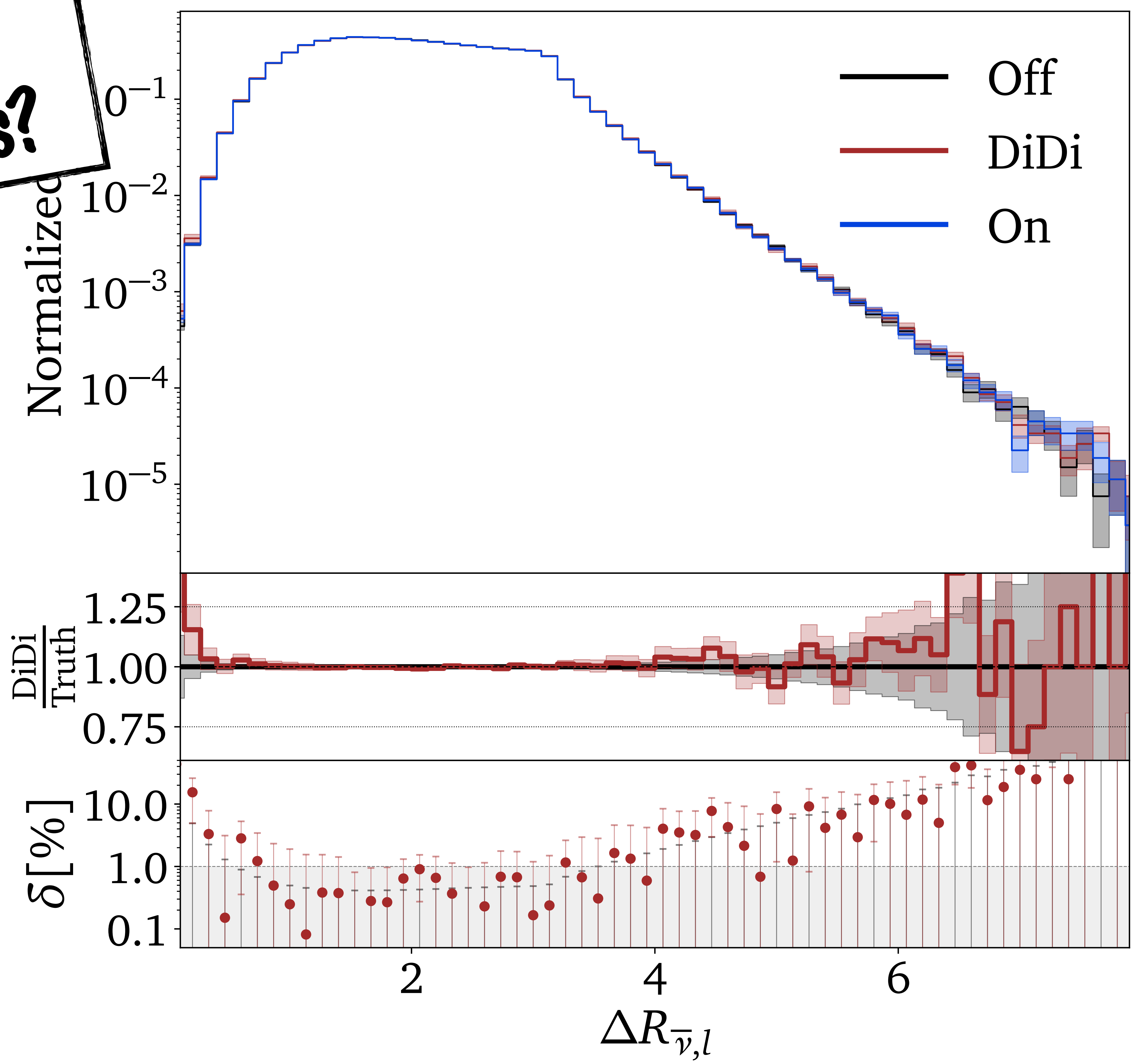
Biased to “optimal” transport (learned by DiDi)

No migration over peak

# Direct Diffusion (DiDi) — Staying put

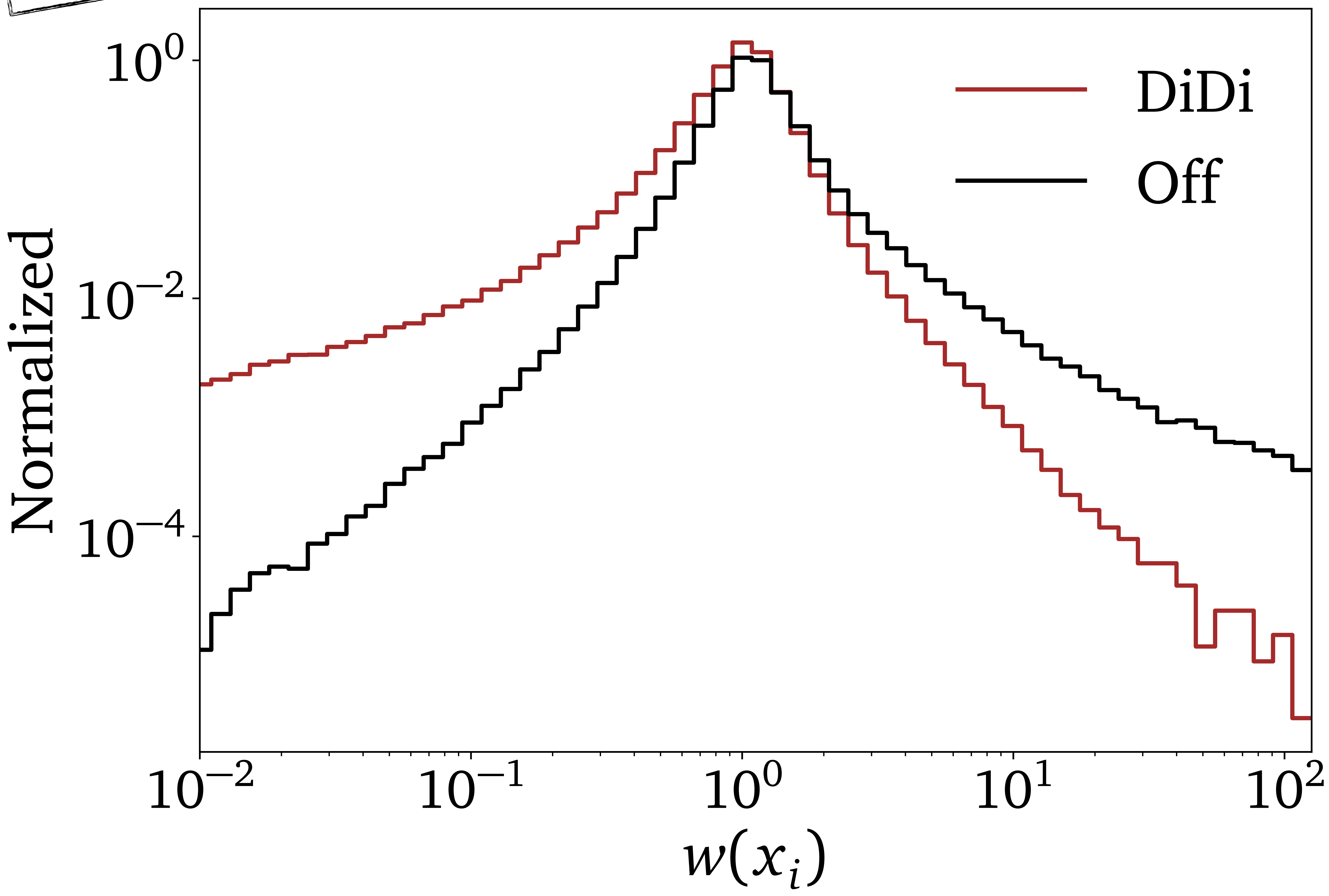



What about similar distributions?



# Understanding our Shortcomings

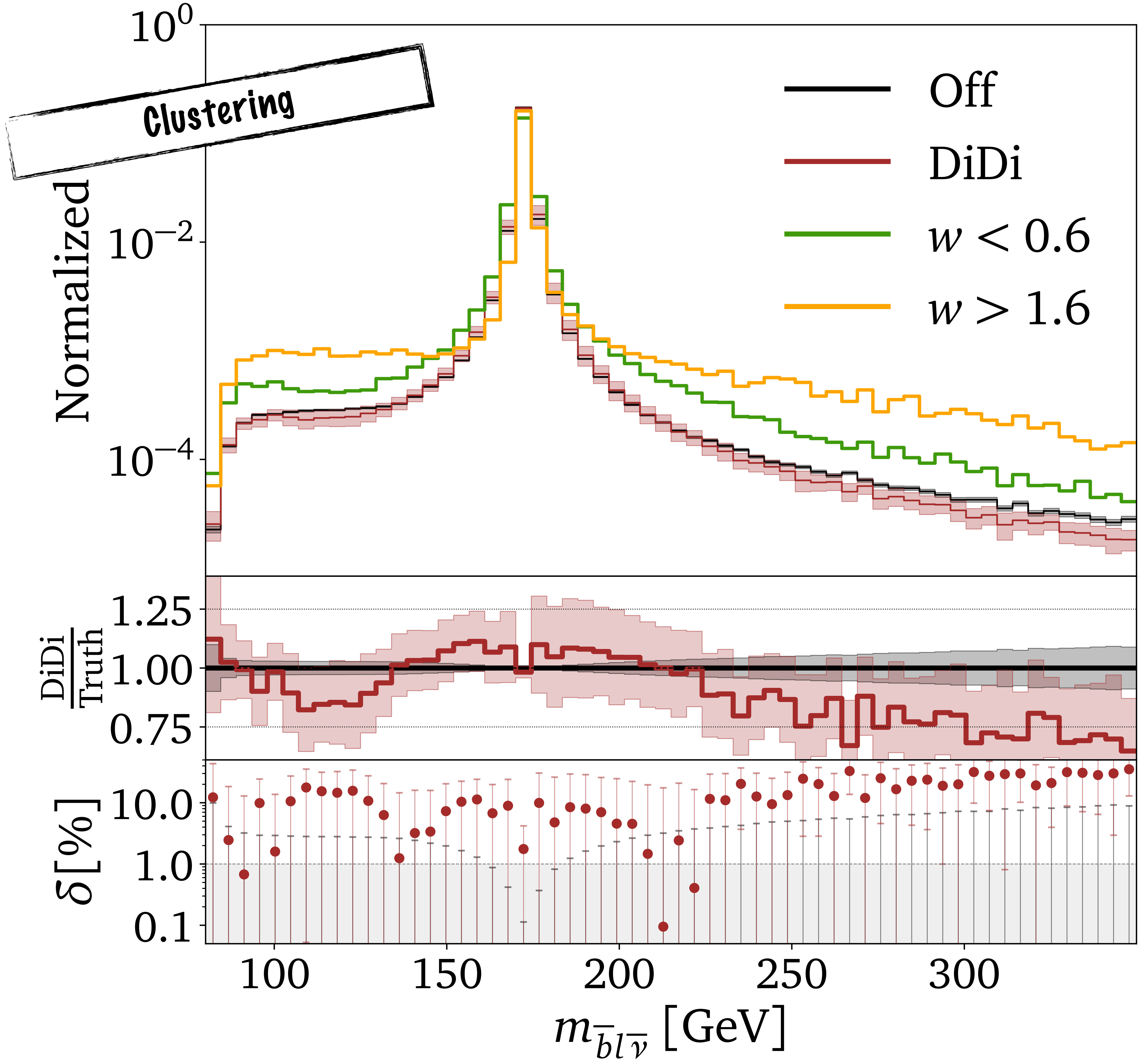
Performance




Train classifier to learn $w = \dfrac{p_{off}}{p_{gen}}$

Allows us to check performance, clustering and reweight distribution

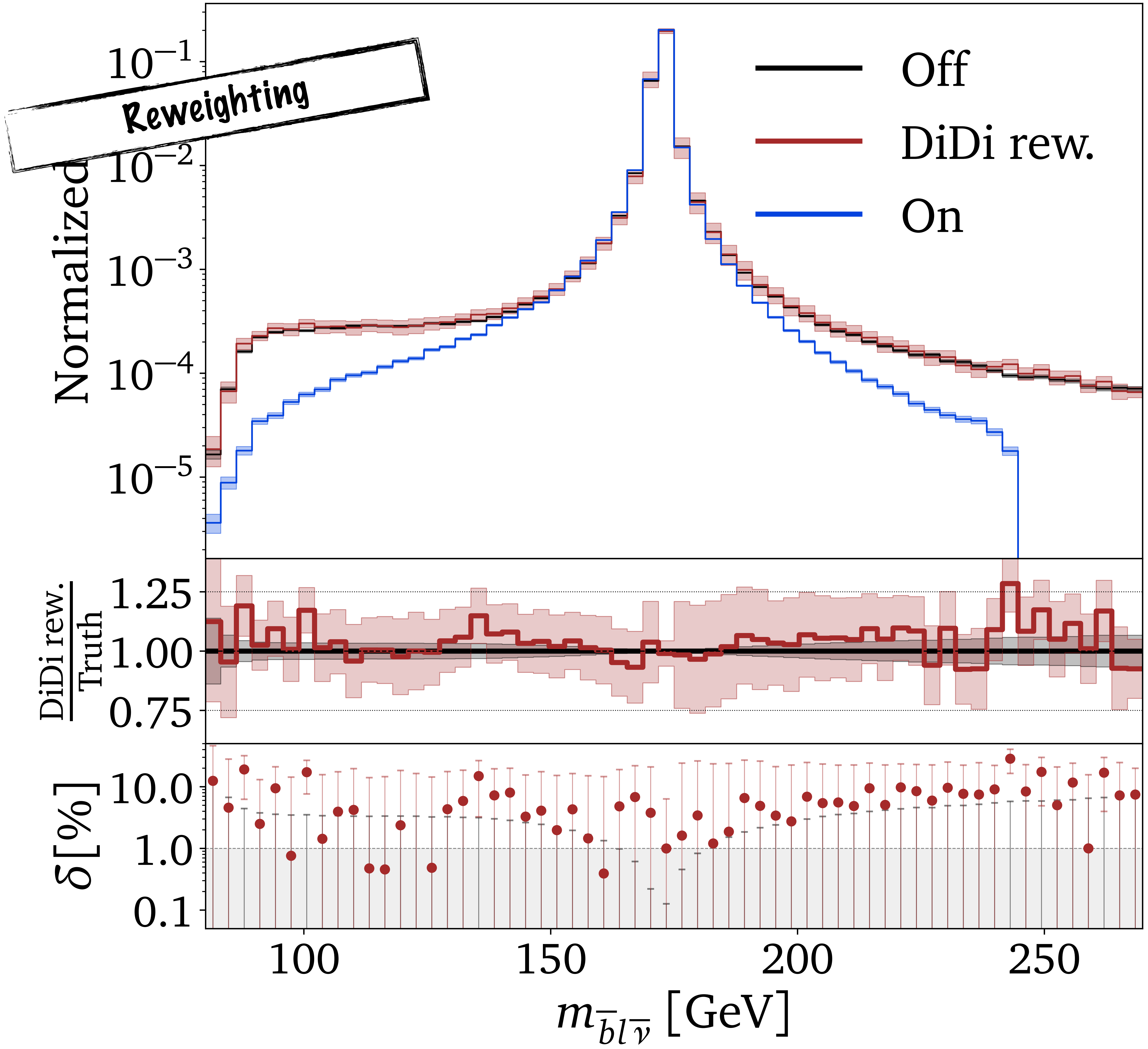
# Understanding our Shortcomings




Train classifier to learn $w = \frac{p_{off}}{p_{gen}}$

Allows us to check performance, clustering and reweight distribution

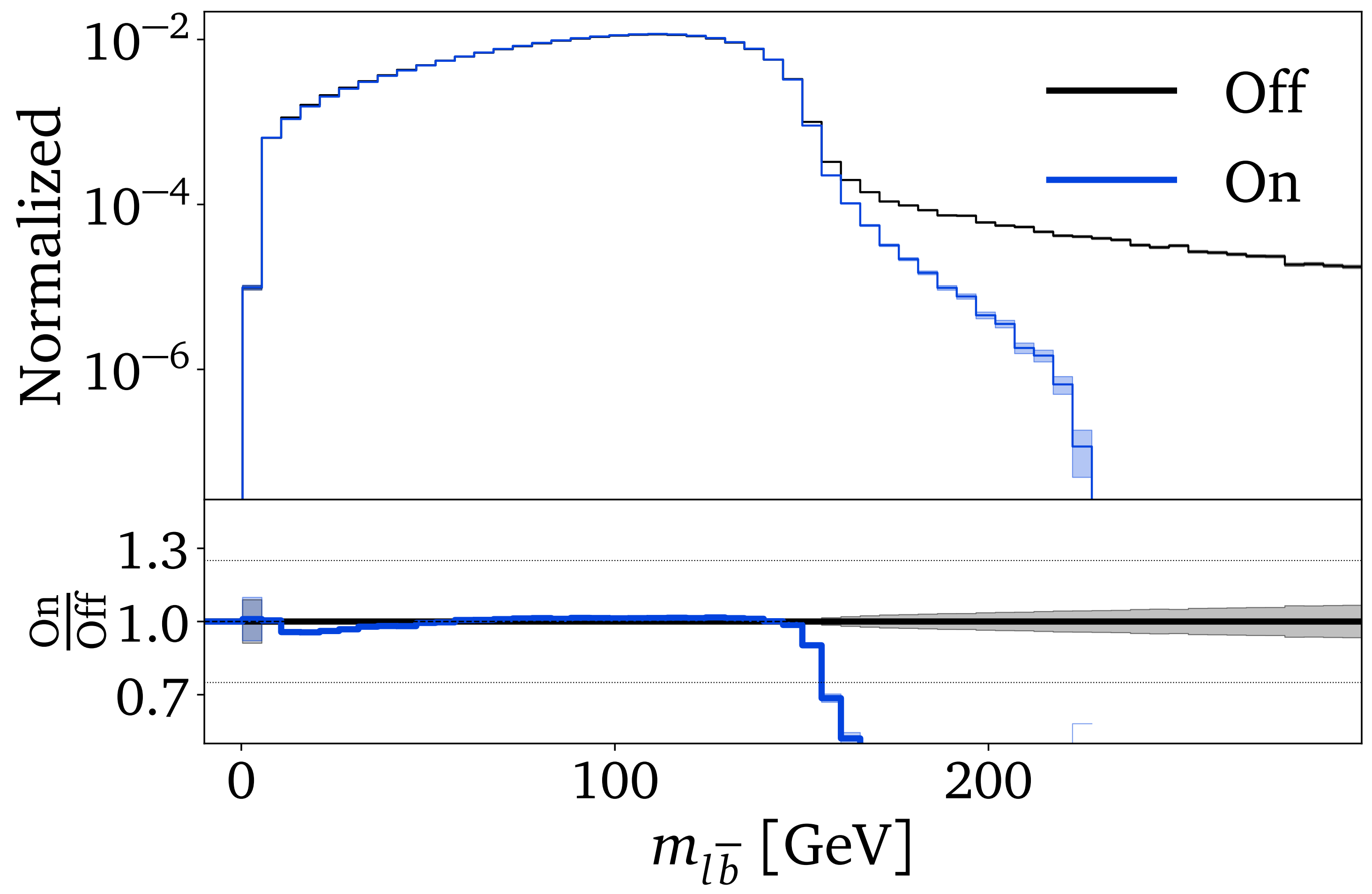
# Understanding our Shortcomings



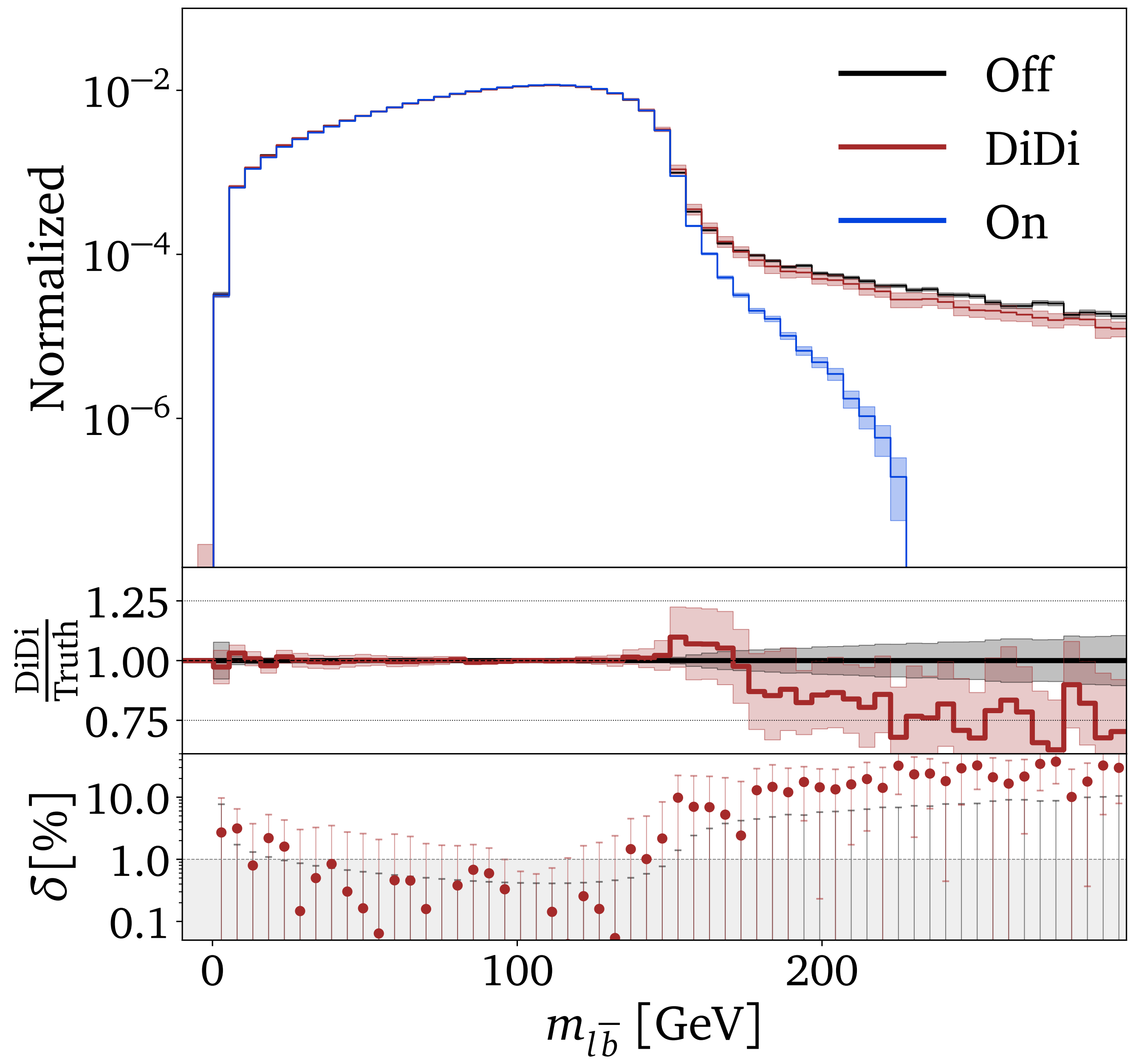

Train classifier to learn $w = \frac{p_{off}}{p_{gen}}$

Allows us to check performance, clustering and reweight distribution

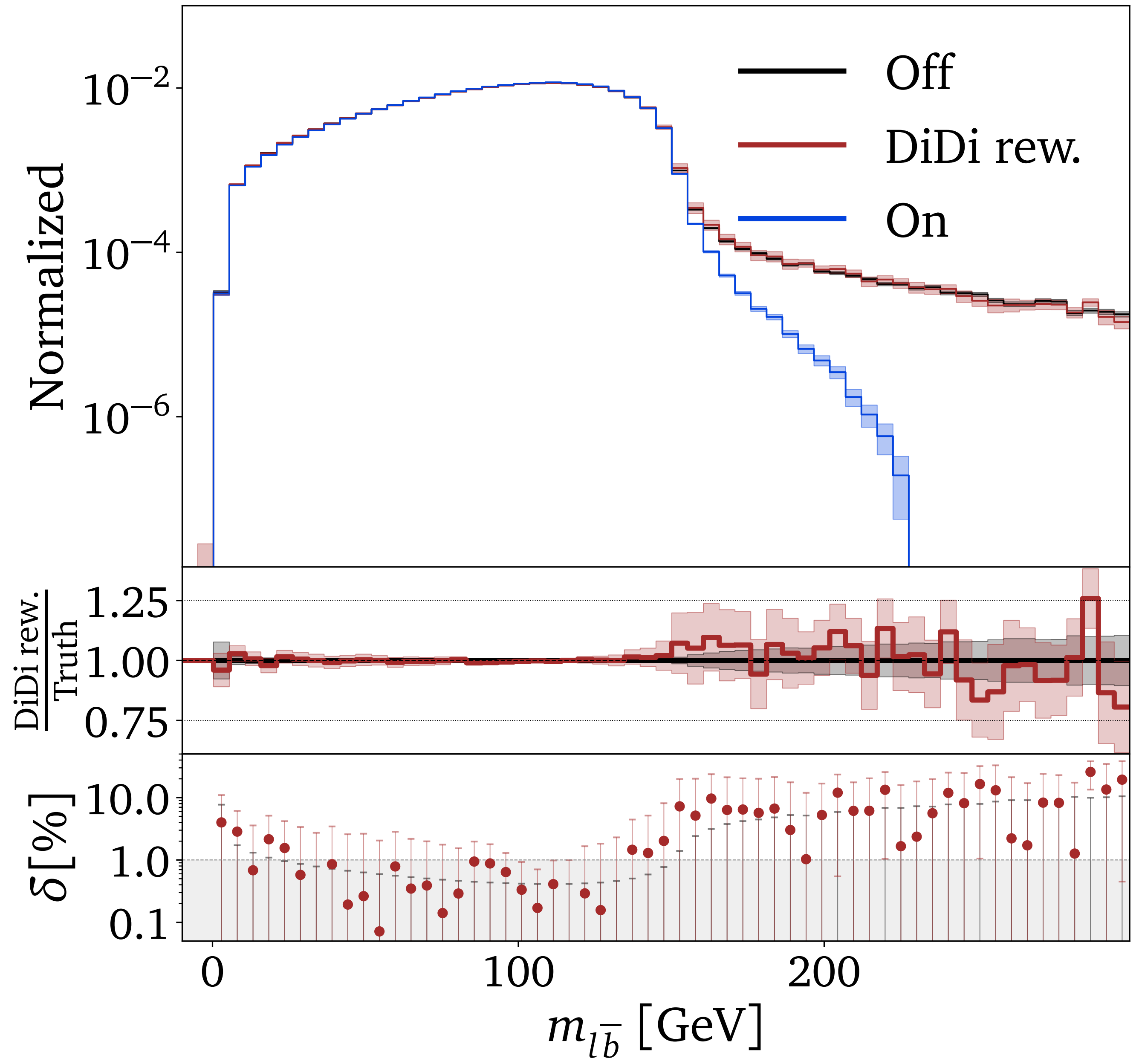
# Summary



Normalized
$10^{-2}$
$10^{-4}$
$10^{-6}$
Off
On
$\frac{\text{On}}{\text{Off}}$
1.3
1.0
0.7
0
100
200
$m_{l\bar{b}}$ [GeV]

Normalized
$10^{-2}$
$10^{-4}$
$10^{-6}$
Off
DiDi
On
$\frac{\text{DiDi}}{\text{Truth}}$
1.25
1.00
0.75
$\delta[\%]$
10.0
1.0
0.1
0
100
200
$m_{l\bar{b}}$ [GeV]

Normalized
$10^{-2}$
$10^{-4}$
$10^{-6}$
Off
DiDi rew.
On
$\frac{\text{DiDi rew.}}{\text{Truth}}$
1.25
1.00
0.75
$\delta[\%]$
10.0
1.0
0.1
0
100
200
$m_{l\bar{b}}$ [GeV]

# And now what?

Generative ML shows great potential to **speed-up** LHC event generation

For end-to-end generation the CFM reaches **state-of-the art precision**

Bayesian Versions seems to estimate **training uncertainty correctly**

Direct Diffusion allows to **morph two unknown, intractable distributions** onto each other

Successfully applied to **generate full off-shell distributions** from on-shell distributions