Signal model parameter scan using Normalizing Flow

29th March 2024

ISGC 2024

ICEPP/UTokyo[a], KEK[b]

Masahiko Saito[a], Masahiro Morinaga[a], Tomoe Kishimoto[b], Junichi Tanaka[a]

Why signal model parameter scan?

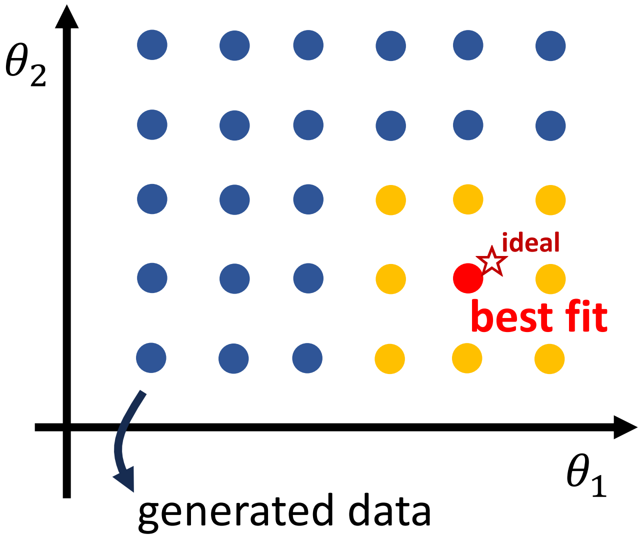
- No clues for beyond the Standard Model (BSM) in LHC experiments so far. **Comprehensive searches** are needed, rather than focusing on the limited phase space.
  - e.g. “anomaly detection”. But not assuming for a specific signal model results in a worse discovery sensitivity.
  - Searches that **assume** a specific signal model and cover the **entire phase space** → **model parameter scan**

- Some BSM (signal) models have **a lot of unpredictable model parameters.**
  - e.g. MSSM: > **100** parameters, pMSSM (reduced model): **19** parameters.
- Scanning such a large phase space is a challenging task.
  - Huge number of model parameter combinations
  - High computational cost to prepare a dataset of each BSM model parameter
    - Data processing (e.g. detector simulation) is a time-consuming task

Some methods for model parameter scan

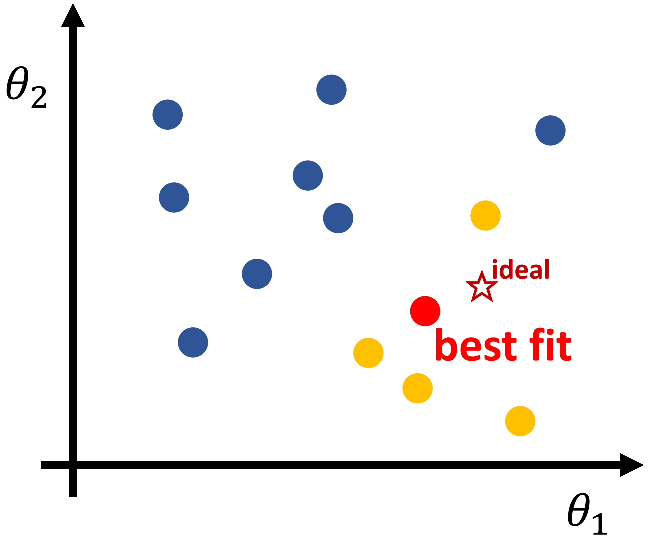
Grid scan

- Cutting space on a grid
- Good: Simple.
- Bad: **Curse of dimensionality**



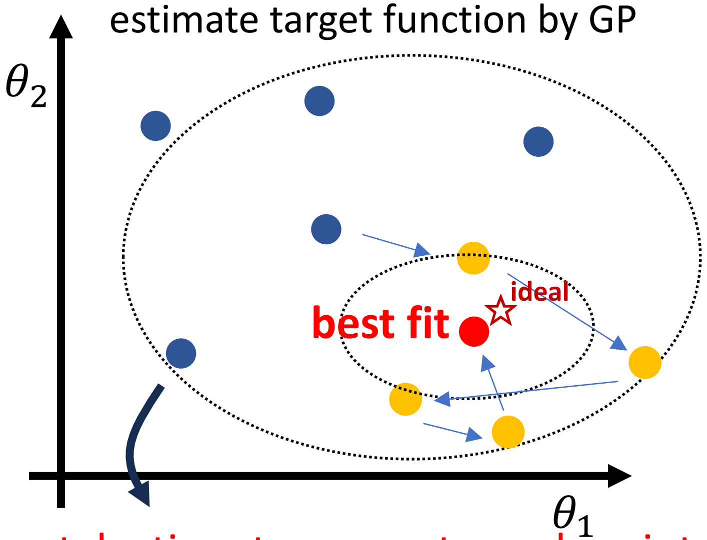
Random sampling

- Sample randomly from phase space
- Good: Easily extended to high dimensionality
- Bad: **Coarse** sampling.



Bayesian optimization

- Sample the next parameter based on the previous results
- Good: Efficient.
- Bad: Difficult to **parallelize** due to sequential evaluation.



Desired method: **flexible interpolation, fast search, tolerant to high-dimensional space**

Signal model parameter scan using Normalizing Flow

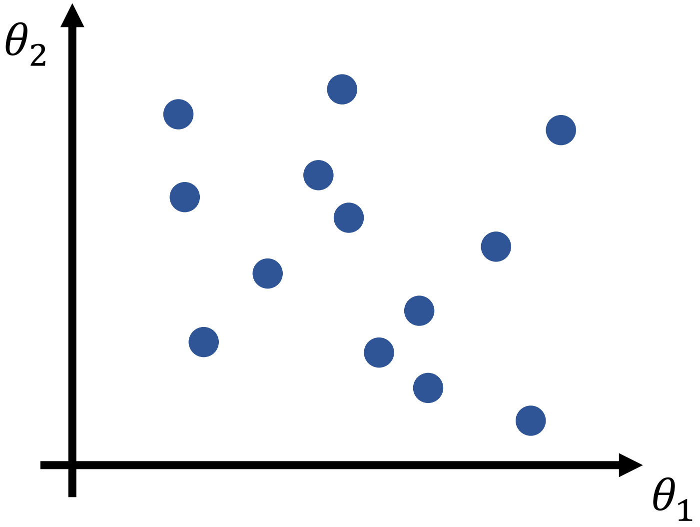
- We propose a new method using **conditional Normalizing Flow (NF)**
- Normalizing Flow is a kind of generative model
  - Model a probability density function ($p(x)$) from data ($\{x_i\}$)
  - Also model a conditional distribution ($p(x|\theta)$)

- We can use Normalizing Flow as
  - **Generator**: generate new events with unseen parameters fast ($x' \sim f_{\mathrm{NF}}(x|\theta)$)
  - **Evaluator**: evaluate a likelihood value for multidimensional observed data ($-\log f_{\mathrm{NF}}(x_{\mathrm{obs}}|\theta)$)
    - and can evaluate gradients for parameters $\nabla_\theta f_{\mathrm{NF}}(x_{\mathrm{obs}}|\theta)$ fast

→ **Optimizer**: **fast, efficient, and continuous** scan for model parameters using a **gradient-based optimization** (e.g., gradient descent), even when the model parameter space is **high dimensional**.

Workflow of parameter scan

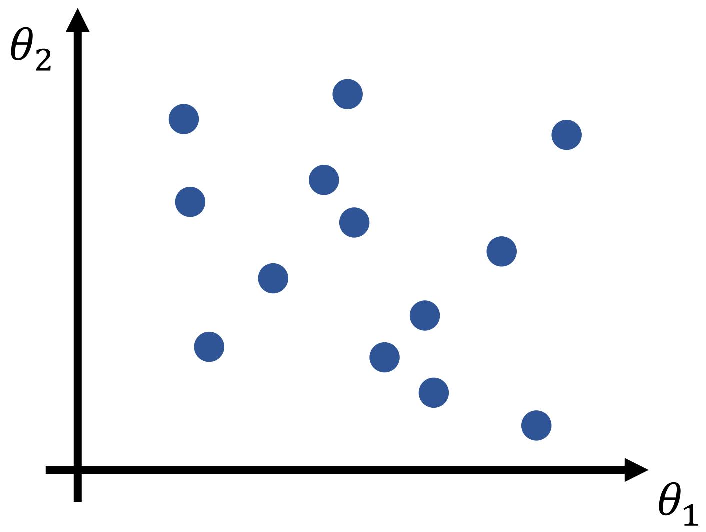
1. Generate simulation samples with various $\boldsymbol{\theta_{sig}}$



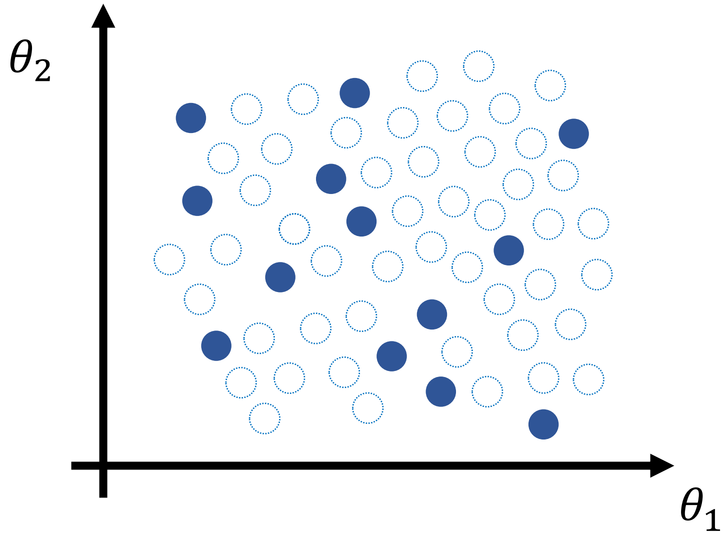
- This is the same as random sampling
- Simulation samples can be generated **in parallel**.

Workflow of parameter scan

1. Generate simulation samples with various $\boldsymbol{\theta_{sig}}$



2. Training NF models (Modeling)

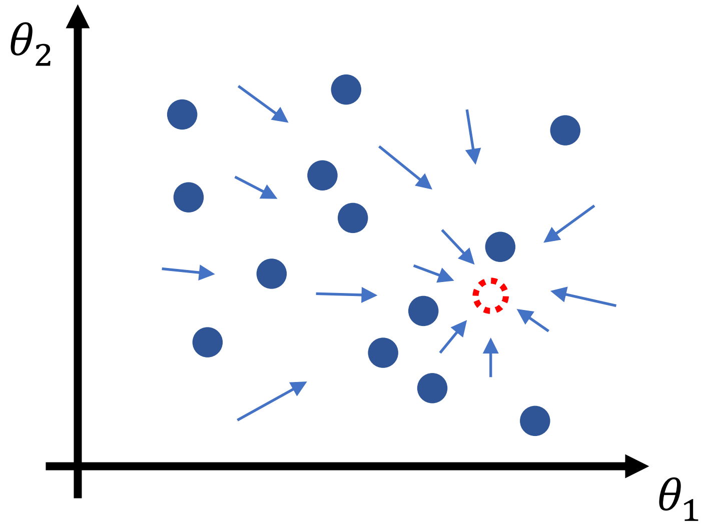


- This is the same as random sampling
- Simulation samples can be generated **in parallel**.

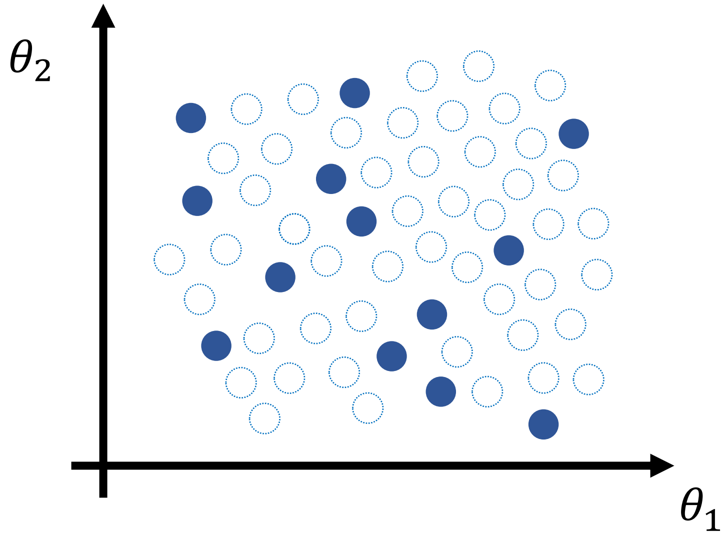
- Train a conditional Normalizing Flow (NF) $f_{\mathrm{NF}}(x|\theta)$ to **model the pdf for all samples**
- The NF can **interpolate in $\boldsymbol{\theta}$ phase space**

Workflow of parameter scan

3. Search for the optimal parameter



2. Training NF models (Modeling)

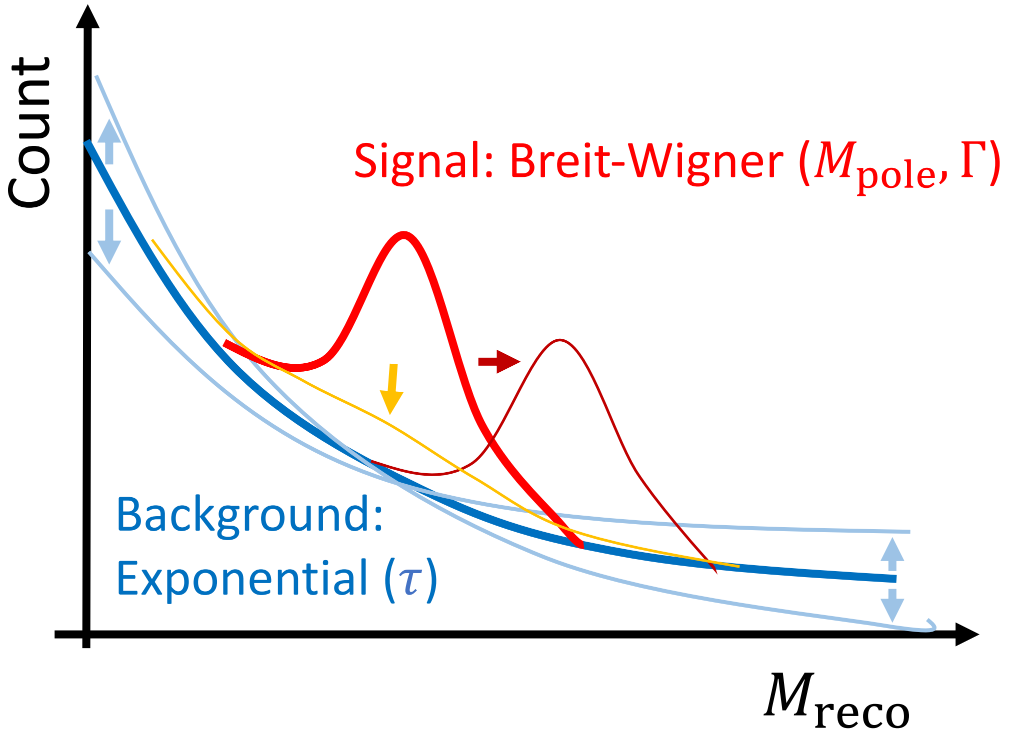


- The NF can **evaluate the NLL value fast** based on the modeled pdf ($f_{\mathrm{NF}}(x|\theta)$)
- Directly calculate **gradients for model parameters** $\nabla_{\theta_{\mathrm{sig}}} p\left(x_{\mathrm{data}}\middle|\theta_{\mathrm{sig}}\right)$ using backpropagation

- Train a conditional Normalizing Flow (NF) $f_{\mathrm{NF}}(x|\theta)$ to **model the pdf for all samples**
- The NF can **interpolate in $\boldsymbol{\theta}$ phase space**

Toy data (bump hunting)

Hunting a Breit-Wigner signal on the exponential background tail

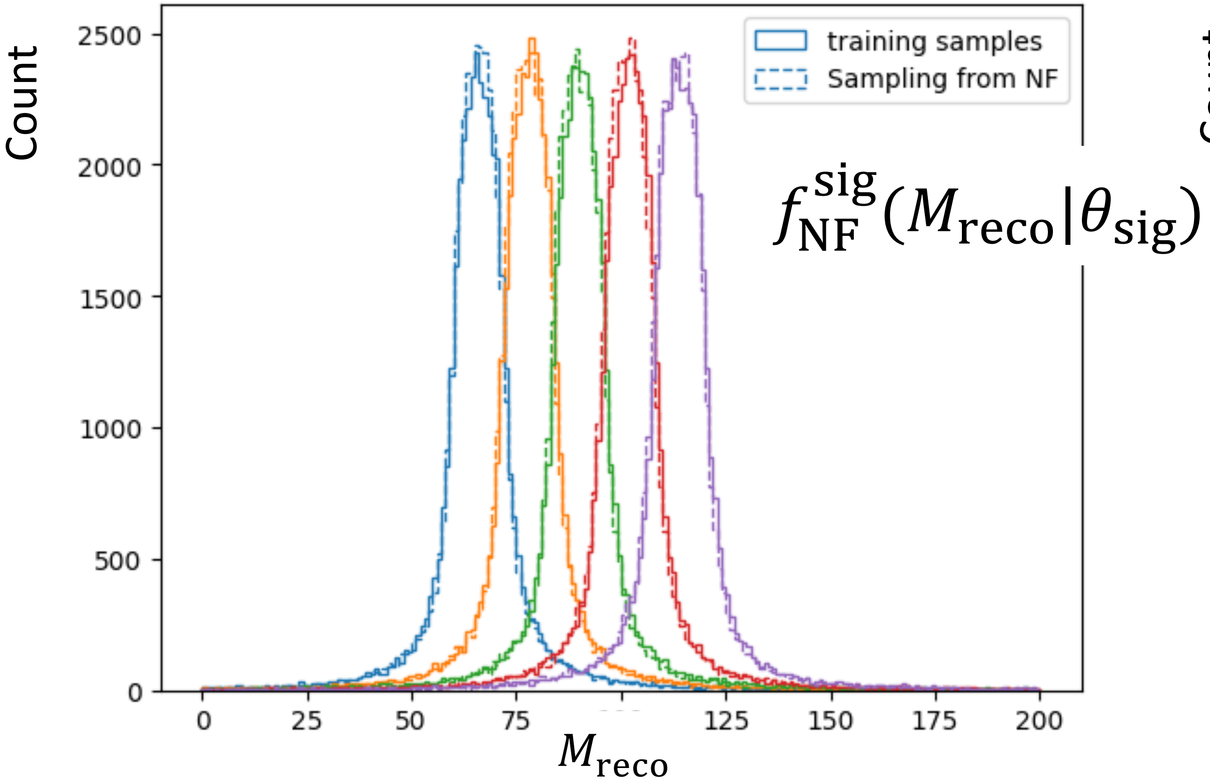


- Use analytic functions ($f_{\mathrm{sig/bg}}\left(M \middle| \theta_{\mathrm{sig/bg}}\right)$) for signal/background pdf
  - $\theta_{\mathrm{sig}} = \{M_{\mathrm{pole}}, \Gamma\}$, $\theta_{\mathrm{bg}} = \{\tau\}$
- Training samples: 200k events with randomly sampled from model parameter space $\theta_{\mathrm{sig}} \in \Theta_{\mathrm{sig}}$, $\theta_{\mathrm{bg}} \in \Theta_{\mathrm{bg}}$
- Pseudo dataset for parameter scan
  - fixed (θ'_{sig}, θ'_{bg}) (unknown.)
  - Signal 1k events, background 100k events

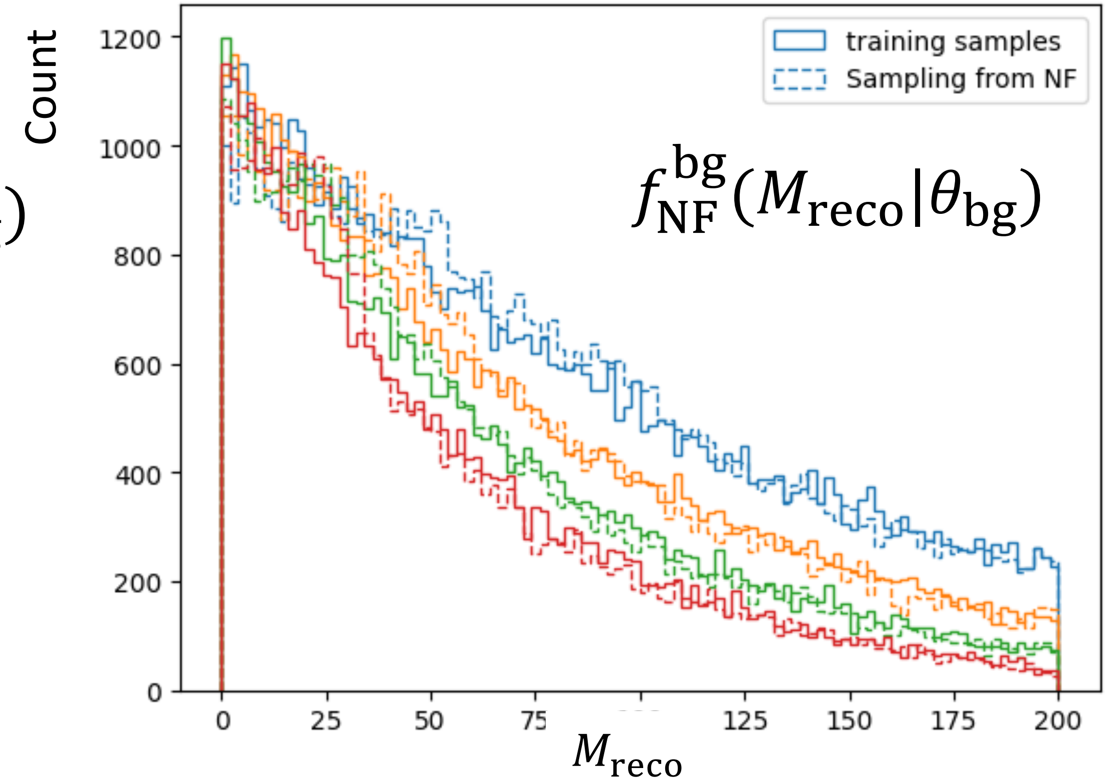
Toy data: Training a Normalizing Flow Model

- Spline Flows (1906.04032) is used as a Normalizing Flow (NF) model
  - Use rational-quadratic splines in each transformation step
  - Flexible modeling capabilities.
- Train two NF models for signal and background, respectively

Signal distribution



Background distribution



Toy data: Parameter scan

- Define likelihood

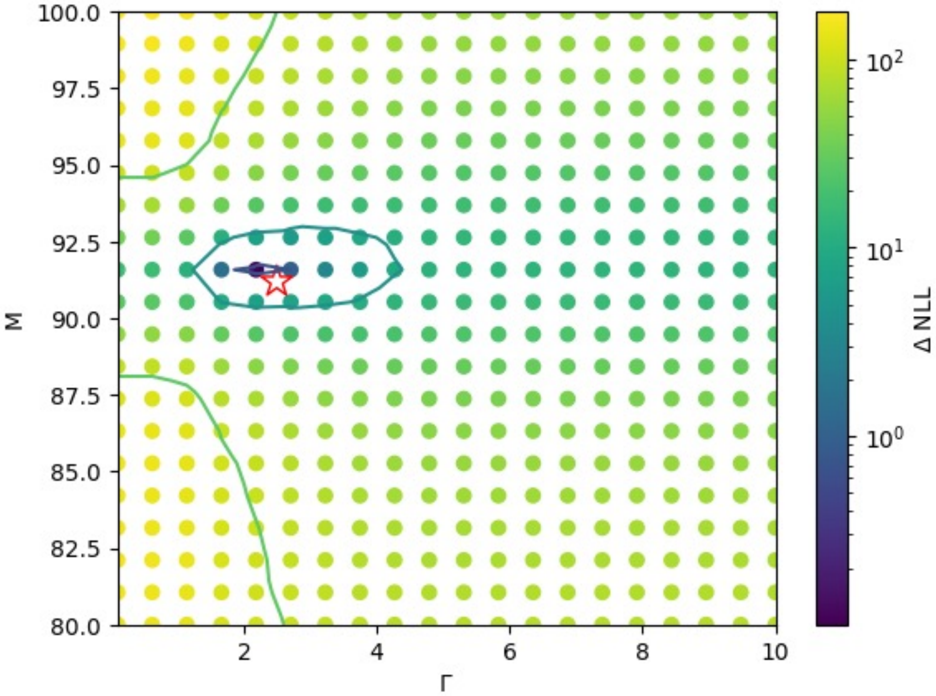
$$p(x|\theta) = \frac{n_{\text{sig}}}{n_{\text{sig}} + n_{\text{bg}}} \cdot \underbrace{f_{\text{NF}}^{\text{sig}}\left(M_{\text{reco}}^{\text{obs}} \mid M_{\text{pole}}, \Gamma\right)}_{\text{evaluated by NF fast}} + \frac{n_{\text{bg}}}{n_{\text{sig}} + n_{\text{bg}}} \cdot \underbrace{f_{\text{NF}}^{\text{bg}}\left(M_{\text{reco}}^{\text{obs}} \middle| \tau\right)}_{\text{evaluated by NF fast}}$$

- Evaluate the sum of NLL ($-\sum_i \log\ p(x_i^{\text{obs}}|\theta)$) for the pseudo dataset ($\{x_i^{\text{obs}}\}$) for any model parameter (θ).

True θ (used in pseudo data generation)

NLL value for $M_{\text{reco}}^{\text{obs}}$ and each θ

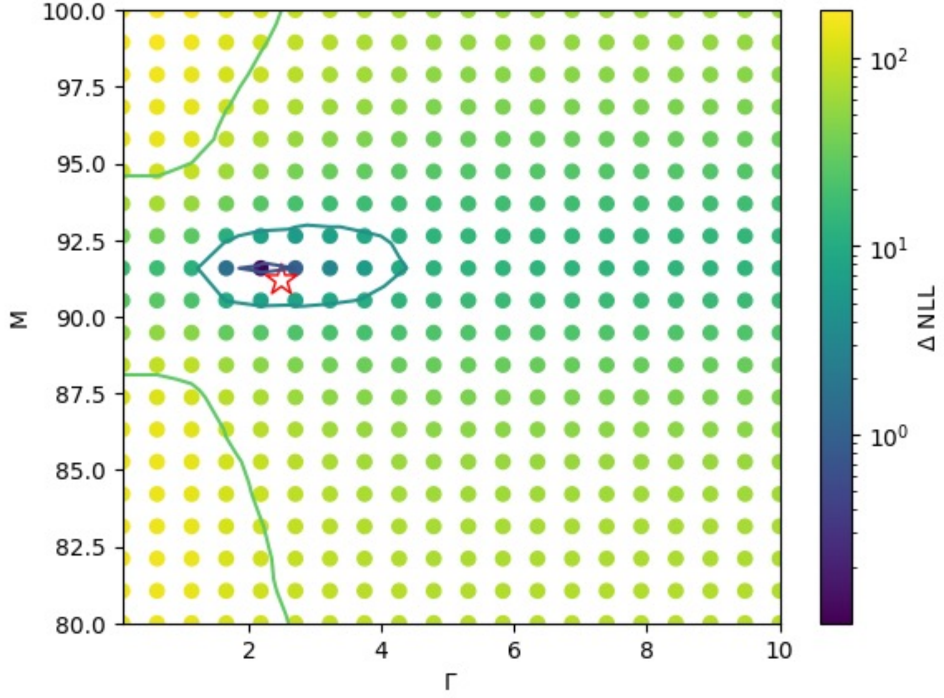
NLL ($\log p(M_{\text{reco}}^{\text{obs}}|\theta_{\text{sig}}, \theta_{\text{bg}})$)



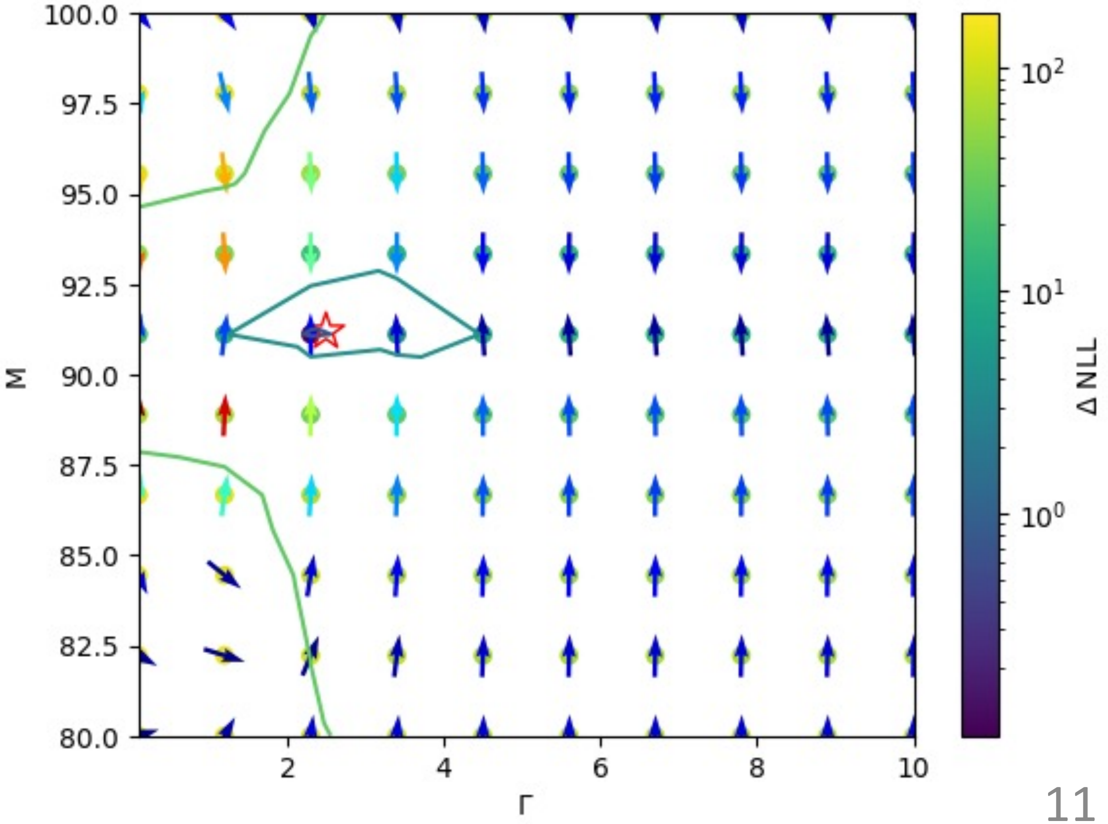
Toy data: Parameter scan

- NLL ($\log p(x_i|\theta)$) is differentiable for model parameters (θ)
  - Fast gradient evaluation ($-\nabla_\theta \log p(x_i|\theta)$)
  - Can use gradients for efficient parameter scan

NLL ($-\log p(M_{\mathrm{reco}}|\theta_{\mathrm{sig}}, \theta_{\mathrm{bg}})$)



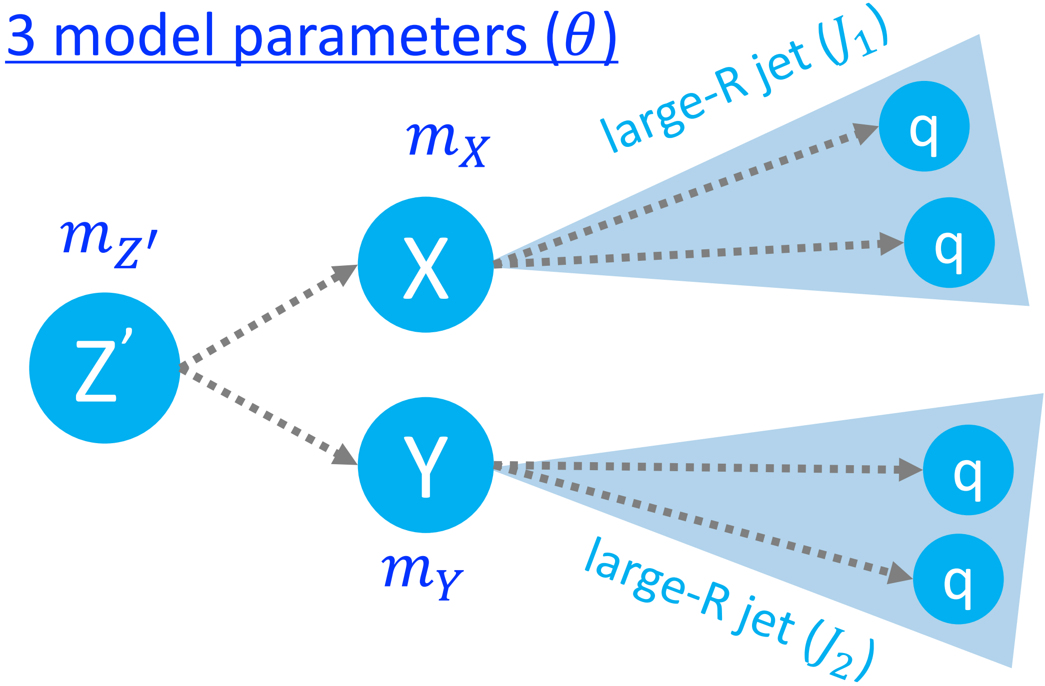
Gradient for model parameters calculating from NF models



More practical data: LHC Olympic 2020 (LHCO) Dataset

- LHC Olympic 2020 (2101.08320) benchmark dataset
  - Signal: $Z' \to XY \to qqqq$ (2 large-R jets), Background: QCD di-jet
  - (R&D) dataset parameter: $((m_{Z'}, m_X, m_Y) = (3500, 500, 100)$ GeV)
  - Enhanced signal samples with various signal parameters for this study
- Input features according to the ANODE paper (2001.04990)

3 model parameters (θ)



5 input (high-level) features (observable, x)

$m_{J_1} (\approx m_X)$ (jet mass)

$\tau_{J_1,21}$ (jet substructure variables)

$m_{J_1 J_2}$ (di-jet mass)

$m_{J_1} - m_{J_2} (\approx m_X - m_Y)$ (jet mass)

$\tau_{J_2,21}$ (jet substructure variables)

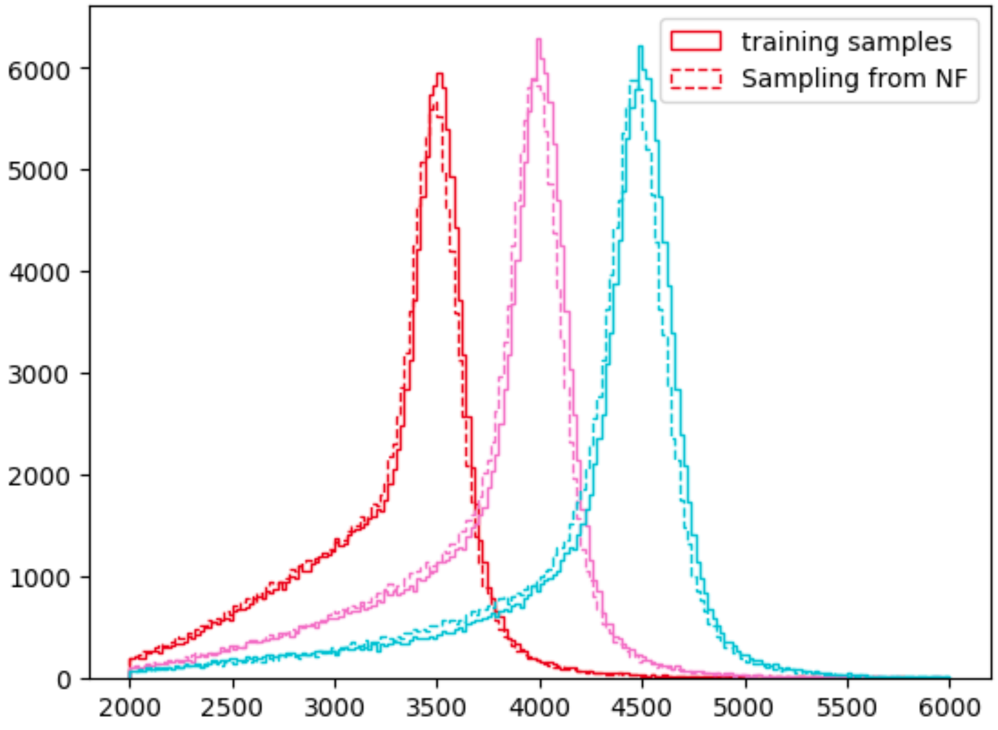
LHCO: Training samples and NF models

- Training samples (enhanced for this study)
  - $m_{Z'}$: [3000, 3500, 4000, 4500] GeV
  - m_X: [250, 500, 750, 1000] GeV
  - m_Y: [50, 100, 150, 200, 250, 300] GeV

    (where satisfied with “$m_{Z'} - m_X > 1000\ \mathrm{GeV}$” and “$m_X - m_Y > 100\ \mathrm{GeV}$”)
  - Total number of signals: 583k, background: 91k
- Preprocessing:
  - Input features are **linearly** normalized in the range 0 ~ 1.
- Normalizing Flow model configuration
  - Masked Autoregressive Flow (MAF) is used.
  - Two normalizing flow models for signal/background
    - NF for signal: 3 conditional parameters ($m_{Z'}, m_X, m_Y$)
    - NF for background: no conditional parameters

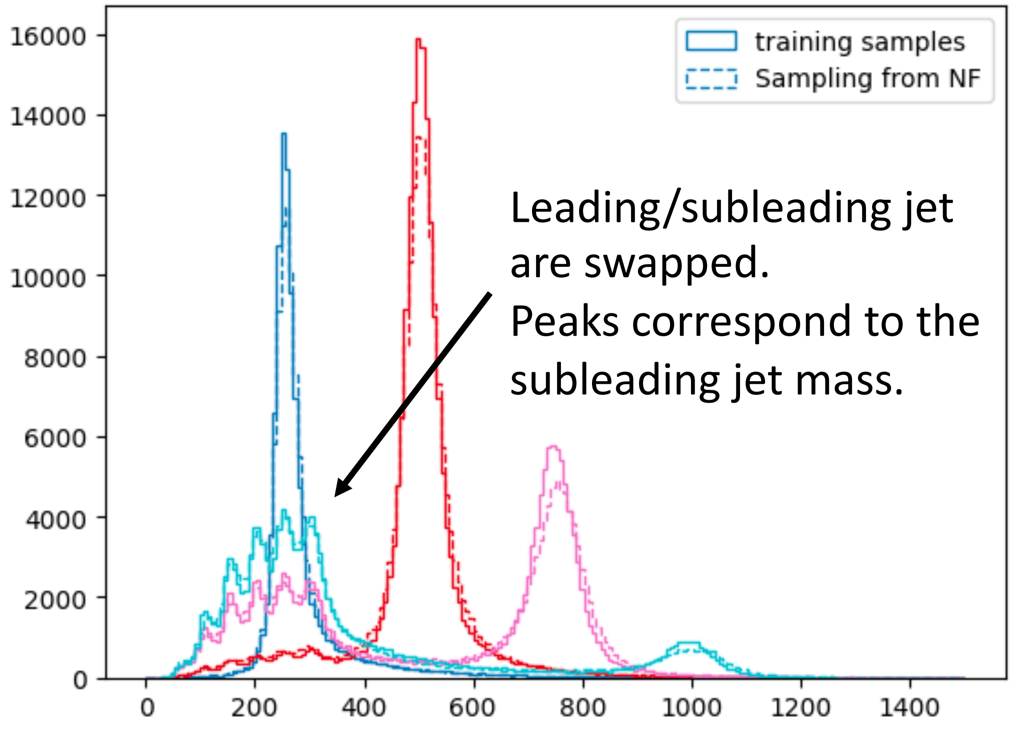
LHCO Dataset: Training the signal NF model

Di large-R jets mass (~Z' mass)



$m_{J_1J_2}$

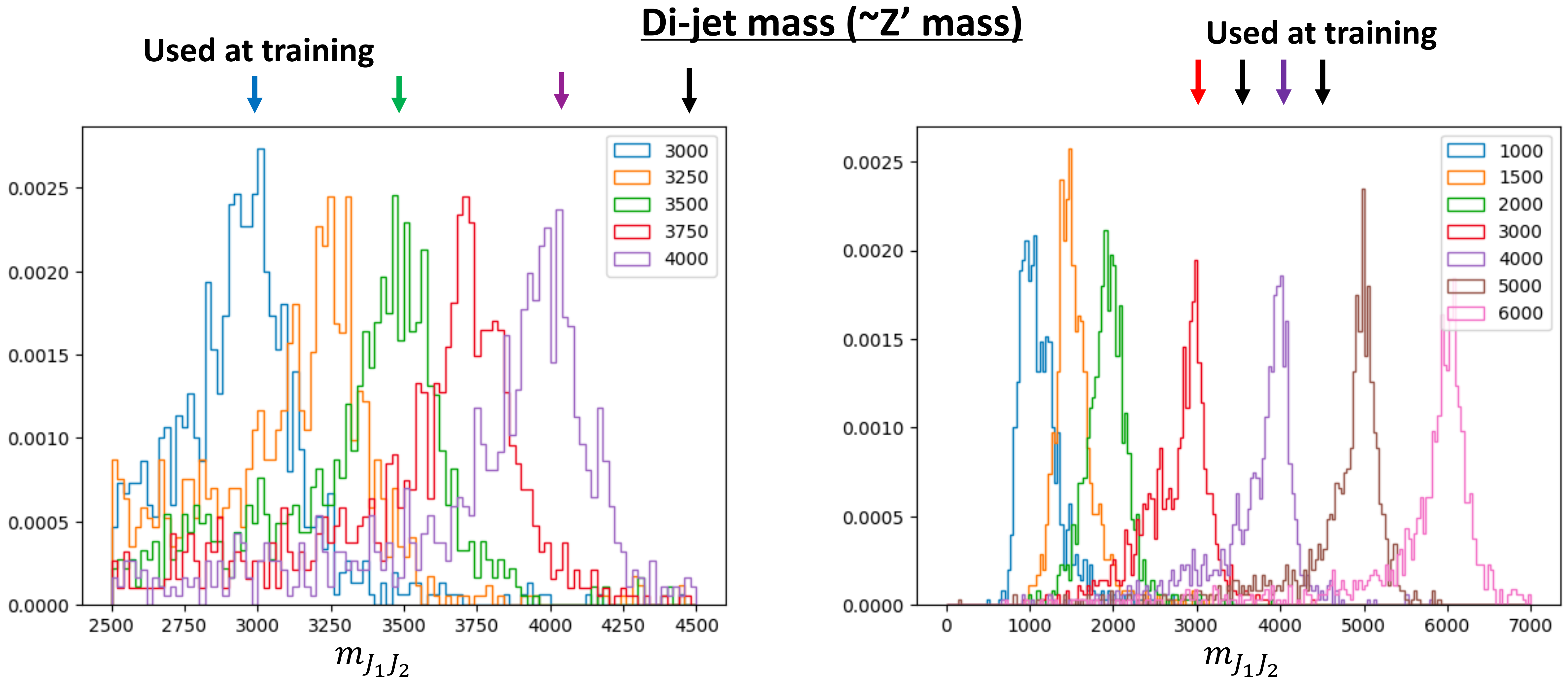
Leading large-R jets mass (~X mass)



$m_{J_1} (\approx m_X)$

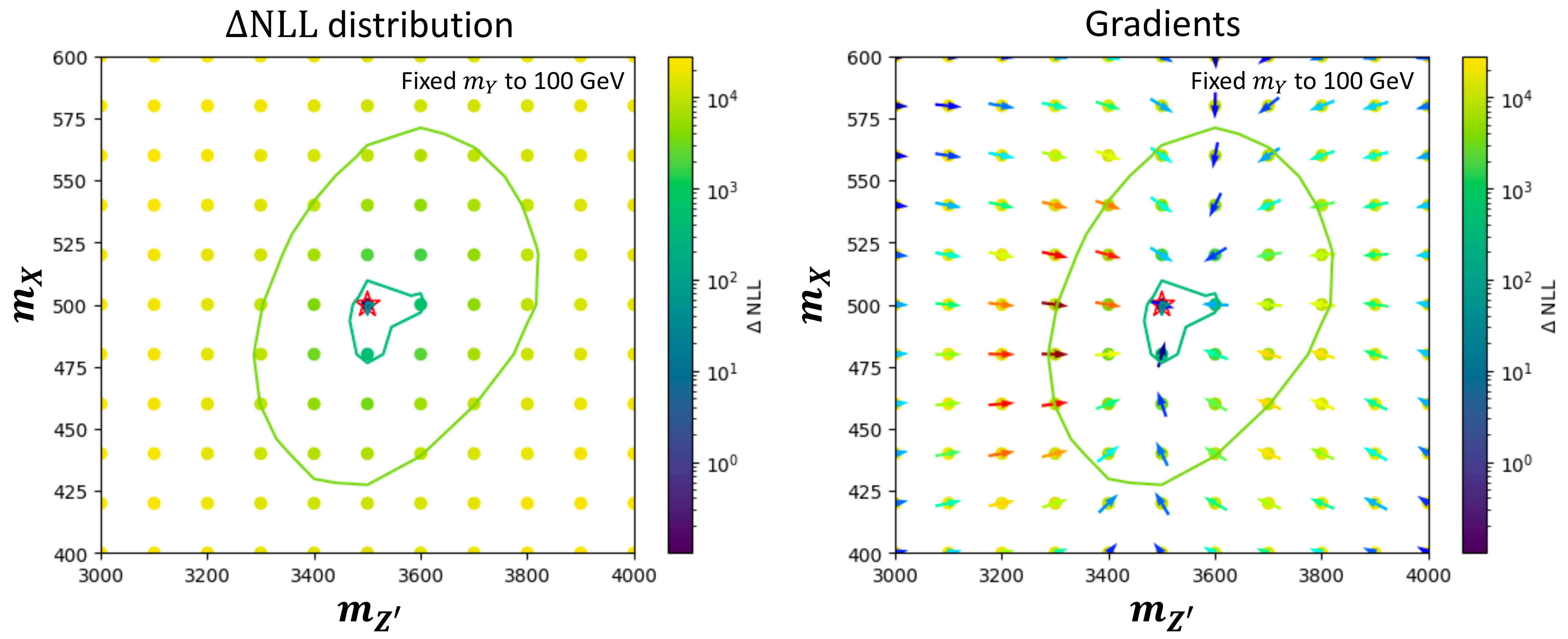
- Now 5 observables (x) and 3 signal model parameters (θ)
- Normalizing Flow model works well for higher dimensional observables/parameters, even if the pdf has an irregular structure.

LHCO Dataset: Interpolation capability



- Samples generated by NF models trained with Z′ mass = [3000, 3500, 4000, 4500] GeV
- Good interpolation and extrapolation capabilities.

LHCO Dataset: Parameter scan



- NLL and gradient evaluation work well for higher dimensional observables/parameters and complex pdfs.

Summary and future work

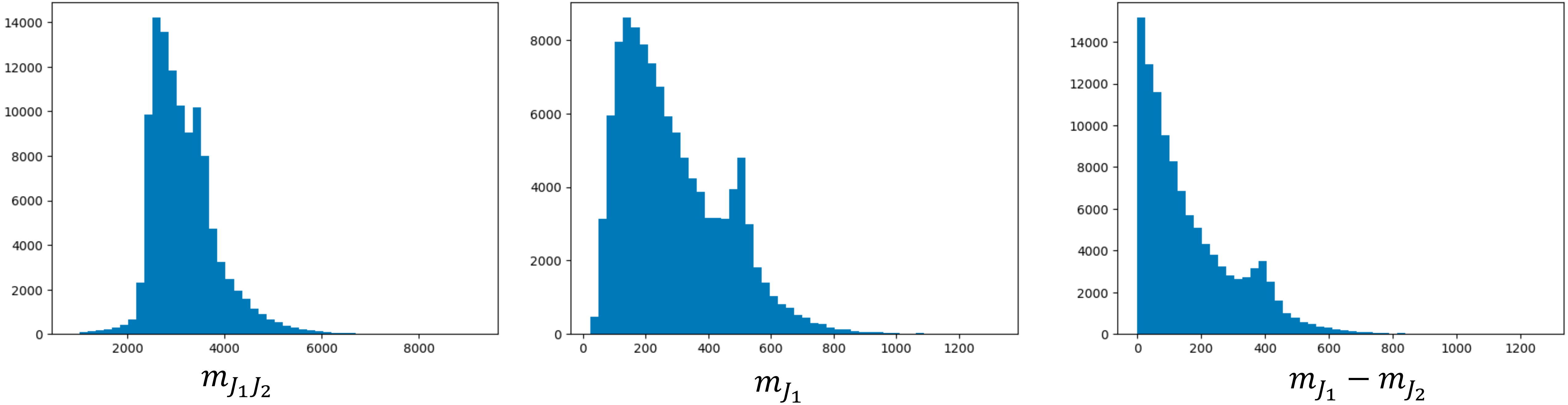
- Proposed an efficient signal model parameter scan technique based on Normalizing Flow
- Demonstrated this technique on toy data and LHC Olympic 2020 benchmark dataset
  - The Normalizing Flow model has good capabilities for modeling complex distributions and interpolating signal model parameter spaces.
  - Gradient of NLL can be evaluated fast.

- Future work
  - Extend to higher dimensional data
    - higher dimensional observables (e.g. set of particle four-vectors)
    - higher dimensional model parameters (e.g. pMSSM, 19 parameters)

Backup

LHCO Dataset: Pseudo dataset and NLL

Pseudo dataset bootstrapped from (Z', X, Y) = (3500, 500, 100) GeV, signal: 10k, Background 100k



PDF

$$p(x|\theta) = \frac{n_{\mathrm{sig}}}{n_{\mathrm{sig}} + n_{\mathrm{bg}}} \cdot f_{\mathrm{NF}}^{\mathrm{sig}}(x \mid M_{Z'}, M_X, M_Y) + \frac{n_{\mathrm{bg}}}{n_{\mathrm{sig}} + n_{\mathrm{bg}}} \cdot f_{\mathrm{NF}}^{\mathrm{bg}}(x)$$ (Fixed $n_{\mathrm{sig}}, n_{\mathrm{bg}}$)

- Find the optimal signal parameter (θ_{best}) that best describes the pseudo dataset ($\{x_i\}$) based on NLL ($-\sum_i \log p(x_i|\theta)$)