Model-independence in machine-learning-based mass reconstruction for inclusive multi-jet search

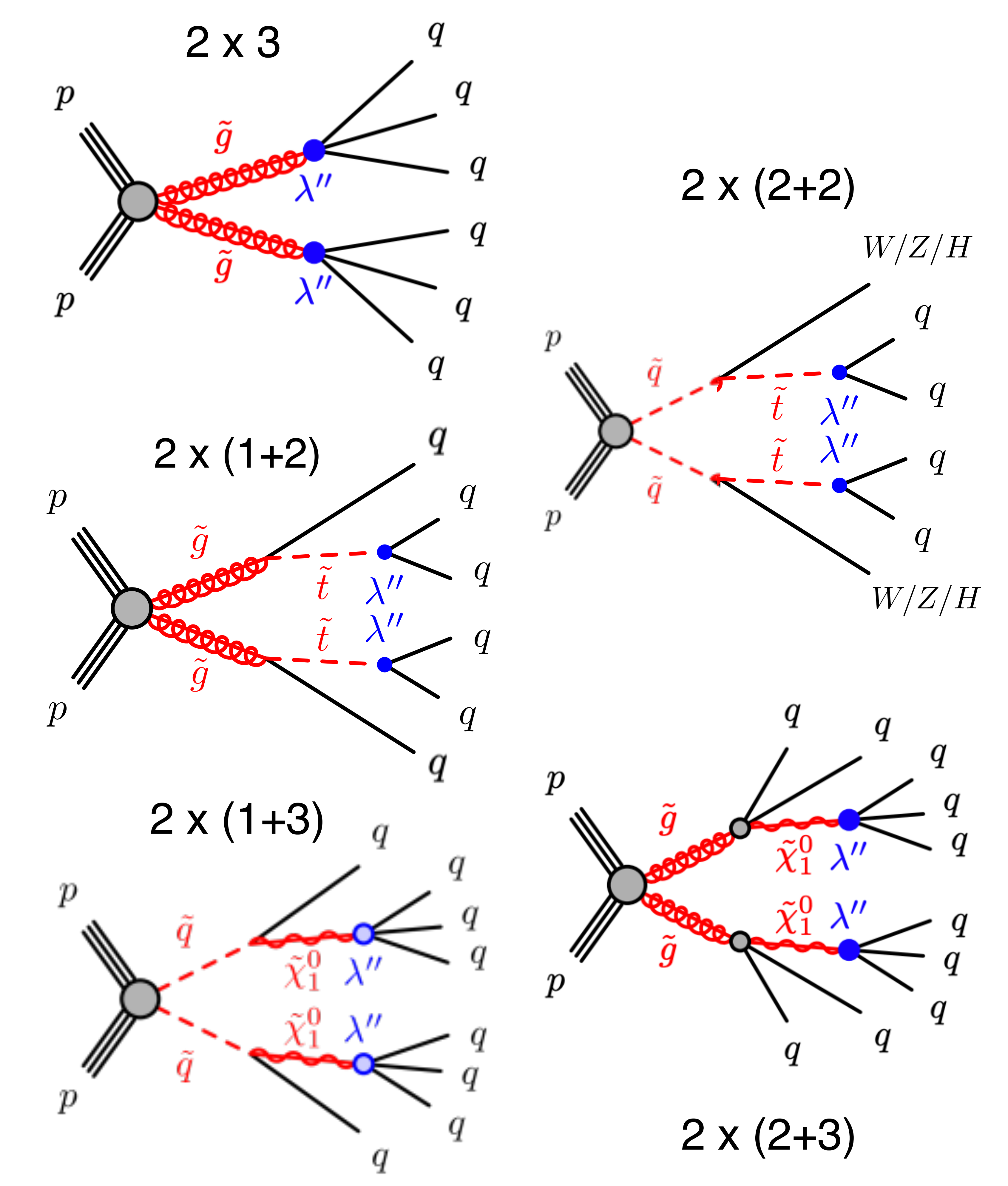
Takane Sano[1], Javier Montejo Berlingen[2], Kunihiro Nagano[3]

[1]Kyoto University,
[2]The Barcelona Institute of Science and Technology,
[3]High Energy Accelerator Research Organization (JP)

Machine Learning at HEP, 10th Jan 2024

Inclusive multi-jet search

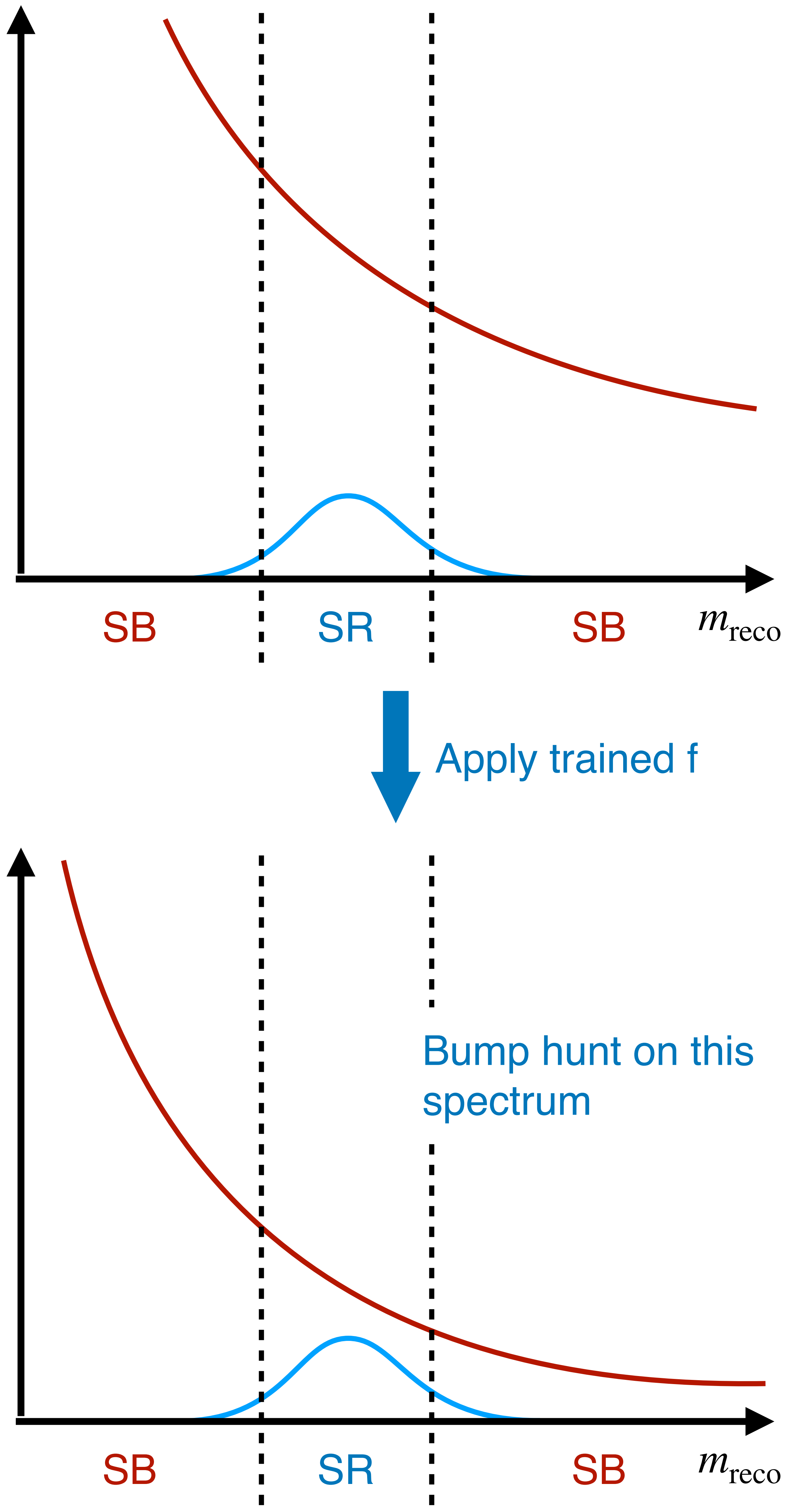
- **Multi-jet final state**
  - Final state with multiple quarks
  - Can be a portal to variety of BSM
  - Especially decays only with light flavor quarks are not well-covered
- **Example: RPV SUSY**
- **Don’t limit the target as far as possible**
  - Pair production of heavy particles
  - Both decay through all hadronic channel
- **Can even cover something not considered yet**



Benchmark channels for RPV SUSY

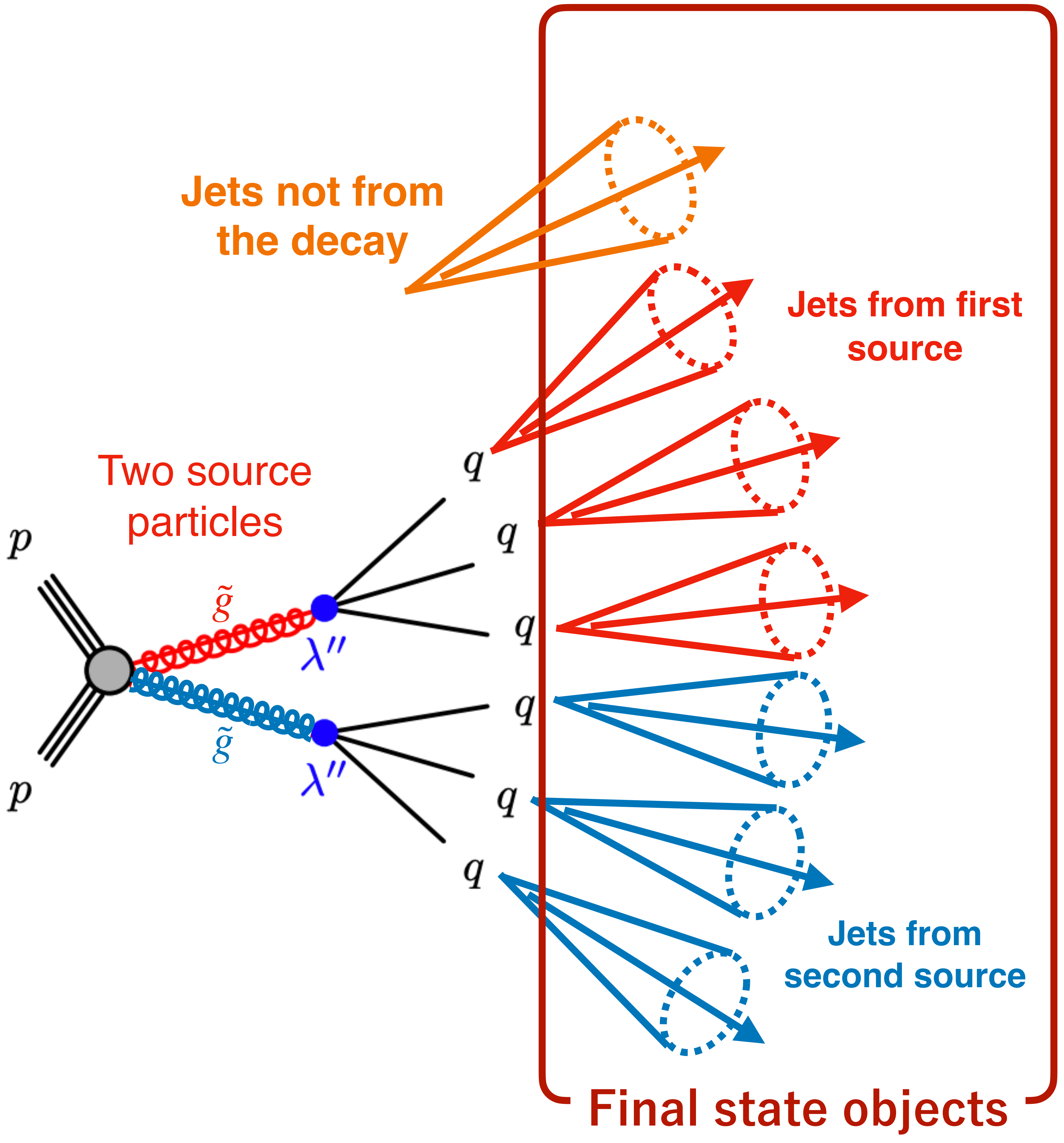
Strategy for inclusive search

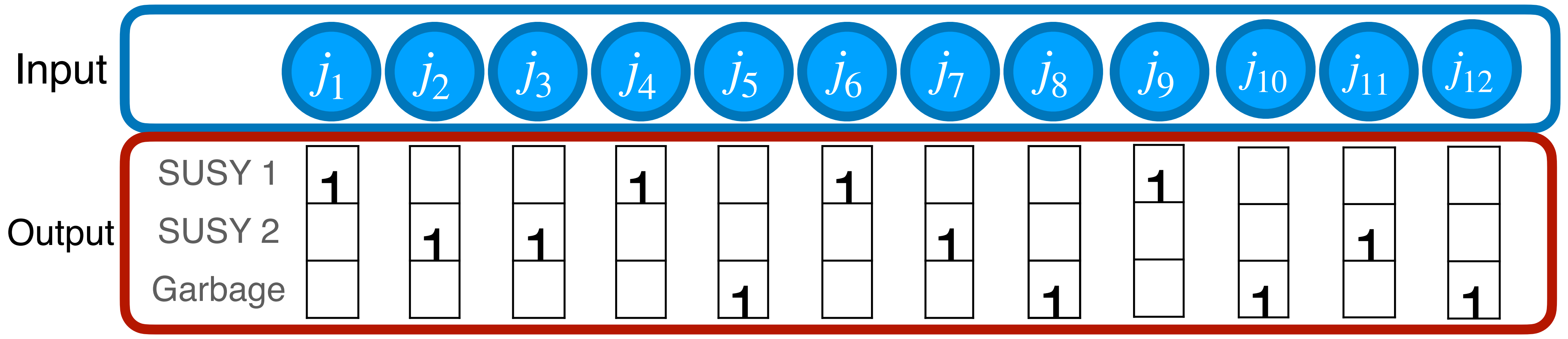
- **Bump hunting on mass spectrum with the help of weakly-supervised ML**
  - Fully data driven
    - CWoLa [arXiv:1708.02949], CATHODE [arXiv:2109.00546], CURTAINs [arXiv:2203.09470], etc.
  - Data driven weak supervision
    - Pick a data event from either SB (Side band) / SR (Signal region)
    - Train SB / SR classifier f (not signal/background)
      - f(event) $\mapsto$ SB or SR
    - **Magically, optimal f is equivalent to optimal signal / background classifier!**
  - Use trained f to enhance significance
    → Do bump hunt on top of that
- **Requirement: signal needs to be localized**
  - Challenge: Model-independent sharp reconstructed mass



Reconstruction of multi-jet events

- **Today's main topic**
- **Jets - source particle matching**
  - To reconstruct the mass, we need the correct combination of jets to build each of the two source particles
  - Source particles are the same kind of particle
    -> Make two groups of jets so that they looks alike
  - Some jets may come outside of the decay
    -> One additional group (garbage group)
  - Develop jets grouping method based on ML
    - Challenge: Model independence



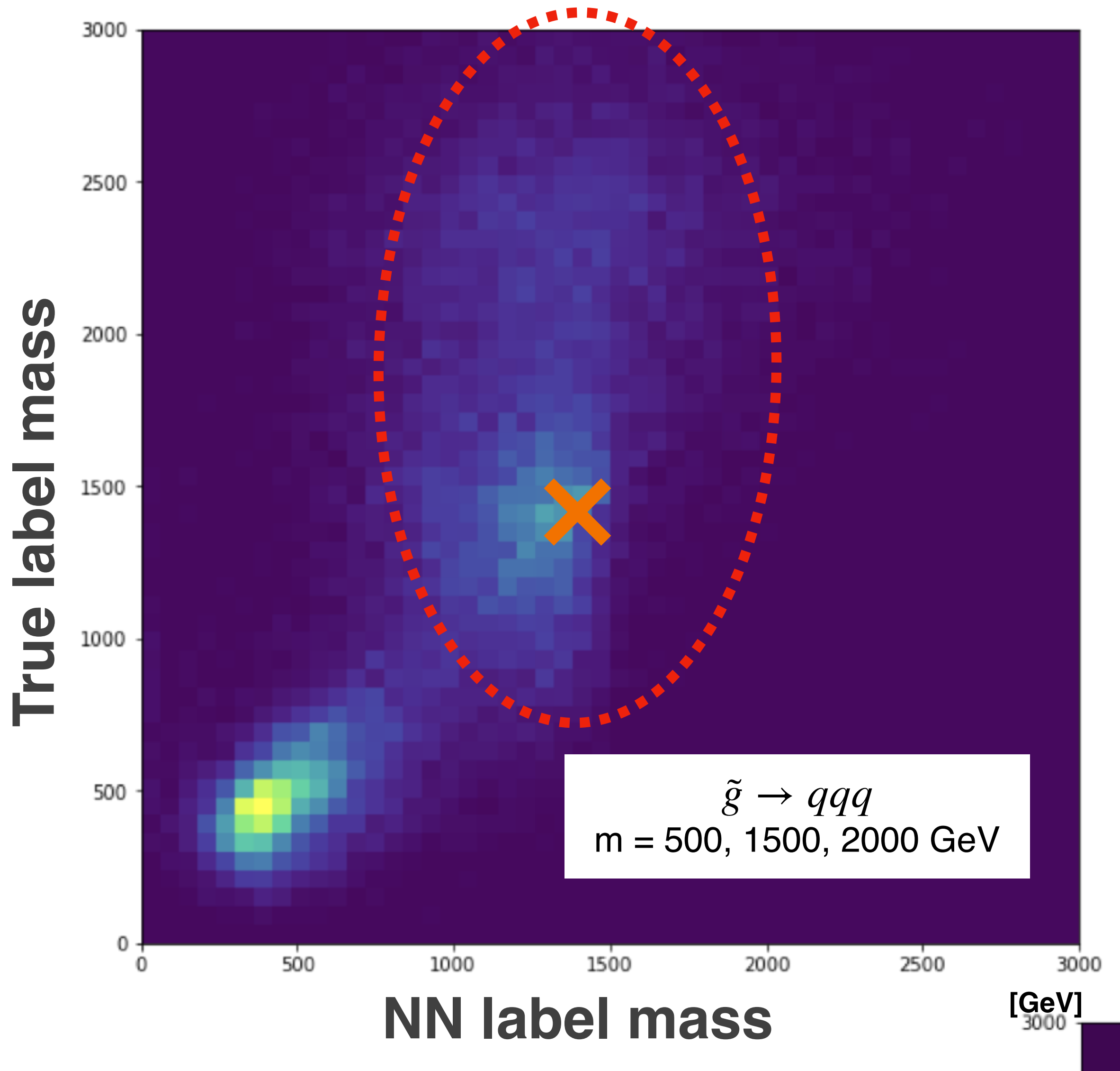


Training sample dependence: mass dependence

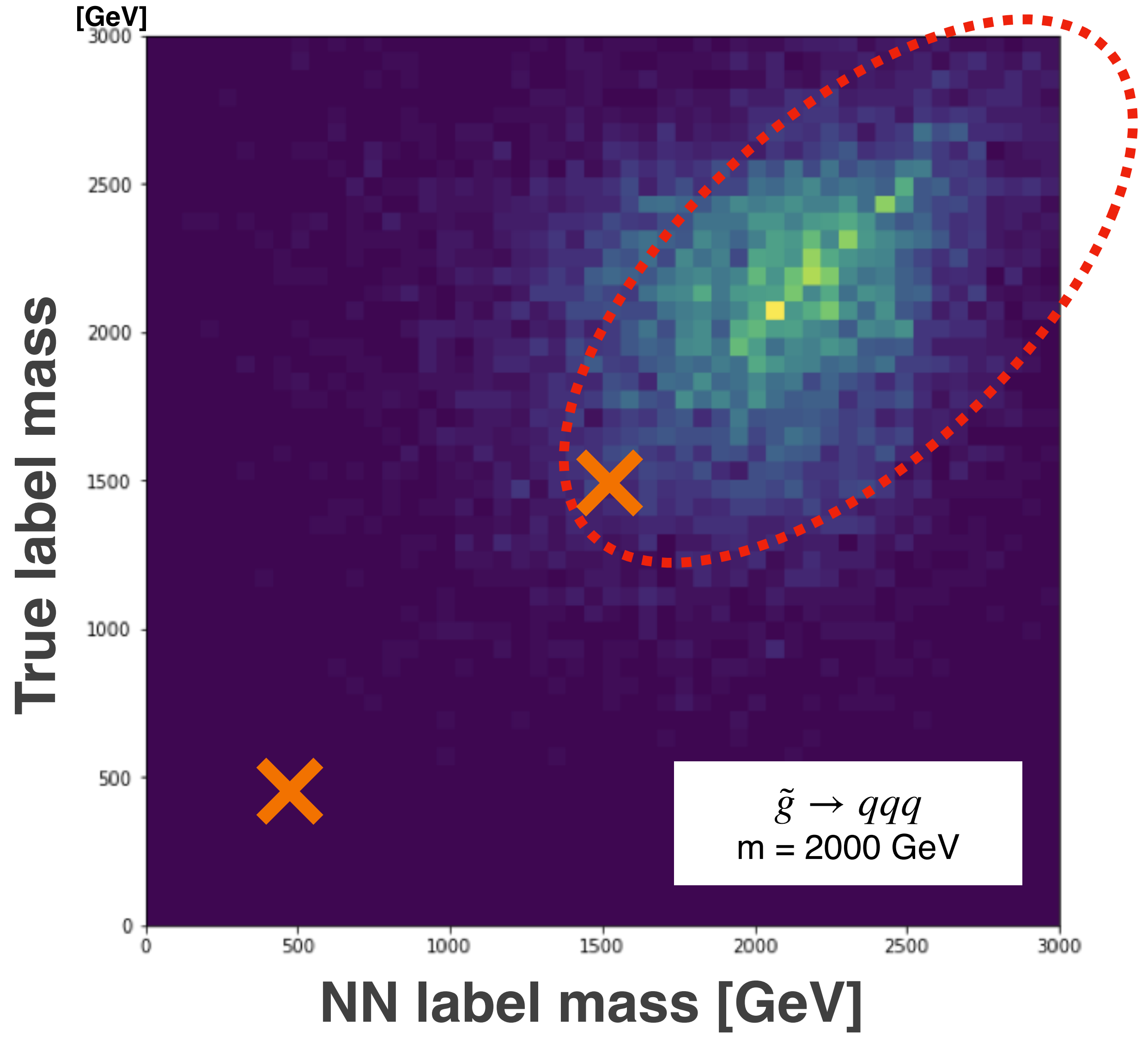
- **Mass dependence**
  - When trained on single mass…
    - Cannot reconstruct mass higher than the given mass
- **What if adding other mass in the training?**
  - Mixing two masses in the training and test on unknown mass

  → Now NN can reconstruct masses higher than the given mass!
- **Just adjusting training sample composition can help mass-independence**
  - Single mass
    -> Learns the mass
  - Mixed mass
    -> Learns to make two similar groups of jets

Trained with single mass



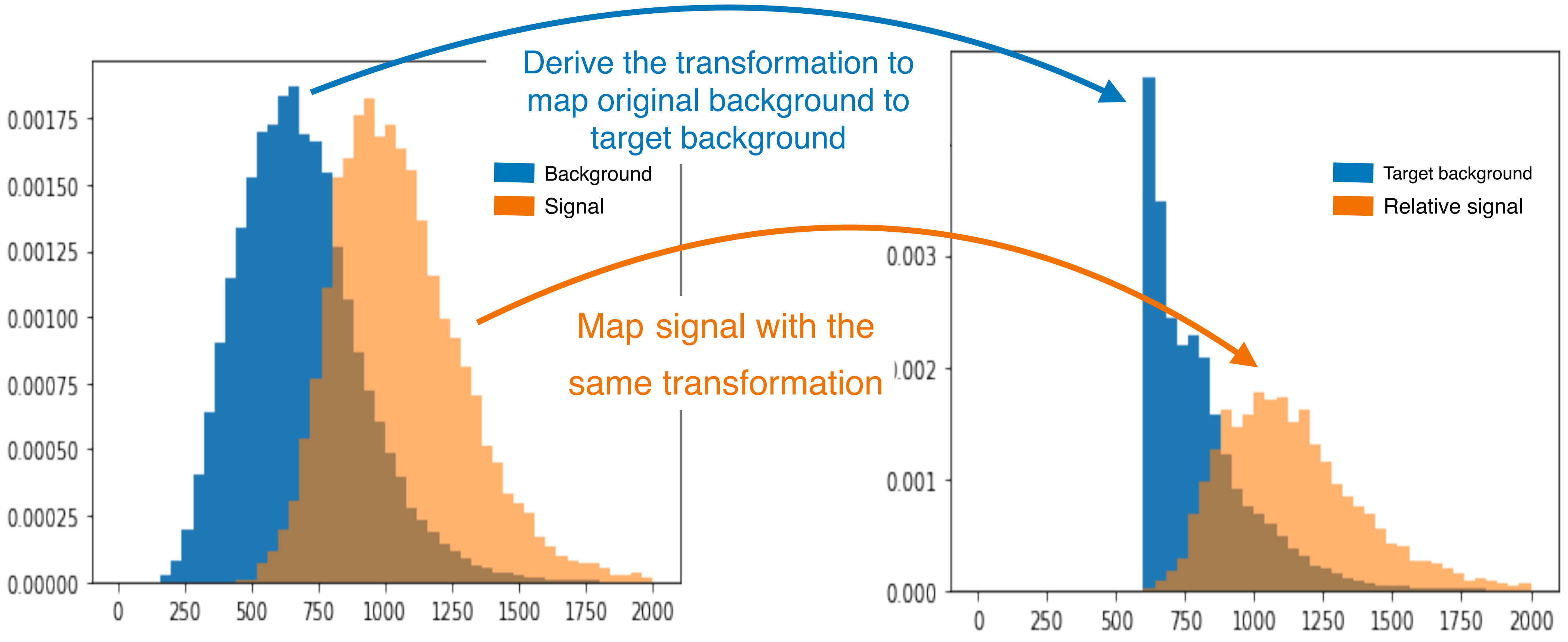
Trained with two masses

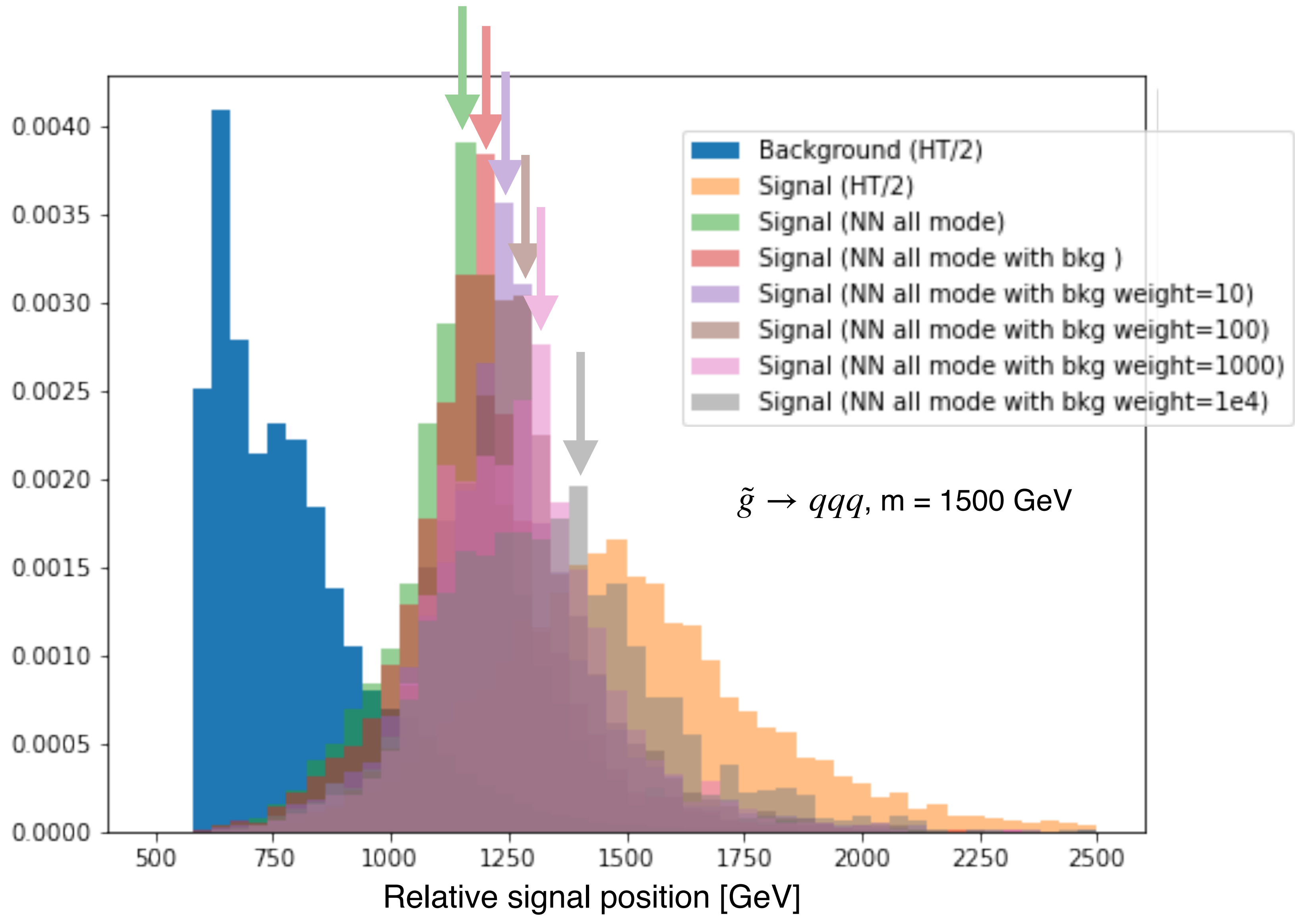


Training sample dependence: background sculpting

- **Background sculpting**
  - NN is trained to reconstruct high mass signals
    -> Tends to reconstruct backgrounds in high mass
- **Can composition adjusting useful in this case too?**
  - → Yes!
  - Mixing background in the training sample
  - → NN learns low mass background
- **Relative signal position**
  - Aim to compare signal shape on the same background distribution
- **Adding backgrounds to training samples with training weights**
  - Relative signal position goes to the right side as the background weight increases
  - → Increasing backgrounds' importance in the training effectively improves the background sculpting

Concept of relative signal position

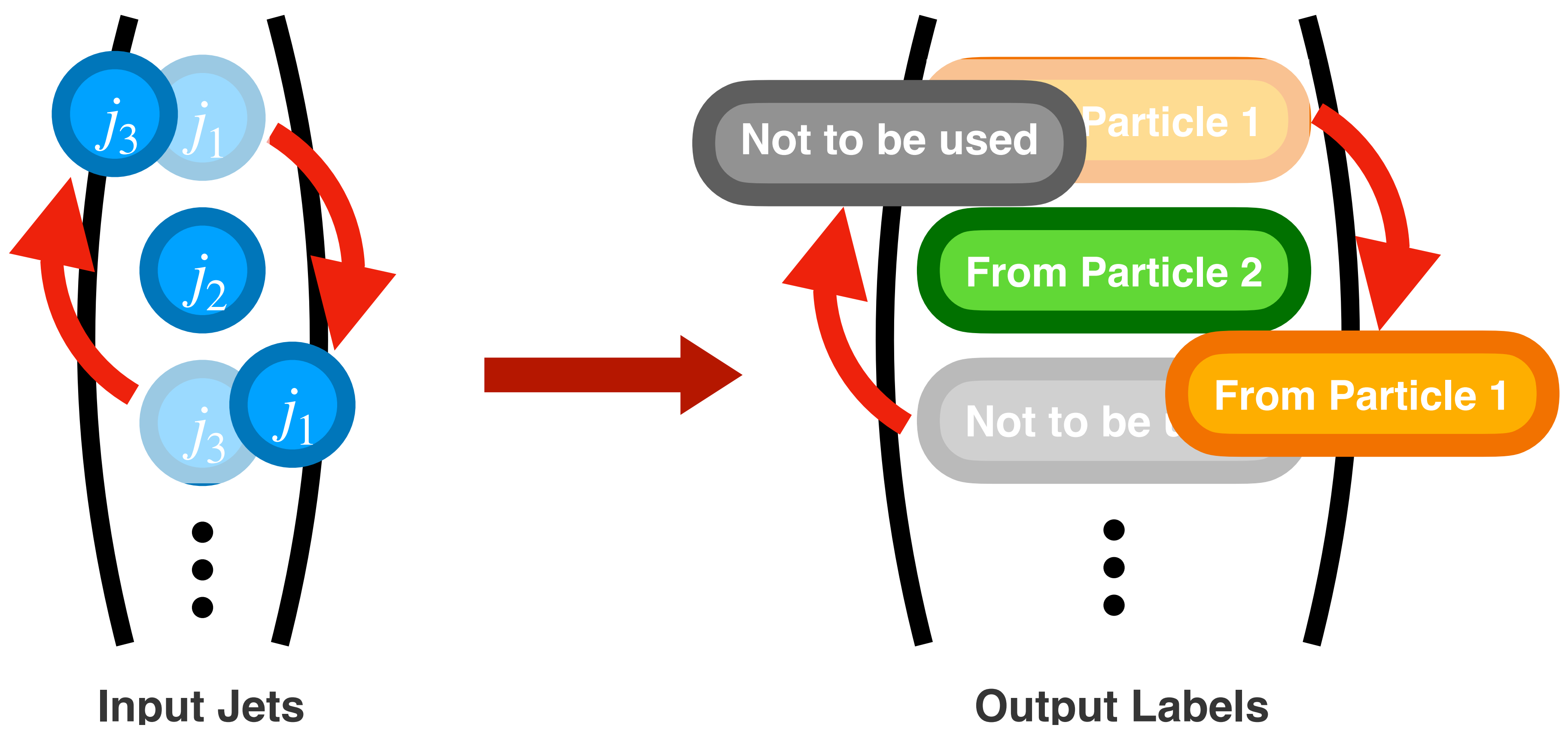




Training sample dependence: number of jets dependence

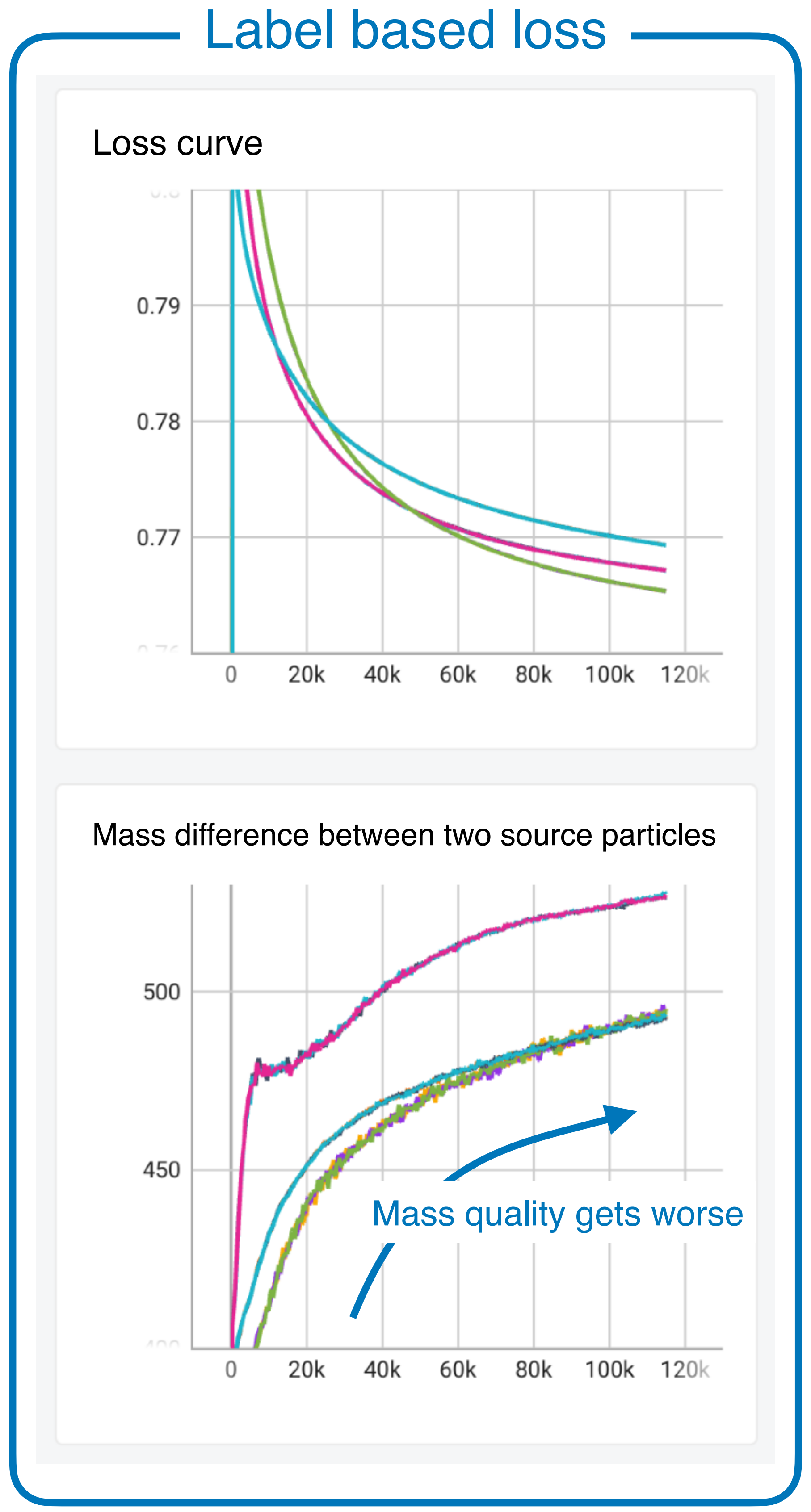
- **Number of jets dependence**
  - Number of jets differs based on decay mode
  - When an event doesn't have enough jets, last parts of the input are masked

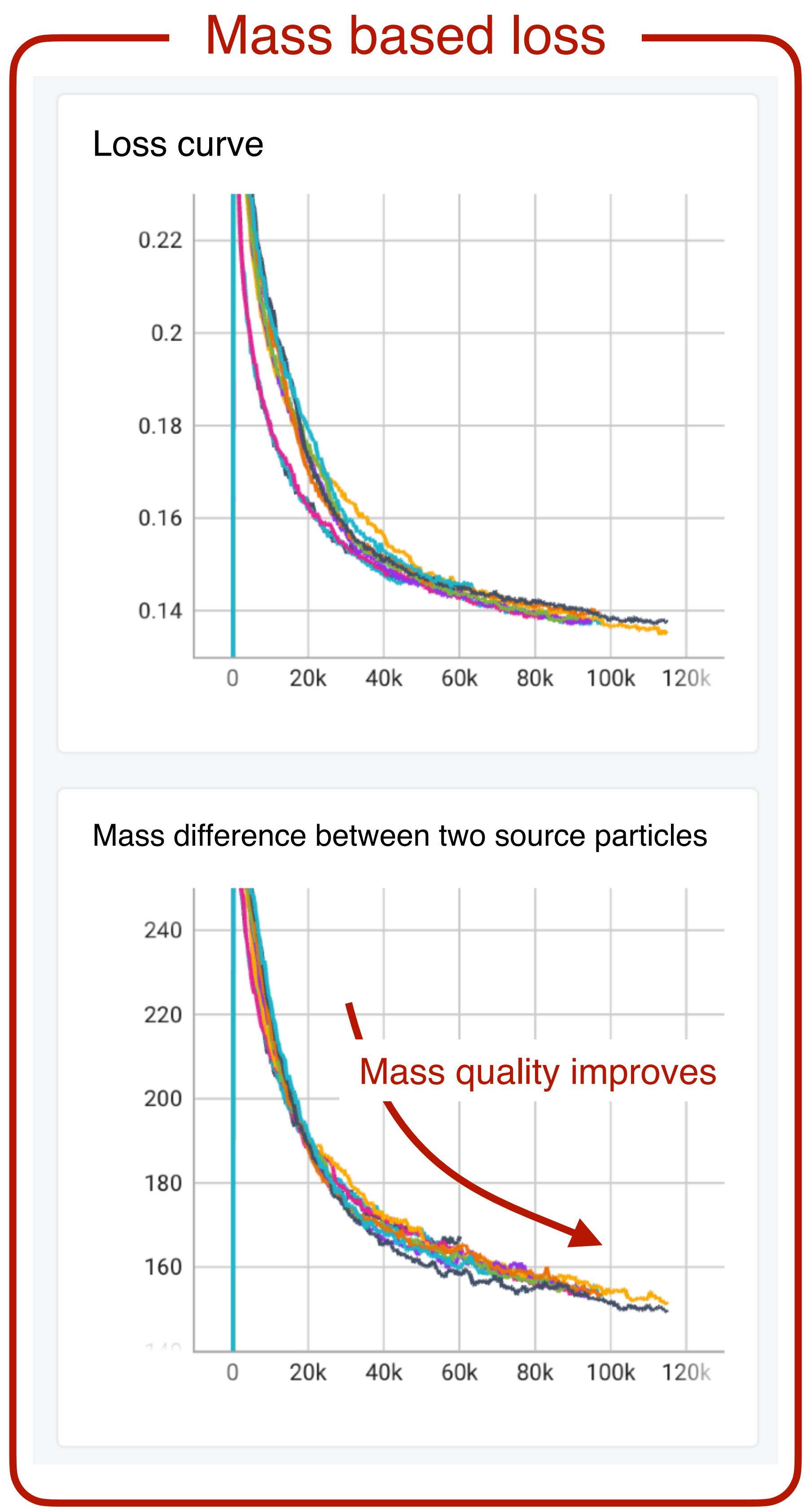
  → NN learns not to use trailing jets in the reconstruction
- **Permutation equivariance**
  - Jet grouping should not depend on jets' order
    - When swap two jets, the output label should be swapped too
  - Permutation equivariant linear transformation can be constructed with per-jet transformation + pooling
  - Replace all dense layer with permutation equivariant layer
    -> Permutation equivariant (and cannot learn jet order) by structure



Loss function definition

- **Loss function**
  - Output of the NN: 3-category label × 12 jets
  - Label based loss
    - Focus on labeling accuracy: Cross-entropy loss
  - Mass based loss
    - Focus on the difference between reconstructed mass and true mass
    - Utilize Gumbel-softmax function for differentiable reconstruction
- **Label based v.s. mass based**
  - Labeling accuracy → label based works better
  - Reconstructed mass quality
    - Mass based works better
    - With label based, mass quality gets worse as the loss decreases (the training progresses)

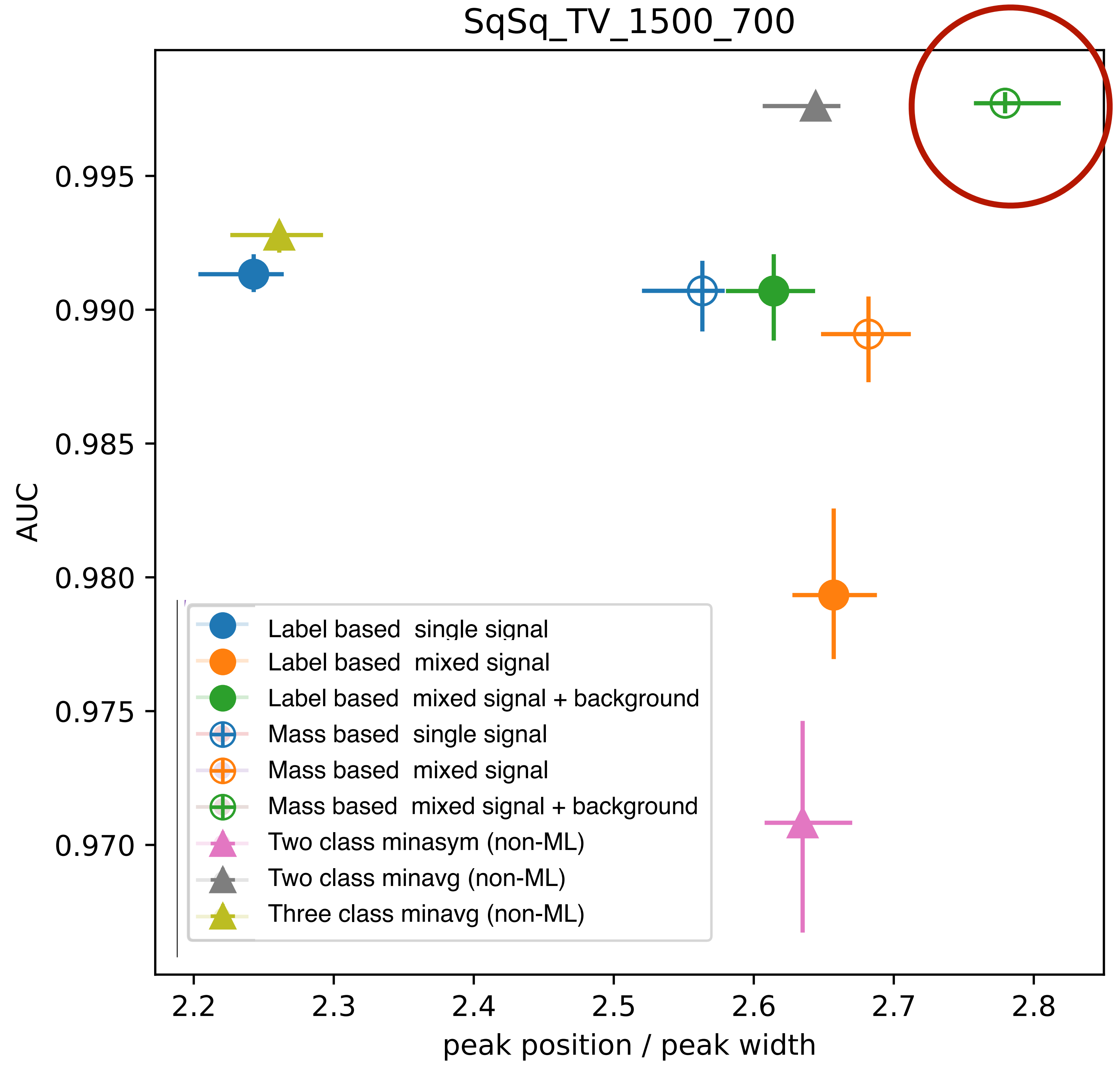




Result - reconstruction characteristics

- **Characteristics evaluation**
  - Evaluate two aspects: background sculpting(AUC) and peak quality
  - Reconstruct signals which is not known in the training
- **Compare different settings, including non-ML method**
  - Six different ML setting
    - Two losses: label based loss, mass based loss
    - Three sample composition:
      single signal, mixed signal, signals + background
  - Three non-ML method
- **Result**
  - Label based (filled circle) v.s. mass based (empty circles)
    -> Mass based outperforms in both aspects, bkg sculpting and peak quality
  - Single signal v.s. mixed signal v.s. signal+background
    - We can see the effects
      - Mixing different signals enhances peak quality
      - Adding background mitigates background sculpting
- **Mass based with signal+background outperforms all**

Reconstruct sq > stop + W/Z signal
(sq, stop) = (1500, 700) GeV



Summary / future challenges

- **Summary**
  - **Bump hunt with weakly supervised ML**
    → Need narrow mass peak as well as model-independence
  - **Model-independence in ML based reconstruction**
    - Although it is simple, training sample adjustment helps a lot
      - Mass dependence, background sculpting, decay mode dependence
    - For the dependence which cannot be dealt with sample adjustment, modify architecture
  - **Mass based loss improves the performance compared to cross-entropy loss**
- **Challenges**
  - How to check model-independence systematically?
    - Currently, exclude one decay mode from the training and test the performance on it
    - Can we use ML to evaluate generalness of the method?
      → Some sort of anomaly detection on "functions' behavior"…?

Backup