Searching for long-lived particles decaying to jets using a deep neural network

6th LHC LLP Workshop

M. Komm, **V.Čepaitis,** R. Bainbridge, A. Tapper, O. Buchmüller, J. Dancu,
on behalf of the CMS collaboration

28/11/2019

Imperial College London

Outline

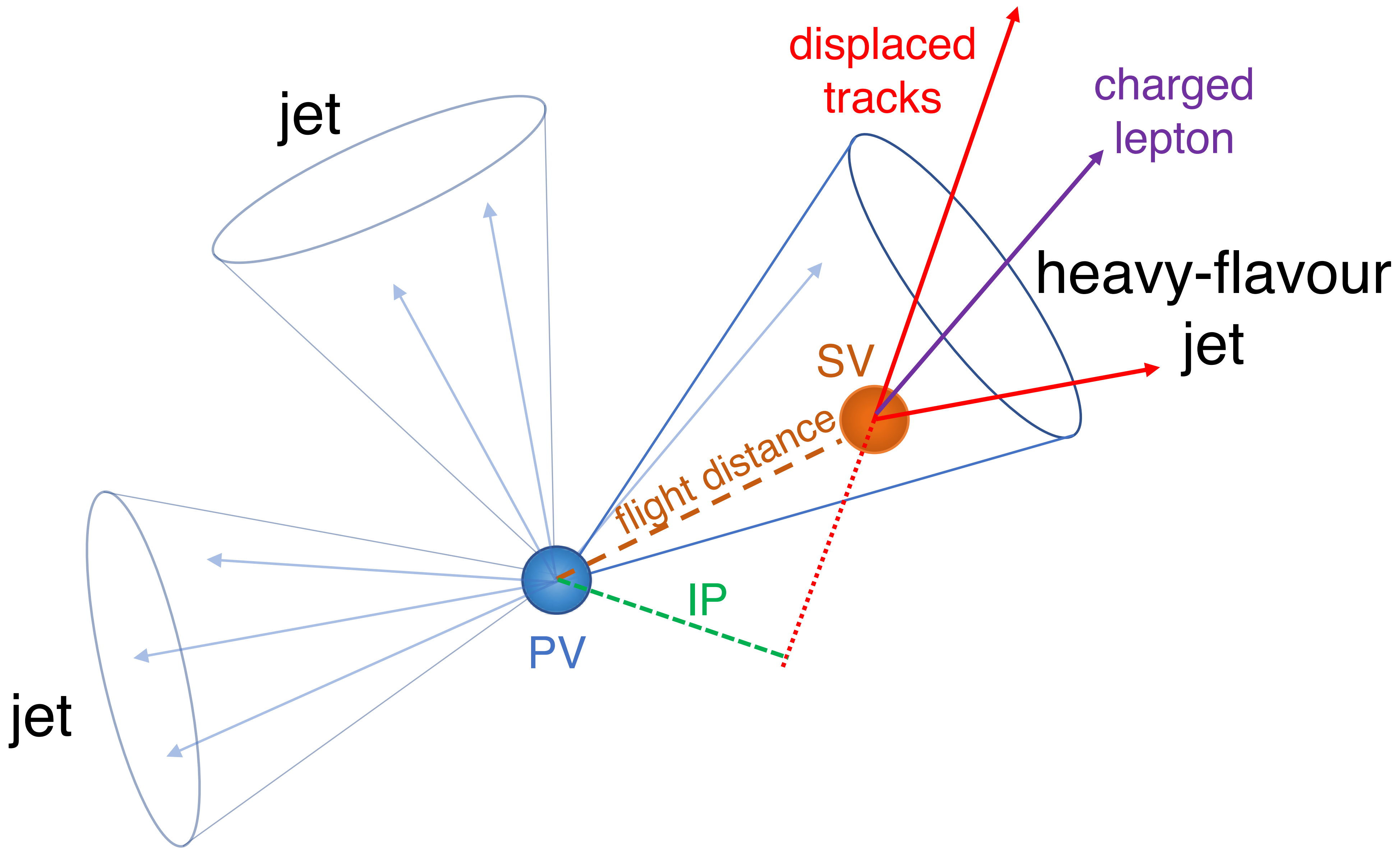
- B-tagging at CMS
  - DeepJet: unprecedented gains in performance from deep learning
  - Basis for a displaced jet tagger

- Developments for displaced jet tagging
  - New truth-level LLP jet definition
  - Network training on data and simulation
  - Network validation in control regions
  - Tagger performance for different LLP models

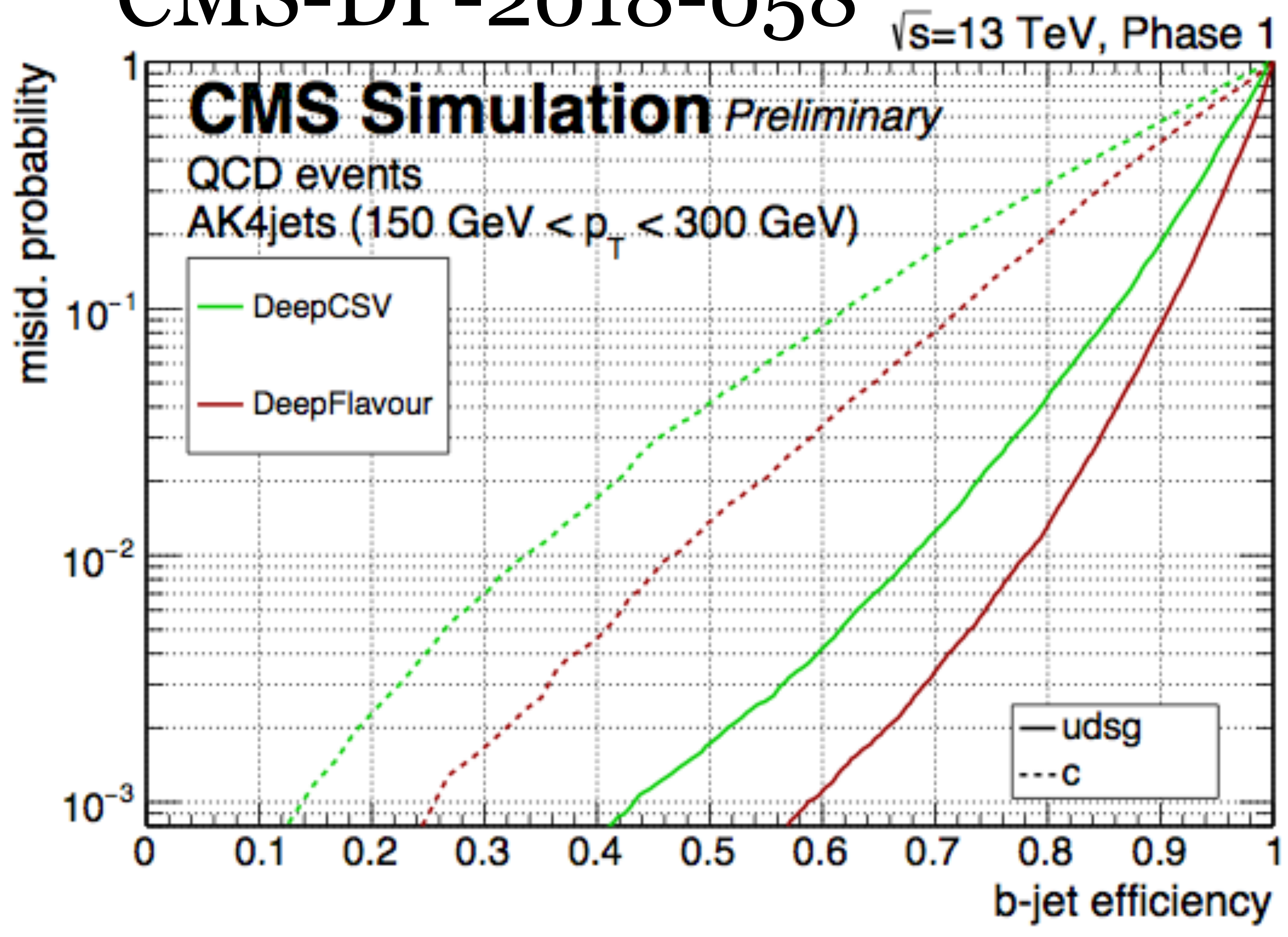
- Tagger application in a showcase search for split SUSY
  - In-situ tagging efficiency constraint

Flavour tagging at CMS

- Exploits features of displaced tracks and secondary vertices
  - Tag jets as b, c or light flavour
- Legacy algorithm: DeepCSV
  - Uses human-engineered high-level features
- Latest algorithm: DeepJet
  - Treats jets as sequences of particles
  - Exploits low-level jet constituent information
  - $\mathcal{O}(10)$ improvement observed in some regions of phase space over DeepCSV
- Good first step for a generic displaced jet tagger
  - b jets like “displaced jets” with $c\tau_0 \approx 1$ mm

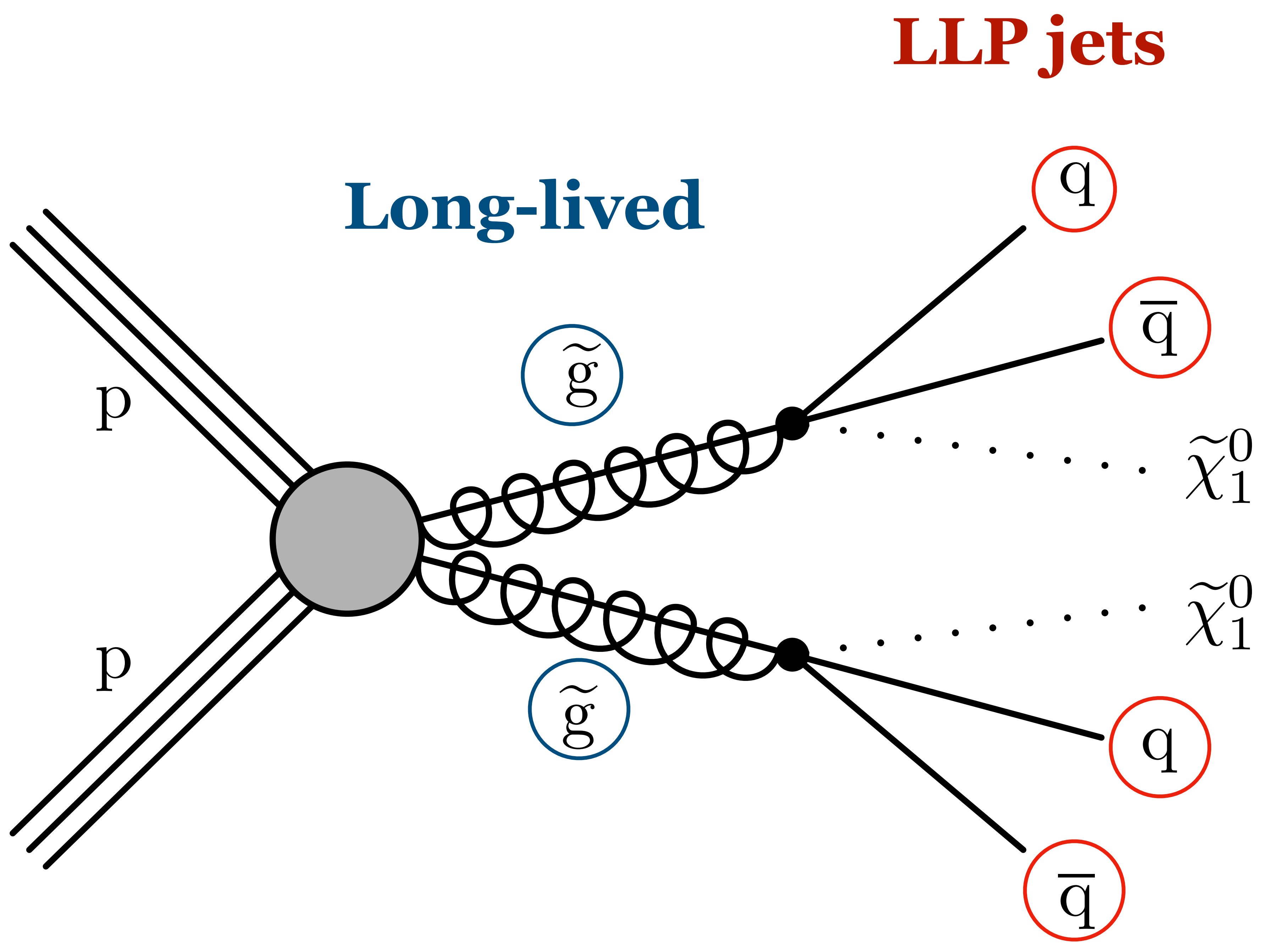


CMS-DP-2018-058



Developing a generic jet tagger

- Benchmark model: $pp \to \tilde{g}\tilde{g}, \tilde{g} \to \tilde{\chi}_1^0 qq$ (split SUSY)
  - Rich phenomenology depending on $m_{\tilde{g}}, m_{\tilde{\chi}}, c\tau_0$:
    - $m_{\tilde{g}} \gg m_{\tilde{\chi}}$: energetic jets
    - $m_{\tilde{g}} \approx m_{\tilde{\chi}}$: challenging "compressed" scenarios
    - $c\tau_0 = 10\,\mu$m: SM-like jets
    - $c\tau_0 = 1$ mm: b-jet like
    - $c\tau_0 = 1$ m: decays outside tracker



- Difficulties compared to b-tagging:
  - What is an "LLP jet" at truth level?
  - Displaced jet signature depends strongly on LLP lifetime
  - No displaced control region to commission tagger
  - No way to measure signal tagging efficiency in data

PAS: EXO-19-011

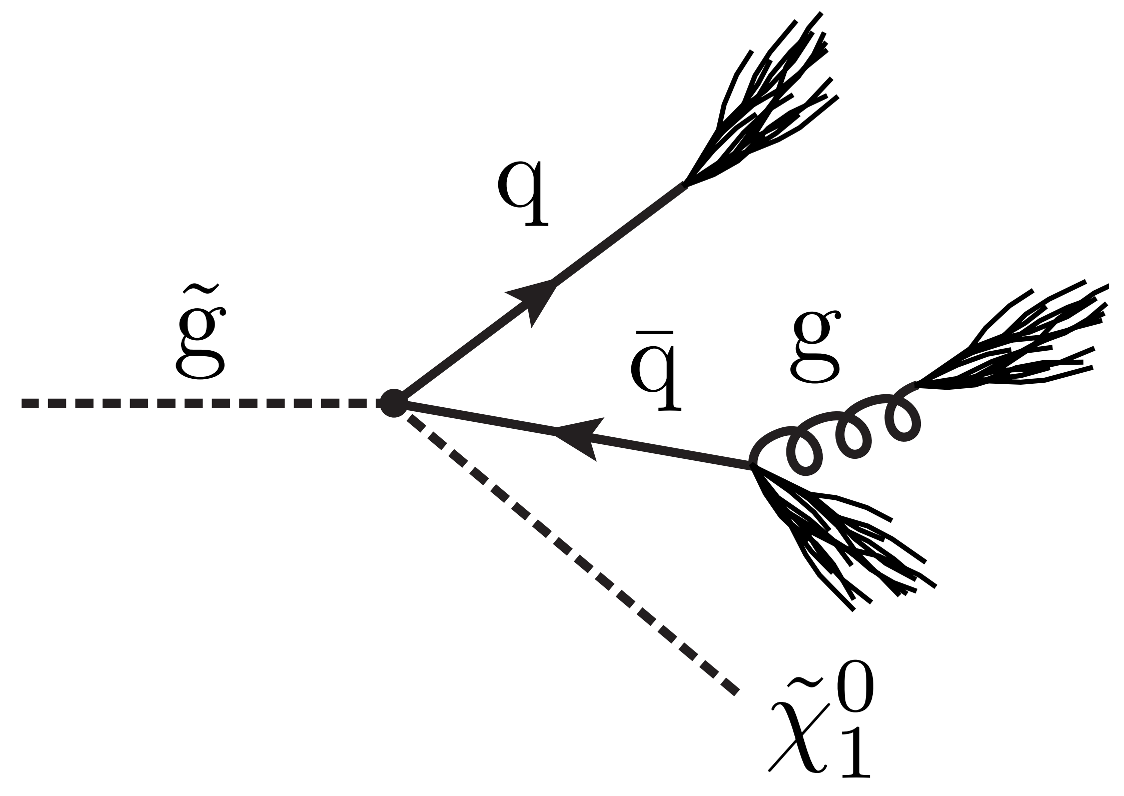
Ghost tagging

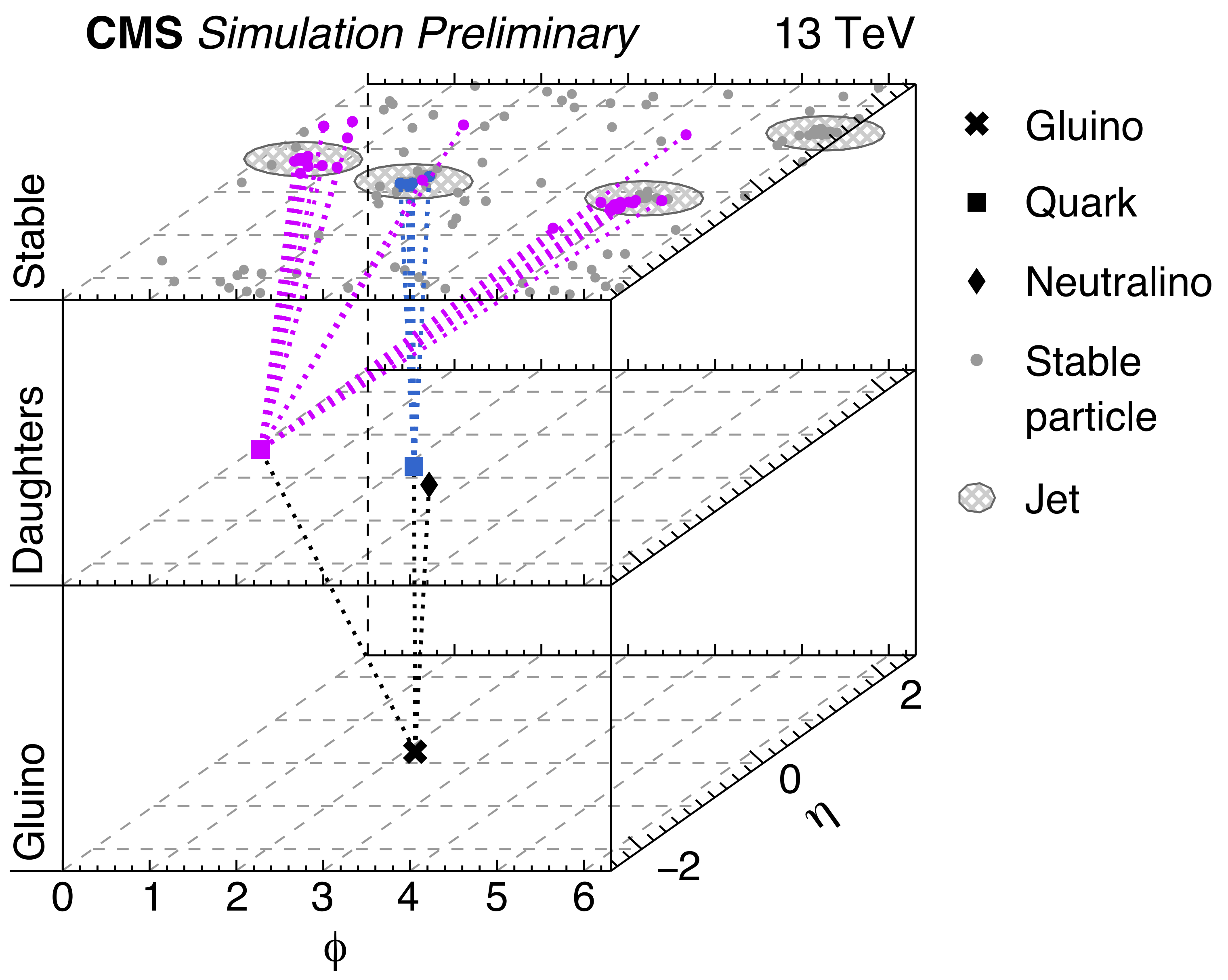
- Need a truth-level definition for a supervised algorithm
  - Try existing methods in literature

- Flavour tagging uses ghost-association:
  - Scale down generator-level b hadron momentum
    - “Ghost” hadron with only directional information
  - Rerun jet clustering treating ghost hadrons as reconstructed particles
  - Label jet as b if ghost hadron included in jet

- Problem: $\tilde{g}$ decay chain leads to loss of directionality
  - Three body-decay ($\tilde{g} \rightarrow \tilde{\chi}_1^0 qq$) leads to non-pointing partons
  - Resulting partons color-connected and interact further

- No good candidate to be used as a ghost from LLP decays





Jet labelling

- New generic LLP jet definition based on generator-level information

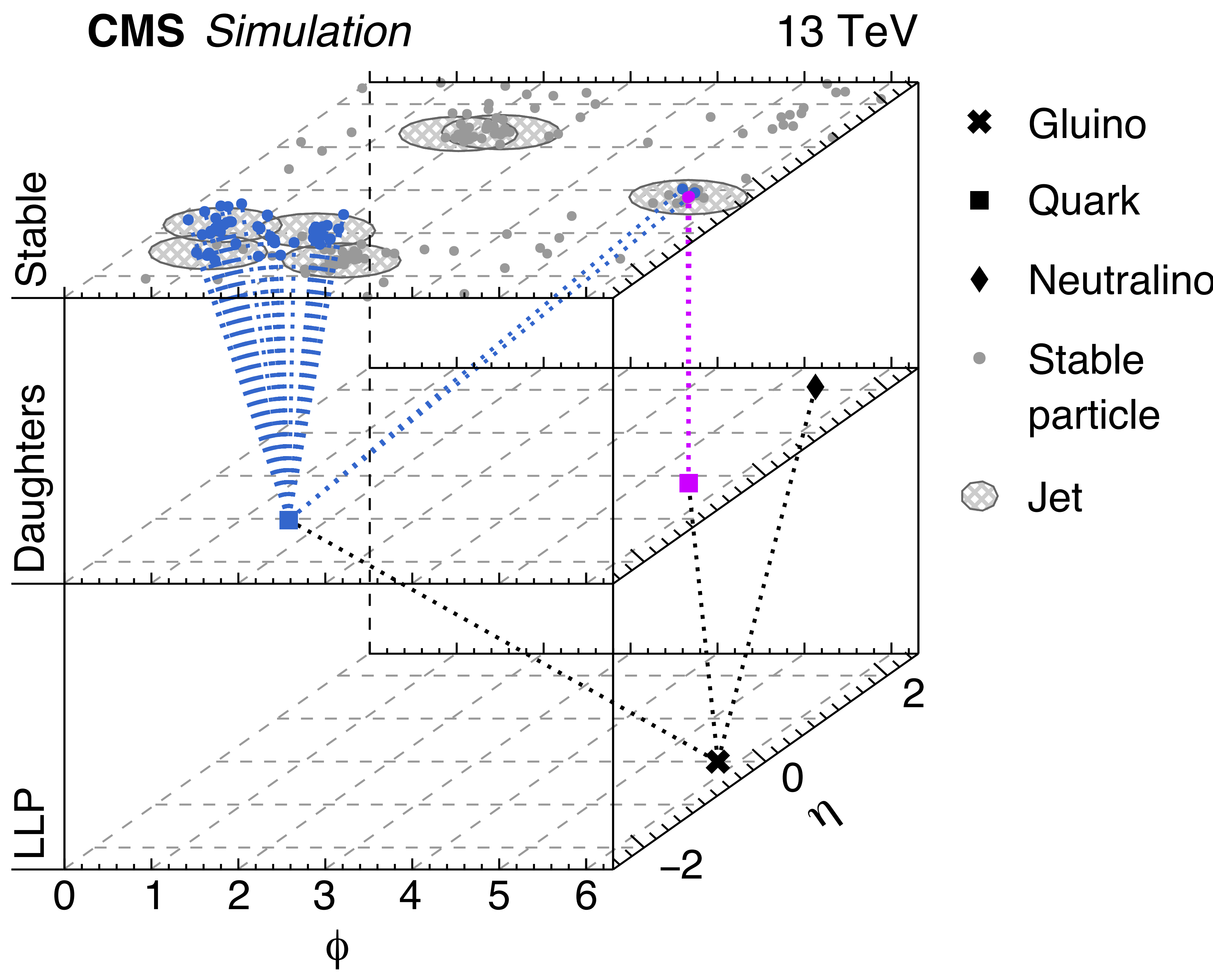
- Form vertices by grouping all stable final-state particles

- Our choice: maximise shared jet-vertex momentum fraction:

$$f_v = \frac{\vec{p}_v \cdot \vec{p}}{p^2}, \quad \vec{p}_v = \Big(\sum_i \vec{p}_i \,\Big|\, i \in v\Big), \quad f_v \in [0;\ 1], \quad \sum_v^{\text{vertices}} f_v = 1.$$

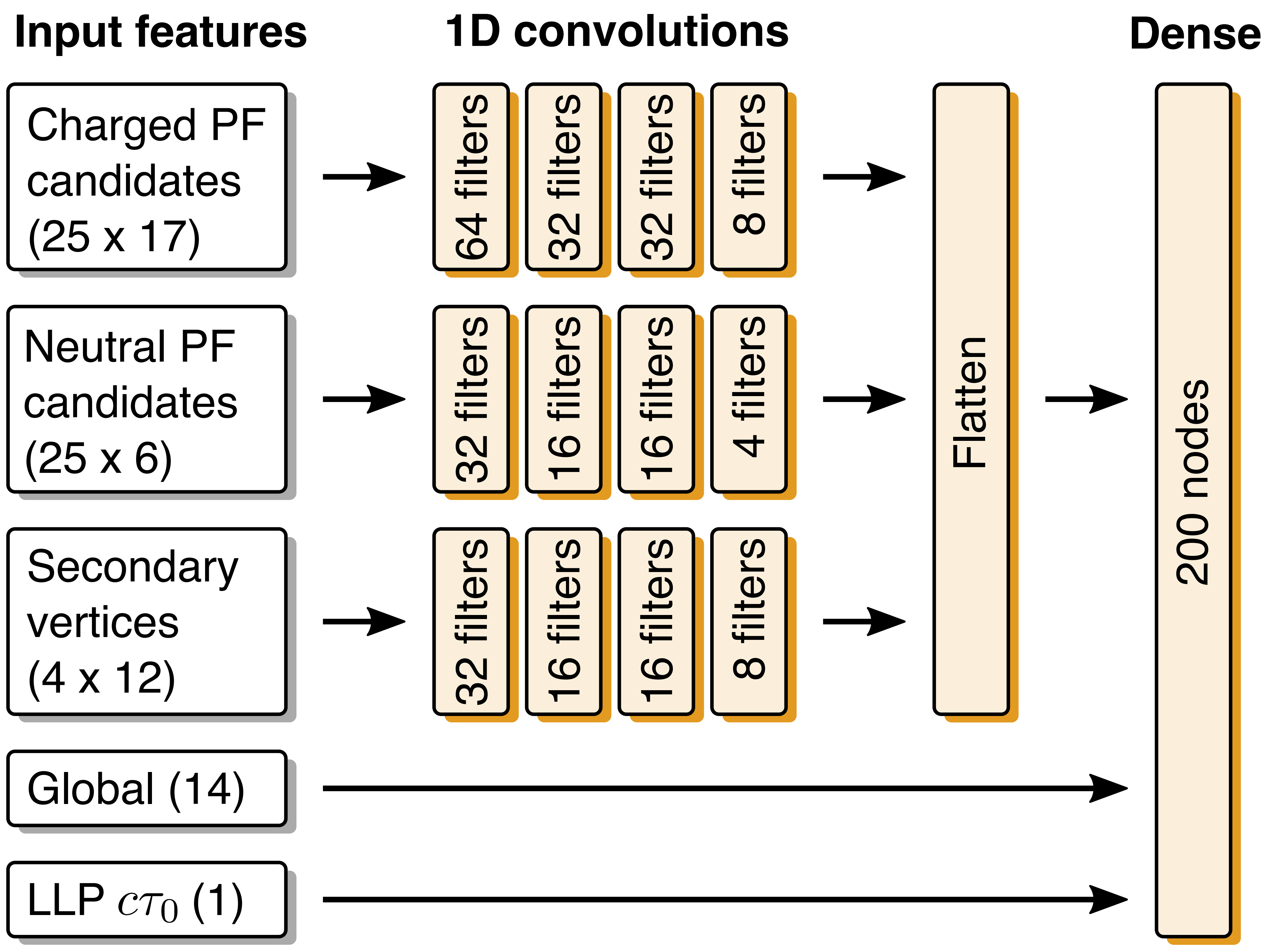
  - $\vec{p}_i$ — jet constituent i momentum, $\vec{p}$ — jet momentum
  - Check if vertex, $\hat{v}$, with maximum $f_{\hat{v}}$ is an LLP decay vertex
  - **i.e. require most of jet momentum stems from an LLP decay vertex**

- Transfer label to reconstructed jet with $\Delta R < 0.4$



Network inputs and feature engineering

- Tagger based on low-level jet constituent features
  - Up to 25 charged tracks
  - Up to 25 neutral particles
  - Up to 4 associated secondary vertices
  - Global jet features
- Architecture progressively compresses and extracts most discriminating features
  - Per jet constituent (track, neutral hadron, SV)
- Network parametrised by LLP jet mother $c\tau_0$
  - "Fake" $c\tau_0$ generated for background jets to avoid discrimination on a latent (truth-level) variable
- Resample jet p_T, η to have same distribution for all classes
  - Discrimination only via correlations with other features

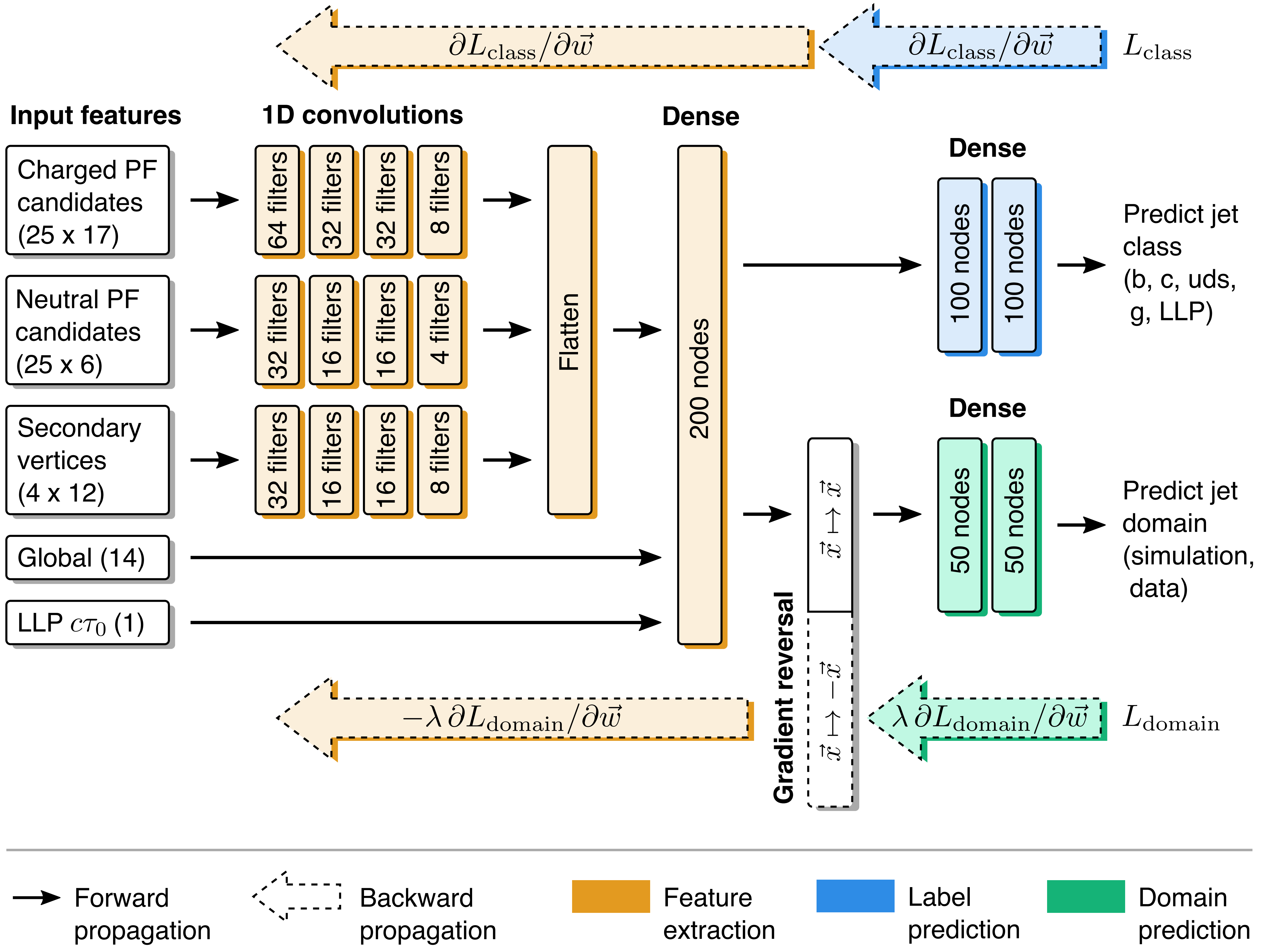


Network architecture

- Multiclass tagger: LLP, b, c, uds, g

- Training for jet label prediction:
  - Background: QCD, $t\bar{t}$
  - Signal: Split SUSY, $c\tau \in [10\ \mu\text{m}, 10\ \text{m}]$, various $m_{\tilde{g}}, m_{\tilde{\chi}}$ configurations

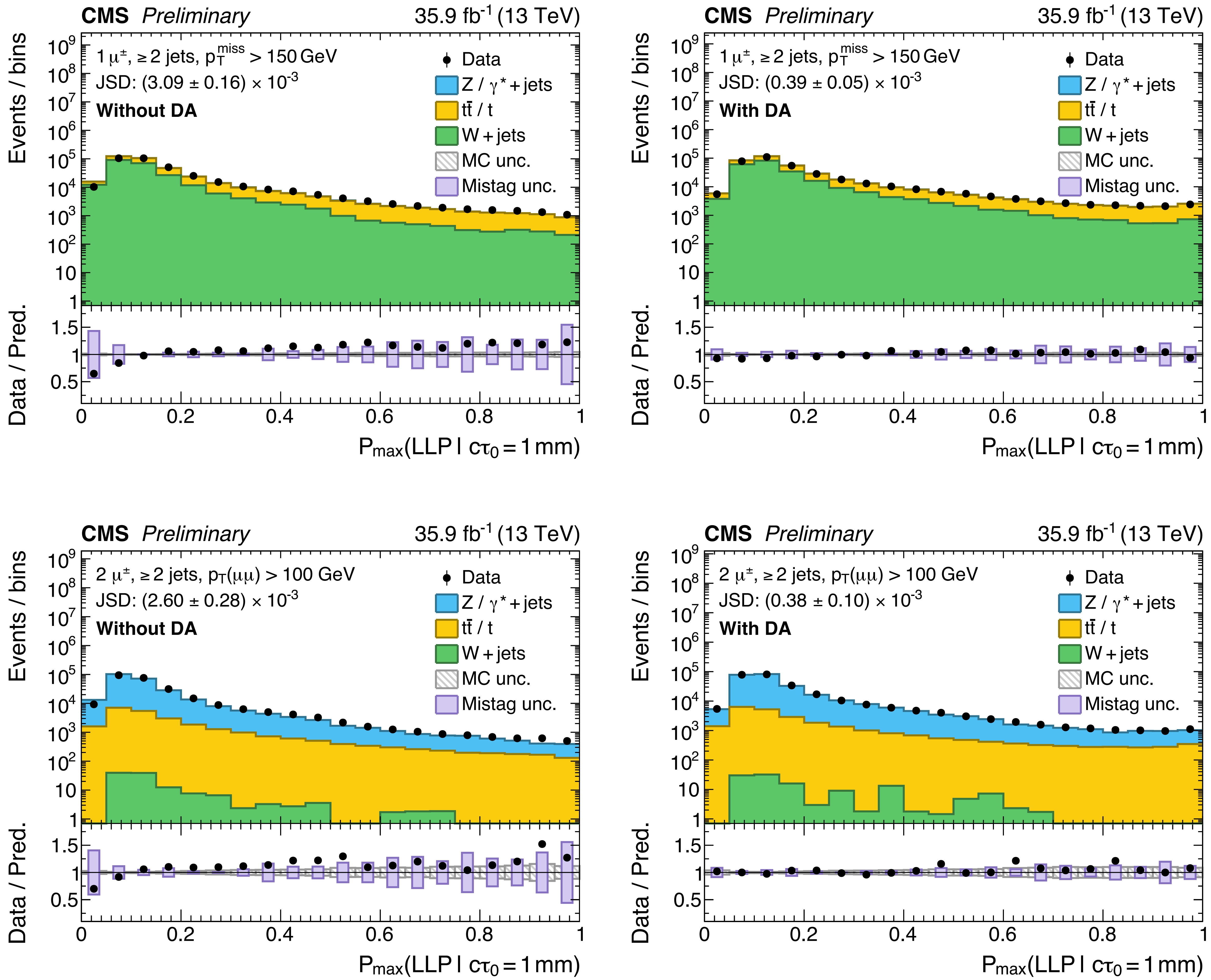
- Domain adaptation by backpropagation:
  - Network setup to deal with mismodelled jet features in simulation
  - Hyperparameter λ controls penalty due to data/MC disagreement
  - No displaced CR, select a prompt one:
    - Data: μ + jets
    - Simulation: W + jets, $t\bar{t}$ simulation



Minimise $L_{\text{class}} - \lambda L_{\text{domain}}$ for feature layer

Network validation

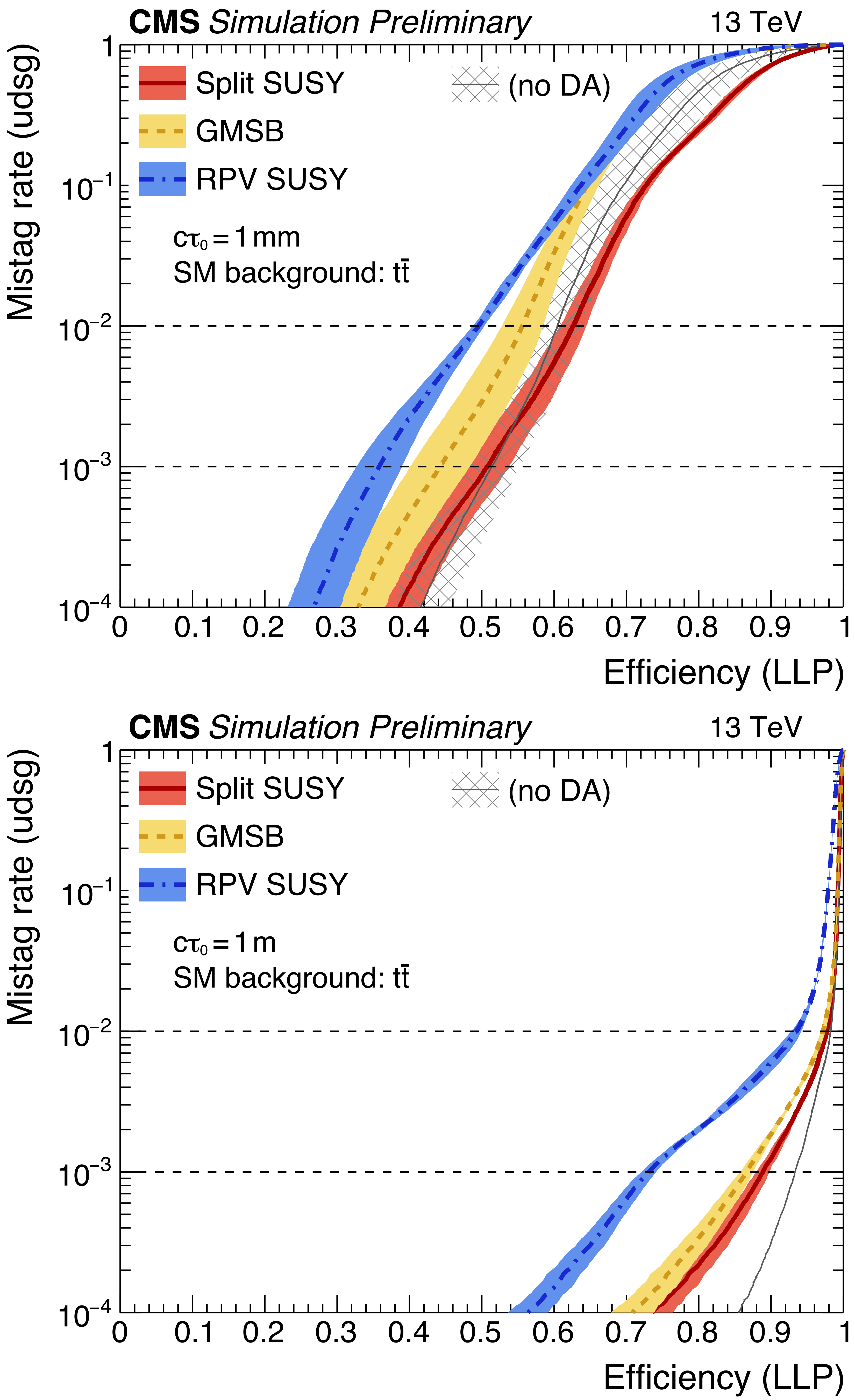
- Compare network trained with and without domain adaptation
  - Per-jet mistag rate (in purple) measured in a low-$p_{\mu\mu}$ μμ+jets control region
- LLP class probability is validated in μ+jets and high-$p_{\mu\mu}$ μμ+jets CRs
- Use Jensen-Shannon divergence to quantify agreement
  - Order of magnitude improvement after training with domain adaptation (DA)



Trained on MC → Trained on data+MC

Tagger performance

- Network trained on split SUSY model
  - Displaced uds jets
- Evaluate performance for benchmark GMSB ($\tilde{g} \rightarrow g\tilde{G}$) and RPV ($\tilde{t} \rightarrow b\ell$) models
  - Displaced g and b jets
- Keep 20-70% of LLP jets for a background rejection factor of 10k
  - 1 mm and 1 m LLP $c\tau_0$
- Some degradation from DA for significant displacement
  - Network exploits mismodelled features when trained exclusively on simulation

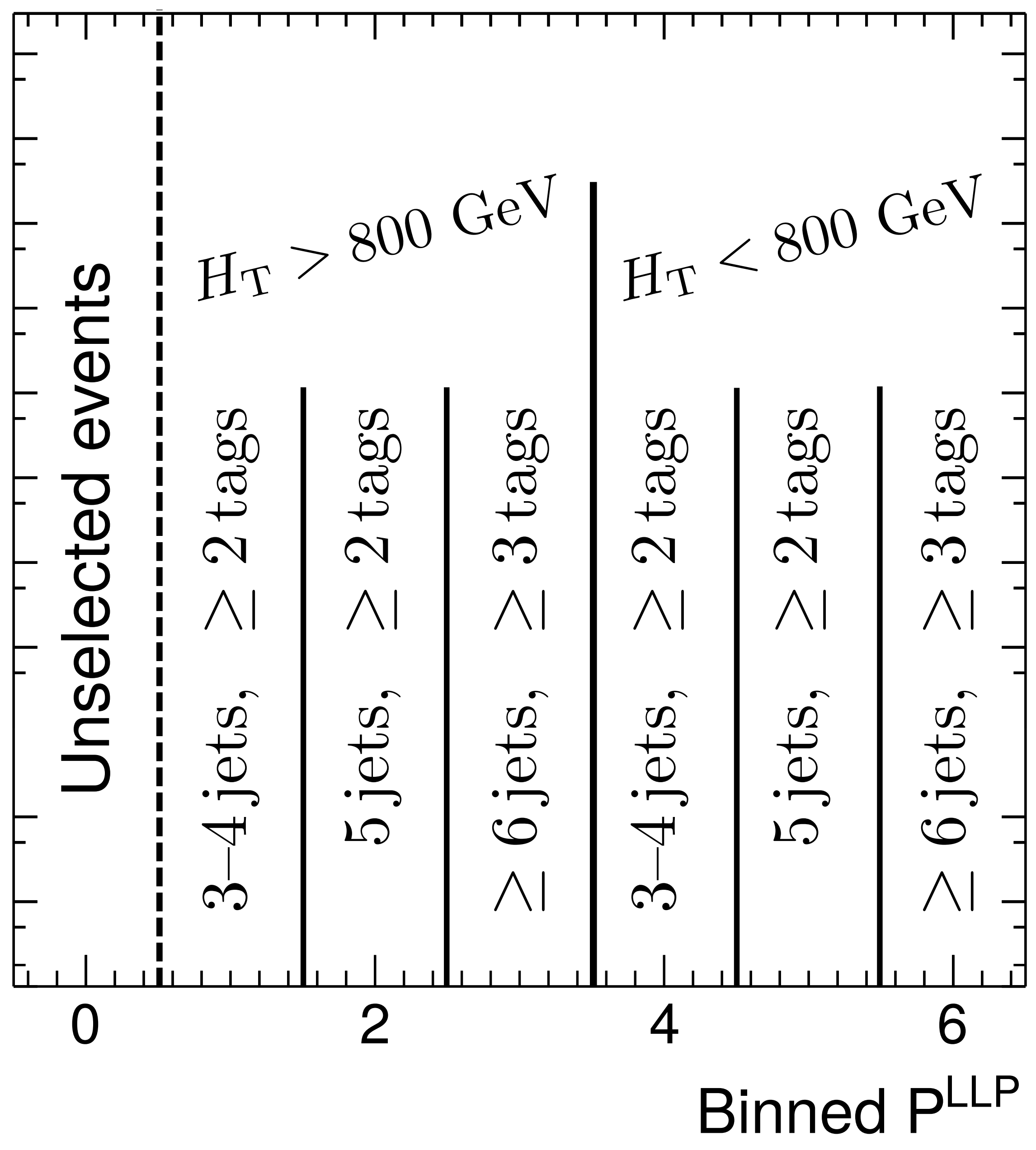


Showcase search for split SUSY

- Set expected limits on split SUSY gluino pair-production
  - Assuming a scenario of $36\ \mathrm{fb}^{-1}$ of 2016 data

- Event pre-selection
  - Veto electrons, muons, forward jets
  - ≥ 3 central jets: $p_{\mathrm{T}} > 30\ \mathrm{GeV}, |\eta| < 2.4$
  - Trigger plateau: $H_{\mathrm{T}} > 300\ \mathrm{GeV}, H_{\mathrm{T}}^{\mathrm{miss}} > 300\ \mathrm{GeV}, H_{\mathrm{T}}^{\mathrm{miss}}/p_{\mathrm{T}}^{\mathrm{miss}} < 1.25$
  - Reject dominant QCD multijet background: $\Delta\phi^*_{\mathrm{min}} > 0.2$

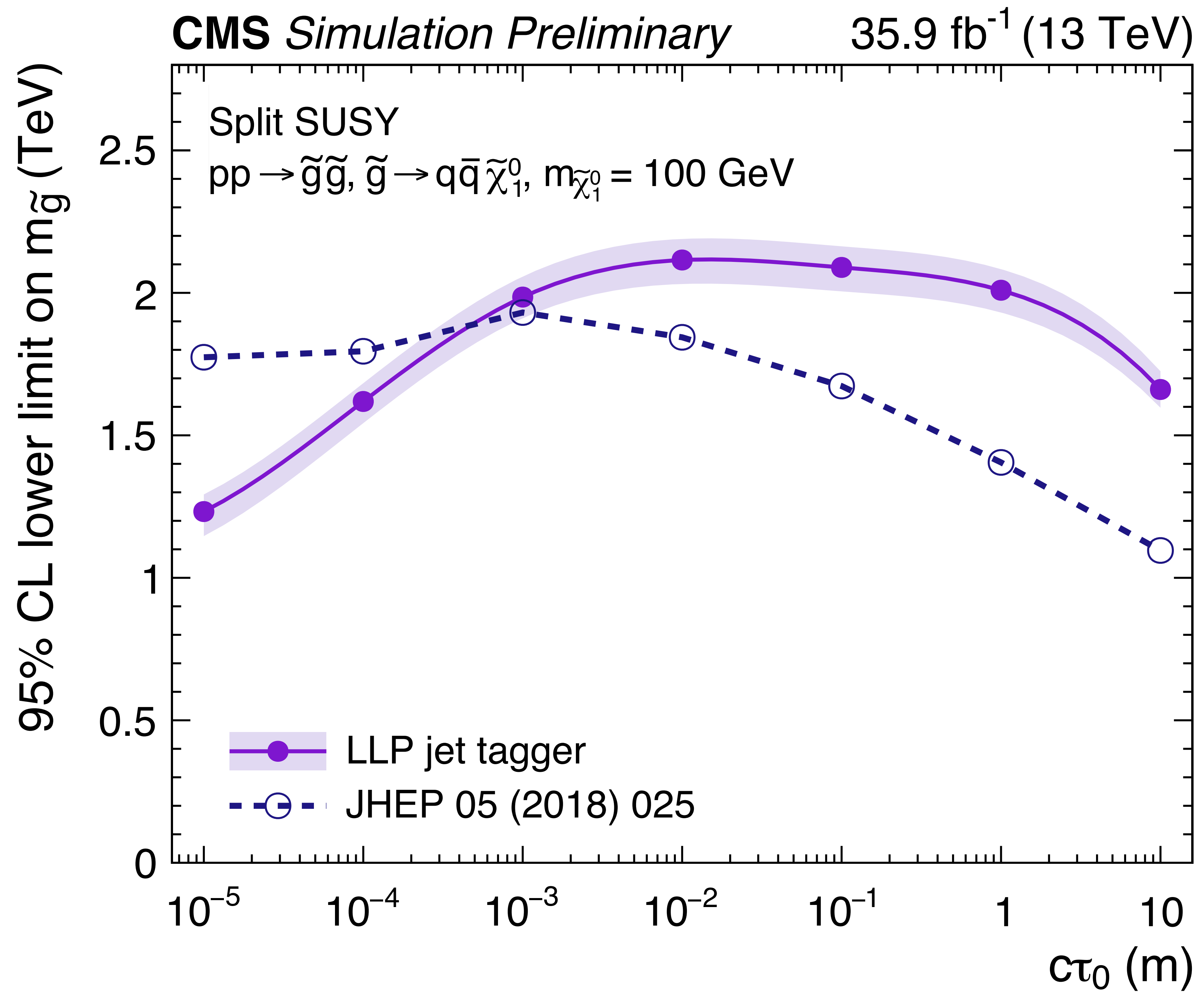
- Event categorisation
  - Separate compressed and uncompressed scenarios by H_{T}
  - Bin events according to $(N_{\mathrm{jet}}; N_{\mathrm{LLP}})$
  - Define conservative working point for each lifetime:
    - All bins must contain at least three expected events from MC
    - Manages dominant systematic uncertainty from finite MC statistics

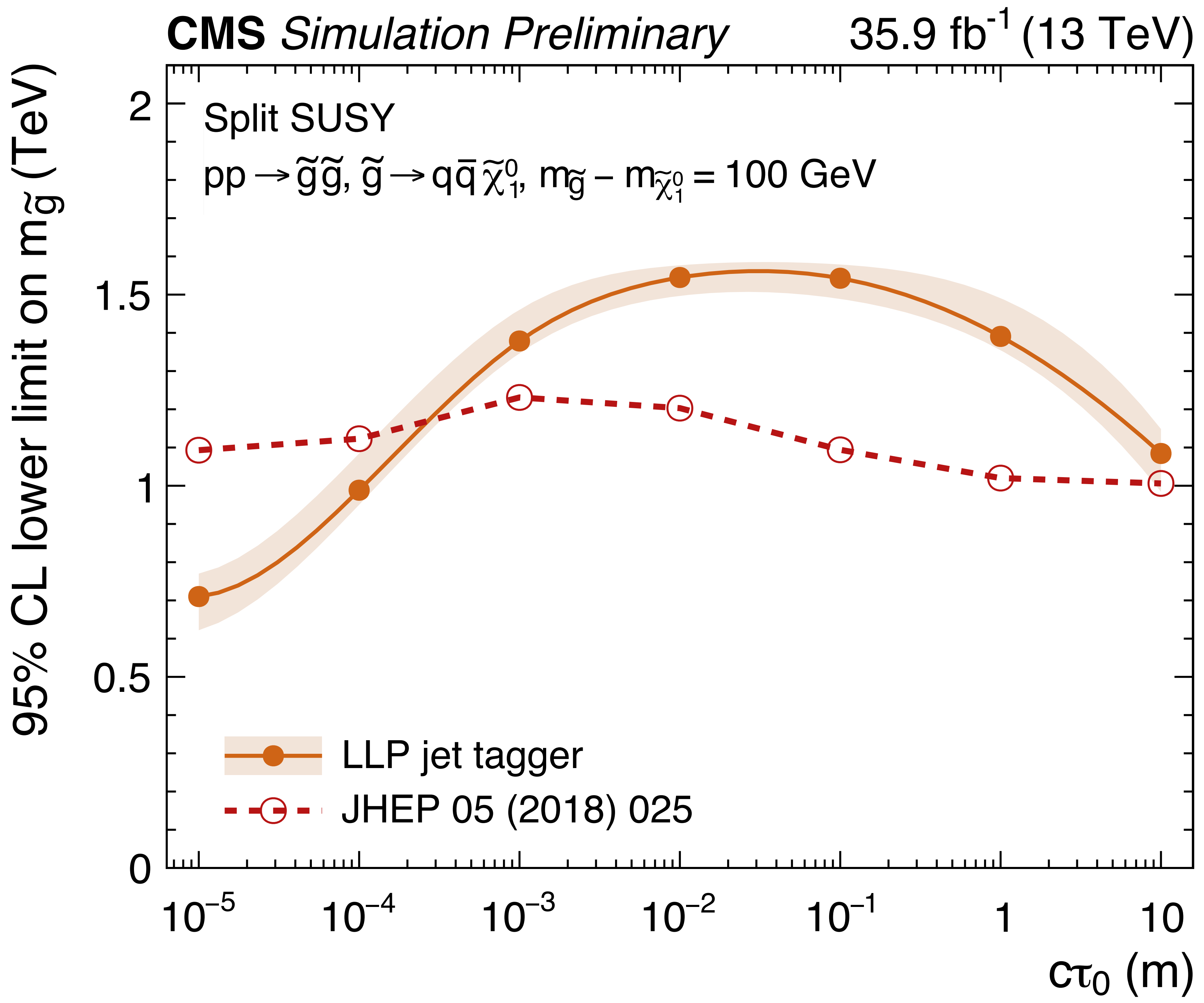


Results

- Compared with previous CMS search for natural SUSY
  - Interpreted with a split SUSY model: enhanced sensitivity for $c\tau = 1$ mm through b-tagging

- Competitive limits obtained
  - NB: simple event categorisation and substantial MC. stat uncertainties
  - Prompt lifetimes: need to exploit event kinematics as tagger performance degrades



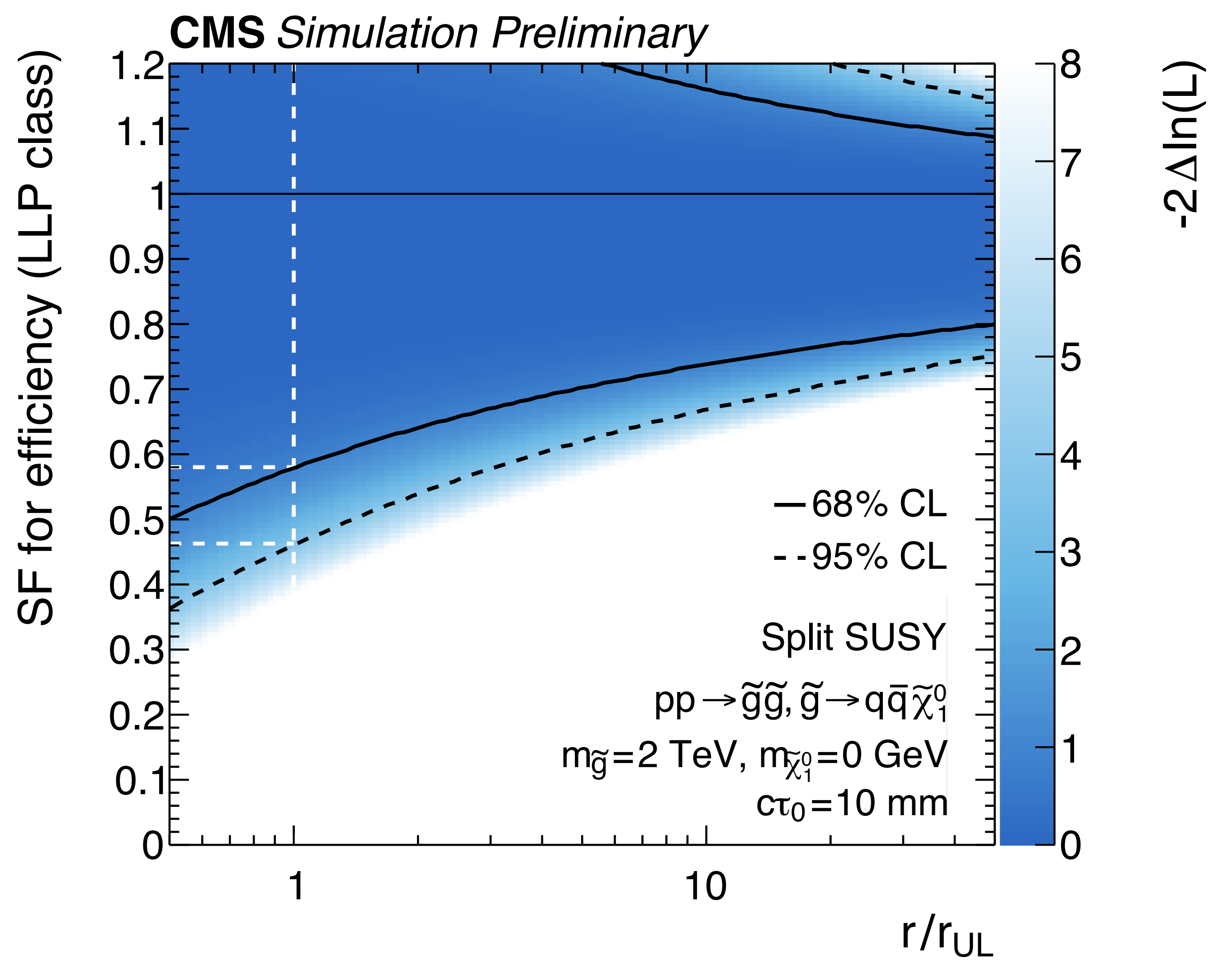


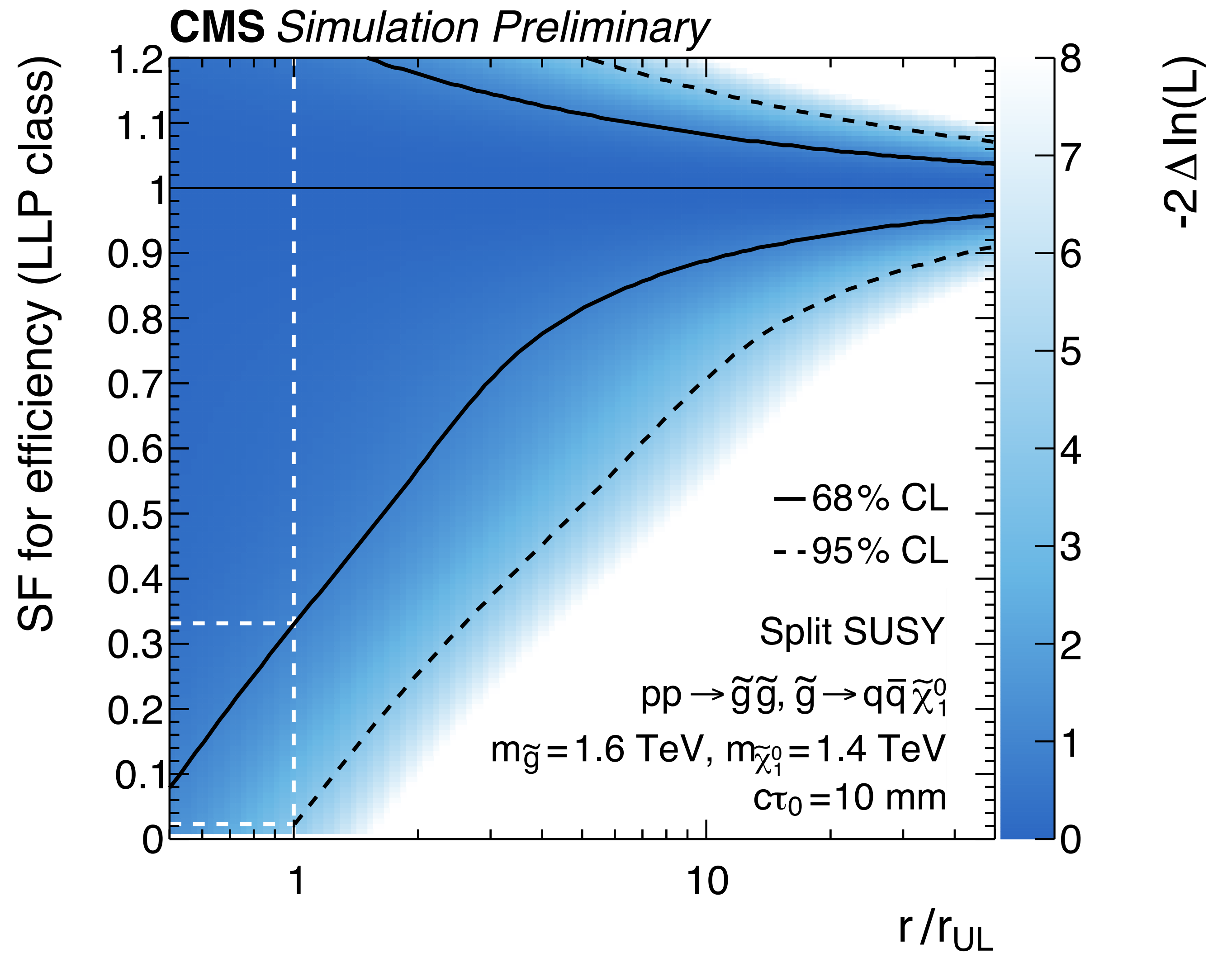
Tagging efficiency modelling

- Signal tagging efficiency in data (ϵ_{data}) a priori unknown
  - Efficiency is unconstrained if there is no signal ($r = 0$)
  - However, assume finite signal strength when setting upper limit, r_{UL}

- Novel procedure: treat tagging efficiency SF as a nuisance parameter
  - MC event weight: $$w = \left(\frac{1 - \text{SF} \cdot \epsilon_{\text{MC}}}{1 - \epsilon_{\text{MC}}}\right)^{\text{N(untagged)}} \times \left(\frac{\text{SF} \cdot \epsilon_{\text{MC}}}{\epsilon_{\text{MC}}}\right)^{\text{N(tagged)}}$$
  - Procedure depends on event categorisation by $(N_{\text{jet}}; N_{\text{tag}})$
  - Assumes $\text{SF}(p_{\text{T}}, \eta) = \text{const}\,.$

- Demonstrated in split SUSY showcase search:
  - SF constrained to ~60-80%





Profile SF for fixed r

Summary

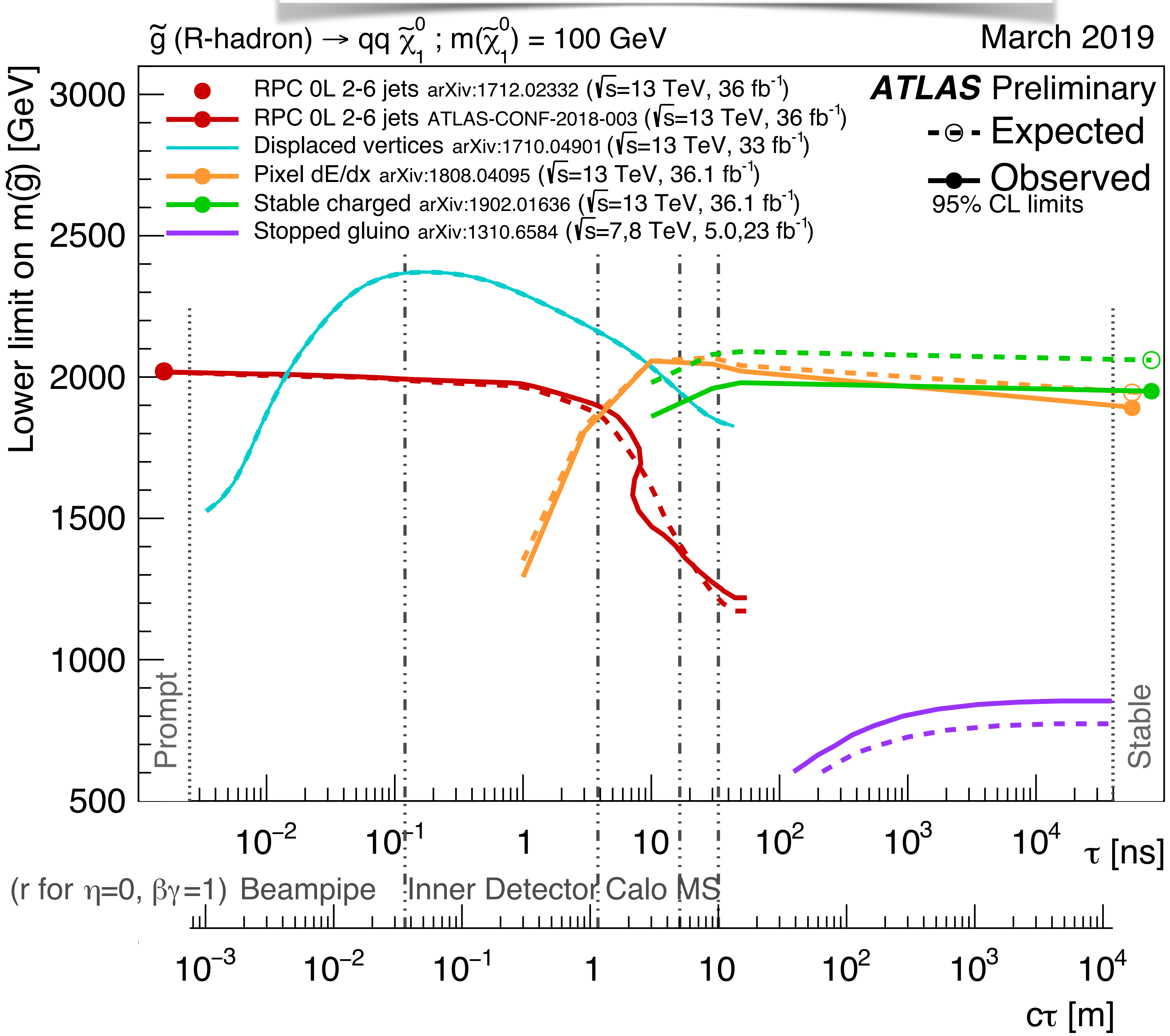
- Presented development of a novel displaced jet tagger
- New truth-level LLP jet definition facilitates training
- Training on CR data results in significantly better modelling of network features
  - Leads only to modest losses in performance (as measured in simulation)
- Tagger applied in a showcase search for split SUSY
  - Competitive limits obtained for 1 mm < $c\tau_0$ < 10 m
- Shows good potential for generalising to other models
- New procedure allows to constrain the signal tagging efficiency in-situ
  - Takes care of residual differences between data/MC after training with DA

PAS: EXO-19-011

Backup

Displaced jet searches

ATL-PHYS-PUB-2019-044



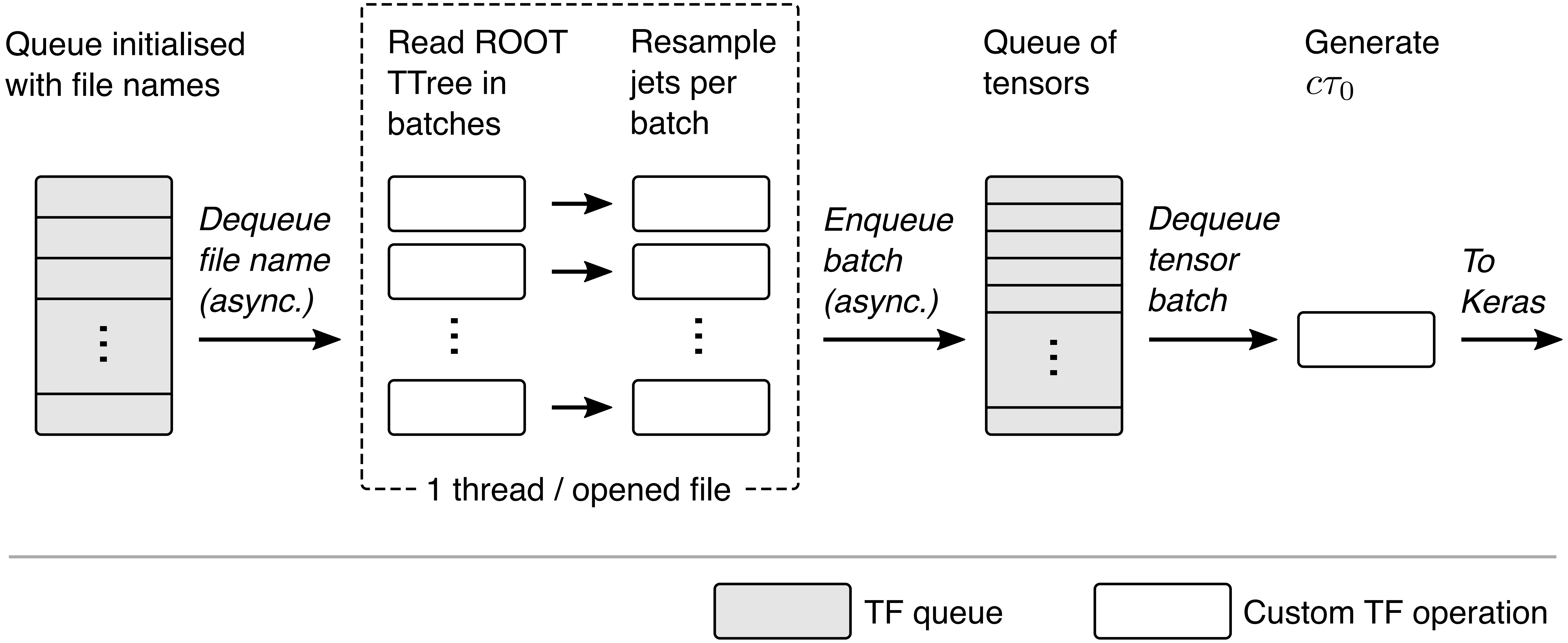
- Provide a variety of unconventional signatures in all parts of the detector

- Several complementary approaches: no search has best sensitivity for all lifetimes
  - Displaced vertices
  - Ionisation losses (dE/dx)
  - Stopped particles
  - Calorimetry timing
  - Decays in the muon systems
  - Re-interpretation of prompt searches

- Several challenges:
  - Non-standard (often sophisticated) reconstruction and identification algorithms
  - Difficulty in accurately estimating backgrounds

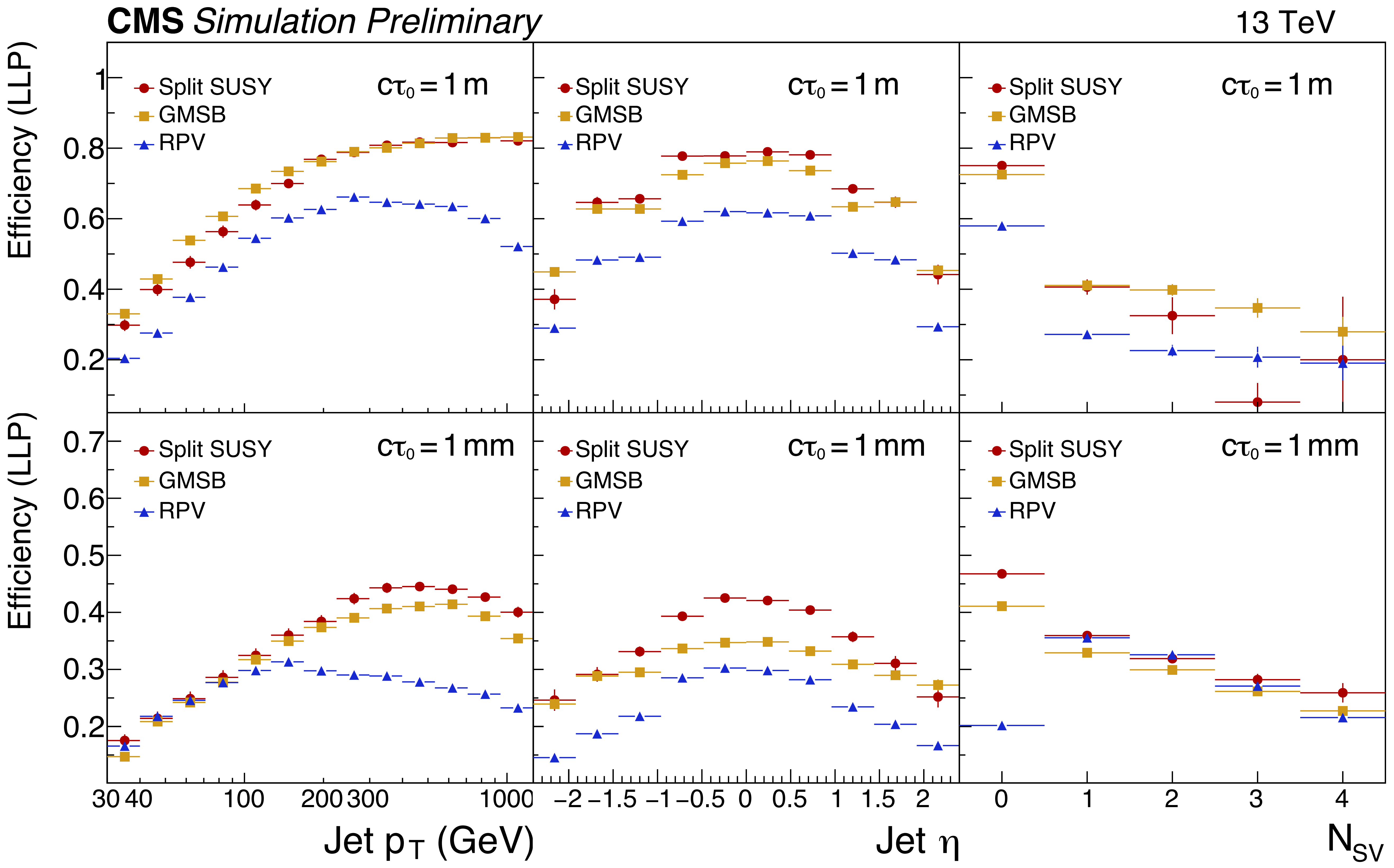
Input pipeline

- Based on a custom TensorFlow-to-ROOT interface and the TensorFlow queue system
  - Demo to be released at https://github.com/matt-komm/ROOT-TF-pipeline
- Performs jet resampling to ensure all classes have same p_{T}, η distribution when training
  - Works on-the-fly: network trained on different jets each epoch
- Generates fake $c\tau_0$ values for background jets by sampling from LLP jet distribution
  - No discrimination based on a latent variable, only through correlations with physical observables



Performance vs p_T, η, N_{SV}

- Apply same WP (0.01% mistag rate)
- Inspect performance for two benchmark lifetimes
  - $c\tau_0 = 1$ mm: exploit tracker
  - $c\tau_0 = 1$ m: exploit calorimeter systems
- Inspect regions of phase space:
  - Better performance for more energetic jets
  - Complementary to searches which require a secondary vertex
  - For RPV SUSY model, decreased performance for $N_{SV} = 0; p_T > 100$ GeV due to confusion between b and LLP classes



Performance vs $c\tau_0$, wrong lifetime test

- Apply a working point: 0.01% light flavour mistag rate
- Best performance for 1 mm < $c\tau_0$ < 10 m
  - Good performance for compressed scenarios with soft jets
- Network is parametric: needs to be supplied $c\tau_0$ to be evaluated
  - When testing a hypothesis, $c\tau_0^{\text{eval}} = c\tau_0^{\text{model}}$
- What happens if the “wrong” $c\tau_0$ is used instead?
  - Best performance when $c\tau_0^{\text{eval}} \rightarrow c\tau_0^{\text{model}}$
  - Potential to classify unknown arbitrary signal in data