Basic guidance on jet algorithms (& FastJet) for FCC-ee

FCC Physics Performance meeting, 27 June 2022

Matteo Cacciari, Gavin Salam and Gregory Soyez

Roles of jets at FCC-ee

- **Higgs physics**: reconstruct hadronic decays of H/Z$^{(*)}$/W$^{(*)}$ and separate them from backgrounds
  E.g. $e^+e^- \to HZ \to (bb)(jj)$ [v. Z→jets, WW→jets, ZZ→jets backgrounds]
- **Top physics:** reconstruct top quarks & separate from backgrounds + QCD studies
- **WW physics:** similar + QCD studies (e.g. colour reconnections)
- **Z physics:** QCD studies & QCD reference for other H/EW/top studies

DISCLAIMER

- Material presented here is based on 3 days' work last week (+ some prior thinking)
- We have focused on approaches we anticipate are well suited to FCC-ee (and ignored / given less priority to algorithms reputed to be pathological, e.g. JADE algorithm, or less suited to FCC-ee, e.g. standard pp algorithms & variants of them like Valencia)
- Case studies in part informed by suggestions from Emmanuel and Patrizia, and focus on reconstruction of hadronic decays of EW resonances

FCC-ee jet finding

and differences wrt other contexts

- Spherical symmetry:
  standard e^+e^- algorithms use distance measures based on energies of particles (E_i) and angles between particles (θ_{ij}); beam direction is not too special
  [LHC has ~ longitudinal boost invariance along beam direction → uses p_T and ΔR_{ij}]

- Absence of underlying event / pileup:
  Every hadron should end up in a jet
  [@LHC: huge amounts of pileup → jet finding is a compromise between losing hadrons from "signal" process, and gaining spurious hadrons from pileup and underlying event]
  [@CLIC&ILC: mild γγ pileup brings related considerations]

Preamble: event generation & analysis for plots

particle-level analyses (i.e. no detector sim.)

Simulation

- Pythia 8.303/8.306 (Monash13 tune), no detector simulation
- long-lived b-hadrons set to be stable
- no crossing angle

Analysis

- visible particles: all particles with $\theta_{i,\mathrm{beam}} > 0.154$, except neutrinos
- missing 4-momentum: total final-state momentum (incl. ν) minus visible 4-momentum
- isolated charged leptons:
  - electrons and muons with $E > 10\,\mathrm{GeV}$
  - with isolation condition: less than 5 GeV other energy within angle of 0.3 radians
- anything that is visible and not an isolated charged lepton is used as input for jet clustering
- if a jet contains one or more b-hadrons, it is considered to be b-tagged

Simplest all-round decent e^+e^- algorithm

the "exclusive" Durham k_t algorithm

- determine $d_{ij} = 2\min(E_i^2, E_j^2)(1 - \cos\theta_{ij})$ between each pair of particles i, j
- recombine i, j pair with smallest d_{ij}, and update all distances
- stop when:

  n$_{\text{jets}}$ mode. you have reached a predetermined number of jets (e.g. $n = 4$ for $ZH \to q\bar{q}b\bar{b}$)

  d$_{\text{cut}}$ mode. all remaining $d_{ij} >$ some threshold (called d_{cut})

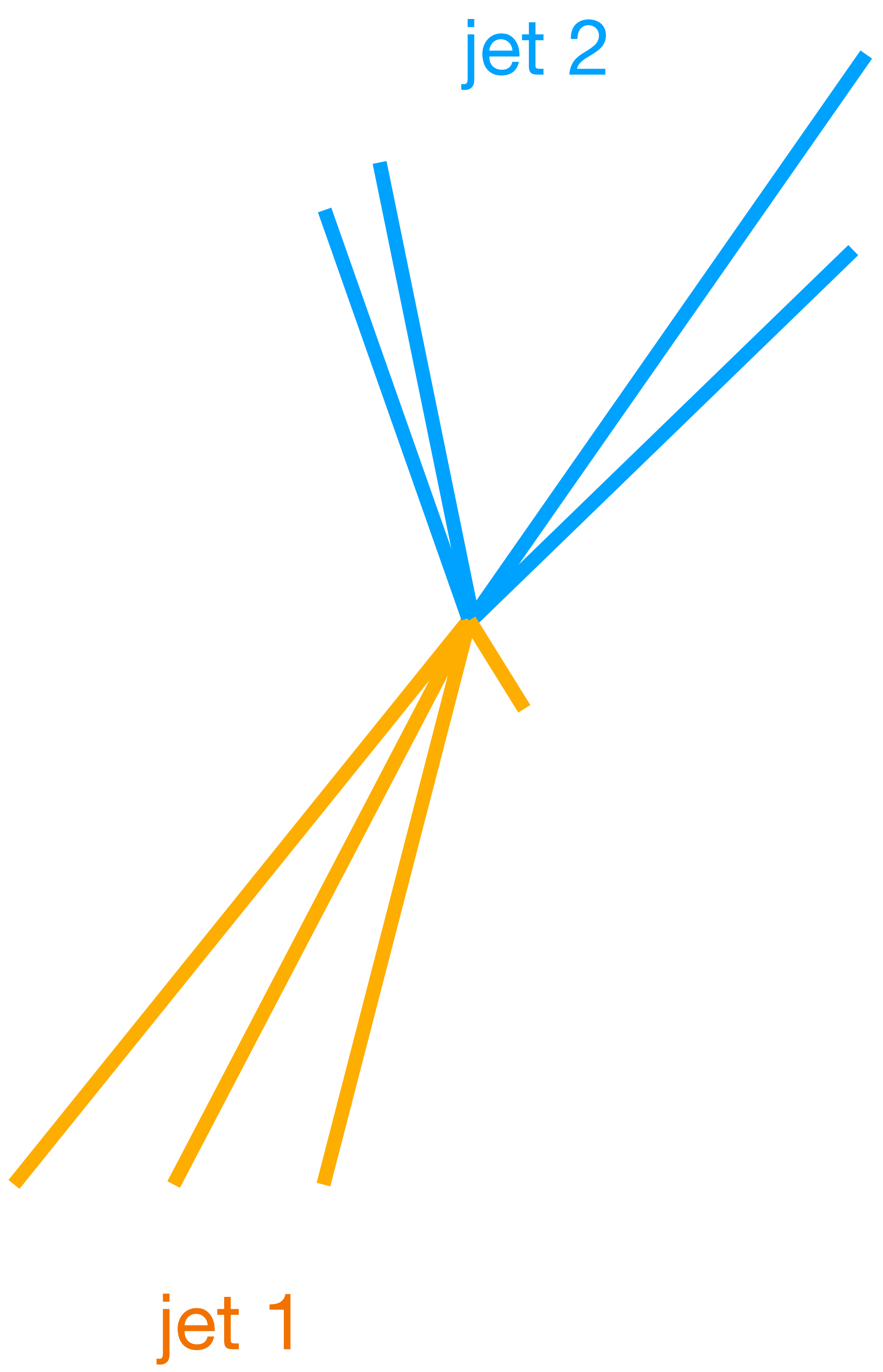
In "n$_{\text{jets}}$ mode", you often want to look at what the next d_{ij} *would* have been and discard the event if it is below some threshold. **Simple and effective.**

In "d$_{\text{cut}}$ mode", you usually make sure you have at least n jets for your process (e.g. 4 for $ZH \to q\bar{q}b\bar{b}$), otherwise discard the event. If you have more than n jets, decide whether to keep the event, and if so which jets to use.

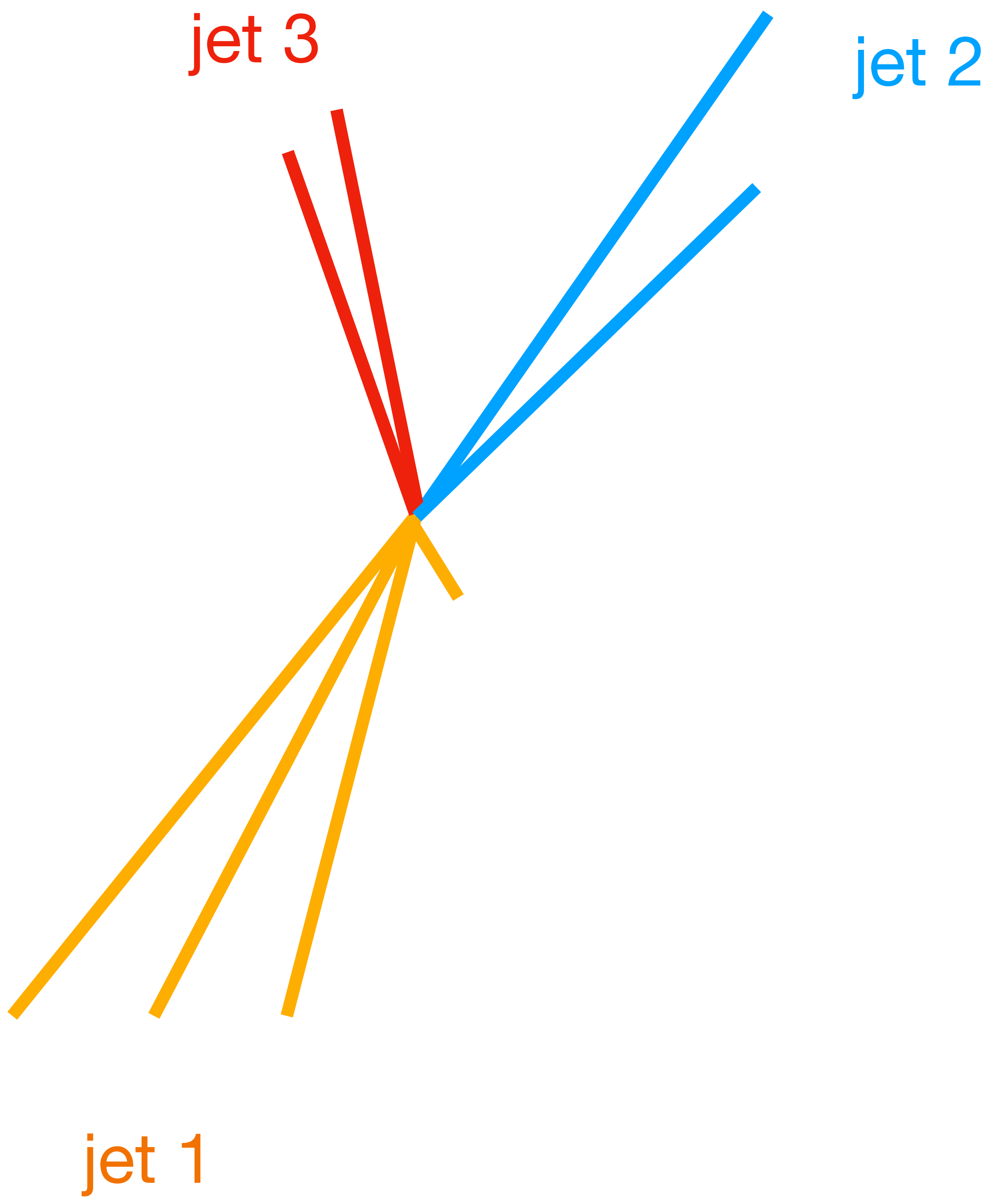
exclusive Durham-k_t example (n_{jets} mode)

stop when you have reached a predetermined number of jets

ask for 2 jets



ask for 3 jets



exclusive Durham-k_t example (n_{jets} mode)

stop when you have reached a predetermined number of jets

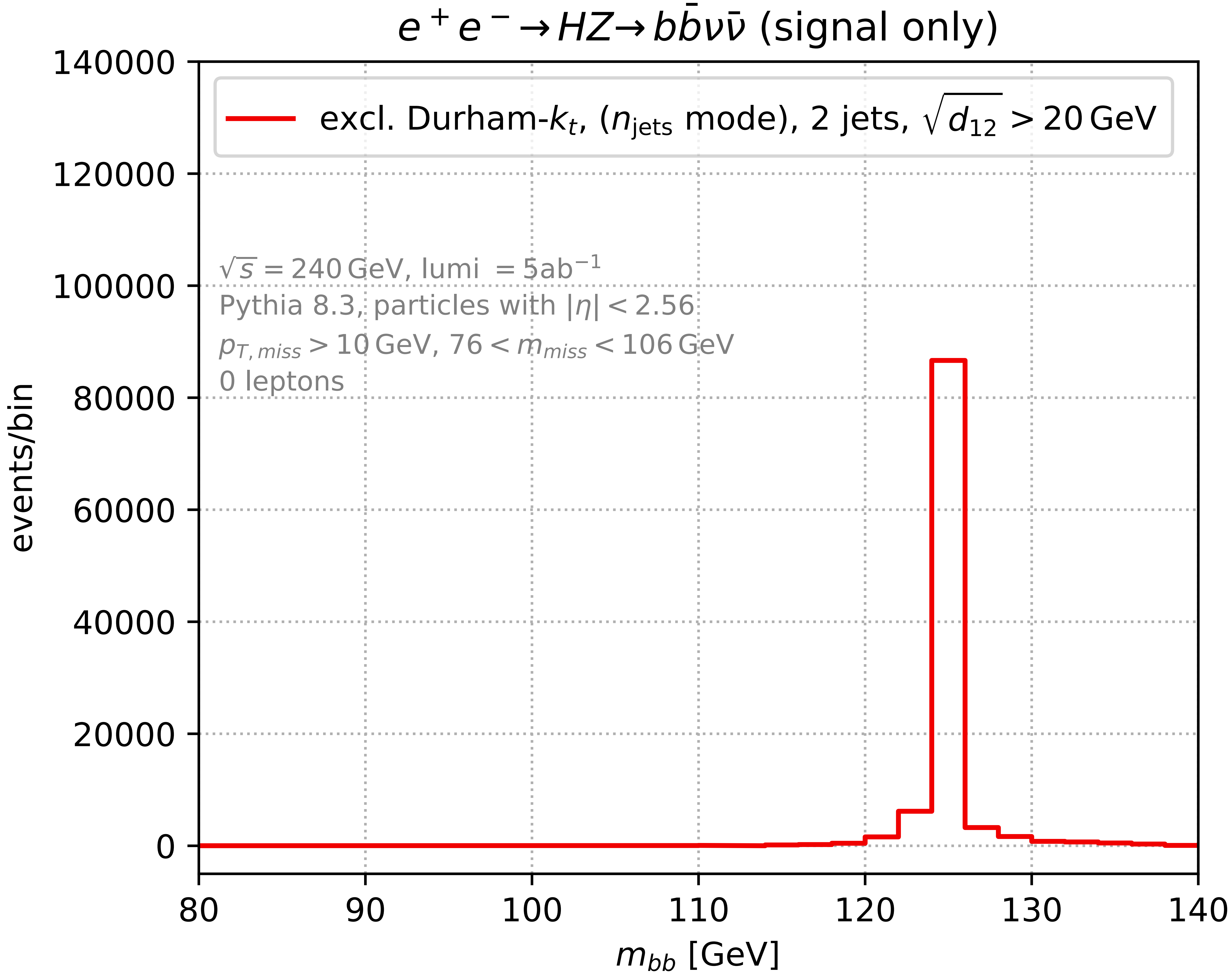
```
#include "fastjet/JetDefinition.hh"
using namespace std;
using namespace fastjet;

vector<PseudoJet> durham_kt_njets_mode(const vector<PseudoJet> & hadrons) {
  JetDefinition kt_jet_def(ee_kt_algorithm);
  ClusterSequence cs(hadrons, kt_jet_def);

  // try to break the event into 2 jets
  int njets = 2;
  vector<PseudoJet> jets = cs.exclusive_jets_up_to(njets);

  // you can also look at the merging scale (relative transverse
  // momentum of the merging) to go from njets to njets-1: this is a sensible
  // discrimination variable against backgrounds with
  // fewer LO jets;
  double rtd = sqrt(cs.exclusive_dmerge(njets-1));
  double rtd_min = 10.0;
  if (rtd < rtd_min) jets.clear();
  // return jets sorted into decreasing energies
  return sorted_by_E(jets);
}
```

Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



Excl. Durham-k_t ($n_{\rm jets}$ mode): Cluster to exactly two jets.

Require both jets to be b-tagged

Plot shows invariant mass of their 4-momentum sum (signal process only)

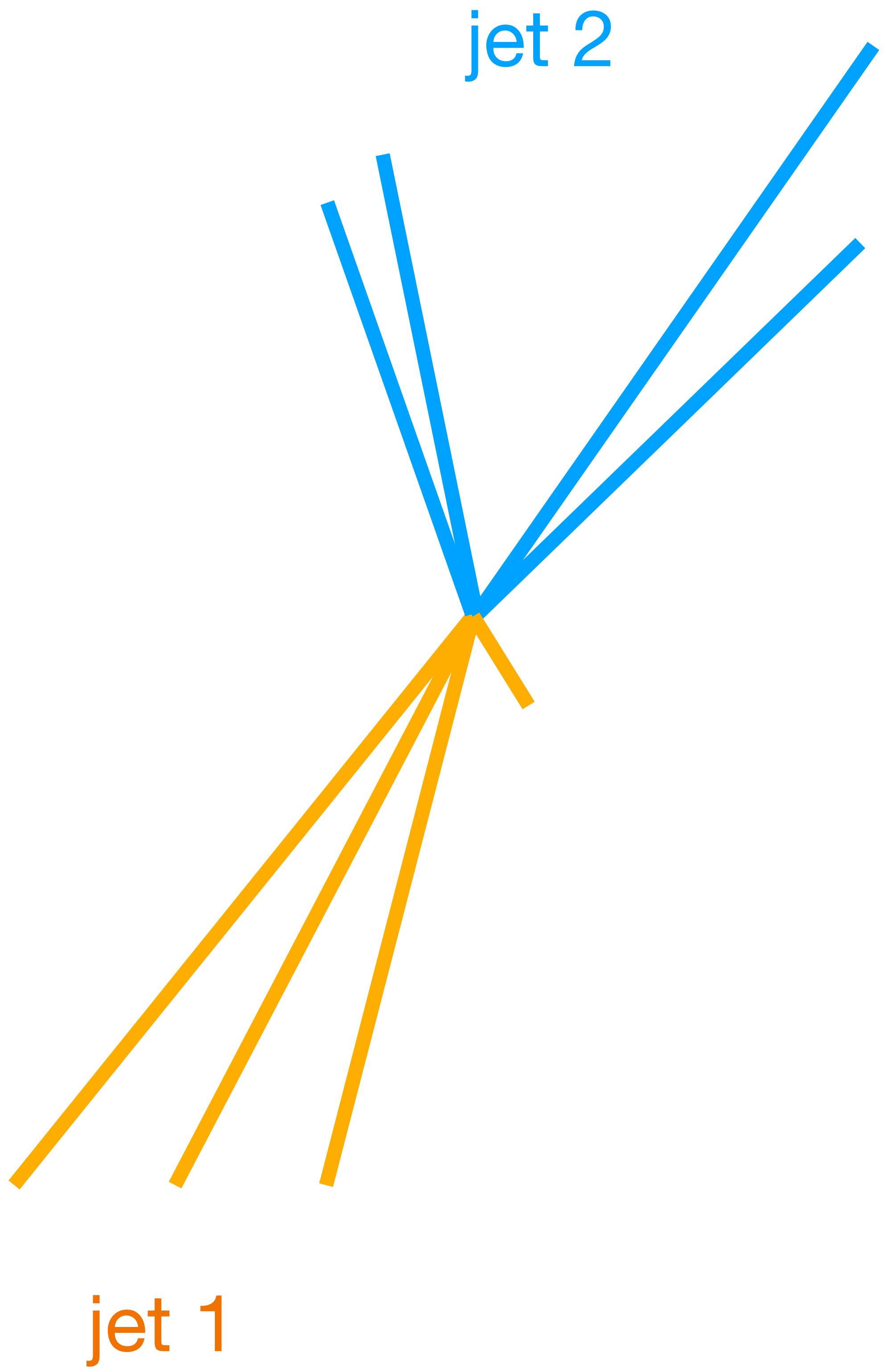
Almost all events are in the peak.

(NB: In ~3% of events, one of the jets not b-tagged, because both b's end up in a single jet)

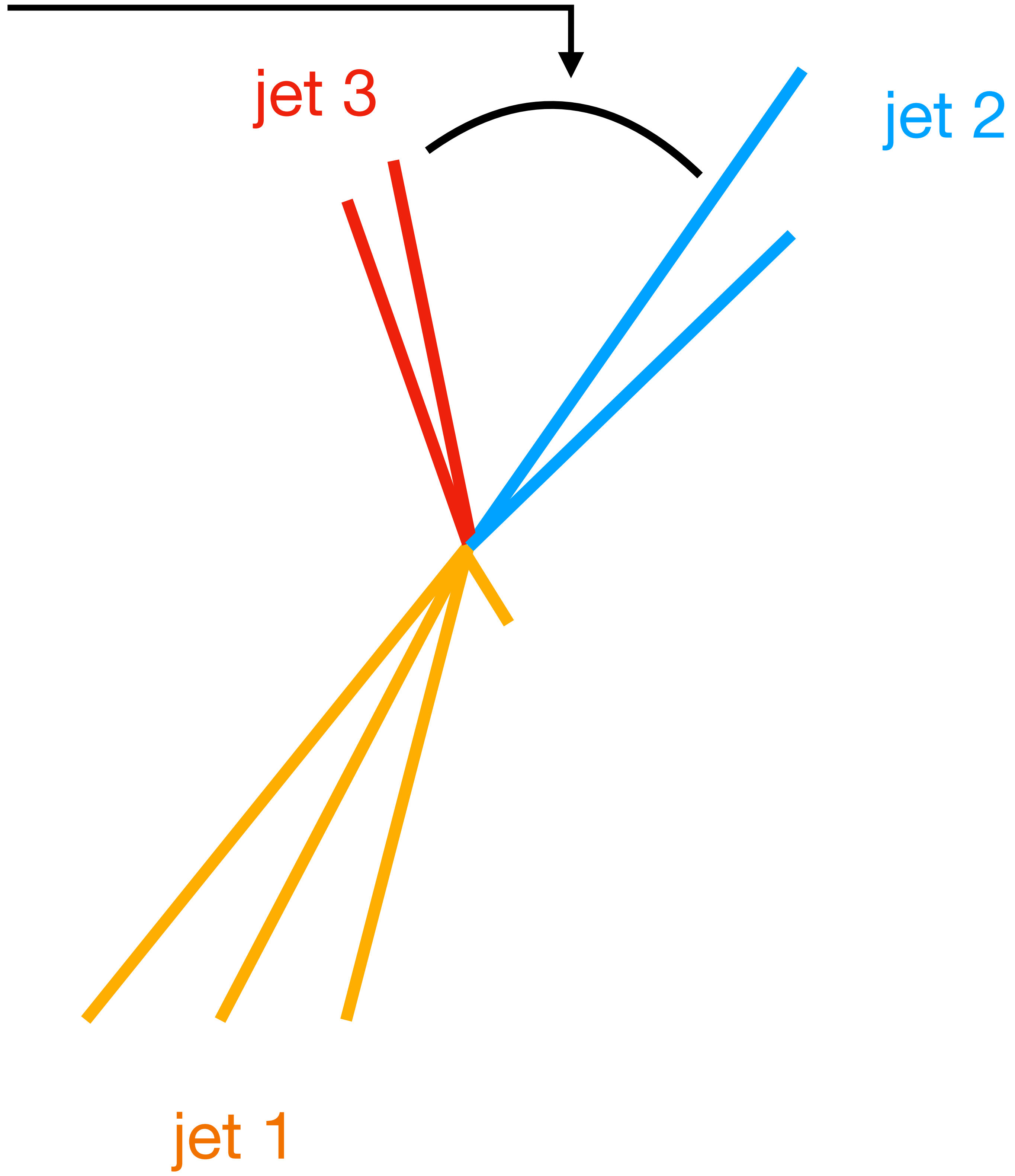
exclusive Durham-k_t example (d_{cut} mode)

stop when all d_{ij} above some threshold (d_{cut})

$$d_{cut} = (40\,\mathrm{GeV})^2$$



$$d_{cut} = (10\,\mathrm{GeV})^2$$



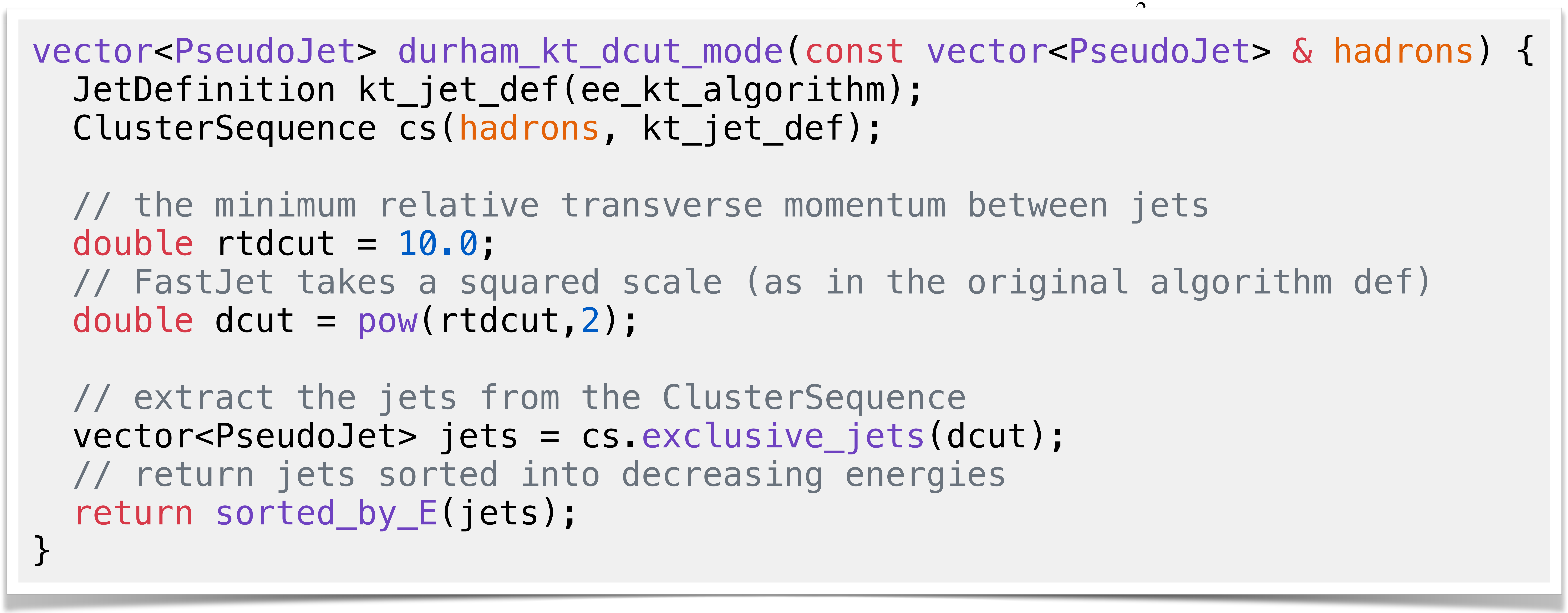
$d_{23} = (12\,\mathrm{GeV})^2$
$> d_{\mathrm{cut}}$ so 2 & 3
remain unmerged

exclusive Durham-k_t example (d_{cut} mode)

stop when all d_{ij} above some threshold (d_{cut})

$$d_{cut} = (40\,\mathrm{GeV})^2$$

$$d_{cut} = (10\,\mathrm{GeV})^2$$



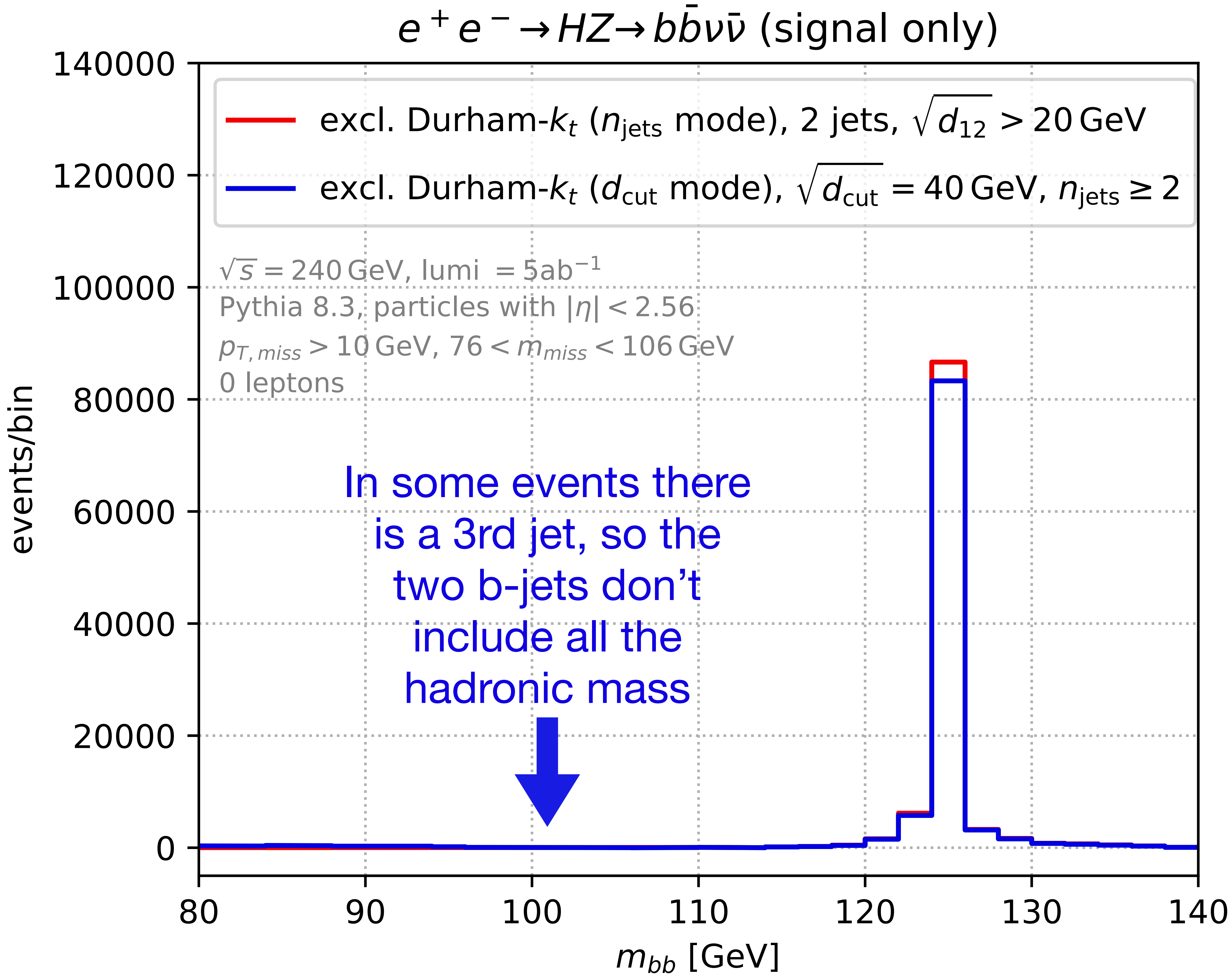
```
vector<PseudoJet> durham_kt_dcut_mode(const vector<PseudoJet> & hadrons) {
  JetDefinition kt_jet_def(ee_kt_algorithm);
  ClusterSequence cs(hadrons, kt_jet_def);

  // the minimum relative transverse momentum between jets
  double rtdcut = 10.0;
  // FastJet takes a squared scale (as in the original algorithm def)
  double dcut = pow(rtdcut,2);

  // extract the jets from the ClusterSequence
  vector<PseudoJet> jets = cs.exclusive_jets(dcut);
  // return jets sorted into decreasing energies
  return sorted_by_E(jets);
}
```

jet 2

Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



In some events there is a 3rd jet, so the two b-jets don't include all the hadronic mass

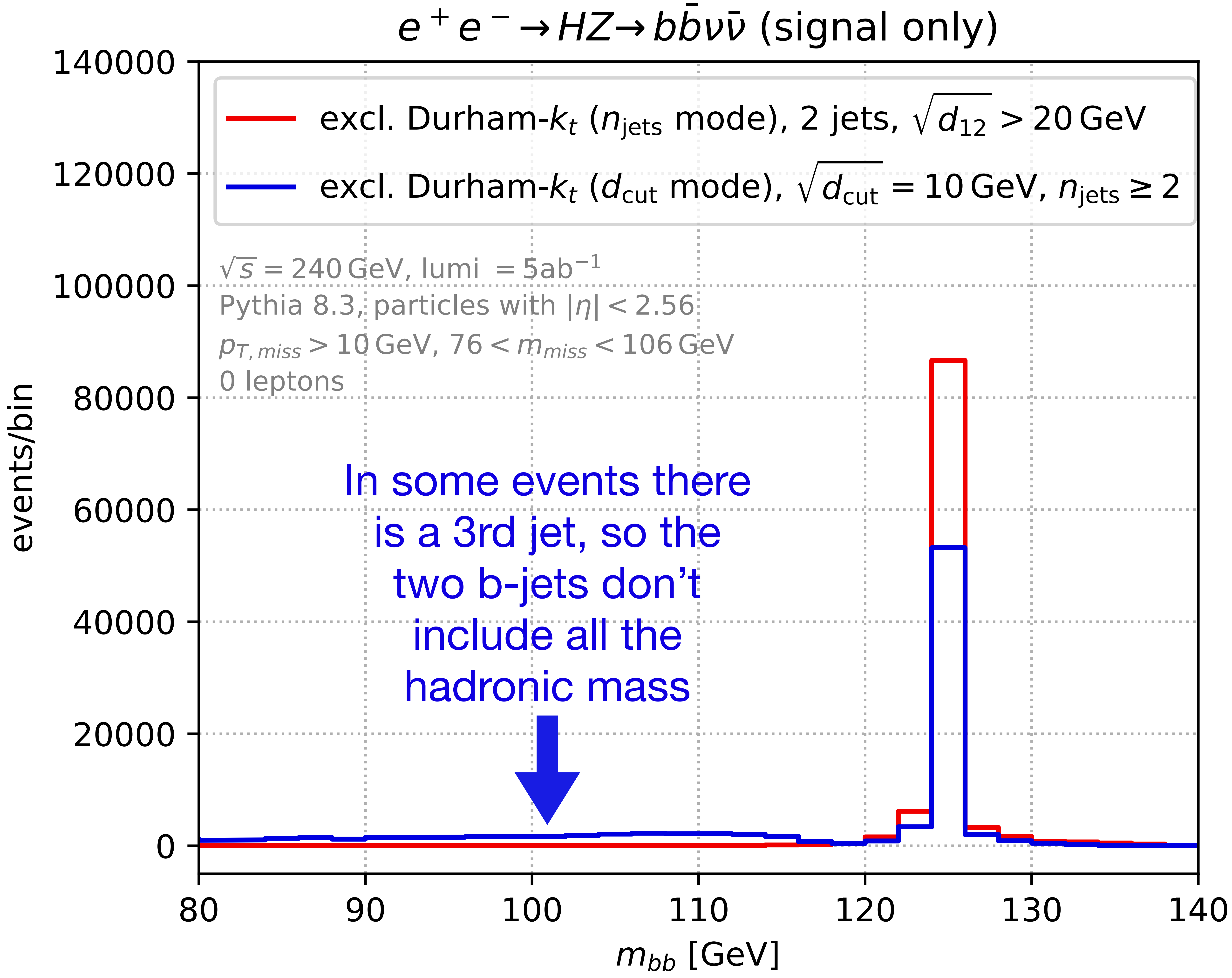
Excl. Durham-k_t (d_{cut} mode):
Stop clustering when all $d_{ij} > d_{cut}$

If there are 2 b-tagged jets, plot their invariant mass.

Hadronic energy may be spread across more than two jets, degrading mass resolution from the two b-jets.

The problem is worse with smaller d_{cut} values.

Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



In some events there is a 3rd jet, so the two b-jets don't include all the hadronic mass

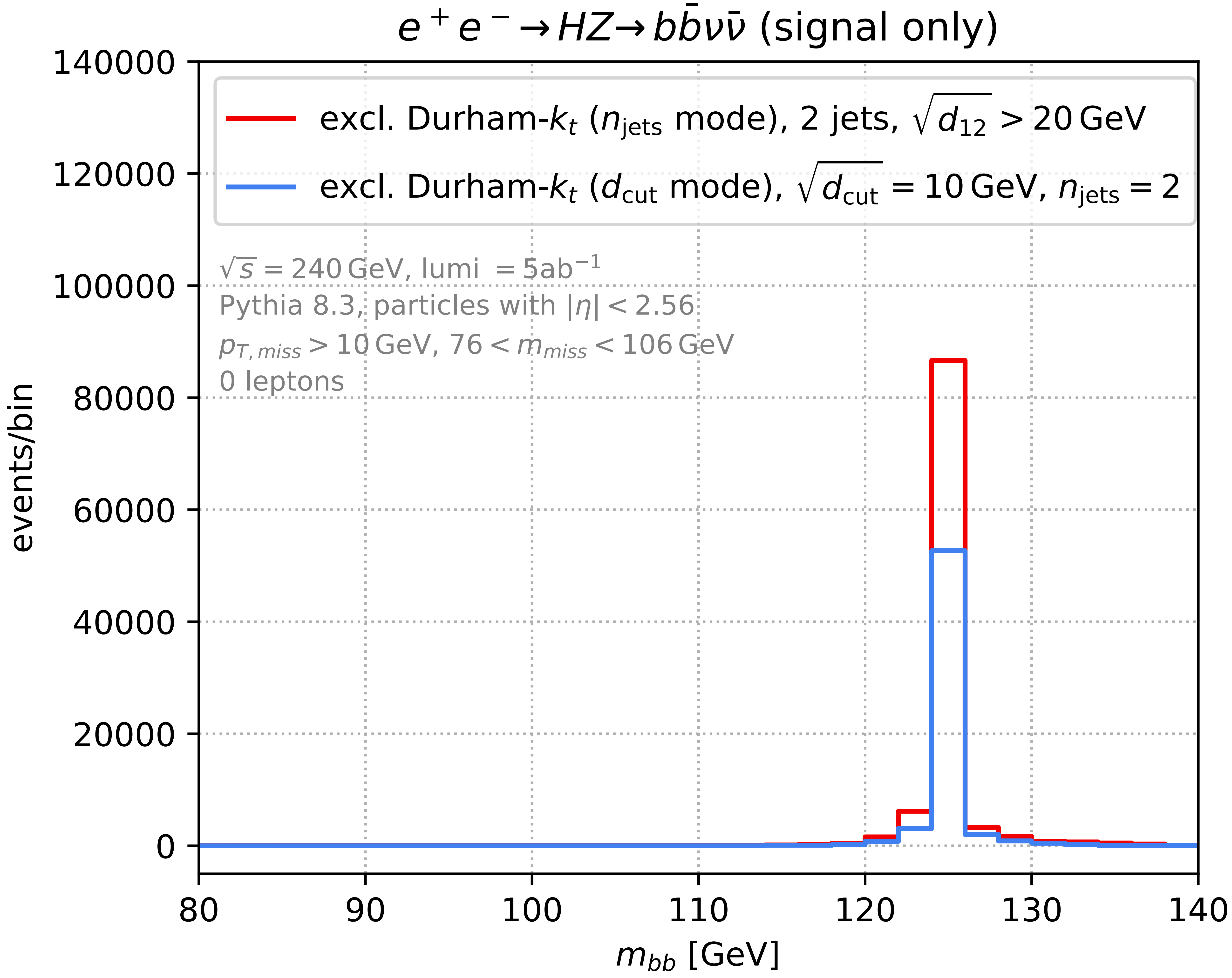
Excl. Durham-k_t (d_{cut} mode):
Stop clustering when all $d_{ij} > d_{cut}$

If there are 2 b-tagged jets, plot their invariant mass.

Hadronic energy may be spread across more than two jets, degrading mass resolution from the two b-jets.

The problem is worse with smaller d_{cut} values.

Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



Excl. Durham-k_t (d_{cut} mode):
Stop clustering when all $d_{ij} > d_{cut}$

If there are 2 b-tagged jets, plot their invariant mass.

Hadronic energy may be spread across more than two jets, degrading mass resolution from the two b-jets.

The problem is worse with smaller d_{cut} values.

Vetoing events with >2 jets removes tail to left but does not improve peak

e^+e^- generalised-k_t algorithm (“inclusive”)

with a jet radius R

- determine $d_{ij} = 2\min(E_i^{2p}, E_j^{2p})(1 - \cos\theta_{ij})/(1 - \cos R)$ between each i, j
- determine $d_{iB} = E_i^{2p}$ for each particle i
- if a d_{ij} is smallest, recombine i and j
- if a d_{iB} is smallest, i becomes a jet and is removed from list of objects

Jets tend to capture particles within angle R

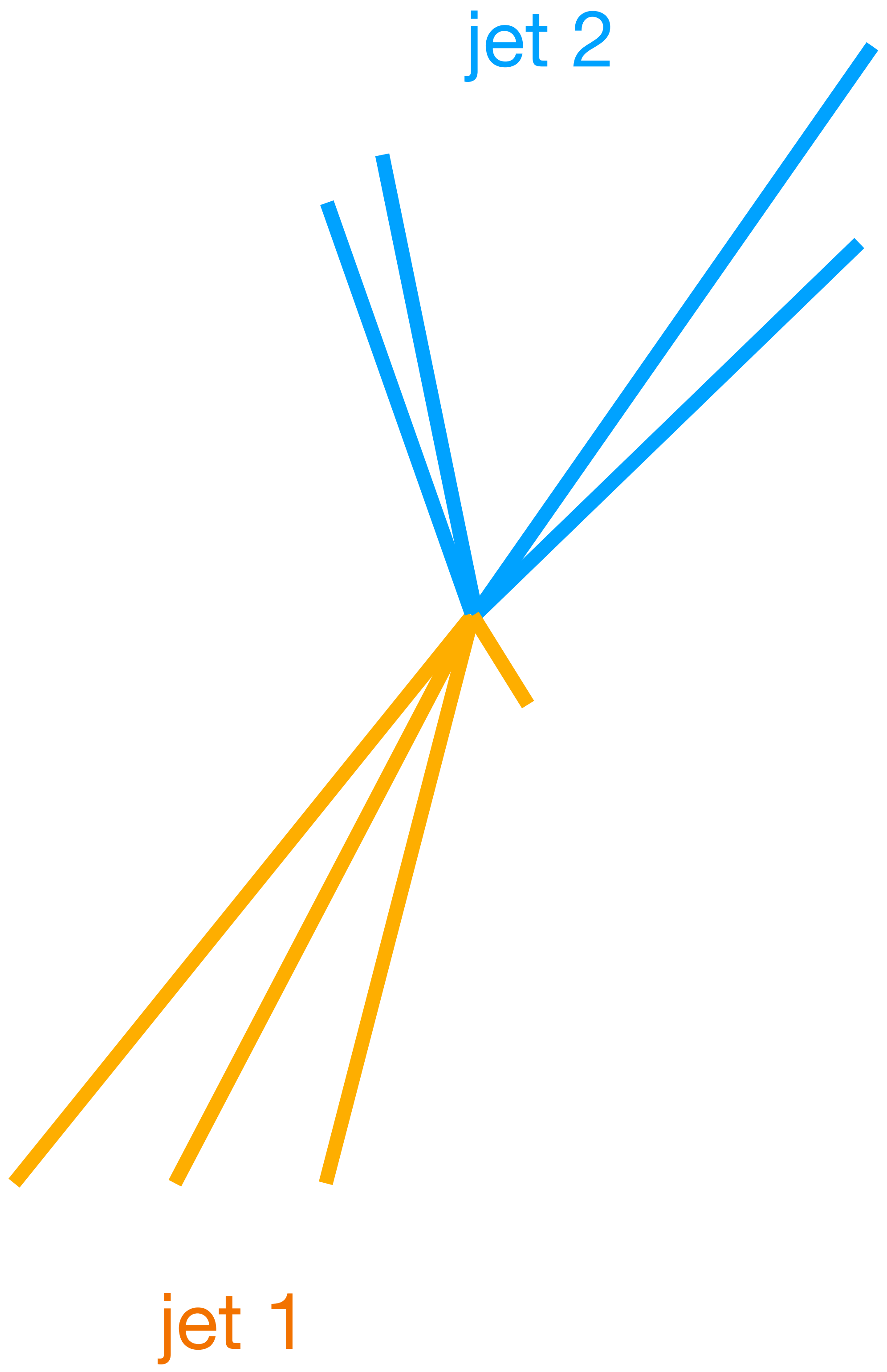
List of jets is only infrared safe if one applies an energy cut $E_{\text{jet}} > E_{\min}$

p=−1: inclusive anti-k_t circular jets
p= 0: inclusive Cambridge/Aachen good for jet substructure
p=+1: inclusive k_t

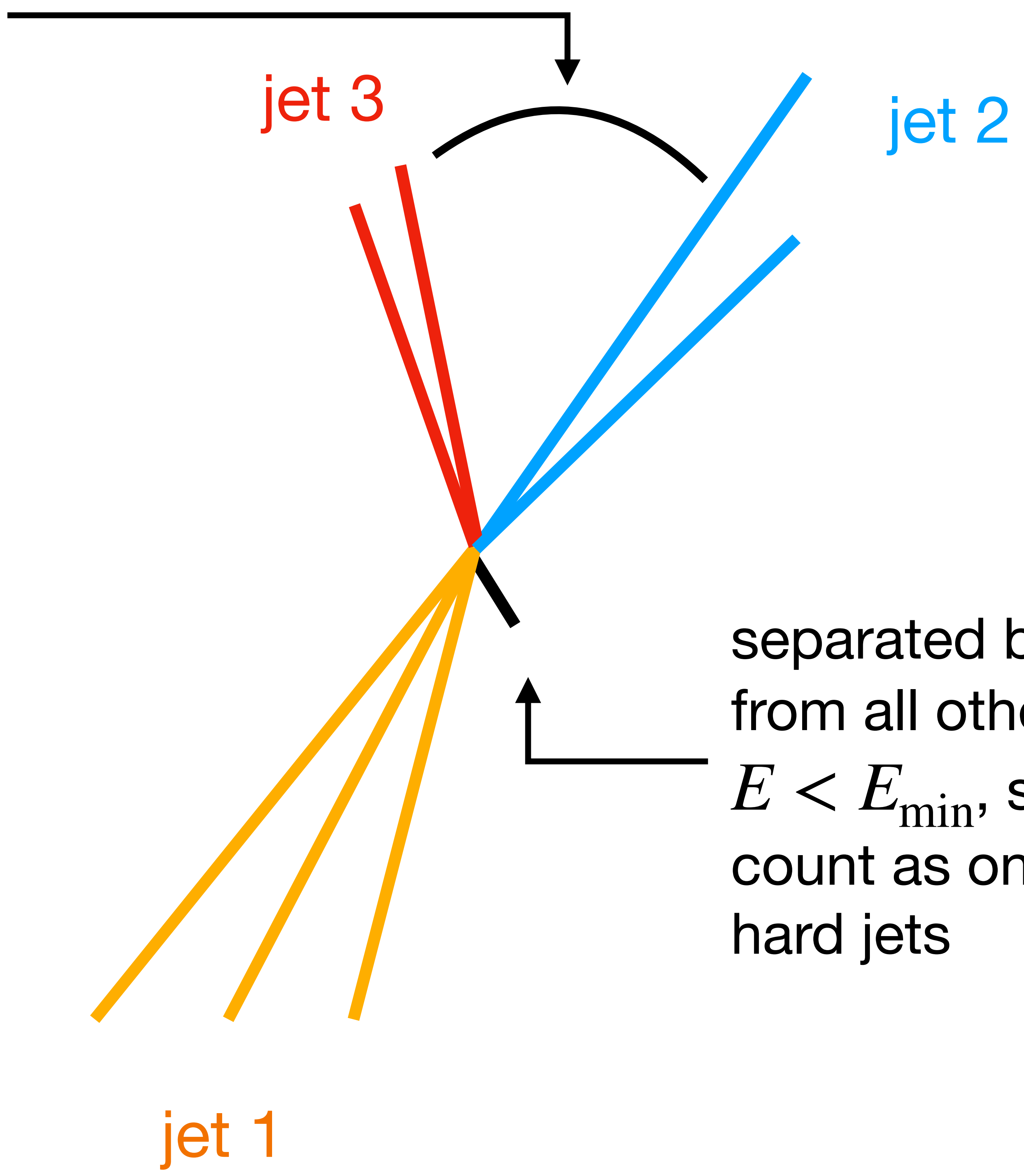
e^+e^- inclusive anti-k_t algorithm

has jet radius R; use with jet energy threshold (for n_{jets} to be infrared safe)

R=1.0, E_{min} = 10 GeV



R = 0.4, E_{min} = 10 GeV

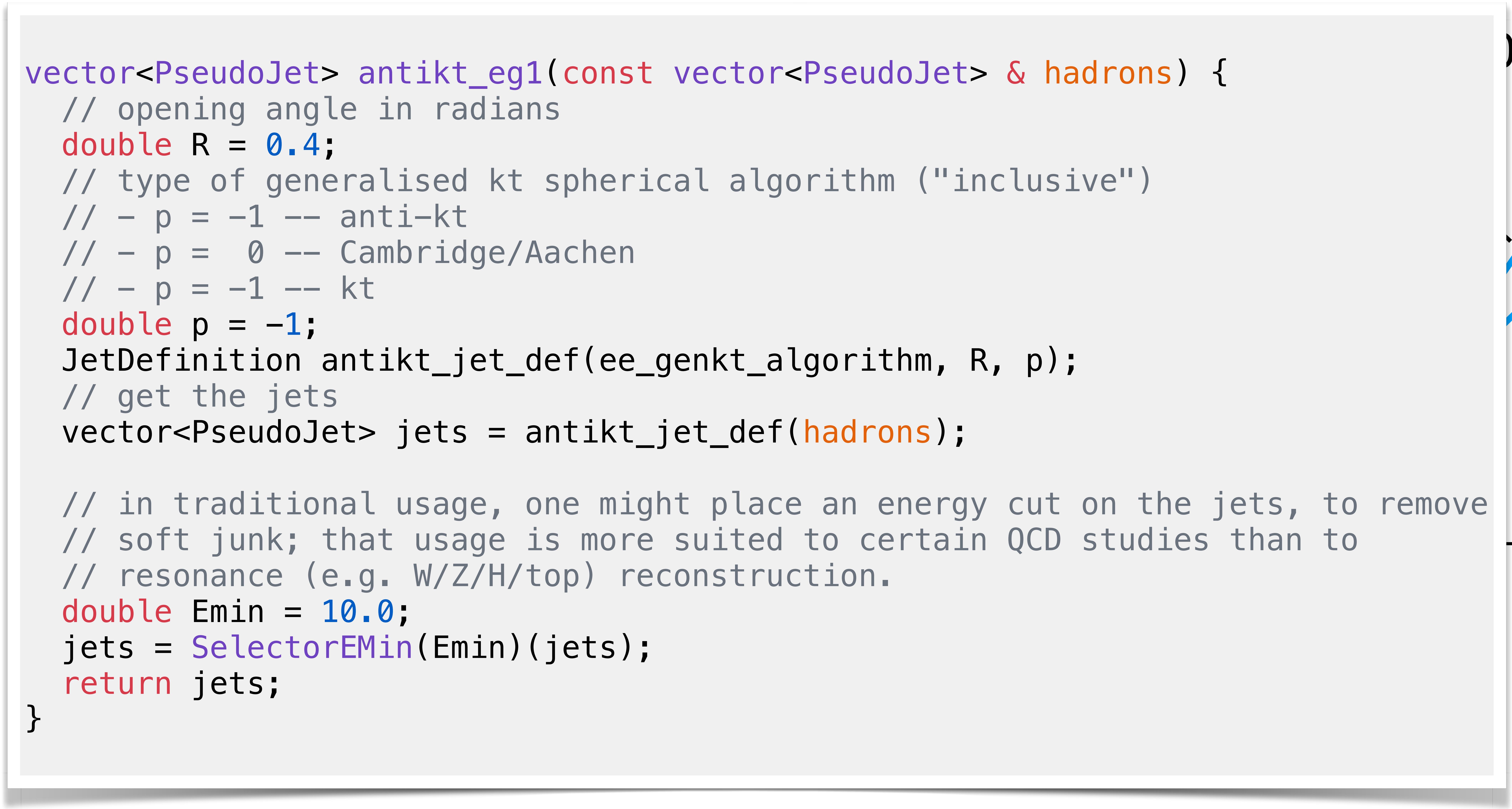


$\theta_{23} > R$
so resolve
separate jets

separated by $\theta > 0.4$
from all other jets;
$E < E_{\min}$, so does not
count as one of the
hard jets

e^+e^- inclusive anti-k_t algorithm

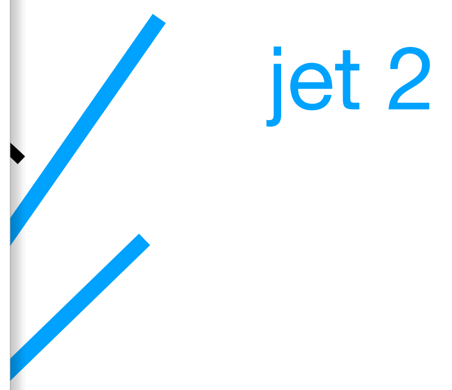
has jet radius R; use with jet energy threshold (for n_{jets} to be infrared safe)



```
vector<PseudoJet> antikt_eg1(const vector<PseudoJet> & hadrons) {
  // opening angle in radians
  double R = 0.4;
  // type of generalised kt spherical algorithm ("inclusive")
  // - p = -1 -- anti-kt
  // - p =  0 -- Cambridge/Aachen
  // - p = -1 -- kt
  double p = -1;
  JetDefinition antikt_jet_def(ee_genkt_algorithm, R, p);
  // get the jets
  vector<PseudoJet> jets = antikt_jet_def(hadrons);

  // in traditional usage, one might place an energy cut on the jets, to remove
  // soft junk; that usage is more suited to certain QCD studies than to
  // resonance (e.g. W/Z/H/top) reconstruction.
  double Emin = 10.0;
  jets = SelectorEMin(Emin)(jets);
  return jets;
}
```

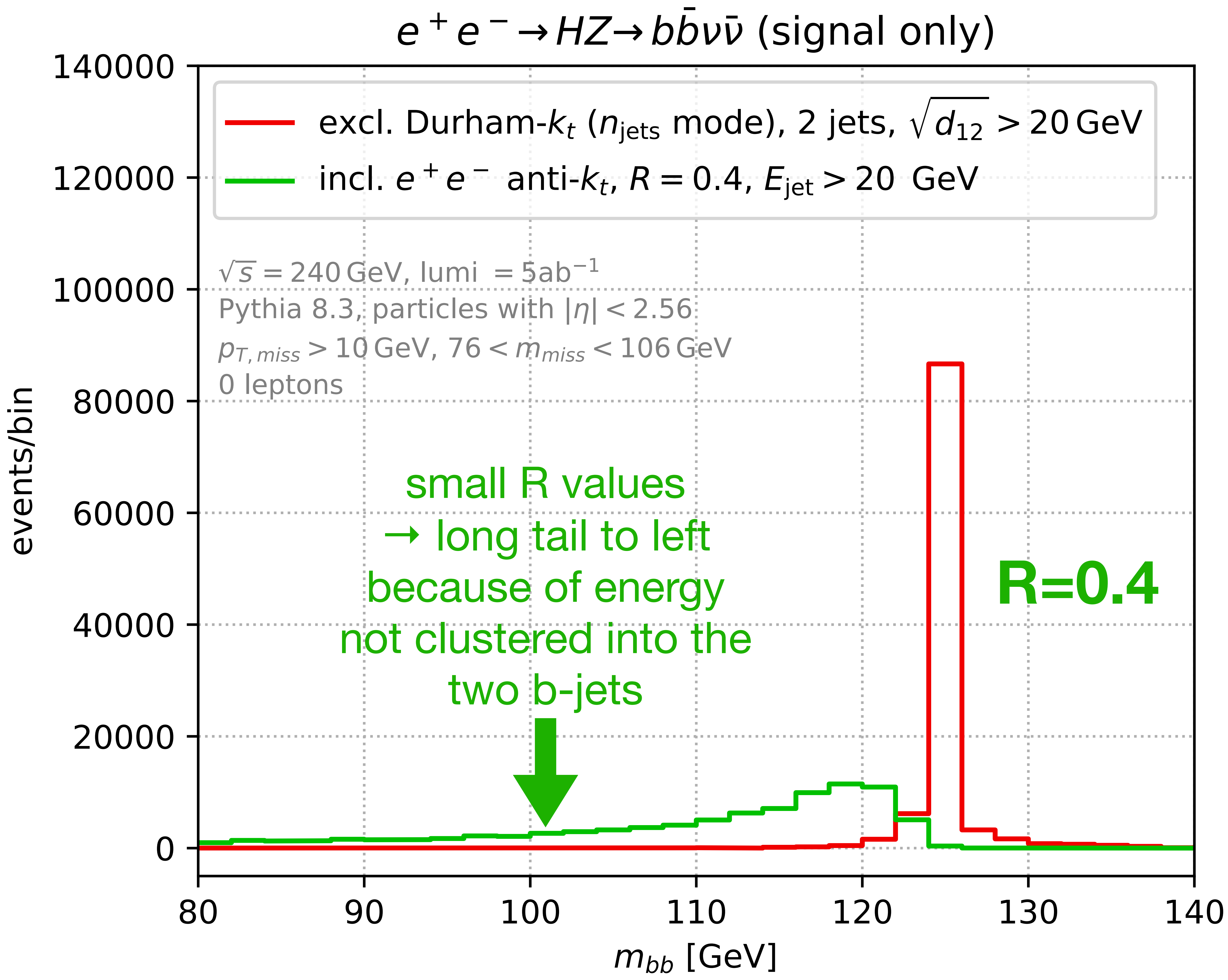
GeV



separated by $\theta > 0.4$ from all other jets;

$E < E_{\rm min}$, so does not count as one of the hard jets

Illustrate with e^+e^- → HZ → bbνν



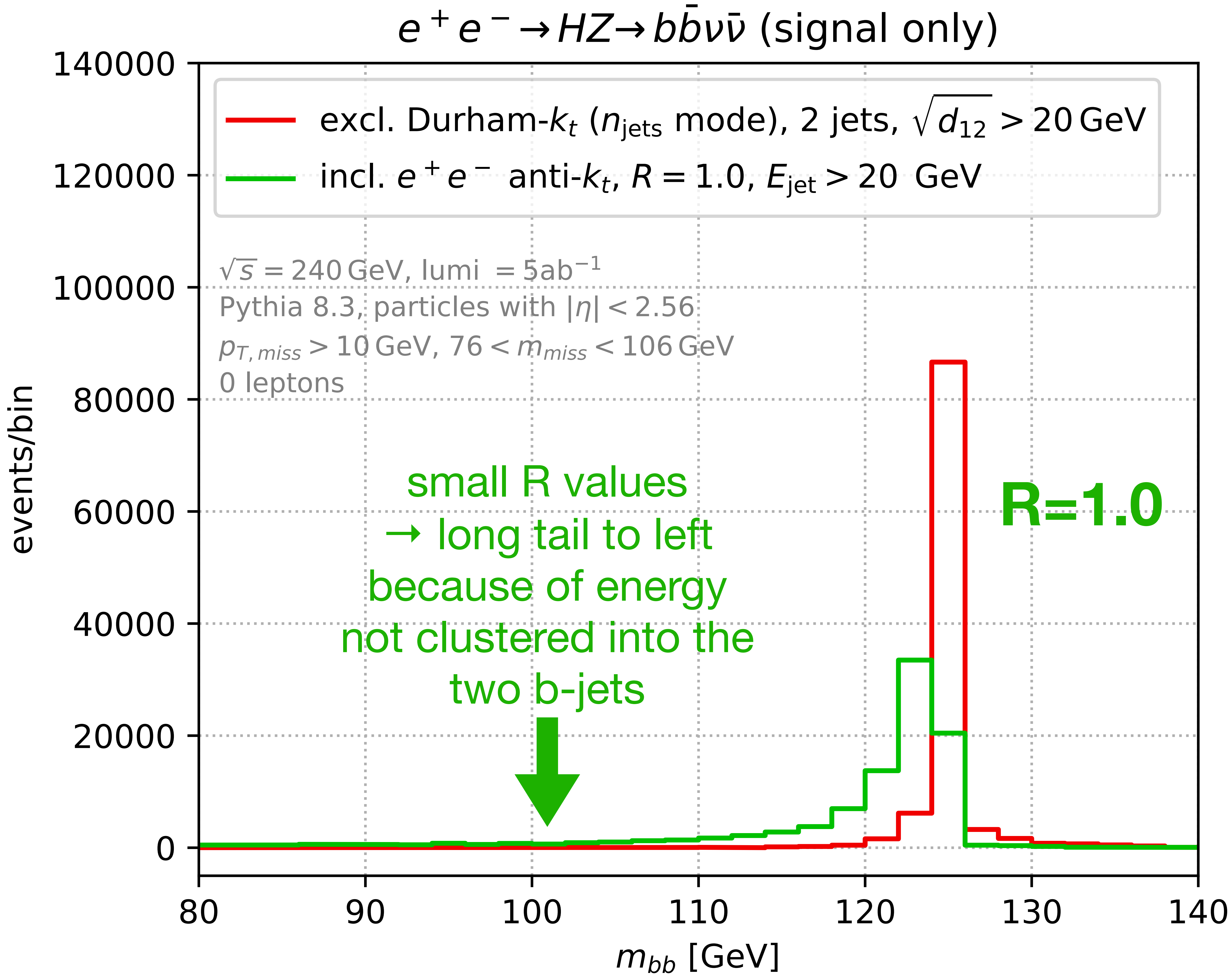
e^+e^- inclusive anti-k_t:
With some fixed jet radius and an energy threshold.

Require exactly two of the jets to be b-tagged jets & plot invariant mass.

Long tail to because of energy left out of the two b-jets. Mitigated when increasing jet radius.

Beware: large jet radius not ideal for multi-jet physics.

Illustrate with $e^+e^- \rightarrow HZ \rightarrow bb\nu\nu$



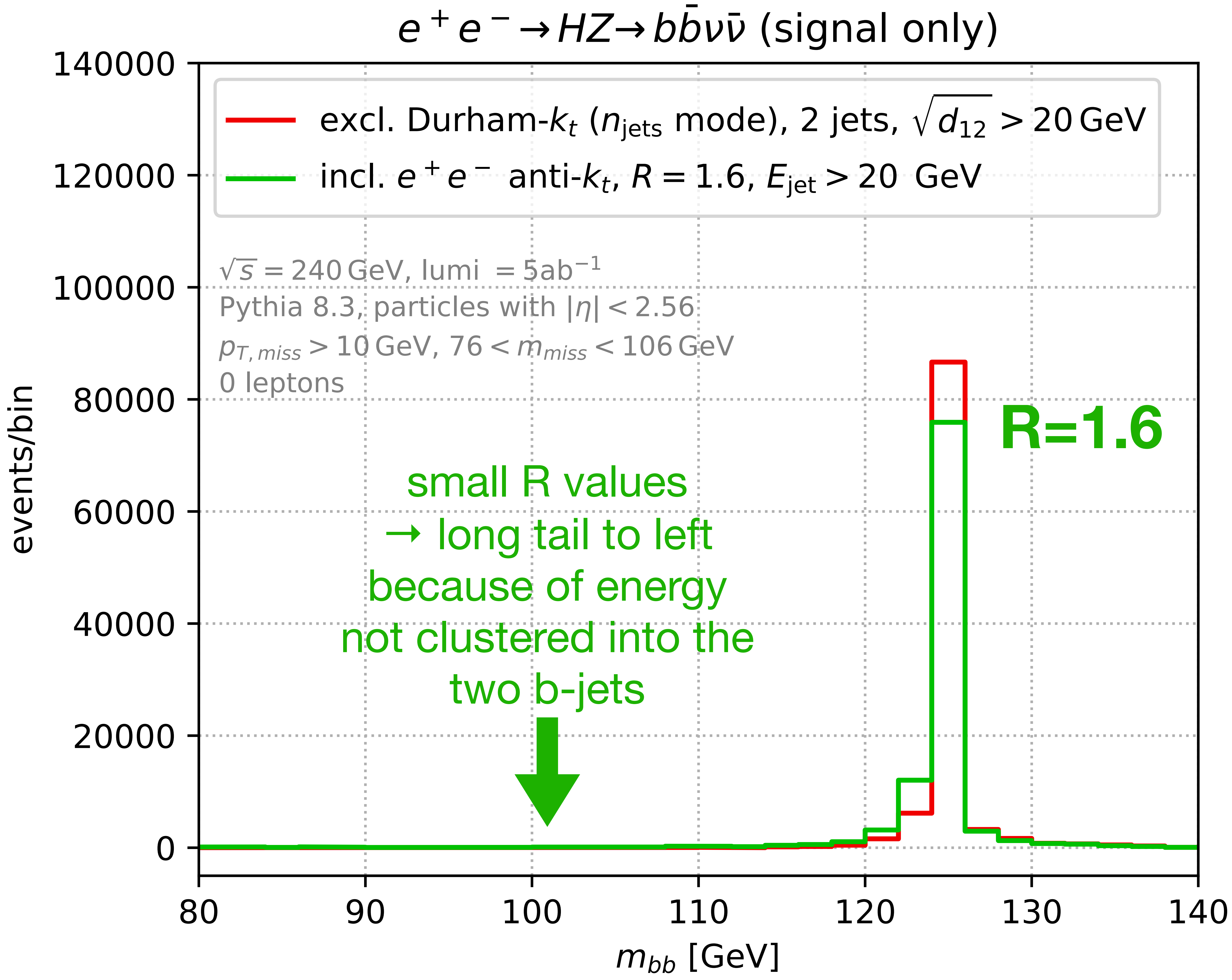
e^+e^- inclusive anti-k_t:
With some fixed jet radius and an energy threshold.

Require exactly two of the jets to be b-tagged jets & plot invariant mass.

Long tail to because of energy left out of the two b-jets. Mitigated when increasing jet radius.

Beware: large jet radius not ideal for multi-jet physics.

Illustrate with e^+e^- → HZ → bbνν



e^+e^- inclusive anti-k_t:
With some fixed jet radius and an energy threshold.

Require exactly two of the jets to be b-tagged jets & plot invariant mass.

Long tail to because of energy left out of the two b-jets. Mitigated when increasing jet radius.

Beware: large jet radius not ideal for multi-jet physics.

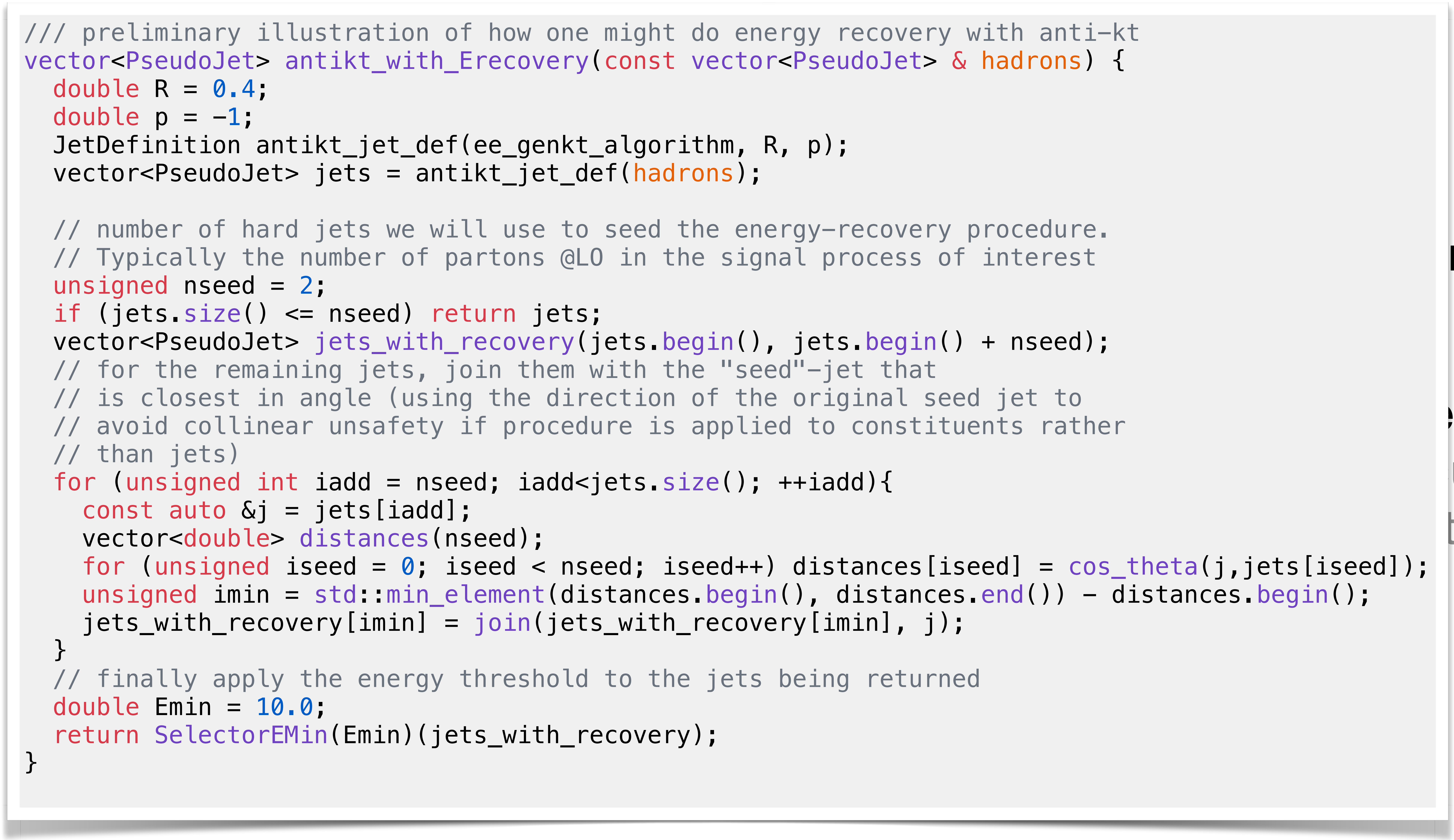
PRELIMINARY: is there a way of using the inclusive algs?

Energy recovery, when you know you are looking for *n* jets

- cluster with your preferred choice of *R*, and no energy threshold
- separate jets into *n* highest energy ones and the remaining (lower-energy) ones
- for each lower energy jet, “join” it with the jet from the “highest-energy” group that is closest in angle [in the implementation used here, we calculate the distance based on the original direction of that highest energy jet; many variants possible]
- apply the energy threshold to jets *after* this procedure

PRELIMINARY: is there a way of using the inclusive algs?

Energy recovery, when you know you are looking for *n* jets



```
/// preliminary illustration of how one might do energy recovery with anti-kt
vector<PseudoJet> antikt_with_Erecovery(const vector<PseudoJet> & hadrons) {
  double R = 0.4;
  double p = -1;
  JetDefinition antikt_jet_def(ee_genkt_algorithm, R, p);
  vector<PseudoJet> jets = antikt_jet_def(hadrons);

  // number of hard jets we will use to seed the energy-recovery procedure.
  // Typically the number of partons @LO in the signal process of interest
  unsigned nseed = 2;
  if (jets.size() <= nseed) return jets;
  vector<PseudoJet> jets_with_recovery(jets.begin(), jets.begin() + nseed);
  // for the remaining jets, join them with the "seed"-jet that
  // is closest in angle (using the direction of the original seed jet to
  // avoid collinear unsafety if procedure is applied to constituents rather
  // than jets)
  for (unsigned int iadd = nseed; iadd<jets.size(); ++iadd){
    const auto &j = jets[iadd];
    vector<double> distances(nseed);
    for (unsigned iseed = 0; iseed < nseed; iseed++) distances[iseed] = cos_theta(j,jets[iseed]);
    unsigned imin = std::min_element(distances.begin(), distances.end()) - distances.begin();
    jets_with_recovery[imin] = join(jets_with_recovery[imin], j);
  }
  // finally apply the energy threshold to the jets being returned
  double Emin = 10.0;
  return SelectorEMin(Emin)(jets_with_recovery);
}
```

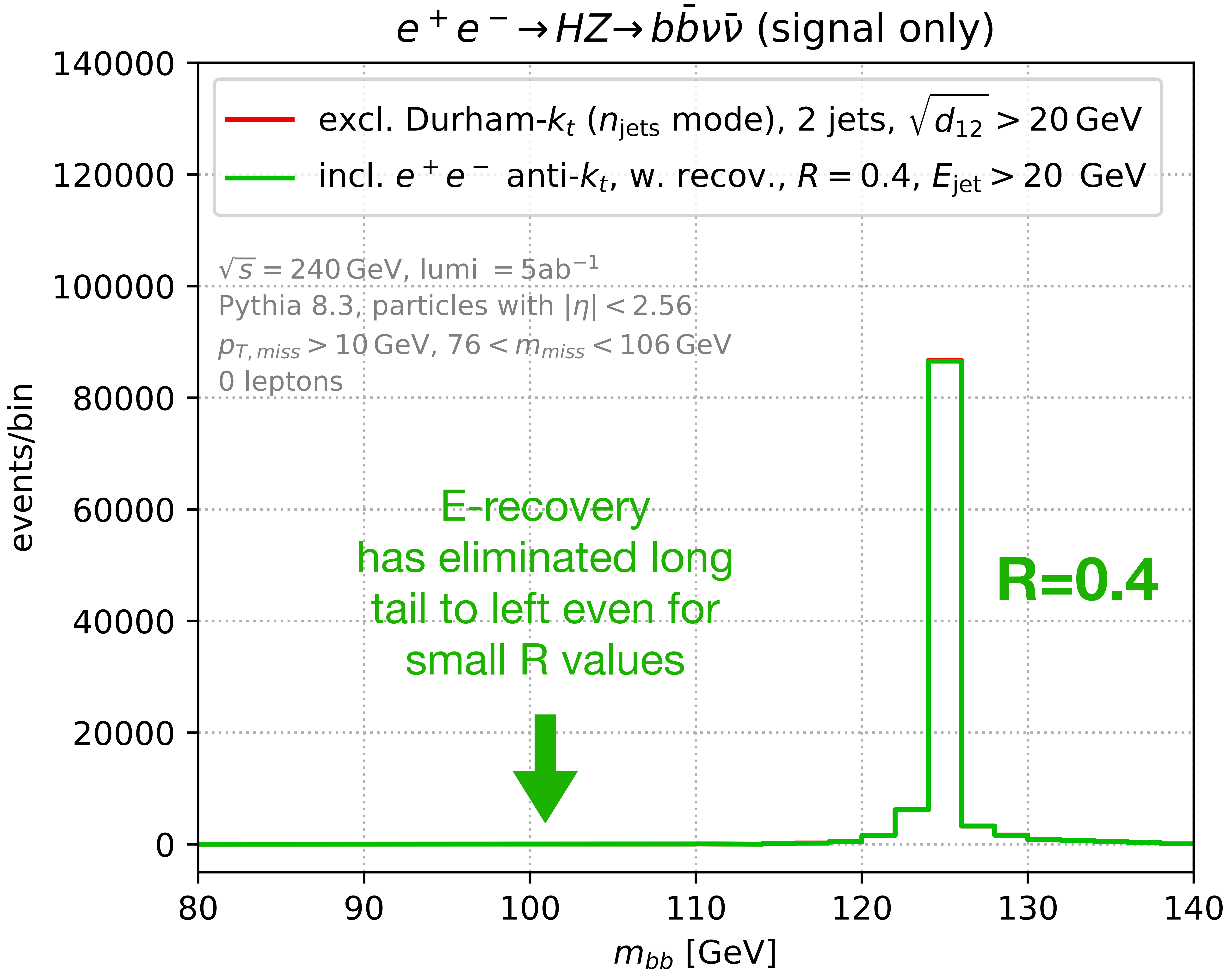
r-energy)

nergy” group

ulate the

t; many

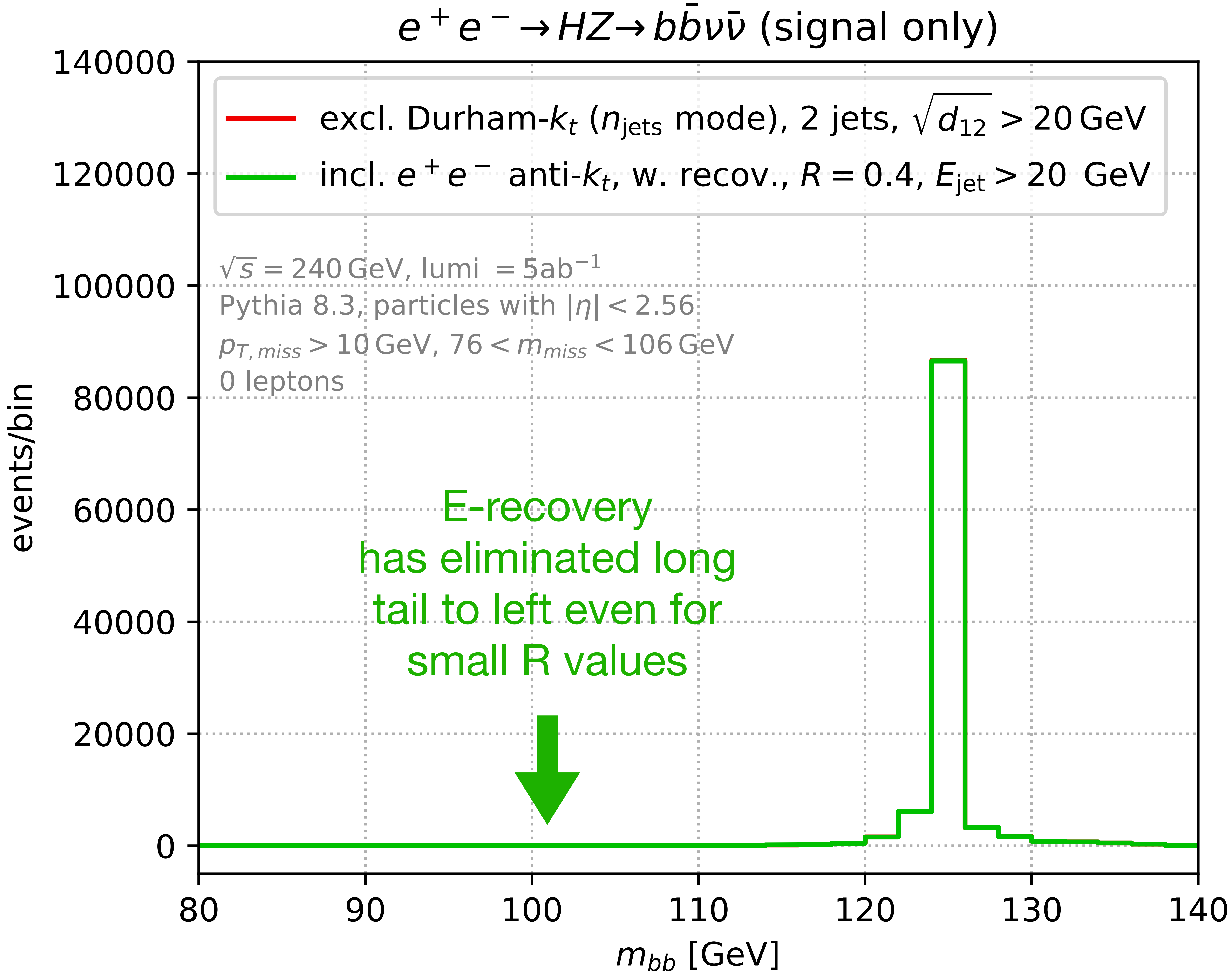
Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



e^+e^- inclusive anti-k_t
w. E-recovery

In this simple case, results indistinguishable from exclusive Durham-k_t ($n_{\rm jets}$ mode), even with a small radius

Illustrate with $e^+e^- \to HZ \to bb\nu\nu$



e^+e^- inclusive anti-k_t
w. E-recovery

In this simple case, results indistinguishable from exclusive Durham-k_t (n_{jets} mode), even with a small radius

multijet example 1: $e^+e^- \rightarrow HZ \rightarrow bbjj$

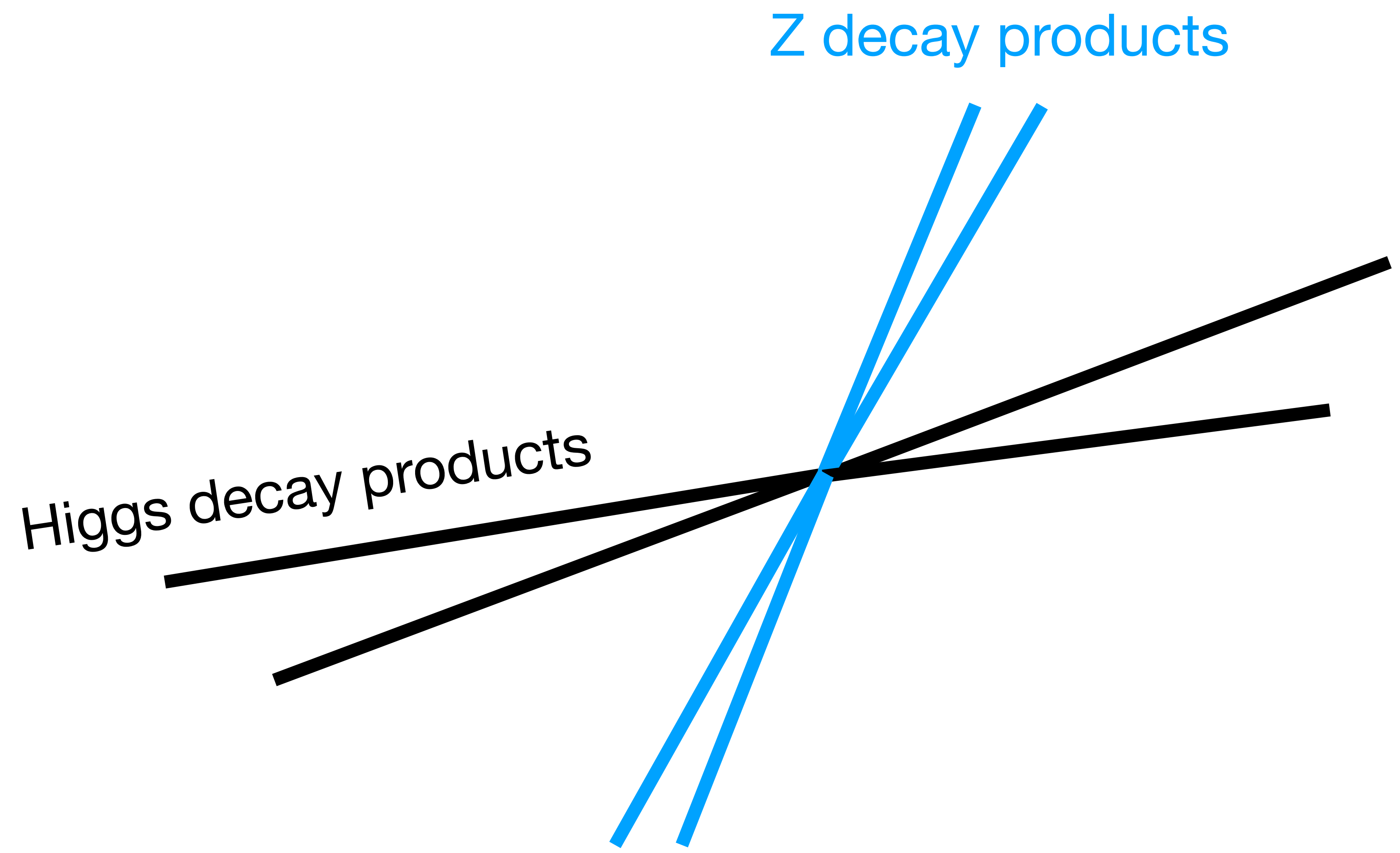
Examples chosen so that we don’t need to worry about combinatorics for associating jets with resonances

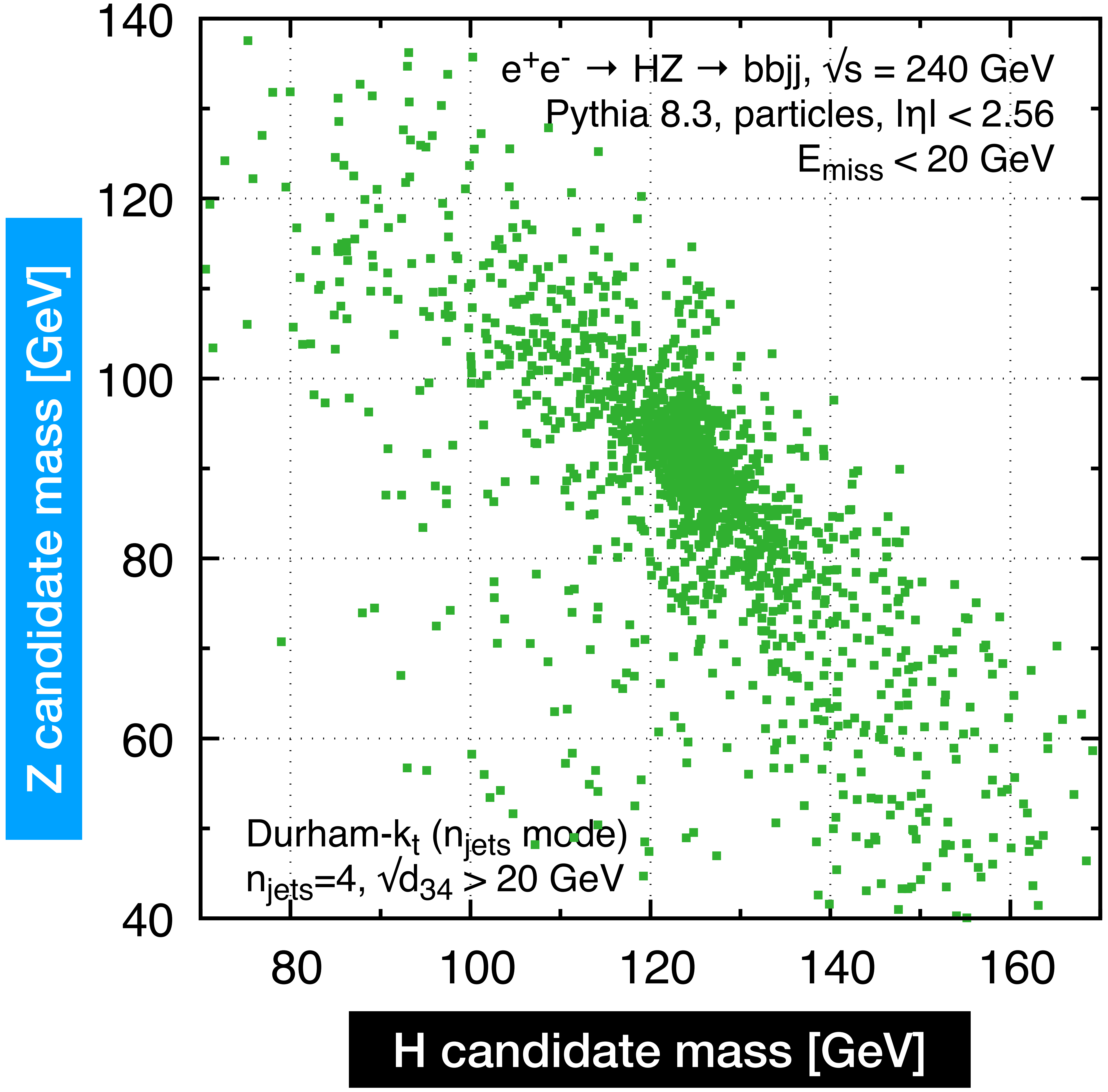
e^+e^- → HZ → bbjj analysis

- Require no leptons, $E_{miss} < 20$ GeV
- Run jet finding, e.g. Durham-k_t (n_{jets} mode), n_{jets}=4, $\sqrt{d_{34}} > 20\mathrm{GeV}$
- Require exactly two of the jets to be b-tagged ("bb") → Higgs candidate
- The two non b-tagged jets ("jj") → Z candidate
  (this leaves out Z→ bb decays, which involve more complex combinatorics)
- Cut & count, where "signal region" is
  - $70 < m_{jj} < 110$ GeV
  - $110 < m_{bb} < 140$ GeV

e^+e^- → HZ → bbjj analysis

Jets question #1: how accurately does the jet alg. assign particles to H v. Z candidate?



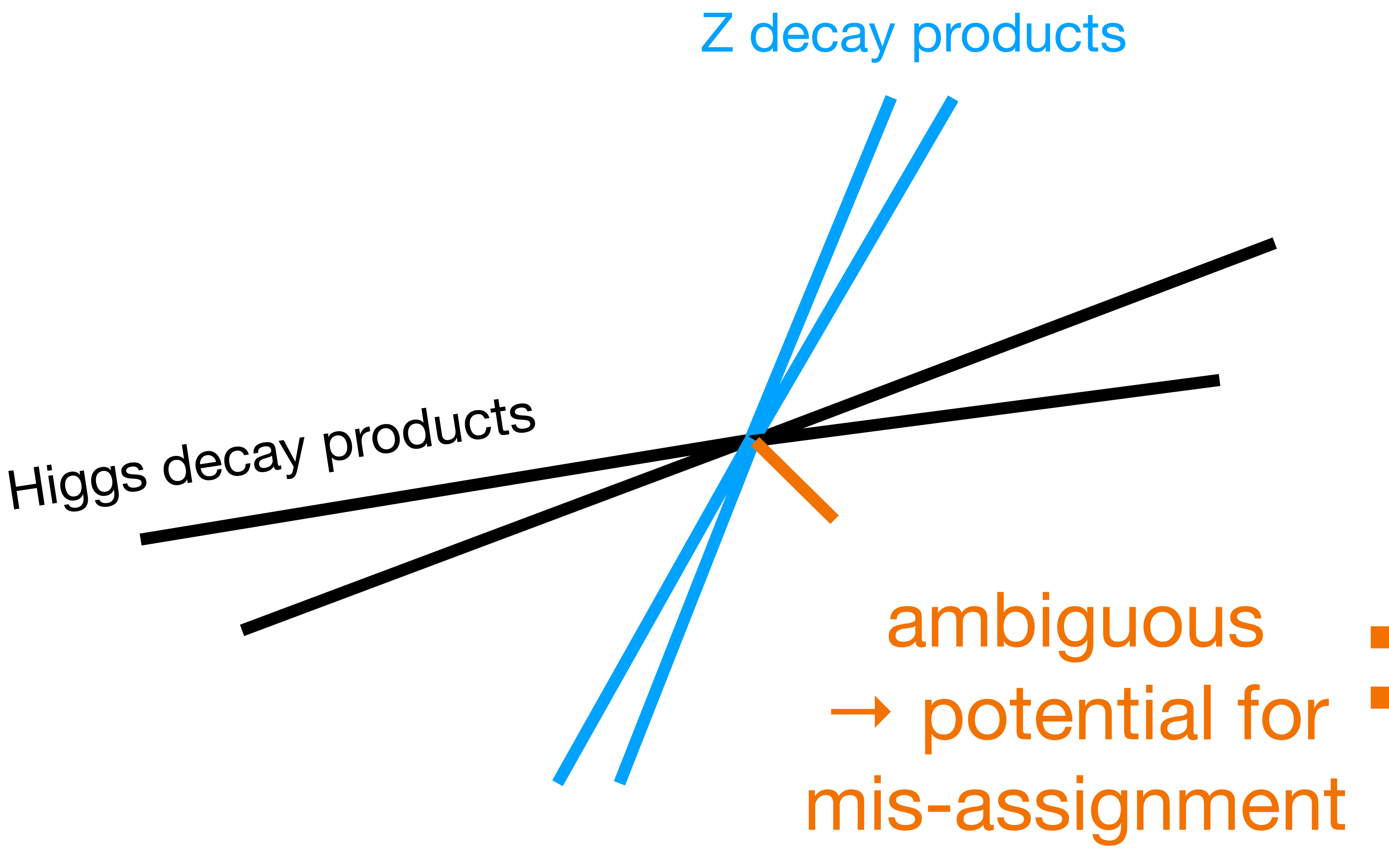


Physically, H ($\Gamma \simeq 4$MeV) & Z ($\Gamma \simeq 2.5$GeV) decay on very different timescales. There is an (almost) unambiguously “right” answer for the assignment. But jet algorithms see only particle kinematics and can make “mistakes”.

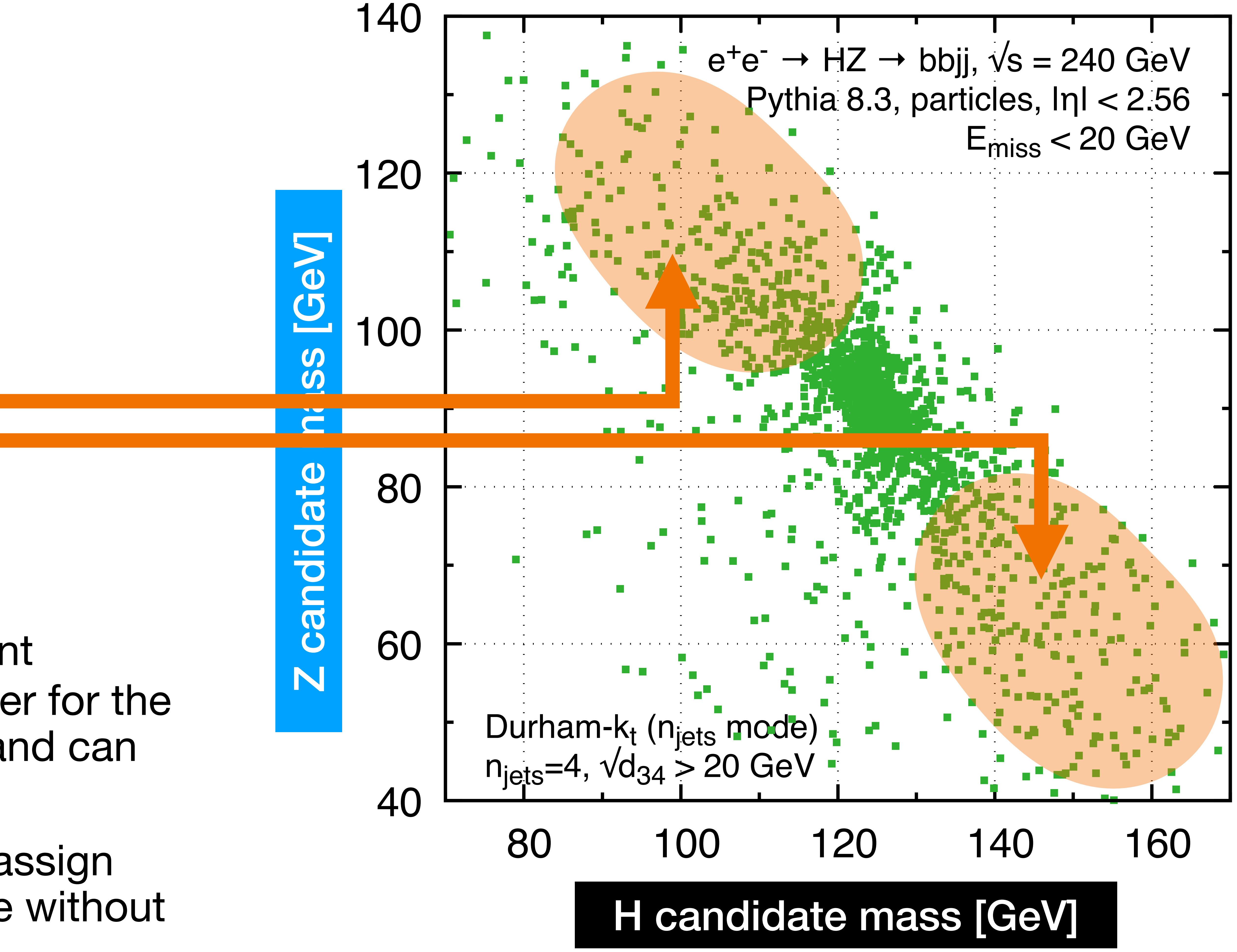
All the algorithms we’ve looked at make similar mistakes, ~ assign particle to jet that is closest in angle. Difficult to do otherwise without scultping backgrounds.

$e^+e^- \to HZ \to bbjj$ analysis

Jets question #1: how accurately does the jet alg. assign particles to H v. Z candidate?



ambiguous
→ potential for
mis-assignment

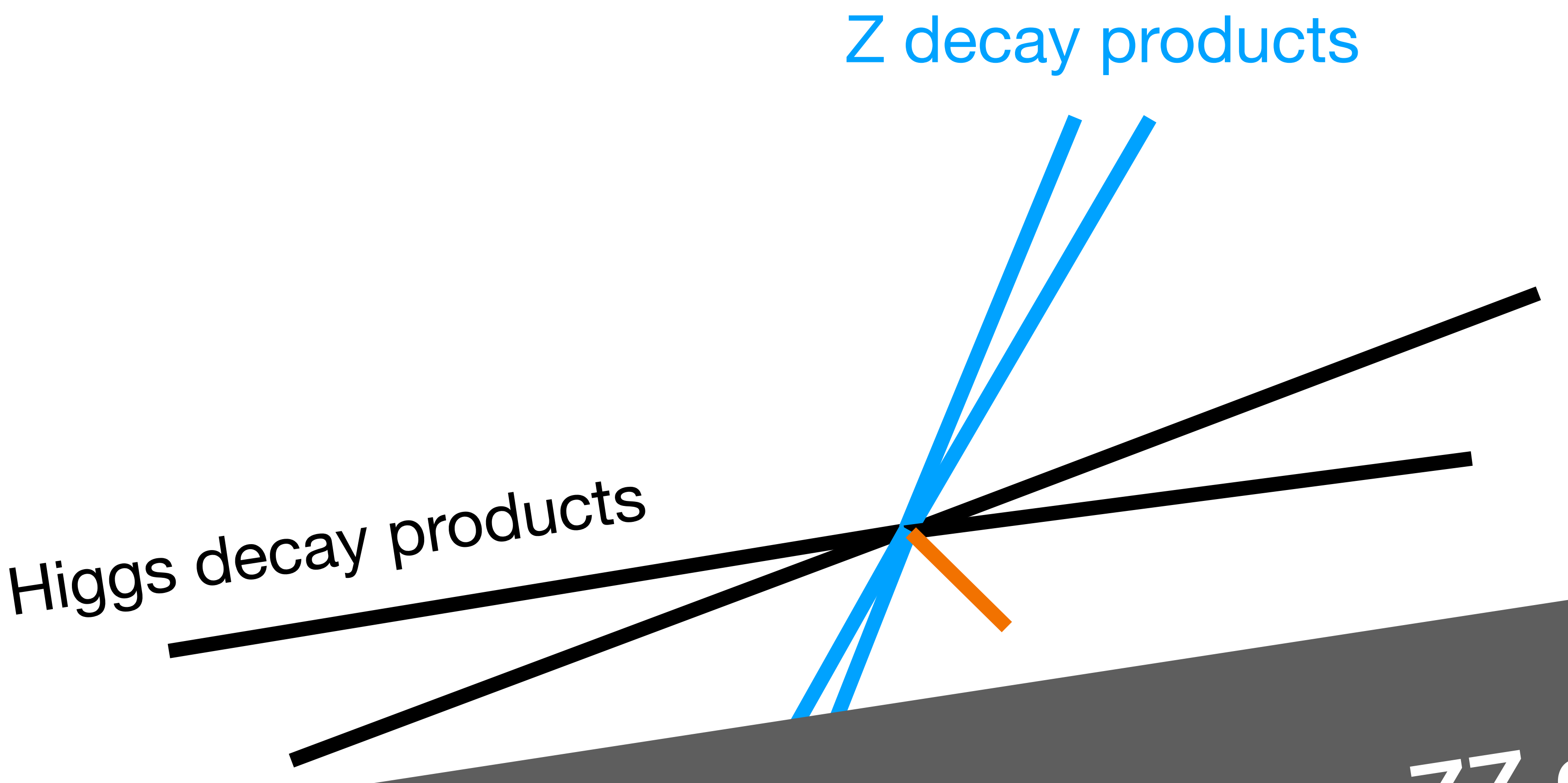


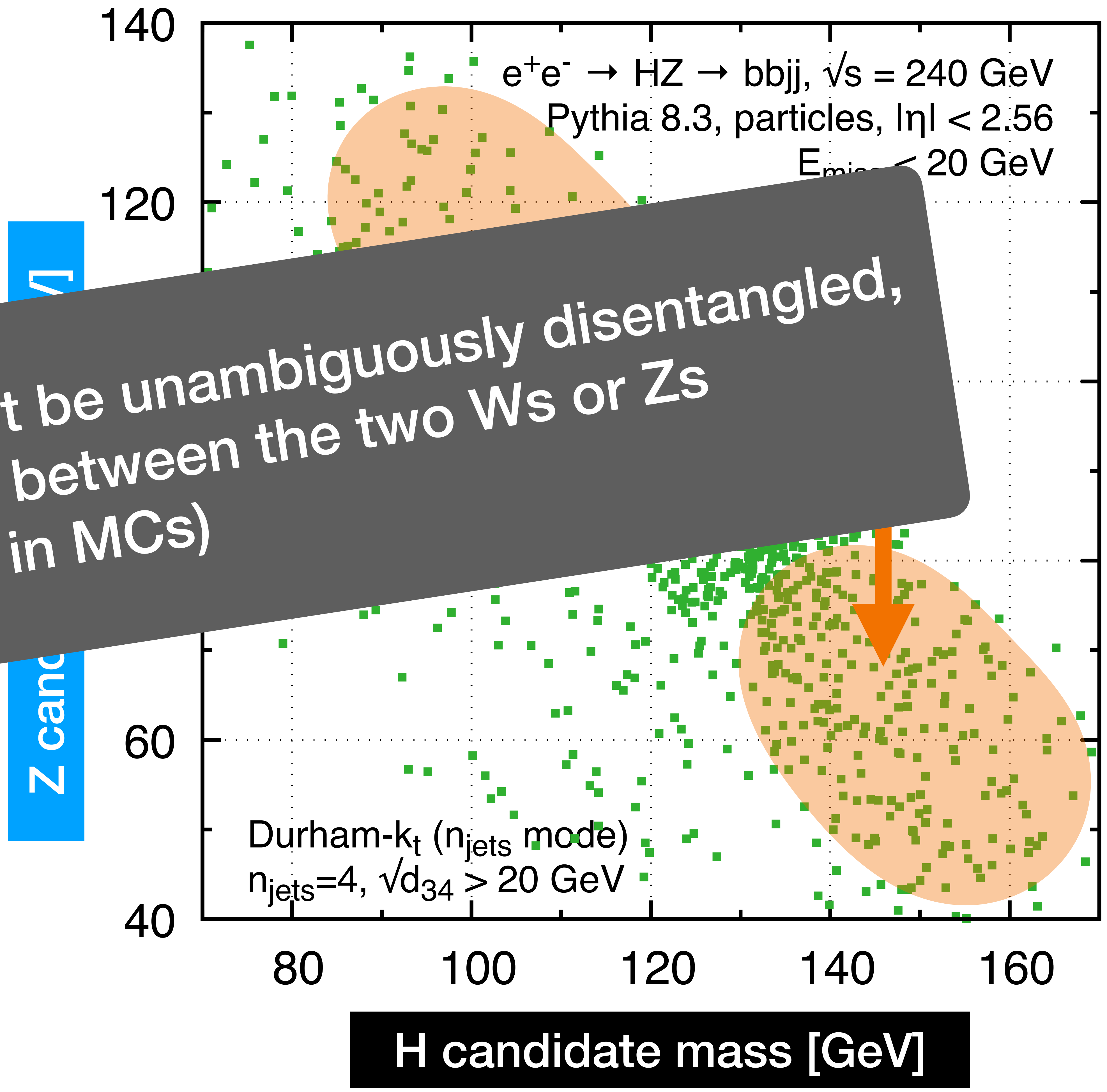
Physically, H ($\Gamma \simeq 4$MeV) & Z ($\Gamma \simeq 2.5$GeV) decay on very different timescales. There is an (almost) unambiguously "right" answer for the assignment. But jet algorithms see only particle kinematics and can make "mistakes".

All the algorithms we've looked at make similar mistakes, ~ assign particle to jet that is closest in angle. Difficult to do otherwise without scultping backgrounds.

$e^+e^- \to HZ \to bbjj$ analysis

Jets question #1: how accurately does the jet alg. assign particles to H v. Z candidate?





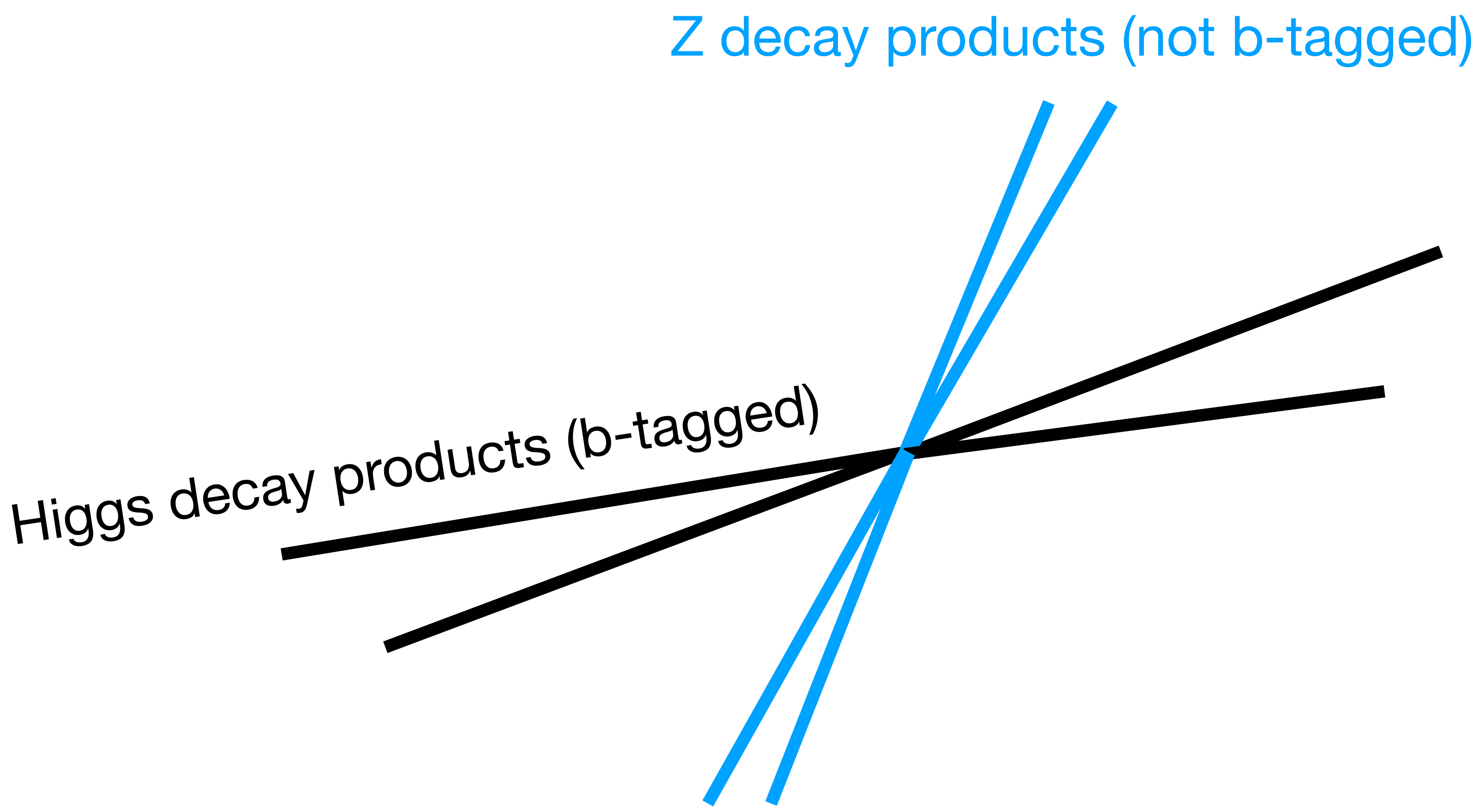
Watch out: particles from WW or ZZ decays cannot be unambiguously disentangled, because hadronisation timescale overlaps between the two Ws or Zs (“colour reconnection” in MCs)

Physically, H on very different timescales. T unambiguously “right” answer for the assignment. But jet algorithms see only particle kinematics and can make “mistakes”.

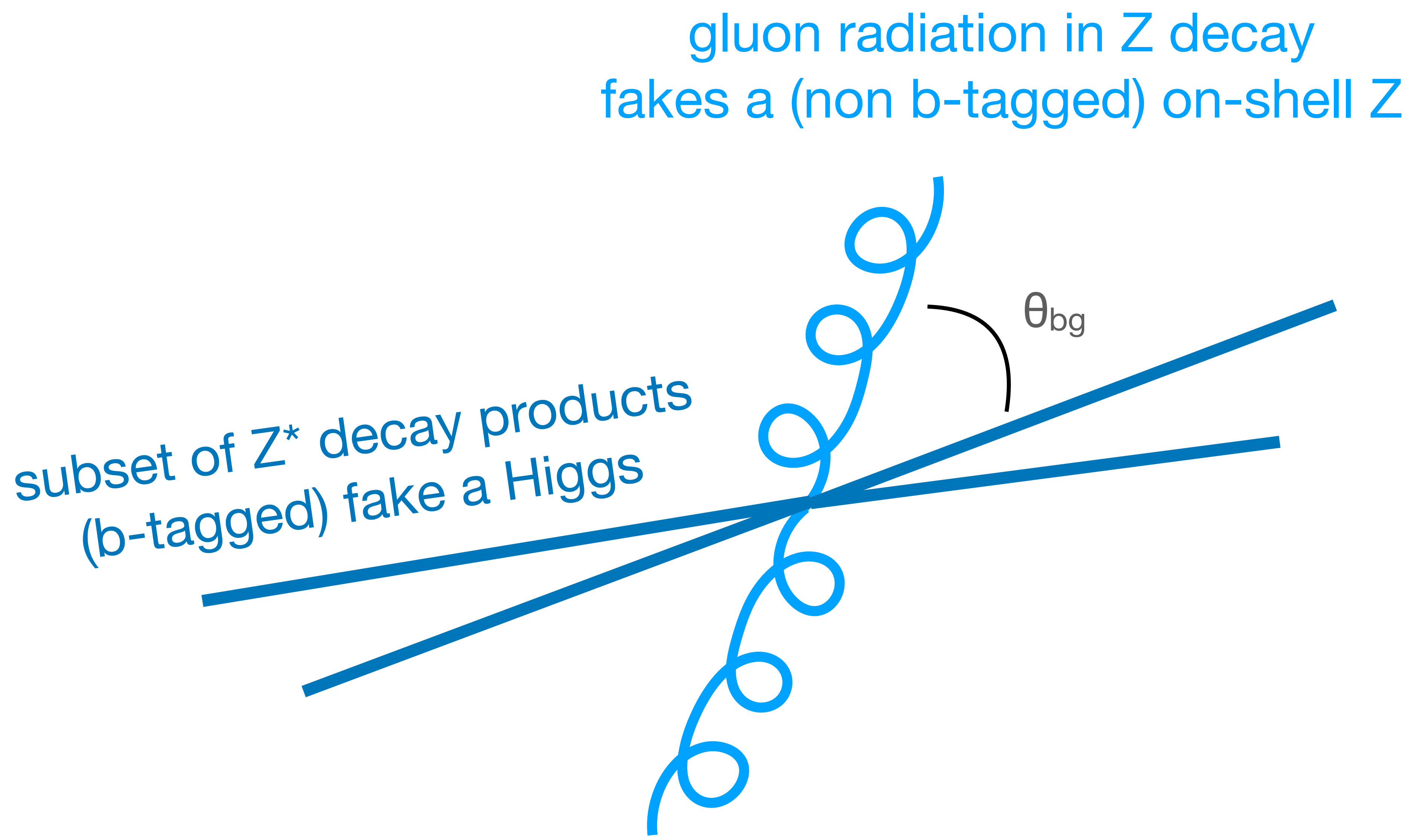
All the algorithms we’ve looked at make similar mistakes, ~ assign particle to jet that is closest in angle. Difficult to do otherwise without scultping backgrounds.

$e^+e^- \to HZ \to bbjj$ analysis

Jets question #2: remove backgrounds, while preserving signal



SIGNAL



BACKGROUND ($Z^* \to$ hadrons)

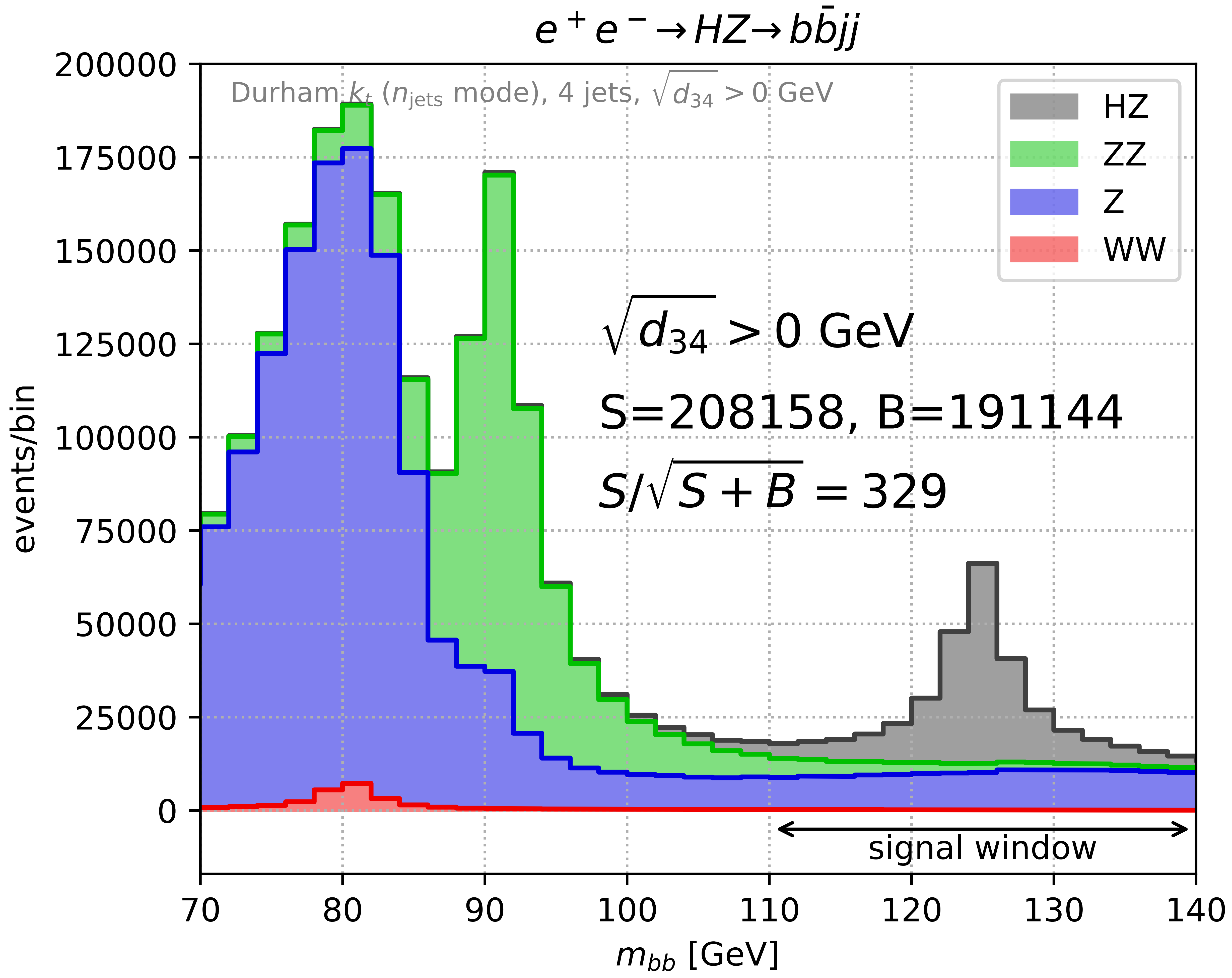
In most cases, backgrounds come from processes with same number of LO jets (e.g. ZZ), or fewer LO jets (e.g. Z*) + gluon radiation.

Gluon radiation tends to be collinear ($\theta_{bg} \ll 1$), while angle between Higgs and Z decay jets is uniformly distributed. Background can be reduced either with d_{34} condition (Durham-k_t) or jet radius parameter R of inclusive generalised k_t algorithms. **Both eliminate small-angle gluon jets.**

e^+e^- → HZ → bbjj analysis

illustration of impact of d_{34} cut

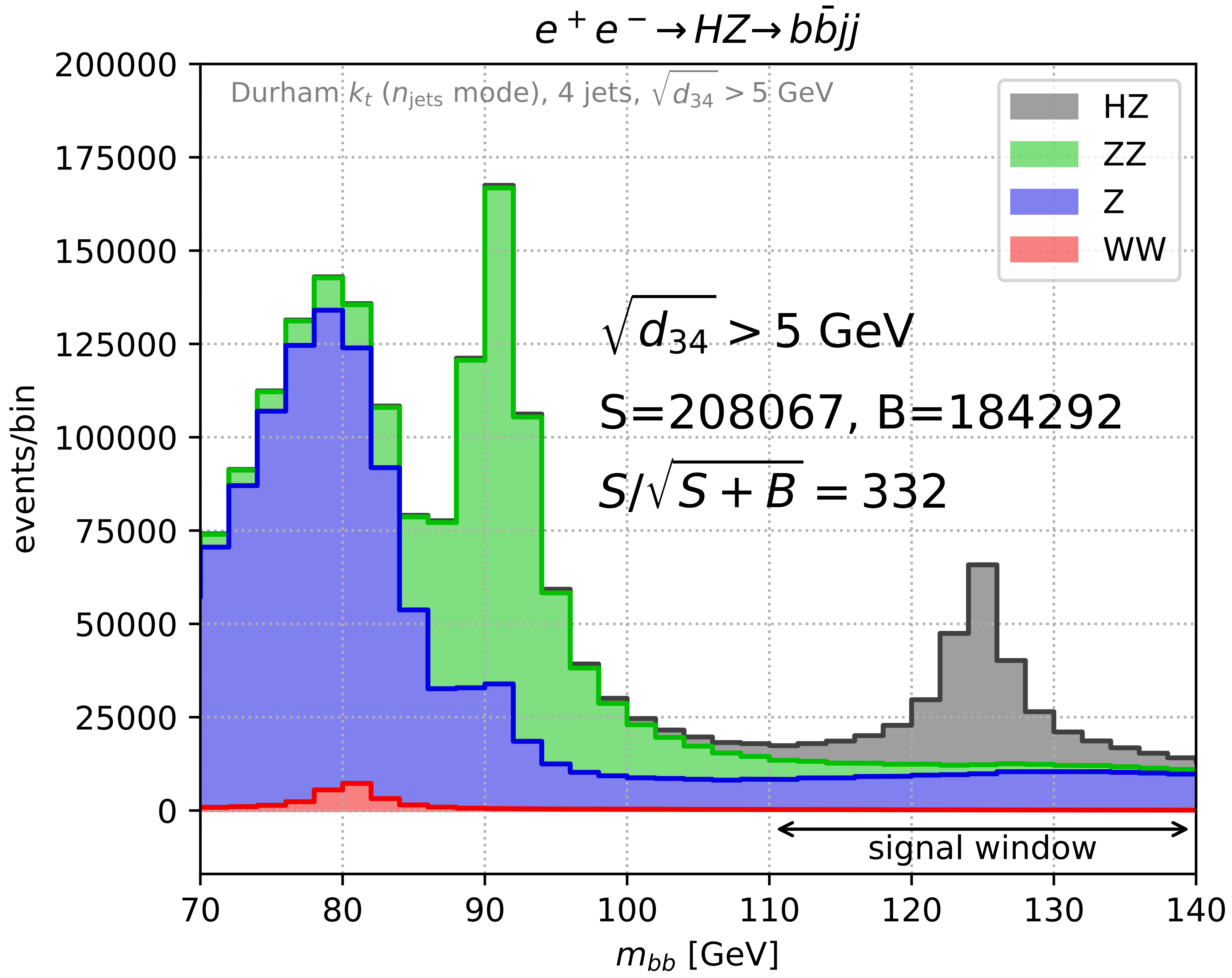
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



$e^+e^- \to$ HZ $\to$ bbjj analysis

illustration of impact of d_{34} cut

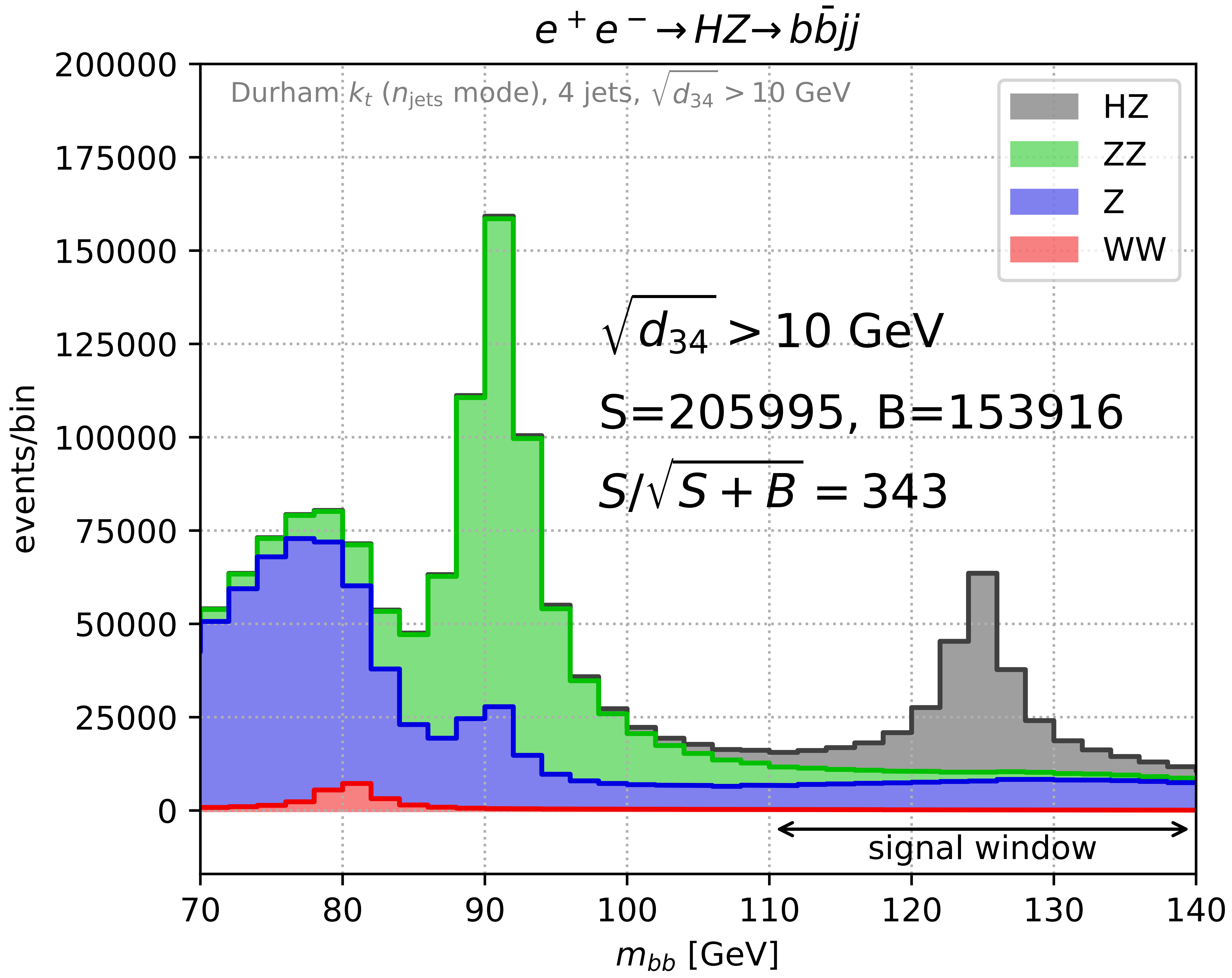
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



e^+e^- → HZ → bbjj analysis

illustration of impact of d_{34} cut

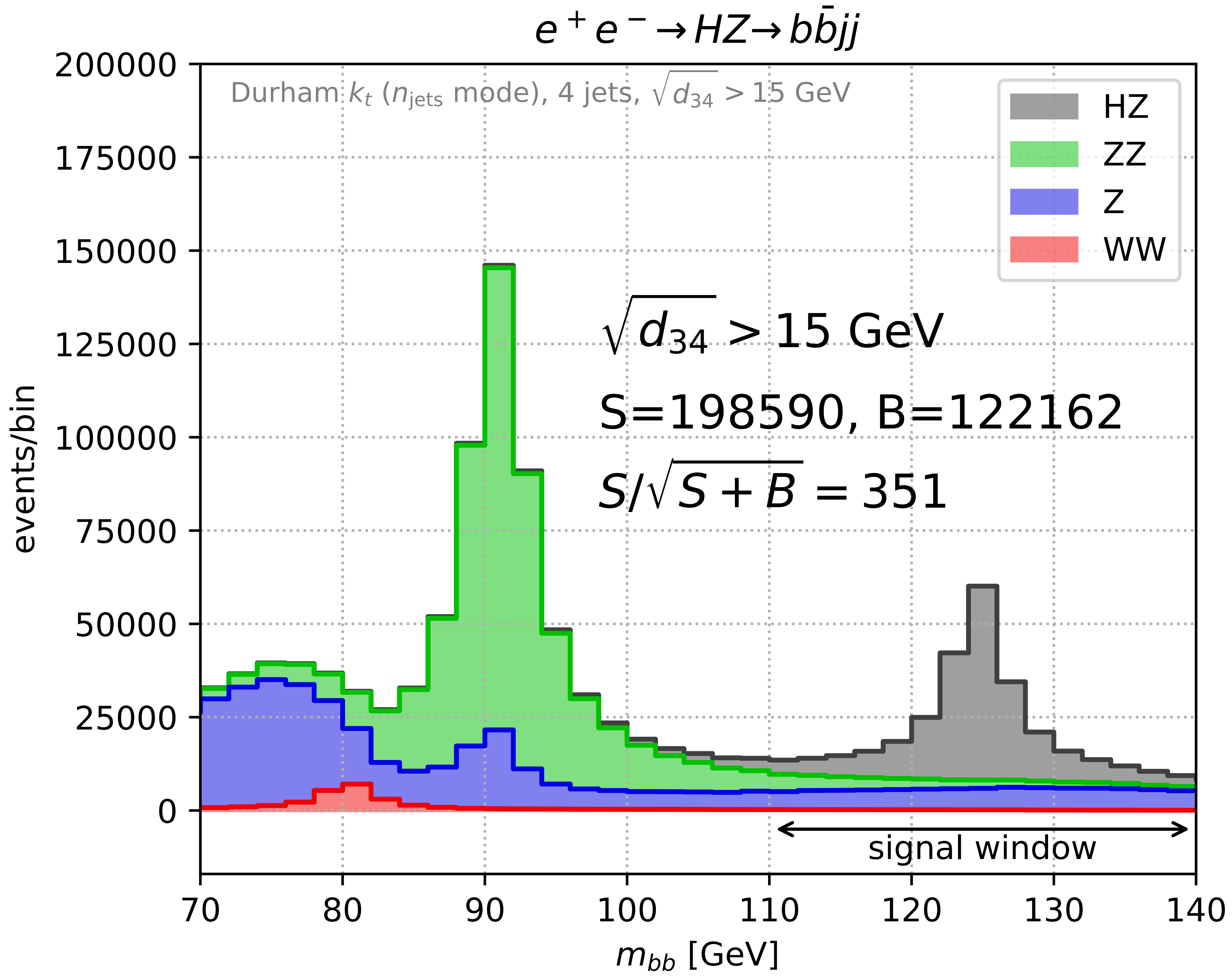
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



e^+e^- → HZ → bbjj analysis

illustration of impact of d_{34} cut

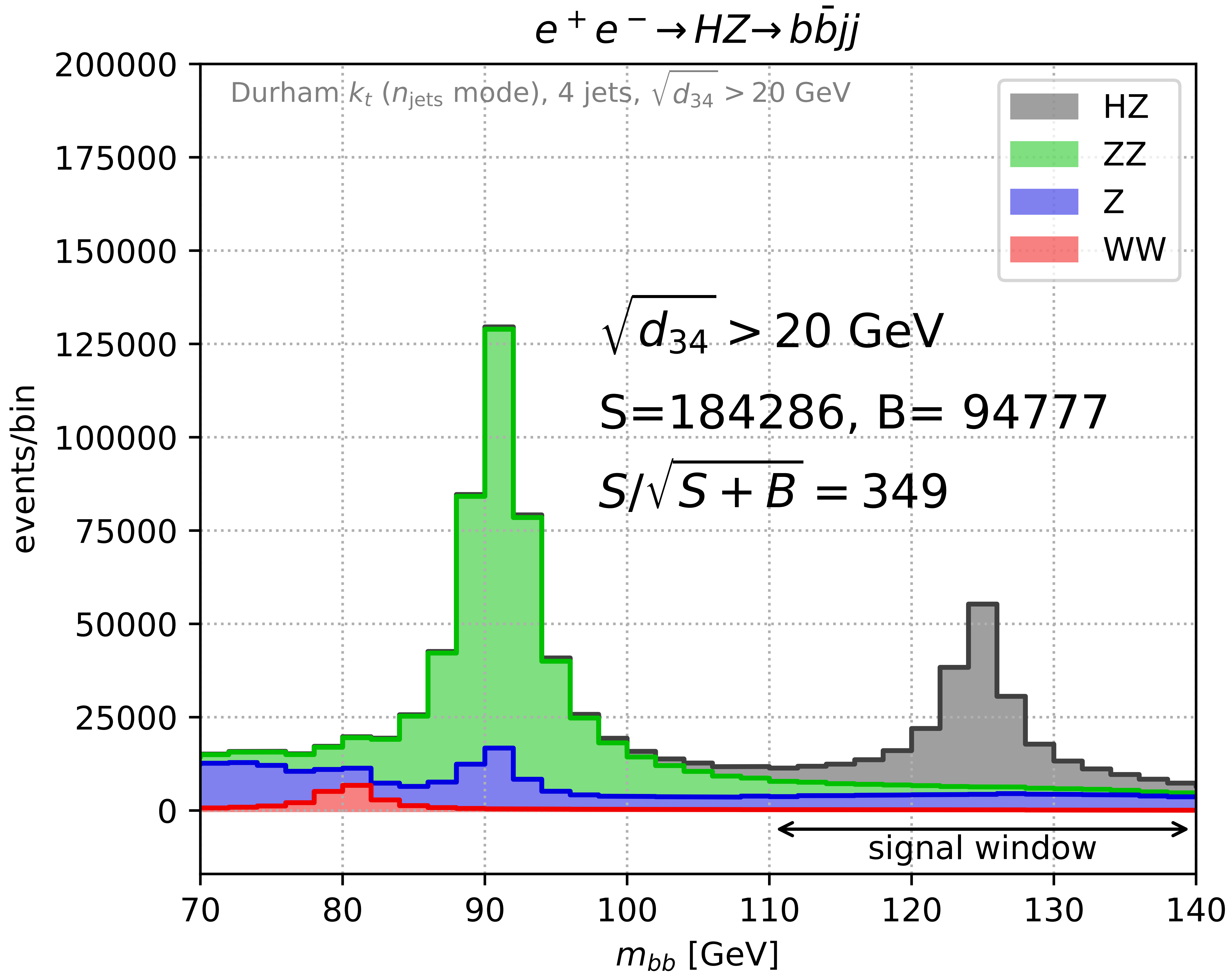
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



$e^+e^- \rightarrow$ HZ $\rightarrow$ bbjj analysis

illustration of impact of d_{34} cut

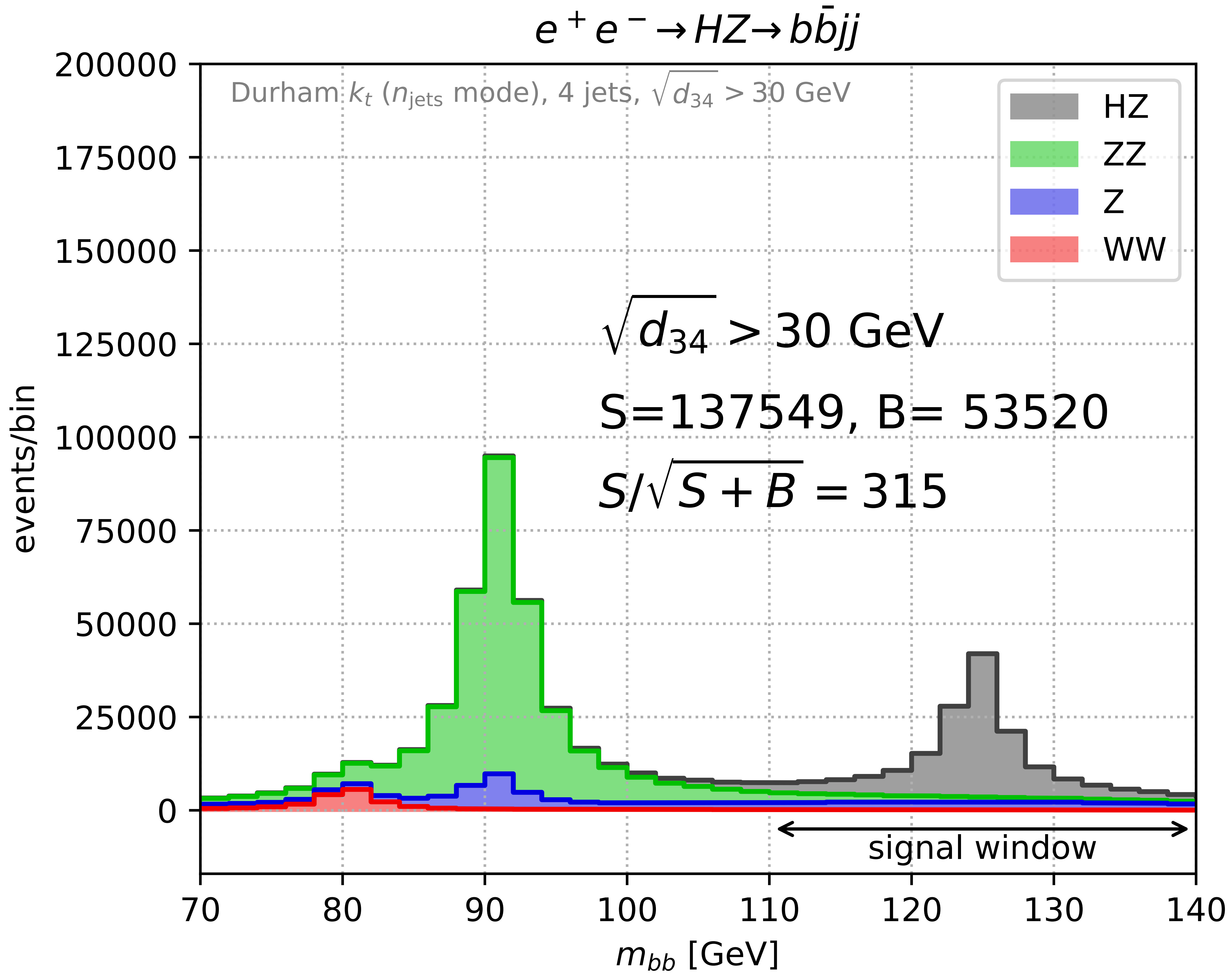
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



$e^+e^- \to$ HZ $\to$ bbjj analysis

illustration of impact of d_{34} cut

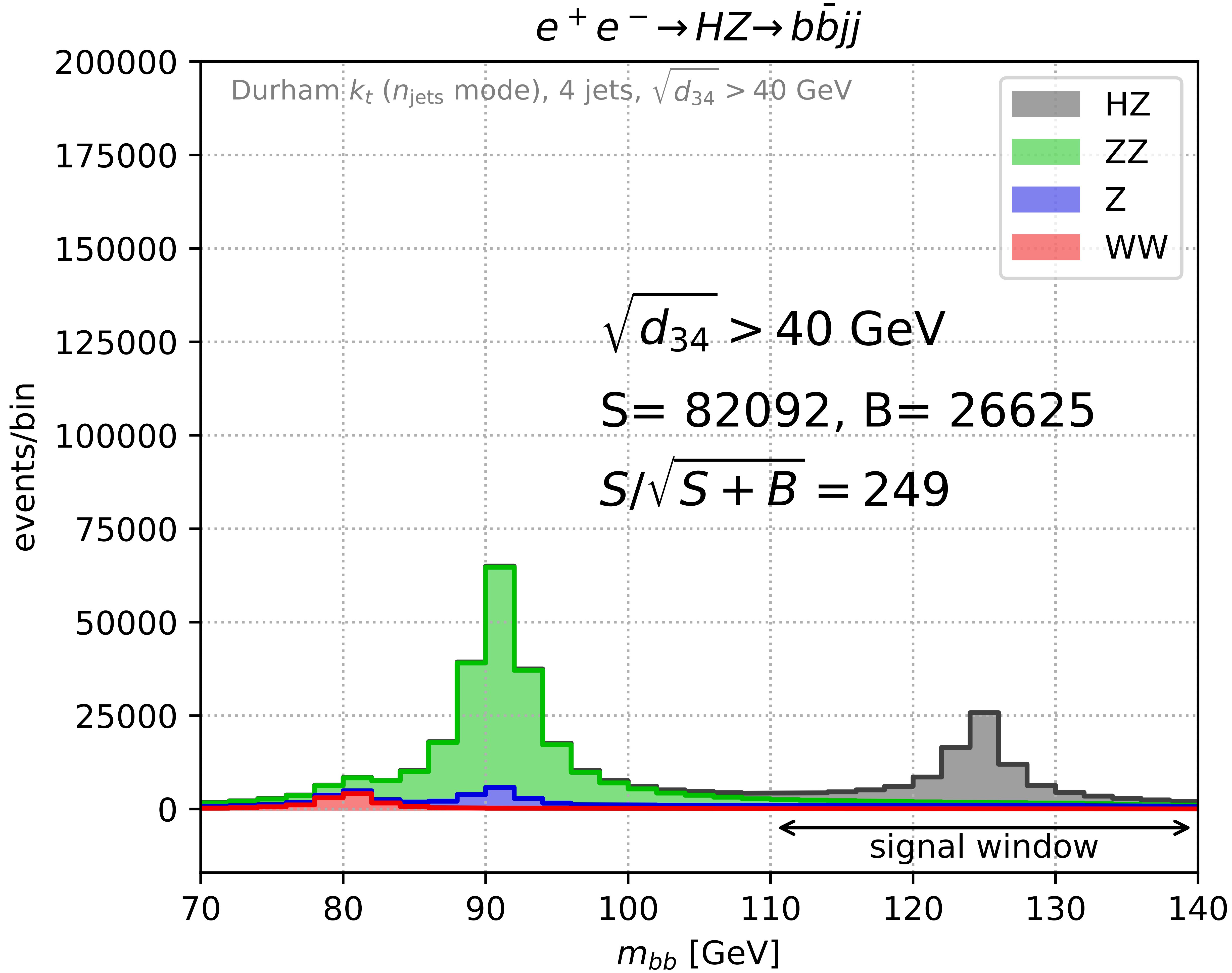
- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



e^+e^- → HZ → bbjj analysis

illustration of impact of d_{34} cut

- too small a limit on d_{34} leads to enhanced background
- too large a limit cuts out large fraction of signal
- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$
- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



$e^+e^- \to$ HZ $\to$ bbjj analysis

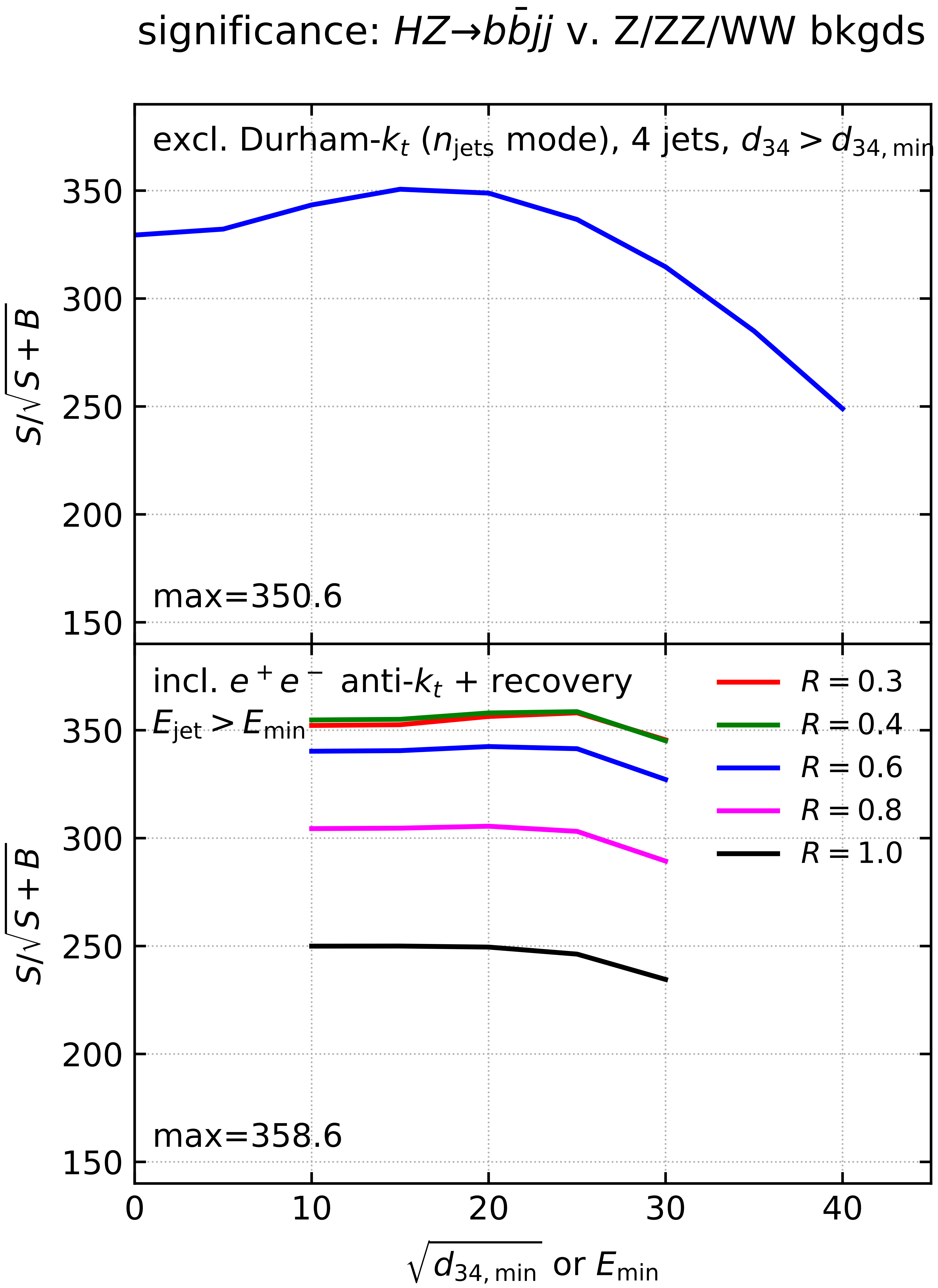
illustration of impact of d_{34} cut

- too small a limit on d_{34} leads to enhanced background

- too large a limit cuts out large fraction of signal

- One can scan over d_{34} cut to optimise $S/\sqrt{S+B}$

- A modern analysis might use the event-by event d_{34} value as a ML input (or full jet momenta)



multijet example 2: $e^+e^- \to HZ \to WW(\to 4j)bb$

Example again chosen so that we don't need to worry about combinatorics for associating jets with resonances

$e^+e^- \rightarrow HZ \rightarrow WW(\rightarrow 4j)bb$ analysis

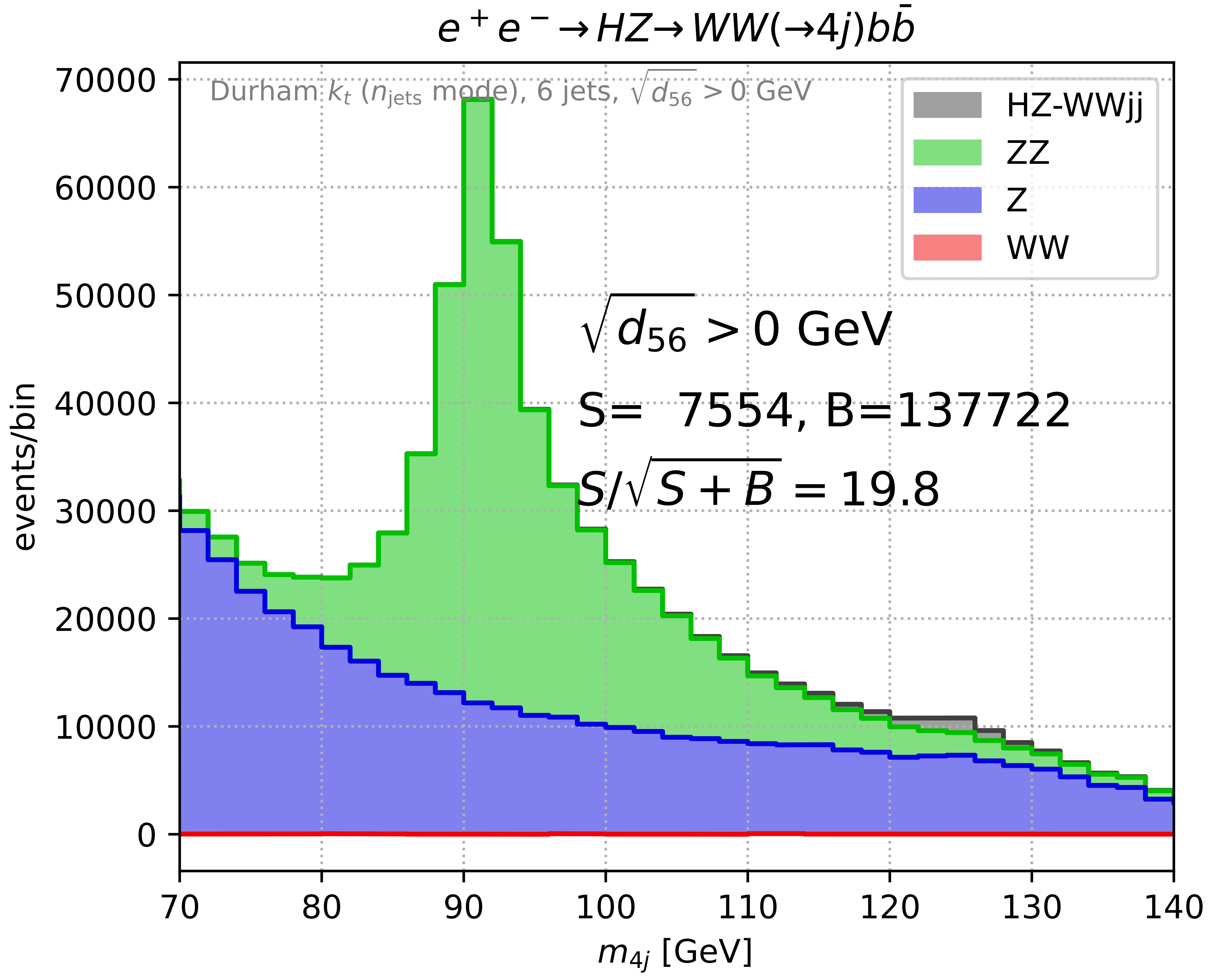
- Require no leptons, $E_{miss} < 20$ GeV
- Run jet finding, e.g. Durham-k_t (n_{jets} mode), $n_{jets}=6$, $\sqrt{d_{34}} > 20\mathrm{GeV}$
- Require exactly two of the jets to be b-tagged ("bb") → Z candidate
  (this leaves out Z→ jj decays, which give more complex combinatorics)
- The four non b-tagged jets ("jjjj") → H candidate
- Cut & count, where "signal region" is
  - $85 < m_{bb} < 110$ GeV (NB: narrow window reduces ZZ background)
  - $110 < m_{4j} < 140$ GeV
- NB: signal in plots is fully hadronic HZ→WWbb; plots **do not include** backgrounds from other Higgs decay channels.

e^+e^- → HZ→WW(→4j)bb analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case
- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY



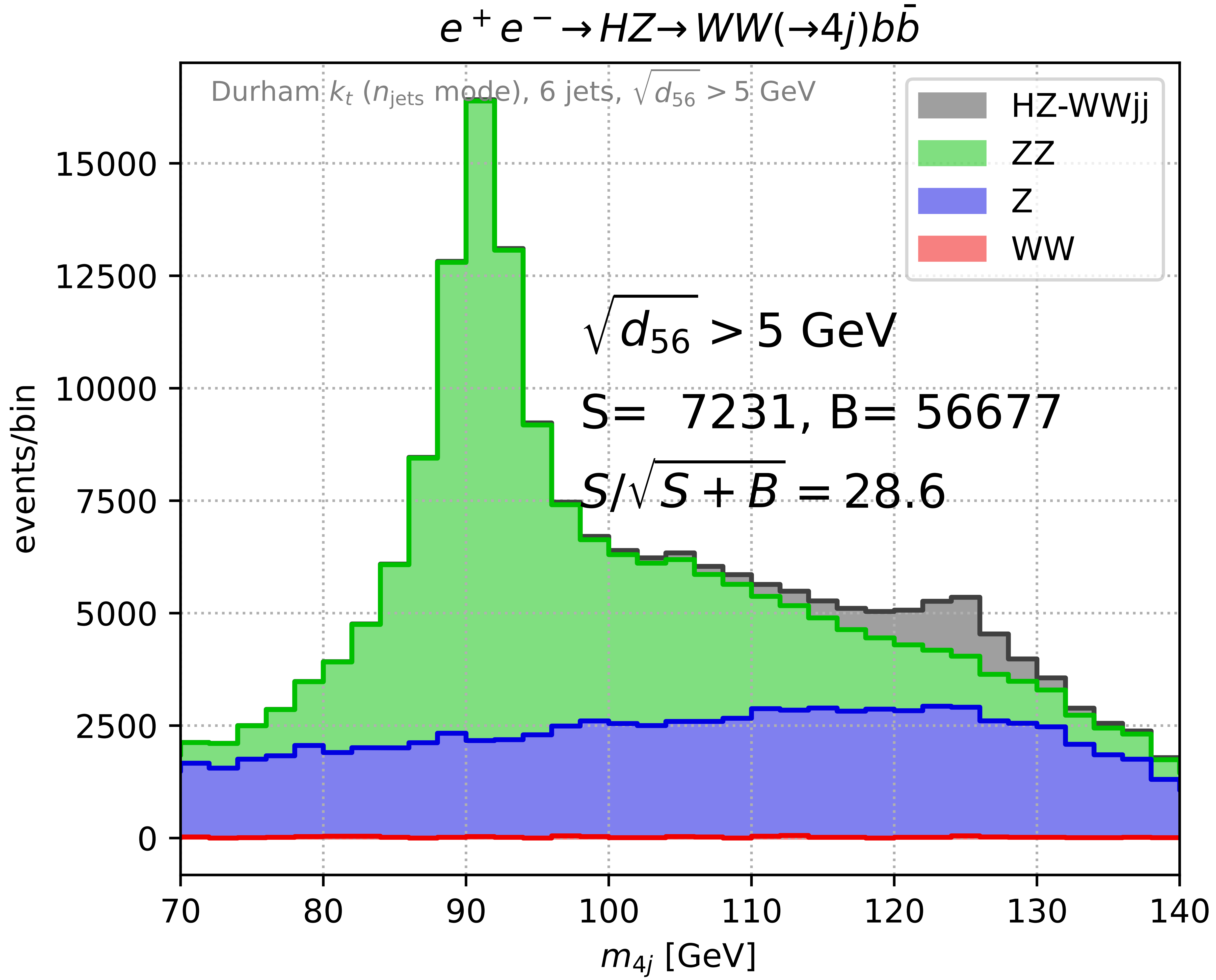
$e^+e^- \rightarrow HZ \rightarrow WW(\rightarrow 4j)bb$ analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case

- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY

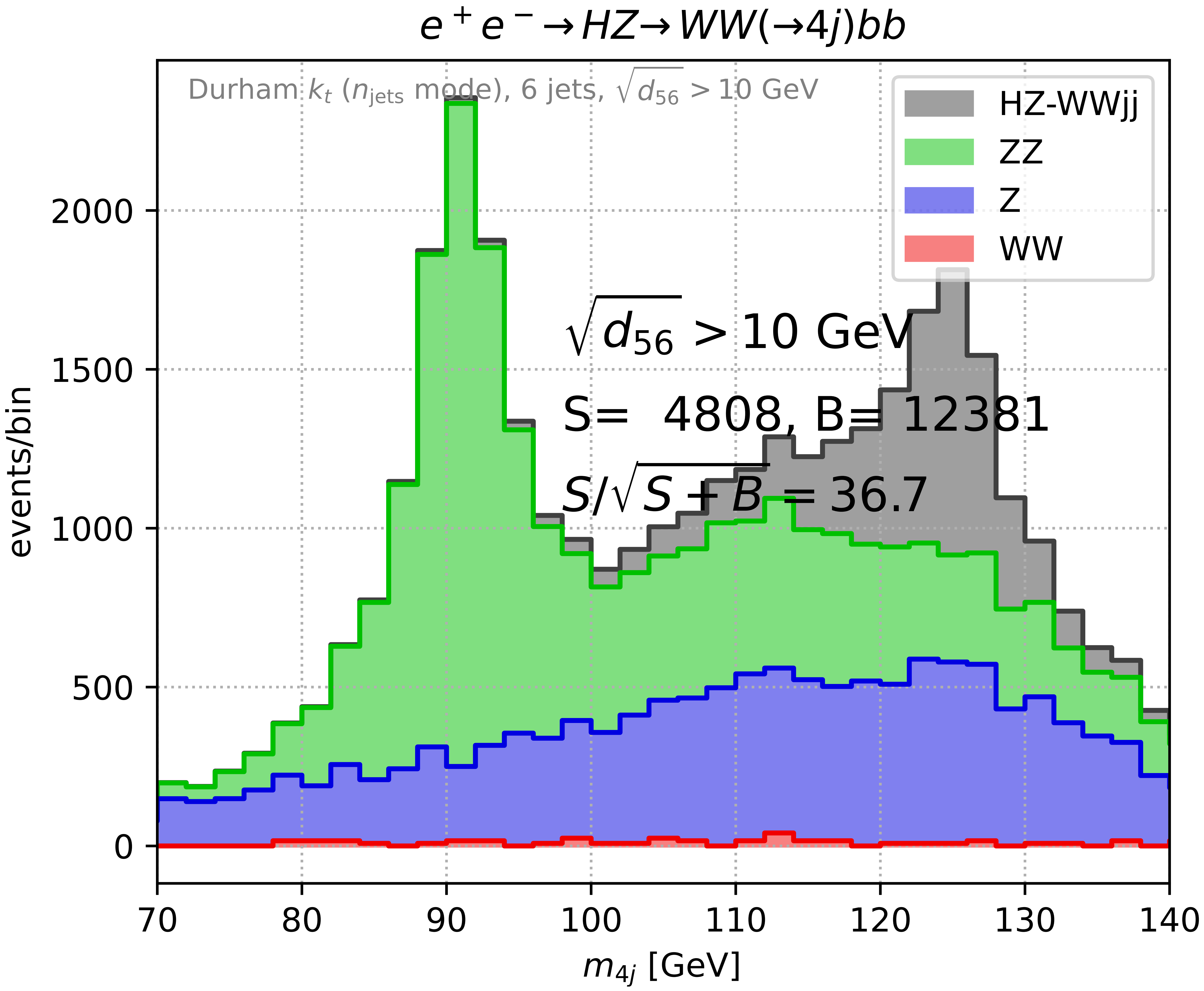


$e^+e^- \rightarrow HZ \rightarrow WW(\rightarrow 4j)bb$ analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case
- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY

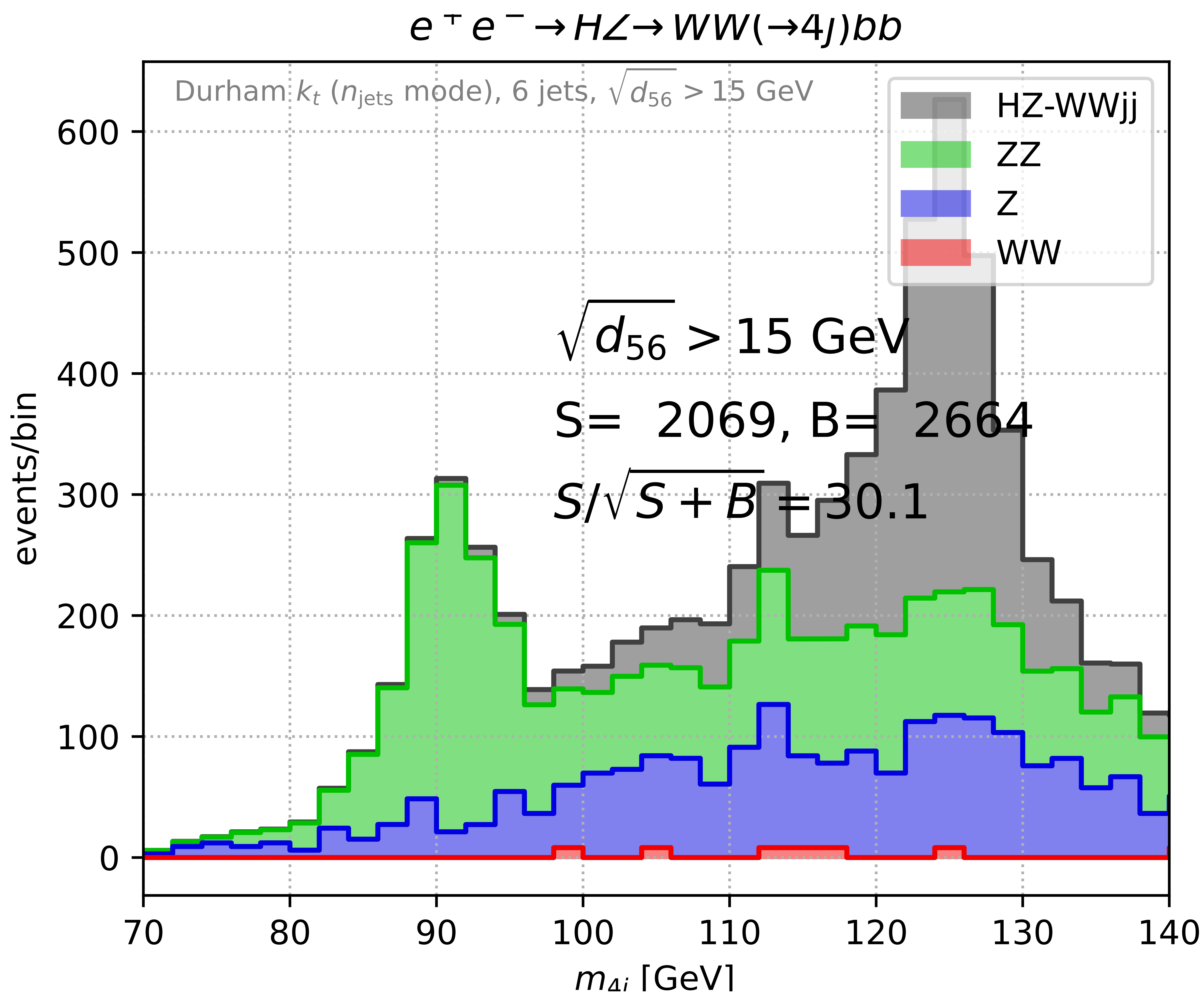


e^+e^- → HZ→WW(→4j)bb analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case
- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY



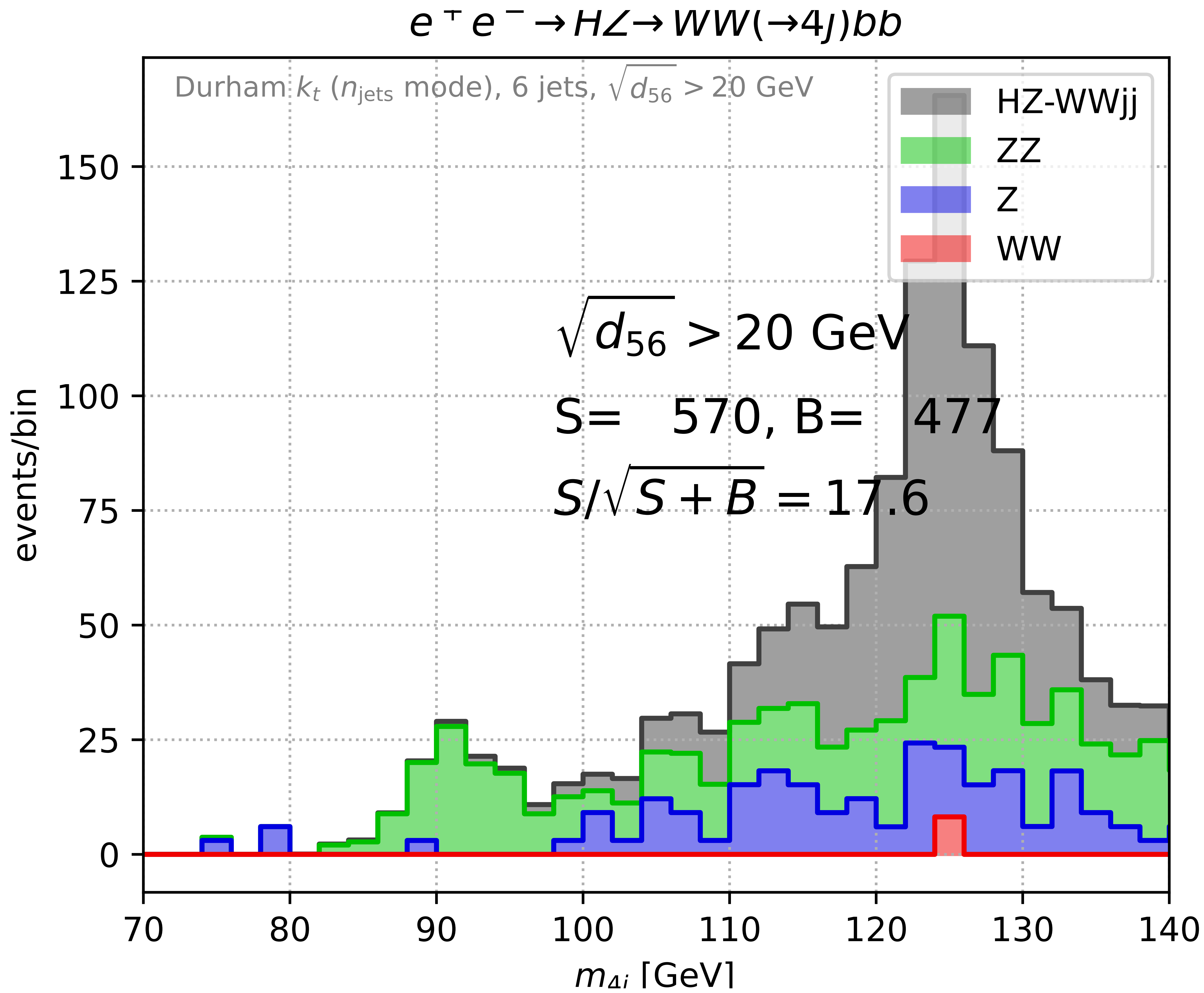
e^+e^- → HZ→WW(→4j)bb analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case

- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY



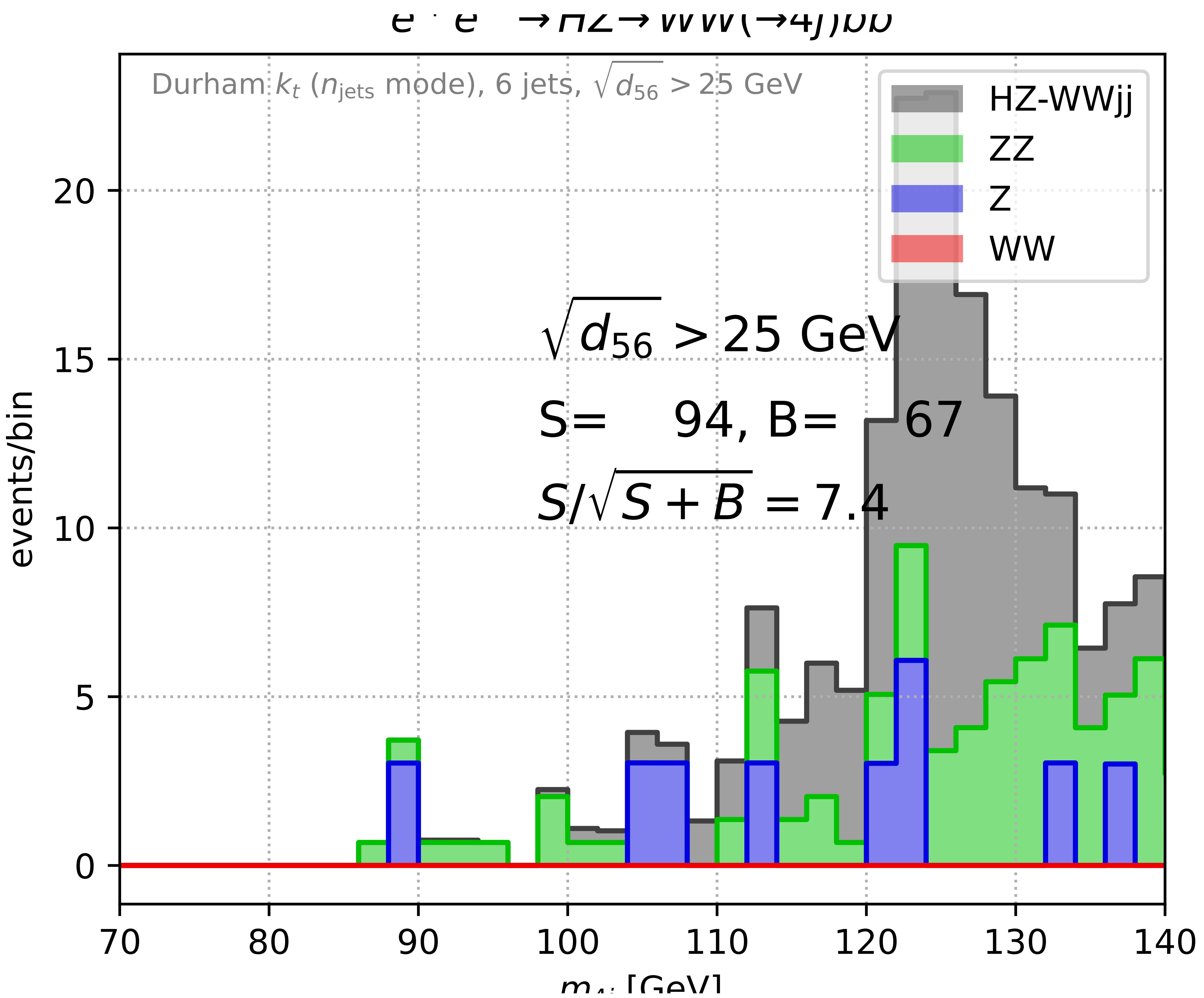
$e^+e^- \rightarrow HZ \rightarrow WW(\rightarrow 4j)bb$ analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case

- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY



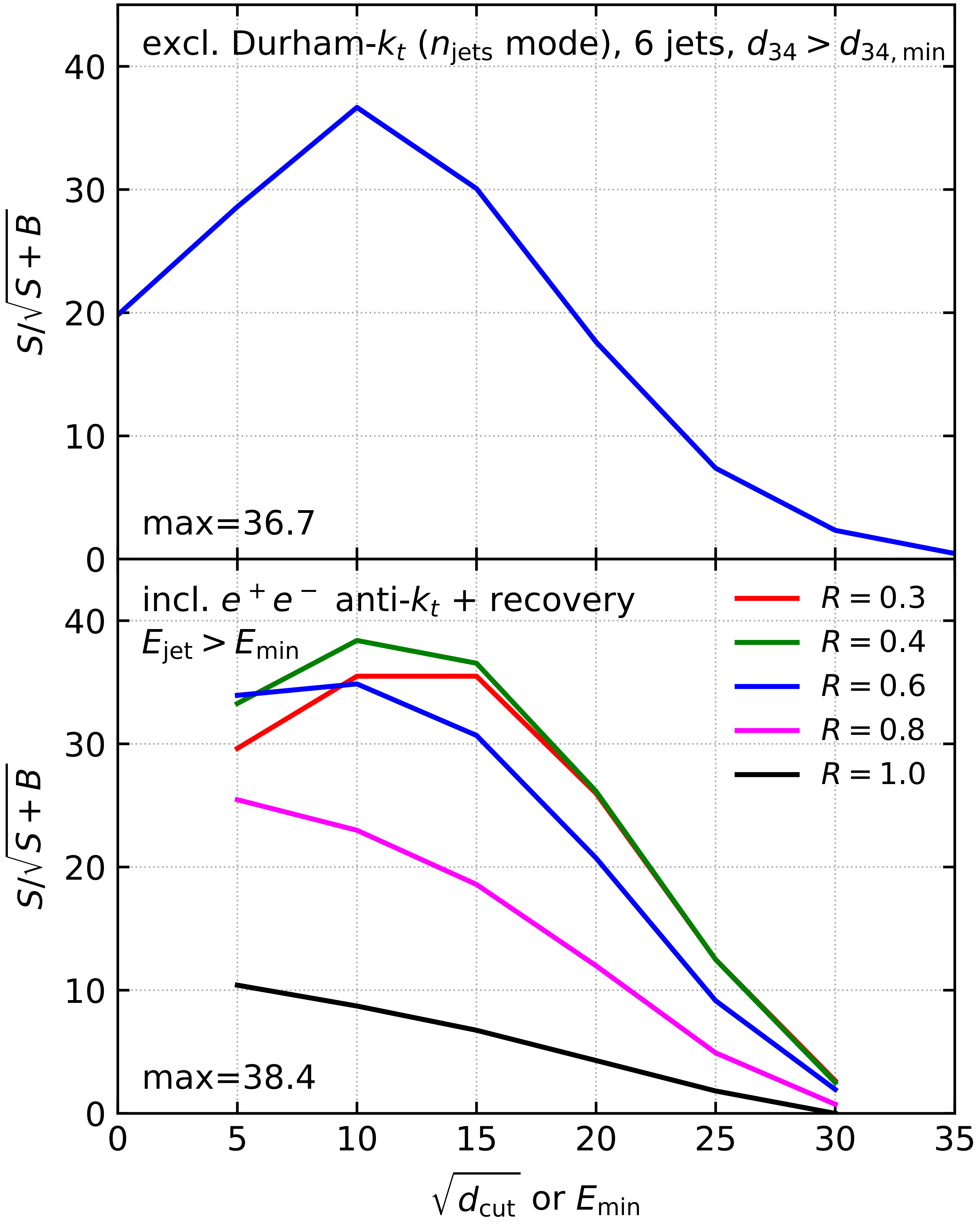
e^+e^- → HZ→WW(→4j)bb analysis

illustration of impact of d_{56} cut

- Dependence on jet-definition parameters & cuts is stronger here than in HZ→bbjj case
- the more jets you have, the more critical the suppression of backgrounds

VERY PRELIMINARY

significance: *HZ*→*WW*(→4*j*)*bb* v. Z/ZZ/WW bkgds



concluding remarks

Other physics points to remember

- use E-scheme recombination, i.e. straightforward 4-momentum sum.
  This is the default in FastJet.
  (Other schemes tend to bias invariant mass reconstructions for resonances — use them only if you have a well motivated reason why they might be better)

- Beware of comparing single-jet energy resolution between algorithms. Prefer resolution for colour-singlet reconstruction (because assignment of particles to one jet or another of the colour singlet decay is a *choice*).

- For QCD studies, the Cambridge algorithm is special. Not shown here, because it's marginally more complicated to operate in an n_{jets} mode (i.e. for resonance studies), but may deserve further investigation.

- For precision physics, flavour is a concept that needs to be defined, e.g. for R_b for Z decays, watch out for contamination from g→bb in "non-b" Z decays (including QM interferences). Algorithms designed to be better for flavour questions are an active research topic.

Other code points to remember

- For the “inclusive” algorithms,
  `auto jets = jet_def(hadrons)`
  returns the jets sorted in decreasing energy (for spherical algorithms) or p_t (for longitudinally invariant algorithms)

- `ClusterSequence::exclusive_jets(…)` and `exclusive_jets_up_to(…)` do not do that (you should manually sort the jets with `sorted_jets = sorted_by_E(jets)`).

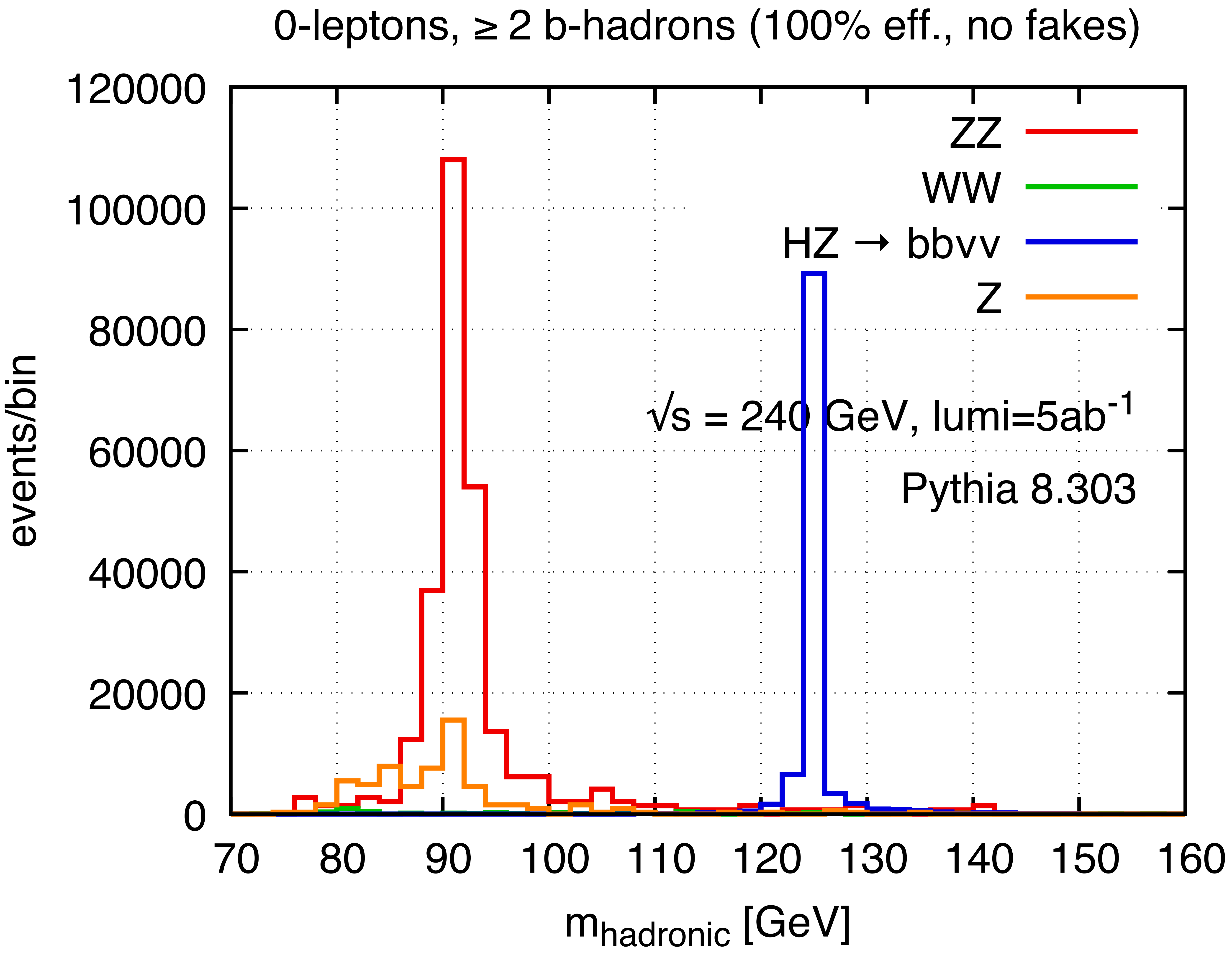
- The reason for the difference is that it’s conceivable that one might want to call `exclusive_jets…` multiple times on a single `ClusterSequence` and the sorting brings an additional overhead: it is up to the user to decide whether they need it.

Overall conclusions

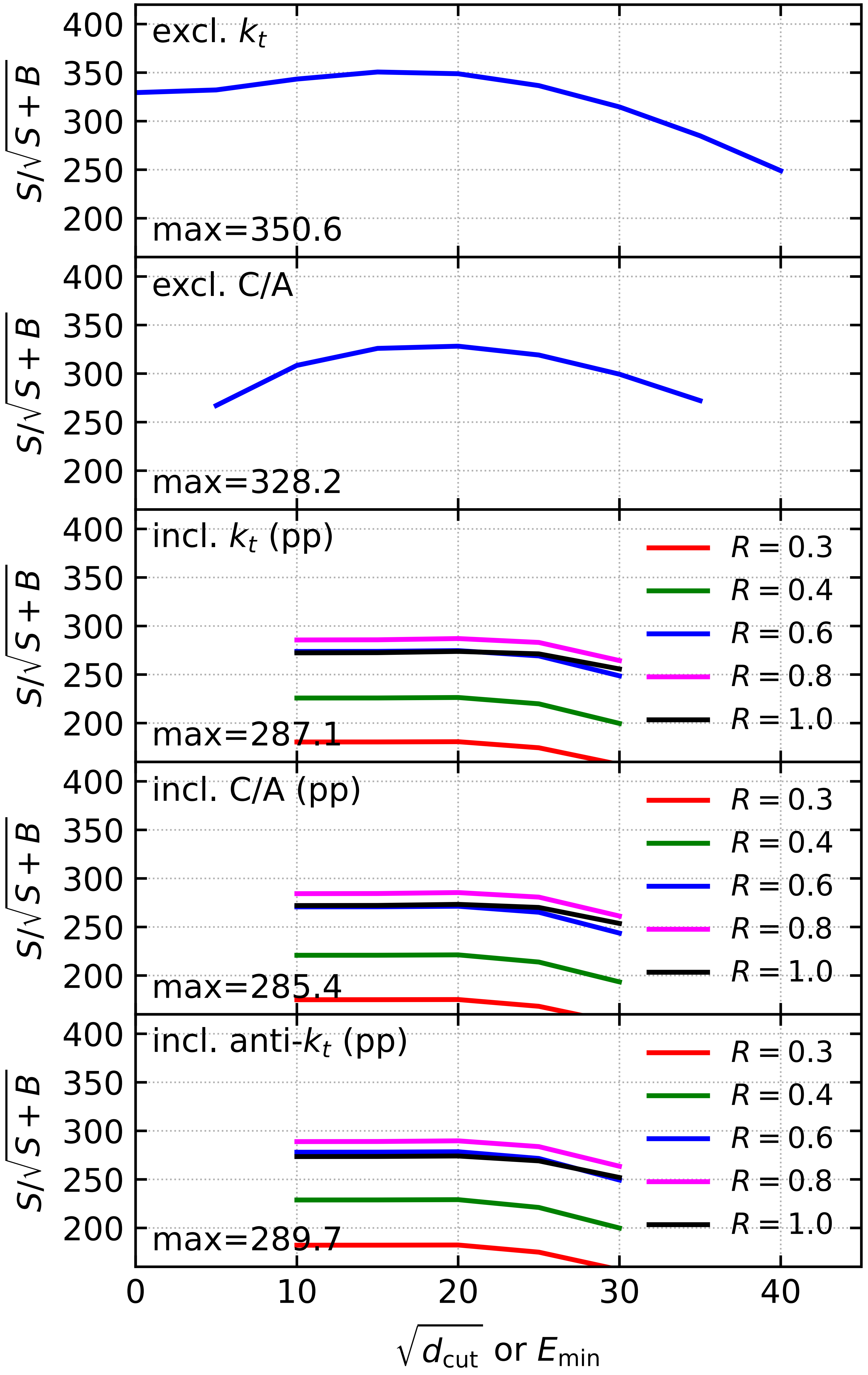
- If you need something simple that just works for EW (& top physics), start with exclusive Durham-k_t (`ee_kt_algorithm`) in the n_{jets} mode, possibly with a cut on $d_{n,n-1}$.
- Other options are possible, e.g. inclusive e^+e^- generalised k_t algorithms with energy recovery from jets beyond the n_{jets} of interest. In the simple studies for this talk, we found slight improvements relative to exclusive Durham-k_t
  There are many ways of implementing energy recovery (and care is needed to avoid infrared divergences). Is the benefit worth this? What is best suited to ML?
- For resonance studies, beware of using variables related to jet substructure (e.g. d_{45} for a 4-jet LO structure): they will make things much more sensitive to QCD details.
- QCD studies are a separate subject, with other choices probably optimal (e.g. Cambridge family of algorithms)

backup

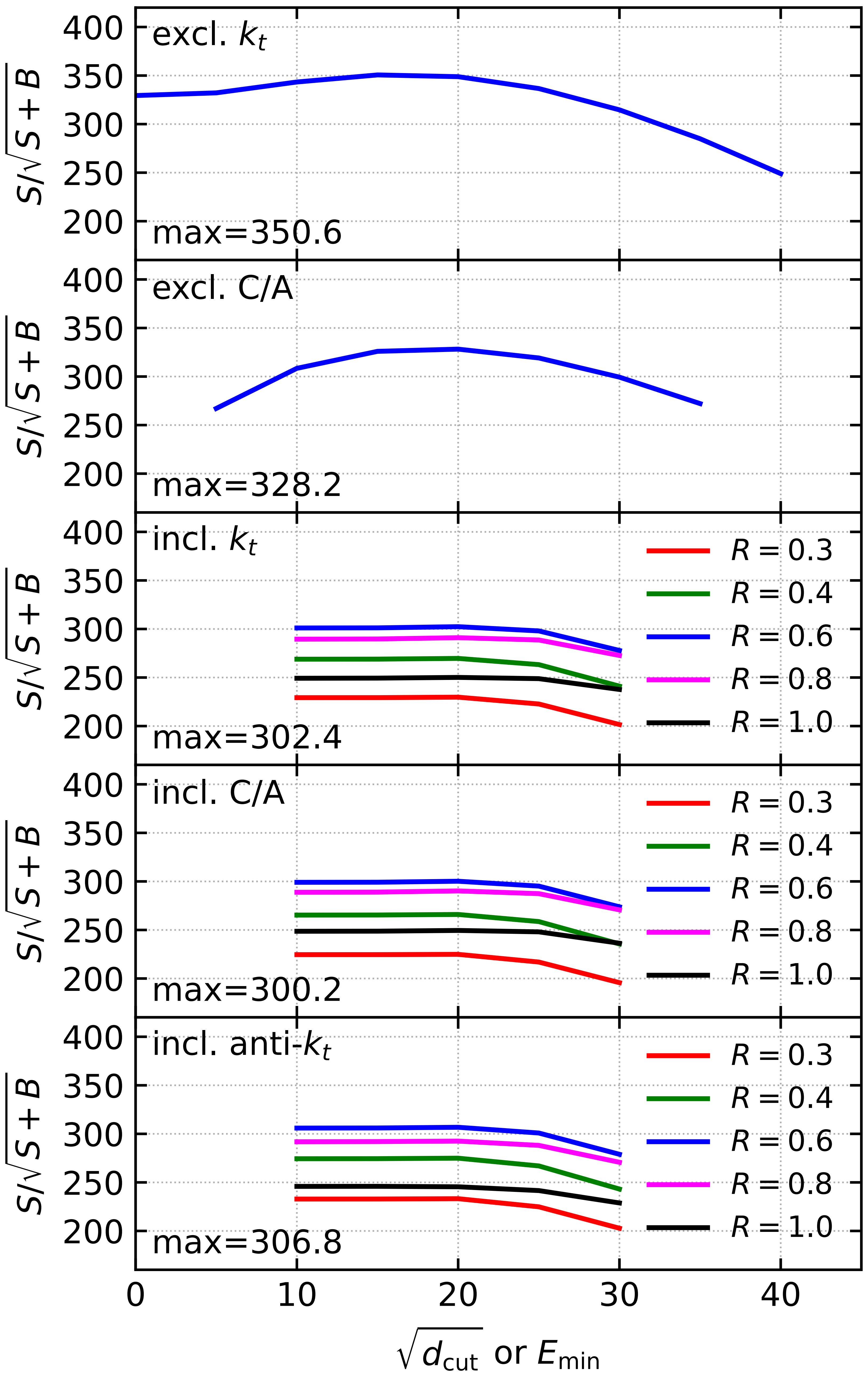
HZ → bbνν, signal & backgrounds



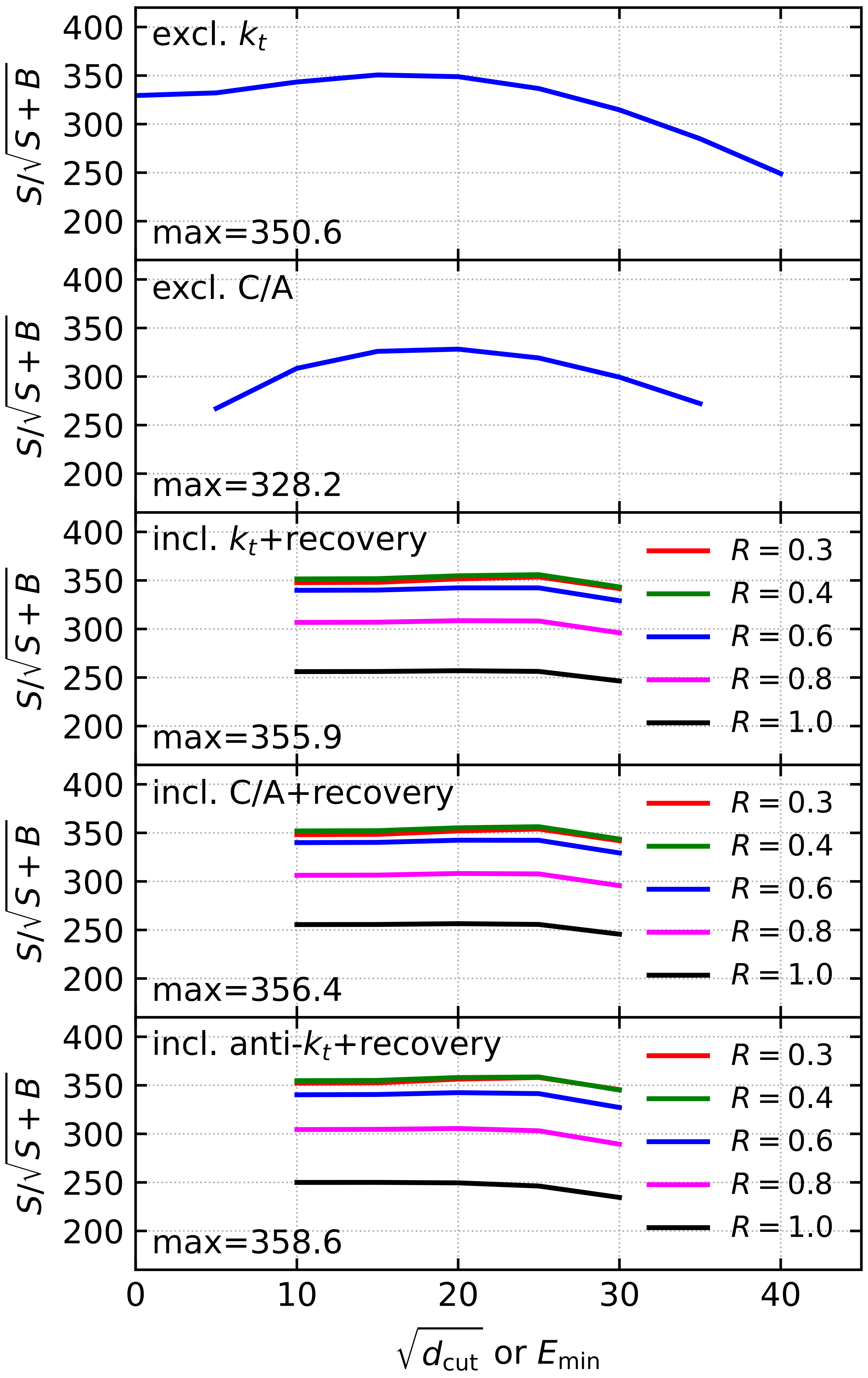
$e^+e^- \rightarrow HZ \rightarrow bbjj$



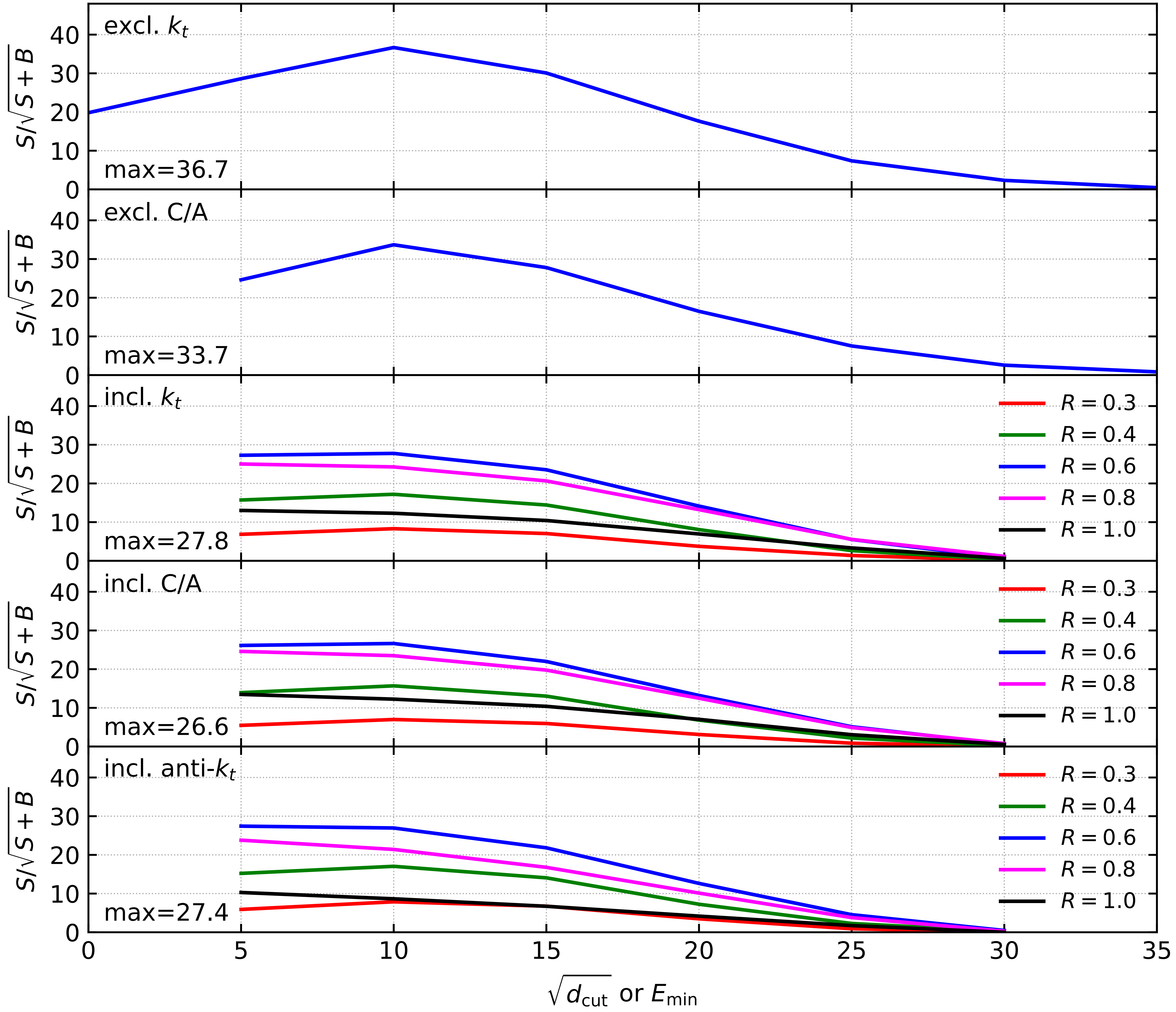
$e^+e^- \rightarrow HZ \rightarrow bbjj$



$e^+e^- \to HZ \to bbjj$



$e^+e^- \to HZ \to WW(\to 4j)bb$ analysis



$e^+e^- \to HZ \to WW(\to 4j)bb$ analysis