Modelling heavy neutral leptons in accelerator beamlines

Komninos-John Plows and Xianguo Lu

DUNE BSM WG meeting, 13/Dec/2022

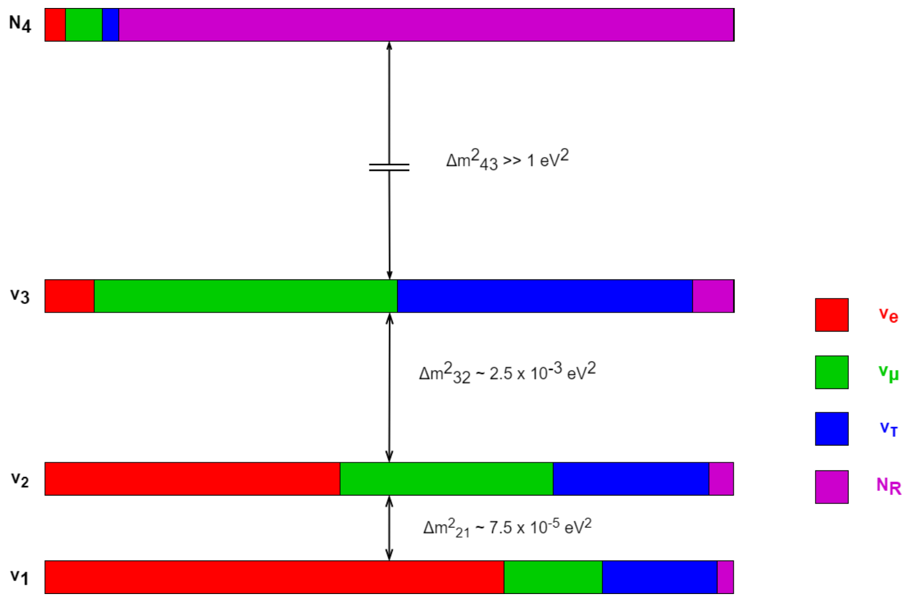
Based on
KJP and XL, “Modelling heavy neutral leptons in accelerator beamlines”, arXiv: 2211.10210

UNIVERSITY OF OXFORD WARWICK

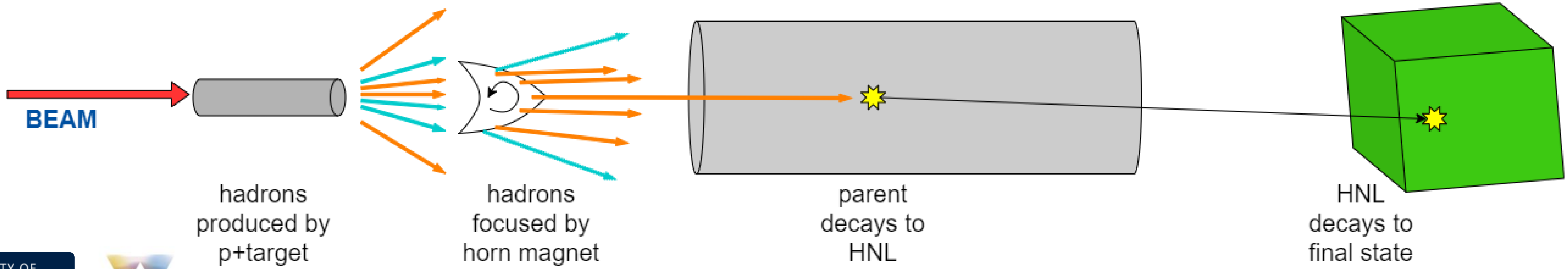
Dramatis personae

- Heavy Neutral Leptons (HNL) : nearly sterile neutrino mass eigenstates
  - Mass range determined by seesaw mechanism
  - $M_{\mathrm{N4}} \geq \mathcal{O}(0.1 - 1\ \mathrm{GeV}/c^2)$

Assume only 1 relevant state, N4

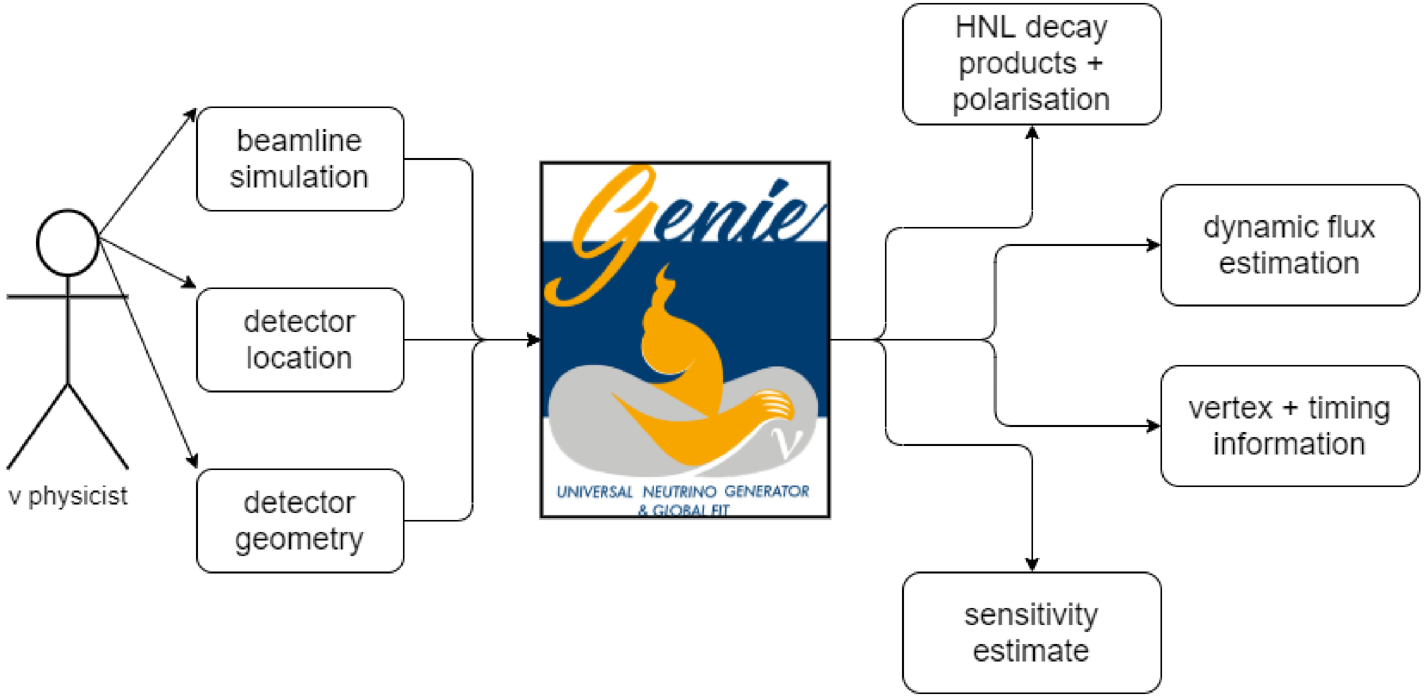


- Production:
  - Indirect (e.g. in 0νββ, upscattering)
  - Direct (e.g. in colliders, in particle decays)



The goal

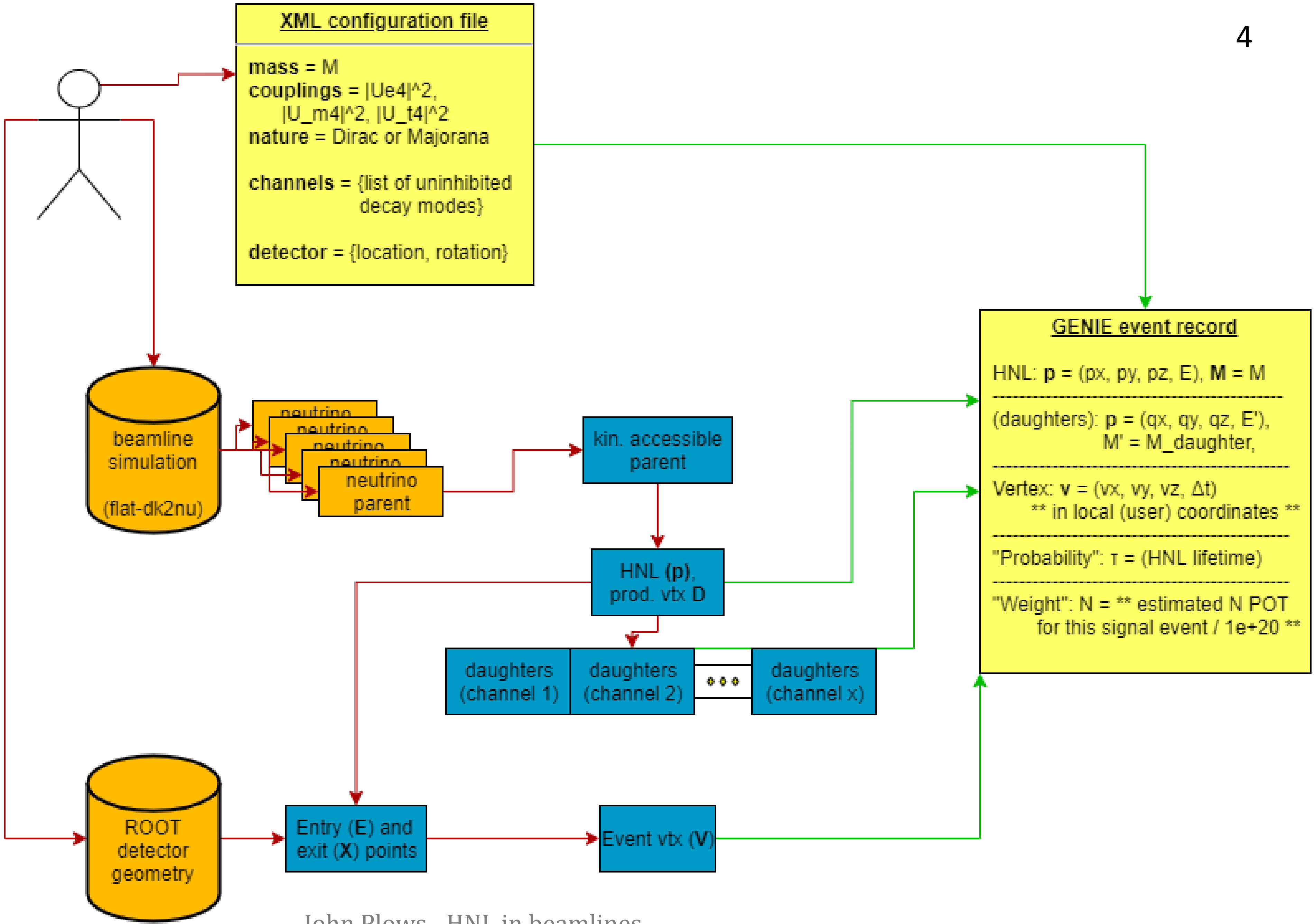
- Reduce model dependence of HNL descriptions
  - Production/decay widths depend on specific Lagrangian **but…**
  - Emission, propagation, arrival : determined by kinematics



- Implemented HNL generator into GENIE (using widths from *EPJ* **C 81** (2021) 78)

Inputs are <u>factorized</u>:

- Beamline sim
- Detector geometry
- Runtime options (e.g. HNL param space)



Emission

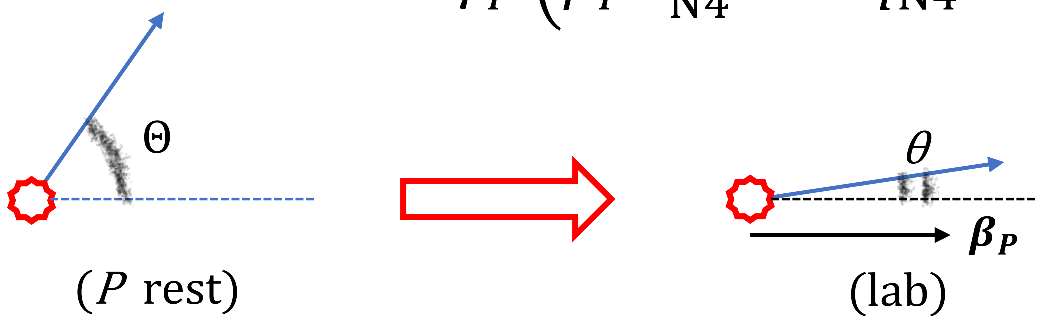
- Pseudoscalar meson decay $P \rightarrow N_4 + \ell$ ($+$ pseudoscalar D)
  - Lorentz boost from parent-rest frame into lab frame dominant factor
  - For a massive neutrino

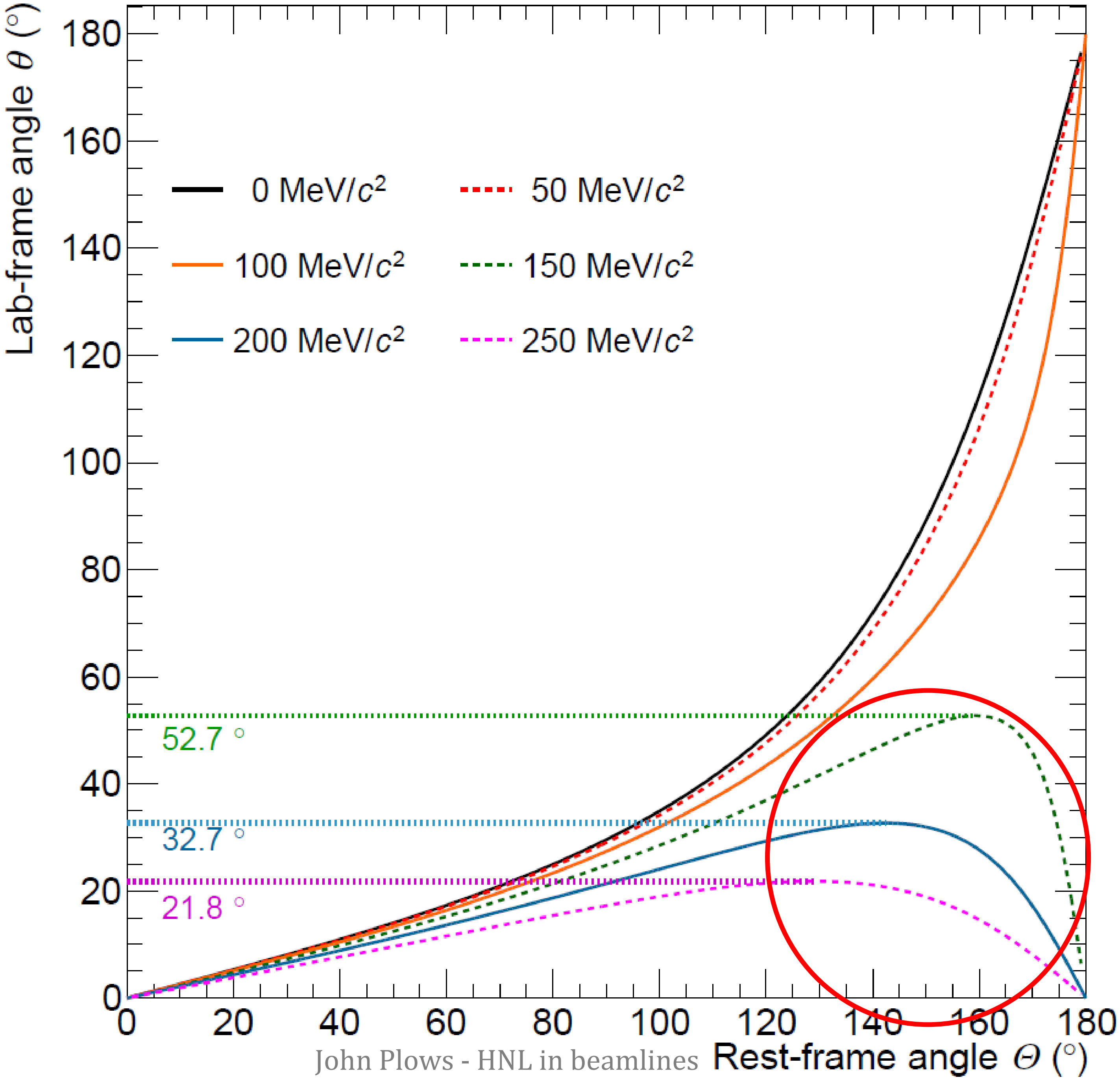
$$\mathcal{B} = \frac{E_{\mathrm{N4}}}{E_{\mathrm{N4}}^{(\mathrm{CM})}} = \frac{1}{\gamma_P(1 - \beta_P \beta_{\mathrm{N4}} \cos\theta_{\mathrm{det}})} \; (1), \; \beta_{\mathrm{N4}} \text{ lab} - \text{frame}$$

$$(\text{cf. } \beta_{\mathrm{N4}} = 1 \text{ for SM})$$

  - Collimation effect:

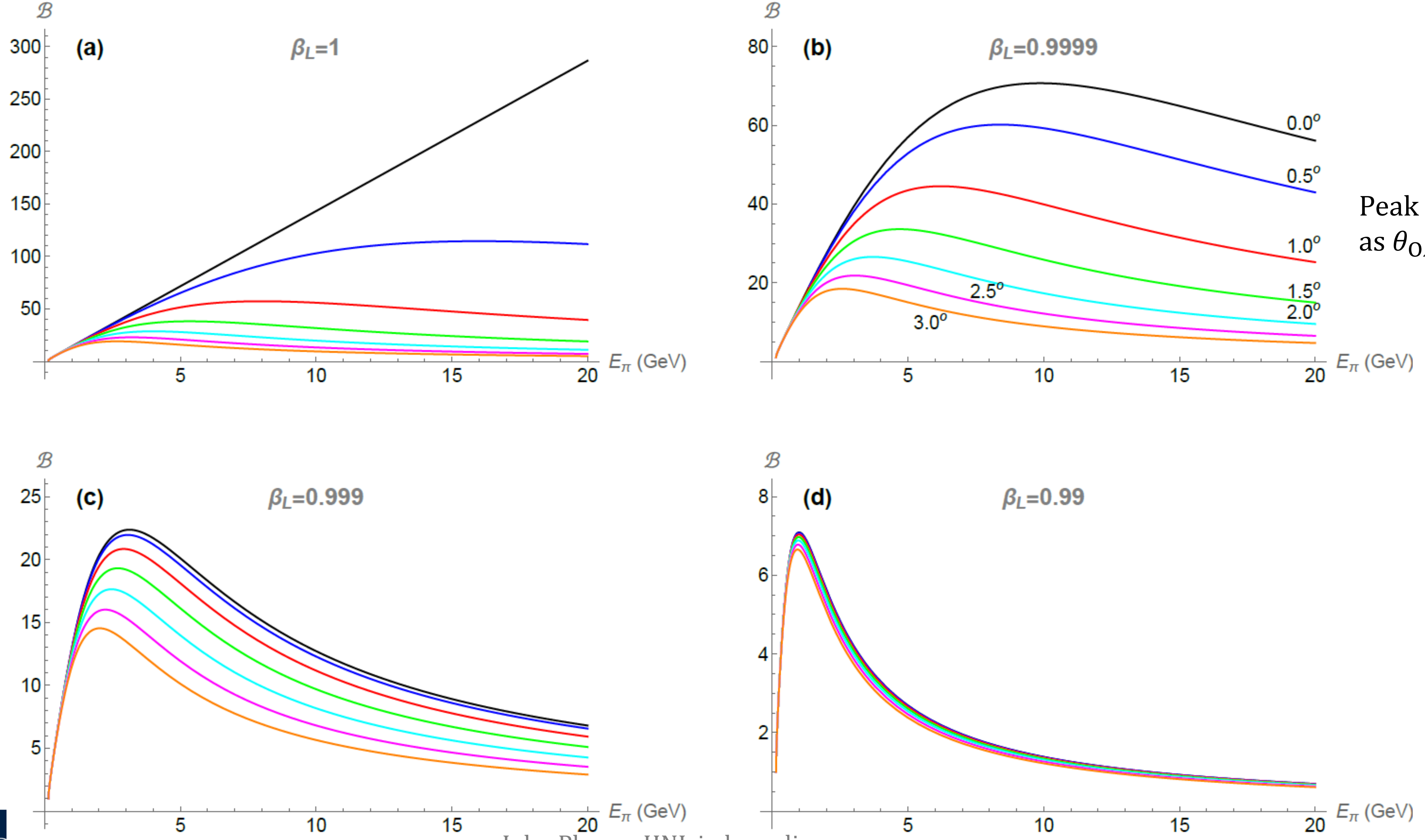
$$\tan\theta = \frac{q_{\mathrm{N4}} \sin\Theta}{\gamma_P\left(\beta_P E_{\mathrm{N4}}^{(\mathrm{CM})} + q_{\mathrm{N4}} \cos\Theta\right)} \; (2)$$



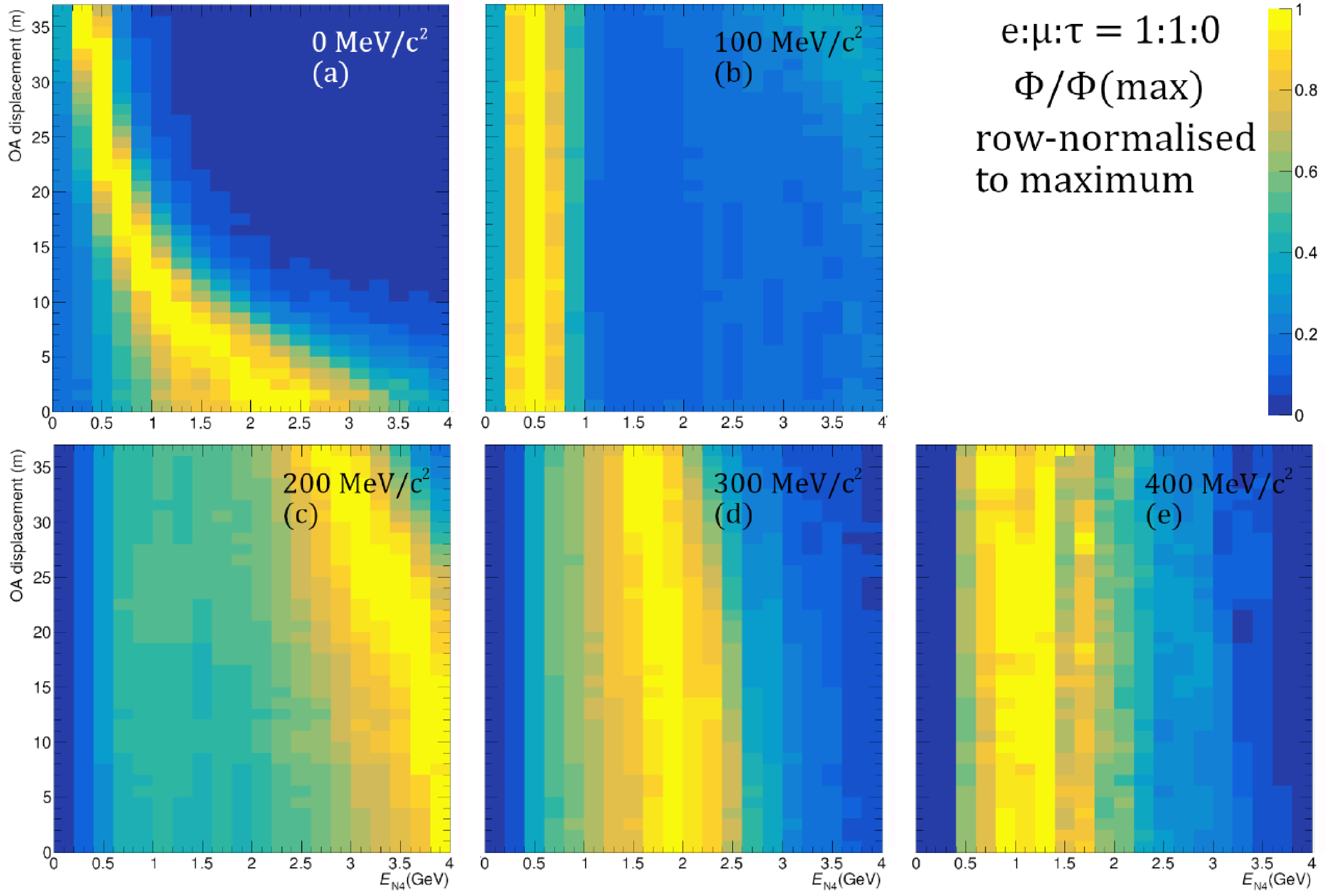


Collimation function dips back down! ⇒ <u>acceptance correction</u> needed

Presence of β_{N4} term in (1) weakens off axis effect compared to SM



Peak $\mathcal{B}$ falls as θ_{OA} rises



Flux shapes, using ND baseline 575m from origin @ beam angle = 5.8° downwards

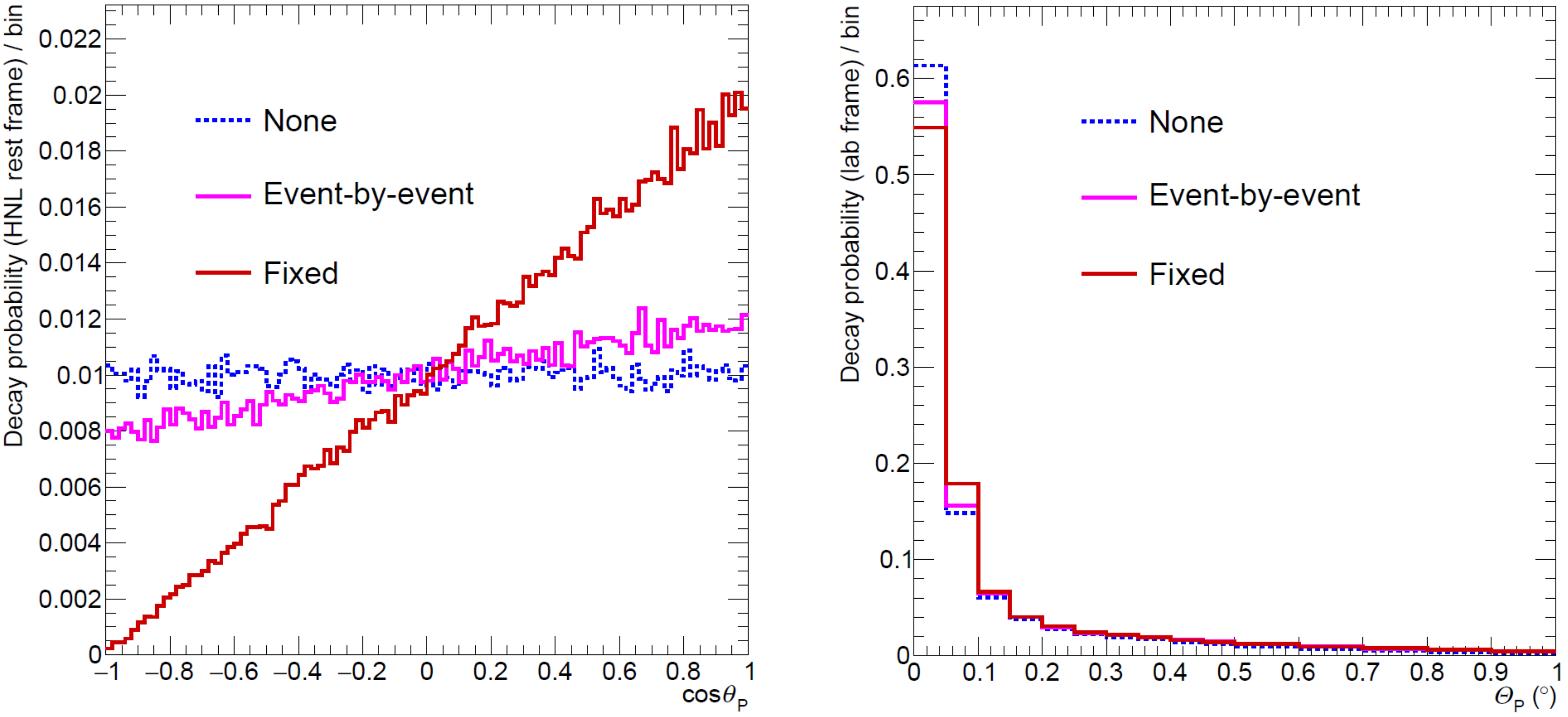
OA effect weakens most close to threshold (panels b, e): heavier HNL are slower

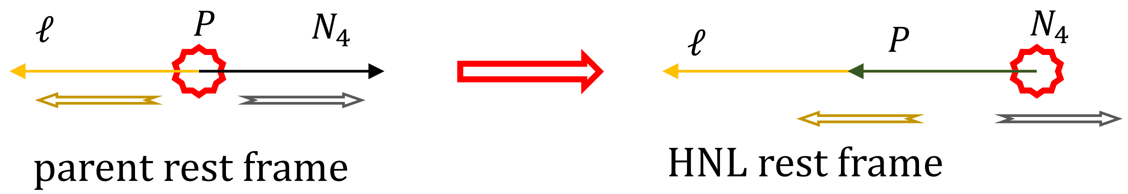
Decay to SM

- User chooses "signal events" (= interesting channels); code inhibits the rest of the decays (but keeps track of their widths)
- Kinematically accessible channels put in pool of candidates chosen at random based on the calculated decay widths
- Daughters of chosen channel generated and added to event record

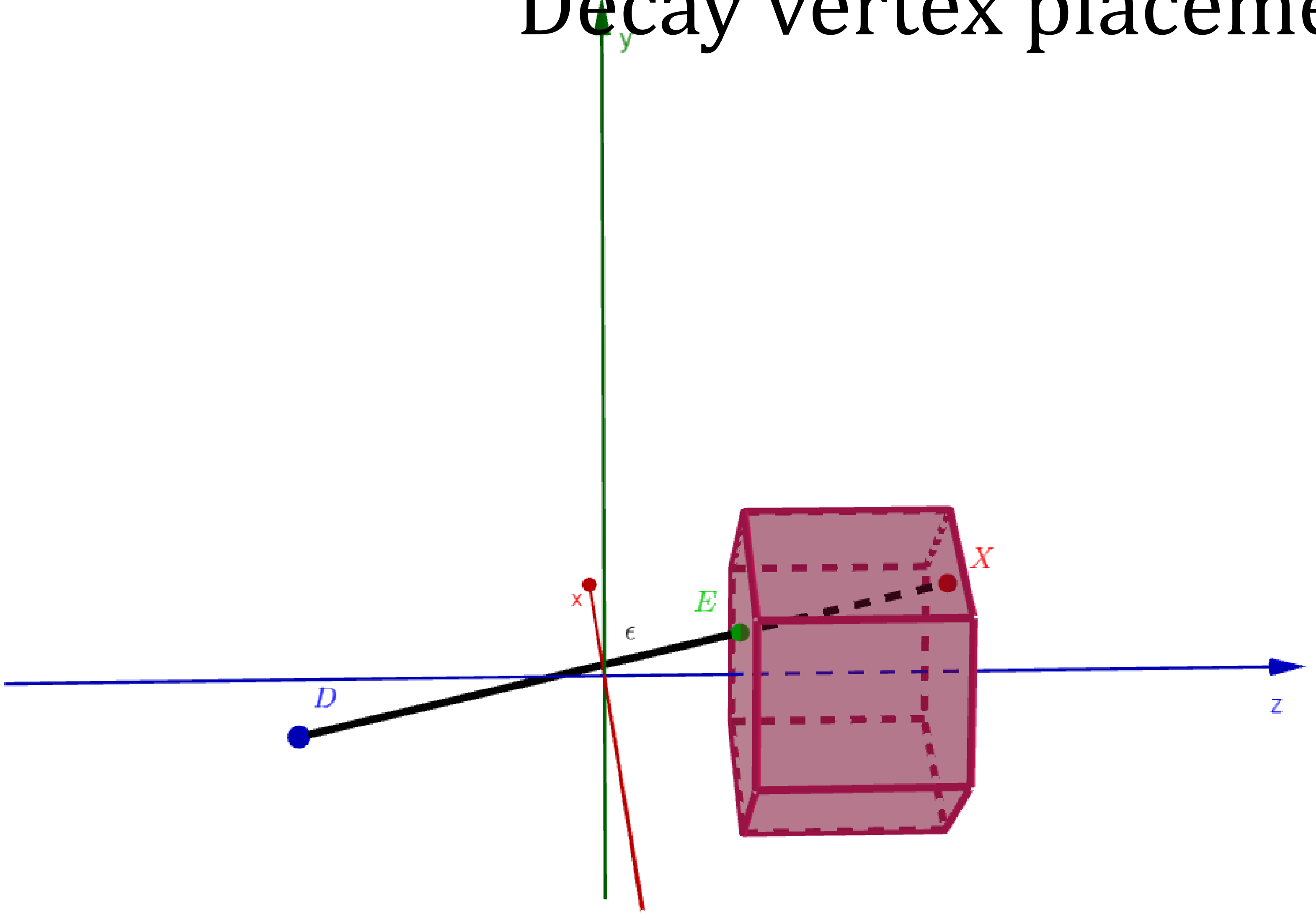
- If polarisation enabled and Dirac HNL, do a "2-to-2" polarised decay based on arXiv: 1805.06419 (valid for 2-body production and 2-body decay of HNL, not strictly correct otherwise)

Polarisation direction kept track of at HNL emission stage





Decay vertex placement



Known: production vertex **D**, HNL momentum $\boldsymbol{p}_{\mathbf{N4}}$,

Sought: Decay vertex **V** such that:
V on HNL trajectory
Conditional probability

$$\frac{\mathrm{P}(\ell_u \in [\ell_E, \ell_V])}{\mathrm{P}(\ell_u \in [\ell_E, \ell_X])}$$

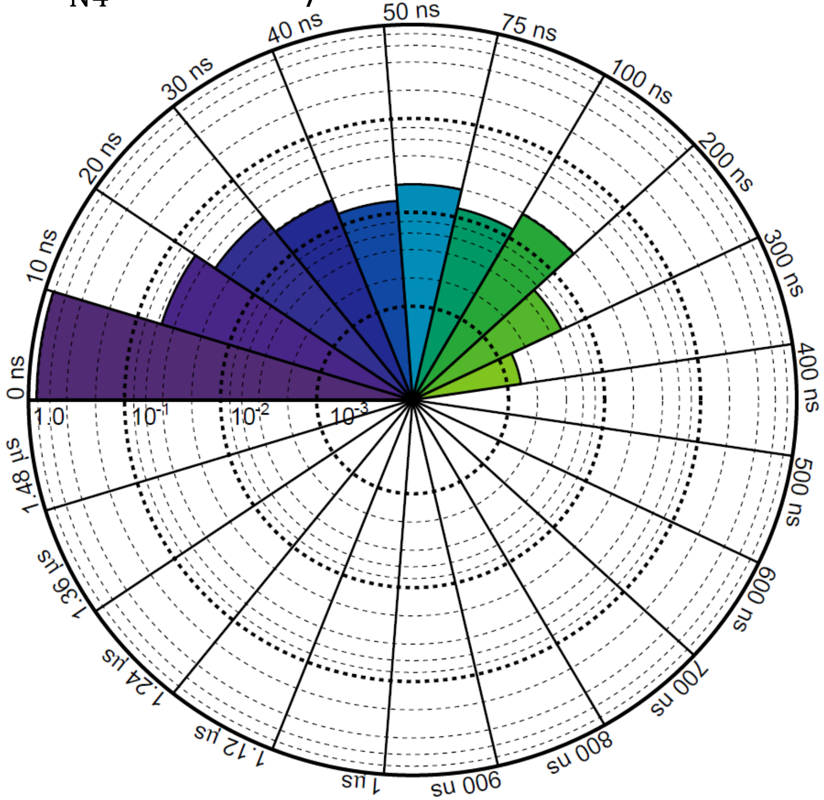
follows exponentially decaying distribution

$$p(\ell_u) = 1 - \exp\left(-\frac{\ell_u - \ell_E}{\beta c \gamma \tau}\right)$$

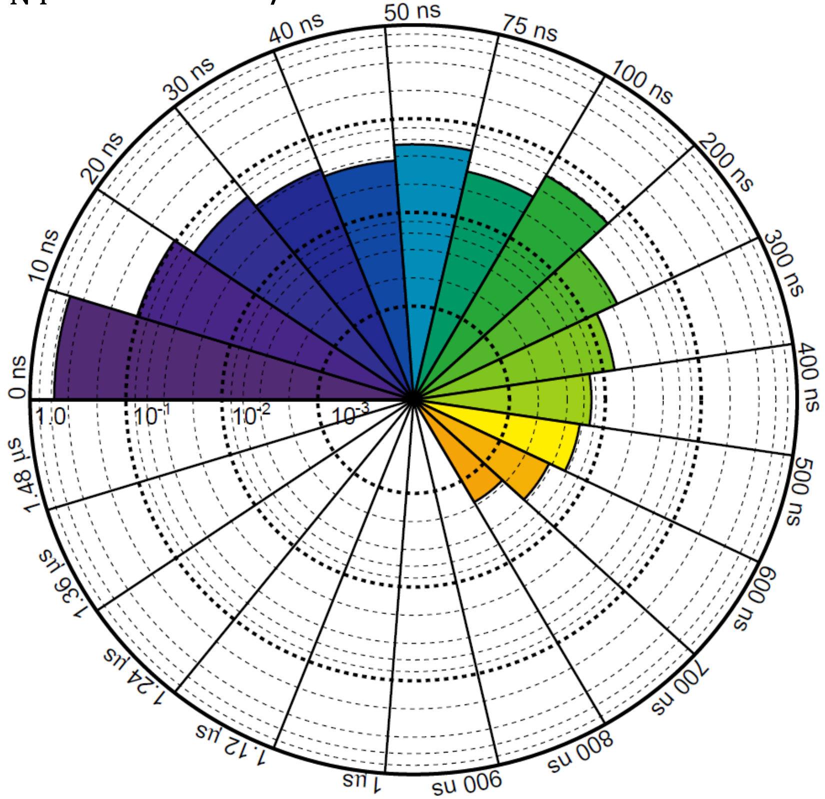
Solution: Map uniform $u \in U(0,1)$ to CDF F of the exponential decay, and get "elapsed length"

$$\ell_u = F(u) \cdot (\ell_X - \ell_E) + \ell_E$$

$M_{N4} = 25\ \mathrm{MeV}/c^2$

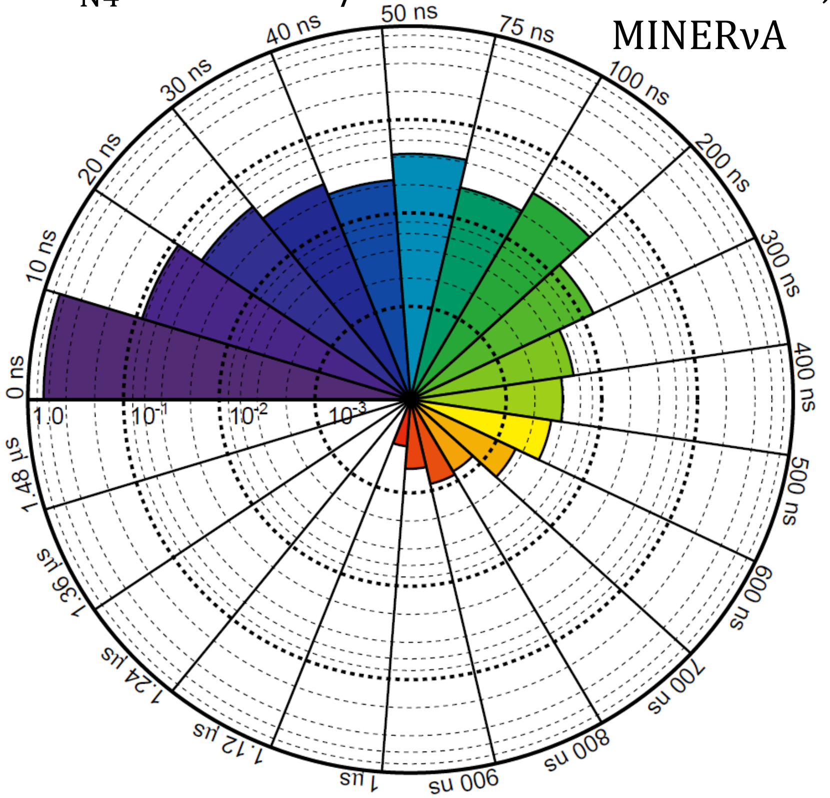


$M_{N4} = 100\ \mathrm{MeV}/c^2$



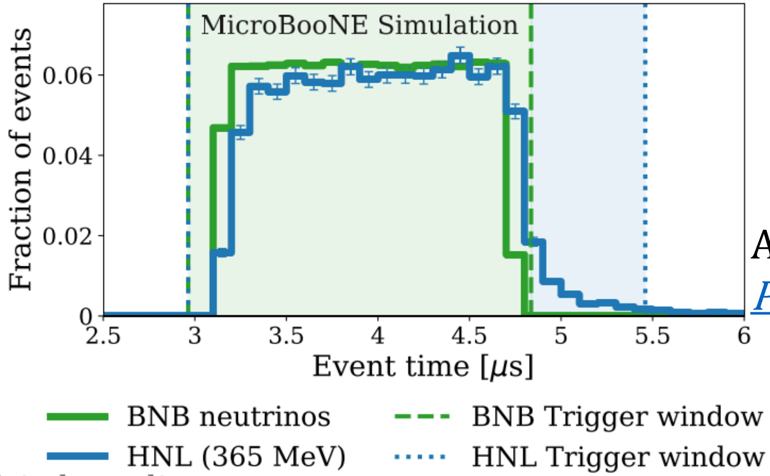
$M_{N4} = 250\ \mathrm{MeV}/c^2$

NuMI beam, MINERvA



Heavier HNL arrive later because they are slower!
⇒ timing triggers possible to reduce SM background

<u>Advantage</u>: Calculation is <u>exact</u> (so up to user exactly what kind of trigger they want to use)



Abratenko et al, *Phys. Rev. D* **101** (2020) 38

UNIVERSITY OF OXFORD

WARWICK

The advantages of BeamHNL

1. Direct interface with GENIE + ghep EventRecord output
   1. Easy for simulation chains!
   2. Makes use of GENIE config ⇒ can iterate over parameter space with arbitrary precision / mixing hypotheses by editing a single file!
2. Factorised input
   1. Beamline simulation can be as sophisticated or as simple as one wants
   2. Detector geometry is likewise “free”
3. Additional tools apart from gevgen_hnl
   1. “gevgen_pghnl” particle gun ready to use (specify original trajectory in config and fire!)
   2. “gevald_hnl” validation App with 3 tests: flux prediction (gives histos of HNL spectra by parentage), decay (gives daughter spectra), geometry (gives TTree with details on entry & exit vertices, decay vertex)

Thank you!

Comments, questions welcome :-)

Backup

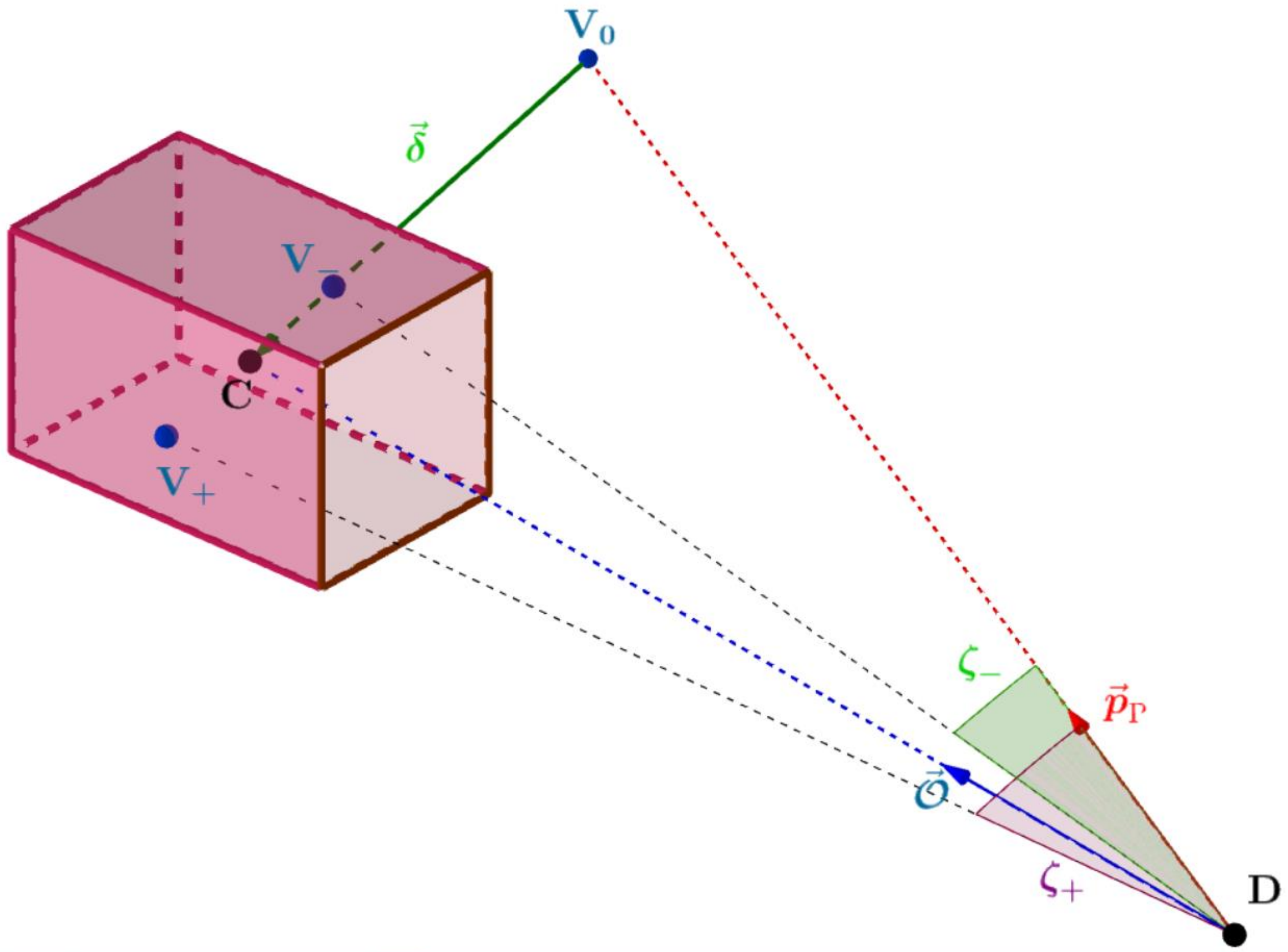
Example: Dirac HNL, $M_{N_4} = 200\ \mathrm{MeV}/c^2, |U_{e4}|^2 = \left|U_{\mu4}\right|^2 = 10^{-7}$

```
|---------------------------------------------------------------------------------------------------------------|
|GENIE GHEP Event Record [print level:   3]                                                                    |
|---------------------------------------------------------------------------------------------------------------|
| Idx |          Name | Ist |        PDG |    Mother   | Daughter   |      Px |      Py |      Pz |       E |      m |
|---------------------------------------------------------------------------------------------------------------|
|   0 |           HNL |   0 | 2000020000 |  -1 |  -1 |   2 |   1 |   0.003 |  -0.262 |   4.447 |   4.459 |  0.200 |
|   1 |           pi+ |   1 |        211 |   0 |  -1 |  -1 |  -1 |   0.007 |  -0.182 |   3.845 |   3.852 |  0.140 |
|   2 |            e- |   1 |         11 |   0 |  -1 |  -1 |  -1 |  -0.004 |  -0.081 |   0.602 |   0.608 |  0.001 |
|---------------------------------------------------------------------------------------------------------------|
|      Fin-Init:                                                    |   0.000 |  -0.000 |   0.000 |   0.000 |        |
|---------------------------------------------------------------------------------------------------------------|
|      Vertex:          HNL @ (x =     0.49658 m, y =    -0.32563 m, z =     7.64406 m, t =    3.435691e-09 s)    |
|---------------------------------------------------------------------------------------------------------------|
| Err flag [bits:15->0] : 0000000000000000    |  1st set:                                                  none |
| Err mask [bits:15->0] : 1111111111111111    |  Is unphysical:    NO |   Accepted:   YES                      |
|---------------------------------------------------------------------------------------------------------------|
| sig(Ev) =       0.00000e+00 cm^2  | dsig(Ev;{K_s})/dK   =     0.00000e+00 cm^2/{K}   | Weight =          0.09407 |
|---------------------------------------------------------------------------------------------------------------|
```

Geometry: MINERνA inner detector (USER ∥ NEAR)

Using NuMI beam: rotated downwards by 0.05830 rad -- $\sin^{-1}(0.262/4.447) \simeq 0.05895$ rad
$t = 3.44$ ns : means $\Delta t \coloneqq t(\text{HNL arrival}) - t(\text{SM } \nu \text{ arrival}) = 3.44$ ns (useful for timing studies)
Weight $= 0.09407$: means that for this signal event, estimated 0.09407×10^{20} POT needed

Angular deviation calculation



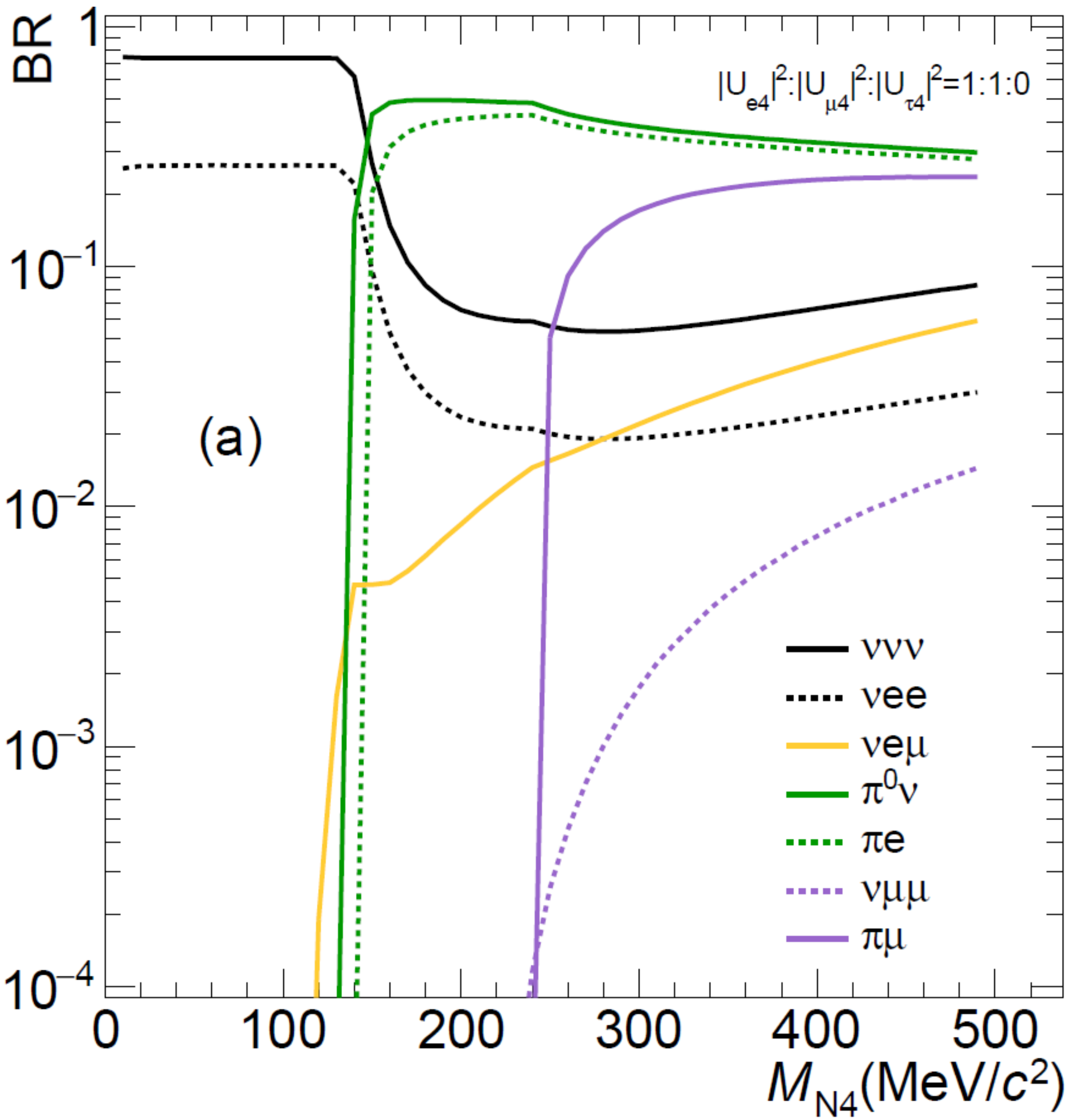
Known: production vertex **D**, parent momentum $\boldsymbol{p_P}$, detector centre **C**

Sought: angles $\zeta_\pm$ such that

$$\langle \boldsymbol{p}_{\mathbf{N4}}, \boldsymbol{p_P} \rangle \equiv \zeta \in [\zeta_-, \zeta_+]$$

$$\Leftrightarrow N_4 \text{ accepted}$$

Solution: Estimate by constructing "sweep" $\boldsymbol{\delta}$ from point $V_0(z = z_C)$ to C and calculating intersections $V_\pm$



Channel calculation

- Calculate decay widths once per run, use them to store branching ratios
  (= map of channels -> widths)
- Obtain reduced map (uninhibited channels only)
- Map uniform random $u \in U(0,1)$ to channel based on score $s_{i+1} = \Gamma_{\mathrm{channel}}/\Gamma_{\mathrm{tot}} + s_i, s_0 = 0$

E.g: at $M_{\mathrm{N4}} = 200\ \mathrm{MeV}/c^2$ with $|U_{e4}|^2 = |U_{\mu4}|^2, |U_{\tau4}|^2 = 0$ there are 5 available channels $\nu\nu\nu, \nu ee, \nu e\mu, \pi^0\nu, \pi e$.
Suppose user wants $\nu ee, \pi e$.
Full map (widths in GeV, $|U_{\alpha4}|^2 = 10^{-6}$):
{ (ννν, 1.46257e-23), (νee, 5.22082e-24), (νeμ, 1.87204e-24), ($\pi^0\nu$, 1.08503e-22), (πe, 9.13852e-23) }
Reduced map:
{ (νee, 5.22082e-24), (πe, 9.13852e-23) }
Scores:
{ (νee, 0.054), (πe, 1.0) }
⇒ 5.4% of simulated events are νee, 94.6% πe

Implemented HNL decay channels

$N \to \nu\nu\nu$, invisible	$M_N \geq 0$ MeV/c^2
$N \to \nu e^{\pm} e^{\mp}$, electron-like	$M_N \geq 1.022$ MeV/c^2
$N \to \nu e^{\pm} \mu^{\mp}$, mixed-lepton	$M_N \geq 106.169$ MeV/c^2
$N \to \pi^0 \nu$, photon-like	$M_N \geq 134.973$ MeV/c^2
$N \to \pi^{\pm} e^{\mp}$, single-e	$M_N \geq 140.081$ MeV/c^2
$N \to \nu \mu^{\pm} \mu^{\mp}$, muon-like	$M_N \geq 211.316$ MeV/c^2
$N \to \pi^{\pm} \mu^{\mp}$, single-mu	$M_N \geq 245.229$ MeV/c^2
$N \to \pi^0 \pi^0 \nu$, 2pi0i	$M_N \geq 269.948$ MeV/c^2
$N \to \pi^0 \pi^{\pm} e^{\mp}$, 2pie	$M_N \geq 275.055$ MeV/c^2
$N \to \pi^0 \pi^{\pm} \mu^{\mp}$, 2pimu	$M_N \geq 380.202$ MeV/c^2

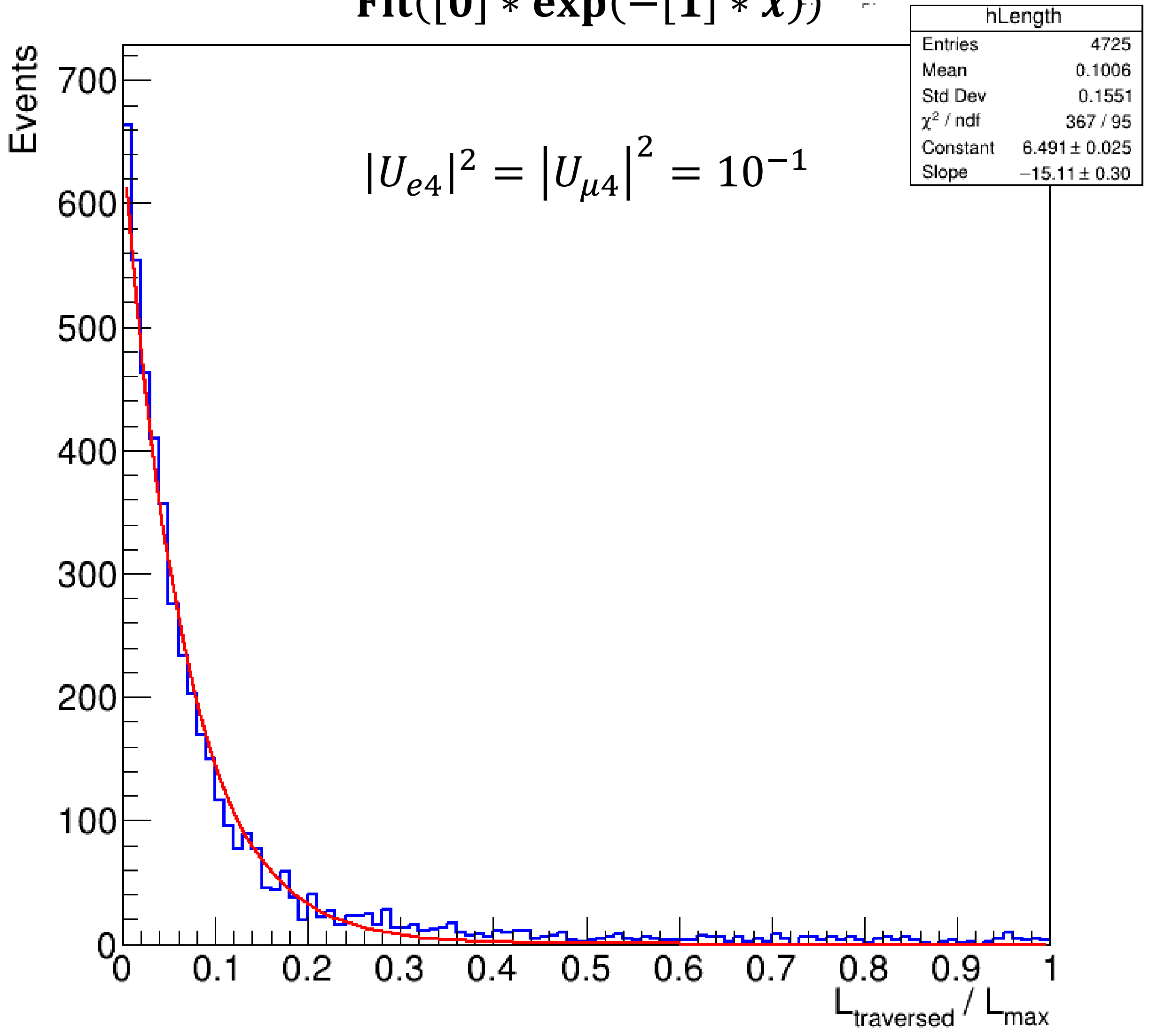
Upon creation of a `SimpleHNL` object, the `HNLBRCalculator` class calculates for each channel c_i the decay width Γ_i

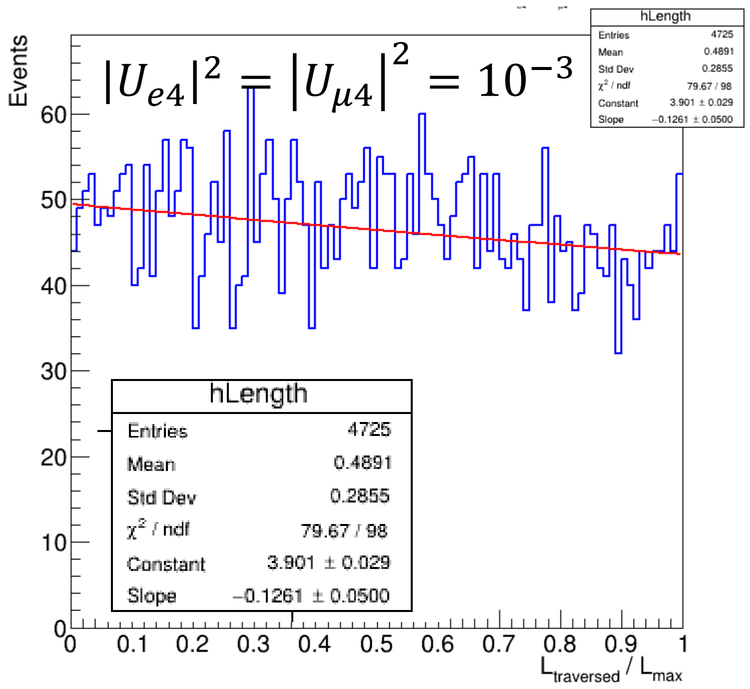
These get inserted into a `std::map<HNLDecayMode_t, double>` that gets attached to the `SimpleHNL` (i.e. a point $\left(M_{N_4}, \{|U_{\alpha 4}|^2\}, \text{isMajorana}\right)$ in parameter space)

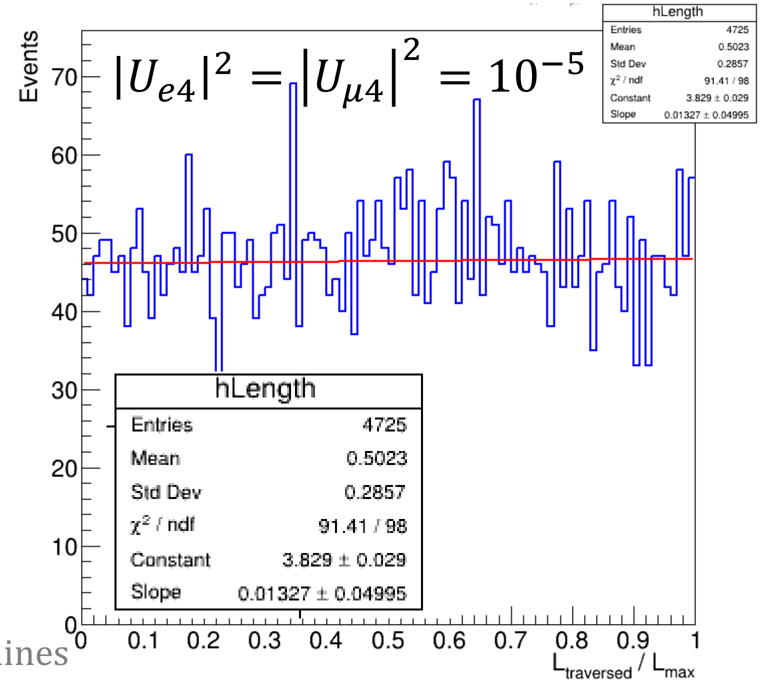
The calculation happens only once for each run of `gevgen_hnl` (one parameter-space point per run, loaded in via config)

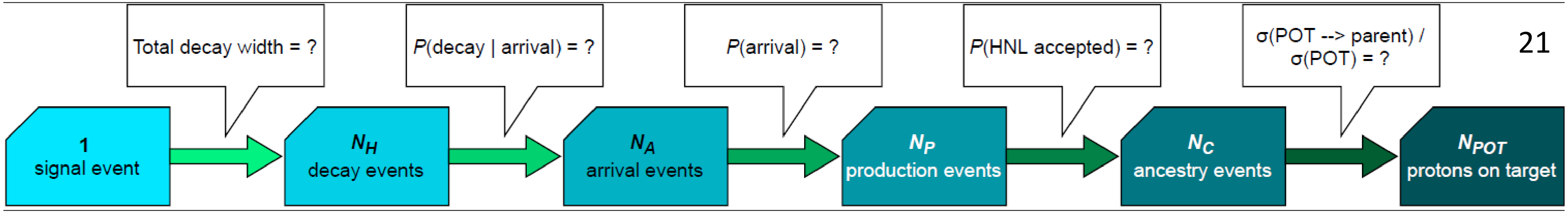
Decay length distributions + fit to expo decay

Fit([0] * exp(−[1] * x))









- Effective POT worked out backwards
  - The "forwards-going" problem (have N POT / x years of exposure, how many signal events?) is in general ill-posed (what's the nature? mass? parameter space? Not straightforward linear scale with $|U_{\alpha 4}|^2$)

- Steps up to N_C self-contained (code works out appropriate multipliers for each of the steps)
- $N_C \to N_{POT}$ step requires external input from beamline
  - N_C is calculated over all hadrons of the event parent's species (e.g. HNL made from K^+, get constraint from all kaon-producing POT).
  - Supply appropriate multiplier: what's $\sigma(\text{POT})/\sigma(\text{POT} \to K^+)$?
  - This can be worked out (in theory) from beamline sim