2021 SIMP Search

HPS Collaboration Meeting

Majd Ghrear

October 20, 2026

Stanford University

Contents

- RC meeting Recap
- Quick updates and dirty laundry
- **Tail fitting**
- **Updated tigh selection optimization**
- ML Methods
- Decisions to make / next steps

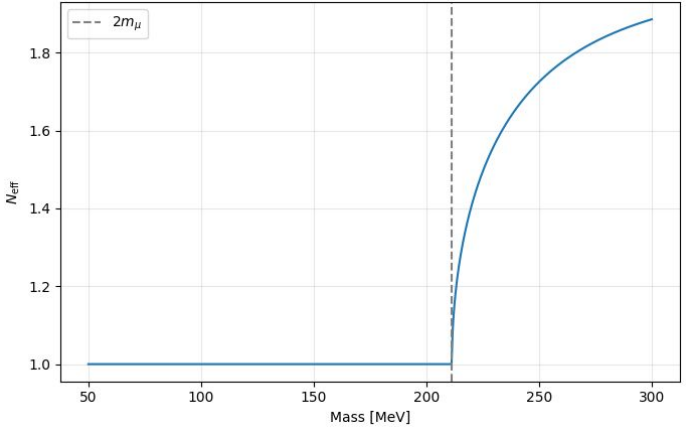
RC meeting Recap

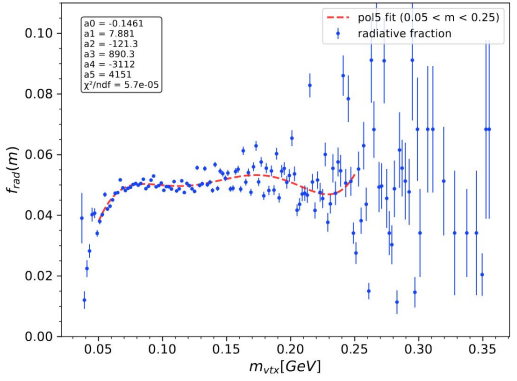
- The SIMP signal estimation has been updated for consistency with the A' analysis
- Optimization procedure
  - $1.5 < P_{sum} < 3.0$ GeV, $z_{vertex} > z_{target}$, $|m_{e^+e^-} - m_{V_D}| < 1.7\sigma(m_{V_D})$
  - Scan grid
  - single selection is chosen for each hit category to minimize the mean epsilon of the Z=1.28 contour
- 1% and 10% optimization completed
  - results suggested tail modeling needed
- ANN on 1% data
  - Good MC/Data agreement
  - similar performance to cut based selection
- Simp model parameters
  - Everything in place for probing m_piD / f_piD
  - New MC needed for m_piD and m_VD
  - Guidance on what to probe

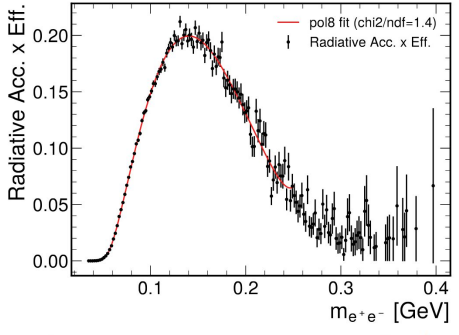
Hit category	VPS	min $\lvert y_0 \rvert_{low}$	Δ min $\lvert y_0 \rvert$
L1L1	2.6 (2.5)	0.27 (0.24)	0.20 (0.10)
L1L2	2.6 (2.8)	0.43 (0.28)	0.15 (0.20)
L2L2	2.2 (3.9)	0.49 (0.36)	0.25 (0.30)
L2L3	1.8 (3.2)	0.47 (0.49)	0.45 (0.25)
L3L3	2.2 (4.5)	0.45 (0.42)	0.45 (0.40)

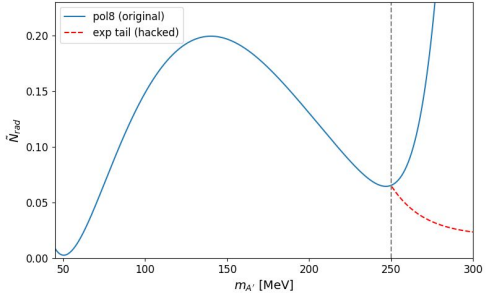
Quick updates and dirty laundry

- Extended search from 250 MeV -> 300 MeV
- f_{rad} = 0.05 assumed above 250 MeV
  - Should we modify this?
- I eyeballed the prompt acceptance factor above 250 MeV
  - Overestimating is conservative
  - Will need a proper extended fit
  - Do not take results => 250 MeV too seriously
- We now account for the di-muon production threshold





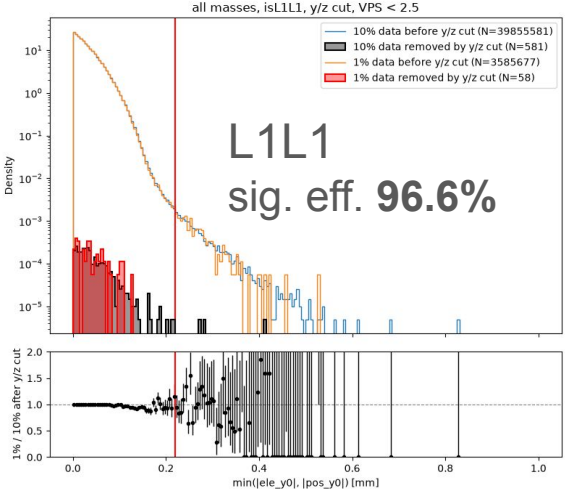


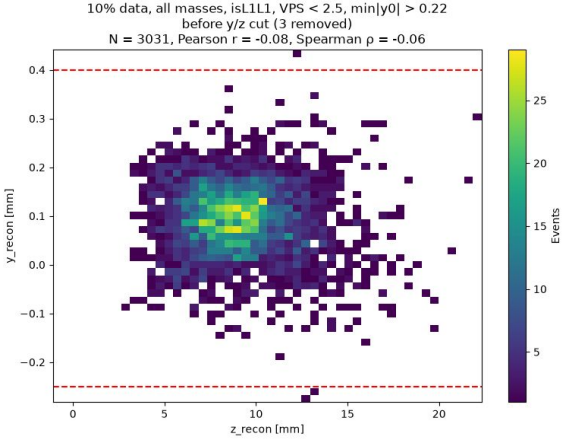


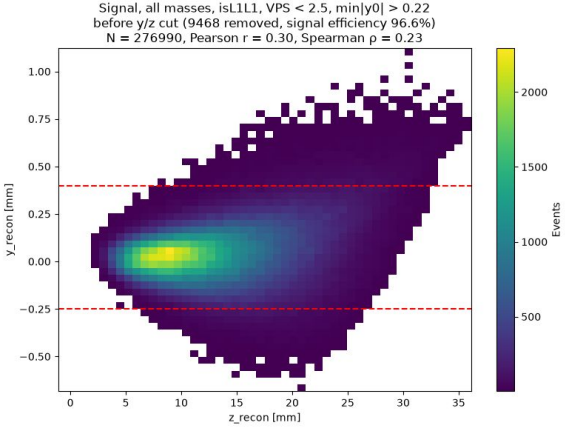
Tail Fitting: Vertex y selection

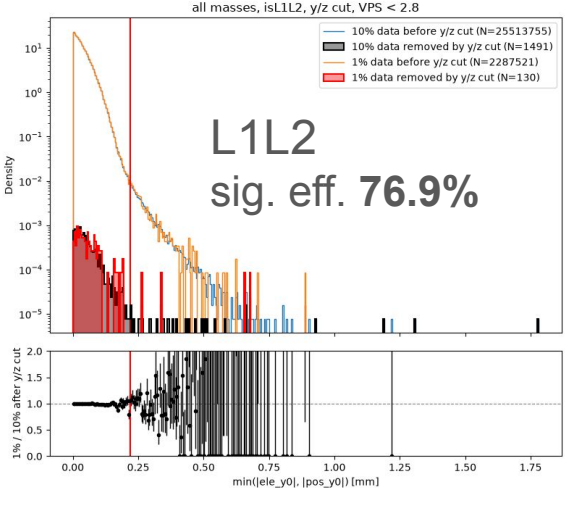
- Early results showed that later hit categories had extended tail that were not well modeled by a single exponential
- Proposed solution: A simple selection on vertex y removing events with vertex y < -0.25mm or vertex y > 0.40 mm
- Vertex y is currently uncalibrated / has run-by-run variations!
  - calibration should give more improvements
- Motivation:
  - Removes trident production on tracker material
  - Vertex y is perpendicular to displacement, so this should not interact with miny0
  - Simple selections are preferred to avoid overfitting to low stat tails!
  - These cuts keep good signal efficiency
  - Deep tail events include cases with low vertex z but high/low vertex y

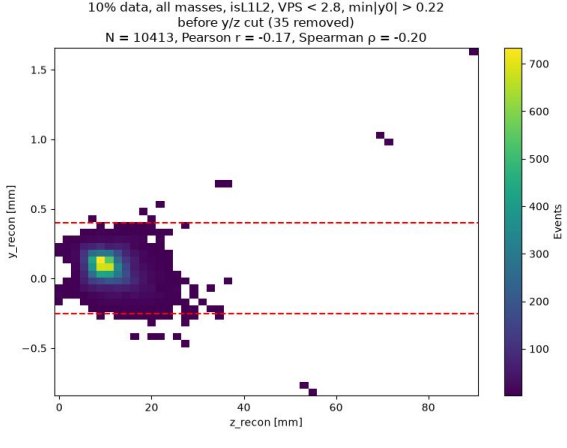
Tail Fitting: Vertex y selection

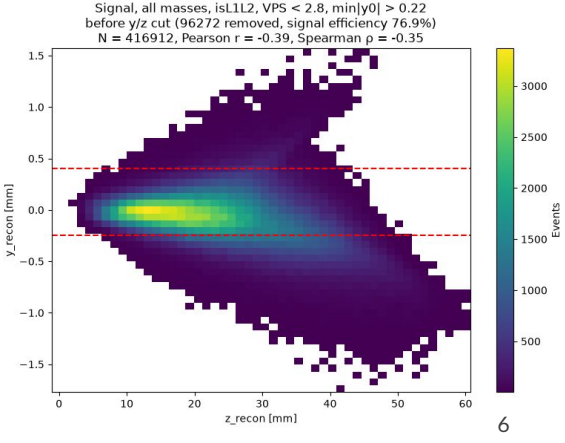




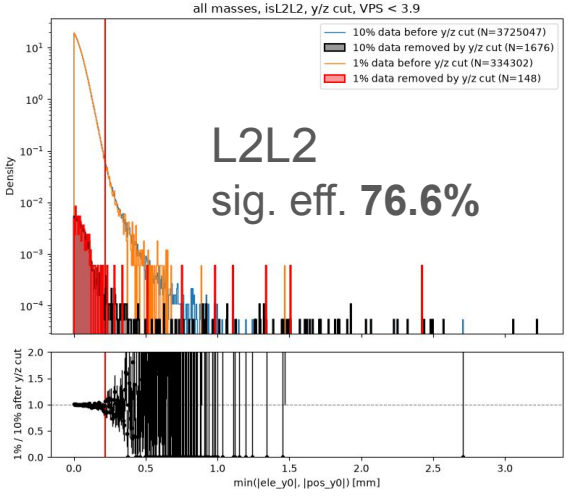


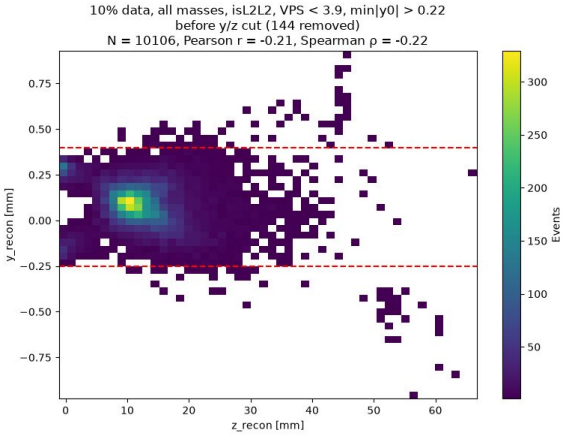


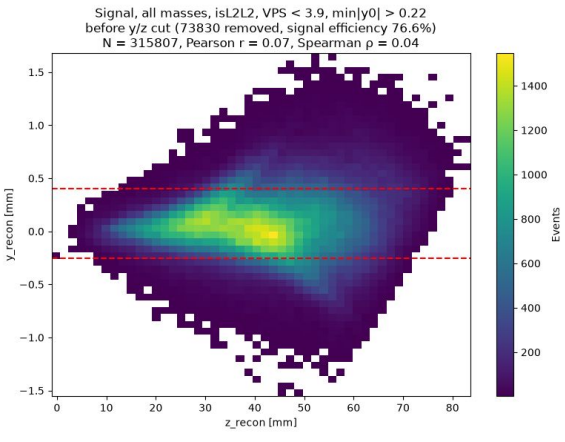


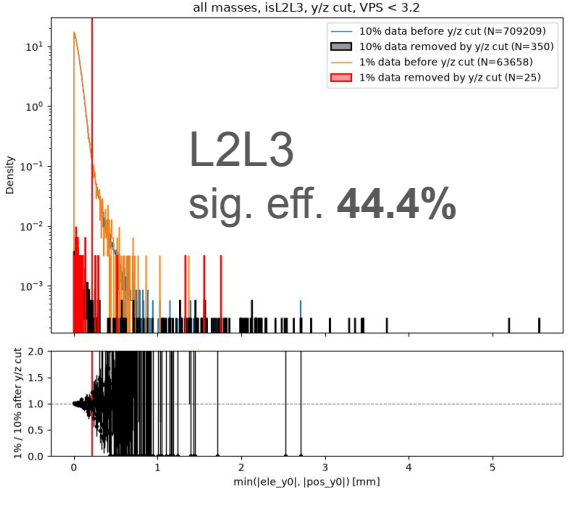


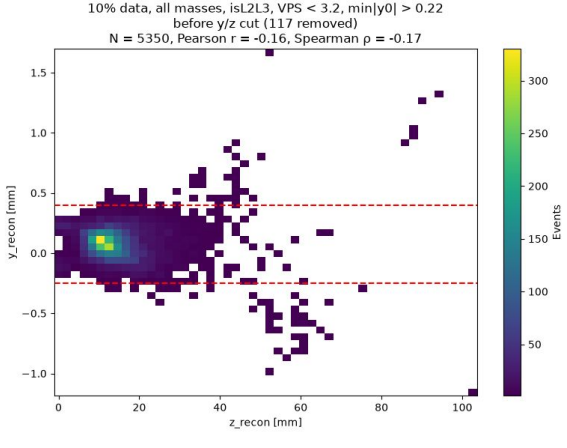
Tail Fitting: Vertex y selection

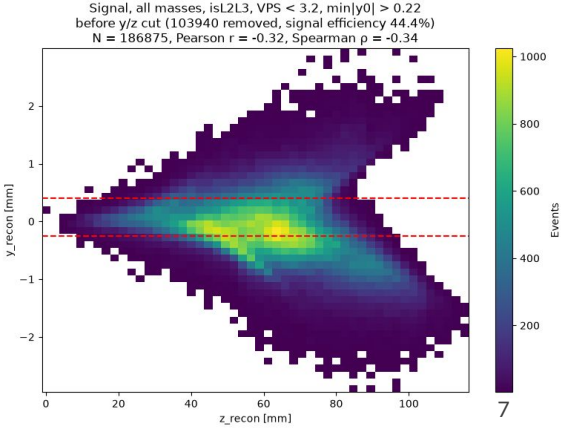




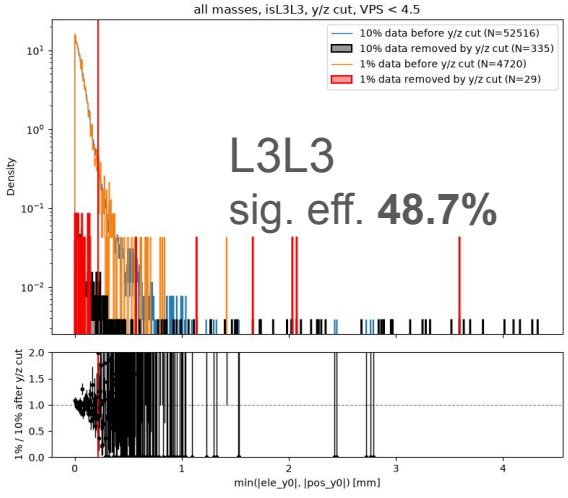


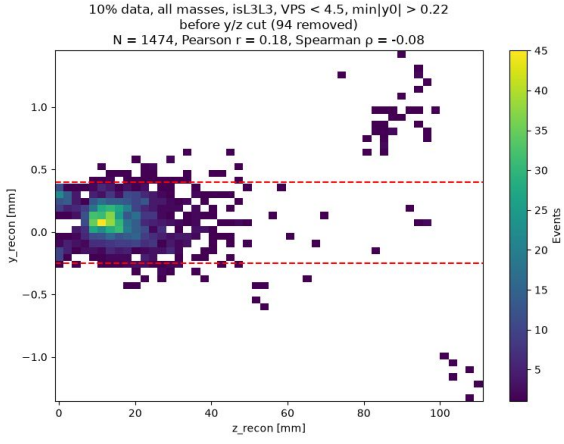


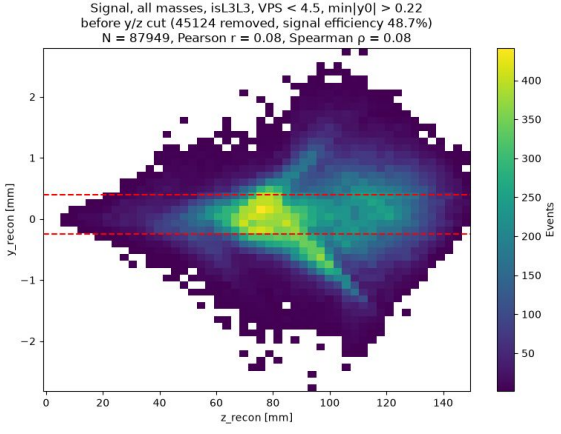




Tail Fitting: Vertex y selection



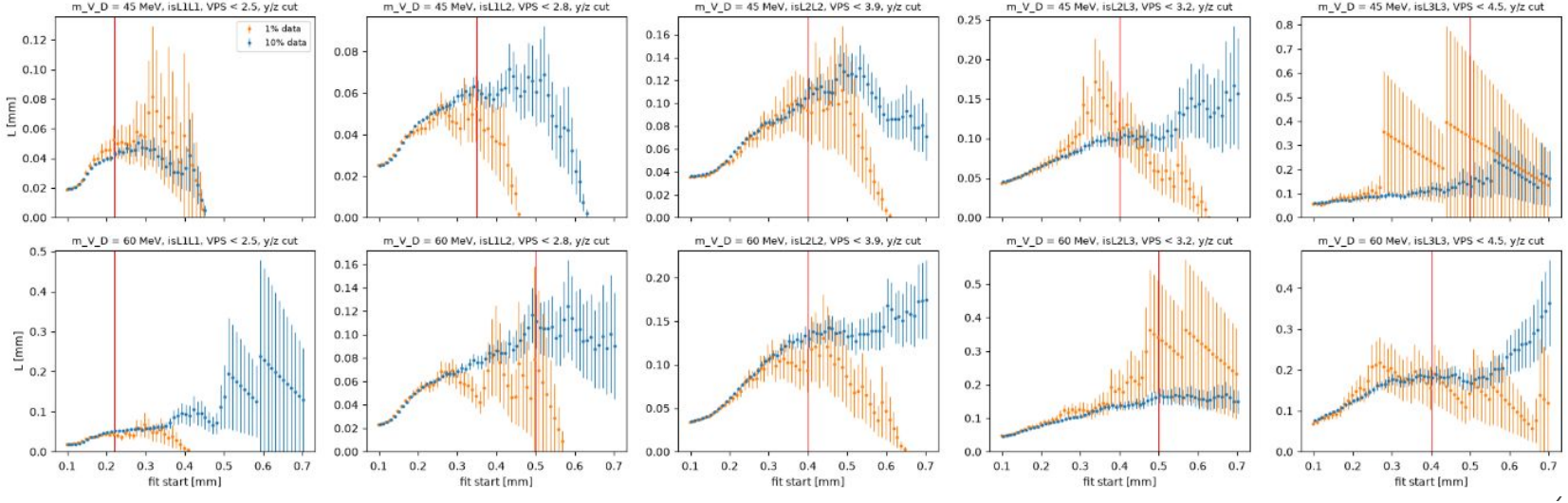




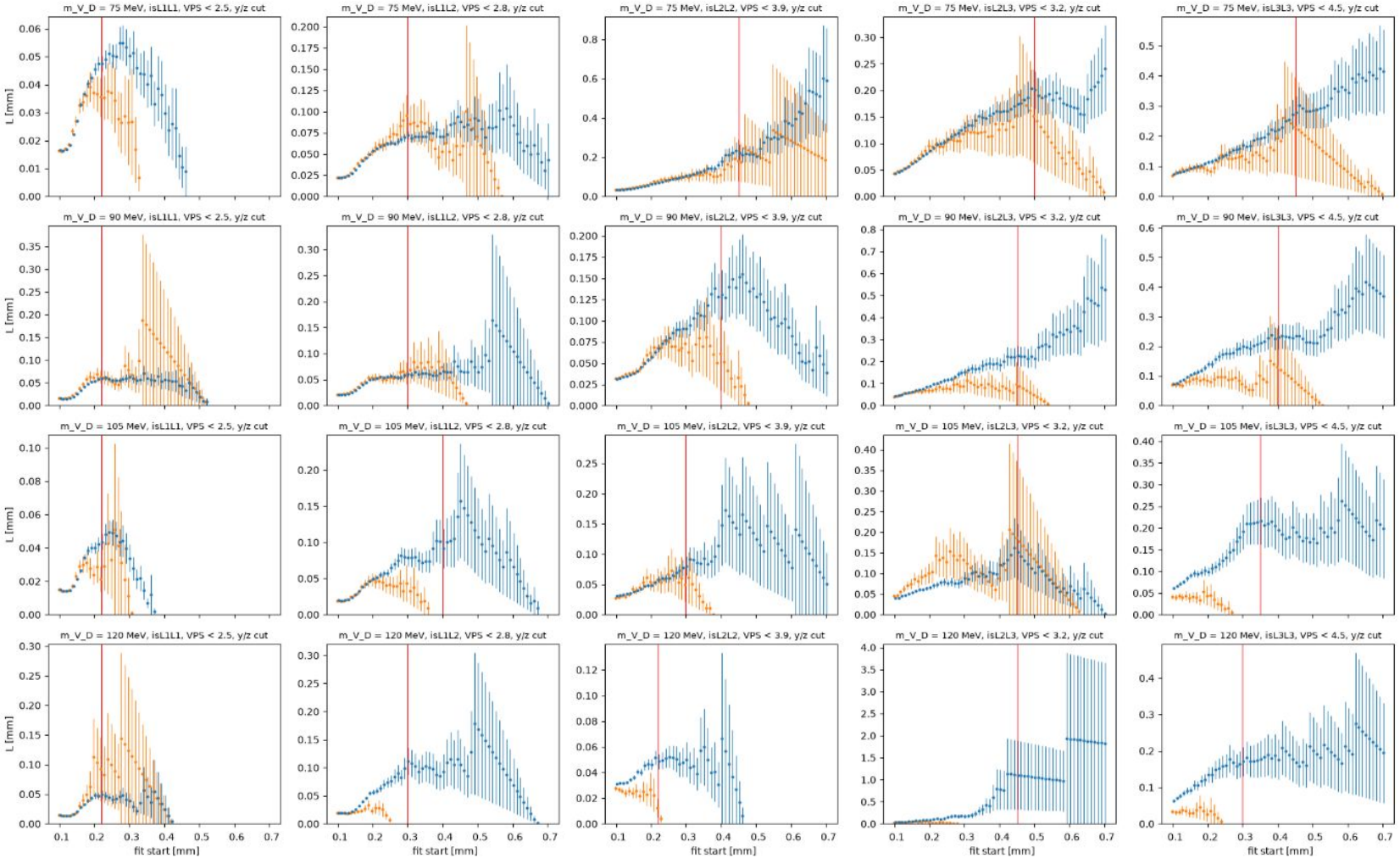
- These selections significantly clean up the tails, so we keep them for now
- Expect them to improve after calibration (better signal eff./ better tail removal)
- The Next step is to decide where to start fitting tails for each distribution

Tail Fitting: Tail starting points

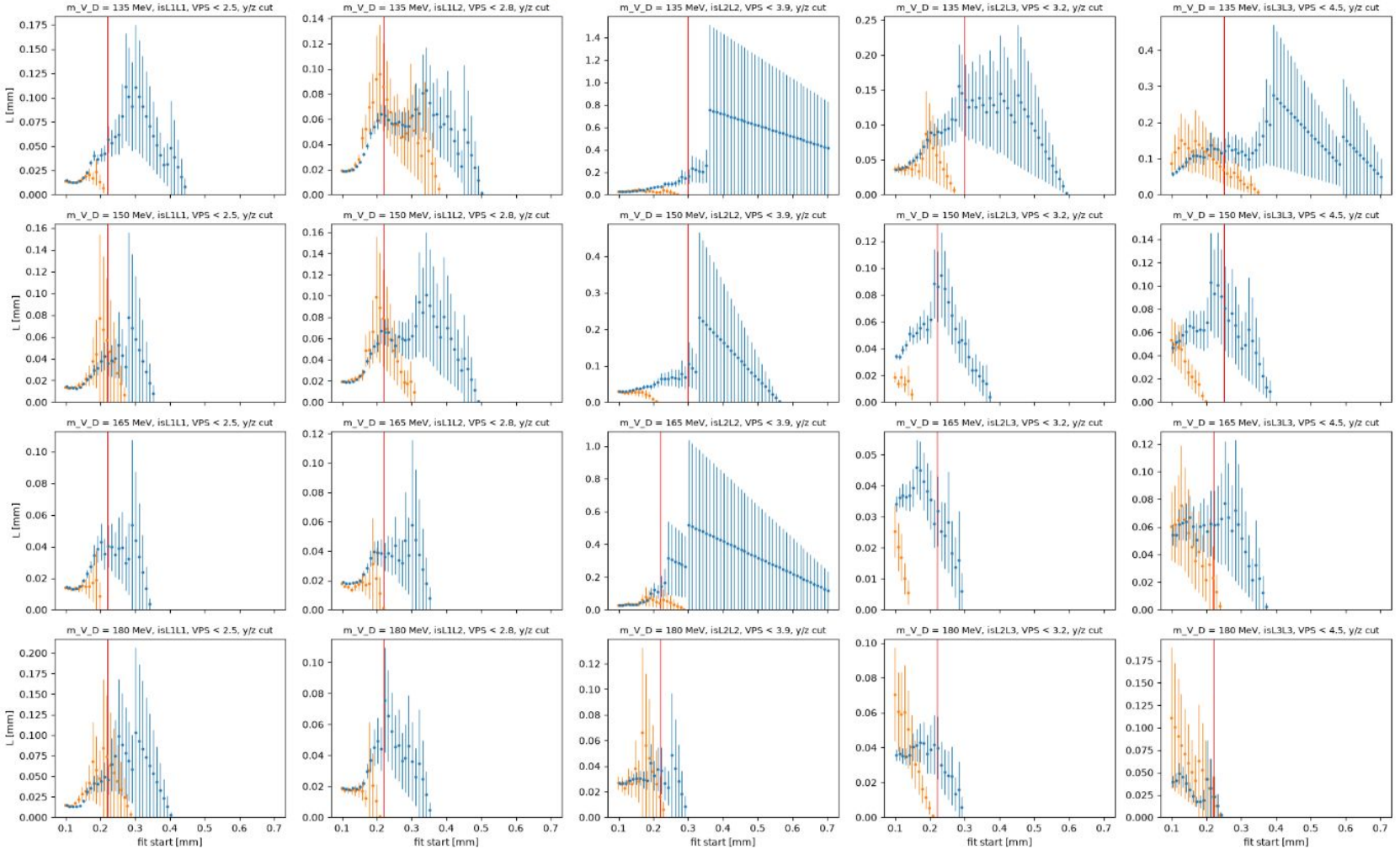
- A simple stability check is performed for each mass / hit category to determine suitable tail starting points
- We scan a range of fit start positions, determining L in each case. Once the fit start position is in the tail, L should stabilize
- Identify the plateau by eye and use that to determine the start position



Tail Fitting: Tail starting points



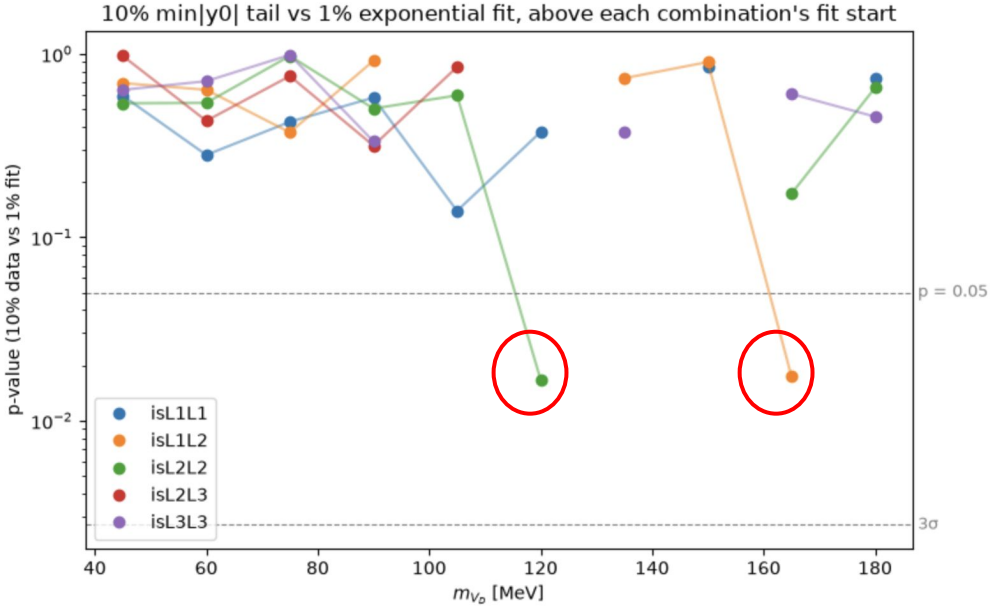
Tail Fitting: Tail starting points



Tail Fitting 1->10% performance

How well does the 1% tail model the 10% data?

- For each case, the 1% sample is fit to an exponential, and the poisson deviance is measured from the 10% data to the 1% fit
- The 10% data is fit to an exponential, the fit is used to draw 1% sample and 10% sample
- The 1% toy is fit to an exponential and the poisson deviance is measured from the 10% toy to the 1% toy fit
- The distribution over 20k toys is used to evaluate the p-value of our observation



$$D = 2\sum_{i=1}^{n}\left(y_i \log\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i)\right)$$

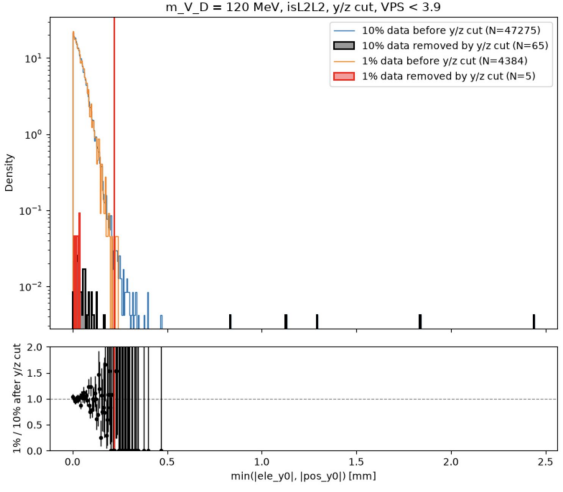
Tail fitting: Bad examples

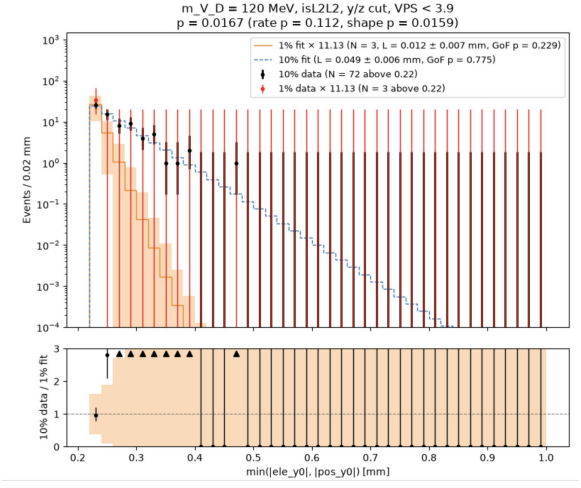
L2L2
m_{VD} = 120 MeV

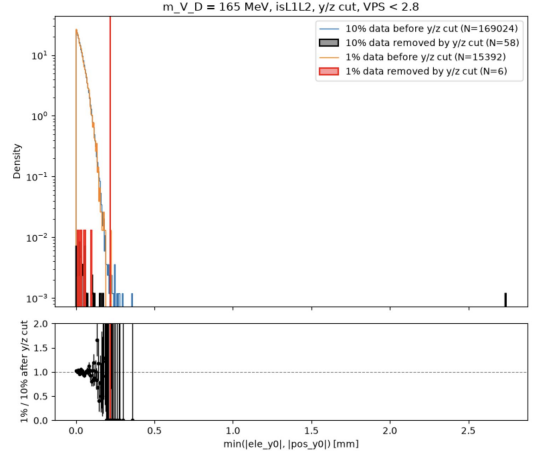
Notes:

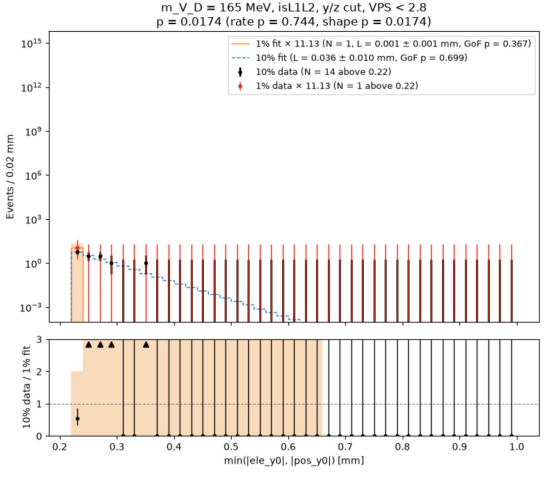
- *The vertex y cut helps*
- *The 1% dataset only has 1 early tail event*

L1L2
m_{VD} = 165 MeV









Updated tight selection optimization

- The tight selection optimization procedure is repeated as before, except background is estimated via the fitted tails
- Two method estimate significance
  - Asymptotic (doesn't account for tail fit uncertainty)
  - Toys (account for tail fit uncertainty)
- Updates to best selection identification procedure
  - previously the Z=1.28 contour is drawn and the mean epsilon of that contour quantifies how good a selection is
  - Now we have a epsilon = 0.001 floor before taking the mean

Updated tight selection optimization

Optimal tight selection configurations

1% Asymptotic

hitcat	combo_idx	VPS	miny0_low	miny0_diff
isL1L1	7816	4.8	0.46	0.20
isL1L2	11669	6.2	0.50	0.35
isL2L2	5888	4.0	0.82	0.40
isL2L3	3273	3.0	0.98	0.55
isL3L3	10373	5.6	0.98	0.65

10% Asymptotic

hitcat	combo_idx	VPS	miny0_low	miny0_diff
isL1L1	7829	4.8	0.48	0.15
isL1L2	3562	3.2	0.62	0.30
isL2L2	3120	3.0	0.76	0.60
isL2L3	1089	2.2	0.98	0.55
isL3L3	3821	3.2	0.98	0.65

1% Toys

hitcat	combo_idx	VPS	miny0_low	miny0_diff
isL1L1	10699	5.8	0.68	0.15
isL1L2	11794	6.2	0.68	0.30
isL2L2	6545	4.2	0.98	0.35
isL2L3	13635	6.8	0.96	0.65

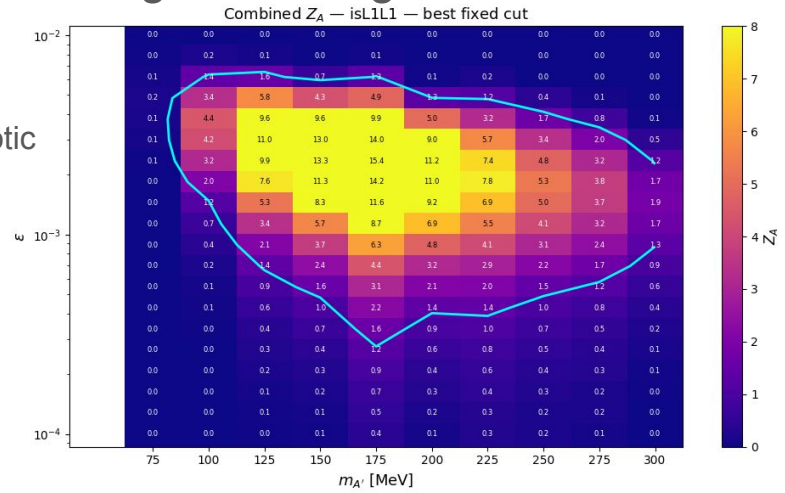
10% Toys

hitcat	combo_idx	VPS	miny0_low	miny0_diff
isL1L1	7843	4.8	0.50	0.15
isL1L2	7934	4.8	0.62	0.50
isL2L2	5987	4.0	0.96	0.45
isL2L3	7628	4.6	0.96	0.60
isL3L3	2729	2.8	0.98	0.65

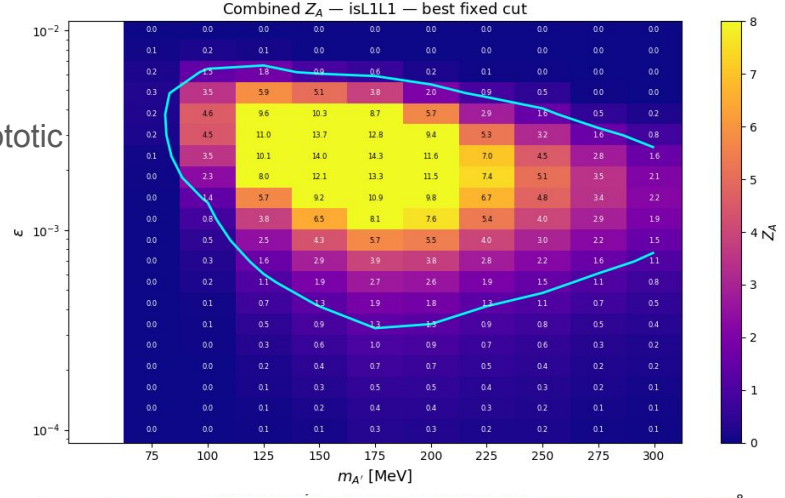
Updated tight selection optimization

L1L1 significance grids

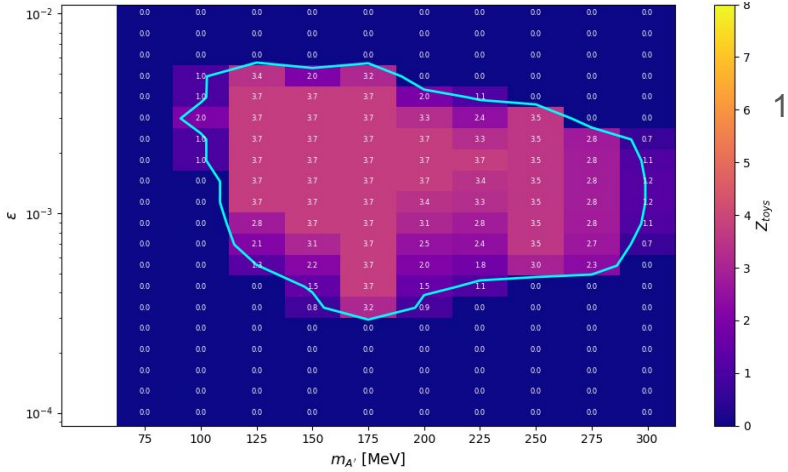
1% asymptotic



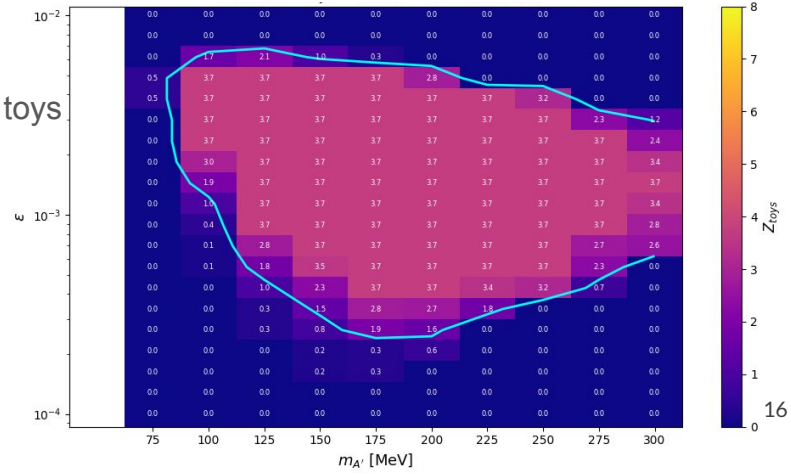
10% asymptotic



1% toys



10% toys

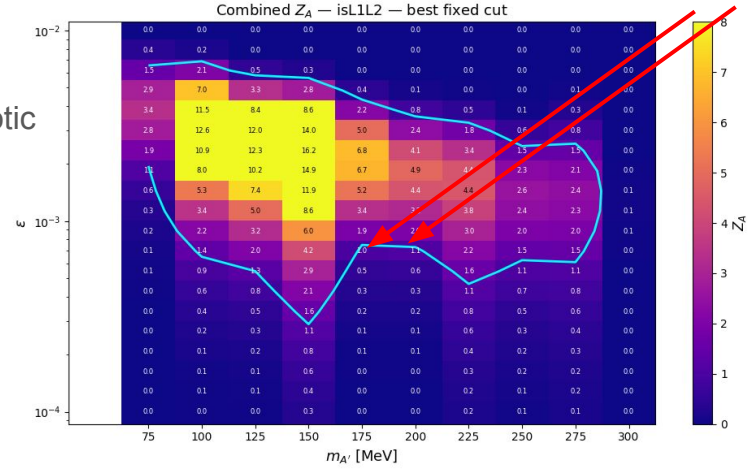


Updated tight selection optimization

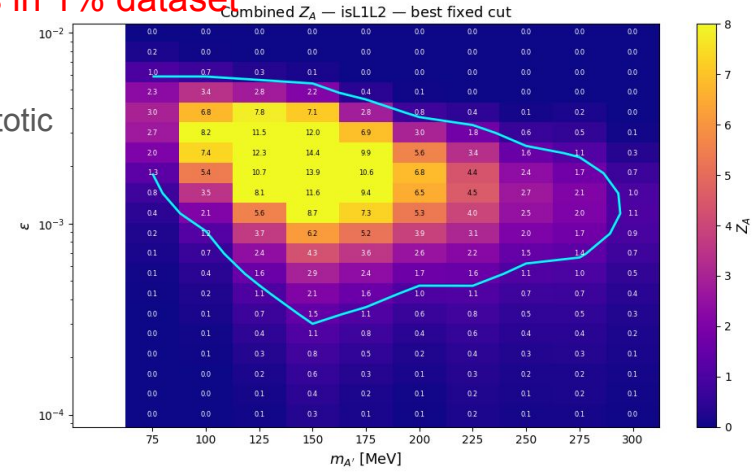
L1L2 significance grids

No tails in 1% dataset

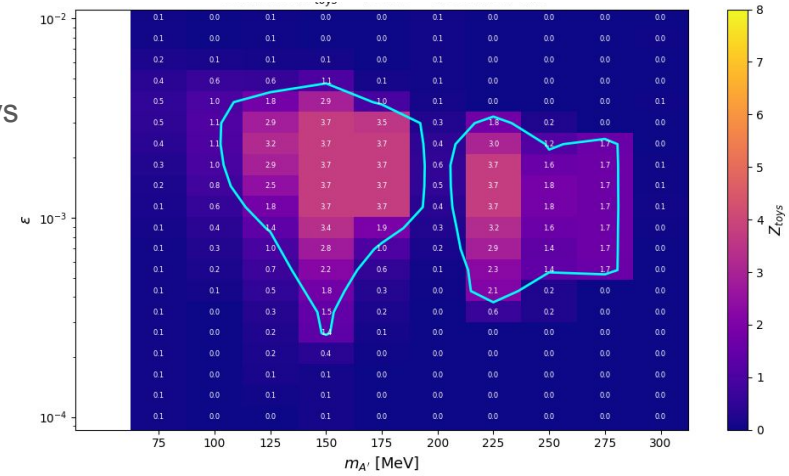
1% asymptotic



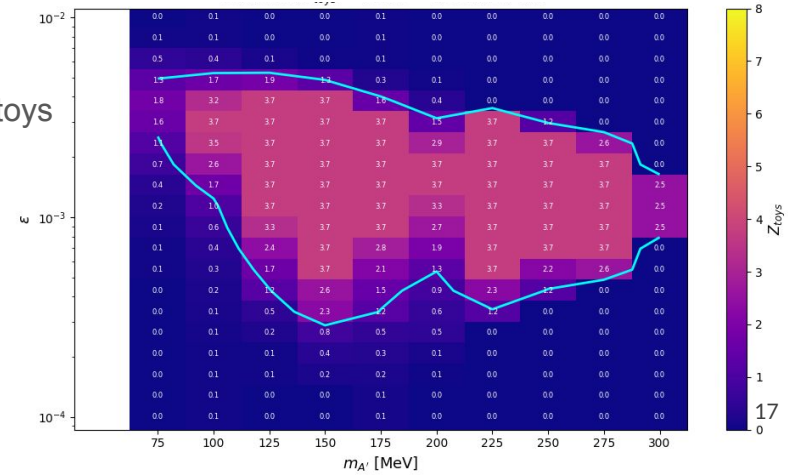
10% asymptotic



1% toys



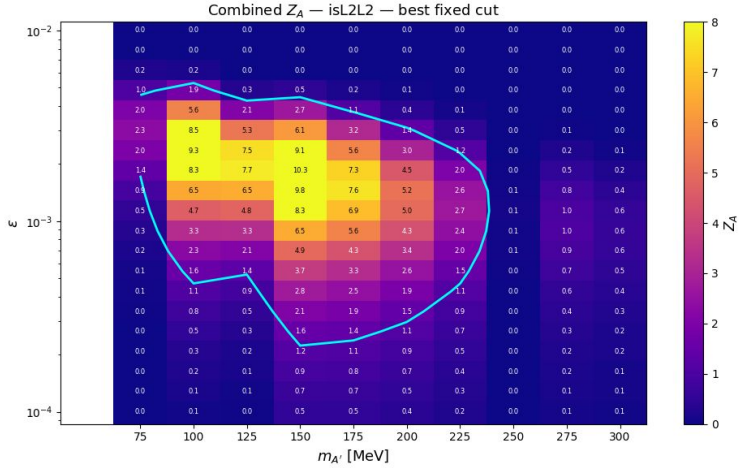
10% toys



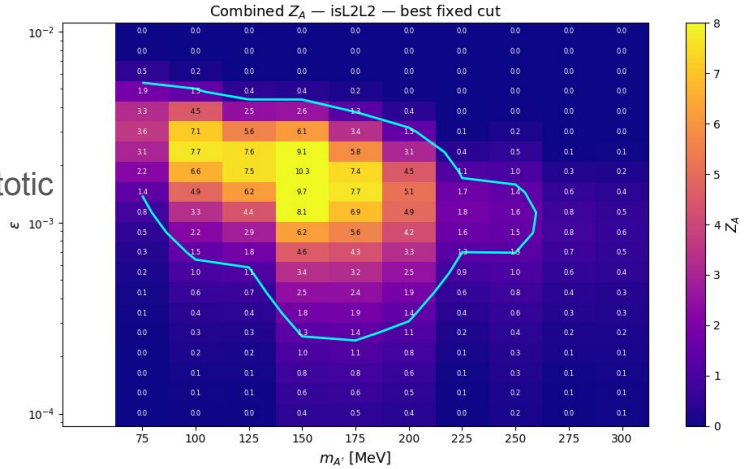
Updated tight selection optimization

L2L2 significance grids

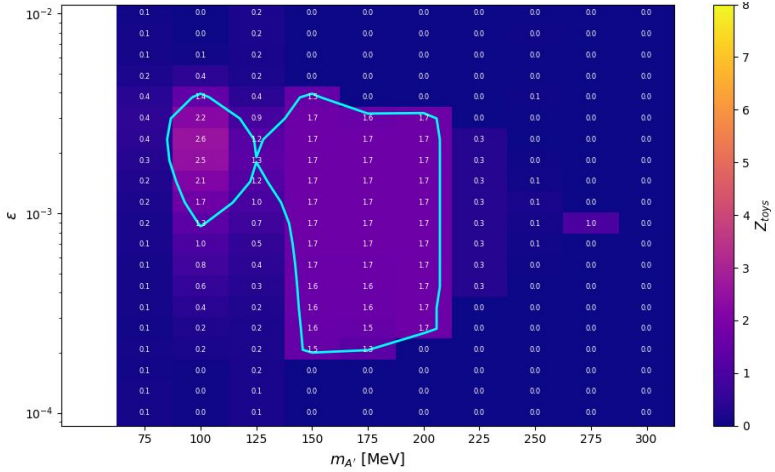
1% asymptotic



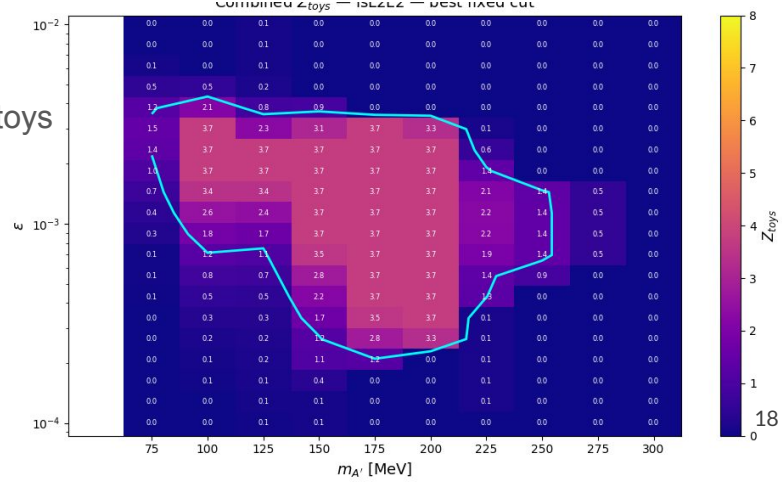
10% asymptotic



1% toys



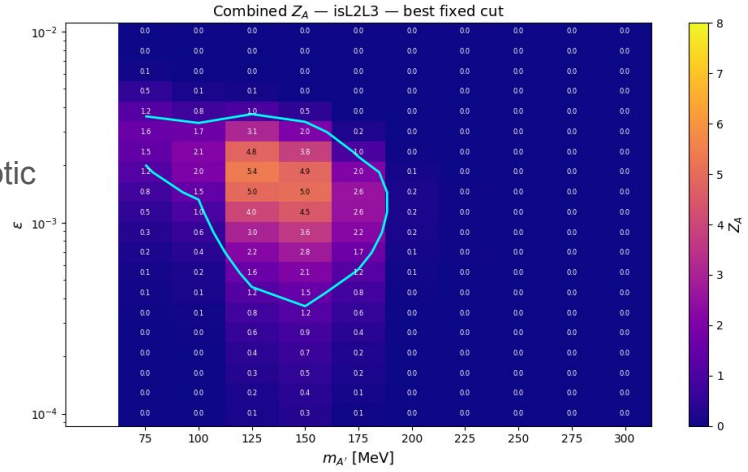
10% toys



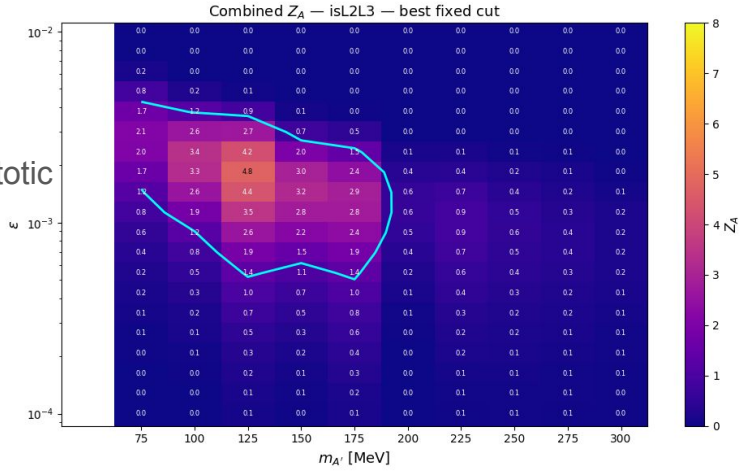
Updated tight selection optimization

L2L3 significance grids

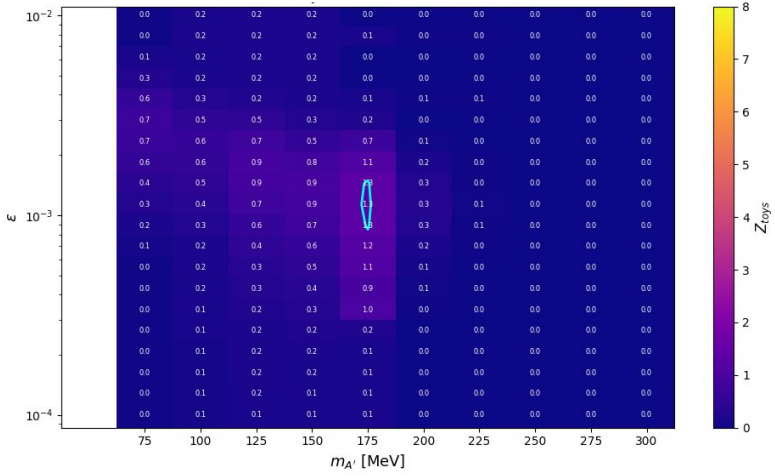
1% asymptotic



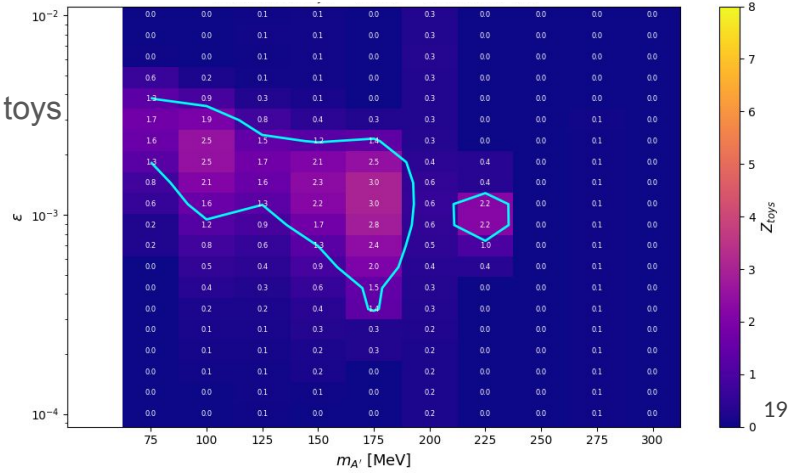
10% asymptotic



1% toys



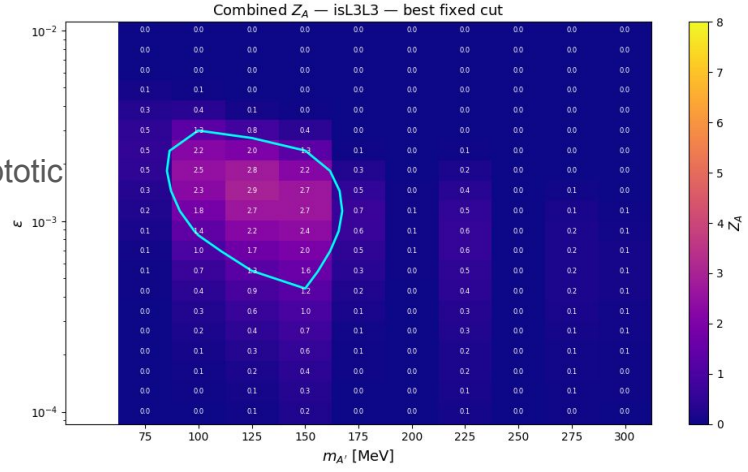
10% toys



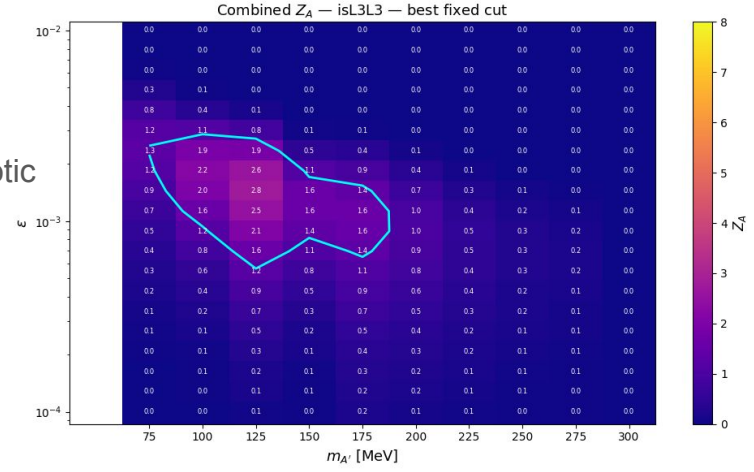
Updated tight selection optimization

L3L3 significance grids

1% asymptotic

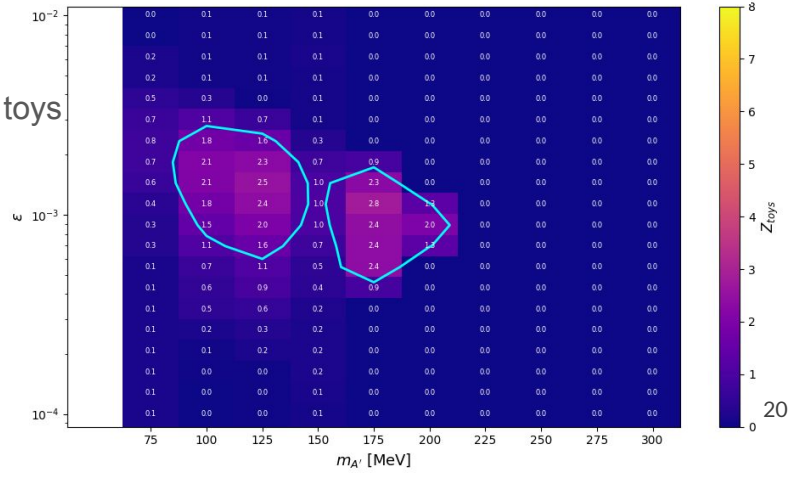


10% asymptotic



1% toys

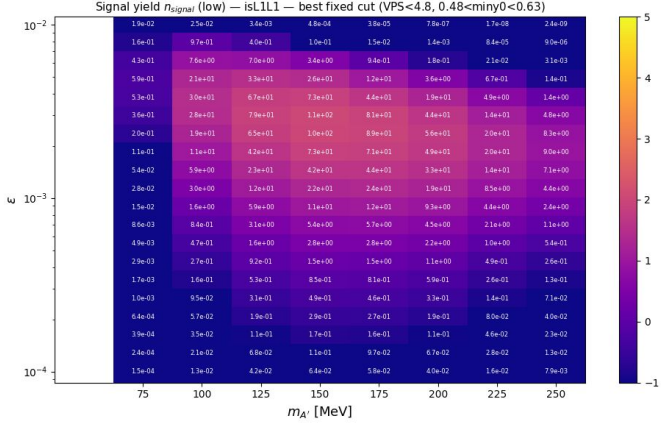
10% toys

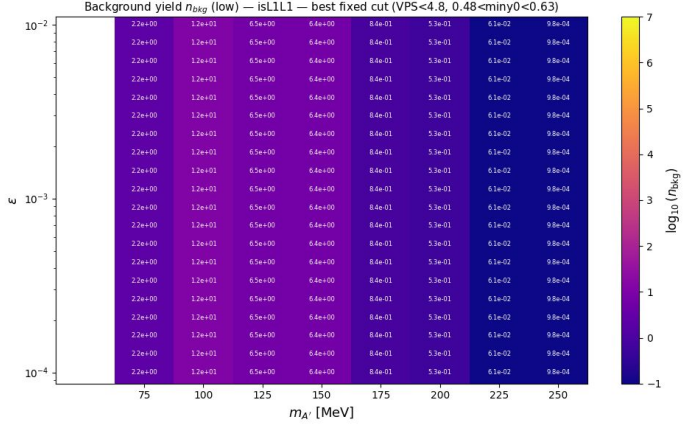


Updated tight selection optimization

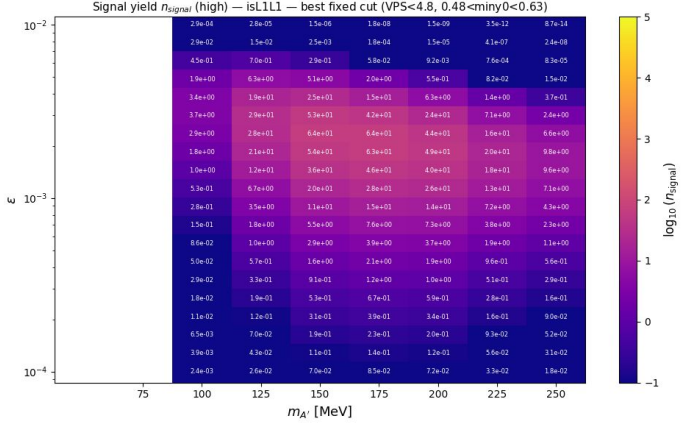
10 percent dataset, Asymptotic approach, L1L1 signal and background counts

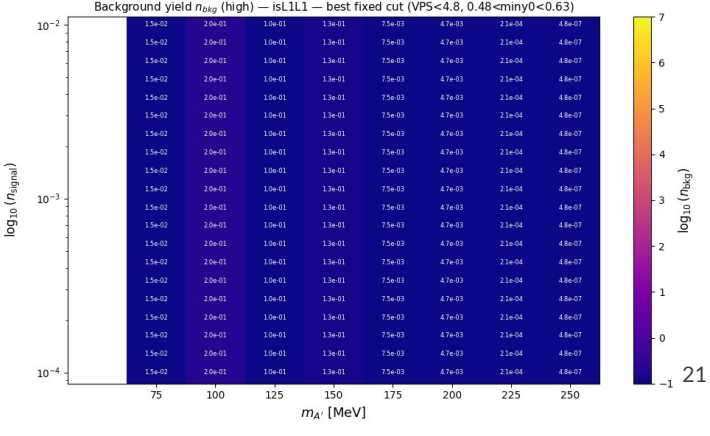
Low signal region





High signal region

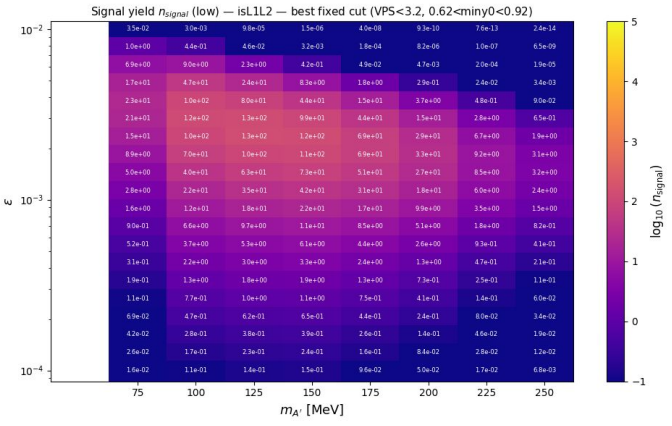


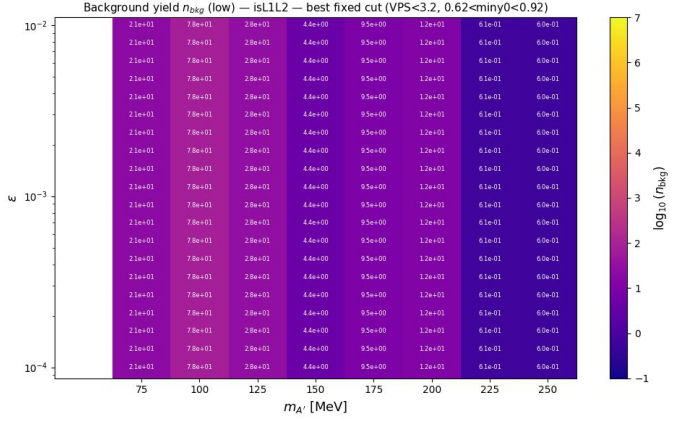


Updated tight selection optimization

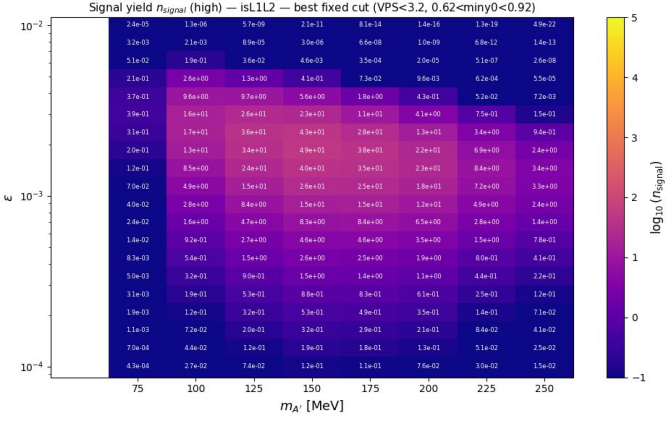
10 percent dataset, Asymptotic approach, L1L2 signal and background counts

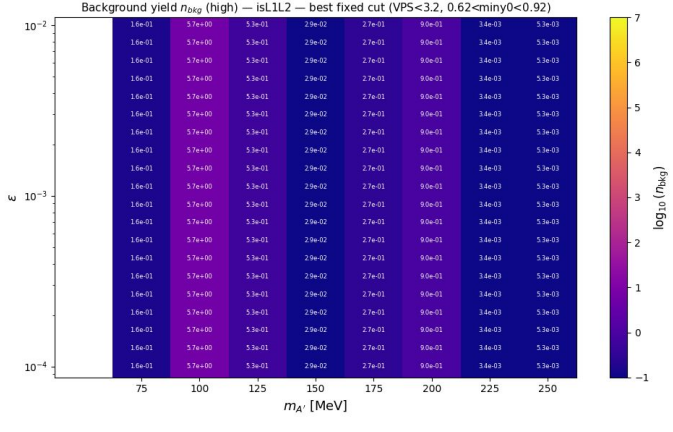
Low signal region





High signal region

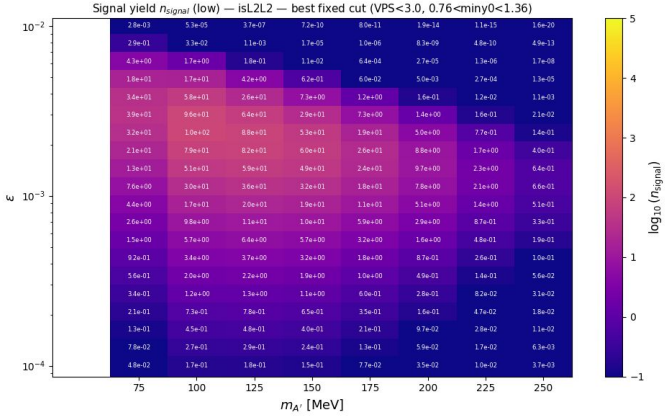


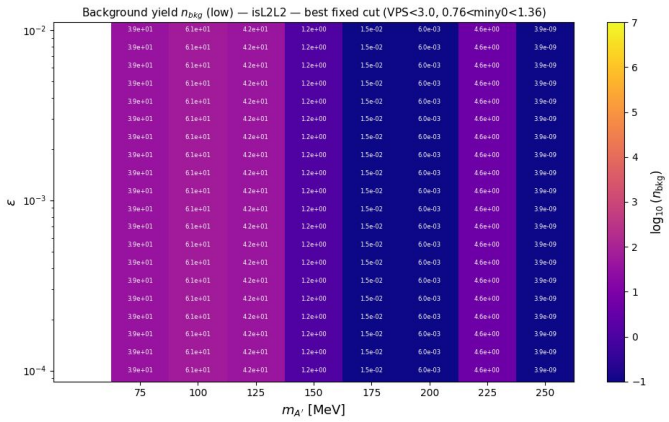


Updated tight selection optimization

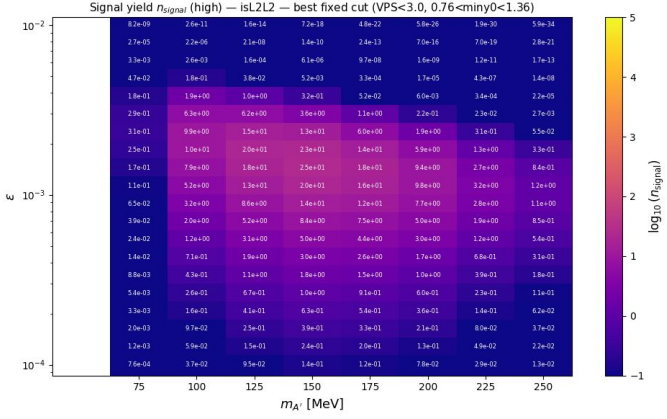
10 percent dataset, Asymptotic approach, L2L2 signal and background counts

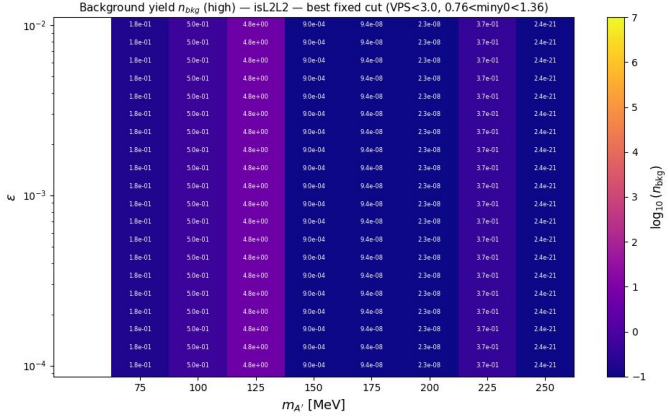
Low signal region





High signal region

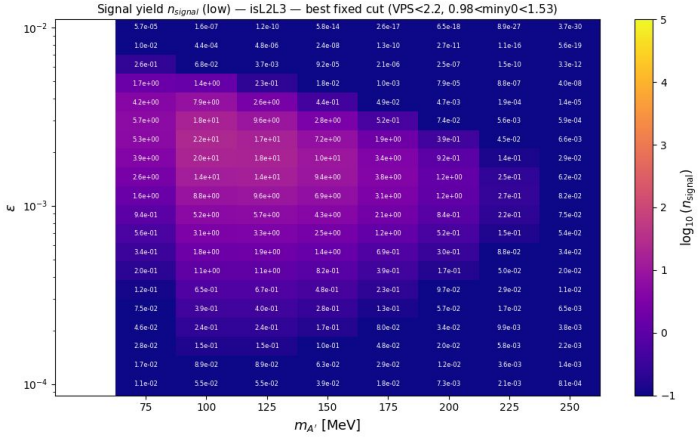


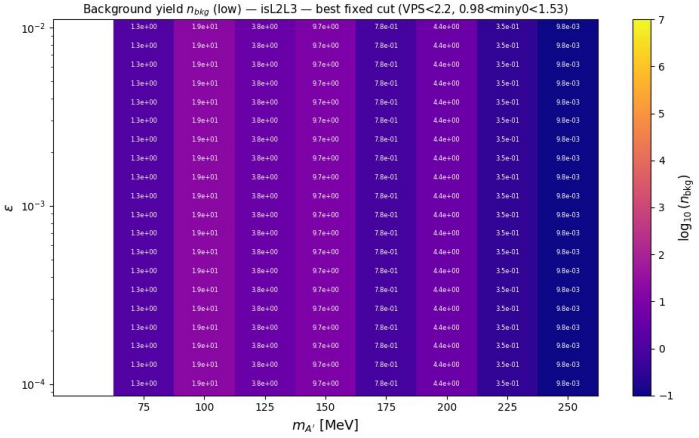


Updated tight selection optimization

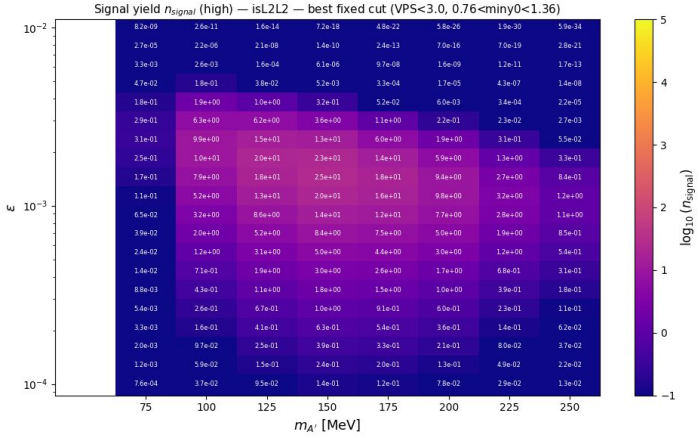
10 percent dataset, Asymptotic approach, L2L3 signal and background counts

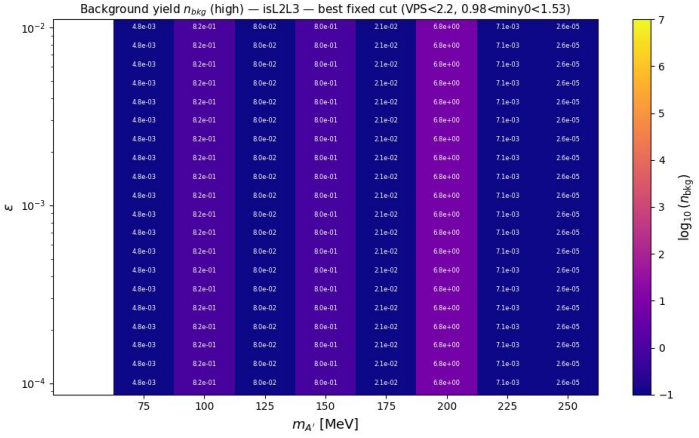
Low signal region





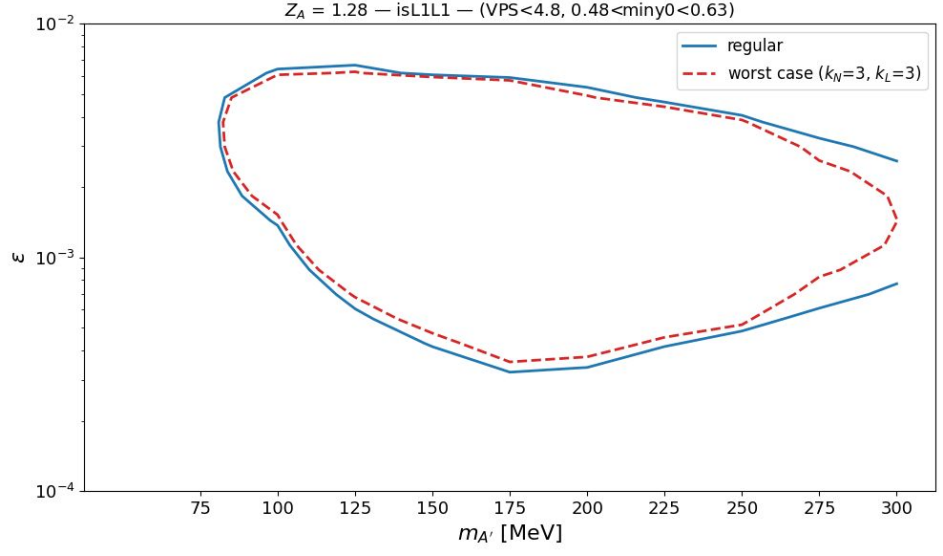
High signal region

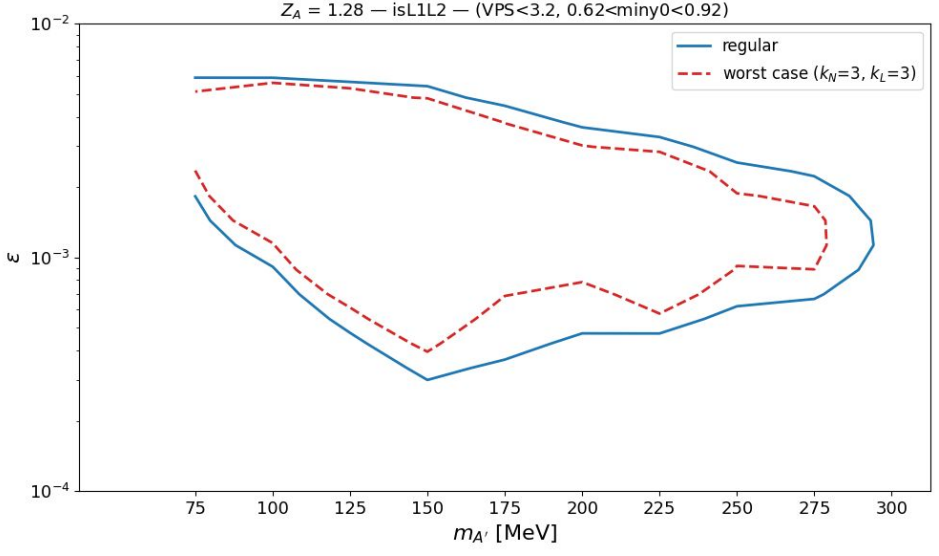




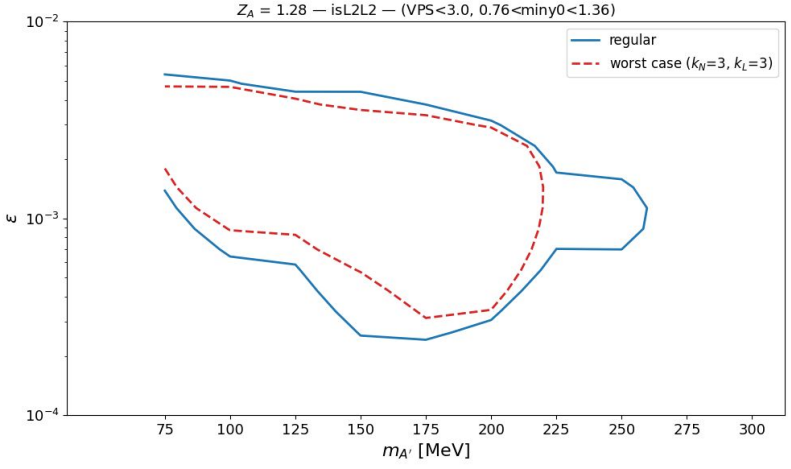
Updated tight selection optimization

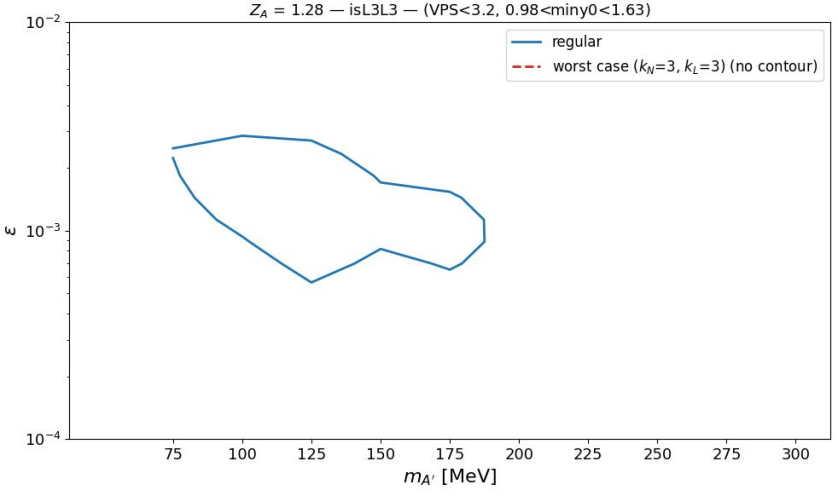
- Worst case scenario test:
  - Tight selections are optimized on the 10% dataset via asymptotic approach
  - The tight selections are fixed, then applied to the 10% dataset with pessimistic tails (3 sigma upper limits are taken on n_tail and L)
  - We assess how much the Z=1.28 contours are affected

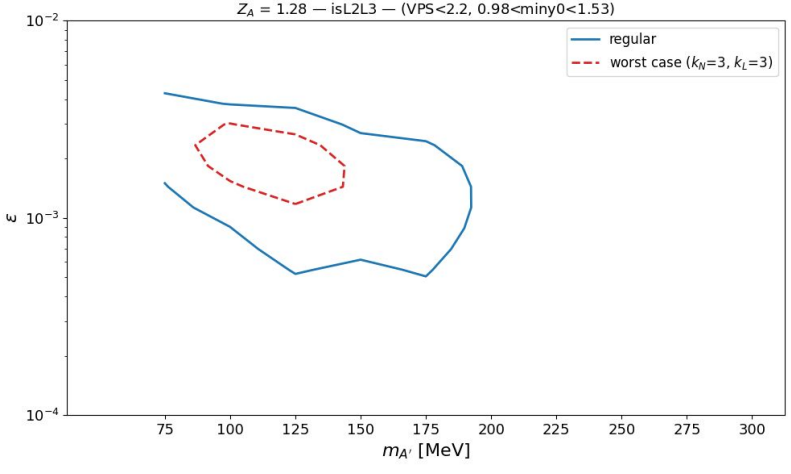




Updated tight selection optimization



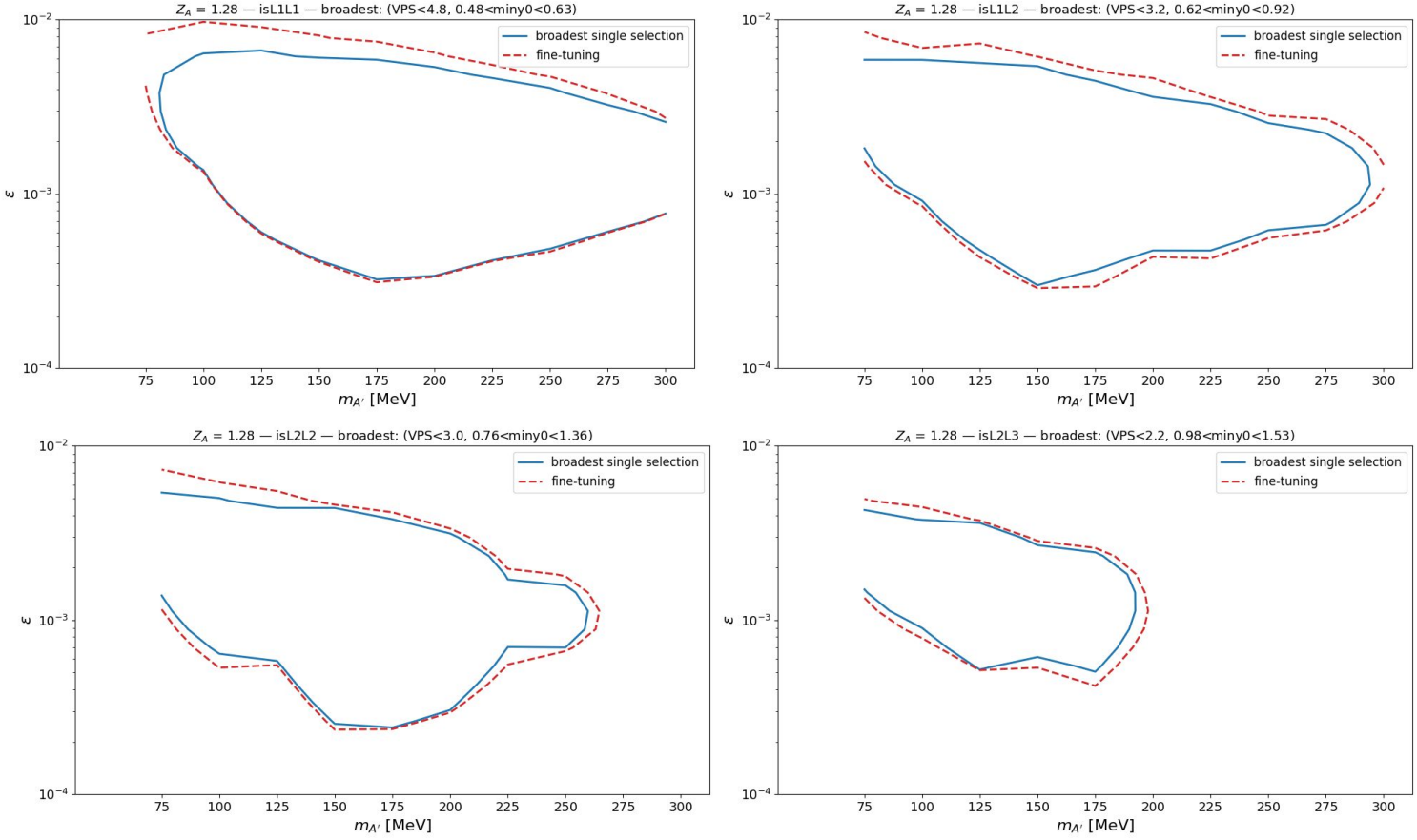




No Z=1.28 contour for L3L3 in the worst case scenario

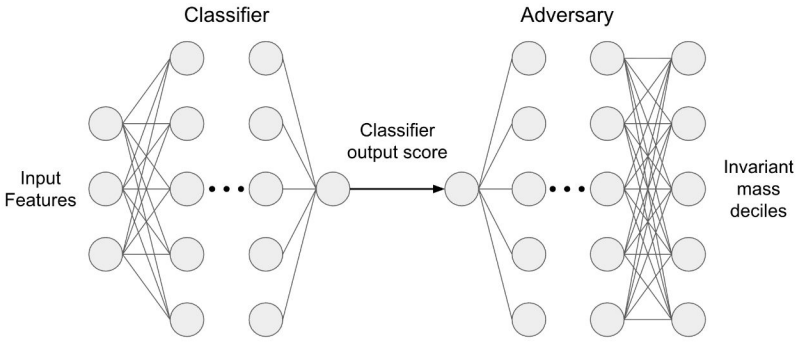
Updated tight selection optimization

- The tight selection optimization has changed
- Compare current method to fine-tuning method



ML Methods

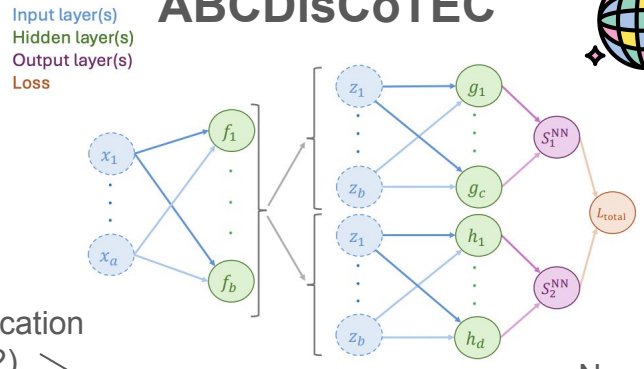
ANN



- ANN is used to provide a discriminant that is uncorrelated from mass
- This is a conservative ANN prescription
  - 5 Features: VPS, Psum, ele_y0, pos_y0, vertex_z
  - One ANN is trained for all masses / epsilons
  - 30% of 1% sample + tritrig / wab MC used as background
- ANN performance quantified via
  - MC/data agreement
  - Classification performance
  - Invariance w.r.t mass

ABCDisCoTEC





Classification (x2)

Non-closure

$$\mathcal{C} = \left|1 - \frac{N_B N_C}{N_A N_D}\right|$$

$$L_{total} = \lambda_{BCE}\left(L_{BCE}^{(1)} + L_{BCE}^{(2)}\right) + \lambda_{DisCo} L_{DisCo} + \lambda_{nonclosure} L_{nonclosure},$$

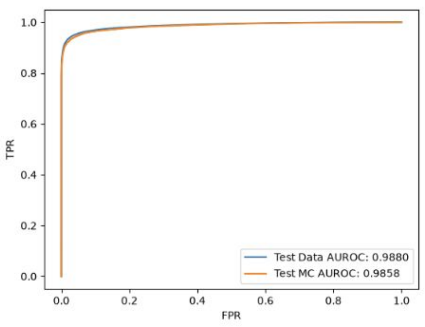
Decorrealation of discriminants

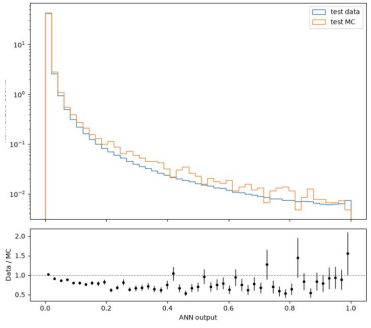
- Dual headed network that procides two uncorrealeated discriminants, suitable for ABCD
- The modified differential method of multipliers

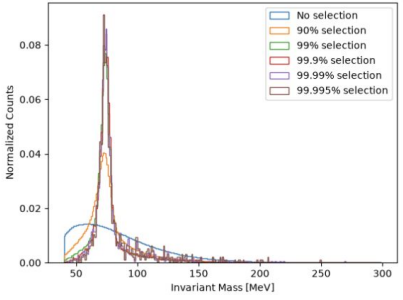
ML Methods

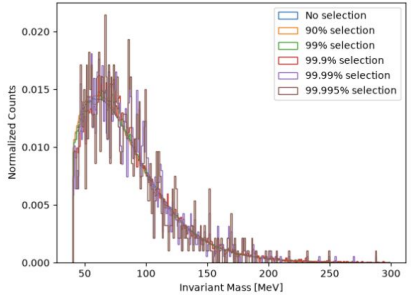
Recap of ANN results:

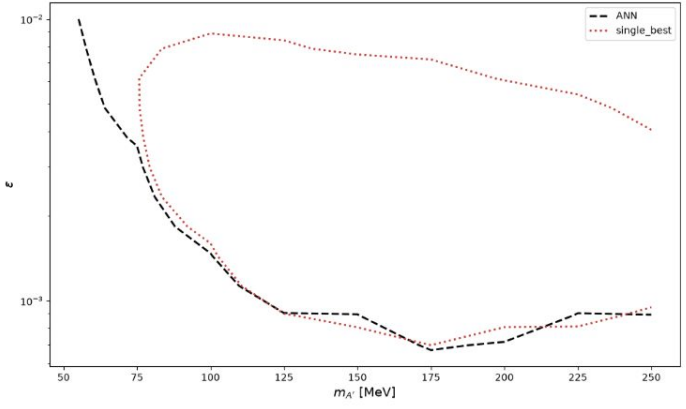
- Good MC/Data agreement
- Good mass deccoreleation
- Comparible performance to square cuts
- Slight improvement possible by training on individual mass / epsilon

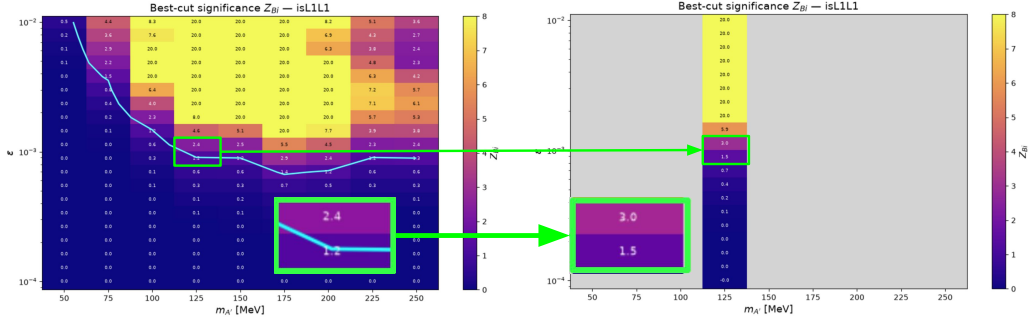






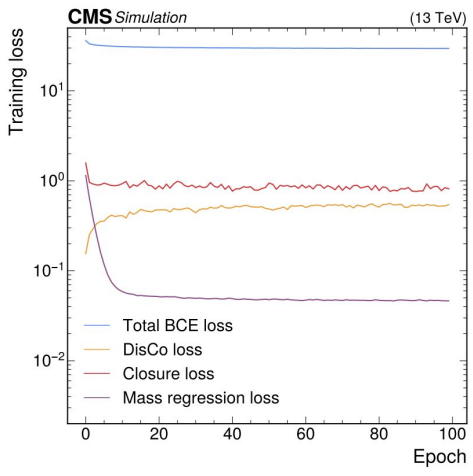


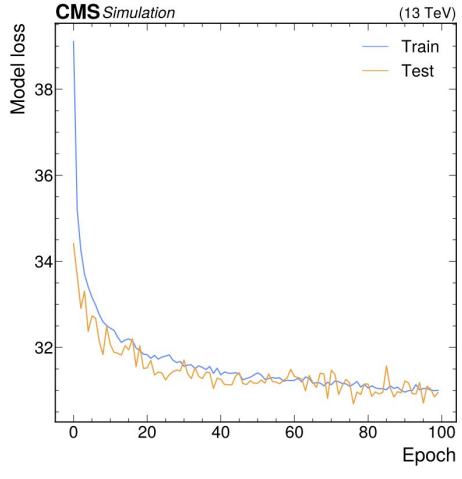




ML Methods

ABCDisCoTEC losses vs epoch in the paper

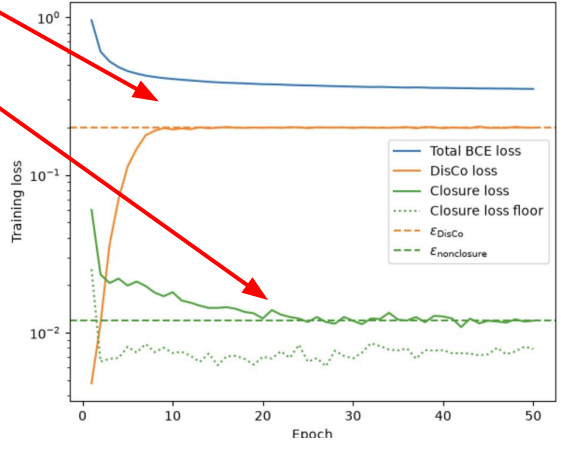


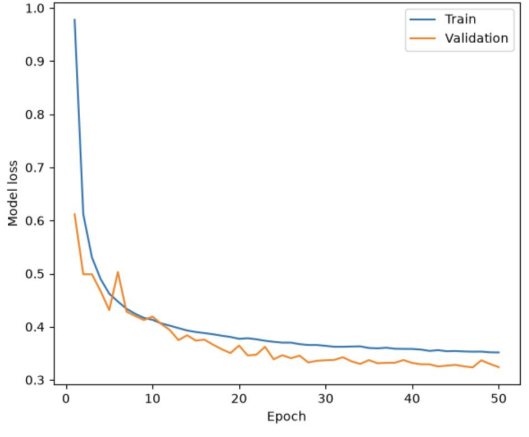


ABCDisCoTEC losses vs epoch in our implementation

- Should reduce DisCo target
- Should reduce closure target

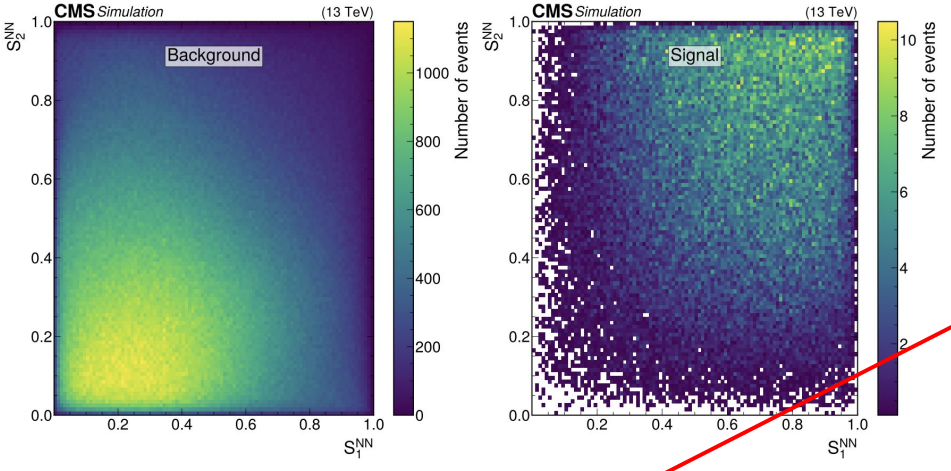
This implementation is still a work in progress





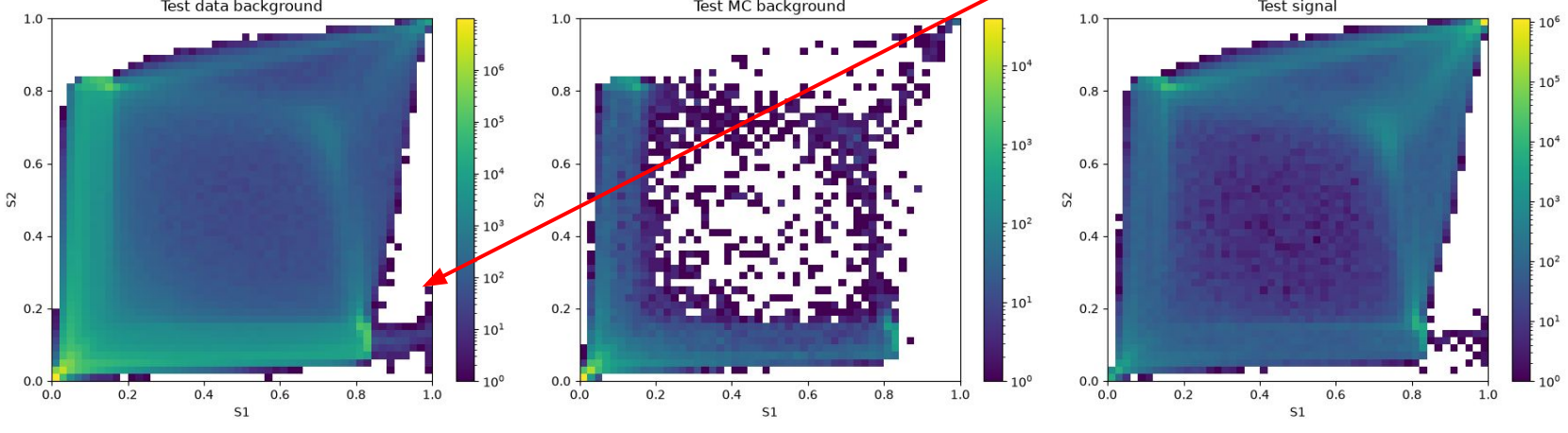
ML Methods

Signal / background distributions on test datasets



This implementation is still a work in progress

b1 and b2 are drawn independently, each uniform between 0.2 and 0.8 on the score scale

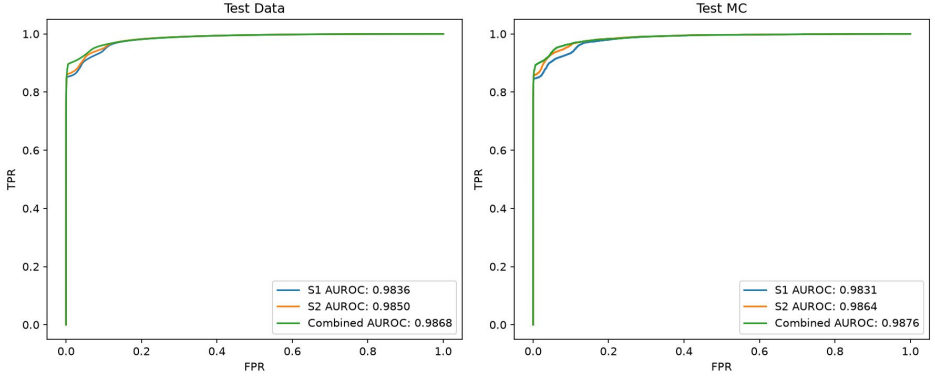


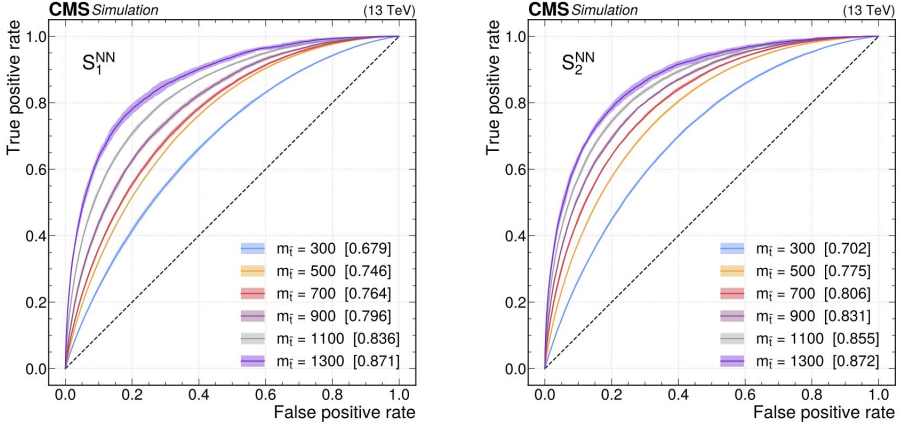
ML Methods

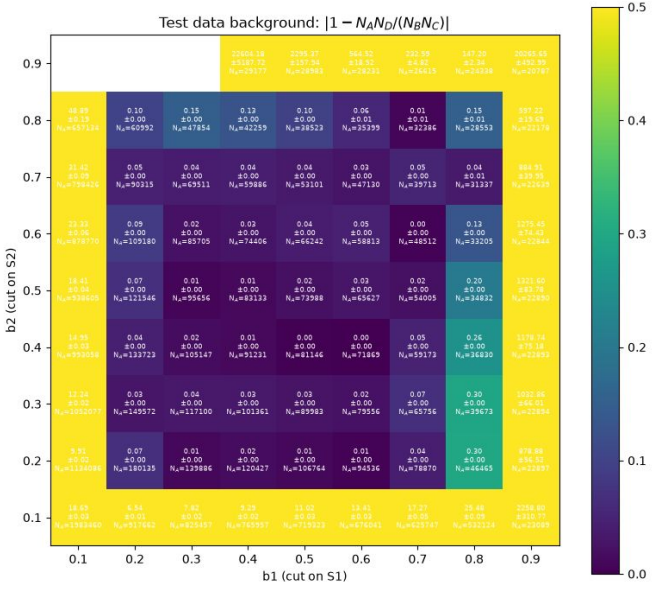
Classification performance and nonclosure

- nonclosure only good in (0.2,0.80) range
- ABCDisCoTEC paper discuusses a more difficult classification problem

This implementation is still a work in progress



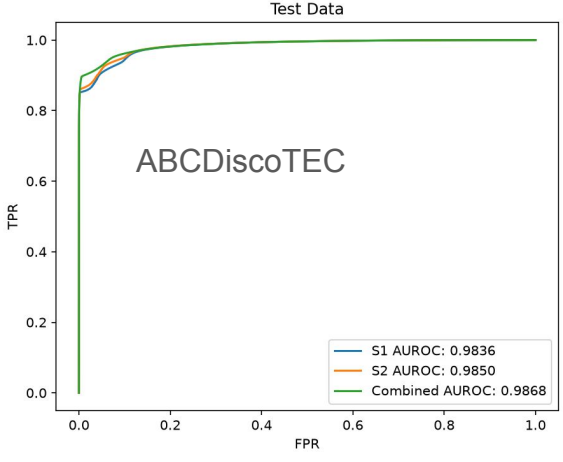


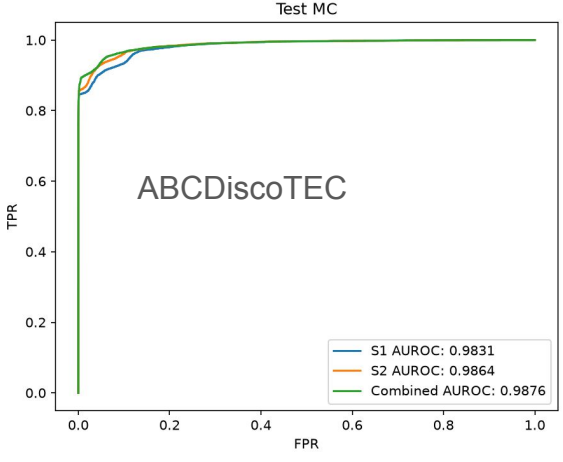


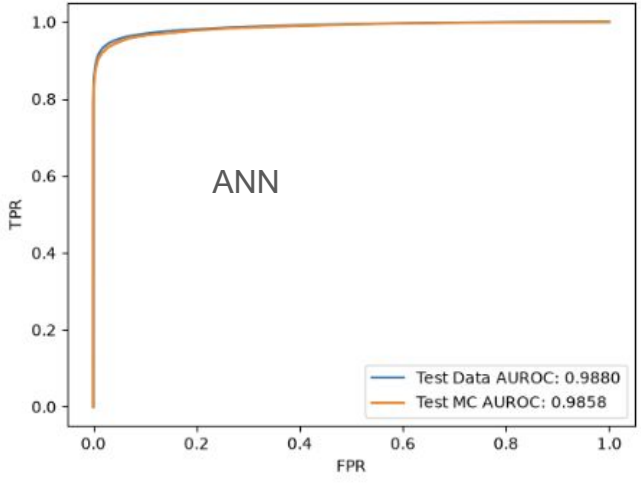
ML Methods

Comparing classification performance with ANN

This implementation is still a work in progress







Early implementation already has comparable performance to ANN!

Decisions to make / next steps

- Get extended fits for f_{rad} and A_{prompt} (may require more mc)
- Review SIMP generation
- Ensure consistent data / SIMP processing
- Unblind 10% dataset for ML
  - study tail fitting method for ML methods
  - ABCDiscoTEC complicates tail fitting
- Test limit setting pipeline with updated preselections
- SIMP models parameters (affect tight selection parameters)