Semantic Segmentation with Sparse Convolutional Neural Network for Event Reconstruction in MicroBooNE

MicroBooNE-Note-1091-Pub

MicroBooNE Collaboration

June 2020

E-mail: microboone_info@fnal.gov

Abstract

We present the performance of a semantic segmentation network, *SparseSSNet* that provides pixel-level classification of MicroBooNE data. MicroBooNE is a short baseline neutrino oscillation experiment employing a liquid argon time projection chamber detector. *SparseSSNet* is a submanifold sparse convolutional neural network and the first machine learning based algorithm applied to one of MicroBooNE's signature ν_e appearance oscillation analyses. The network is trained to segment five types of classes relevant to this analysis; however those are re-classified into two track and shower. The improvement of this network with respect to previous algorithm is both in accuracy and computing resource utilization. The accuracy achieved on the test sample is $\geq 99\%$ out of all non-zero pixels; for full neutrino interaction simulations, the time for processing one image is $\sim 0.5\,\text{sec}$ and the memory is O(GB), which allows utilizing CPU worker machines at FNAL and the Open Science Grid, without adaptations.

1 Introduction

The main goal of the MicroBooNE experiment is to search for Low Energy Excess (LEE) electron like events, specifically in the region seen by the MiniBooNE experiment [1]. As the dominant cross section in the observed LEE energy range is Charge Current Quasi-Elastic (CCQE), the approach adopted

by the Deep Learning (DL) LEE analysis is to isolate highly pure data samples of CCQE ν_e (ν_μ) interactions. The topology of these interactions are fairly simple and manifest in most case as 1 lepton and 1 proton; where the lepton is either an electron or a muon for ν_e or ν_μ respectively. The outgoing proton generates a short track and is common to both interactions; however the outgoing lepton produces a shower or track depending upon whether it is an electron or a muon. Therefore classifying outgoing particles whether they produce a shower or a track is crucial for this analysis. Convolutional Neural Networks (CNN) are the state of the art algorithm for solving many problems among the task of semantic segmentation [2]. In this note, we describe a deep learning based algorithm to distinguish showers from tracks in data from the MicroBooNE Liquid Argon Time Projection Chamber (LArTPC).

The MicroBooNE detector [3] operating at the Booster Neutrino Beam (BNB) at Fermilab, consist of a $\sim$ 170 ton Liquid Argon Time Projection Chamber (LArTPC). The charge readout is done by three wire planes; two induction planes (U and V) and one collection plane (Y). In the DL-LEE analysis the data is represented as a set of 2-dimensional images (one for each wire plane), with wire number along the x-axis and drift time along the y-axis. The intensity of each "pixel" is given by the sum of the noise-filtered, deconvolved signal from six TPC time-ticks (3μs) [4].

After applying a set of initial data selection criteria for reducing low energy backgrounds and tracks originating from cosmic muons, the images are fed into a CNN in order to semantic segment them, i.e., dividing the "pixels" into various classes. The output of the network is a normalized probability vector (also referred as scores) for each class ($\vec{p}$), the predicted label is then defined to be the class with the highest probability.

Within the current DL-LEE analysis there are only two classes track and shower; however the network is able to distinguish five different classes Highly Ionizing Particles (HIP); Minimum Ionizing Particle (MIP); shower; delta; and Michel which are re-classified into two classes where track consist of HIP and MIP and shower consist of shower, delta, and Michel.

2 The Network

The current network, "*SparseSSNet*" (Sparse Semantic Segmentation Network) is a modification over "SSNet" [5], in which the image data is processed as a sparse matrix as opposed to a dense matrix [6]. Using *SpaseSSNet* provides several benefits over using the dense SSNet. First it allows for training on smaller images than the inferred ones; therefore, although the training images are only 512 X 512 pixels, the inferred images can be much larger, for

instance 1008 X 3456 pixels. Thus, unlike in the dense representation where the inferred image is too large to be processed on a standard GPU and had to be cropped into ~ 64 images, in the sparse representation, the entire image can be processed and no cropping is needed. In addition, the nature of the sparse representation as well as fewer images to process (no cropping) improves drastically the time for processing an event from $\sim 64 \times 5\,\mathrm{s}$ (for 64 crops) to $(\sim 0.5)\,\mathrm{s}$. Moreover as pixels of no interest (e.g., 0 intensity) are not saved, the memory consumption is reduced as well from $\sim 6\,\mathrm{GB/crop}$ to $\sim 1\,\mathrm{GB/image}$. Due to these two advantages the image inferring can be done utilizing worker machines at FNAL and the Open Science Grid, without "optimization" of the network.

The architecture of the network (U-Res-Net) is a hybrid of U-Net [7] and Res-Net [8]. The network is constructed with 32 filters in the initial layer and has 5 layers. A softmax[1] classifier and a cross entropy loss function summed over all non-zero pixels is used. In addition a weighting scheme is applied for preventing class imbalance (see section 4). A designated set of network weights is derived for each plane and no data is shared between the different planes.

Prior to training *SparseSSNet* masking is performed on the image which distinguishes between important and non important pixels. In this analysis all pixels with intensity < 10 or > 300 are not included in the sparse representation, which reduces the number of pixels to $\sim 0.5\%$ of the total pixels in an image. Once a pixel is not included in the sparse representation, it is disregarded and not stored; moreover pixels which do no pass the masking process cannot appear in hidden layers due to convolutional operations (see fig. 1). This prevents "dialation"of the image and improves the accuracy of the network.

3 The Data samples

We perform a supervised learning. The data samples consist of $\sim 143,000$ images with pixel intensity. Pixels with ionization associated with a certain particle are defined as a cluster. For each event particle propagation as well as detector effects are applied to derive the final input image.

[1]Softmax is a mathematical function that takes as input a vector of real numbers, and maps it into a probabilities summed to one, with larger input values corresponding to higher probabilities.

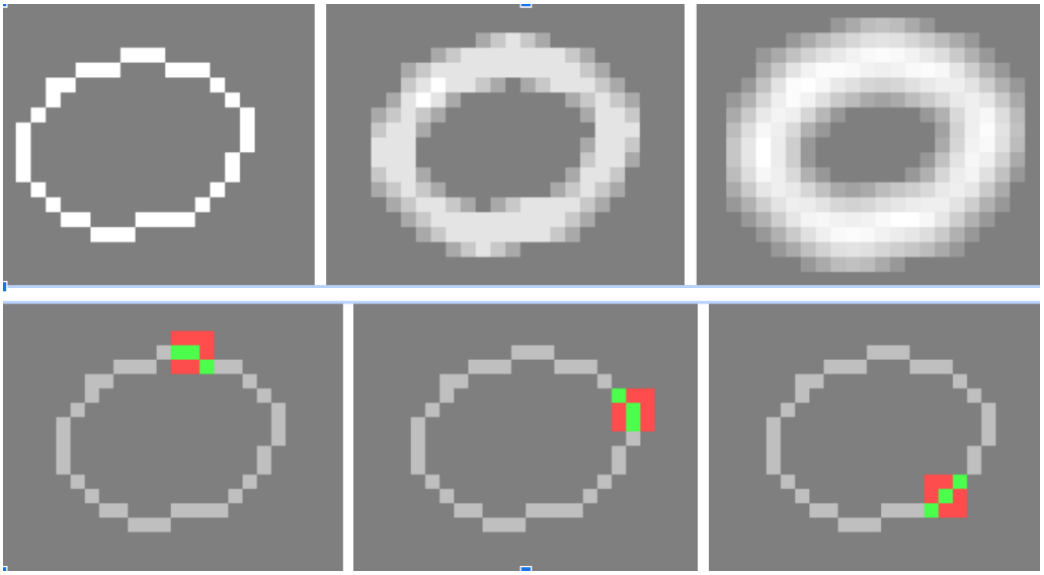


Figure 1: Top) an example of an image getting dilated after two convolutional layers using a kernel with size 3 X 3, weights 1 / 9, and stride 1. Bottom) a non dilated image, the green label represents pixels which are kept for consecutive layers and the red pixels represents pixels which were supposed to get values in regular CNNs however are not in *SparseSSNet*

3.1 The Particle Sample

For each image a random location in the detector is drawn from a uniform distribution, and a "Particle Bomb" is generated originating at that location to mimic an interaction point. The particle bomb is produced by generating a random number of particles drawn from a uniform distribution in the range of [1, 6], the direction of each particle is chosen from an isotropic distribution. The number of particles from a specific type which can be produced is shown in table 1. $\sim$ 85% of the sample contains particles with typical kinetic energies (E_k) produced upon neutrino interactions within MicroBooNE the energy range (E) for this sample for each particle is shown in table 1 a smaller sample $\sim$ 15% is generated with different configuration which are more oriented to low energy interactions as particle identification becomes more difficult. The momentum (P) range for each particle from the low energy sample is shown in table 1. Finally a random number of muons in the range of [5, 10] are generated in both samples to mimic cosmic rays. The data sample is divided into two: a training sample ($\sim$ 120k images) and a test sample ($\sim$ 23k images).

Particle	e	γ	μ	$\pi^{\pm}$	p	cosmic μ
Multiplicity	0–2	0–2	0–2	0–2	0–3	5–10
E_k [MeV]	50 -1,000	50–1,000	50–3,000	50-2,000	50–4,000	5,000–20,000
P (low E) [MeV/c]	30–100	30–100	85–175	95–195	300–450	5,000–20,000 (E_k)

Table 1: The data sample particle content. For each particle type the multiplicity/event, the kinetic energy range for the regular sample, and the momentum for the low E sample is given. Notice that unlike the particles originating at the simulated "interaction point" the cosmic muons for the low E sample are still defined by their kinetic energy as they are the same for both samples.

3.2 Labels

There are five labels for the supervised learning.

1. **HIP**, protons, typically manifests in a short highly ionized track.
2. **MIP**, muons and charged pions, typically manifests in a longer track.
3. **Shower**, induced by electrons, positrons, and photons above a minimal energy ($\sim$ 33MeV in LAr).
4. **Delta rays**, electrons from hard scattering of other charged particles, mainly muons.
5. **Michel electrons**, produced from a decay of muons.

Within the DL-LEE analysis these are re-classified into two classes

1. **Track**, instead of the HIP and MIP labels.
2. **Shower**, instead of the shower, delta ray, and Michel electron labels.

Notice that the re-classification does not require new inferring; rather it is just a mapping of these original five labels to the newly defined two labels.

4 Pixel Weighting Scheme

To prevent class imbalance, a case where one class is dominating the loss function and the penalty for incorrect predicting of other classes is negligible, we apply a pixel weighting scheme, defining the loss function

$$Loss = \Sigma_i w_i \cdot (\vec{l_i} \cdot log(\vec{P_i})) \tag{1}$$

where w_i is the weight defined for each pixel, $\vec{l_i}$ is the label vector of pixel i (e.g., (1,0,0,0,0) for a HIP) and $\vec{P_i}$ is the softmax probability vector for pixel i.

The sum of two types of weighting is assigned to each pixel: **Cluster weighting** and **vertex weighting** (see fig. 2) .

- **Cluster weighting:** big clusters contain many pixels, which makes it easier to correctly label as there is more information. Labeling a big cluster correctly reduces the loss function by a large amount. Small clusters should therefore be treated with more care to prevent the loss function of being governed by one correctly labeled big cluster. We apply a weight which is inversely proportional to the size of the cluster in the range of $(0.02 - 2) \times 10^{-2}$

- **Vertex Weighting:** pixels at center of a cluster are easier to identify as they cannot be confused by other pixels in their close vicinity. However, pixels near "different label" clusters are the hardest to recognize and impact vertex reconstruction dramatically. We apply a 0.02 weight to pixels within three pixel distance from a pixel with a different label, and zero to all other pixels.

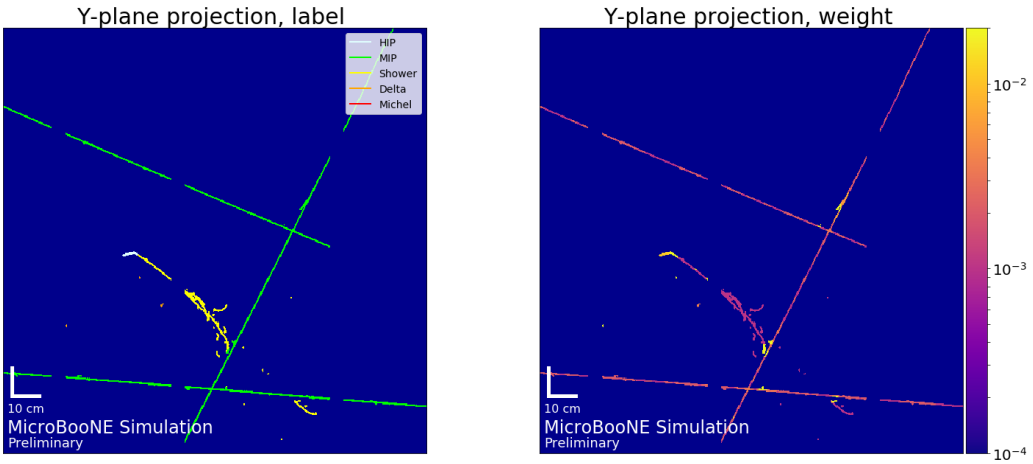


Figure 2: Example from the test data sample. Left) The labels assigned for each pixel according to generated particles. Right) Weighting scheme: for each cluster a weight proportional to the inverse of its size is assigned ($[2 \times 10^{-4} - 2 \times 10^{-2}\ ]$). For crossing type pixels a weight of 2×10^{-2} is assigned.

5 Results

The accuracy (with respect to non-zero pixels only) and loss on the training sample both with and without weighting is shown in fig. 3, along with the accuracy on the test sample.

The five class semantic segmentation confusion matrices obtained from the test sample for the Y plane is shown in fig. 4. The matrices for the U and V planes are fairly similar and are presented in appendix A. The number of pixels obtained for each class is $> 10^5$; thus statistical uncertainties are negligible.

By re-classifying and defining the track (HIP, MIP) and shower (shower, delta, Michel) classes, the two class semantic segmentation confusion matrices can be obtained for each plane (see fig. 5). The number of pixels obtained for each class is $>10^7$; thus statistical uncertainties are negligible.

An example of an event from the test sample is presented in fig. 6. This display encapsulates the weighting scheme and the performance of the network and contains the pixel intensity, the truth level label, the weight applied to each pixel, and the *SparseSSNet* predictions. For more event displays see appendix B.

Finally, examples of simulated ν_e and ν_μ from the MicroBooNE detector are shown in fig. 7 and fig. 9 respectively. Zoomed in versions of this events are shown in fig. 8 and fig. 10 respectively. Each figure consists of the generated interaction particles type (top), the ADC counts of the generated interaction overlaid with cosmic rays extracted from off-beam data samples (middle), and the *SparseSSNet* predictions (bottom). For more event displays supporting the consistency of the results see appendix C.

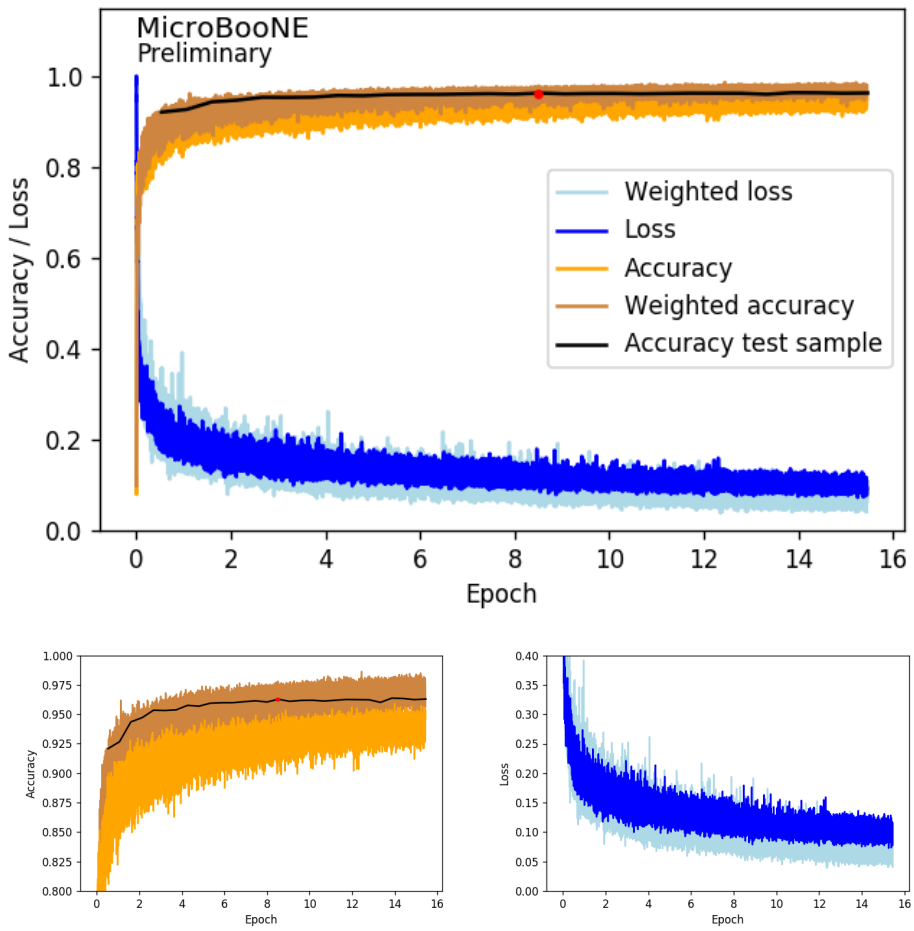


Figure 3: The accuracy and loss of the network on the training and inference, Y plane data sets. The accuracy before (orange) and after (peru) applying weighting, the loss function normalized by the loss after first iteration before (blue) and after (light blue) applying weighting. The accuracy on the inference data sample is indicated in black. The selected network weights is indicated by the red dot. The bottom plots are zoomed in.

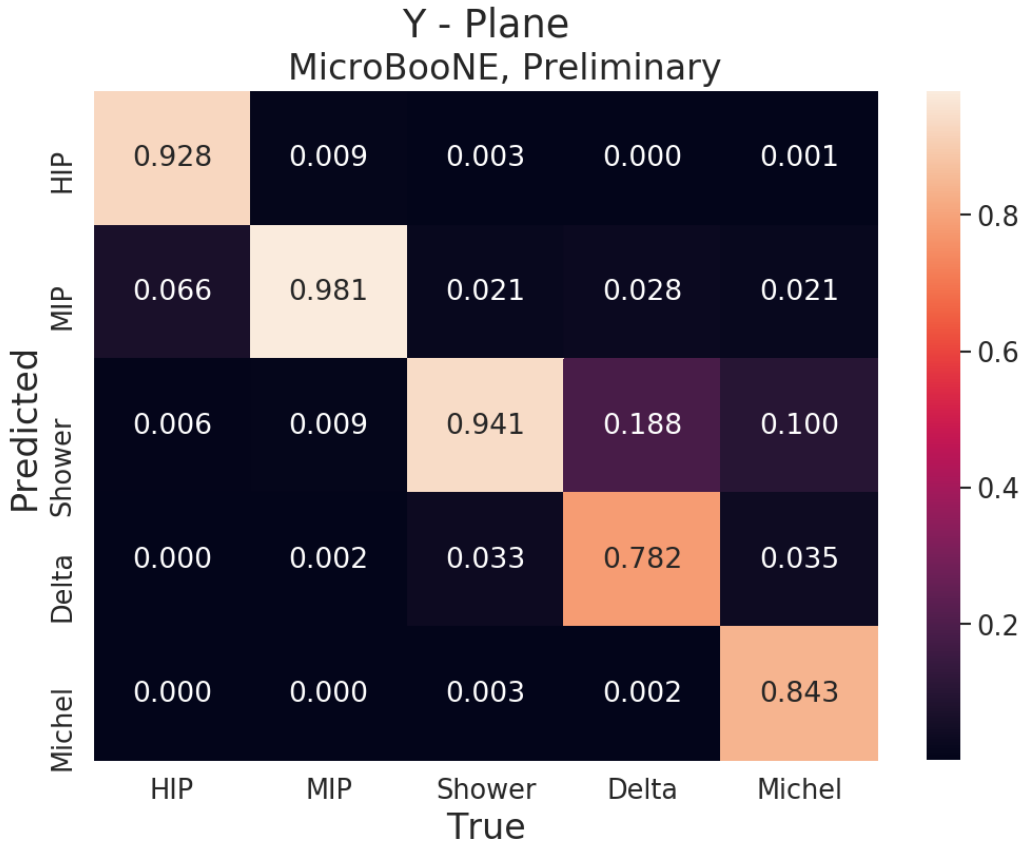
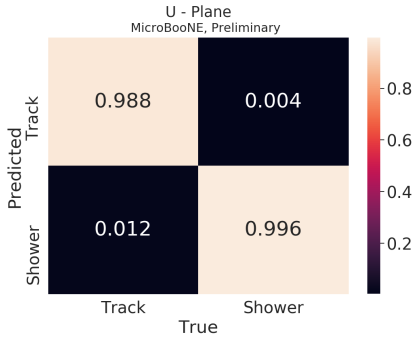
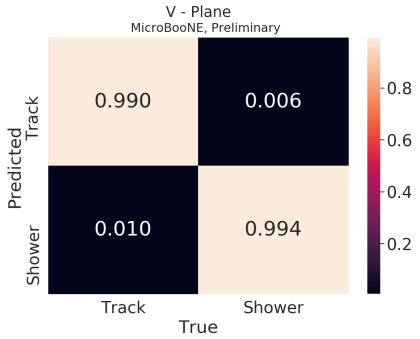


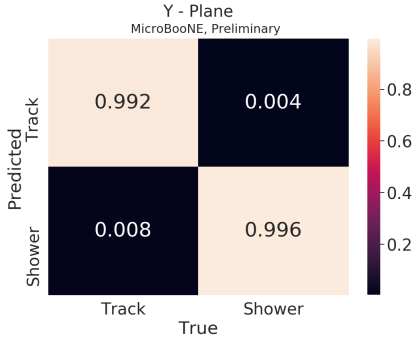
Figure 4: Five class confusion matrices obtained from validation sample for the Y-Plane, each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y axis from the test sample. The smallest number of pixels is O(10^5) for Michel all other classes vary between $10^6 - 10^7$ pixels.



(a)



(b)



(c)

Figure 5: Two class confusion matrices for all three planes, obtained from re-normalizing the five class matrices. Each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y axis from the test sample. The number of pixels associated with each class is $O(10^7)$ pixels.

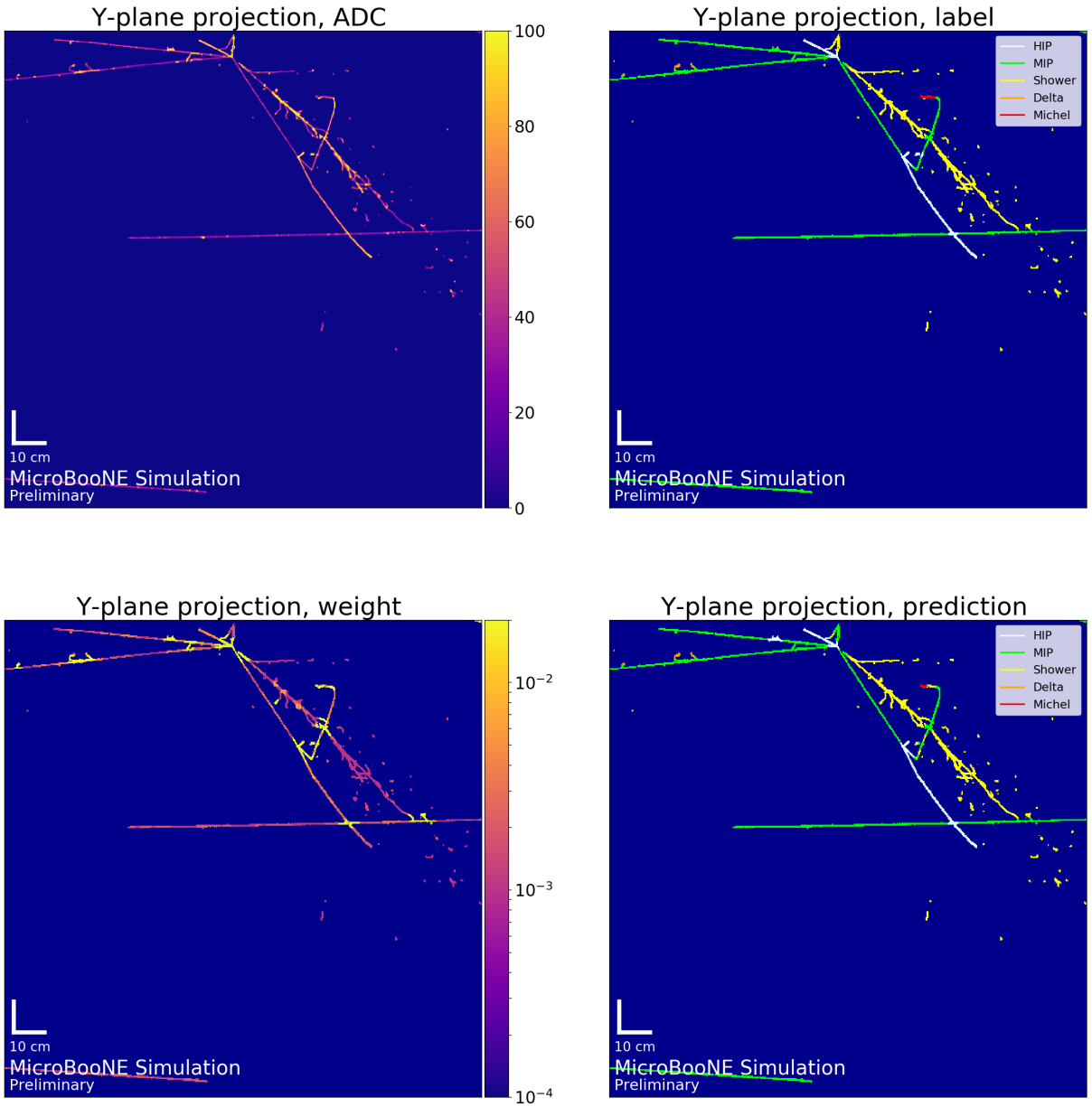


Figure 6: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SpsarseSSNet* predictions

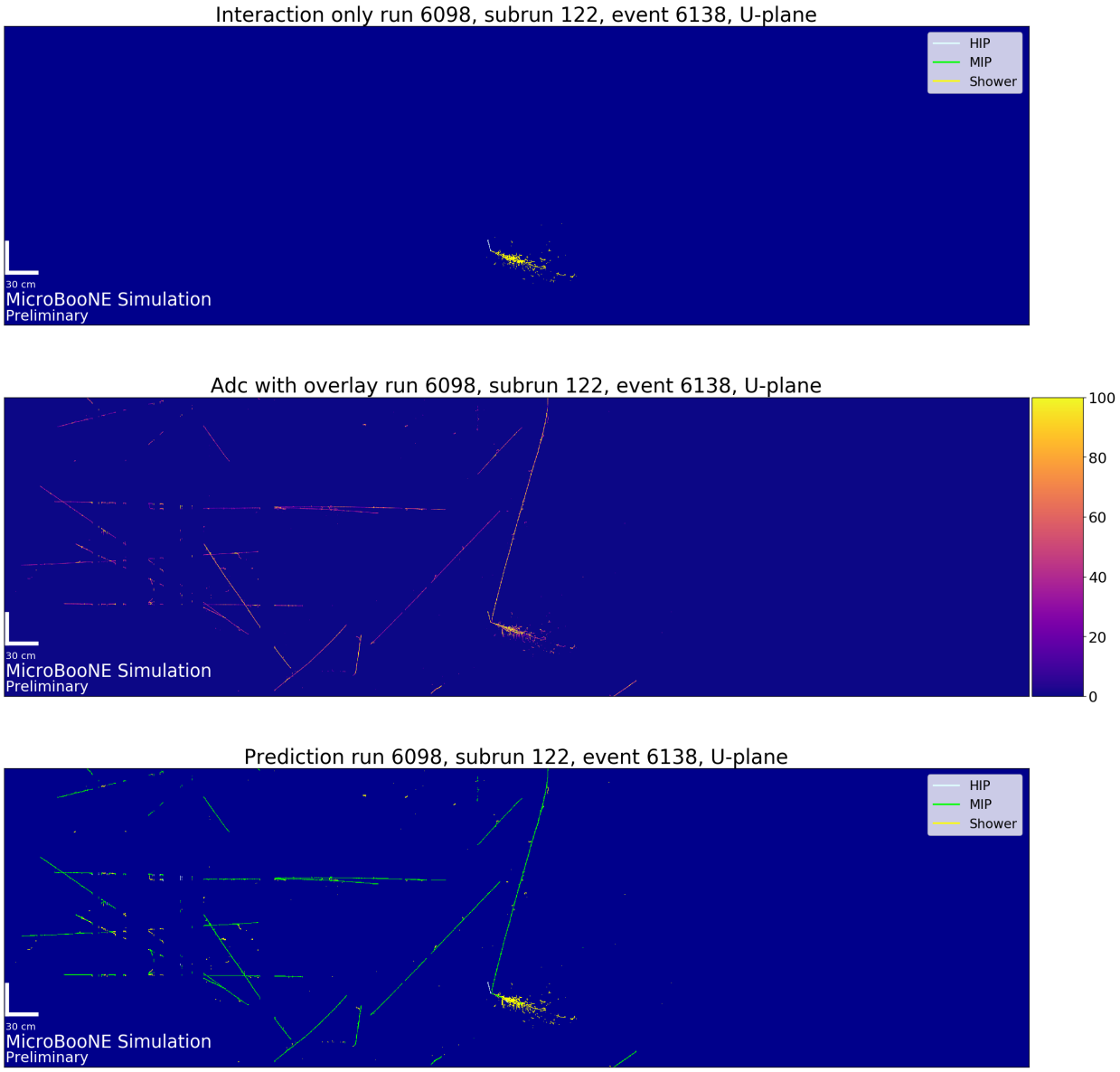


Figure 7: An example of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

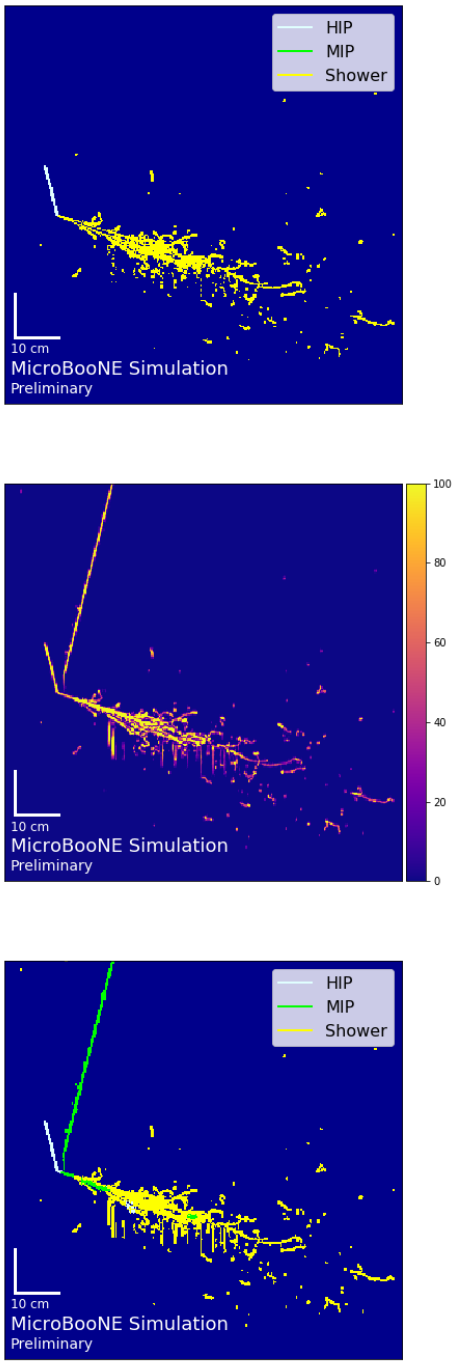


Figure 8: A zoomed version of fig. 7 of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions

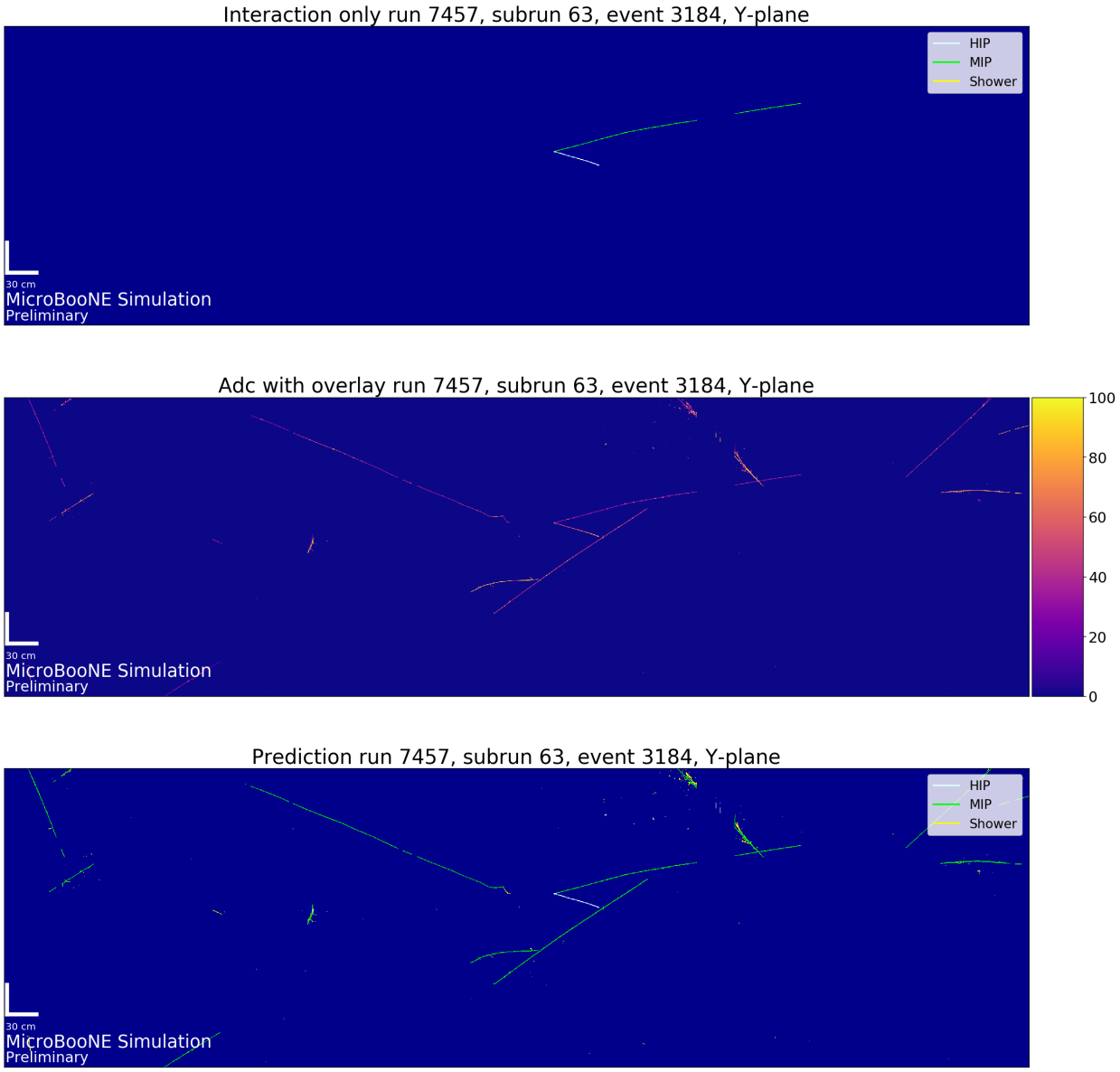


Figure 9: An example of a ν_μ interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

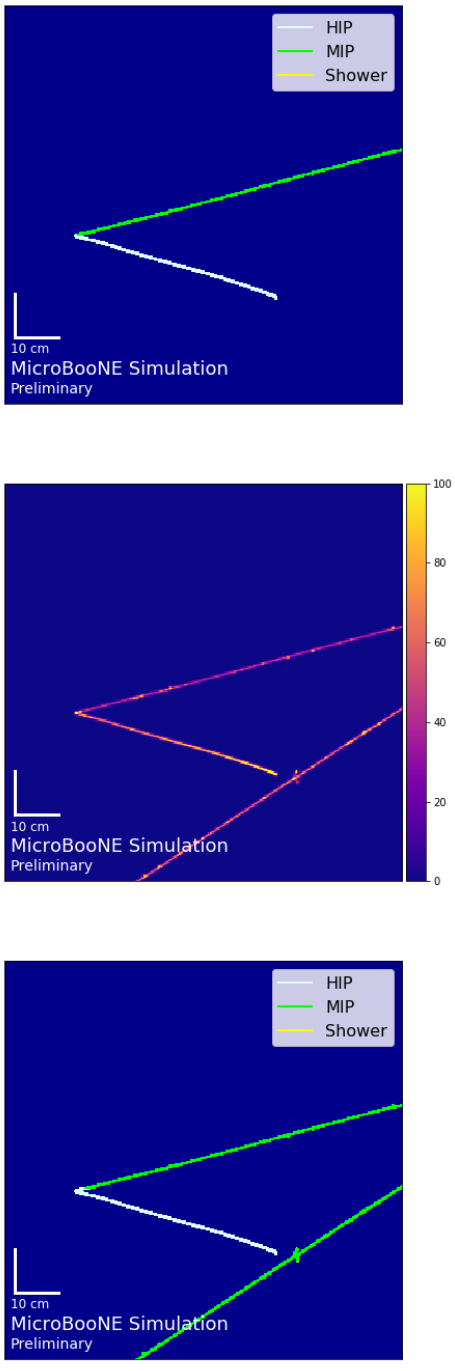


Figure 10: A zoomed version of fig. 9 of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions

6 Summary

We have presented the performance of *SparseSSNet* in the task of semantic segmentation on simulated data from the MicroBooNE detector. This is the first machine learning algorithm in the DL-LEE analysis chain. The adaptation to sparse representation improves dramatically the inference time from $\sim 64 \times 5\,\mathrm{s}$ to $\sim 0.5\,\mathrm{s}$ as well as the memory usage from $\sim 64 \times 5\,\mathrm{GB}$ to $\sim 1\,\mathrm{GB}$[1], where the 64 stands for the amount of cropped image needed per event. In addition there is an improvement in the accuracy on the test sample due to no dilation. The current analysis uses only two classes (track and shower), however the network produces five class segmentation which we hope to exploited in future analyses.

References

[1] A.A. Aguilar-Arevalo et al. Significant Excess of ElectronLike Events in the MiniBooNE Short-Baseline Neutrino Experiment. *Phys. Rev. Lett.*, 121(22):221801, 2018.

[2] Y. LeCun, Y. Bengio, and G.Hinton. Deep learning. *Nature*, 521:436, 2015.

[3] R. Acciarri et al. Design and Construction of the MicroBooNE Detector. *JINST*, 12(02):P02017, 2017.

[4] R. Acciarri et al. Noise Characterization and Filtering in the MicroBooNE Liquid Argon TPC. *JINST*, 12(08):P08003, 2017.

[5] C. Adams et al. Deep neural network for pixel-level electromagnetic particle identification in the MicroBooNE liquid argon time projection chamber. *Phys. Rev. D*, 99(9):092001, 2019.

[6] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks. *CoRR*, abs/1706.01307, 2017.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[1] The performance test where done on a Intel core i7-8750H CPU 2.2GHz

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

A Confusion matrices

The confusion matrices of images from two induction planes, the results are similar to the one presented in 4, and are added here for completeness.

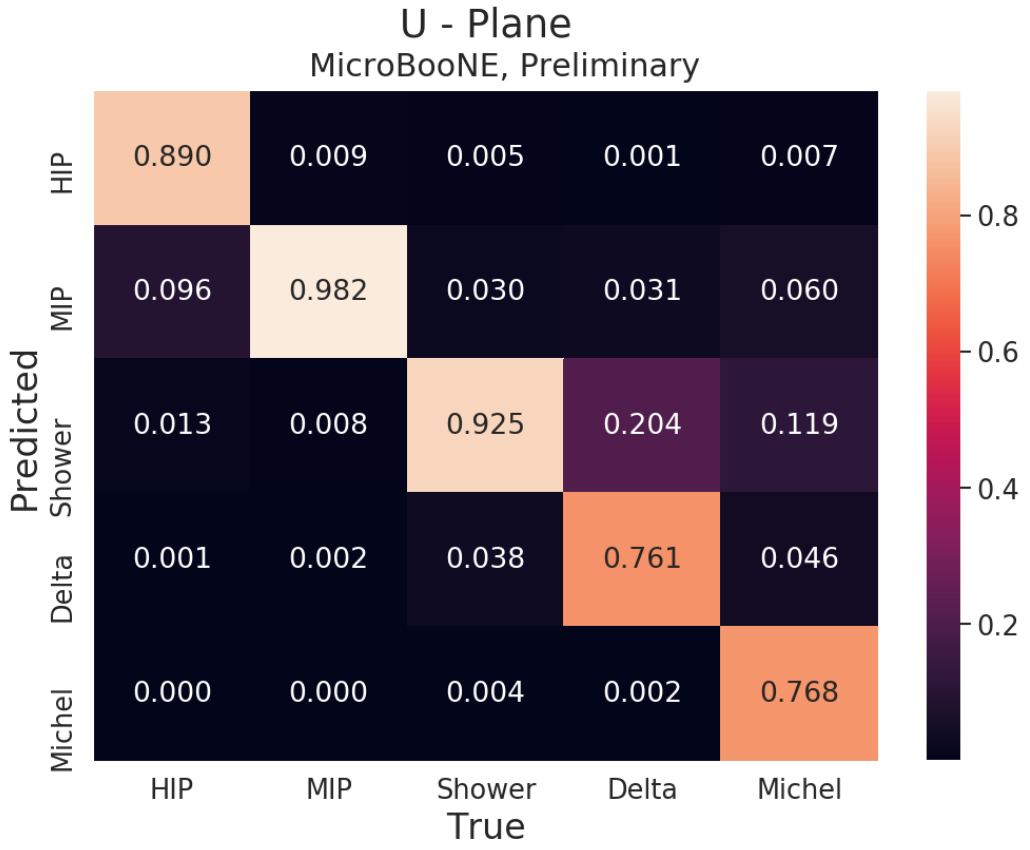


Figure 11: Five class confusion matrices for the U-Plane, each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y axis from the test sample. The smallest number of pixels is O(10^5) for Michel. All other classes vary between $10^6 - 10^7$ pixels.

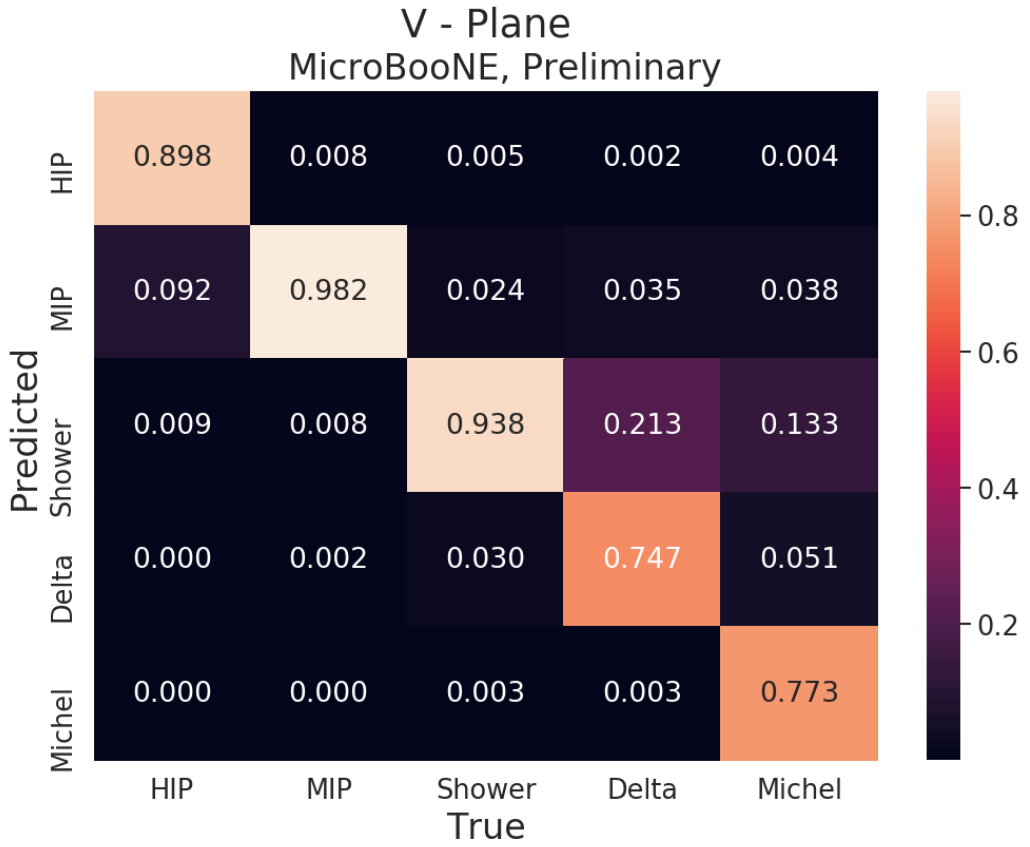


Figure 12: Five class confusion matrices for the V-Plane, each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y axis from the test sample. The smallest number of pixels is O(10^5) for Michel all other classes vary between $10^6 - 10^7$ pixels.

B Test sample event displays

More event displays from the test sample supporting the consistency of *SparseSSNet* performance, as well as showing more information about the pixel-weighting scheme.

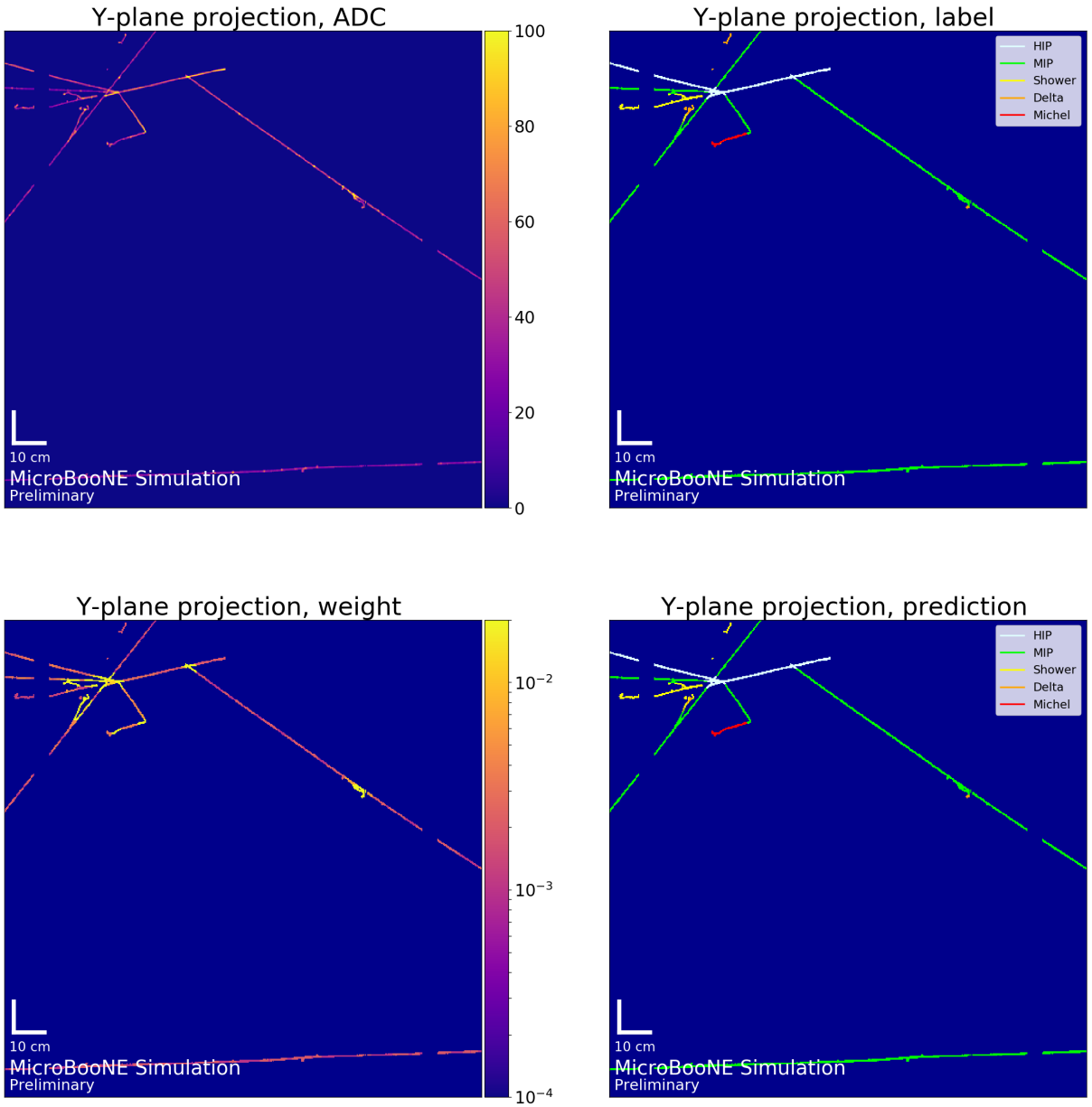


Figure 13: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

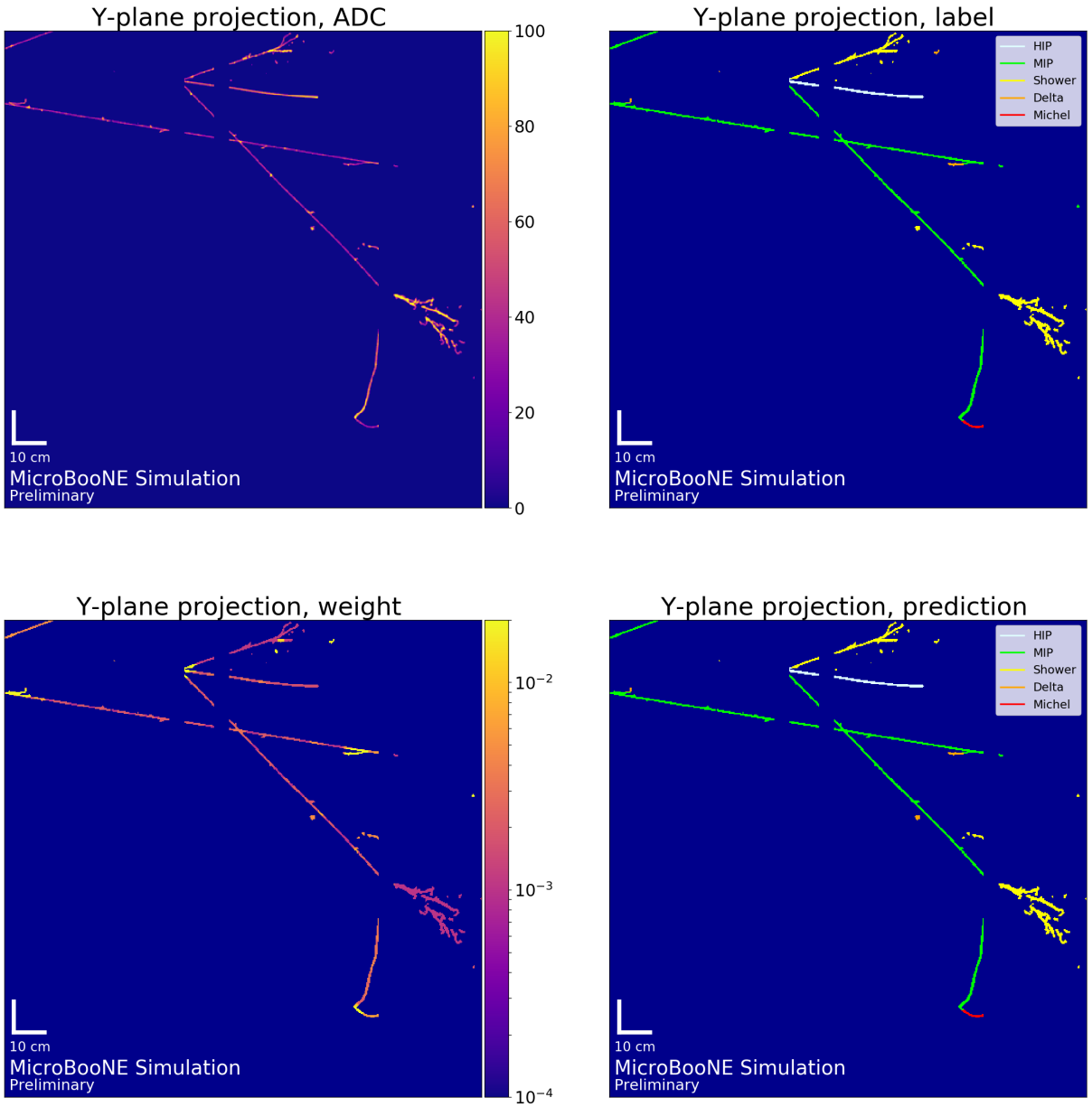


Figure 14: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

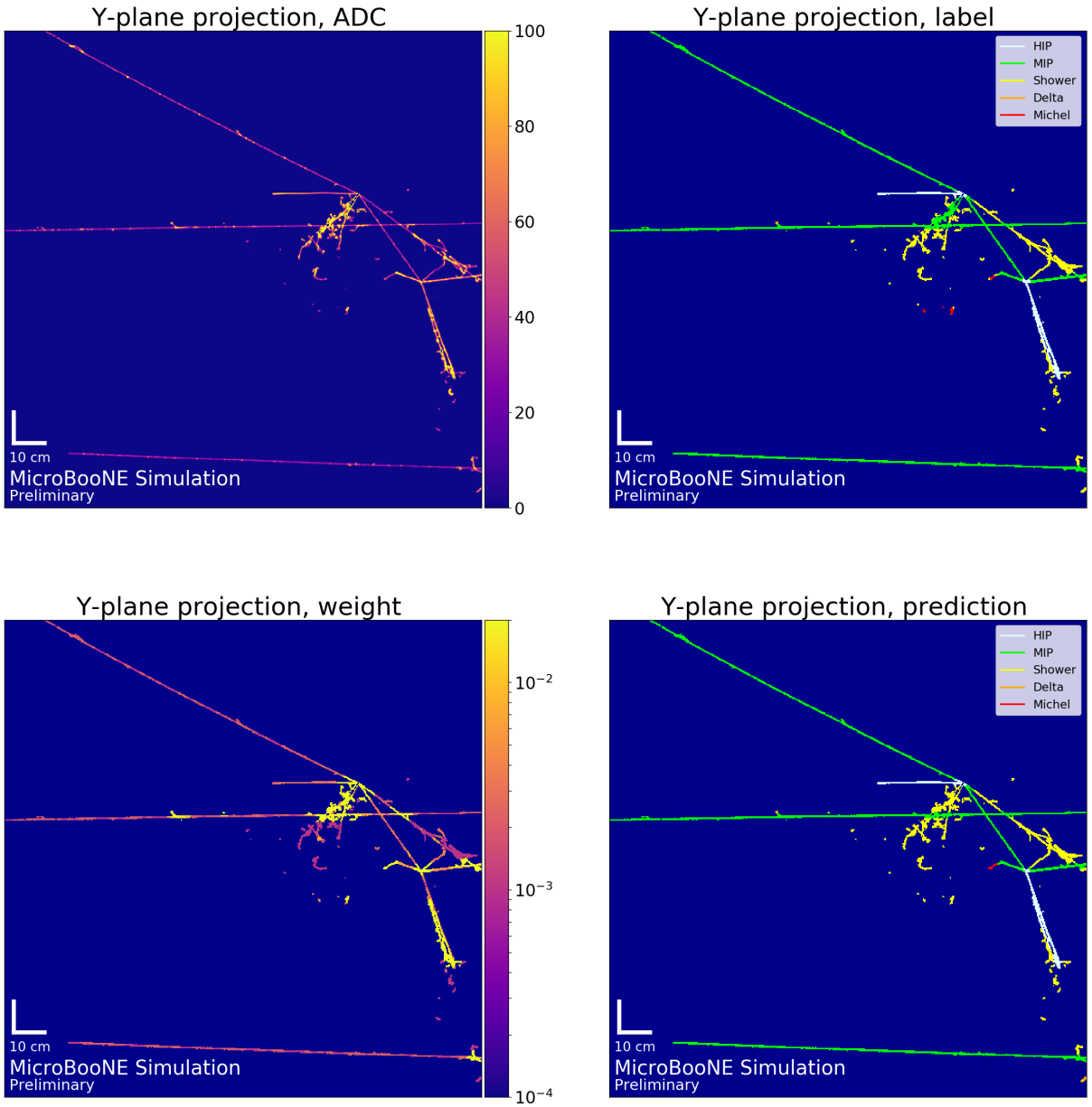


Figure 15: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

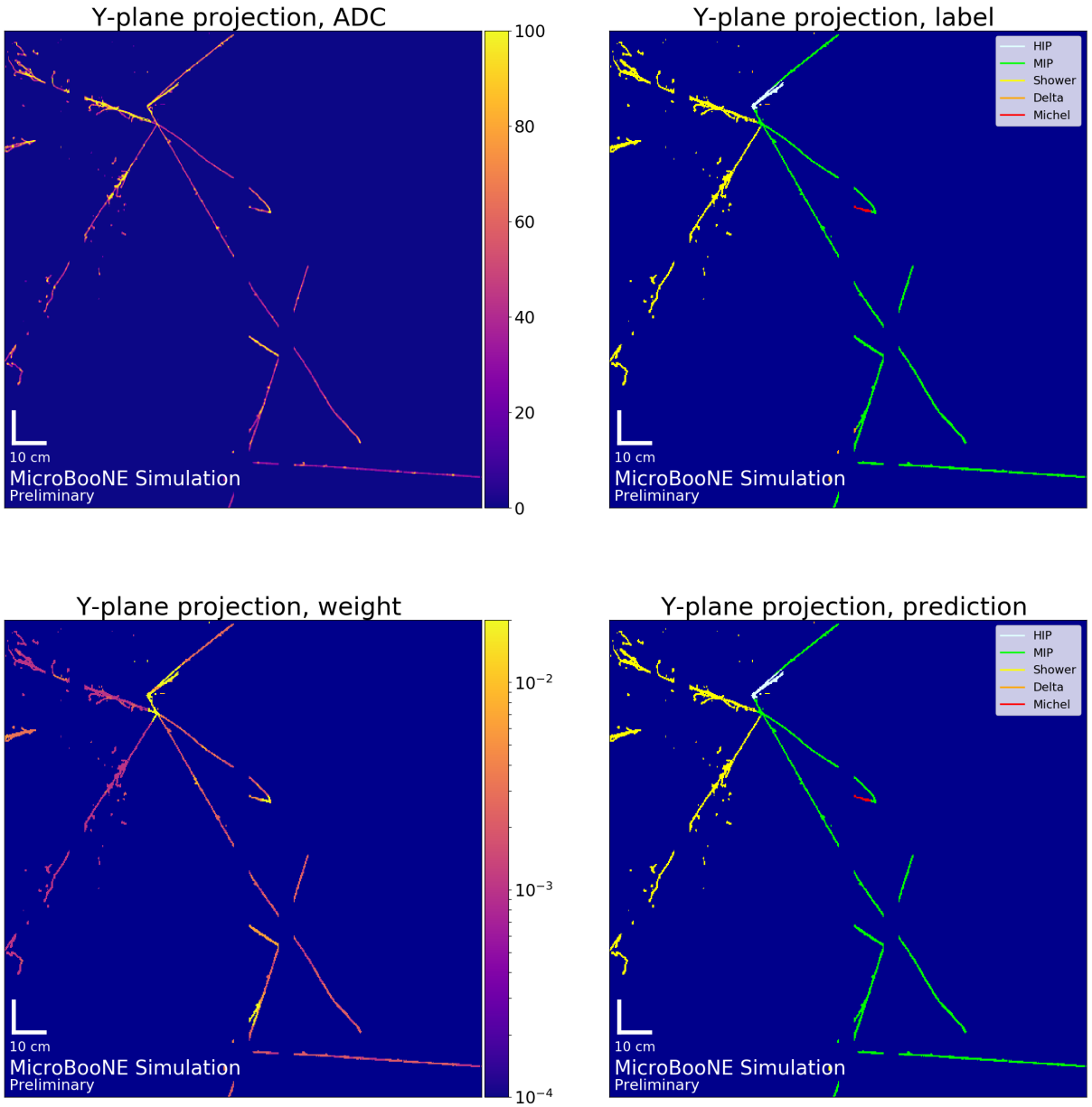


Figure 16: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

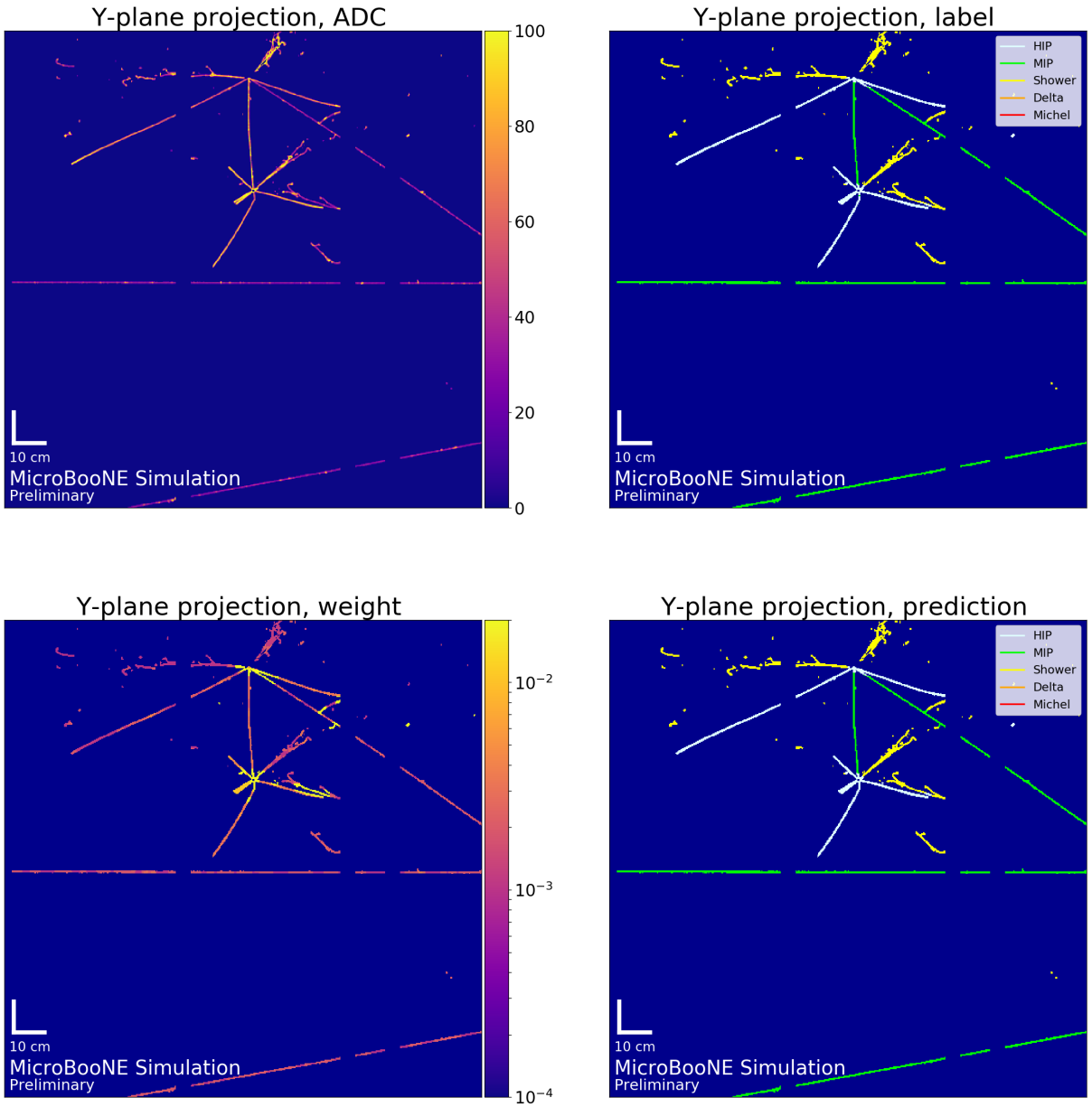


Figure 17: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

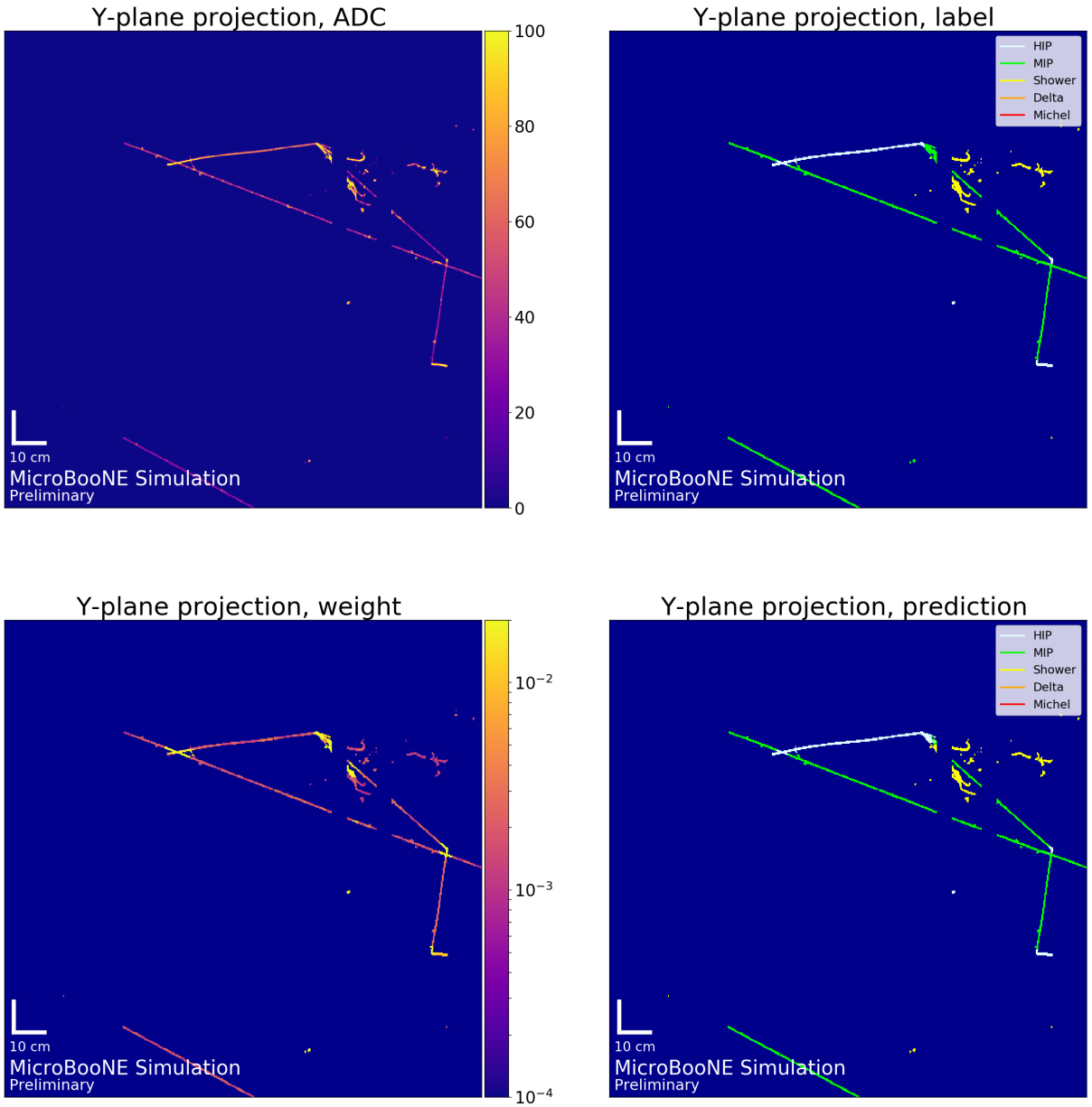


Figure 18: An example of an event from the test sample. Top left) pixel intensity (ADC). Top right) truth label. Bottom left) weight applied to each pixel. Bottom right) *SparseSSNet* predictions

C ν interaction event displays

Additional event displays from full ν interactions. These events support the consistency of *SparseSSNet* performance on different track/shower sizes as well as an example of a Michel electron (fig. 25)

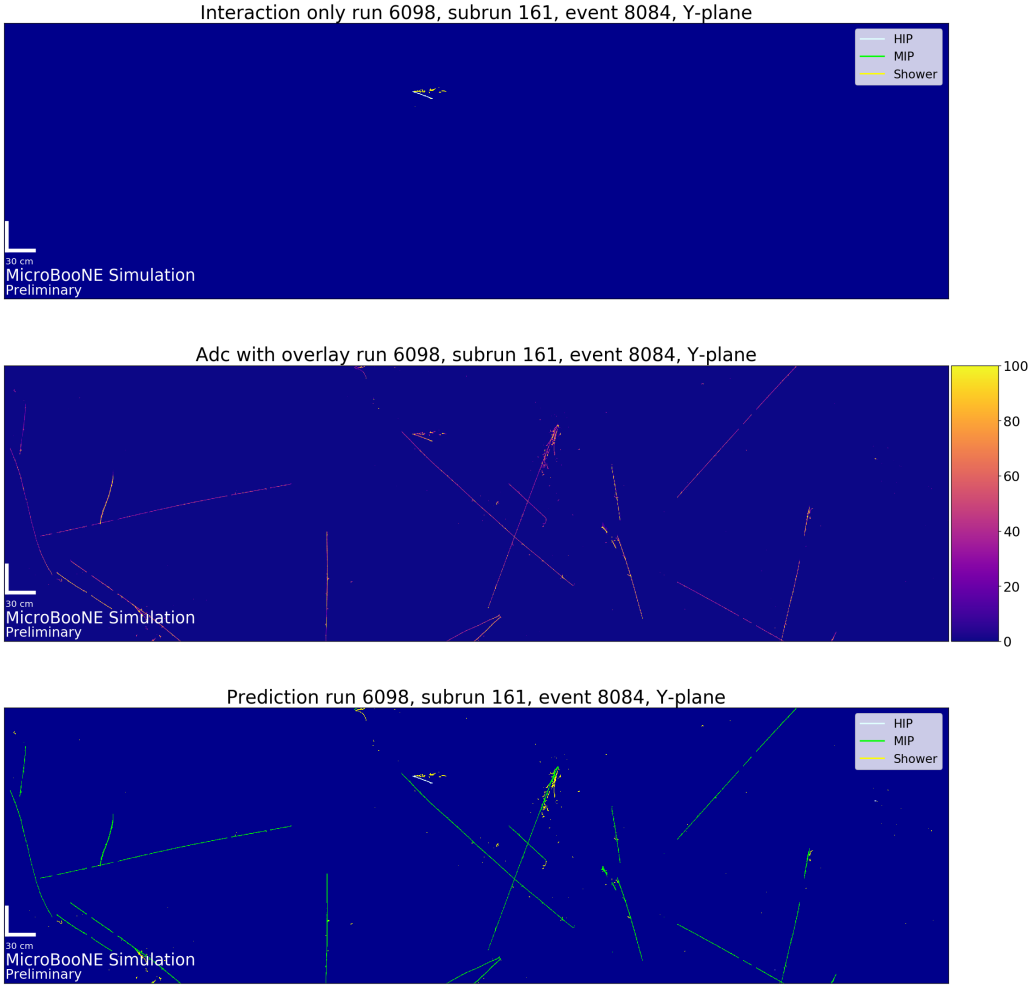


Figure 19: An example of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

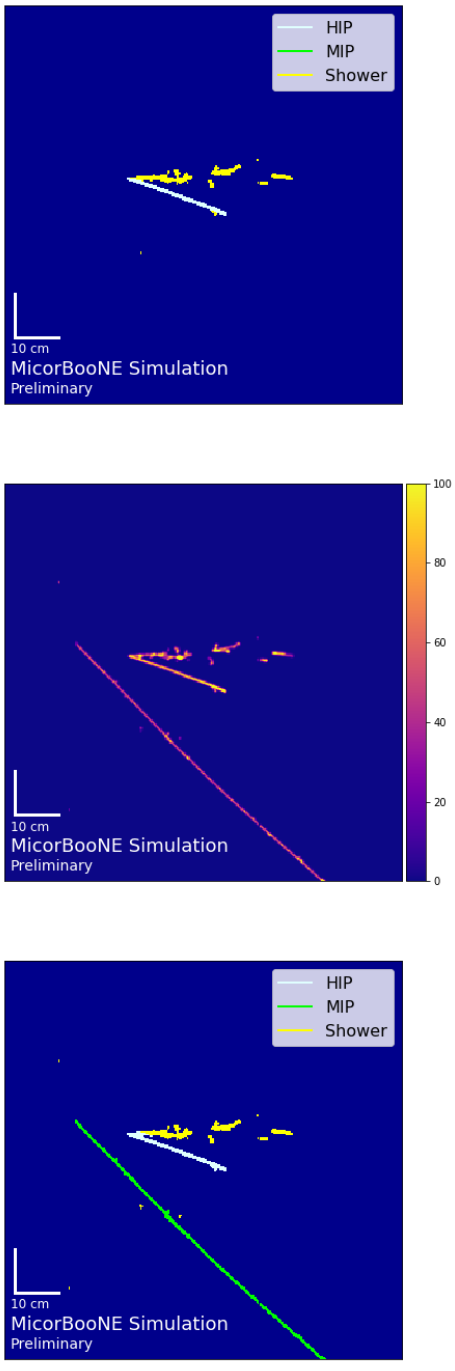


Figure 20: A zoomed version of fig. 19 of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions

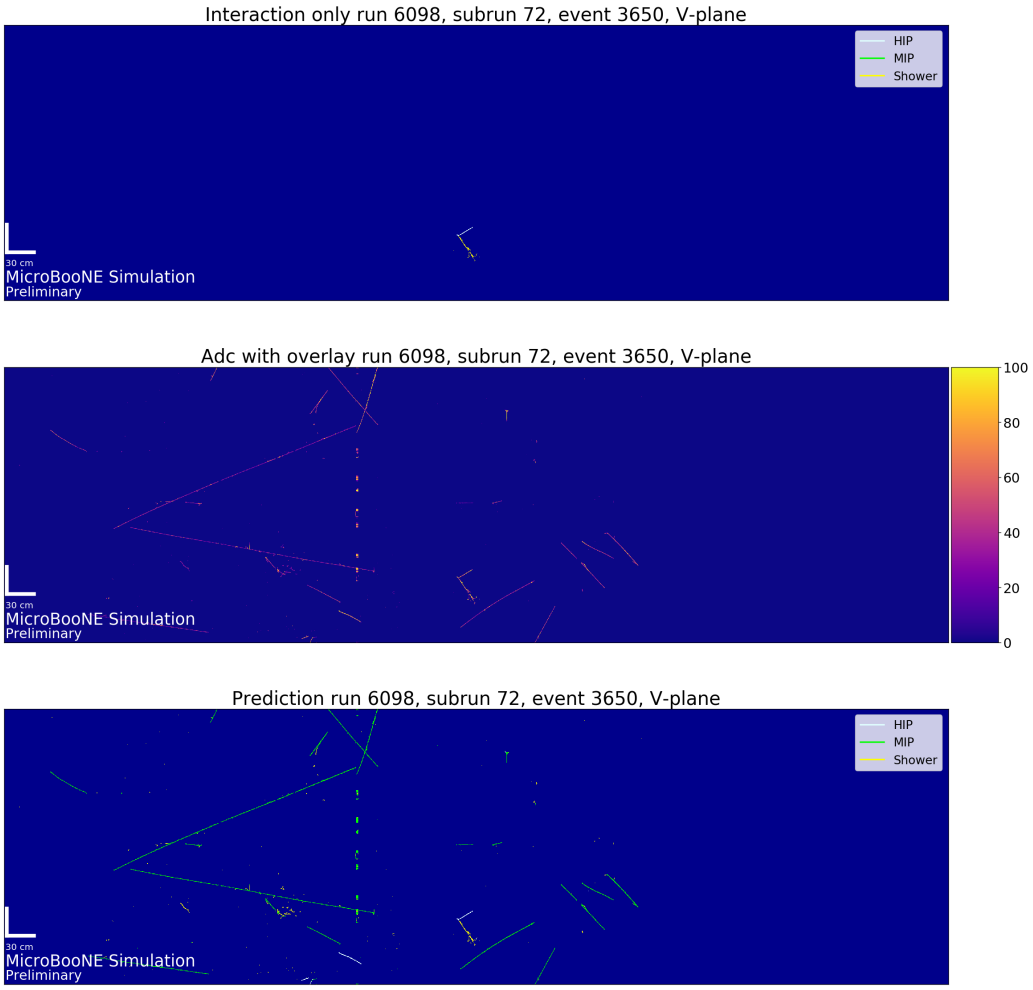


Figure 21: An example of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

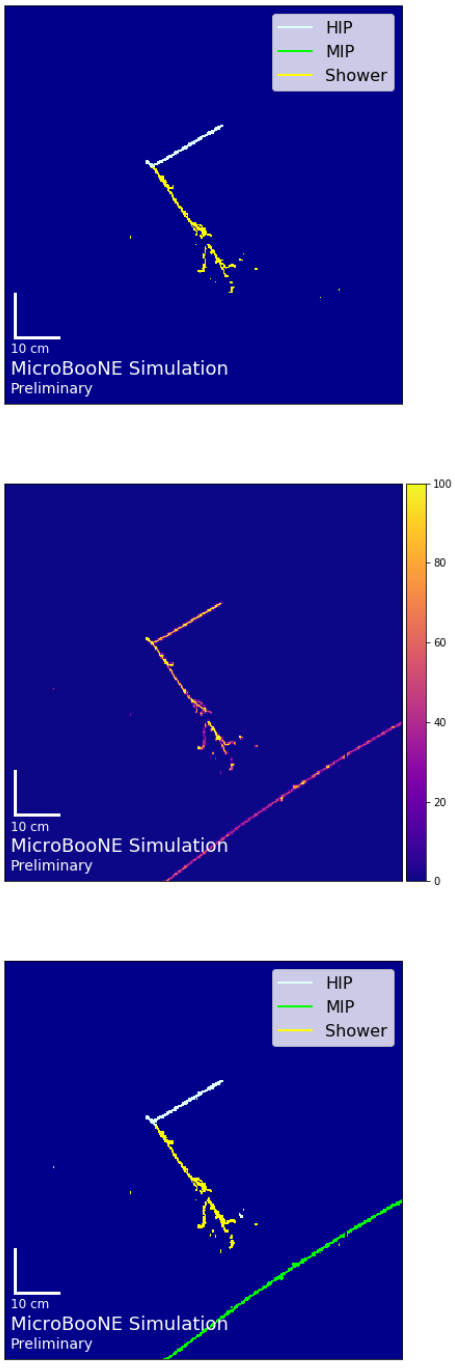


Figure 22: A zoomed version of fig. 21 of a ν_e interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions

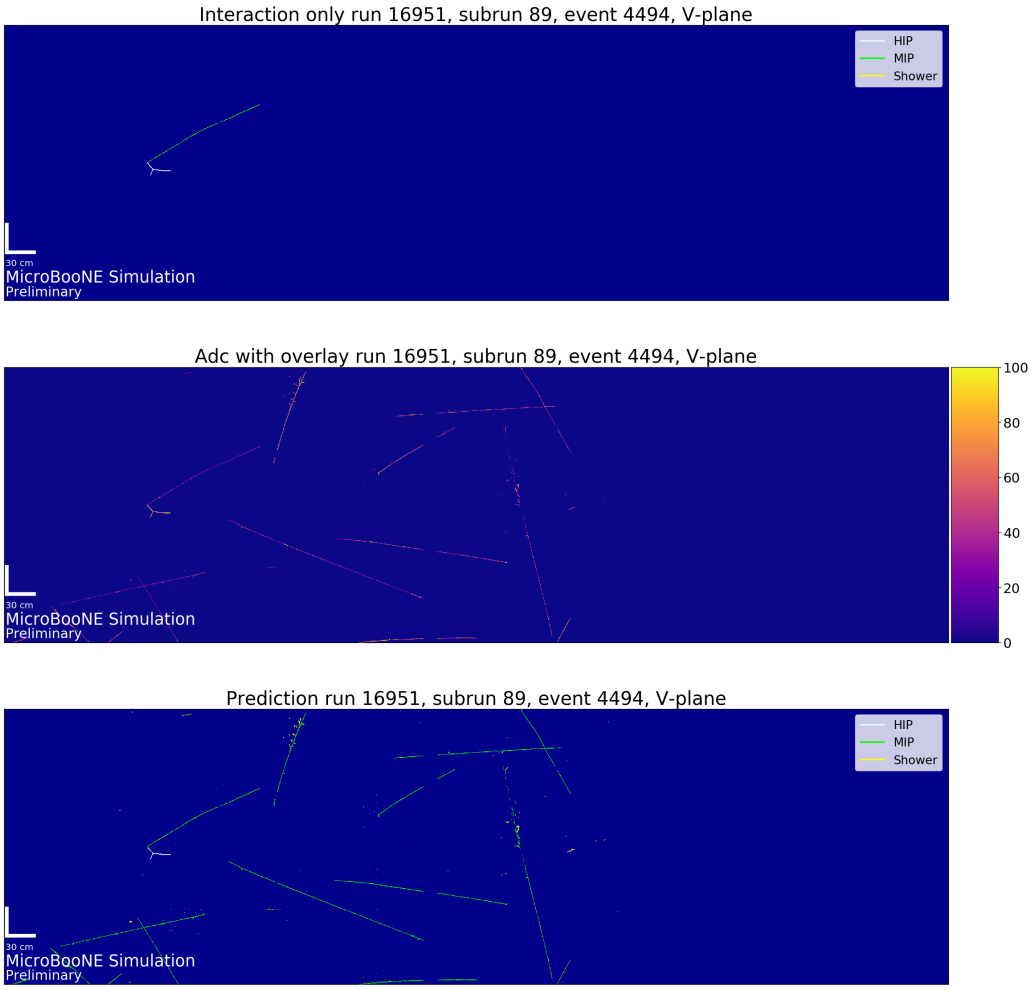


Figure 23: An example of a ν_μ interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

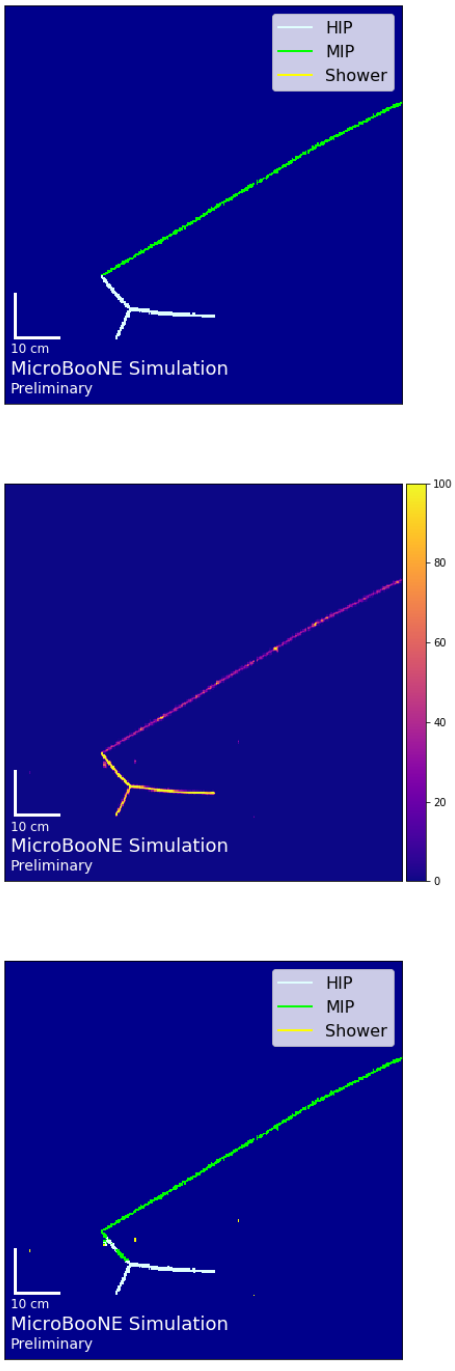


Figure 24: A zoomed version of fig. 9 of a ν_μ interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions

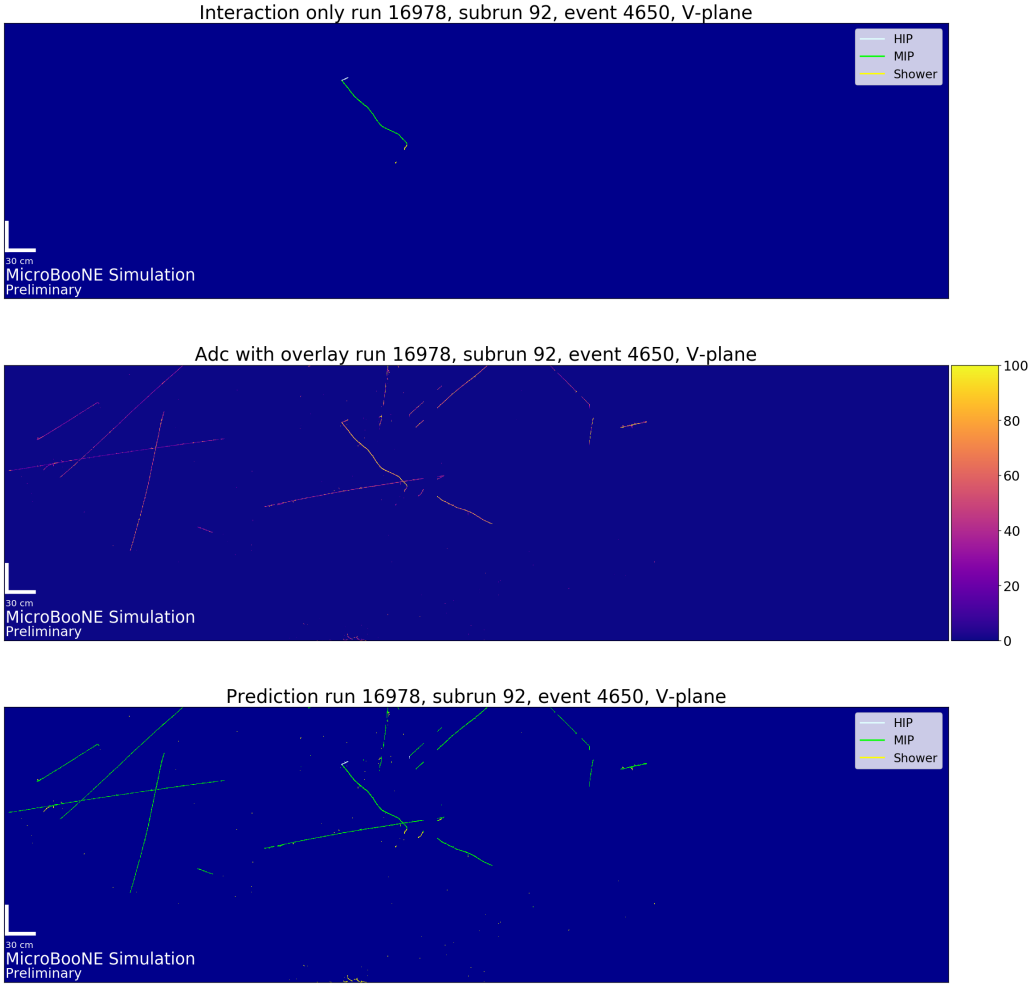


Figure 25: An example of a ν_μ interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions.

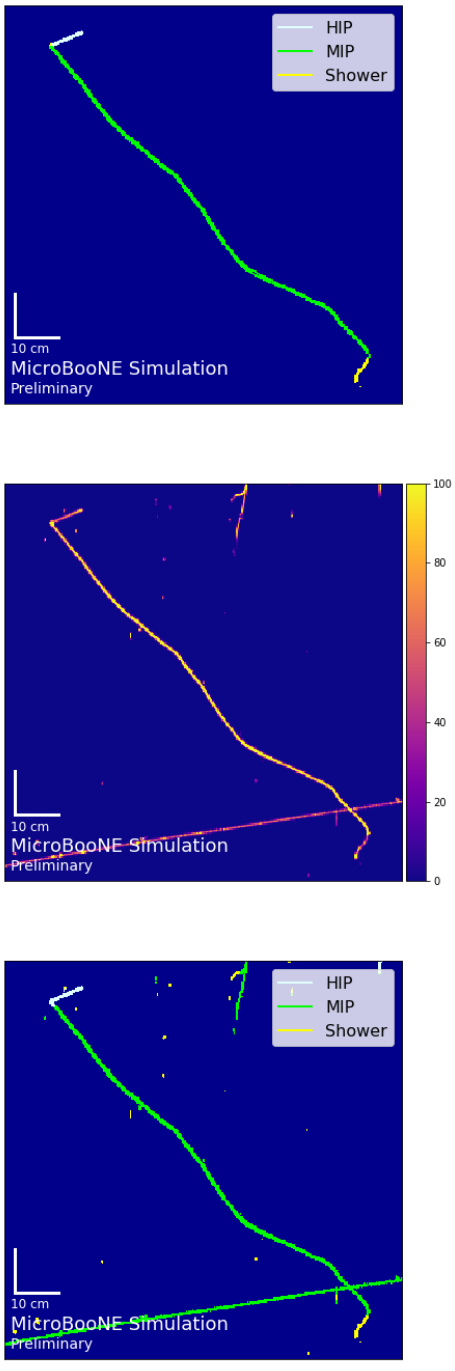


Figure 26: A zoomed version of fig. 25 of a ν_μ interaction. Top) the produced particles upon interaction. Middle) pixel intensity of interaction overlaid with cosmic rays. Bottom) *SparseSSNet* predictions