

Semantic segmentation with a sparse convolutional neural network for event reconstruction in MicroBooNE

P. Abratenko,³⁶ M. Alrashed,¹⁵ R. An,¹⁴ J. Anthony,⁴ J. Asaadi,³⁵ A. Ashkenazi,^{19,33} S. Balasubramanian,³⁹ B. Baller,¹¹ C. Barnes,²⁰ G. Barr,²⁴ V. Basque,¹⁸ L. Bathe-Peters,¹³ O. Benevides Rodrigues,³² S. Berkman,¹¹ A. Bhandari,¹⁸ A. Bhat,³² M. Bishai,² A. Blake,¹⁶ T. Bolton,¹⁵ L. Camilleri,¹⁰ D. Caratelli,¹¹ I. Caro Terrazas,⁹ R. Castillo Fernandez,¹¹ F. Cavanna,¹¹ G. Cerati,¹¹ Y. Chen,¹ E. Church,²⁵ D. Cianci,¹⁰ J. M. Conrad,¹⁹ M. Convery,²⁹ L. Cooper-Troendle,³⁹ J. I. Crespo-Anadón,⁶ M. Del Tutto,¹¹ S. R. Dennis,⁴ D. Devitt,¹⁶ R. Diurba,²¹ R. Dorrill,¹⁴ K. Duffy,¹¹ S. Dytman,²⁶ B. Eberly,³¹ A. Ereditato,¹ J. J. Evans,¹⁸ G. A. Fiorentini Aguirre,³⁰ R. S. Fitzpatrick,²⁰ B. T. Fleming,³⁹ N. Foppiani,¹³ D. Franco,³⁹ A. P. Furmanski,²¹ D. Garcia-Gamez,¹² S. Gardiner,¹¹ G. Ge,¹⁰ S. Gollapinni,^{34,17} O. Goodwin,¹⁸ E. Gramellini,¹¹ P. Green,¹⁸ H. Greenlee,¹¹ W. Gu,² R. Guenette,¹³ P. Guzowski,¹⁸ L. Hagaman,³⁹ E. Hall,¹⁹ P. Hamilton,³² O. Hen,¹⁹ G. A. Horton-Smith,¹⁵ A. Hourlier,¹⁹ R. Itay,²⁹ C. James,¹¹ J. Jan de Vries,⁴ X. Ji,² L. Jiang,³⁷ J. H. Jo,³⁹ R. A. Johnson,⁸ Y.-J. Jwa,¹⁰ N. Kamp,¹⁹ N. Kaneshige,³ G. Karagiorgi,¹⁰ W. Ketchum,¹¹ B. Kirby,² M. Kirby,¹¹ T. Kobilarcik,¹¹ I. Kreslo,¹ R. LaZur,⁹ I. Lepetic,²⁷ K. Li,³⁹ Y. Li,² B. R. Littlejohn,¹⁴ W. C. Louis,¹⁷ X. Luo,³ A. Marchionni,¹¹ C. Mariani,³⁷ D. Marsden,¹⁸ J. Marshall,³⁸ J. Martin-Albo,¹³ D. A. Martinez Caicedo,³⁰ K. Mason,³⁶ A. Mastbaum,²⁷ N. McConkey,¹⁸ V. Meddage,¹⁵ T. Mettler,¹ K. Miller,⁷ J. Mills,³⁶ K. Mistry,¹⁸ A. Mogan,³⁴ T. Mohayai,¹¹ J. Moon,¹⁹ M. Mooney,⁹ A. F. Moor,⁴ C. D. Moore,¹¹ L. Mora Lepin,¹⁸ J. Mousseau,²⁰ M. Murphy,³⁷ D. Naples,²⁶ A. Navrer-Agasson,¹⁸ R. K. Neely,¹⁵ P. Nienaber,²⁸ J. Nowak,¹⁶ O. Palamara,¹¹ V. Paolone,²⁶ A. Papadopoulou,¹⁹ V. Papavassiliou,²² S. F. Pate,²² A. Paudel,¹⁵ Z. Pavlovic,¹¹ E. Piasetzky,³³ I. D. Ponce-Pinto,¹⁰ S. Prince,¹³ X. Qian,² J. L. Raaf,¹¹ V. Radeka,² A. Rafique,¹⁵ M. Reggiani-Guzzo,¹⁸ L. Ren,²² L. Rochester,²⁹ J. Rodriguez Rondon,³⁰ H. E. Rogers,⁵ M. Rosenberg,²⁶ M. Ross-Lonergan,¹⁰ B. Russell,³⁹ G. Scanavini,³⁹ D. W. Schmitz,⁷ A. Schukraft,¹¹ W. Seligman,¹⁰ M. H. Shaevitz,¹⁰ R. Sharankova,³⁶ J. Sinclair,¹ A. Smith,⁴ E. L. Snider,¹¹ M. Soderberg,³² S. Söldner-Rembold,¹⁸ S. R. Soleti,^{24,13} P. Spentzouris,¹¹ J. Spitz,²⁰ M. Stancari,¹¹ J. St. John,¹¹ T. Strauss,¹¹ K. Sutton,¹⁰ S. Sword-Fehlberg,²² A. M. Szec,¹⁸ N. Tagg,²³ W. Tang,³⁴ K. Terao,²⁹ C. Thorpe,¹⁶ M. Touns,¹¹ Y.-T. Tsai,²⁹ M. A. Uchida,⁴ T. Usher,²⁹ W. Van De Pontseele,^{24,13} B. Viren,² M. Weber,¹ H. Wei,² Z. Williams,³⁵ S. Wolbers,¹¹ T. Wongjirad,³⁶ M. Wospakrik,¹¹ W. Wu,¹¹ E. Yandel,³ T. Yang,¹¹ G. Yarbrough,³⁴ L. E. Yates,¹⁹ G. P. Zeller,¹¹ J. Zennaro,¹¹ and C. Zhang²

(The MicroBooNE Collaboration)*

¹Universität Bern, Bern CH-3012, Switzerland

²Brookhaven National Laboratory (BNL), Upton, New York, 11973, USA

³University of California, Santa Barbara, California, 93106, USA

⁴University of Cambridge, Cambridge CB3 0HE, United Kingdom

⁵St. Catherine University, Saint Paul, Minnesota 55105, USA

⁶Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (CIEMAT), Madrid E-28040, Spain

⁷University of Chicago, Chicago, Illinois, 60637, USA

⁸University of Cincinnati, Cincinnati, Ohio, 45221, USA

⁹Colorado State University, Fort Collins, Colorado, 80523, USA

¹⁰Columbia University, New York, New York, 10027, USA

¹¹Fermi National Accelerator Laboratory (FNAL), Batavia, Illinois 60510, USA

¹²Universidad de Granada, Granada E-18071, Spain

¹³Harvard University, Cambridge, Massachusetts 02138, USA

¹⁴Illinois Institute of Technology (IIT), Chicago, Illinois 60616, USA

¹⁵Kansas State University (KSU), Manhattan, Kansas, 66506, USA

¹⁶Lancaster University, Lancaster LA1 4YW, United Kingdom

¹⁷Los Alamos National Laboratory (LANL), Los Alamos, New Mexico, 87545, USA

¹⁸The University of Manchester, Manchester M13 9PL, United Kingdom

¹⁹Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts, 02139, USA

²⁰University of Michigan, Ann Arbor, Michigan, 48109, USA

²¹University of Minnesota, Minneapolis, Minnesota, 55455, USA

²²New Mexico State University (NMSU), Las Cruces, New Mexico, 88003, USA

²³Otterbein University, Westerville, Ohio, 43081, USA

²⁴University of Oxford, Oxford OX1 3RH, United Kingdom

²⁵Pacific Northwest National Laboratory (PNNL), Richland, Washington, 99352, USA

²⁶*University of Pittsburgh, Pittsburgh, Pennsylvania, 15260, USA*²⁷*Rutgers University, Piscataway, New Jersey, 08854, USA*²⁸*Saint Mary's University of Minnesota, Winona, Minnesota, 55987, USA*²⁹*SLAC National Accelerator Laboratory, Menlo Park, California, 94025, USA*³⁰*South Dakota School of Mines and Technology (SDSMT), Rapid City, South Dakota, 57701, USA*³¹*University of Southern Maine, Portland, Maine, 04104, USA*³²*Syracuse University, Syracuse, New York, 13244, USA*³³*Tel Aviv University, Tel Aviv, Israel, 69978*³⁴*University of Tennessee, Knoxville, Tennessee, 37996, USA*³⁵*University of Texas, Arlington, Texas, 76019, USA*³⁶*Tufts University, Medford, Massachusetts, 02155, USA*³⁷*Center for Neutrino Physics, Virginia Tech, Blacksburg, Virginia, 24061, USA*³⁸*University of Warwick, Coventry CV4 7AL, United Kingdom*³⁹*Wright Laboratory, Department of Physics, Yale University, New Haven, Connecticut, 06520, USA*

(Received 17 December 2020; accepted 1 March 2021; published 26 March 2021)

We present the performance of a semantic segmentation network, SparseSSNet, that provides pixel-level classification of MicroBooNE data. The MicroBooNE experiment employs a liquid argon time projection chamber for the study of neutrino properties and interactions. SparseSSNet is a submanifold sparse convolutional neural network, which provides the initial machine learning based algorithm utilized in one of MicroBooNEs ν_e -appearance oscillation analyses. The network is trained to categorize pixels into five classes, which are reclassified into two classes more relevant to the current analysis. The output of SparseSSNet is a key input in further analysis steps. This technique, used for the first time in liquid argon time projection chambers data and is an improvement compared to a previously used convolutional neural network, both in accuracy and computing resource utilization. The accuracy achieved on the test sample is $\geq 99\%$. For full neutrino interaction simulations, the time for processing one image is ≈ 0.5 sec, the memory usage is at 1 GB level, which allows utilization of most typical CPU worker machine.

DOI: [10.1103/PhysRevD.103.052012](https://doi.org/10.1103/PhysRevD.103.052012)

I. INTRODUCTION

The primary goal of the MicroBooNE experiment is to search for electronlike events, specifically in the kinematic region where an anomaly was reported by the MiniBooNE experiment [1]. The MiniBooNE experiment observed an excess with respect to their background predictions in electron neutrino events below 500 MeV. This excess is often referred to as the MiniBooNE low energy excess (LEE). Four independent analyses from MicroBooNE are targeted at explaining this excess. The analyses differ in reconstruction techniques and in their approach of targeting different signal topologies. Specifically, the deep learning (DL) LEE analysis uses a combination of machine learning algorithms [2] and traditional tracking tools [3]. As charged current quasielastic (CCQE) is the dominant cross section in the LEE energy range, the approach adopted by the DL-LEE analysis is to study high-purity data samples of CCQE ν_e and ν_μ interactions. The topologies of these interactions are much simpler than other interaction types since they manifest in most cases as one lepton and one proton ($1l1p$), where the lepton is either an electron ($1e1p$) or a muon ($1\mu1p$) for ν_e or ν_μ interactions, respectively. Focusing on

these topologies allows an easier and more precise event selection than trying to select topologies with neutral particles in final states or hadronic interactions.

The MicroBooNE experiment has been collecting data since 2015 and is part of the Short-Baseline Neutrino (SBN) program [4] operating at Fermi National Accelerator Laboratory (FNAL) along the Booster Neutrino Beamline (BNB) [5]. The detector [6] is a 10.4 m long, 2.6 m wide, and 2.3 m high liquid argon time projection chamber (LArTPC), consisting of 170 tons (85 tons in the active volume). The readout time window is 4.8 ms and is digitized into 9600 readout time ticks. Upon neutrino-argon interaction, various particles are produced depending on the interaction channel. The final state charged particles produce prompt scintillation light as well as ionization electrons along their paths in the liquid argon. The light is detected by an array of 32 Hamamatsu 5912-02MOD photomultiplier tubes (PMTs), while the ionization electrons drift in an electric field of 273 V/cm toward the readout wire planes leading to a drift time of 2.3 ms for the maximal distance. There are three wire planes: two induction planes (referred to as U and V) consisting of 2400 wires each and one collection plane (referred to as Y) consisting of 3456 wires. The induction plane wires are aligned at $\pm 60^\circ$ with respect to those of the collection

*microboone_info@fnal.gov

plane, while the collection plane wires are vertical. The distance between two adjacent wires (on the same plane) is 3 mm for all planes.

In LArTPCs the ionization pattern generated by a proton or a muon is a straight line referred to as a track [7]. It originates at the interaction point and extends to the point where the particle does one or several of the following: loses all its energy and comes to a stop; interacts again; decays; or exits the detector active volume. The most significant difference between a proton track and a muon track is in the energy deposition per unit length (dE/dx). The electromagnetic shower pattern generated by an electron ($E \geq 39$ MeV [8]) has a richer topology that is similar to a tree with many random branches. This is due to the electron losing energy to ionization as well as stochastically emitting photons. The emitted photons produce an electron-positron pair or Compton scatter to produce electrons. At this stage, the same processes occur again until all energy is deposited in the detector or the shower exceeds the detector boundaries. This cascade generates a pattern referred to as a shower [8]. The lower the electromagnetic shower energy is, the less branches are created and its topology becomes less distinct from a track. Correctly classifying the signature generated by charged particles in the detector as a shower or a track is a key ingredient of the DL-LEE analysis.

In the MicroBooNE detector, the data from the wires are retrieved as waveforms. As a first step, the waveforms are be subjected to signal processing that reconstructs the original ionization charge and zero-suppresses nonsignal regions (see [6]).

In the DL-LEE analysis, the data is represented as a set of three two-dimensional images (one for each wire plane), with wire number plotted along the x-axis and drift time plotted along the y-axis. The intensity of each “pixel,” measured in pixel intensity units (PIUs), gives a measure of the number of ionization electrons arriving at the corresponding location and time. Along the time axis, the waveform is integrated over six TPC time-ticks ($3 \mu\text{s}$). Thus the effective size of each pixel is 3.3 mm along the y-axis and 3 mm along the x-axis. After signal-processing, the resulting image will contain sparse regions of interest (ROIs) that are the charge-signals from the interaction. Pixels outside of these ROIs, are set to zero. To assure images from all wire planes are the same size, the images from the induction planes are padded with zero value pixels for wires 2401–3456 yielding a final images size of 3456×1008 .

Convolutional neural networks (CNNs) are the state of the art algorithms for solving many problems in image processing [9]. In recent years machine learning techniques in general, and specifically CNNs, have seen many applications in physics [10]. Particularly in the field of neutrino physics, many data analyses exploit the power of CNNs for various tasks such as event classification, background

rejection, energy reconstruction, and more [11–15]. As the MicroBooNE data can be represented by sets of images, it is natural to exploit the excellent performances of CNNs. Recently, an implementation of CNNs oriented at sparse data sets, named Submanifold Sparse Convolutional Networks (SSCNs), was proposed by the Facebook AI team [16,17] and has drawn much attention from several experiments in many experimental physics application. SSCNs were demonstrated to perform better than a dense CNN for the task of semantic segmentation on an open data set sample [18], as well as on background rejection using calibration data in a gas xenon TPC [14]. As the images produced by the MicroBooNE detector are very sparse macroscopically (after applying low frequency noise-filtering and signal-processing), but hold rich dense data in the vicinity of an interaction, the use of the SSCN adaptation, is appropriate. Moreover, as explained in Sec. II, an SSCN’s resource consumption scales linearly with the amount of data and thus is of much interest in future LArTPC detectors with a much larger data volume such as ICARUS [19] and DUNE [20] (for more examples of SSCN use in LArTPCs see [21–23]).

In this paper, we describe SparseSSNet (Sparse Semantic Segmentation Network), a deep-learning-based algorithm designed to distinguish showers from tracks in MicroBooNE. After applying a set of initial data selection criteria for reducing low energy backgrounds and tracks originating from cosmic muons, SparseSSNet is applied, replacing the previously used CNN [13]. The network is used for the task of semantic segmentation at the pixel level [24], before any physical entities are identified (e.g., interaction vertex, grouping energy depositions originating from the same particle, etc.). It is worth mentioning that although SparseSSNet is currently used only within the DL-LEE analysis, It is not tuned to that specific analysis and can be used in the search of any type of search in a LArTPC. This study was done in a Singularity [25] software container, [26] and the implementation of SparseSSNet is available on GitHub [27].

II. THE NETWORK

SparseSSNet treats each wire-plane separately, therefore three networks are trained one for each wire-plane. The architecture of the networks is similar however, the masking and the derived set of network weights are unique to each network. No data are shared between the different planes.

The main idea behind SparseSSNet is that, prior to training, masking is performed on the image to distinguish between important and nonimportant pixels, the masking is done on the feature vector (the pixel intensity this work). The input data therefore are reduced from $N \times \vec{f}$ to $N_{\text{th}} \times d \times \vec{f}$. N is the total number of pixels in the image and scales exponentially with the image dimension

($N = 3456 \times 1008$ for images produced by MicroBooNE). \vec{f} is the feature vector (the intensity of the pixel for the input layer in this analysis), d is the dimension of the image ($d = 2$ in this analysis), and N_{th} is the number of pixels passing selection criteria (masked). Using the sparse representation, the data stored, as well as the number of computations done for a convolutional layer, scales linearly with the number of pixels passing selection criteria instead of as a function of the total number of pixels.

SparseSSNet processes images using a sparse algebra as opposed to a dense one. Using SparseSSNet provides several benefits over using the dense CNN. The key benefit is the reduction in the time and memory consumption. The nature of the sparse representation drastically improves the time for processing a 512×512 image (this is the size of an image from the training sample see Sec. III) from ≈ 5 s to ≈ 0.5 s (The performance tests were done on an Intel core i7-8750H CPU 2.2 GHz). This improvement in the computation time reduces it below the time required of I/O which becomes the current bottleneck. In addition, as pixels of no interest (e.g., 0 intensity) are not saved, the memory consumption for inference is reduced from ≈ 6 GB to ≈ 1 GB. As a consequence, a full image can be inferred in a timely manner on a single CPU, unlike in the dense case where the images were cropped into ≈ 64 smaller images. Looking at the total CPU wall-time needed to analyze the current MicroBooNE open data set of $\approx 5 \times 10^{19}$ protons on target, a total of 195875 events (three planes each) requires ≈ 2267 days for the dense case and only ≈ 1.13 days in the sparse case (assuming a batch size of 1 as required by the dense case, due to CPU memory limitation). Due to these advantages, the image inference can be performed utilizing commonly available computing resources provided by numerous high throughput computing facilities.

Given the design of the MicroBooNE detector electronics, the pixel value (integrated counts over six TPC time ticks) distribution of minimum ionizing muons peaks at ≈ 40 PIU and the maximal pixel value of a muon would produce a pixel value of ≈ 220 PIU. In this analysis, all pixels with intensity smaller than 10 PIU or larger than 300 PIU are not included in the sparse representation. The lower value is set by the ability to distinguish signal from noise, while the higher removes unphysical noise originating in the data-processing and is set well above any expected value. This threshold reduces the number of relevant pixels to $\approx 0.5\%$ of the total pixels in an image.

Once a pixel is not included in the sparse representation, it is disregarded and not stored in the input data. In any convolution operation, an output pixel will be nonzero if and only if the central pixel of the receptive field is nonzero in the input feature map. Thus, the sparseness of the data is retained through a convolution, constraining only pixels which were nonzero in the input layer to be activated in hidden layers. While a dense convolution will spread

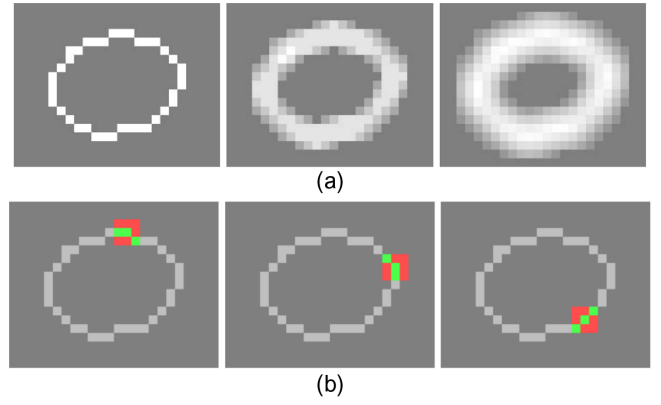


FIG. 1. (a) an example of an image being dilated after two dense convolution operations using a filter with size 3×3 , weights $1/9$, and stride 1. (b) a nondilated image using sparse convolutional layers, the green label represents pixels that are kept for consecutive layers and the red label represents pixels that would have acquired values in dense CNNs, but do not in sparseSSNet (image taken from [17]).

information to pixels which originally contained no information (this process is referred to as “image dilation”) the sparse convolution restricts changes to only those pixels which satisfy the selection criteria. This prevents image dilation (see Fig. 1) and improves the accuracy of the network.

The architecture of SparseSSNet is U-Res-Net (see Fig. 2) which is a hybrid of U-Net [28] and Res-Net [29]. This network consists of two parts: an encoder and a decoder. At the encoder, the image is downsampled using a stride = 2, and the network extracts features at various scales and hierarchical correlations. The number of downsampling steps is referred to as the depth of the network. The decoder upsamples the output of the encoder using transpose convolutions, and learns how to interpolate back to higher spatial resolution images, until reaching the original size. The feature map from the encoder is concatenated to the decoder feature map of the same size to help the decoder to restore the original image. Each block of convolutions (up or down sampling) is made of two convolutional layers followed by a batch normalization operation and a rectified linear unit (ReLU) function. Additional skip connections are added according to the Res-Net architecture.

SparseSSNet is constructed with 32 filters in the initial block and has a depth of five. The number of filters for a specific block is $i \times 32$, where i is the block’s depth (the input layer has a depth of 1). The filter size is 3×3 and a stride of two is used at each down-sample process,¹ decreasing the image size by a factor of two. A softmax¹

¹Softmax 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.

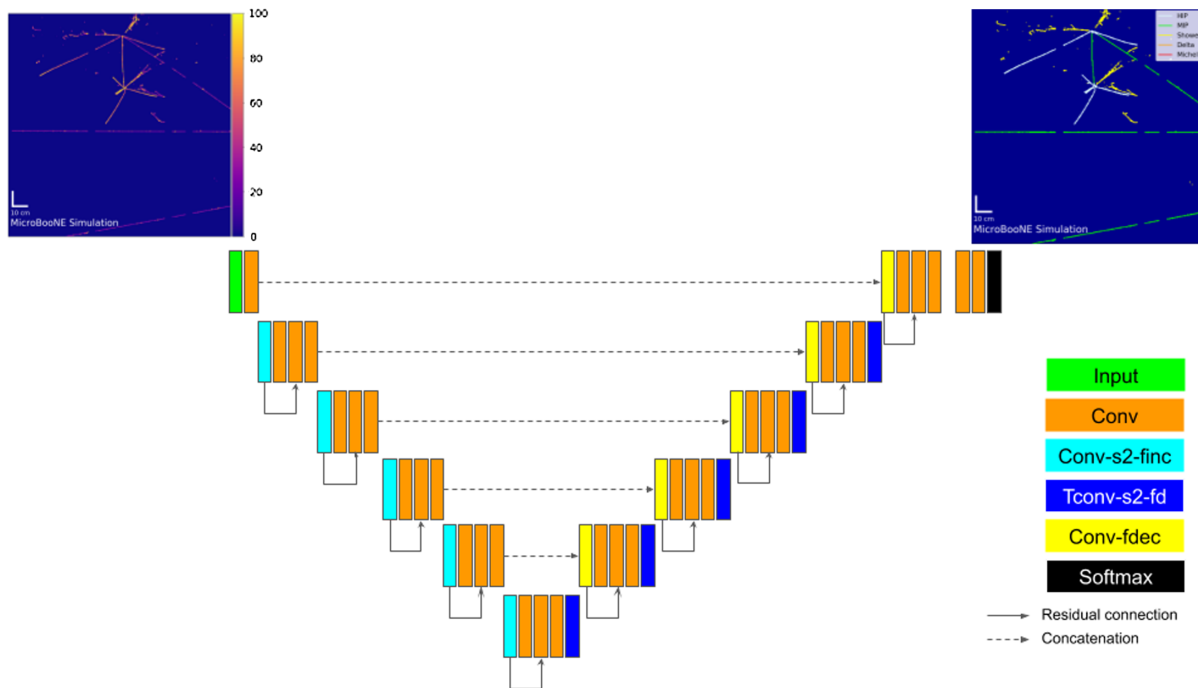


FIG. 2. SparseSSNet's architecture. Light blue boxes represent convolutional layers with stride two (decreasing the spatial size) and an increased number of filters. Orange boxes represent convolutional operations. Dark blue boxes represent transpose convolution with stride two (increasing the spatial size) and a decreased number of filters. Yellow boxes are convolutional layers with stride one and decreased number of filters. The depth of the network is defined as five since there are five down-sampling operations. The spatial size is constant along the horizontal axis.

classifier, an ADAM optimizer [30] and a cross-entropy loss function summed over all nonzero pixels are used. In addition, a pixel-weighting scheme is applied for preventing class imbalance (see Sec. IV).

The output of the network is a pixel-wise normalized five-dimensional probability vector \vec{p} (also referred to as scores); the predicted pixel label is then defined to be the class with the highest probability.

III. SIMULATED DATA SAMPLES

The data samples consist of images that contain neutrino interactions, as well as many particles from cosmic rays, since the MicroBooNE detector runs on the surface and uses a long integration time to collect ionization over the long drift length. The neutrino interactions consist of electrons (e), photons (γ), muons (μ), charged pions (π^\pm) and protons (p). Most of the cosmic ray particles are muons. These muons are higher in energy than the muons that are produced in neutrino interactions. Training is performed on simulated images consisting of the above mentioned particles in addition to higher energy muons I. An example of a simulated image with cosmic muons is shown in Fig. 3.

The training and test data samples consist of $\approx 120,000$ and $\approx 23,000$ simulated images respectively. For each image, pixel intensity and class labels (see III B) are

produced on which supervised learning is performed. For validation tests, several samples of $\approx 15,000$ neutrino interactions assuming different variations on the detector response model are used.

Pixels are associated with the simulated particles that contributed to their intensity. All pixels associated with a given particle are grouped and considered a cluster for that particle (for a shower all pixels are considered part of the original particle (electron or gamma) initiating the shower). For each event, particle propagation as well as detector effects are applied to derive the final input image. The data samples are produced using the LArSoft [31] v08_05_00_06 and UBOONECODE [32] v08_00_00_13a packages. The simulation of the wire response is performed by the Wire-Cell simulation code, common to several current LArTPC experiments [33,34].

A. Particle sample

SparseSSNet is trained to be sensitive only to outgoing charged particles traveling in the LAr and producing ionization electrons (translated into PIUs); thus, the simulation sample contains only randomly produced charged particles and gammas with no assumption on the interaction that produced them. For each image in the training and test samples, a random location in the detector is drawn from a uniform distribution. A random number $N(=\sum n_i)$

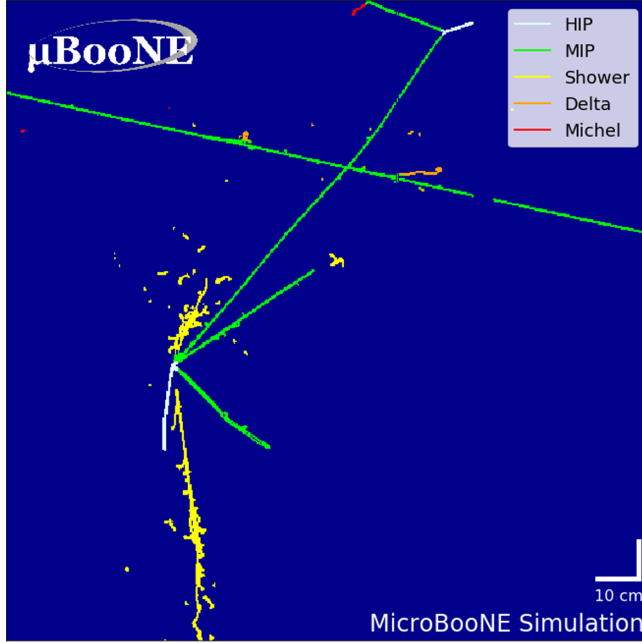


FIG. 3. An example of a simulated event projected on the collection plane, taken from the training sample. Multiple particles are generated at a specific location and propagated throughout the detector to mimic a neutrino interaction. In addition a higher energy muon is simulated to mimic cosmic muons.

of particles are generated at this location where N is the total number of particles and n_i is the number of particles from type i . N is drawn from a uniform distribution in the range of 1–6, whereas n_i is drawn from a uniform distribution in the ranges specified in Table I. The momentum vector direction of each particle is chosen from an isotropic distribution. Approximately 85% of the sample contains particles with kinetic energies (E_k) consistent with neutrino interactions within MicroBooNE. The energy range (E) for this sample for each particle is shown in Table I. A smaller sample ($\approx 15\%$) is generated with a different configuration targeted at low energy interactions where particle identification becomes more difficult. The momentum (P) range for each particle from the low energy (low E) sample is shown in Table I. The number of particles generated (N) and their multiplicity (n_i) remains the same. Finally, a random number of muons in the multiplicity

range of 5–10 and kinetic energy range of (5,000–20,000) MeV are generated in both samples to mimic cosmic rays.

B. Training labels

In the preparation of the samples, training labels are assigned to each pixel according to the particle contributing to their intensity. A total of five different labels are used. For the training and test sets, a pixel can be a sum of ionization electrons produced by two different charged particles reaching the same wire at the same time. Therefore, we follow a one-hot label scheme, i.e., a pixel can have only one label. In the case a pixel can be assigned to more than one label, the label assigned to it will be the highest according to the following order. This improves the performance of downstream analysis tasks such as identifying vertices [3].

1. *Heavily ionizing particles (HIP)*, produced by protons, typically manifest in a short, highly ionized track.
2. *Minimum ionizing particles (MIP)*, produced by muons and charged pions, typically manifest in a longer, fainter (lower dE/dx) track.
3. *Showers*, produced by electrons, positrons, and photons above a minimal energy, ≈ 39 MeV in liquid argon [8].
4. *Delta rays*, produced from ejected atomic electrons from a hard scattering of other charged particles, mainly muons.
5. *Michel electrons*, produced from a decay at rest of muons.

Within the current DL-LEE analysis, only two classes are used for particle ID, track and shower; thus, the previously mentioned five-classes are mapped into two new classes.

1. *Track*, either HIP or MIP labels.
2. *Shower*, either shower or delta ray or Michel electron labels.

Notice that the reclassification does not require new inferring; rather it is just a mapping of these original five labels to the newly defined two labels (i.e., if a pixel is classified by the network as a MIP in the DL-LEE analysis it will be considered as a track). We intend to exploit the full class feature set in future analyses.

TABLE I. The data sample particle content. For each particle type the multiplicity per event, the kinetic energy range for the full sample, and the momentum for the low energy (E) sample are 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.

Particle	e	γ	μ	π^+	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)

C. Full neutrino interaction sample

The neutrino simulation is performed in two stages in order to model the cosmic background in a more realistic manner. The first stage is simulating the neutrino interaction itself using the GENIE [35] v3_00_04 and GEANT4 [36] v4_10_3_03c software packages in addition to the LArSoft and UBOONECODE packages. The second is using a sample of beam-off data, taken from beam-off periods triggered on cosmic rays, and overlaying it on the neutrino interaction. As particles from the beam-off data sample are not from simulation, no labels can be assigned to them.

There are two Monte Carlo (MC) simulation samples used for the study. The full-BNB sample includes all types of neutrino interactions that are expected to occur in MicroBooNE. The intrinsic ν_e sample comprises events due to the electron-flavor neutrinos predicted to be in the flux, with all neutrino interaction types included. Both samples are overlayed with beam-off data.

SparseSSNet's predictions may vary depending on several factors, but the primary variation is from modeling different aspects of the detector response such as electric field, space-charge effects [37], wire response, etc. Therefore to assess systematic uncertainties (see Sec. V B) samples with variation in the detector response model are generated with $O(15,000)$ events per each of the eight variations explored. The sample with the nominal detector response is referred to as the central value simulation sample. These samples are used to verify the performance of SparseSSNet within the context of the DL-LEE analysis; therefore, only the two-class semantic segmentation is studied.

IV. PIXEL WEIGHTING

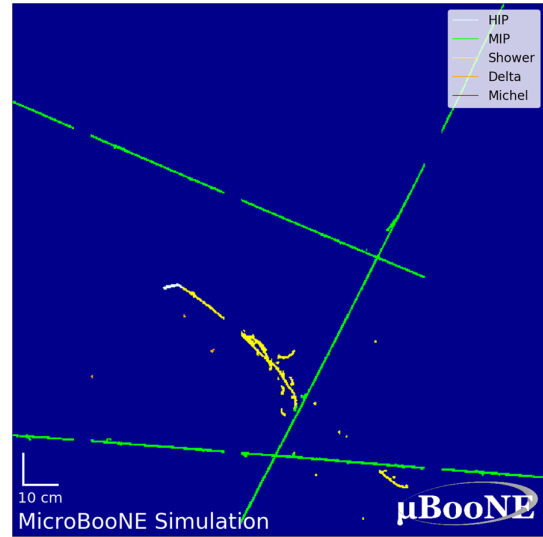
To prevent class imbalance [38], a case where one class dominates the loss function and the penalty for incorrect prediction of other classes is negligible, we apply a pixel weighting scheme, defining the loss function

$$\text{Loss} = \sum_i w_i \cdot (\vec{l}_i \cdot \log(\vec{P}_i)) \quad (1)$$

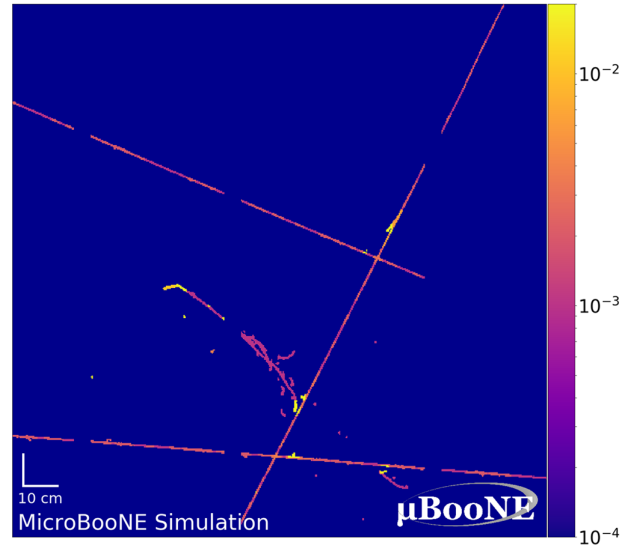
where w_i is the weight defined for each pixel, \vec{l}_i is the label vector of pixel i (e.g., (1,0,0,0,0) for a HIP) and \vec{P}_i is the scores vector for pixel i (e.g., (0.8,0.2,0,0,0) for 80% HIP and 20% MIP).

The sum of two types of weighting is assigned to each pixel: cluster weighting and vertex weighting (see Fig. 4).

- (i) *Cluster weighting*: large clusters contain many pixels. This makes it easier to correctly label them, as they contain more information. Moreover, labeling a large cluster correctly reduces the loss function by a significant amount (proportional to the number of pixels in the cluster). Therefore, small clusters should be treated with more care to prevent the loss function from being governed by one correctly labeled big cluster. We apply a cluster weight that



(a)



(b)

FIG. 4. An example of a simulated event projected on the collection plane, taken from the training sample. (a) The labels assigned for each pixel according to generated particles. (b) Pixel weighting: for each cluster a weight proportional to the inverse of its size is assigned in the range of $(0.02-2) \times 10^{-2}$. Notice that even though the shower has gaps in it (due to gammas) all pixels of the shower are associated with the original electron cluster for the purpose of cluster weighting. For crossing type pixels a weight of 2×10^{-2} is assigned.

is inversely proportional to the size of the cluster in the range of $(0.02-2) \times 10^{-2}$.

- (ii) *Vertex weighting*: pixels at the center of a cluster are easier to identify, as they cannot be confused by other pixels in their close vicinity. On the other hand, pixels near a cluster labeled as a “different class” are the most difficult to recognize and impact the vertex

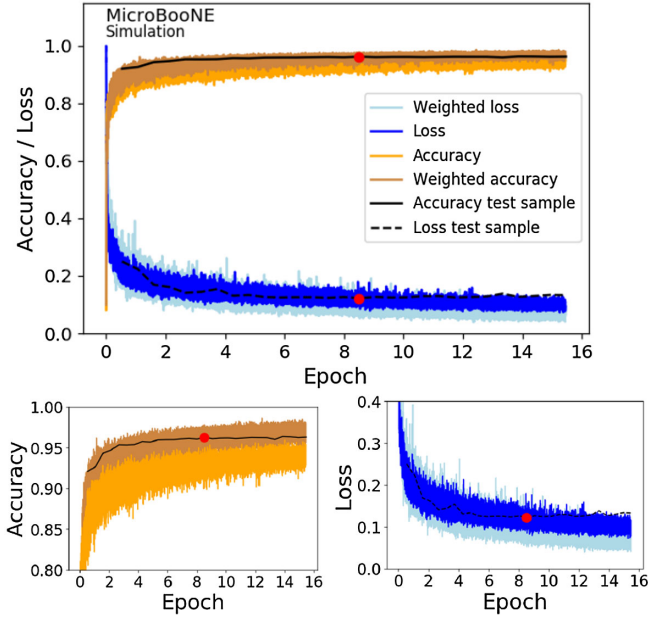


FIG. 5. Top: the accuracy and loss of the network on the training and inference data sets obtained from the collection plane. The accuracy before (orange) and after (light brown) applying weighting, the loss function normalized by the loss after the first iteration, before (blue) and after (light blue) applying weighting. The accuracy of the inference data sample is indicated in solid black and the loss on the test sample is indicated in dashed black. The selected network weights are indicated by the red dot (8.5 epochs). Bottom: plots are zoomed in.

reconstruction algorithm [3] (the next stage of the DL-LEE analysis) dramatically (e.g., near a vertex of a proton and a muon it is harder to distinguish exactly which pixel is associated with the proton and which with the muon). We apply a vertex weight of 0.02 to pixels within a three-pixel distance from a pixel from a different class, and vertex weight of zero to all other pixels.

V. RESULTS

A. Test sample

For each plane, SparseSSNet is trained for ≈ 15 epochs,² the chosen weights are obtained from ≈ 8.5 epochs of training to achieve the best performance without over fitting the network. Although the performance of the signal processing is plane dependent [34], the SparseSSNet results from all planes are similar (averaged over the entire phase-space mentioned in table I) and therefore we discuss only results obtained from the collection plane. We define the accuracy

²An epoch refers to one cycle through the entire data set. In this analysis, as images are selected randomly for each batch, an epoch refers to a cycle through a number of images equal to the sample size ($\approx 120,000$ images).

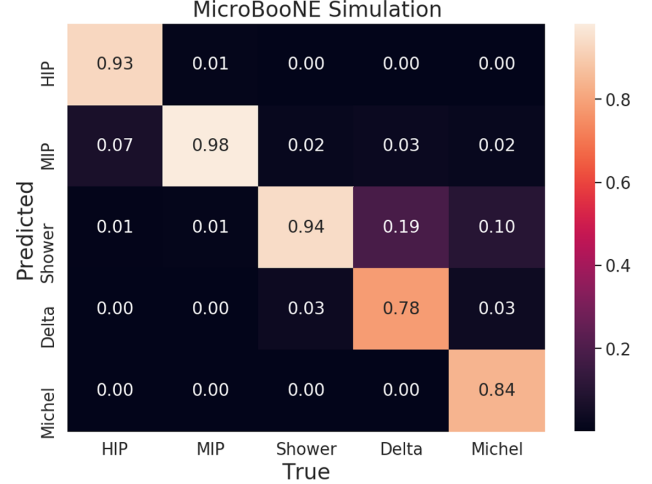


FIG. 6. Five-class confusion matrix obtained from the collection plane test sample. Each box represents the fraction of pixels that are from the class stated in the x-axis and predicted as class stated in y-axis from the test sample. The smallest number of pixels is $O(10^5)$ for Michel electrons. All other classes vary between 10^6 – 10^7 pixels.

as the number of correctly classified pixels with respect to nonzero pixels only.

The total accuracy obtained from the collection plane test sample is 96% for the final configuration of the network. The accuracy and the loss function obtained from the training sample both with and without weighting are shown in Fig. 5, along with the accuracy and loss obtained from the test sample. Applying the pixel weighting improves the total accuracy from 92% to 96%.

A better quantification of the performance with respect to each class is achieved by the confusion matrices. The five-class semantic segmentation confusion matrix obtained from the test sample for the collection plane is shown in Fig. 6.

The number of pixels obtained for each class is $O(10^5)$ for Michel electrons, and varies between 10^6 and 10^7 for other classes; thus statistical uncertainties are small. The matrices for the induction planes are fairly similar and are presented in the Appendix A.

TABLE II. sparsessnet's track and shower accuracy, for the test sample and the neutrino interaction central value simulation sample (both full-BNB and intrinsic ν_e). Results are obtained from the collection plane. The number of pixels associated with each class is $O(10^7)$ pixels except for the full-BNB shower which is $O(10^5)$. The drop in the shower accuracy for the neutrino interaction sample is explained in Sec. V B.

	Test	Intrinsic ν_e	Full-BNB
Track	0.992	0.992	0.998
Shower	0.996	0.859	0.823

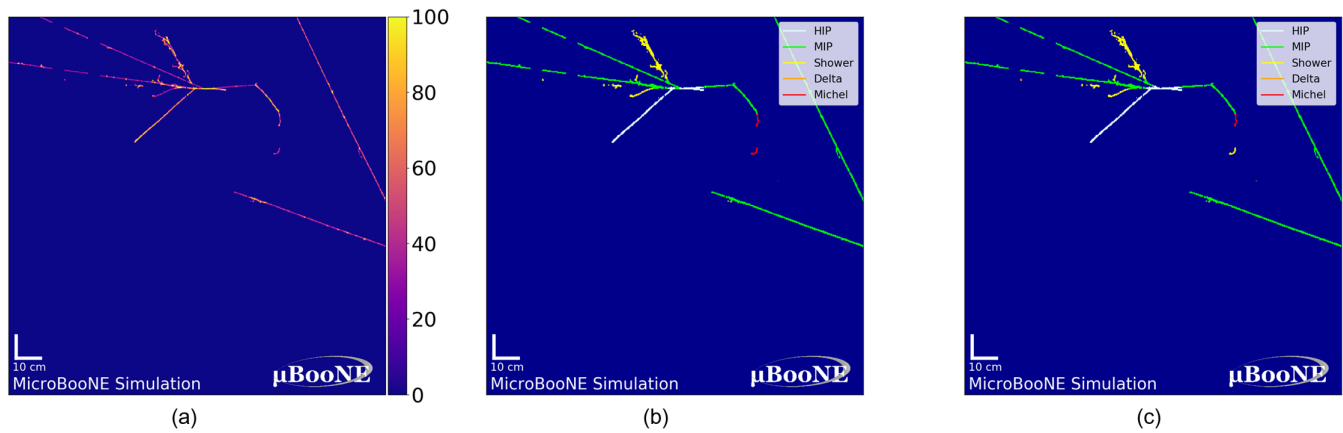


FIG. 7. An example of a simulated event from the test sample projected on the collection plane. (a) pixel intensity. (b) Truth label. (c) SparseSSNet predictions. All images are 512×512 pixels crops from a full detector simulation.

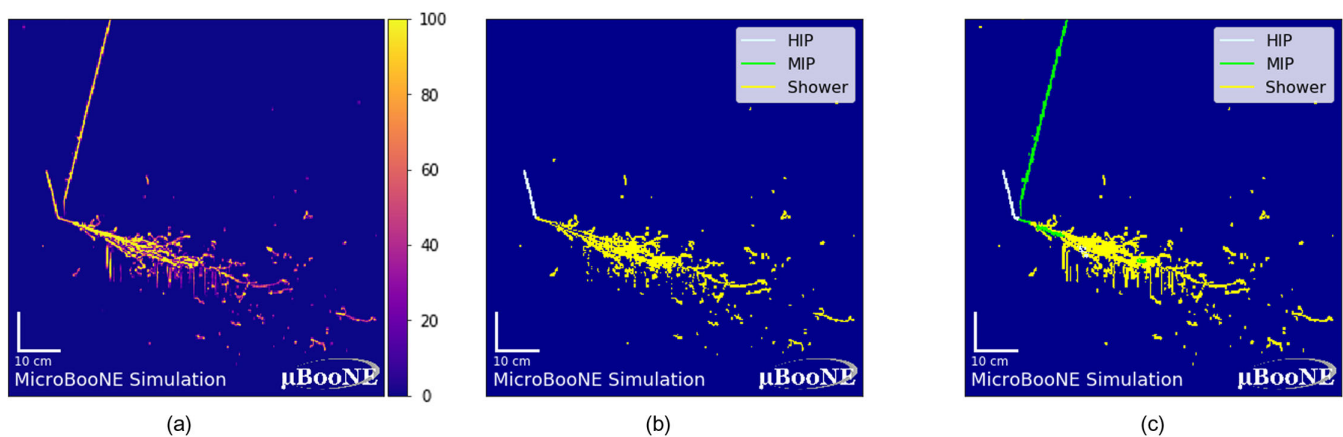


FIG. 8. An example of a simulated ν_e interaction projected on the first induction plane. This is a (300×300) pixels crop from the full detector image shown in Fig. 13. (a) Pixel intensity of interaction overlayed with cosmic rays. (b) The produced particles upon interaction, before overlaying cosmic rays. (c) SparseSSNet predictions. Notice that although some shower pixels are misclassified, the fraction of showerlike pixels is larger than 0.2 and therefore this interaction will be correctly classified as ν_e interaction (see Sec. V D).

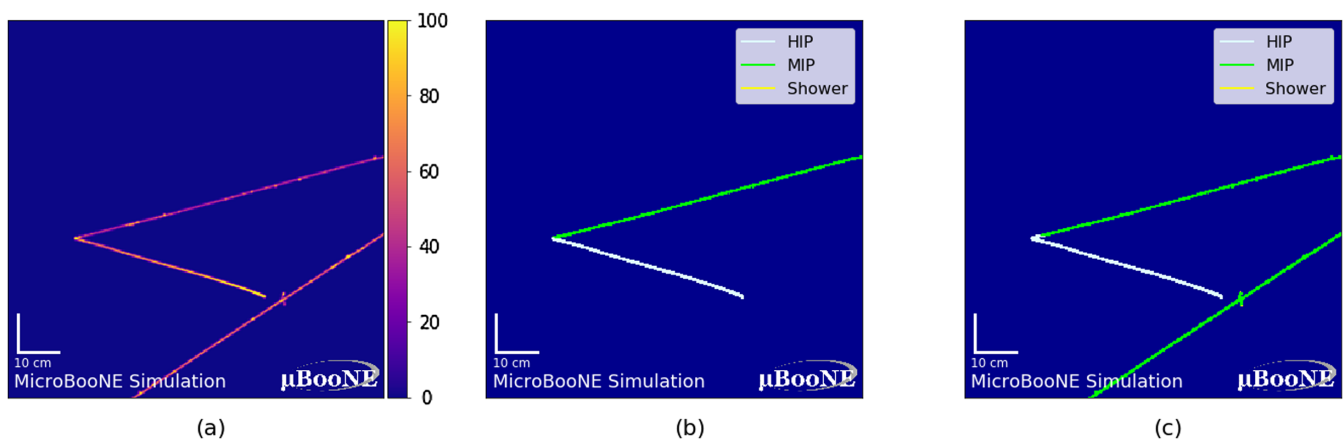


FIG. 9. An example of a simulated ν_μ interaction projected on the collection plane. This is a (300×300) pixels crop from the full detector image shown in Fig. 14. (a) Pixel intensity of interaction overlayed with cosmic rays. (b) The produced particles upon interaction, before overlaying cosmic rays. (c) SparseSSNet predictions.

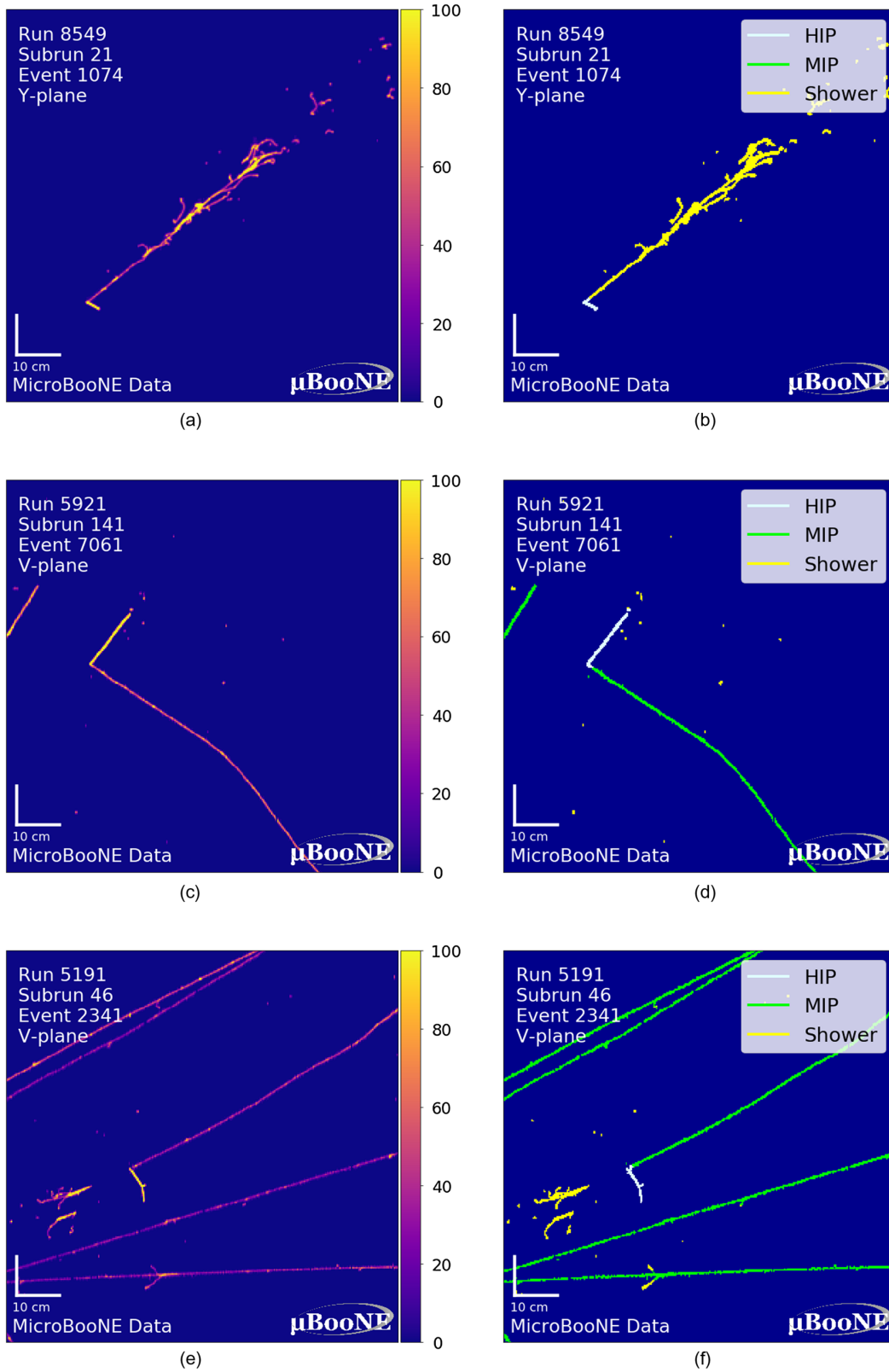


FIG. 10. Selected example from BNB data, image cropped to (300×300) pixels roughly around the interaction point. The left column is the pixel intensity (a,c,e) and the right column is SparseSSNet predictions. The top (a,b) event is an example of a $1e1p$ final state topology, the middle (c,d) is an example of a $1\mu1p$ final state topology, and the bottom (e,f) is an example of a $1\mu1p1\pi^0$ final state topology. These events show good performance of sparsessnet on data

In the DL-LEE analysis, only two classes are defined (see Sec. III B). The accuracy for each class for the two-class semantic segmentation obtained for the collection plane (Y) is shown in Table II. The results from the two induction planes are similar and are presented in Appendix A. The number of pixels obtained for each class is $>10^7$; as with the case of the five-class scheme, the statistical uncertainties are negligible. Comparing these results to the previous network used [13], yields an improvement in the shower accuracy from 95.9% to 99.6% and in the track accuracy from 97.4% to 99.2%. The accuracy improvement is due to the no-image-dilation effect, the lack of background class, and to fewer boundaries in the image (all zero and nonzero pixels)

An example of an event from the test sample is presented in Fig. 7. This display encapsulates the performance of the network and contains the pixel intensity, the truth level label, and the SparseSSNet's predictions.

B. Neutrino interaction sample

The neutrino interaction sample contains beam-off data which are not assigned with labels (see Sec. III C). Due to labeling priorities (see Sec. III B), when a cosmic muon crosses a shower, SparseSSNet is trained to predict the joint pixels as MIPs and not as showers. These pixels are considered as wrongly predicted which biases the network's shower accuracy. The accuracy calculated for the two-class semantic segmentation task for both the full-BNB and intrinsic ν_e samples are shown in Table II. The central value simulation samples are used to calculate these results.

The track accuracy is comparable to the one calculated from the test sample. The lower shower accuracy is attributed to the bias explained above. The track accuracy compared to previous analysis yields an improvement from 95.7% to 99.8% and 86.2% to 99.2% for the full-BNB (compared with the ν_μ sample) and intrinsic ν_e (compared with the ν_e sample), respectively. Notice that the previous analysis did not use beam-off data and did not see the bias in the shower accuracy. Examples of simulated ν_e and ν_μ from the neutrino interaction sample are shown in Fig. 8 and Fig. 9 respectively. These images are 300×300 pixels crops, roughly centered at the interaction point. For the full detector images see Appendix B. Each figure consists of the pixel intensity of the generated interaction with beam-off data, the generated interaction particles labels, and the SparseSSNet's predictions.

C. Beam data

As MicroBooNE has revealed part of the collected data, a final check of the performance of the network can be done on real BNB data. To quantitatively assess the performance of the network on BNB data, a more thorough analysis,

including reconstruction and selection needs to be performed which is beyond the scope of this paper. However, we have hand scanned many events (>1000 selected 111p events) and saw no strange bias. We present example event-displays showing SparseSSNet predictions. The three selected events are presented in Fig. 10, the left column is the PIU which is the input to the network and right column is SparseSSNet prediction. The events themselves present a $1e1p$ (a and b), $1\mu1p$ (c and d), and $1\mu1p1\pi^0$ (e and f) final states.

D. Robustness of results

The output of SparseSSNet is used within the DL-LEE analysis in two tasks: pixel classification and cluster classification. The samples with variation in the detector model (see III C) are used to verify that any mismodeling of the detector does not strongly affect these classification tasks.

The pixel classification is used for vertex finding, shower reconstruction and energy estimation, cluster classification, and more (ongoing work). It is performed by setting a threshold on the shower score $p_{\text{shower}} \geq 0.5$. The amount of misclassified shower/track pixels have a $\approx 0.5\%$ variation between all different simulation models.

The cluster classification is used to distinguish between a ν_e CCQE interaction ($1e1p$) and a ν_μ CCQE interaction ($1\mu1p$) by applying a selection criterion of $f_s \geq 0.2$ for at least one cluster, where f_s is the fraction of shower-like pixels in a cluster. The variations on shower clusters are $\approx 1\%$.

As discussed previously, the systematic errors are evaluated by running SparseSSNet on samples with detector variations. We find that, for the DL-LEE analysis, the uncertainty coming from SparseSSNet is negligible compared to uncertainties from the traditional algorithms used for track/shower reconstruction (ongoing work).

VI. SUMMARY

We have presented the performance of SparseSSNet in the task of semantic segmentation on simulated data from the MicroBooNE detector. The output of SparseSSNet plays an important role in many tasks such as vertex finding, shower reconstruction and energy reconstruction, and neutrino selection in the DL-LEE search in MicroBooNE, and is the first usage of SSCN in LArTPC data (beam-off) and realistic MC (neutrino interactions). The adaptation to sparse representation dramatically improves the inference time from ≈ 5 s to ≈ 0.5 s as well as the memory usage from ≈ 5 GB to ≈ 1 GB. In addition, there is an improvement in the accuracy of the test sample due to no-image-dilation preserving the sparsity and locality of the information. The current analysis uses only two classes (track and

shower), however the network produces five-class segmentation which we plan to exploit in future analyses.

The method that we have described here is transferable to other LArTPC detectors. This includes ICARUS, SBND, that are about to begin running on the same neutrino beamline as MicroBooNE, as well as for the DUNE experiment.

ACKNOWLEDGEMENT

This document was prepared by the MicroBooNE collaboration using the resources of the Fermi National Accelerator Laboratory (Fermilab), a U.S. Department of Energy, Office of Science, HEP User Facility. Fermilab is managed by Fermi Research Alliance, LLC (FRA), acting under Contract No. DE-AC02-07CH11359. MicroBooNE is supported by the following: the U.S. Department of Energy, Office of Science, Offices of High Energy Physics and Nuclear Physics; the U.S. National Science Foundation; the Swiss National Science Foundation; the Science and Technology Facilities Council (STFC), part of the United Kingdom Research and Innovation; the European Union's Horizon 2020 Marie Skłodowska-Curie Actions; and The Royal Society (United Kingdom). Additional support for the laser calibration system and cosmic ray tagger was provided by the Albert Einstein Center for Fundamental Physics, Bern, Switzerland.

APPENDIX A: RESULTS FROM INDUCTION PLANES

The five-class semantic segmentation confusion matrices produced from the data samples from the U plane (Fig. 11)

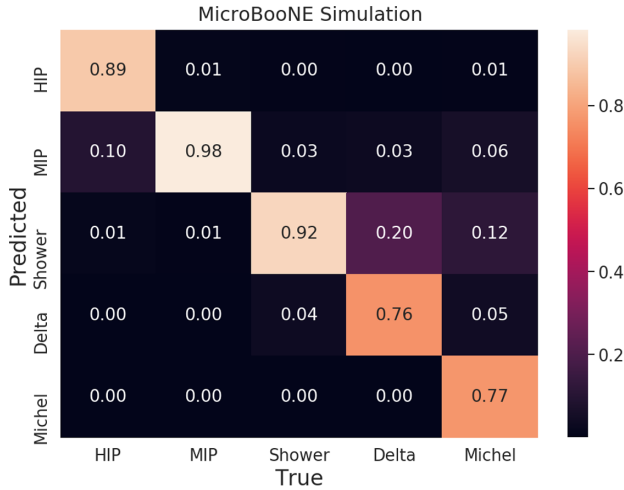


FIG. 11. Five-class confusion matrices for the first induction plane test sample. Each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y-axis. The smallest number of pixels is $O(10^5)$ for Michel electrons. All other classes vary between 10^6 – 10^7 pixels.

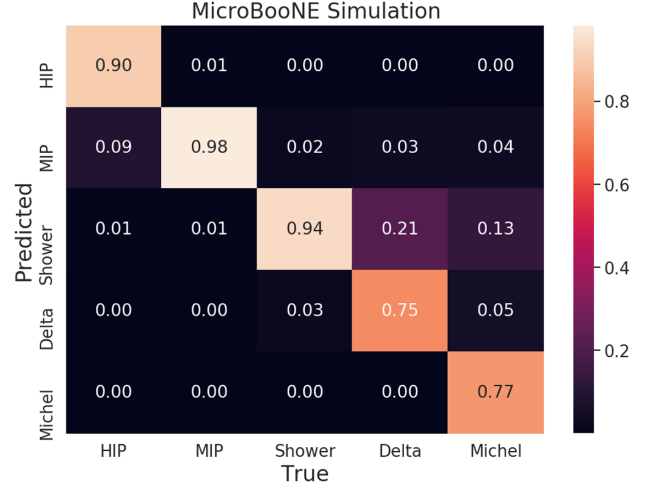


FIG. 12. Five-class confusion matrices for the second induction plane test sample. Each box represents the fraction of pixels which are from the class stated on x-axis and predicted as class stated in y-axis. The smallest number of pixels is $O(10^5)$ for Michel electrons. All other classes vary between 10^6 – 10^7 pixels.

and V plane (Fig. 12) are similar to the one presented in Fig. 6, and are presented here for completeness. Notice that these matrices are obtained from different planes which yields different images (though coming from same interactions) and different networks; this can explain the small accuracy differences between the planes. The number of pixels obtained for each class is similar to the collection plane and is $O(10^5)$ for Michel electrons, and varies between 10^6 and 10^7 for other classes; thus statistical uncertainties are small.

The accuracy for each class for the two-class semantic segmentation task obtained from the two induction planes on all simulation samples are similar to the those presented in Sec. II and presented in Table III for completeness. The drop in the shower accuracy for the full-BNB and intrinsic ν_e samples is explained in Sec. V B.

TABLE III. SparseSSNet track and shower accuracy for the test sample and the neutrino interaction central value simulation samples (both full-BNB and intrinsic ν_e). The results are obtained from the two induction planes. The number of pixels associated with each class is $O(10^7)$ pixels except for the full-BNB shower which is $O(10^5)$. The drop in the shower accuracy for the neutrino interaction sample is explained in V B.

	Test		Intrinsic ν_e		Full-BNB	
	U	V	U	V	U	V
Track	0.988	0.990	0.990	0.989	0.997	0.998
Shower	0.996	0.994	0.823	0.858	0.809	0.821

APPENDIX B: NEUTRINO INTERACTION EVENT DISPLAYS

Event displays of ν_e (Fig. 13) and ν_μ (Fig. 14) interactions from the full detector (3456×1008 pixels) of the events shown in Sec. V B. Notice that induction planes (U & V) contains only 2400 wires; hence, to keep the size of the images identical, an additional 1056 ($2400-3456$) are added but are empty. One can also see the sparsity of events from these figures.

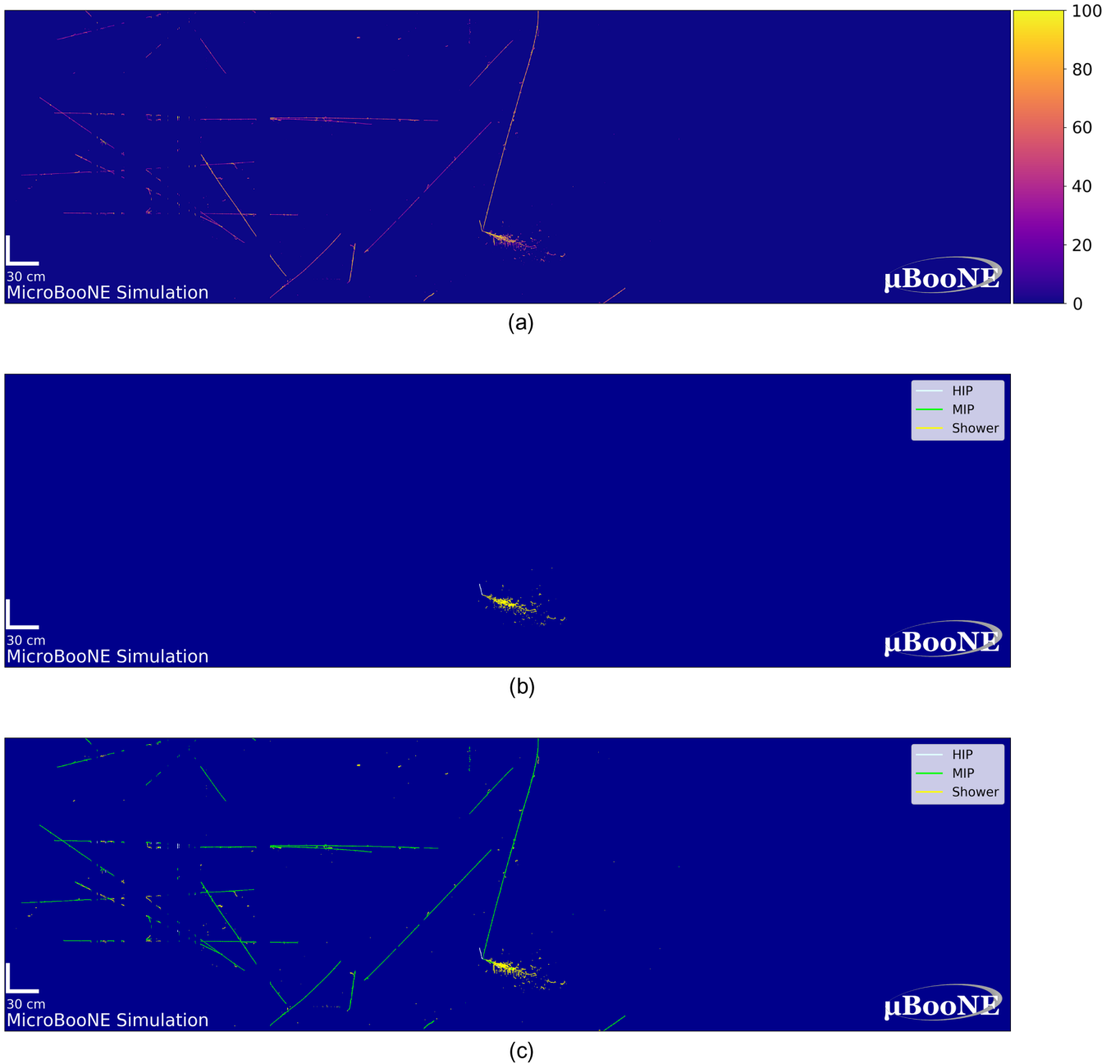


FIG. 13. An example of a simulated ν_e interaction projected on the first induction plane. This is the full detector image of the ν_e interaction shown in Fig. 8. (a) Pixel intensity of interaction overlaid with cosmic rays. (b) The label assigned to the simulated neutrino interaction. As cosmic rays are taken from beam-off data they are not assigned with labels. (c) SparseSSNet's predictions.

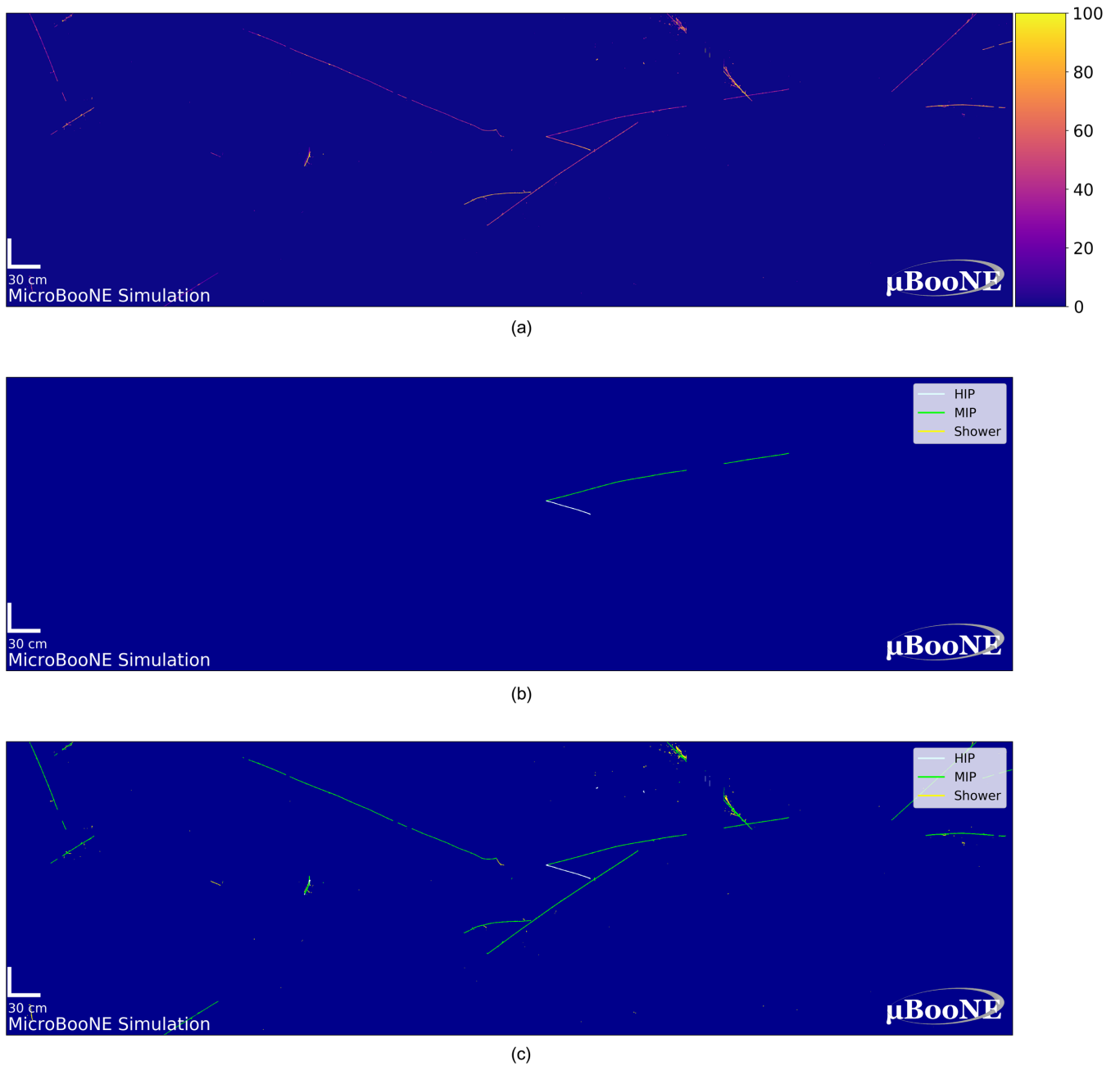


FIG. 14. An example of a simulated ν_μ interaction projected on the first induction plane. This is the full detector image of the ν_e interaction shown in Fig. 9. (a) Pixel intensity of interaction overlaid with cosmic rays. (b) The label assigned to the simulated neutrino interaction. As cosmic rays are taken from beam-off data they are not assigned with labels. (c) SparseSSNet's predictions.

-
- [1] A. Aguilar-Arevalo *et al.* (MiniBooNE Collaboration), *Phys. Rev. Lett.* **121**, 221801 (2018).
 - [2] P. Abratenko *et al.* (MicroBooNE Collaboration), [arXiv:2010.08653](#).
 - [3] P. Abratenko *et al.* (MicroBooNE Collaboration), *J. Instrum.* **16**, P02017 (2021).
 - [4] M. Antonello *et al.* (MicroBooNE, LAr1-ND, and ICARUS-WA104 Collaborations), [arXiv:1503.01520](#).

- [5] A. Aguilar-Arevalo *et al.* (MiniBooNE Collaboration), *Phys. Rev. D* **79**, 072002 (2009).
- [6] R. Acciarri *et al.* (MicroBooNE Collaboration), *J. Instrum.* **12**, P08003 (2017).
- [7] P. Abratenko *et al.* (MicroBooNE Collaboration), *Phys. Rev. Lett.* **123**, 131801 (2019).
- [8] C. Adams *et al.* (MicroBooNE Collaboration), *J. Instrum.* **15**, P02007 (2020).
- [9] Y. LeCun, Y. Bengio, and G. Hinton, *Nature (London)* **521**, 436 (2015).
- [10] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Rev. Mod. Phys.* **91**, 045002 (2019).
- [11] A. Aurisano, A. Radovic, D. Rocco, A. Himmel, M. Messier, E. Niner, G. Pawloski, F. Psihas, A. Sousa, and P. Vahle, *J. Instrum.* **11**, P09001 (2016).
- [12] R. Acciarri *et al.* (MicroBooNE Collaboration), *J. Instrum.* **12**, P03011 (2017).
- [13] C. Adams *et al.* (MicroBooNE Collaboration), *Phys. Rev. D* **99**, 092001 (2019).
- [14] M. Kekic *et al.* (NEXT Collaboration), *J. High Energy Phys.* **01** (2021) 189.
- [15] S. Delaquis *et al.* (EXO Collaboration), *J. Instrum.* **13**, P08023 (2018).
- [16] B. Graham and L. van der Maaten, [arXiv:1706.01307](https://arxiv.org/abs/1706.01307).
- [17] B. Graham, M. Engelcke, and L. van der Maaten, [arXiv:1711.10275](https://arxiv.org/abs/1711.10275).
- [18] L. Domin and K. Terao (DeepLearnPhysics Collaboration), *Phys. Rev. D* **102**, 012005 (2020).
- [19] S. Amerio *et al.* (ICARUS Collaboration), *Nucl. Instrum. Methods Phys. Res., Sect. A* **527**, 329 (2004).
- [20] R. Acciarri *et al.* (DUNE Collaboration), [arXiv:1601.02984](https://arxiv.org/abs/1601.02984).
- [21] L. Dominé, P. C. de Soux, F. Drielsma, D. H. Koh, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher, [arXiv:2006.14745](https://arxiv.org/abs/2006.14745).
- [22] D. H. Koh, P. C. de Soux, L. Domin, F. Drielsma, R. Itay, Q. Lin, K. Terao, K. V. Tsang, and T. L. Usher (DeepLearnPhysics Collaboration), [arXiv:2007.03083](https://arxiv.org/abs/2007.03083) [Phys. Rev. D (to be published)].
- [23] C. Adams, M. Del Tutto, J. Asaadi, M. Bernstein, E. Church, R. Guenette, J. M. Rojas, H. Sullivan, and A. Tripathi, *J. Instrum.* **15**, P04009 (2020).
- [24] J. Long, E. Shelhamer, and T. Darrell, [arXiv:1411.4038](https://arxiv.org/abs/1411.4038).
- [25] K. G. Sochat, C. J. Prybol, G. M. Kurtzer, *PLoS One* **12**, e0188511 (2017).
- [26] <https://singularity-hub.org/containers/6555>
- [27] <https://github.com/ranitay/SparseSSNet>
- [28] O. Ronneberger, P. Fischer, and T. Brox, in *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015*, editor edited by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi (Springer International Publishing, Cham, 2015), pp. 234–241.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.
- [30] D. P. Kingma and J. Ba, [arXiv:1412.6980](https://arxiv.org/abs/1412.6980).
- [31] E. Snider and G. Petrillo, *J. Phys. Conf. Ser.* **898**, 042057 (2017).
- [32] <https://cdcvs.fnal.gov/redmine/projects/uboonecode>.
- [33] C. Adams *et al.* (MicroBooNE Collaboration), *J. Instrum.* **13**, P07007 (2018).
- [34] C. Adams *et al.* (MicroBooNE Collaboration), *J. Instrum.* **13**, P07006 (2018).
- [35] C. Andreopoulos, C. Barry, S. Dytman, H. Gallagher, T. Golan, R. Hatcher, G. Perdue, and J. Yarba, [arXiv:1510.05494](https://arxiv.org/abs/1510.05494).
- [36] S. Agostinelli *et al.* (GEANT4 Collaboration), *Nucl. Instrum. Methods Phys. Res., Sect. A* **506**, 250 (2003).
- [37] P. Abratenko *et al.* (MicroBooNE Collaboration), *J. Instrum.* **15**, P12037 (2020).
- [38] N. Japkowicz and S. Stephen, *Intelligent Data Anal.* **6**, 429 (2002).