Scalable, Proposal-free Instance Segmentation Network for Liquid Argon Time Projection Chambers

Dae Heun Koh*, Kazuhiro Terao
*koh0207@stanford.edu

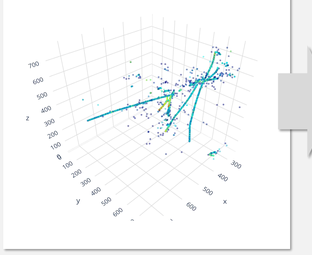
Stanford University

SLAC

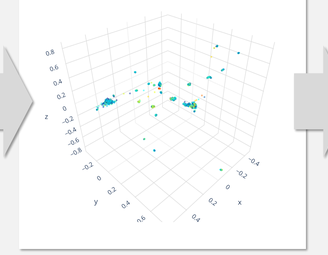
I. Introduction

We introduce a fast and scalable deep learning algorithm for particle clustering in liquid argon time projection chamber data using sparse convolutional neural networks. Particle clustering refers to the task of grouping 3D image voxels into different particle instances that may share the same semantic label (particle type). Building on previous works on sparse convolutional neural networks (SCNNs) and proposal free instance segmentation, we build an end-to-end trainable instance segmentation network that learns an embedding of the image voxels to perform point cloud clustering in a transformed space, and apply it to liquid argon time projection chamber data.

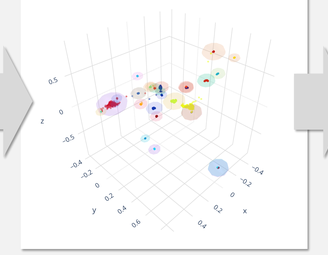
1. INPUT 3D IMAGE



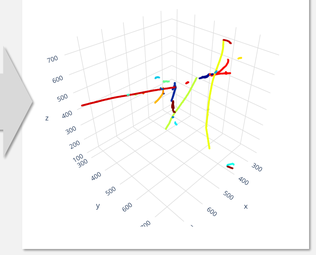
2. GENERATE EMBEDDINGS



3. POST-PROCESSING



4. CLUSTER PARTICLES



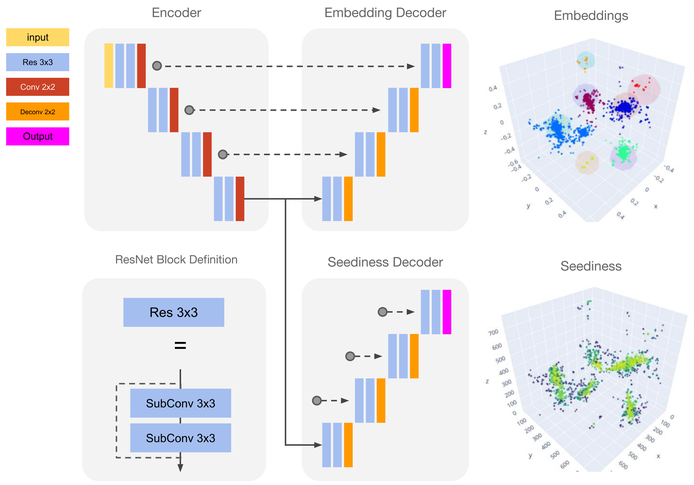
II. Network Architecture

Our architecture design closely follows previously studied instance segmentation techniques in computer vision [1, 2].

- **UResNet (ref. Poster # 373)**: Convolutional neural network architecture based on submanifold sparse convolutions [3, 4]. For particle clustering, we use UResNet with a shared encoder and two decoding branches.
- **Embedding**: pixels that belong to same particle instances are grouped together
- **Seediness**: trained to predict a score between 0 and 1 indicating proximity to the embedding space centroid of each cluster of pixels.

During inference, the particle labels are generated in the following way:

1. Find the point with the highest seediness score → likely candidate for centroid μ_k.
2. Define a kernel map $p(\cdot\,; \mu_k, \sigma_k)$ and compute scores p_{ik} for all pixels with a given pixel type.
3. Threshold via a predetermined score value p_0 to assign voxels to the centroid μ_k. Repeat until all pixels are clustered into different particle instances.



ALGORITHM DESCRIPTION & LOSS DEFINITIONS

We include a brief summary of the key loss definitions presented in [1] and [2] that allows us to train the instance segmentation network:

1. Instance Kernel Map

Given a particle instance C_k and an embedded point $f_{emb}(x_i)$, we compute the instance centroid μ_k and the associated margin value σ_k:

$$\mu_k = \frac{1}{|C_k|}\sum_{i\in C_k} f_{emb}(x_i), \quad \sigma_k = \frac{1}{|C_k|}\sum_{i\in C_k}\sigma(x_i)$$

This defines a gaussian kernel map $p(\cdot\,; \mu_k, \sigma_k)$ defined over the set of all pixels x_i in a given image:

$$p_{ik} \equiv p(x_i; \mu_k, \sigma_k) = \exp\left[-\frac{\|f_{emb}(x_i) - \mu_k\|^2}{2\sigma_k^2}\right]$$

One can interpret p_{ik} as the probability of pixel x_i belonging to particle instance C_k. We parametrize both the embedding function $f_{emb}(x_i)$ and the margin map $\sigma(x_i)$ with neural networks, optimizing the binary classification loss such as cross entropy:

2. Embedding Loss†

$$\mathcal{L}_{BCE}^{(k)} = -\frac{1}{N}\sum_{i=1}^{N}[y_{ik}\log(p_{ik}) + (1-y_{ik})\log(1-p_{ik})]$$

3. Centroid Separation Loss

$$\mathcal{L}_{inter} = \frac{1}{K(K-1)}\sum_{i<j}[2\delta_d - \|\mu_i - \mu_j\|]_+^2$$

3. Seediness Loss

$$\mathcal{L}_{seed}^{(k)} := \frac{1}{|C_k|}\sum_{i\in C_k}\|s_i - p(x_i; \mu_k, \sigma_k)\|^2$$

The seediness is a predicted measure of a given pixel embeddings' proximity to centroid μ_k. During inference, one begins with a highest seediness point and cluster pixels sequentially by thresholding on p_{ik} until all pixels are clustered.

†: In practice, we follow [1] and use the Lovasz Hinge binary classification loss instead of binary cross entropy.

DATASET DESCRIPTION

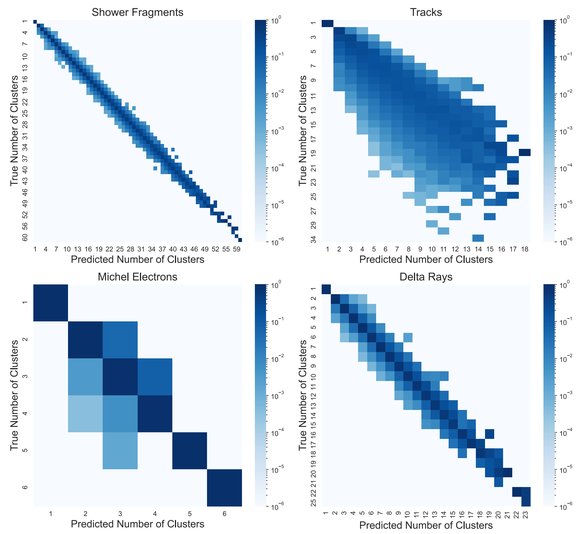
- **80k event training set and 20k event test set.**
- **Total number of clusters (particles) in test set: ~500k**

	Average # per Event	Voxel Counts	Percentage # Particles
Shower Fragments	15 ± 9	21.4M	56.9%
Tracks	6 ± 3	47.7M	24.7%
Michel	1 ± 1	0.793M	3.0%
Delta Rays	5 ± 3	1.11M	15.4%

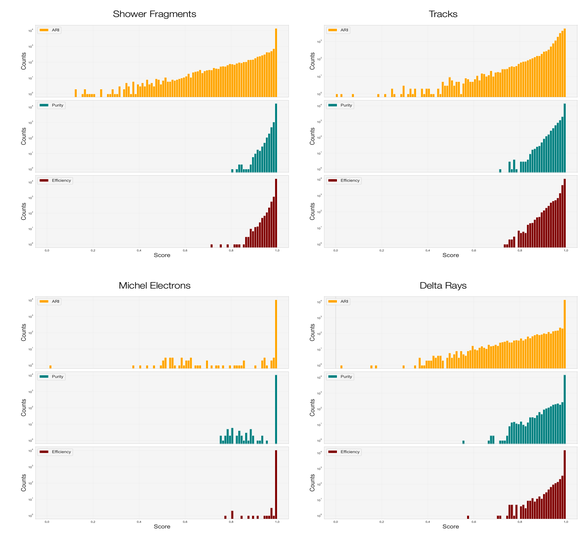
III. Results

Particle clustering is done per semantic class using ground truth semantic labels; we mask a given image according to voxel type labels (tracks, shower fragments, delta rays, and Michel electrons) and then run post-processing within each class. Given the LArTPC semantic segmentation performance reported in previous study [5], this apparent limitation is far from being overly optimistic. We demonstrate that the proposed algorithm has an average per-event inference time of 0.73 seconds on a NVIDIA V100 GPU, with an average adjusted rand index (ARI) score of 0.973, 99.3% purity and 99.3% efficiency.

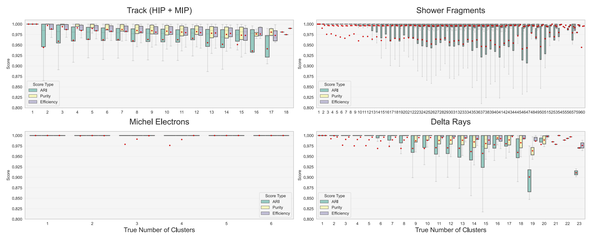
PREDICTED NUMBER OF PARTICLES



DISTRIBUTION OF ACCURACY METRICS



DEPENDENCE ON TRUE NUMBER OF PARTICLES



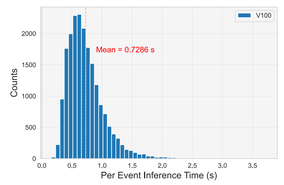
RESOURCE USAGE

Training (V100, Batch Size = 32)

- Training Memory Usage: 10.87GB
- Average time for 1 iteration: 9.82 s

Inference

- Average prediction time per event: 0.7286 s



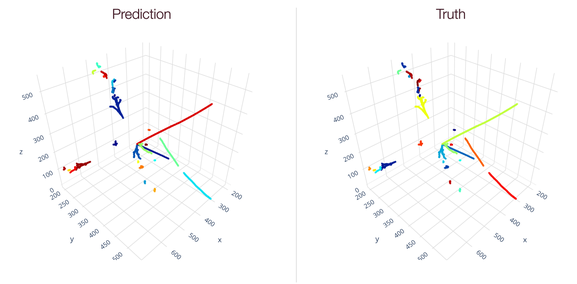
SUMMARY

- Evaluated on approx. 20k event test set
- ARI (Adjusted Rand Index) = **0.973**
- Purity = **0.993**
- Efficiency = **0.993**
- SBD (Symmetric Best Dice) = **0.903**

DIVISION BY CLASS

	Purity	Efficiency	ARI	SBD
Shower Fragments	0.997	0.996	0.968	0.978
Tracks	0.988	0.983	0.961	0.717
Michel	0.999	1.000	0.998	0.998
Delta Rays	0.992	0.996	0.978	0.982

EXAMPLE PREDICTION



IV. References

[1] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool, Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2019) pp. 8837–8845.
[2] B. De Brabandere, D. Neven, and L. Van Gool, Semantic instance segmentation for autonomous driving, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops* (2017).
[3] B. Graham and L. van der Maaten, Submanifold sparse convolutional networks, arXiv preprint arXiv:1706.01307 (2017).
[4] B. Graham, M. Engelcke, and L. van der Maaten, 3d semantic segmentation with submanifold sparse convolutional networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018) pp.9224–9232.
[5] L. Domine and K. Terao, Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data (2019), arxiv:1903.05663.