Machine Learning & Data Processing for Beam Profile Measurements

Javier Martínez Samblas, Manuel Gonzalez-Berges

Contributions by:
Glenn Anta, Clara Marie Fleisig, Verena Kain, Mark Mclean, Hampus Sandberg, James Storey

07/10/2023

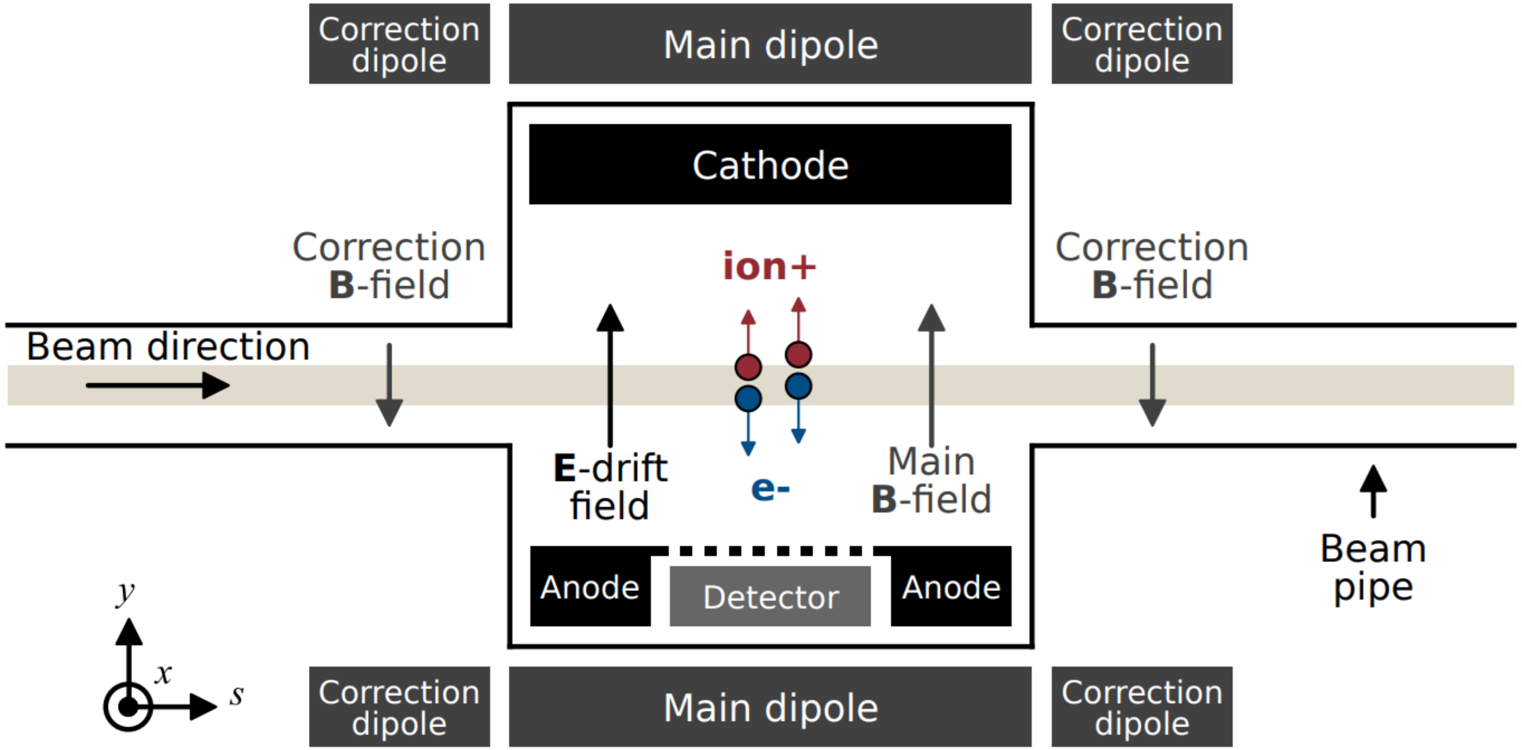
Outline

1. Introduction
2. Noise Removal
3. Extra Data Processing
4. Preliminary Results
5. Conclusions & Future Work

1. Introduction

BGI: Beam Gas Ionization

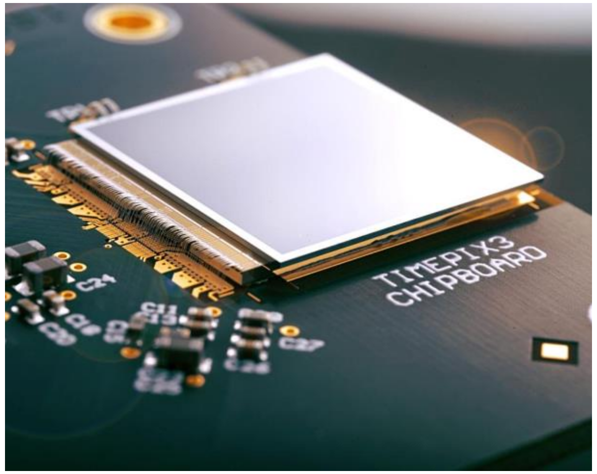
Motivation: “To obtain non-invasive beam profile measurements.”



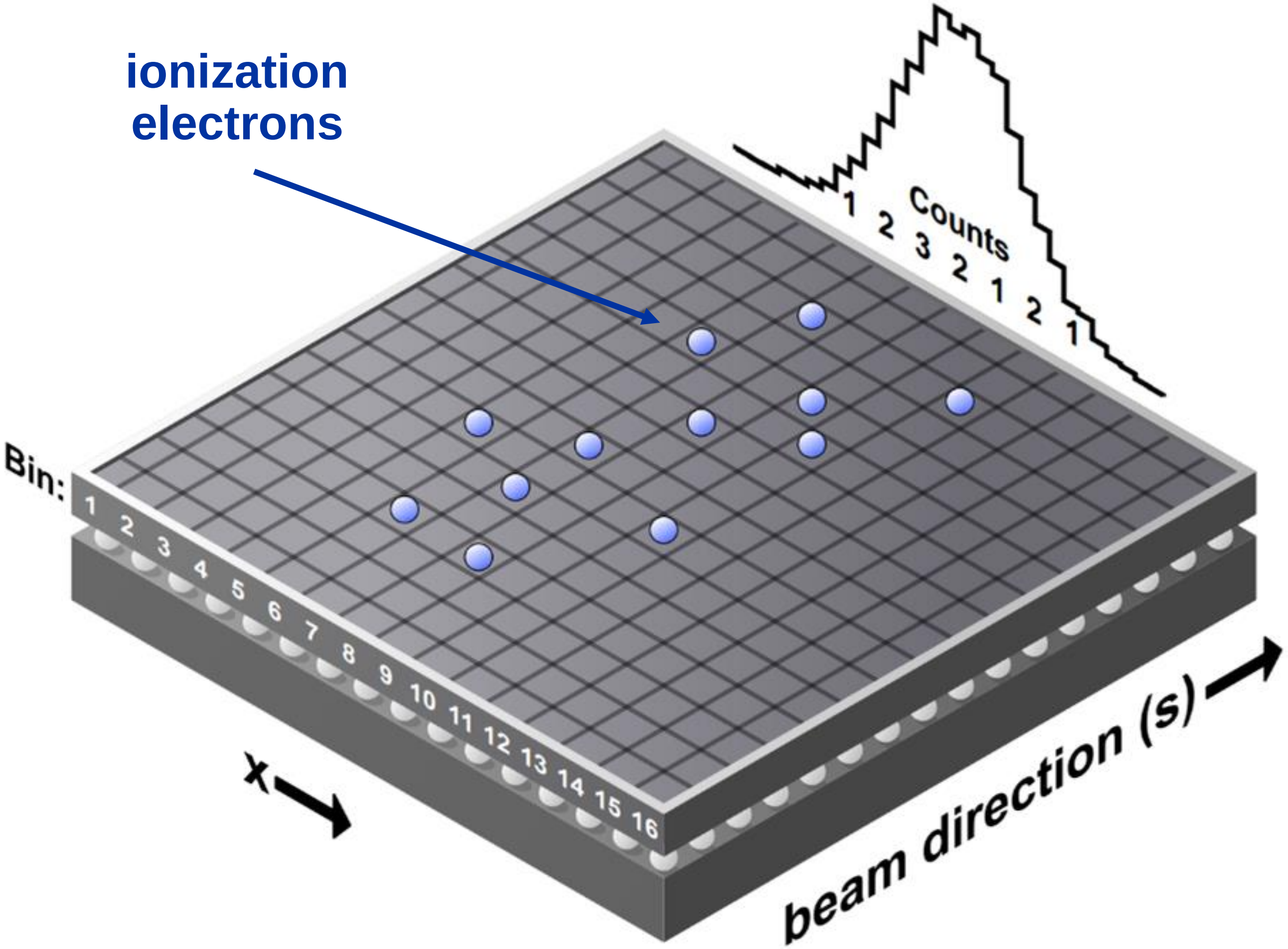
Source: Commissioning of Timepix3 Based Beam Gas Ionisation Profile Monitors for the CERN Proton Synchrotron (Swann Levasseur)

The Detector

Timepix3



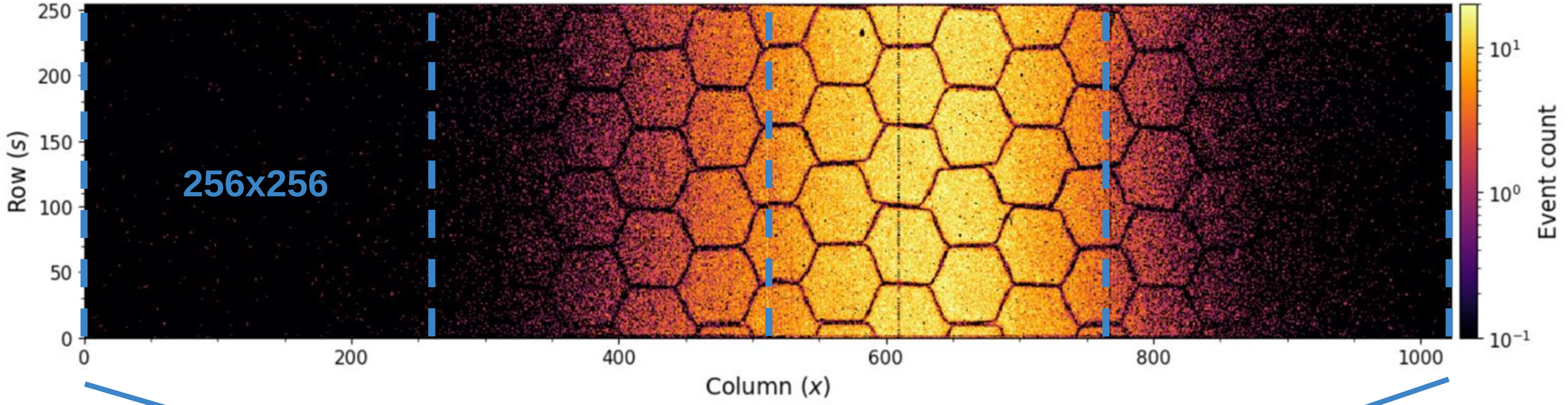
256x256 px
14x14 mm



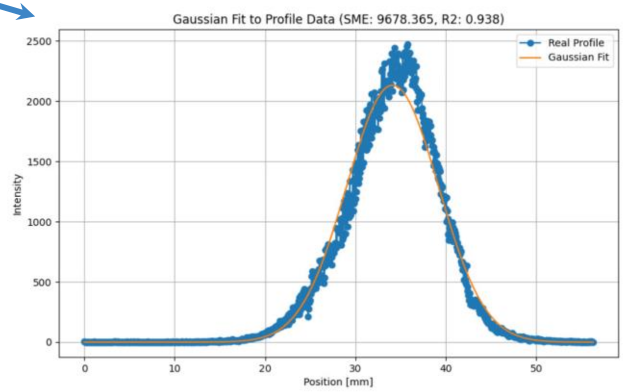
Source: Measuring the Beam Profile by Counting Ionization Electrons (Hampus Sandberg)

Beam Profile

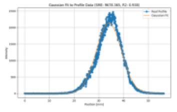
All Events



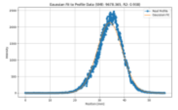
Sum (project) columns to obtain the beam profile



BGIH (horizontal plane)



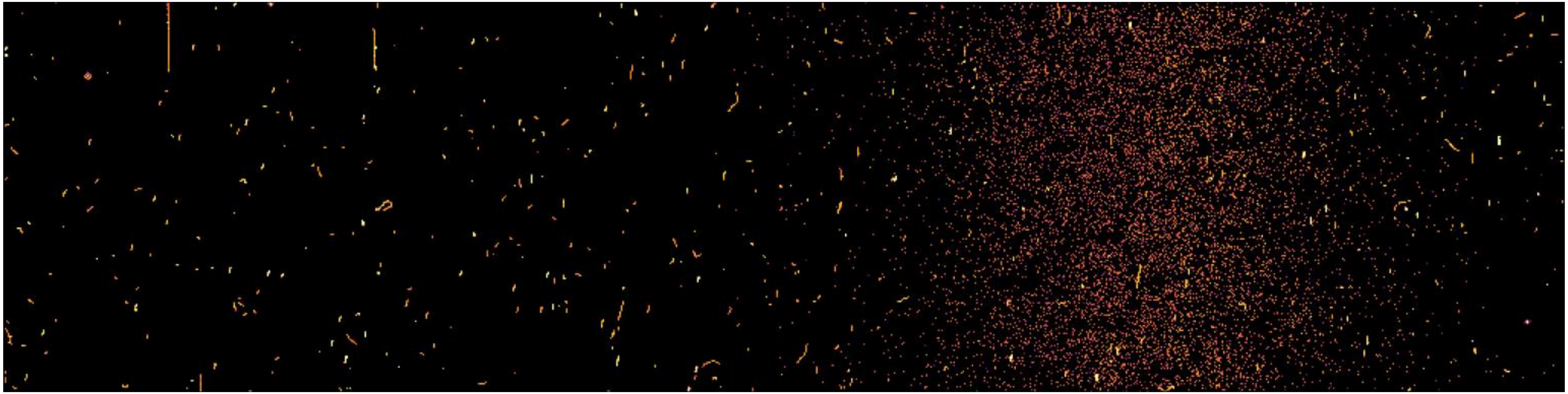
BGIV (vertical plane)



Acquisition Over Time

Acquisitions can actually be considered as a sequence of frames (video) that can be merged based on some predefined **integration time** 🎥

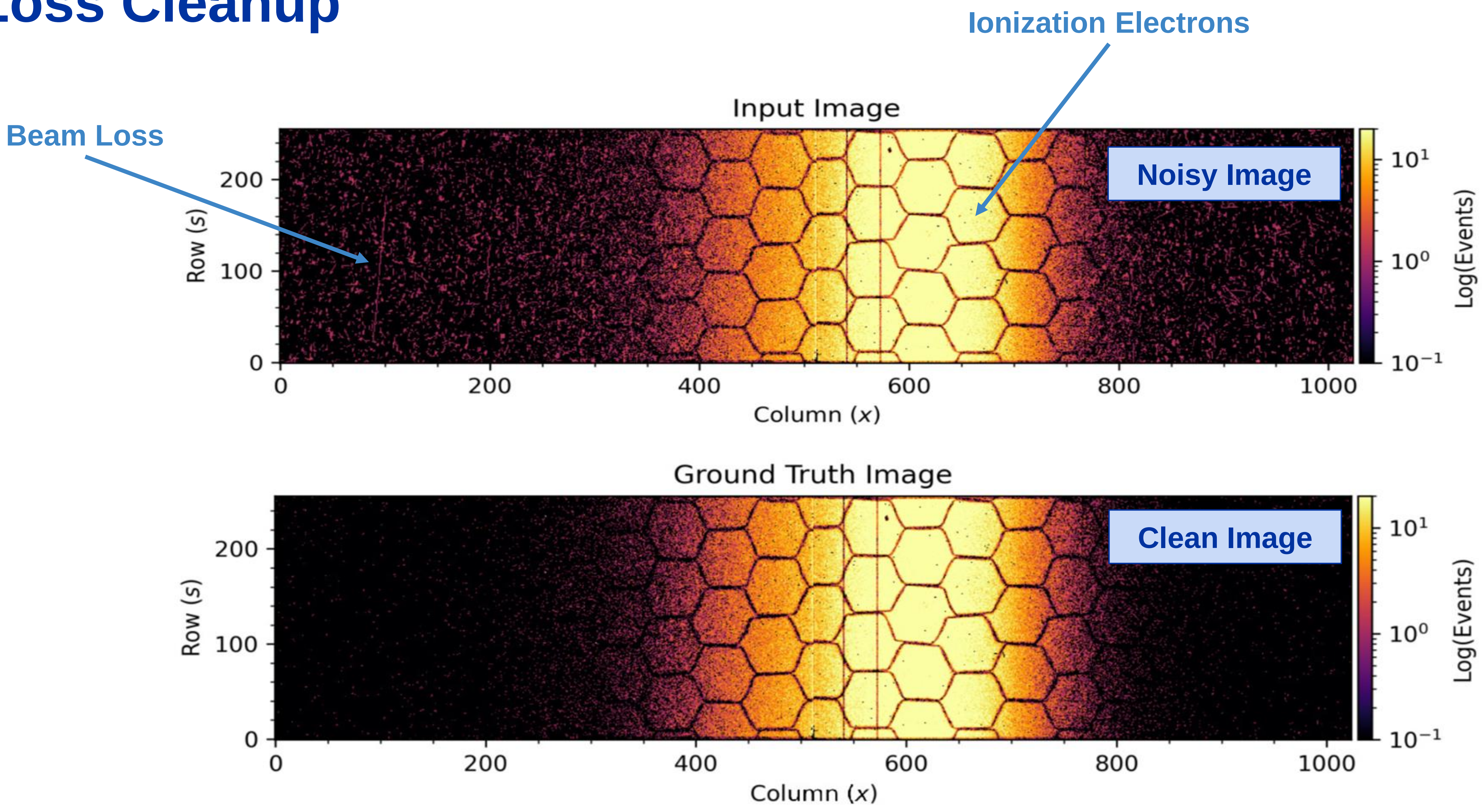
All Events Video Sequence



Source: https://bgi.web.cern.ch/

2. Noise Removal

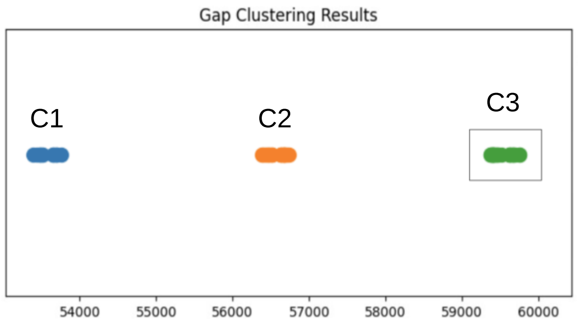
Loss Cleanup



Previous work: DBSCAN Clustering

Expert knowledge: "Beam losses refer to clusters of points that are sufficiently close together at a given time; everything else is signal (**ionization electrons**)."

TIME CLUSTERING

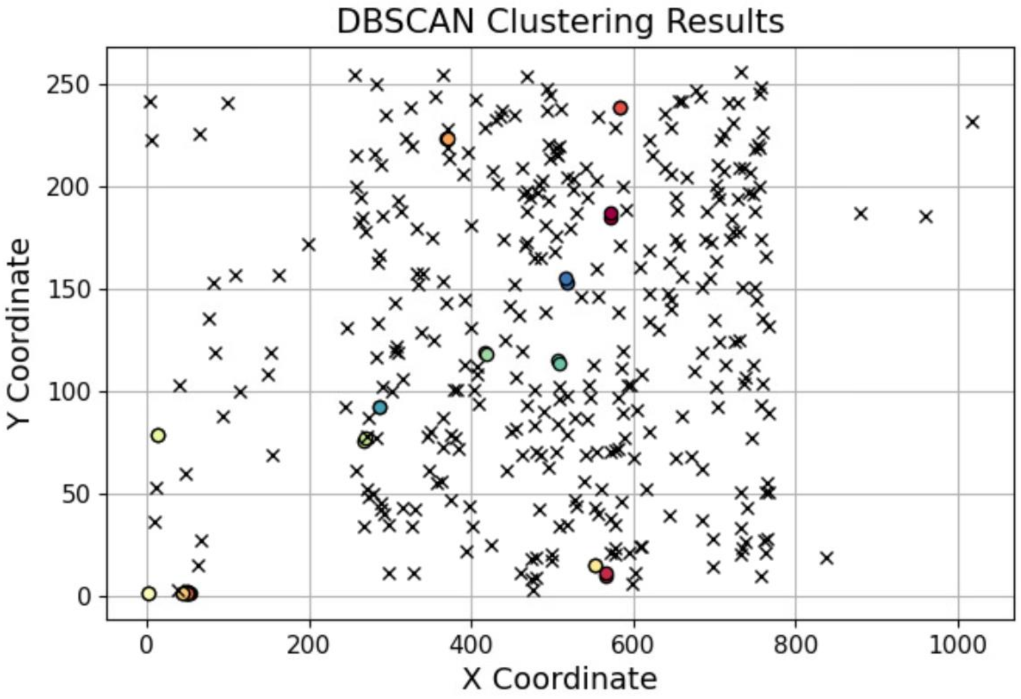


nanoseconds

apply spatial clustering to every time cluster found

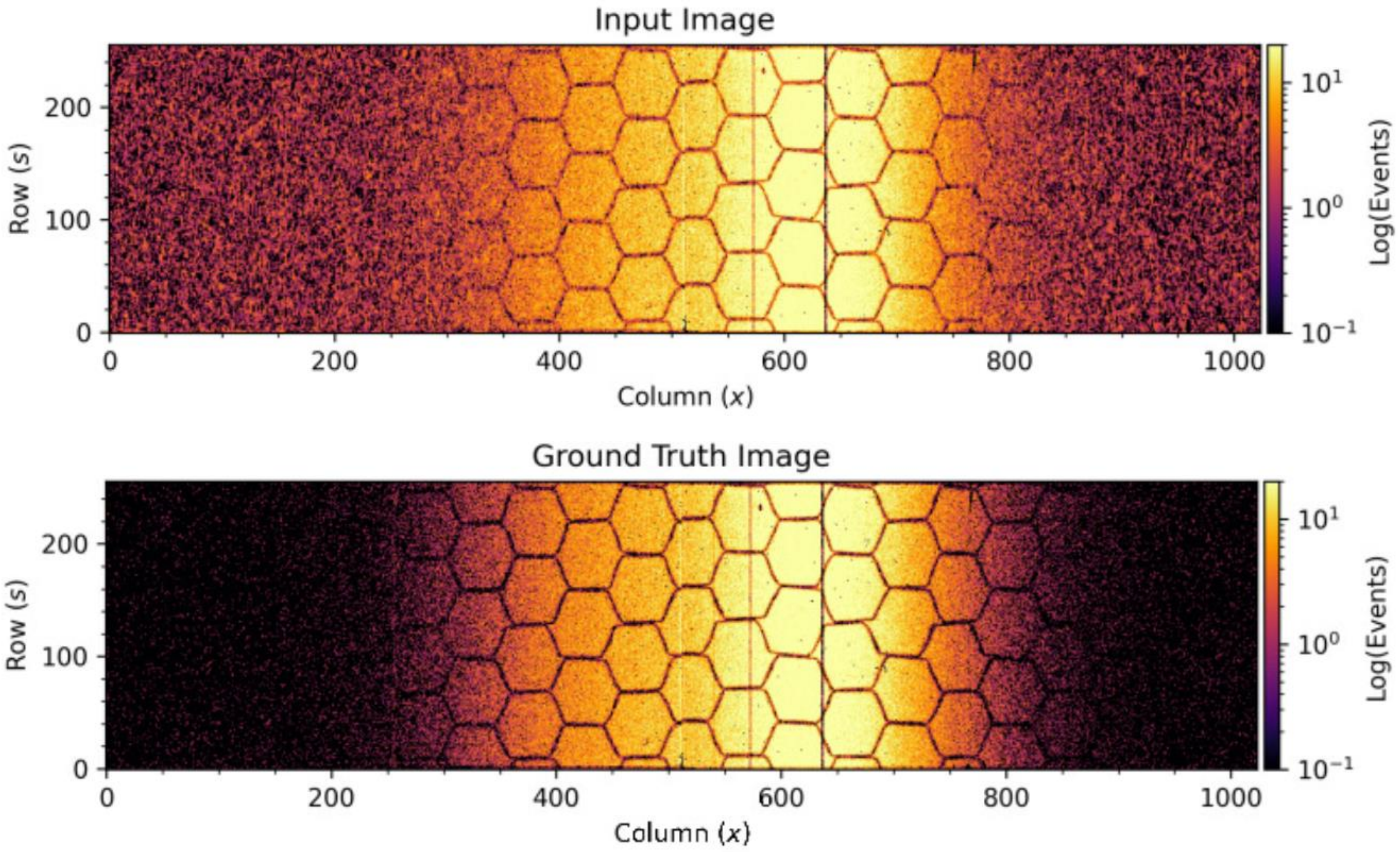
C3

SPATIAL CLUSTERING



It works but it is too slow…

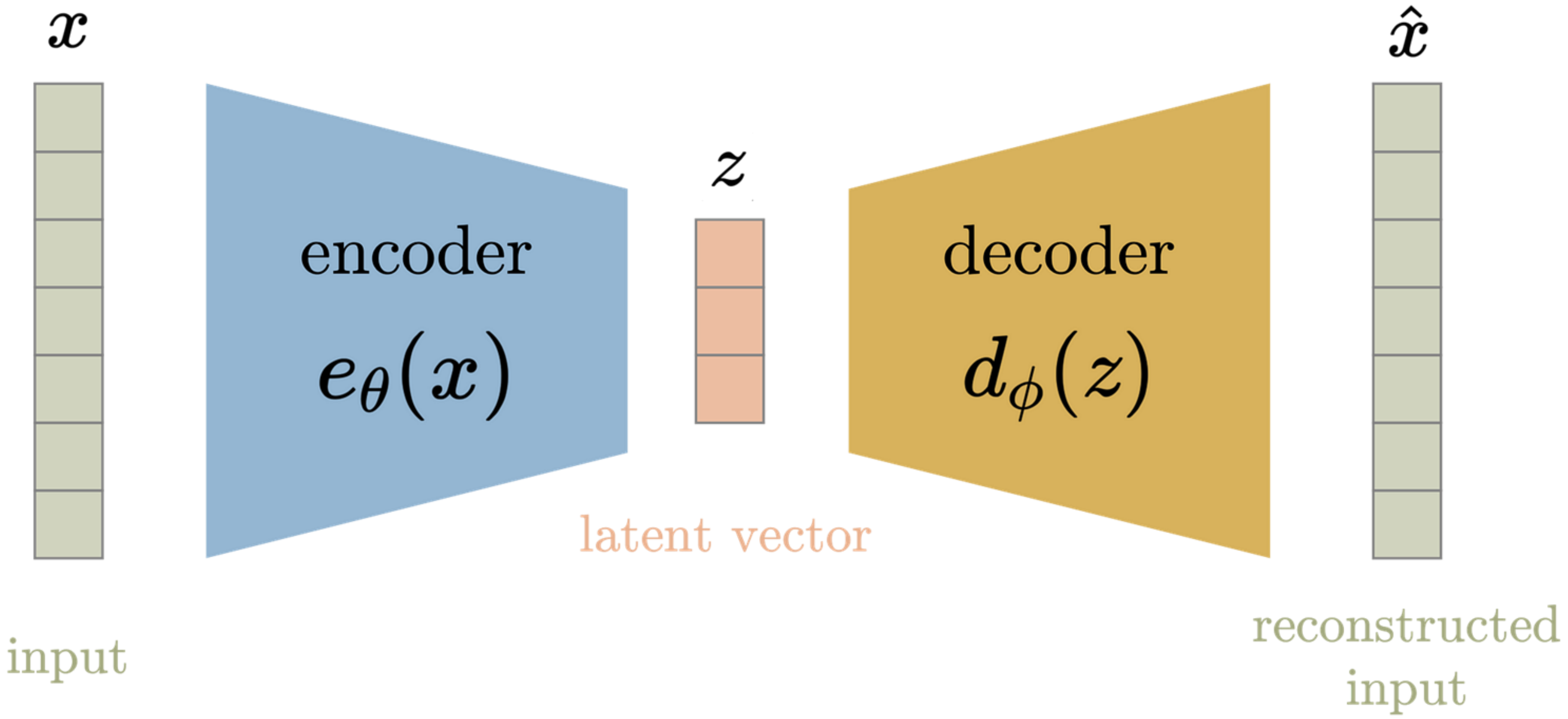
It takes from **20 to 100 seconds** to analyze **each acquisition** (even with multiprocessing on several CPUs)



How can ML help us?

- **Unsupervised learning**. Trying to directly detect the noise without labeled data.
- **Supervised learning**. Using denoised data as our ground truth.
- **Self-supervised learning**. Creating noisy synthetic images from clean data and train in a supervised way.
- **Algorithms:**
  - Traditional methods (e.g. median filtering, wavelet denoising, etc.)
  - **Autoencoders (our selection / baseline)**
  - Generative Adversarial Networks (GANs)
  - Vision Transformers (ViT)

What is an autoencoder?

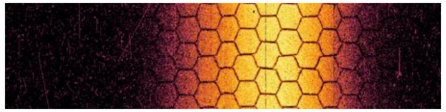


Source: Difference between AutoEncoder (AE) and Variational AutoEncoder (VAE) (Aqee Anwar) (https://towardsdatascience.com/)

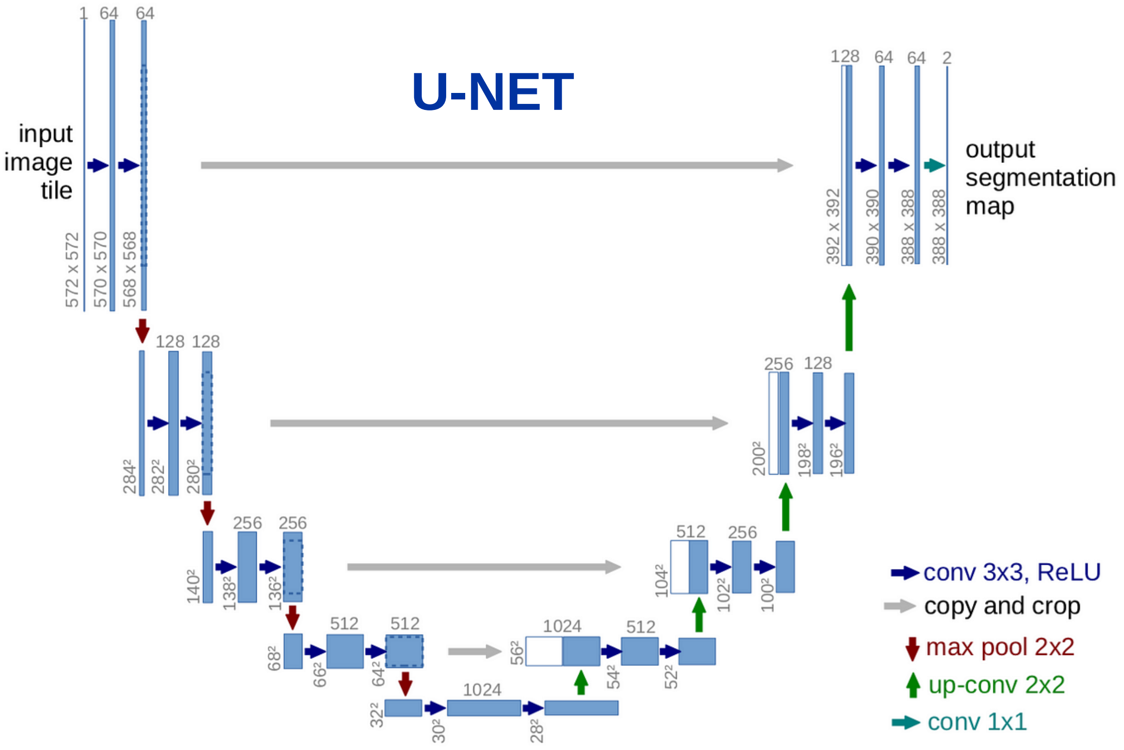
Proposal: using autoencoders to speed up the process

Assumption: "DBSCAN works. Let’s train a neural network in a supervised way **to learn the mapping** from the noisy to the denoised DBSCAN outputs.”

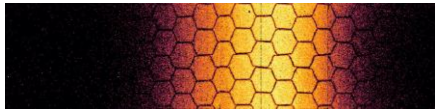
NOISY



U-NET



DENOISED



Source: U-Net: Convolutional Networks for Biomedical Image Segmentation

Training Stage

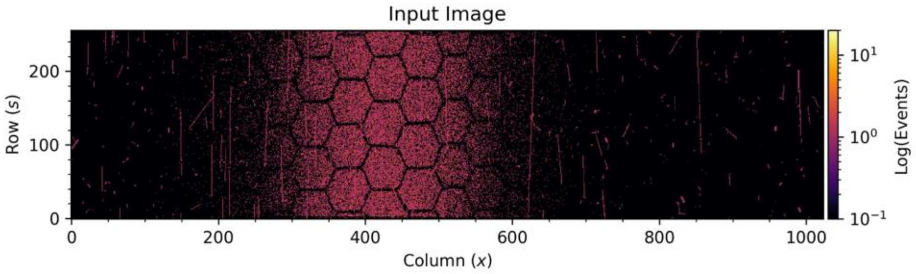
DENOISE USING DBSCAN (SLOW)

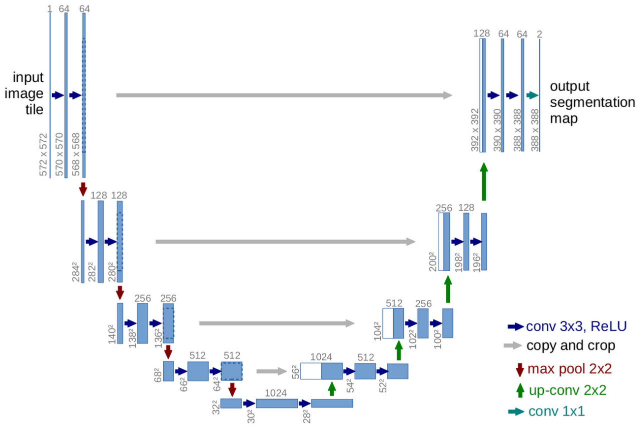
DBSCAN OUTPUT

binarize

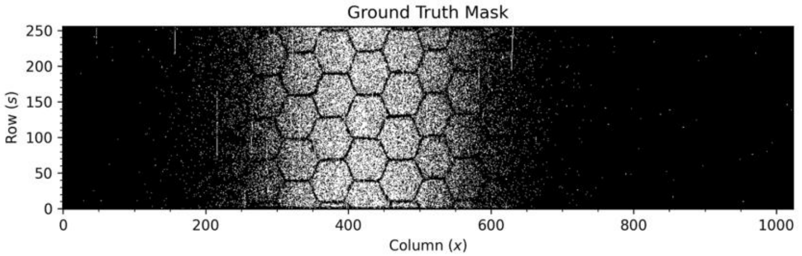
U-NET ARCHITECTURE

NOISY IMAGE

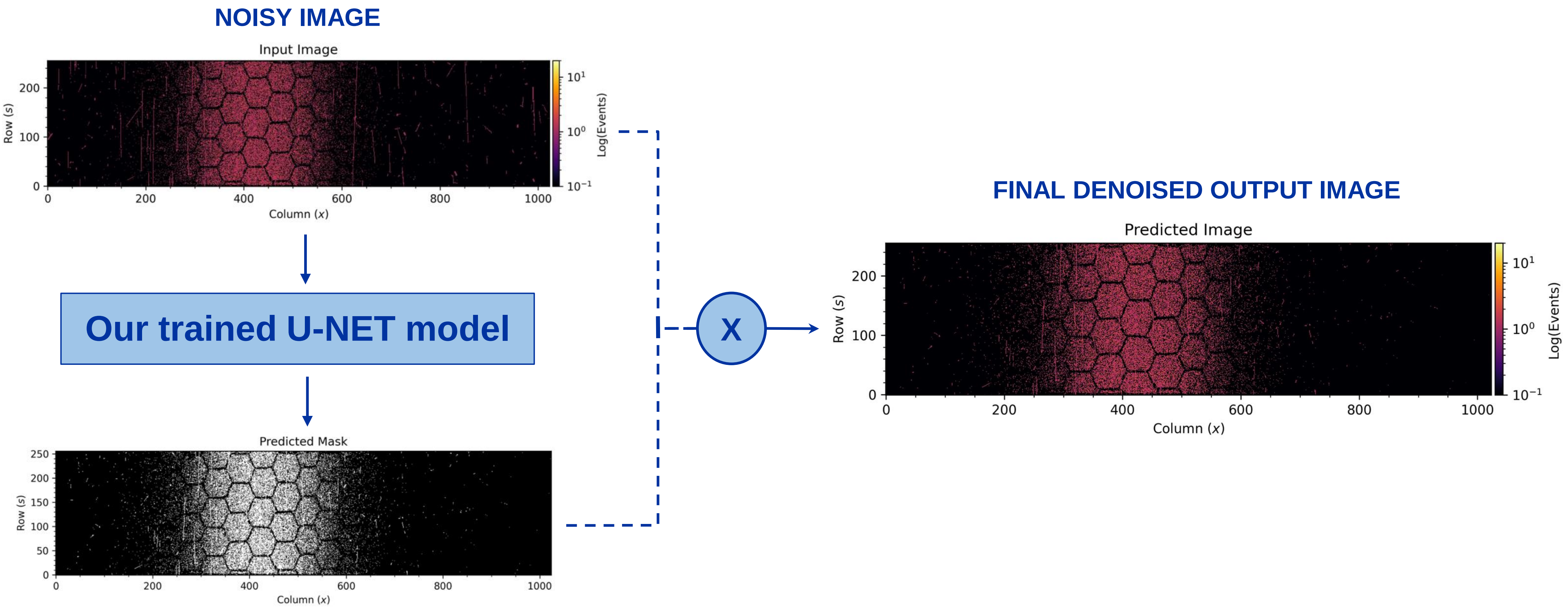




DENOISED BINARY IMAGE



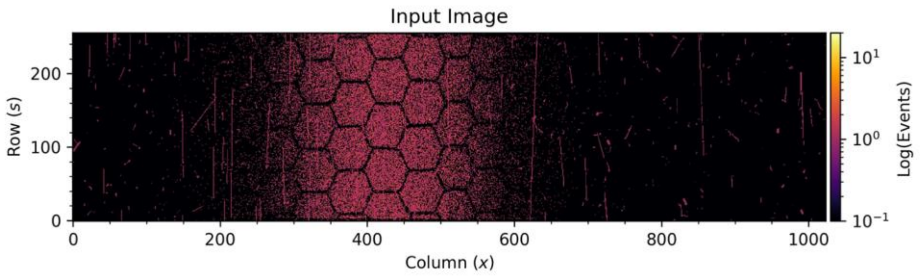
Inference


NOISY IMAGE
Input Image
Row (s)
Column (x)
Log(Events)
Our trained U-NET model
Predicted Mask
X
FINAL DENOISED OUTPUT IMAGE
Predicted Image

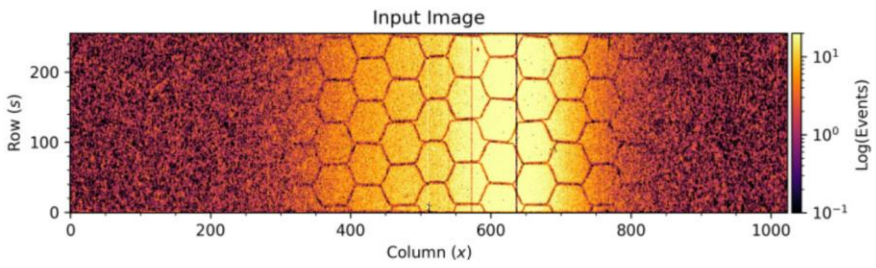
Dataset Details

- We have analyzed about **1500 image pairs** from BGIH 2021 and 2022 data.
- Not all of the data is actually valid (e.g. images with detectors OFF, not enough beam or recorded events, bad fits, etc.)
- Trained one model on **high integration** time images (~100 imgs)
- Trained another on **low integration** time images (~350 imgs)
- 75% training set and 25% validation set
- Preprocessing: **Min-Max Norm** and **Log Transformation**

LOW INTEGRATION TIME



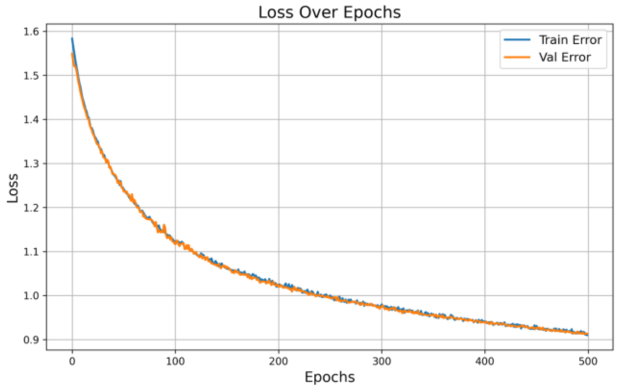
HIGH INTEGRATION TIME

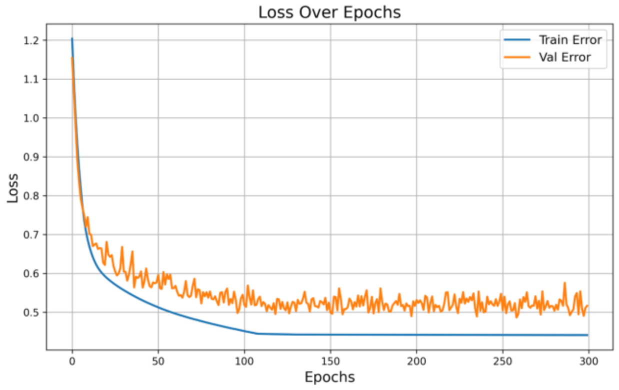


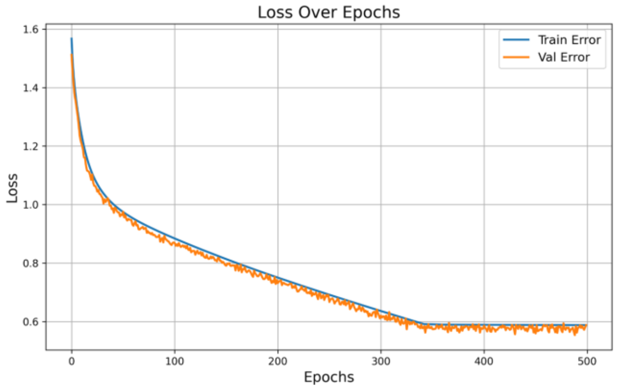
Training Configuration

Our preliminary study shows that it is actually possible to learn the mapping 😊

- batch size = 1-8 imgs
- lr = 1e-6 to 1e-8 (lr_scheduler)
- weight decay = 1e-8
- optimizer = AdamW
- Loss = Binary Cross Entropy (BCE) + Dice Loss







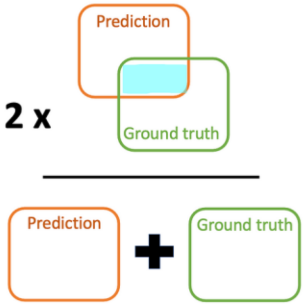
Sum Loss = BCE + Dice Loss

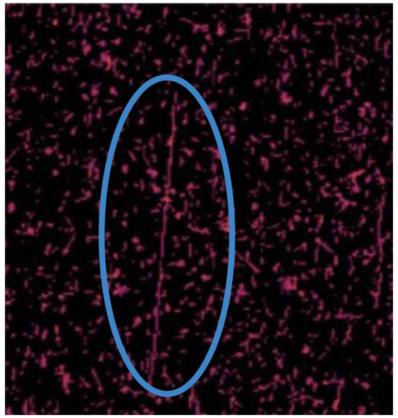
Binary Cross Entropy for pixel-wise accuracy

$$BCE = -\frac{1}{N}\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) + (1 - y_i) \cdot log(1 - \hat{y}_i)$$

DL = (1 - Dice Coefficient) for contextual spatial features

$$\text{Dice} = \frac{2\ X\ Area\ of\ overlap}{Total\ area} =$$



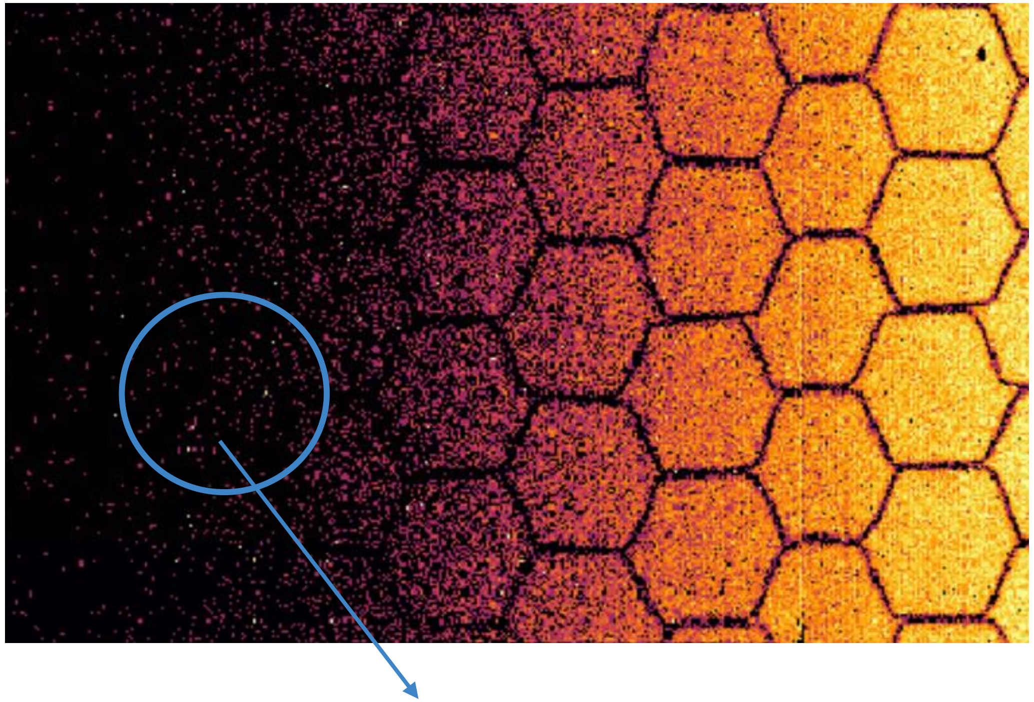


3. Extra Data Processing

Extra #1: Time Over Threshold (ToT) Filtering

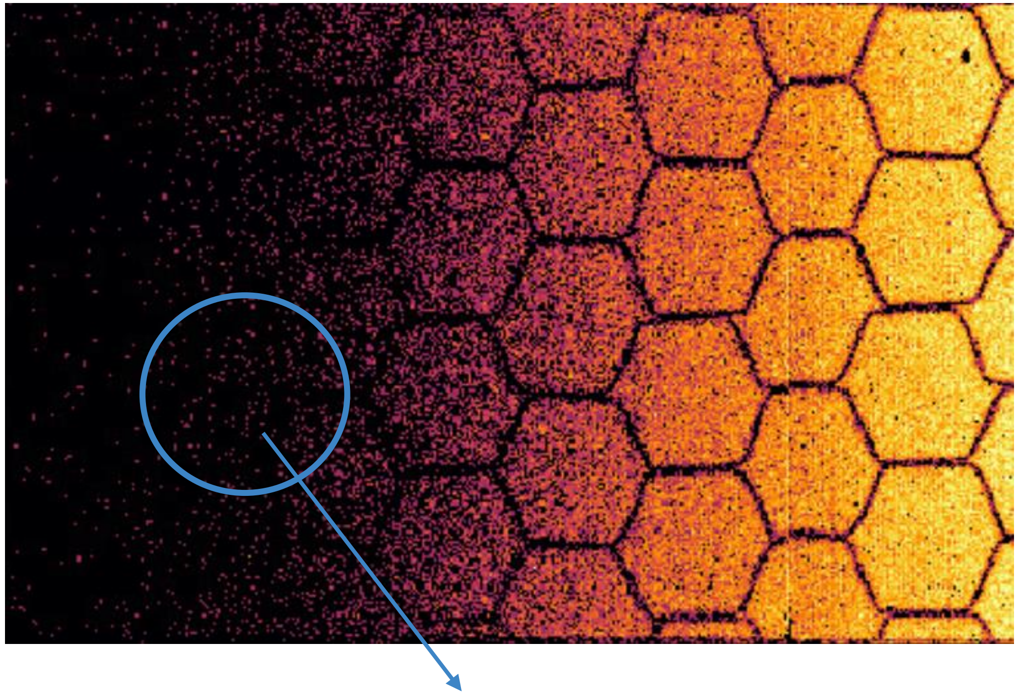
Assumption: "Events with ToT > 20 are likely to be beam loss."

After DBSCAN



Highly saturated pixels!

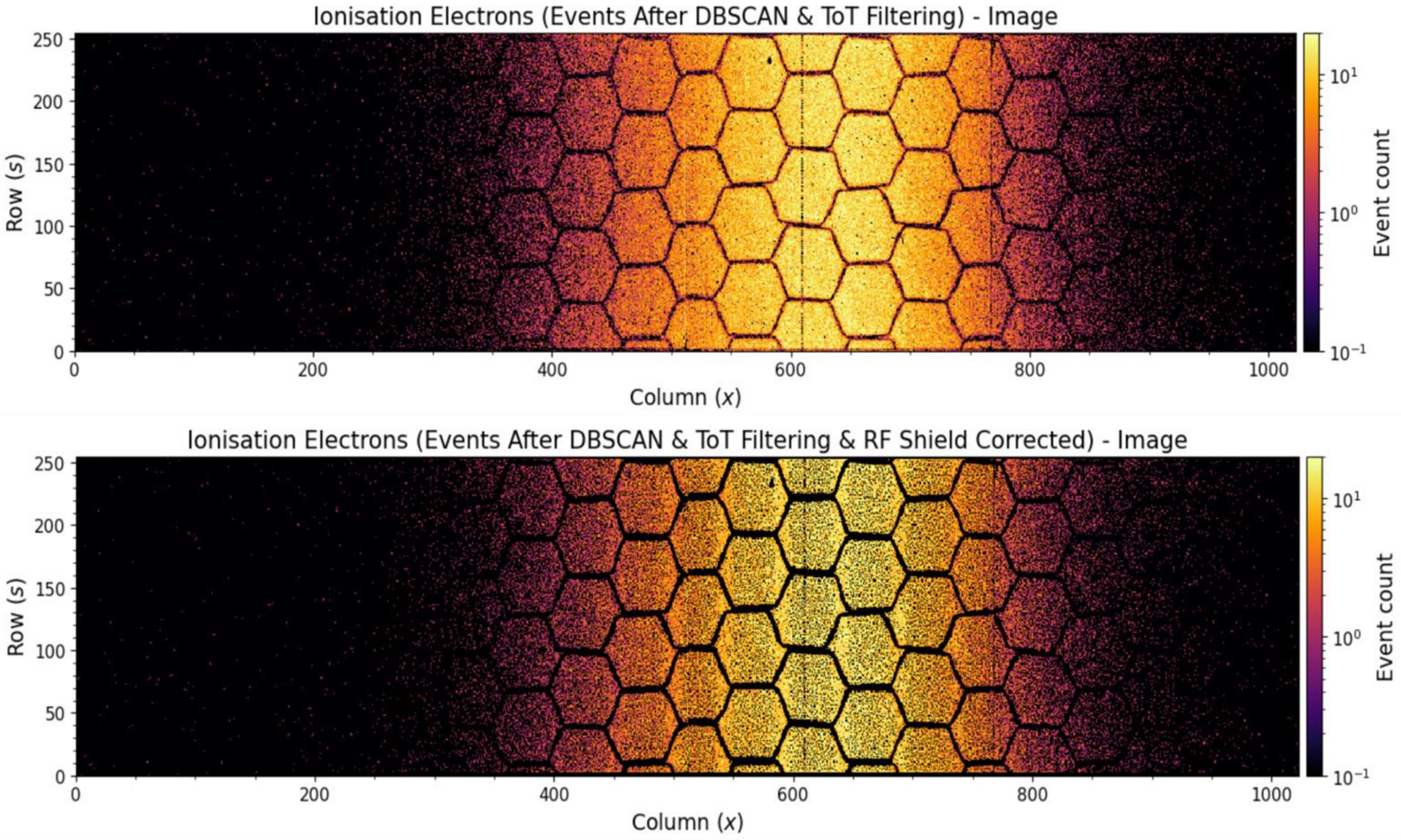
After DBSCAN + ToT Filtering



Saturation is gone 😊

Extra #2: RF Shield Correction

Filter out honeycomb and manually annotated noisy pixels 🐝



The RF Shield helps to remove any kind of electromagnetic interference

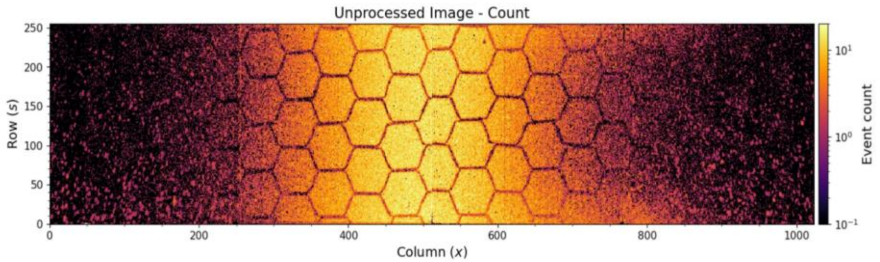
But it is annoying for properly calculating the beam profiles

correction

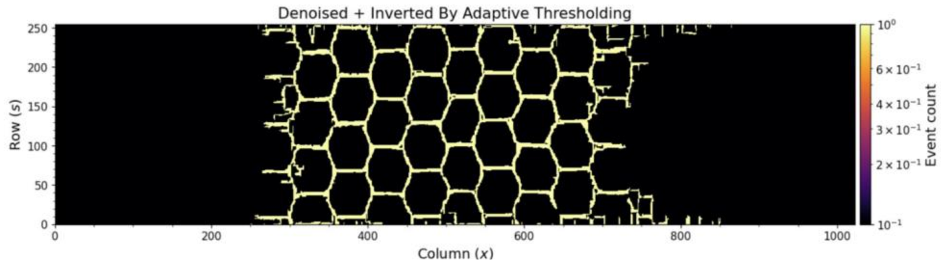
Extra #2: RF Shield Correction

We can also interpolate the honeycomb using TELEA[1] inpainting!

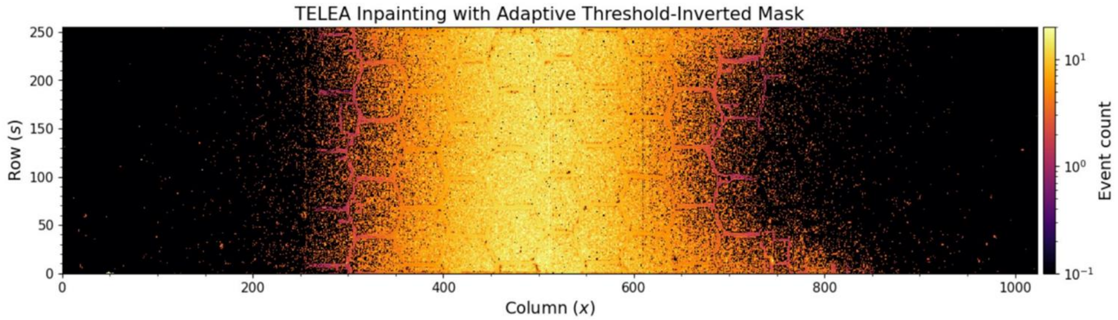
Noisy Image



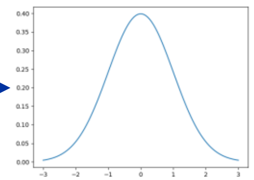
RF Shield



Denoised and Inpainted



Improves the beam profile

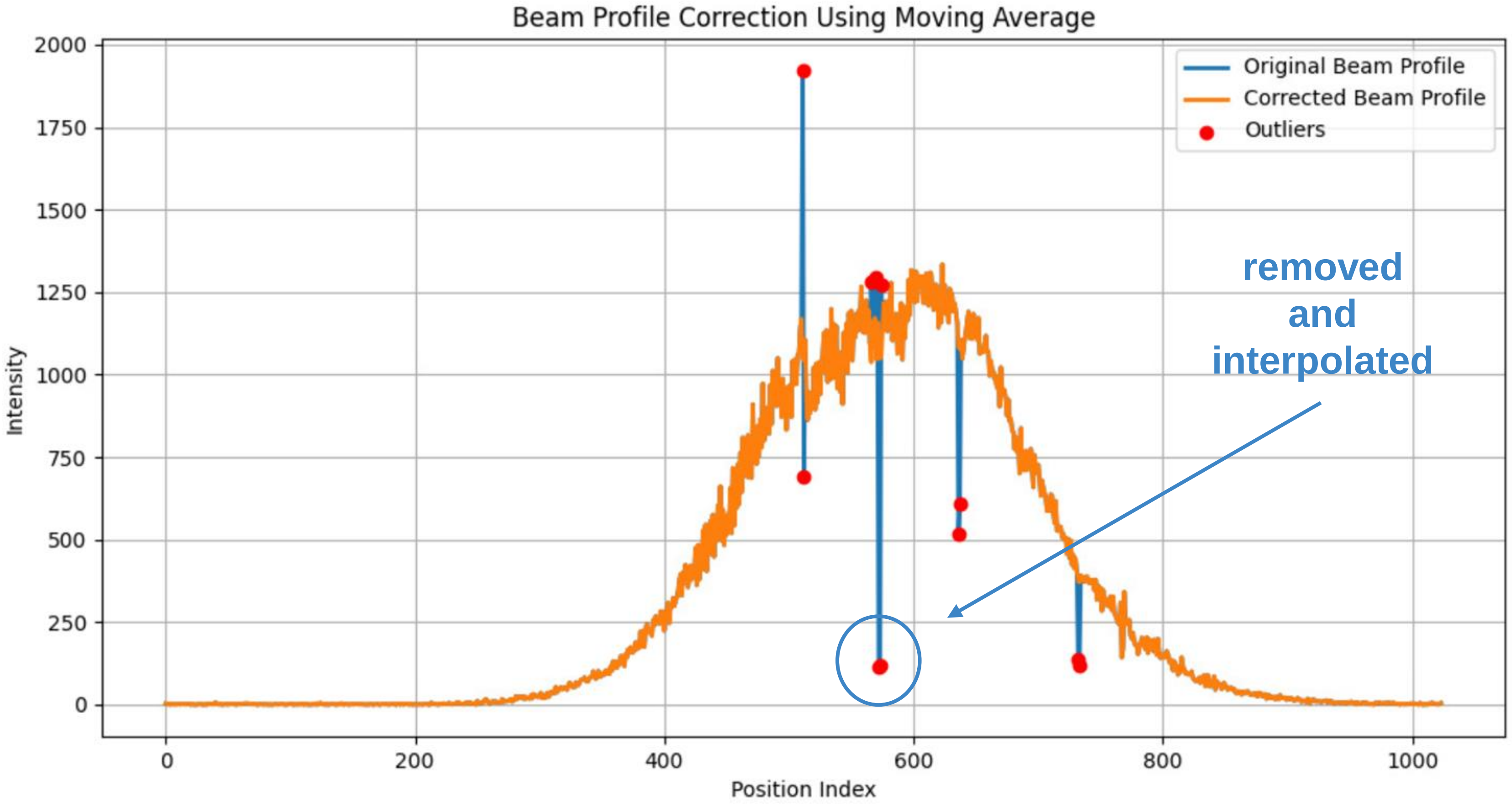


1. Telea, Alexandru (2004). An Image Inpainting Technique Based on the Fast Marching Method. Journal of Graphics Tools. 9.10.1080/10867651.2004.10487596.

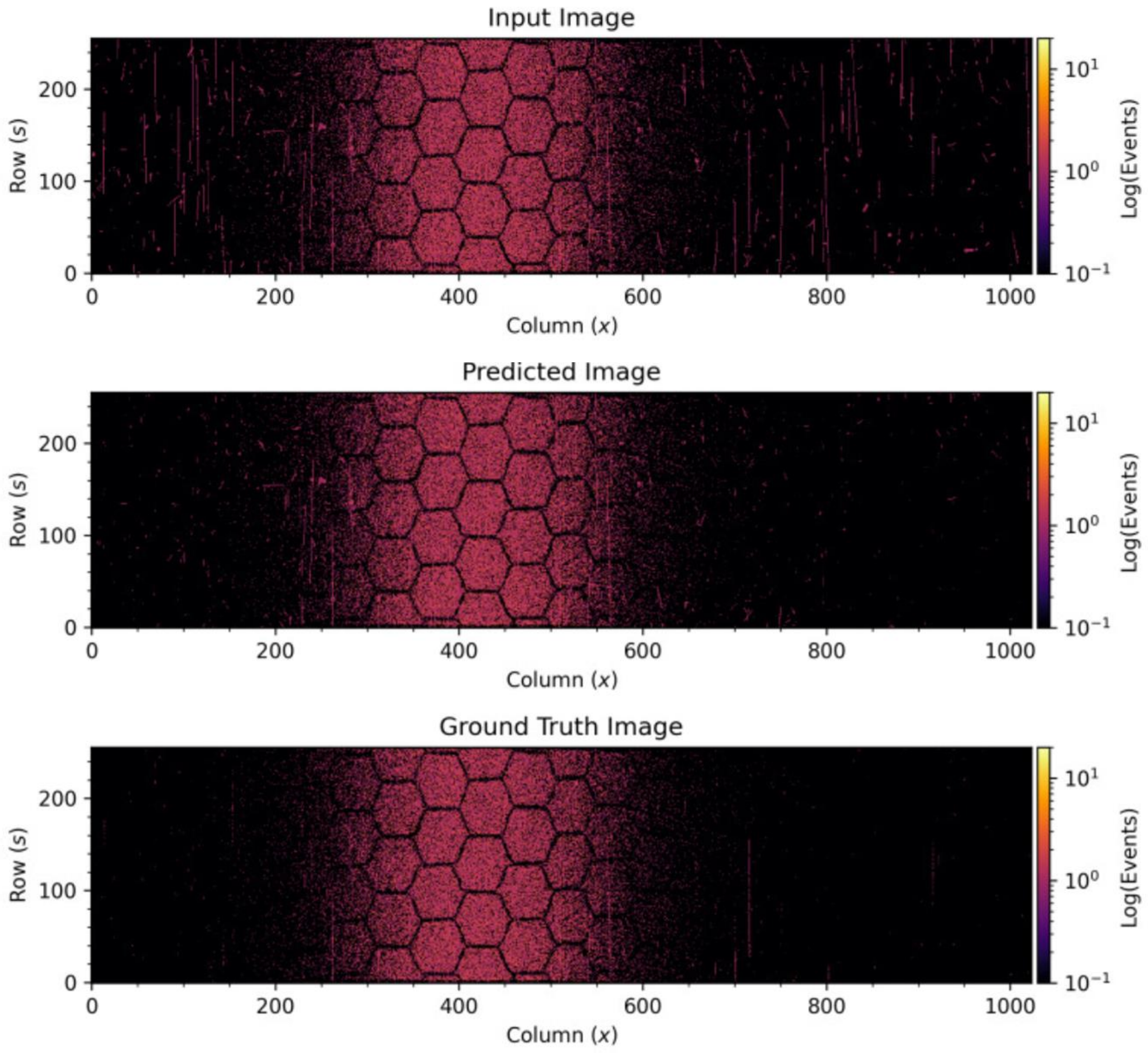
Slide courtesy of Glenn Anta

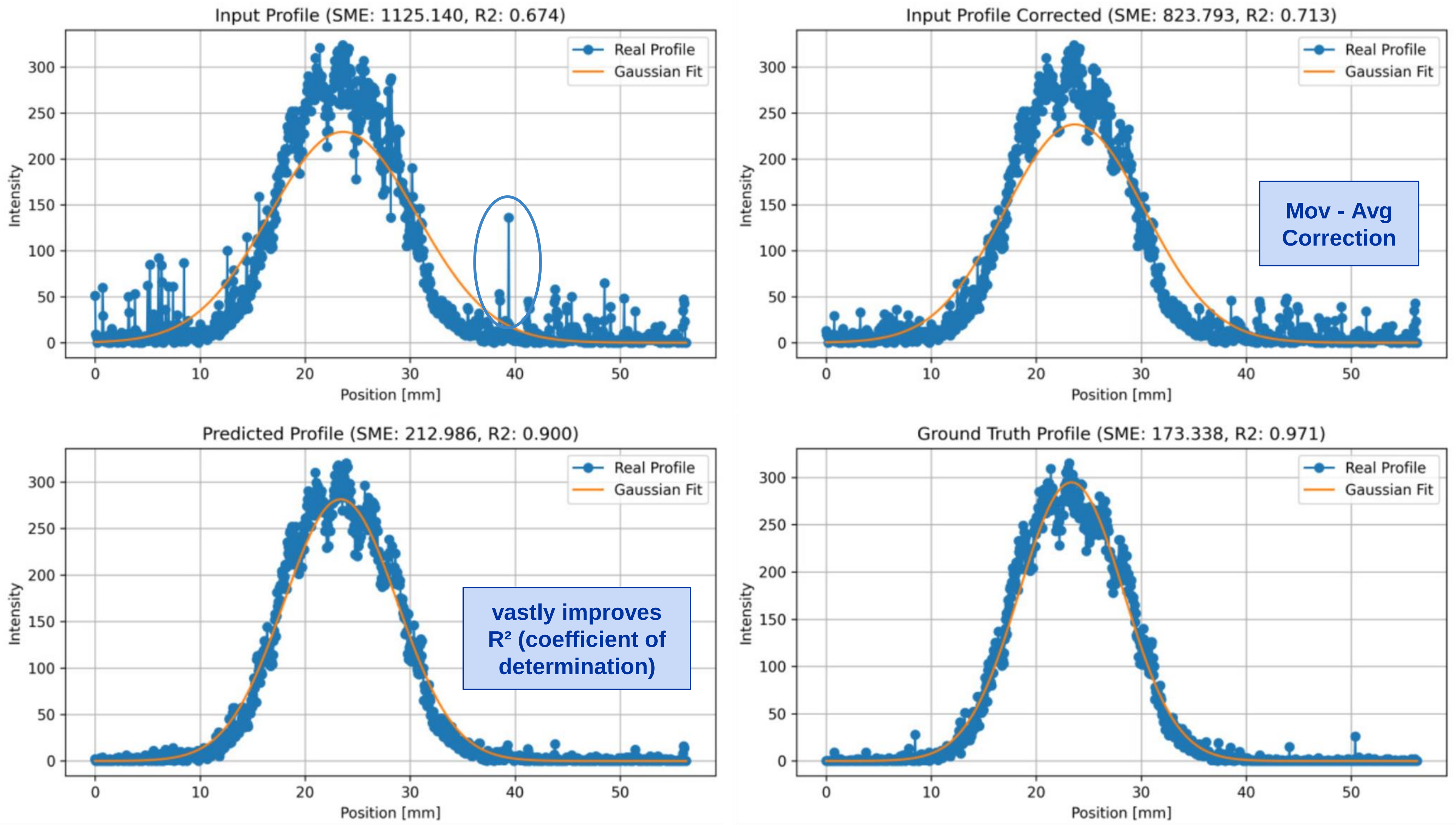
Extra #3: Moving Average Thresholding

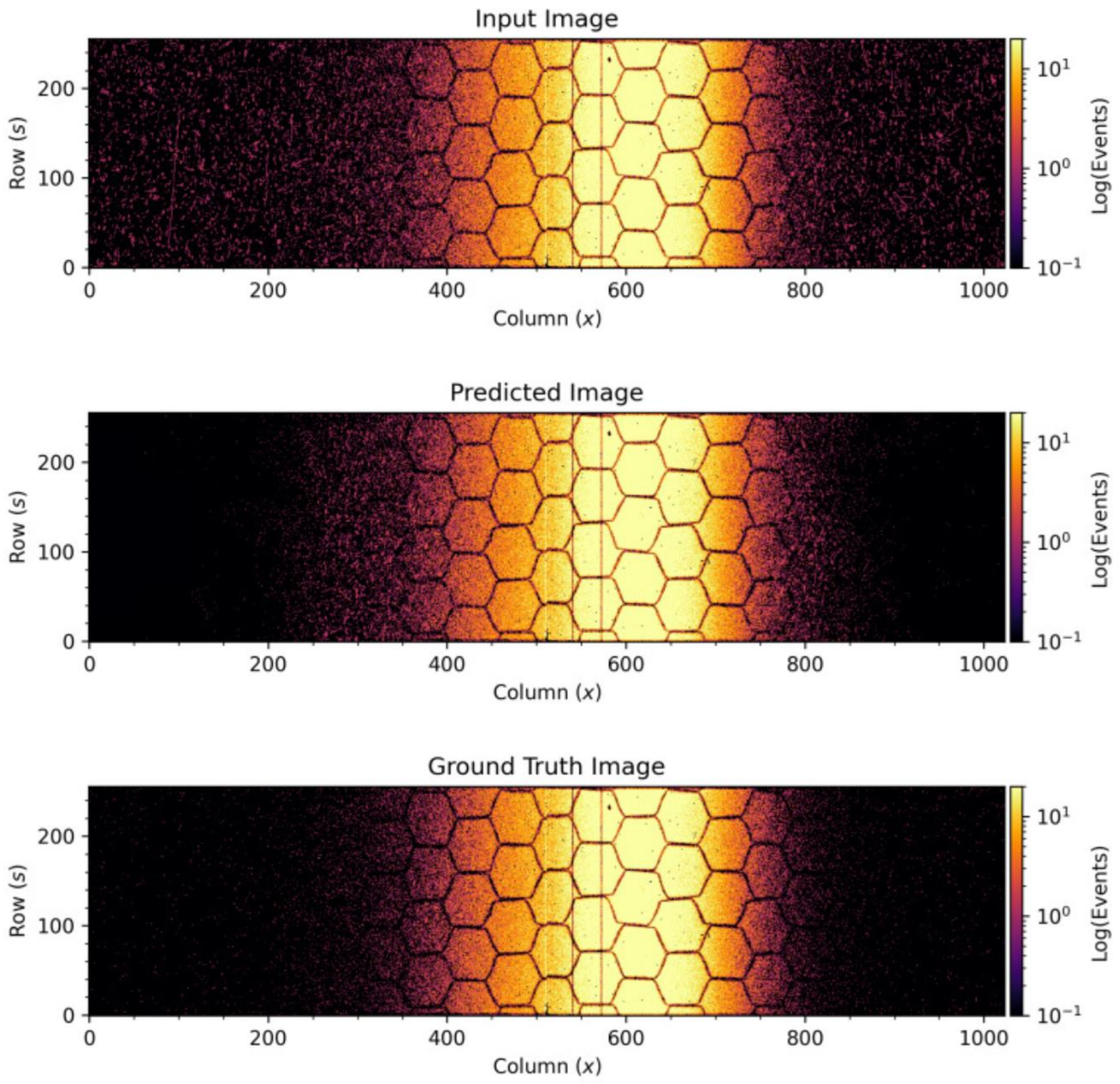
Improve the final beam profile by detecting noisy-pixel outliers in a very safe way 😁

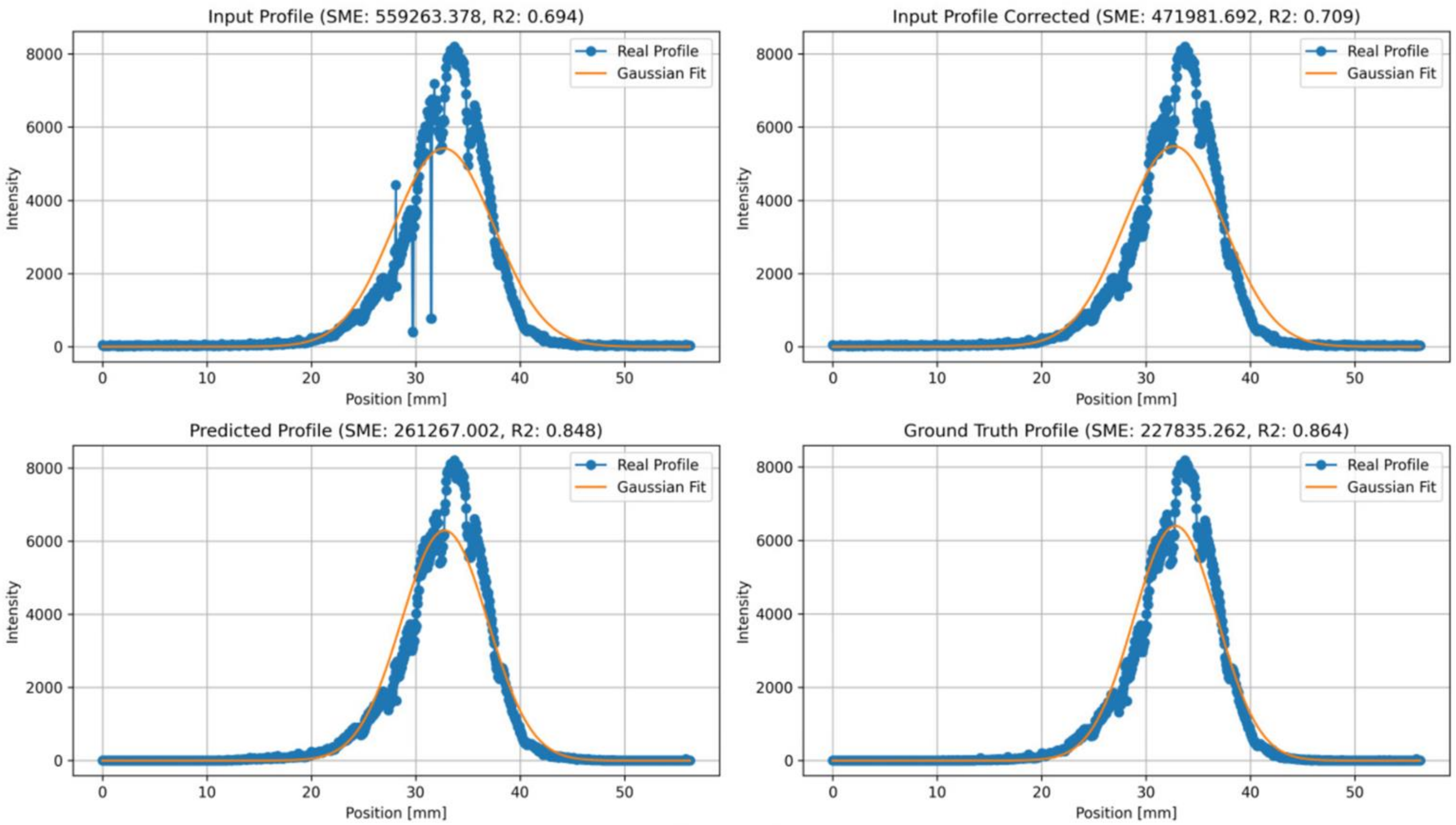


4. Results


Input Image
Row (s)
Column (x)
Log(Events)
Predicted Image
Row (s)
Column (x)
Log(Events)
Ground Truth Image
Row (s)
Column (x)
Log(Events)


Input Profile (SME: 1125.140, R2: 0.674)
Input Profile Corrected (SME: 823.793, R2: 0.713)
Predicted Profile (SME: 212.986, R2: 0.900)
Ground Truth Profile (SME: 173.338, R2: 0.971)
Real Profile
Gaussian Fit
Intensity
Position [mm]
Mov - Avg Correction
vastly improves R² (coefficient of determination)


Input Image
Predicted Image
Ground Truth Image
Row (s)
Column (x)
Log(Events)
0
100
200
400
600
800
1000
10^{1}
10^{0}
10^{-1}


Input Profile (SME: 559263.378, R2: 0.694)
Input Profile Corrected (SME: 471981.692, R2: 0.709)
Predicted Profile (SME: 261267.002, R2: 0.848)
Ground Truth Profile (SME: 227835.262, R2: 0.864)
Real Profile
Gaussian Fit
Intensity
Position [mm]
0
2000
4000
6000
8000
0
10
20
30
40
50

5. Conclusions & Future Work

Good news

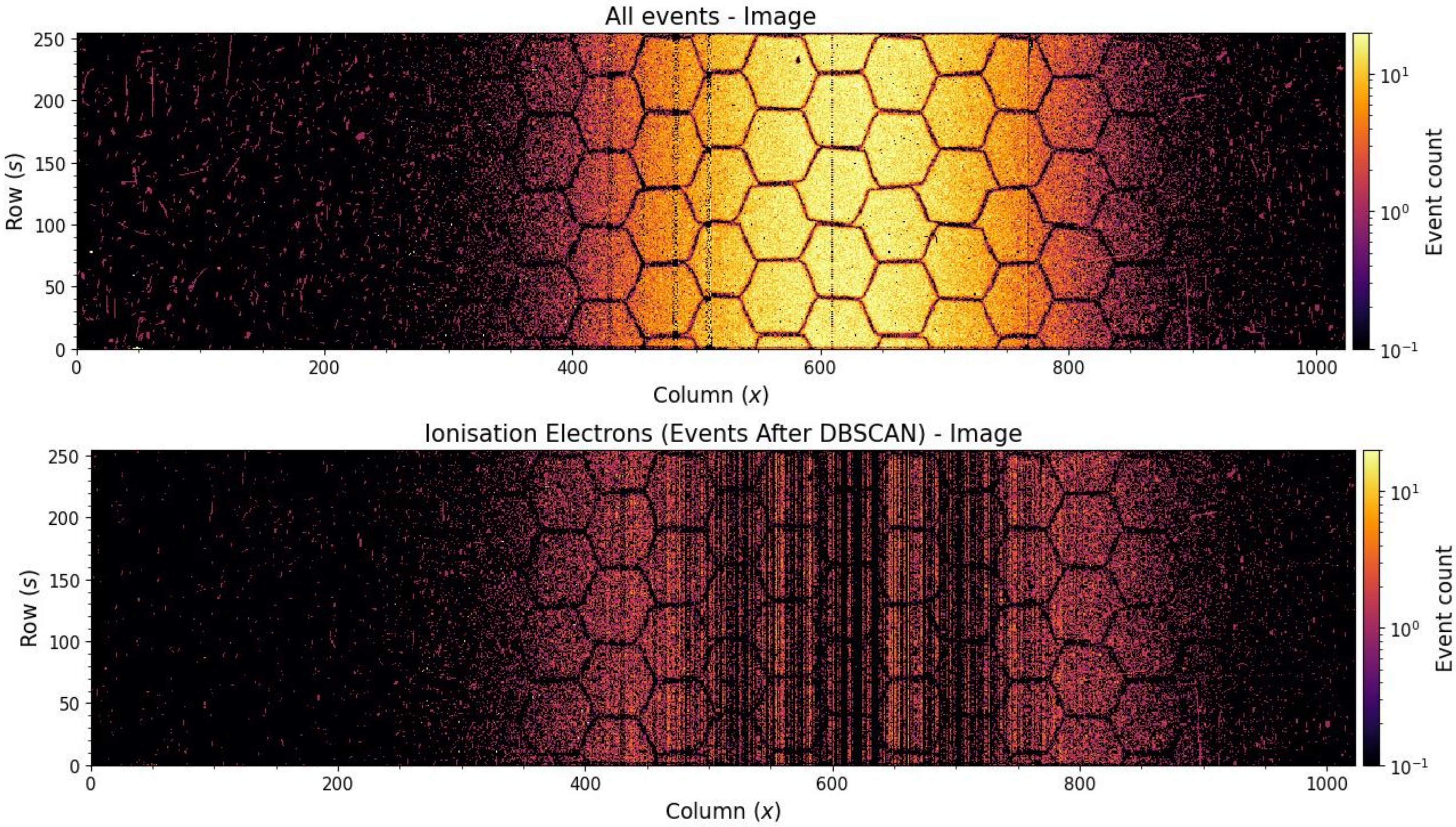
- DBSCAN is not viable for **real-time** operational purposes but it is robust and can be used during post-processing and for training ✔
- **U-Net** has been proven to effectively learn **DBSCAN's mapping** ✔
- Standard **signal processing techniques** (e.g. Mov-Avg Confidence Intervals) can vastly improve the profiles in a safe way ✔
- **ToT filtering** and **honeycomb** adjustments are also safe and fast (but it requires manual pixel mask adjustments) ✔
- We have now scripts to easily generate, filter and analyze our **image dataset** from the raw files (it will be essential for future training and test) ✔

Known limitations

- We need **more data** and more updated data (right now we only trained on BGIH files from 2021 and 2022) 📖
- We need to test the models in actual **test** data taken from different days and under **different conditions or beam types** 🌧
- In any case, we need some consensus on **integration times**, as images differ a lot depending on how many milliseconds we integrate 🤝
- We are ruling out **time information** so profiles will not always be perfect 😔
- The ultimate goal would be to use the whole video to obtain the beam profile (e.g. using **Transformers**) 🤖

Thanks!

DBSCAN is not perfect: it sometimes fails but it is very rare…



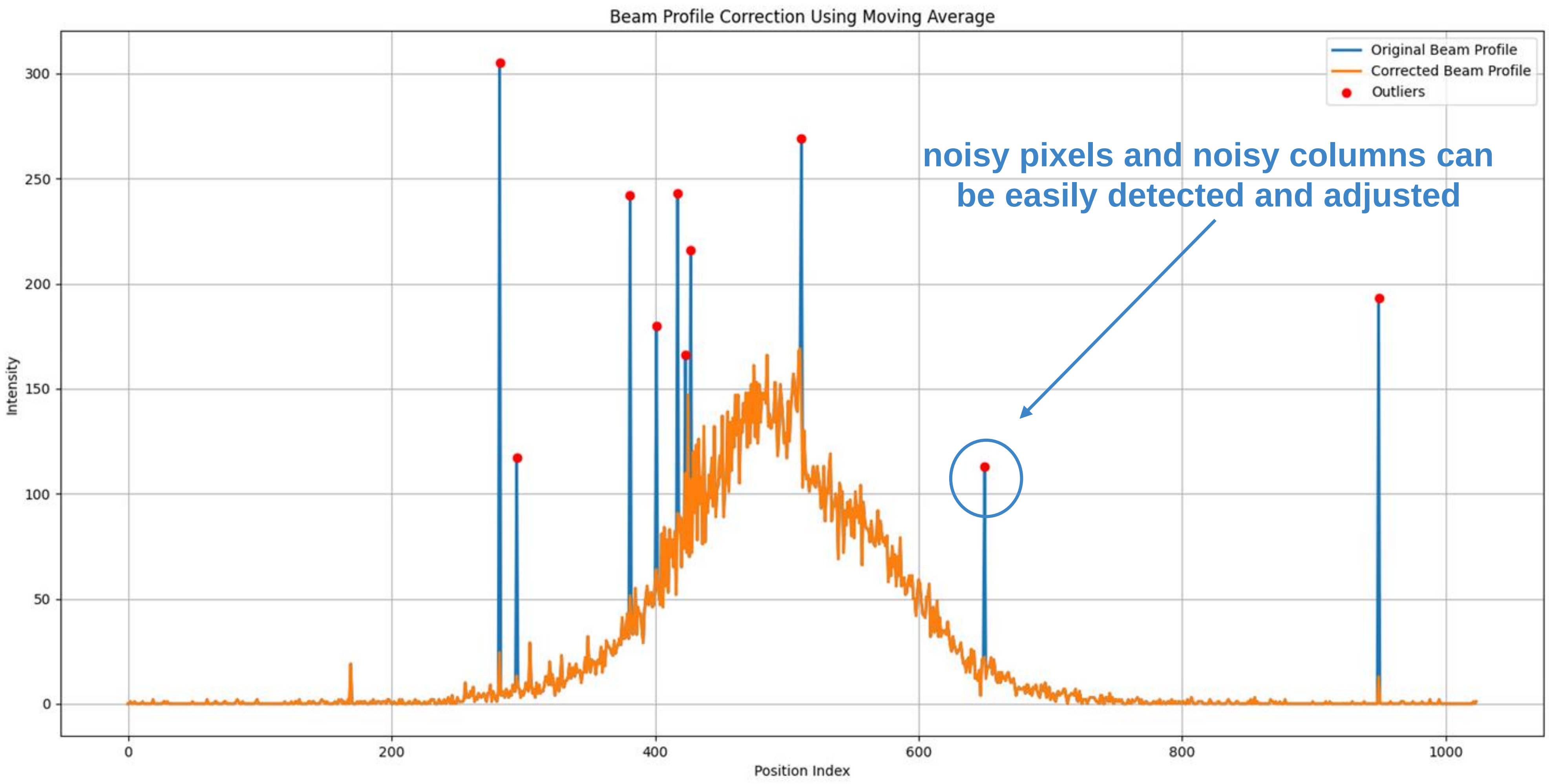
Hardware and GPU Information

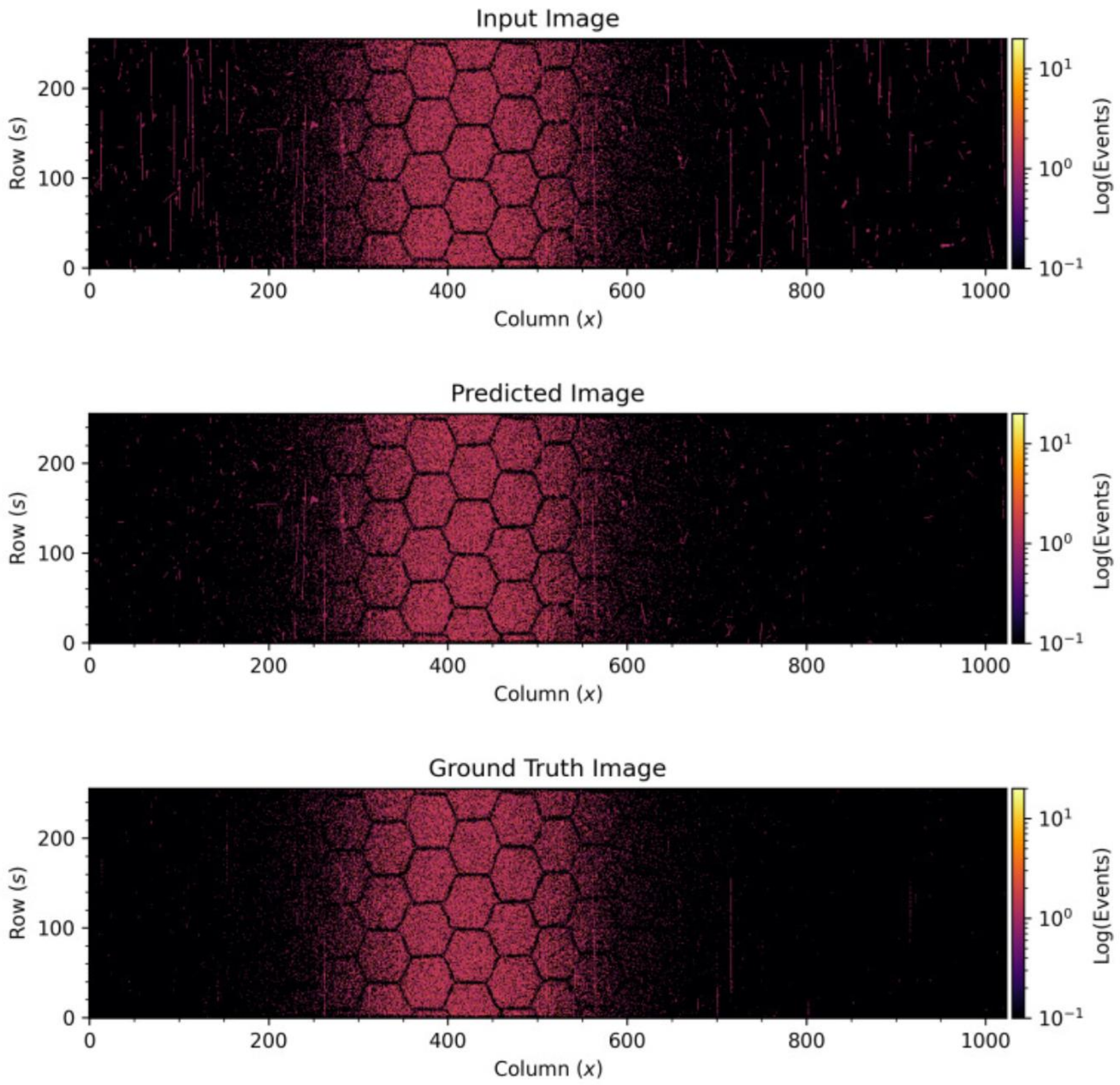
The model was trained using CERN's infrastructure: https://ml.cern.ch

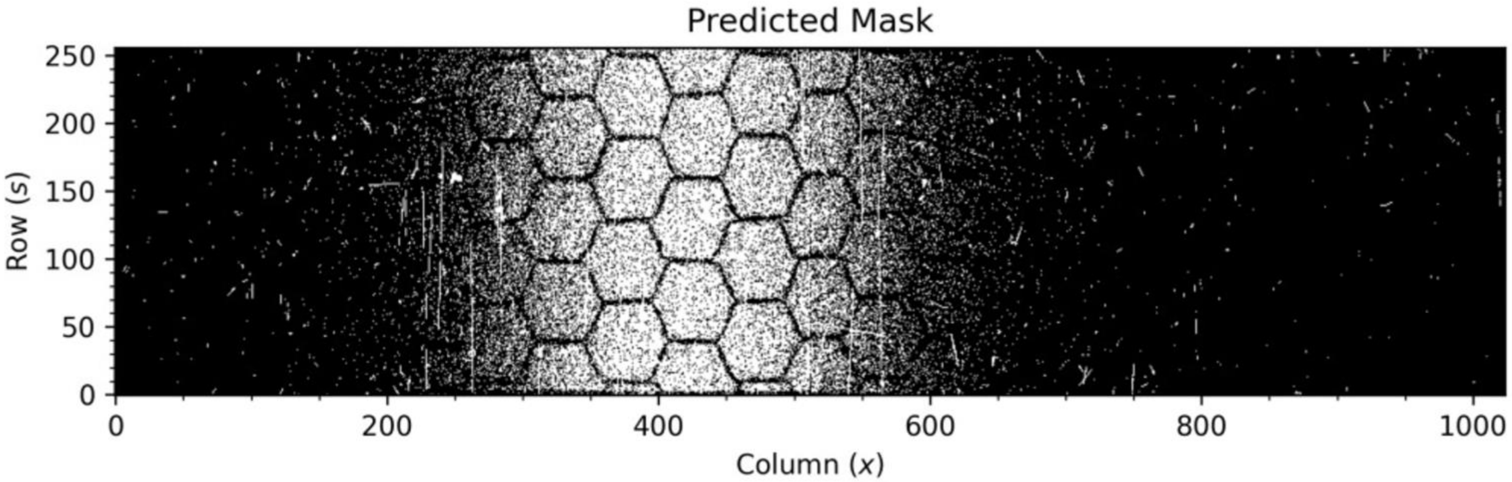
NVIDIA TESLA T4 - THE MOST VERSATILE GPU

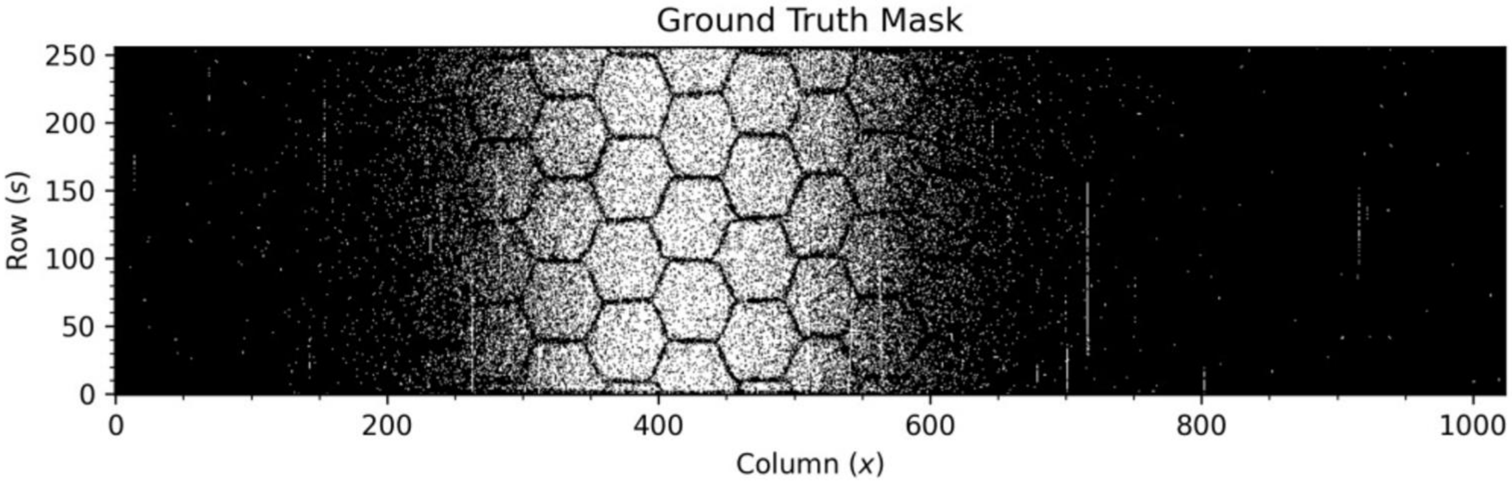
	T4	P4	M60	M10
GPUs / Board (Architecture)	1 (Turing)	1 (Pascal)	2 (Maxwell)	4 (Maxwell)
CUDA Cores	2,560	2,560	4,096 (2,048 per GPU)	2,560 (640 per GPU)
Memory Size	16 GB GDDR6	8 GB GDDR5	16 GB GDDR5 (8 GB per GPU)	32 GB GDDR5 (8 GB per GPU)
vGPU Profiles	1 GB, 2 GB, 4 GB, 8 GB, 16 GB	1 GB, 2 GB, 4 GB, 8 GB	0.5 GB, 1 GB, 2 GB, 4 GB, 8 GB	0.5 GB, 1 GB, 2 GB, 4 GB, 8 GB
Form Factor	PCIe 3.0 Single Slot (rack servers)	PCIe 3.0 Single Slot (rack servers)	PCIe 3.0 Dual Slot (rack servers)	PCIe 3.0 Dual Slot (rack servers)
Power	70W	75W	300W (225W opt)	225W
Thermal	passive	passive	active/passive	passive
	PERFORMANCE Optimized			DENSITY Optimized

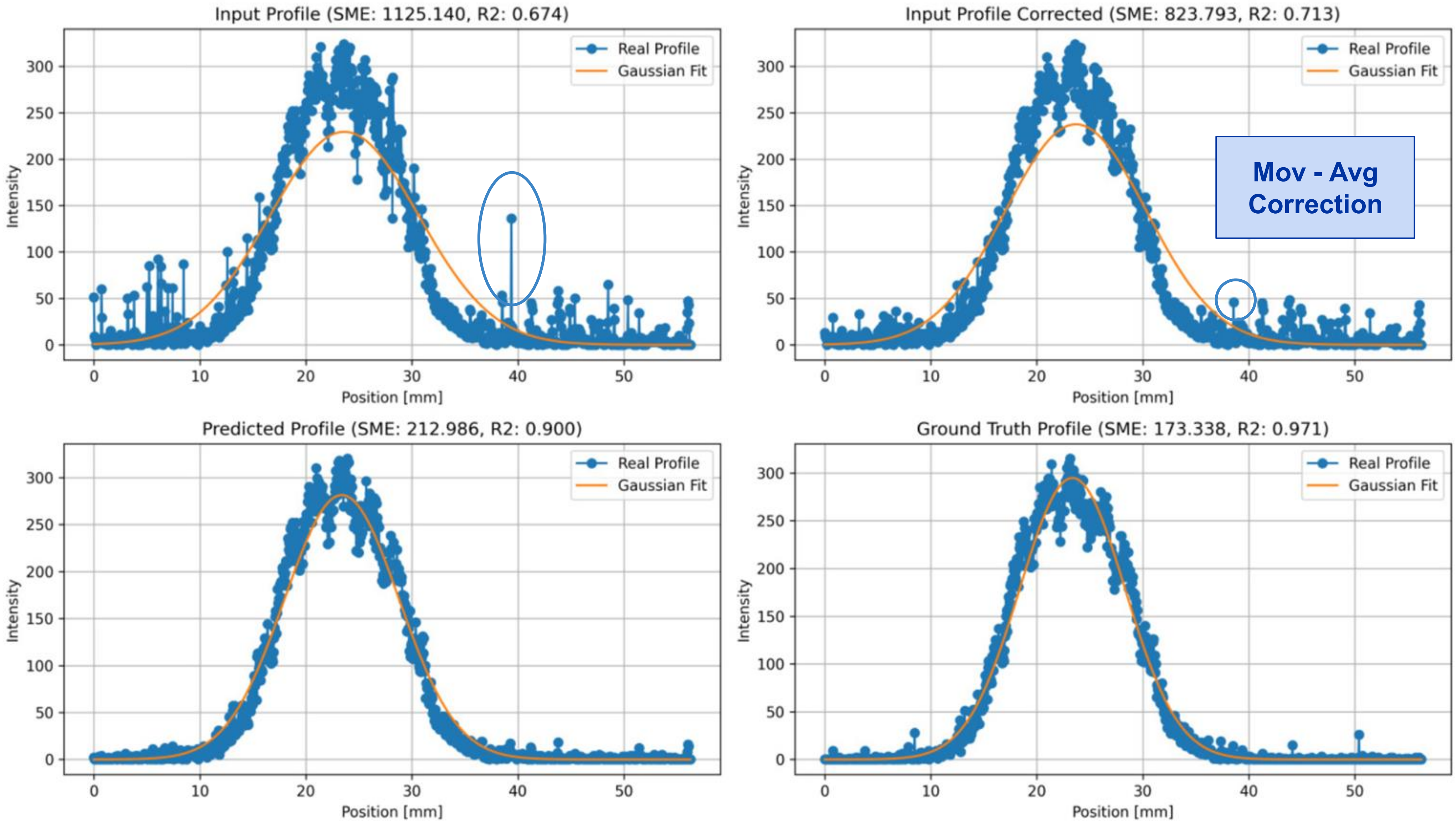
Source: NVIDIA Tesla T4 Powers Next Generation of Virtual Workstations (NVIDIA Blog)


Beam Profile Correction Using Moving Average
Original Beam Profile
Corrected Beam Profile
Outliers
noisy pixels and noisy columns can be easily detected and adjusted
Intensity
Position Index
0
50
100
150
200
250
300
0
200
400
600
800
1000

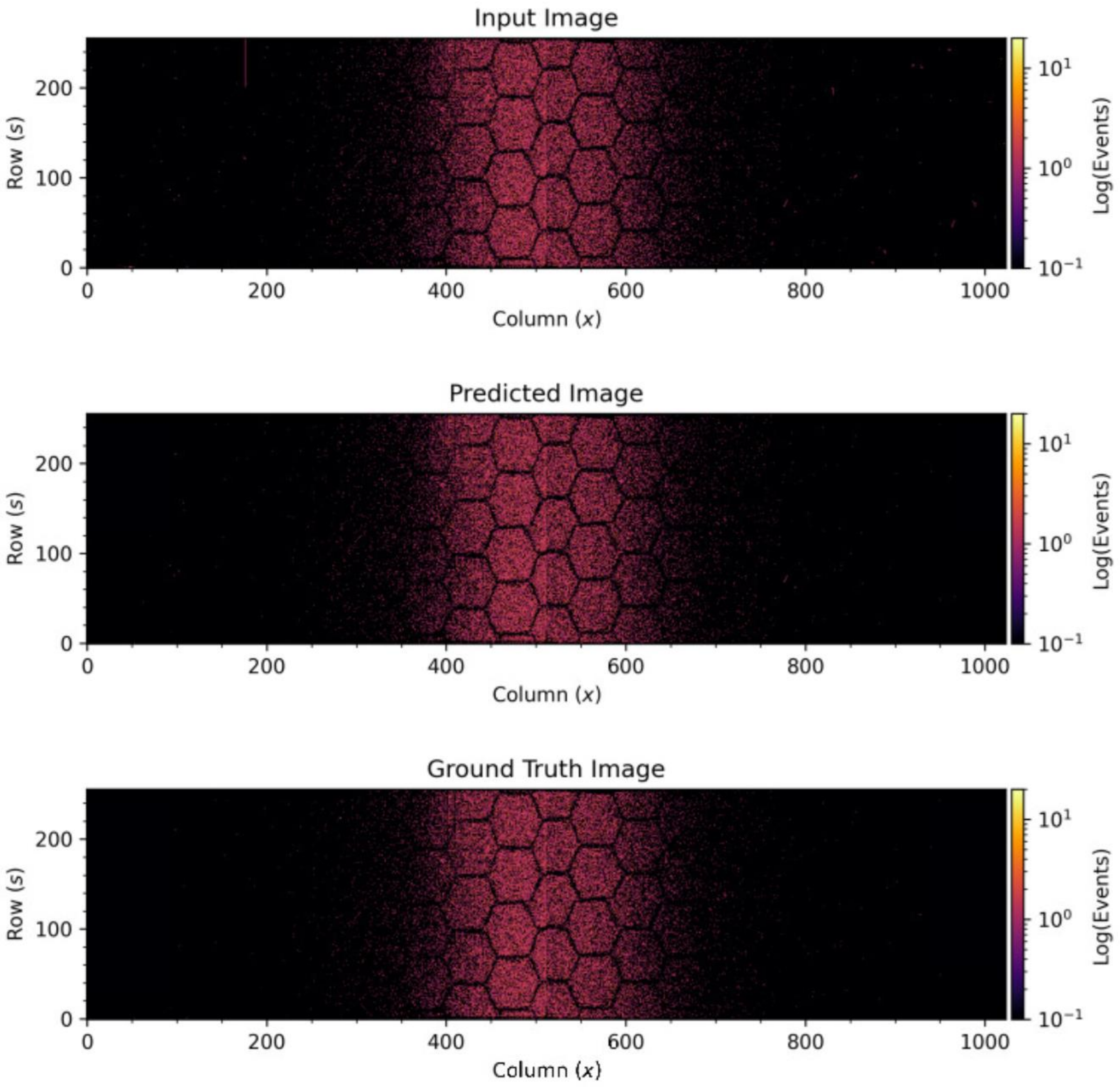

Input Image
Row (s)
Column (x)
Log(Events)
Predicted Image
Row (s)
Column (x)
Log(Events)
Ground Truth Image
Row (s)
Column (x)
Log(Events)

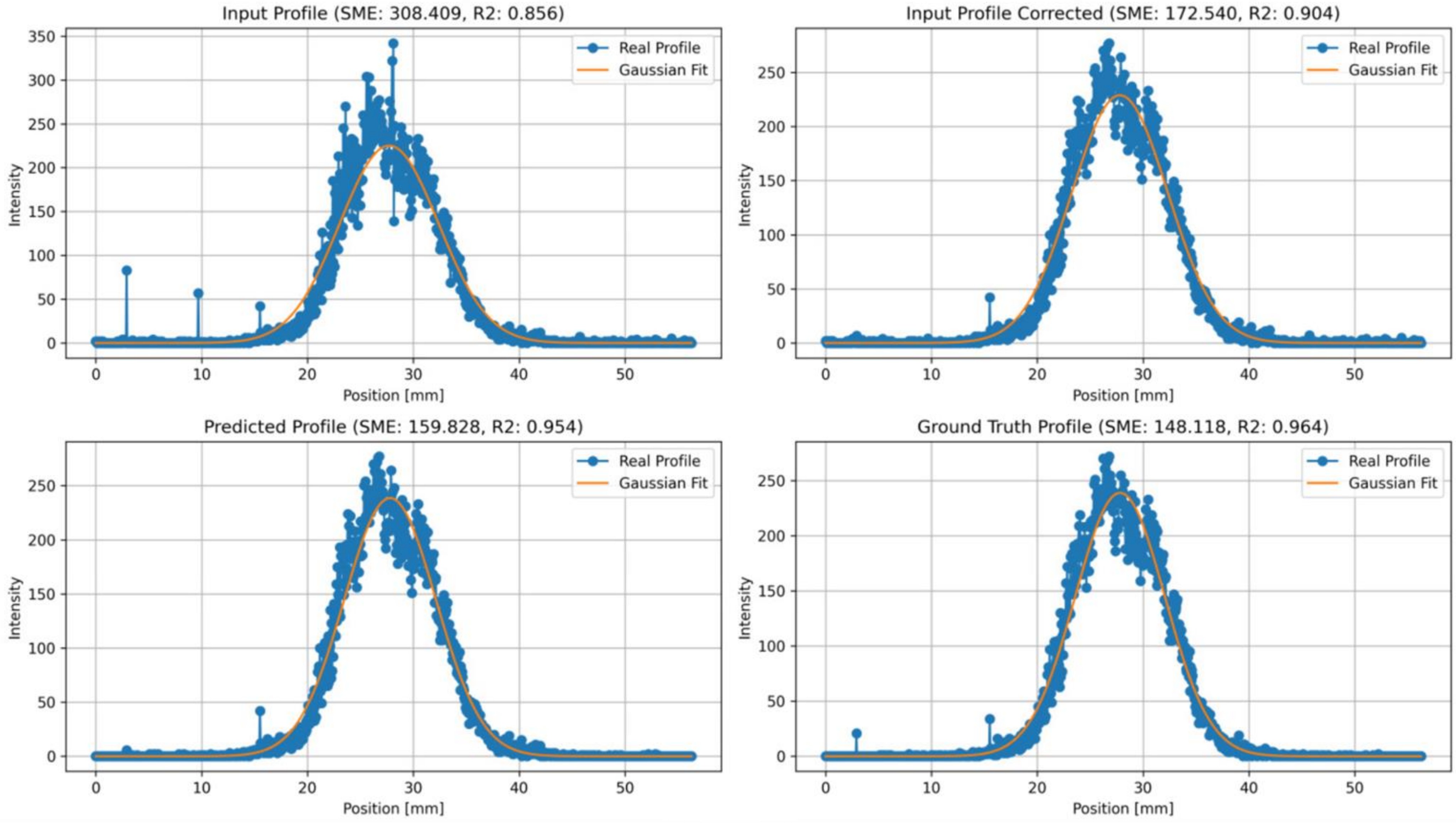

Predicted Mask
Row (s)
0
50
100
150
200
250
0
200
400
600
800
1000
Column (x)

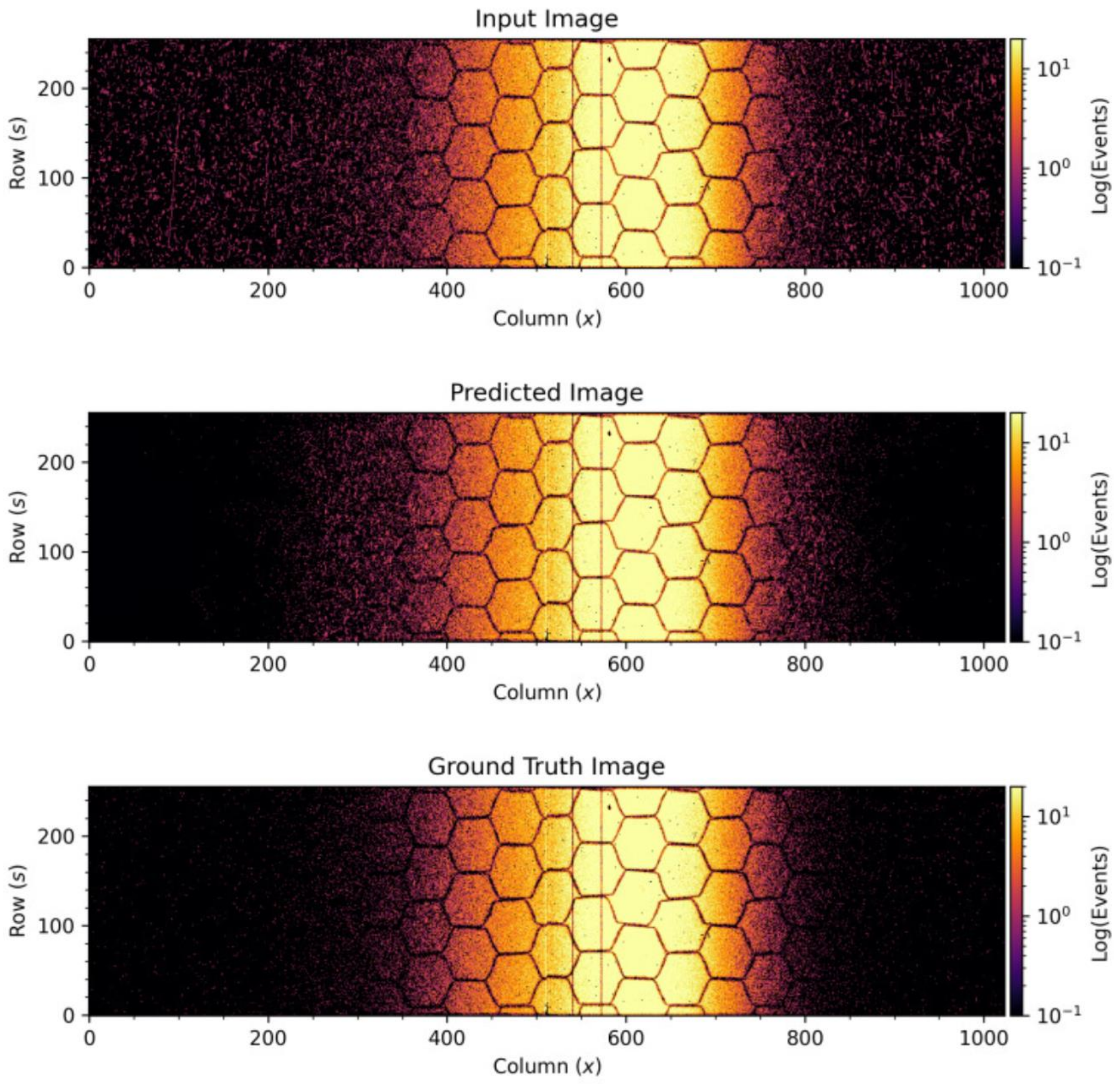

Ground Truth Mask
Row (s)
0
50
100
150
200
250
0
200
400
600
800
1000
Column (x)

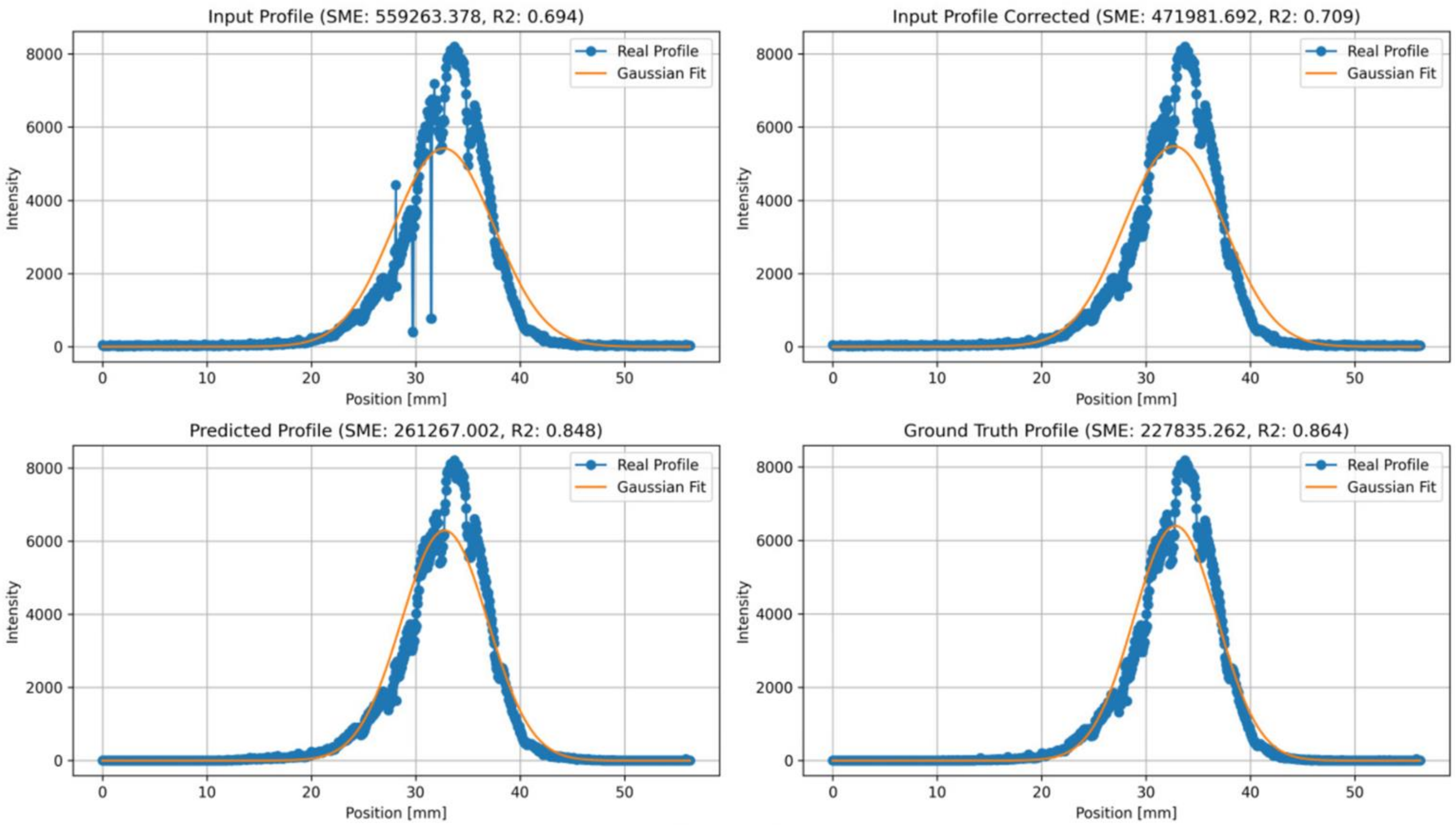


Input Profile (SME: 1125.140, R2: 0.674)
Real Profile
Gaussian Fit
Intensity
Position [mm]
Input Profile Corrected (SME: 823.793, R2: 0.713)
Mov - Avg Correction
Predicted Profile (SME: 212.986, R2: 0.900)
Ground Truth Profile (SME: 173.338, R2: 0.971)


Input Image
Row (s)
Column (x)
Log(Events)
Predicted Image
Row (s)
Column (x)
Log(Events)
Ground Truth Image
Row (s)
Column (x)
Log(Events)


Input Profile (SME: 308.409, R2: 0.856)
Input Profile Corrected (SME: 172.540, R2: 0.904)
Predicted Profile (SME: 159.828, R2: 0.954)
Ground Truth Profile (SME: 148.118, R2: 0.964)
Real Profile
Gaussian Fit
Intensity
Position [mm]


Input Image
Predicted Image
Ground Truth Image
Row (s)
Column (x)
Log(Events)
0
100
200
400
600
800
1000
10^{1}
10^{0}
10^{-1}


Input Profile (SME: 559263.378, R2: 0.694)
Input Profile Corrected (SME: 471981.692, R2: 0.709)
Predicted Profile (SME: 261267.002, R2: 0.848)
Ground Truth Profile (SME: 227835.262, R2: 0.864)
Real Profile
Gaussian Fit
Intensity
Position [mm]
0
2000
4000
6000
8000
0
10
20
30
40
50