Proceedings of the 4th International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/2026.35208

Heavy-Ion Jet Reconstruction and Transverse-Momentum Recovery with Analytical, Machine Learning, and U-Net Methods

Chengyu Zhang[1*], **Yunan Chen**[2]

[1]*Wuxi Tianyi High School, Wuxi, China*
[2]*Shenghua Zizhu Academy, Shanghai, China*
[*]*Corresponding Author. Email: zcycycycy@gmail.com*

Abstract. The academic project research of the course described in this article involves the reconstruction of heavy-duty ionized jets, covering the entire process from the stage of the event to the final resolution analysis. In the absence of background correction, the thermal particles cause the momentum of the reconstruction jet to deviate from the original actual value. To assess the practical role of various correction methods, this report makes a horizontal comparison of the region-based subtraction method, the characteristics-based scalar-regression method, the image-dependent CNN regression method, and the paired image U-Net reconstruction method. This article will first explain the specific process of PYTHIA event reading. The operating steps of the thermal background injection of different particle species and the v_2 modulation mixture are then described. The FastJet clustering and matching will be followed by the construction of matching jet tables. The follow-up will also involve feature learning, jet image learning, and specific processing of image-to-image denoising. The effect of the scalar-regression branch will be evaluated with mean squared error MSE, mean absolute error MAE, R^2, and $\sigma(\Delta p_T)$, where $\Delta p_T = p_T^{pred} - p_T^{true}$. The evaluation of the U-Net branch is carried out at the image level, using indicators such as image-space MSE, MAE, accuracy of peak positions, and correlation. The report synchronizes the actual performance of the Area-based approach, the deep neural network DNN method, and the CNN method. The report also includes feature-based benchmark results, as well as images of U-Net reconstructions.

Keywords: Heavy-ion collisions, jet reconstruction, background subtraction, machine learning, U-Net

1. Research background

1.1. Scientific motivation

This article mainly does the research on quark-gluon plasma, and the research work relies on relativistic heavy ion collisions to carry out. Jets are valuable probes in such studies, all of which are generated by the hard scattering process, which then passes through the entire reaction medium. The phenomenon of their energy loss, which we observe, and shape changes, is generally discussed in the concept of jet quenching.[1,2] In this specific research project, the practical challenges that need

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.35208

to be solved are more specific than the previous common research. The hot background particles are mixed with hard jet and do not exist alone. Therefore, the transverse momentum of the reconstruction must first make the corresponding adjustment, and the adjustment can be completed before it can be compared with the real jet.

1.2. International and domestic research status

This article now does the survey of jet background correction, which includes the analysis of subtraction methods, as well as the combination of various data-driven reconstruction technologies. Region-based subtraction methods, composition-level subtraction methods, and tools such as SoftKiller can produce interpretable baseline results to eliminate build-up problems or redundant background interference. However, if the fluctuations between different events are relatively large, the remaining residuals after these methods are treated may increase.[3,4,5] In recent research on heavy ion-related research, we have also used neural models and image models in the work of jet correction and background subtraction to see the actual role. These models include deep neural networks, also known as DNNs, and convolutional neural networks, or CNNs, which are regressions made with different image reconstruction strategies.[6,2,7]

1.3. Project positioning

This paper divides the research methods used into three main groups, namely, analytical rectification, feature-based scalar regression, and image-based learning. When we do the synchronization core comparison, we will use the regional baseline as the basis, using the deep neural network, that is, the DNN, and the convolutional neural network, which is the CNN, which is used to do the calculation related to the direct jet-by-jet p_T regression. On the matching jet table, we also used several complementary technologies, such as linear regression, random forest, extreme gradient enhancement/general-purpose enhancement regression, that is, XGBoost/GBR, and multi-layer perceptron, MLP. We separate the U-Net to set up, mainly because it is to reconstruct the image, not to predict the momentum of a single scale.

2. Technical scheme and methodology

2.1. Data sources and task definition

The research in this paper uses four interrelated data products, namely, no processed PYTHIA event records, matching completed jet tables, jet-image tensors of three different channels, and paired image tensors dedicated to reconstruction work. The original event generated by PYTHIA 8.3 will be used as a particle-level signal sample for subsequent research.[8] After that, we will superimpose a hot background in these original events to simulate the interference in the actual environment. The matching jet table belongs to the region-related baseline content and is the primary input material for scalar regression of the Area-based baseline, the synchronized DNN branch, and subsequent feature-based machine learning benchmarks. The jet-image tensor is given to the CNN regressor, and the paired image tensor is left to the U-Net reconstruction branch. In this report, the scalar-regression mission is to retrieve matching real-life momentum data p_T^{true} from observables of the tainted jet. The purpose of image reconstruction is to convert a jet image contaminated with a hot background into a real image without a background signal. The output of these two tasks is completely different, so their indicators are stored separately in different result tables for subsequent review and analysis.

DOI: 10.54254/2753-8818/2026.35208

2.2. Algorithm selection

The project described in this article starts from a set of clear correction links, and then switches to the corresponding learning mode after taking this step. The region-based subtraction method is preserved as a physical baseline, and the method itself fully follows the core concept of the FastJet activity area, and does not require any additional training work when actually used.[3,4] The table model used calls for matching jet observables that have been matched, which contain unprocessed p_T data, corresponding regional parameters, background density values, quantitative statistics, and substructure markers. Deep neural networks, that is, DNNs, linear regression models, random forest models, XGBoost or general-purpose enhanced regression, that is, GBR, and multi-layer perceptrons, MLPs, all rely on these structured functions that have been sorted to achieve corresponding performance enhancements.[6,9,10] Convolutional neural networks, also known as CNNs, are specifically used to process the representation of jet images. In this application scenario, the local spatial pattern in the image may convey additional information that can not be fully captured by manually selected variables. U-Net will be added to the paired denoising link, its encoder-decoder structure with skip connections, compared with the direct scalar regression method, better adapts to the image-to-image processing task.[11,12,7]

2.3. Method comparison

Table 1 summarizes the role of each method.

Table 1. Method comparison in this project context.

Method	Strengths	Limitations	Use case
Area-based	Interpretable and stable; does not require training	Limited nonlinear response under large background fluctuations	Analytical baseline and fast correction
DNN	Neural regressor for structured observables with nonlinear response	Depends on feature quality and does not explicitly encode spatial locality	Main tabular neural baseline in the synchronized comparison
Linear Regression	Very interpretable, fast to train, good as sanity-check baseline	Only linear response and limited robustness to complex residual structure	Quick benchmark on matched-jet features
Random Forest	Captures nonlinear feature interactions and is robust to feature scaling	Prediction can be less smooth and model size can grow quickly	Nonlinear tabular regression with moderate engineering cost
XGBoost/ GBR	Good boosting-based accuracy and useful bias correction for residual learning	Hyperparameter-sensitive and less transparent than linear models	Optimized tabular benchmark in the later run
MLP	Standard feedforward benchmark, easy to compare with classical models	Still feature-dependent and can underperform if tuning is weak	Secondary neural benchmark on tabular features
CNN	Learns local spatial features and can improve resolution on image inputs	More data-hungry and more prone to overfitting on small samples	Jet-image based high-dimensional regression
U-Net	Image-to-image denoising; skip connections help preserve fine jet structure	Higher parameter count and not a direct scalar regressor by itself	Auxiliary paired-image reconstruction and background-suppression diagnostics

DOI: 10.54254/2753-8818/2026.35208

2.4. Implementation workflow

The experimental process in this paper begins with a complete record of events generated by PYTHIA. The thermal background is incorporated by sampling the corresponding particle masses from the orientation angle of the multi-species model and the ϕ modulation distribution. We then used FastJet's anti-k_T algorithm, activity area assessment, and median ρ assessment to complete the reconstruction of the jet in a contaminated event.[13,3] After the entire process of geometric truth-reco matching is completed, the matching sample will serve as a common input for all the scalar-regression branches. In the analysis branch, we calculate the subtraction results based on the area. The role of the tabular branch is to train deep neural networks, that is, DNNs, and classical machine learning ML regressors. The image branch will train the convolutional neural network, the CNN scalar regressor, and the attention U-Net denoiser. At the end of the entire process, we will export the required indicators, residual plots, resolution curves, and multi-channel examples of these materials.

3. System design

3.1. System organization

The overall workflow organized in this article can be divided into six separate modules. In the initial event preparation phase, the original particle records need to be collected in advance, and then these records are used to complete the creation of the thermal background. Subsequent jet reconstruction and matching modules will use proven technologies such as FastJet clustering, active-zone extraction, background-density assessment, truth and reconstruction data correlation. After the data conversion module, it will first get a matching jet, and then convert these jets into scalar-regression input content, as well as a supporting standardized guide. A specialized model training script includes feature regression, a synchronous deep neural network, that is, the DNN branch, and the convolutional neural network, which is the CNN branch. This processing path will be paired with multi-channel images, which will then be used to train the U-Net to reconstruct the model. The final step of the entire workflow is to collect the indicators obtained by each link and then produce the corresponding comparative figures for the final report.

3.2. Data flow and script architecture

The main data flow designed in this paper is not integrated into a single comprehensive script, but is split into multiple sequentially advanced analytical stages, each with corresponding processing goals and verification links. In the initial processing stage, we will first retrieve all the original events that need to be used, and the relevant thermal background data will be merged into the processing process, and then use the FastJet tool to group the jets, and finally record all the successfully matched jet samples for subsequent use. Next, we will do area-based subtraction operations and compare the processing effects of different technologies, such as linear regression, random forest, XGBoost/GBR, MLP, and synchronous DNN branches, which use common scalar characteristics when running, without additional individual building feature systems. When using the CNN pathway, we will use the same test index to do image-based scalar regression, and will also use the same index to carry out resolution analysis to ensure that the basic conditions of the two groups of analysis are exactly the same. In the final reconstruction path, we will first build a paired image tensor, then use these data to train the attention U-Net model, and finally generate the corresponding channel-wise image index, as well as the corresponding graphic representation, so as to facilitate the subsequent intuitive viewing of the analysis results.

DOI: 10.54254/2753-8818/2026.35208

3.3. Design decisions

The model comparison work carried out in this paper is managed by three well-determined choices in advance. The scalar-regression branch will always adhere to the 80/20 data division rule and will also store a common test index to ensure that all subsequent model comparisons are consistent. When we select the analytical reference point, we choose two indicators: activity area and median ρ, without using the previous common constant-area simplification method. Each mode of operation has its own unique normalization method. The processing of the scalar feature is independent of the channel processing, which is adapted to the reasonable percentile normalization requirements of the U-Net tensor. So, the corresponding score of the scalar regression and image-space denoise will display it separately in a separate table.

4. Experiments and analysis

4.1. Experimental setup and effective sample size

The synchronous matching jet data set for direct scalar regression in this paper has a total of 12,066 samples, and the source and annotations of these samples have been verified by the previous basic verification to meet the basic model training requirements. This paper makes a whole division of the data set according to the fixed 80/20 ratio. The training and verification set contains a total of 9,652 jets related data, and the remaining 2,414 jets are divided into test sets, which will not participate in the training process throughout the process, and finally test the generalization ability of the model. The paper also set an additional one-fifth of the validation ratios in 9,652 samples on the training side, and 7,721 training samples were split, and 1,931 validation samples were verified to assess the current effect of the model after each round of training. In the initial synchronous comparison experiment, the DNN branch did a total of 200 epochs of training, the batch size of the training was set to 256, and the learning rate was also specially adjusted according to the characteristics of the branch. The training settings of the CNN branch and the DNN branch have some differences. The CNN branch has done a total of 150 epochs of training, the batch size is set to 128, the learning rate used is consistent with the DNN branch, and the early termination rule is also set during the training process to prevent the model from being fitted. The experiment also set up a feature-based benchmarking method that reused the same matching jet table and shared the same test set split results with the previous model to ensure the fairness of the experimental comparison. This benchmark approach integrates common algorithms such as linear regression, random forests, XGBoost/GBR, and MLP to form an effective comparison with subsequent deep learning models. The U-Net branch deals with paired three-channel 32×32 images, and the preprocessing and pairing rules for images are unified with the previous dataset settings. The U-Net branch is trained using up to 10,000 event data, and the training set and validation sets are also split in an 80/20 ratio, consistent with the previous model split logic. This article also sets up an online enhancement strategy for the U-Net branch, optimizer uses AdamW, learning rate scheduling uses warm-cosine scheduling, and the loss function combines focal loss, L1 loss, and SSIM loss to deal with the training problems caused by sparse jet image targets.

4.2. Core quantitative results

Table 2 gives the results data corresponding to the initial synchronous scalar regression, and Table 3 also lists the corresponding training indicators throughout the training process. These two tables are the core reference content of the whole report, we do this part of the comparison of the baseline is based on the Area-based set, but also the synchronous deep neural network, synchronous convolu-

DOI: 10.54254/2753-8818/2026.35208

tional neural network, CNN branch in the same experimental conditions to do the comparison, the data split rules used are exactly the same, the subsequent evaluation program did not make any changes.

Table 2. Core regression metrics for the synchronized Area-based/DNN/CNN comparison.

Method	**MSE**	**MAE**	R^2	$\sigma(\Delta p_T)$
Area-based	686.455	20.960	0.153	25.975
DNN	118.893	8.226	0.853	10.863
CNN	17.568	2.174	0.978	4.189

Table 3. Selected training-stability statistics for the synchronized DNN and CNN branches.

Model	**Best Epoch**	**Best Val Loss**	**Final Train Loss**	**Final Val Loss**
DNN	197	0.01428	0.05829	0.01336
CNN	68	0.00666	0.01148	0.00677

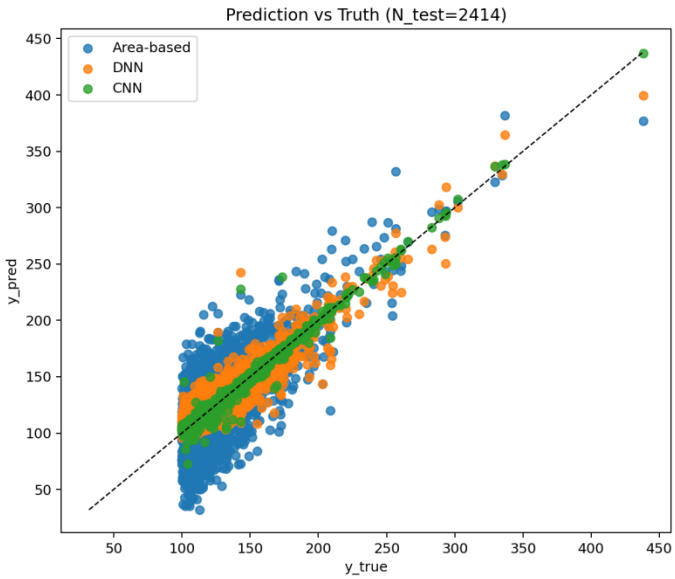

Figure 1. Prediction-versus-truth comparison for the synchronized Area-based, DNN, and CNN branches.

The figures collated in this article can help us to visualize the data in Table 2. In Figure 1, the predicted results of the convolutional neural network CNN output as a whole are the closest to the diagonal position. In Figure 2, the corresponding residual distribution of CNNs is the narrowest of all models, and the results obtained by using the region-based baseline method have a much wider range of residual numerical distributions. Figure 3 is specifically designed to show the comparison of the effects of the same model in the true-p_T bins grouping above. The test results of deep neural networks have improved compared to the previously used analytical baseline. However, in this synchronized model run test, the overall resolution given by CNN is the smallest of all the models that participated in the test.

DOI: 10.54254/2753-8818/2026.35208

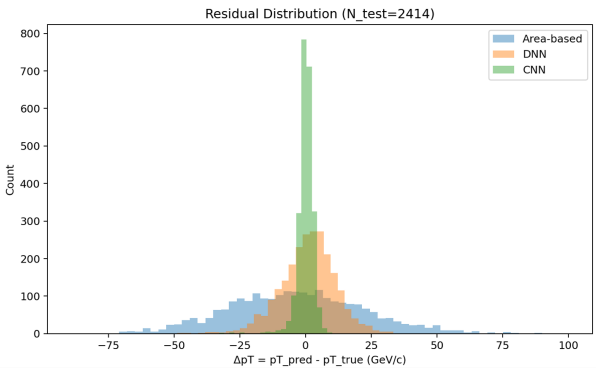

Figure 2. Residual histogram comparison for the synchronized scalar-regression branches.

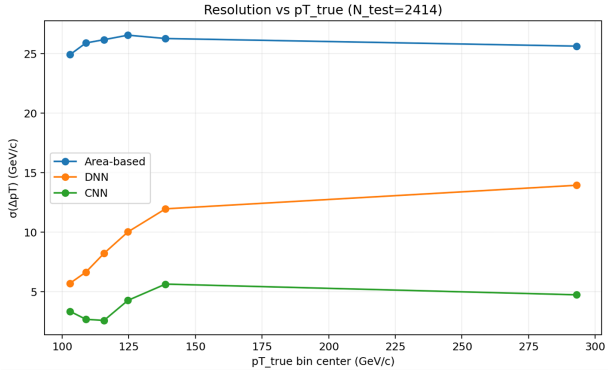

Figure 3. Resolution $\sigma(\Delta p_T)$ versus p_T^{true} bins for the synchronized comparison.

4.3. Extended feature-based benchmark

In the process of expanding scalar-regression operation, the classic regression and machine learning regression are integrated into the same matching jet feature table. We have compiled all the results obtained from this operation; the specific content is listed in Table 4, you can directly check the content of the corresponding table. These new results are not used to replace the old data already in Table 2; we think of them as a supplementary extension of the original data. This setting occurs because this specific operation process uses an optimized baseline, and a larger reference family is used to test, which is significantly different from the previous operating parameters.

Table 4. Extended feature-based benchmark from the optimized repository run.

Method	**MSE**	**MAE**	R^2	$\sigma(\Delta p_T)$
Area-based optimized baseline	61.347	5.760	0.902	7.472
Linear Regression	34.308	4.580	0.945	5.854
Random Forest	35.359	4.425	0.943	5.945
XGBoost/GBR	38.744	4.705	0.938	6.223
MLP	45.262	5.392	0.927	6.726

After combing through the long-term experimental data, this paper found that the overall performance of the four trained table models is much better than the optimized analysis benchmark. Figure 4 visualizes the same benchmark trend across true-p_T bins. The results of the linear regression model and the random forest model output are close to the reference standard 5.9 GeV/c. In all labeled true-p_T bins, the output values of the boost-regression branch are basically kept within the same error range. A multi-layer perceptron is often referred to as MLP, and the optimization analysis benchmark does have a certain performance improvement, but in the course of this specific experiment, its performance has not yet exceeded the best performance of the table models. The specific values obtained by these experiments also show that matching the jet feature table has saved a lot of valuable correction data before we carry out the image-based learning step, which can provide a good data basis for the subsequent experimental links.

DOI: 10.54254/2753-8818/2026.35208

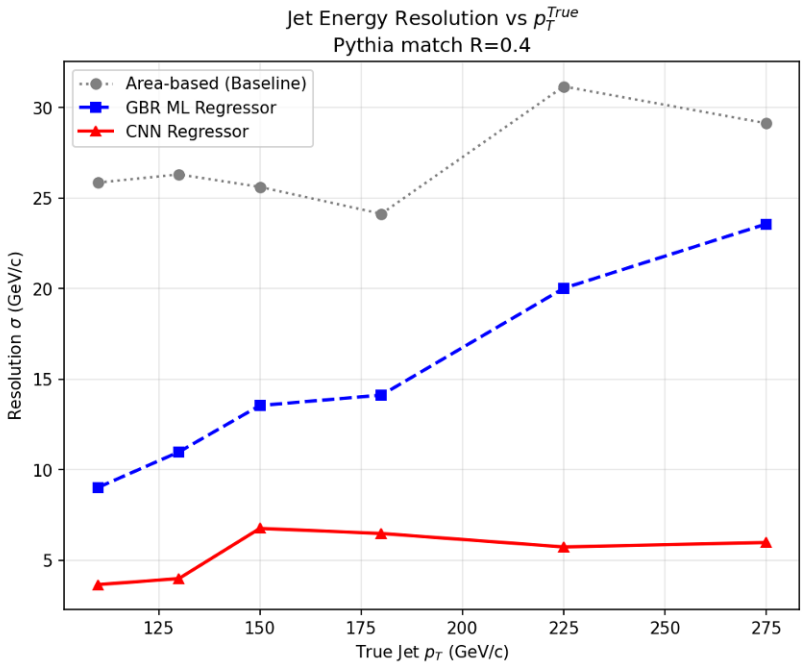

Figure 4. Resolution overlay versus true p_T for the optimized feature-based benchmark.

4.4. U-Net reconstruction diagnostics

This article locates the U-Net branch as an auxiliary test module for image reconstruction, not for direct scalar regression. It generates a three-channel image that removes the interference-free surface, preserving the key image information we need. The evaluation of this branch focuses on seeing whether the predicted image can restore the original structure of the real image. The path-by-channel diagnostic results in our previously saved evaluation summary have been sorted into the corresponding Table 5. The peak accuracy measure is used to determine whether the pixel with the highest predicted value falls within the range of two pixels around the true value of the highest pixel. When we selected the model, we completely followed the original design ideas of the original U-Net and Attention U-Net, and did not make any extra changes. The idea for this application comes from recent research on image reconstruction in the direction of heavy ion background subtraction.[11,12,7]

This article discusses the relevant conversations about U-Net as briefly as possible, because the main focus of this part of the research is on scalar-regression comparison. In Table 5, the relevant values of the Total-p_T channel show the lowest mean square error MSE, and also get the highest correlation value. For this model, the two parts of Multiplicity and Charged-p_T are more challenging than other content. This is completely consistent with their own sparse image structure. Figure 8, Figure 9, and Figure 10 provide representative reconstruction and training examples in the supplementary appendix.

Table 5. U-Net reconstruction diagnostics on normalized multi-channel jet images.

Channel	MSE	MAE	Peak Acc.	Corr.
Total p_T	0.000032	0.001688	0.8290	0.8363
Multiplicity	0.000417	0.008130	0.5875	0.5311
Charged p_T	0.001206	0.007974	0.5785	0.5735

DOI: 10.54254/2753-8818/2026.35208

5. Course reflection and future work

5.1. Learning outcomes

This related work carried out in this article has accumulated practical experience from the starting point of the research to the end of the whole process, and at the beginning, we must first determine the specific query content that matches the physical direction, and then step by step, and finally get a set of complete plots that have been verified by multiple rounds. In the course of my research, I did a series of different specific tasks, including the reading of event data, the introduction of thermal background operations, the FastJet reconstruction step, and the matching work between the truth data and the reconstruction data, the scalar-regression model construction process, and the subsequent image reconstruction and various kinds of measurement extraction work. We evaluated and analyzed the actual performance of the baseline method, the table-like machine learning method, the CNN regression method, and the U-Net denoising method, which is an unwise choice to judge whether the method is good or bad based on a single comprehensive measure. If you are dealing with exactly the same tasks, these different research methods use a unified common data splitting method to process the data. If there is a difference in the output type of different methods, it is necessary to use different diagnostic measures to make an effective judgment.

5.2. Future directions

The current model of this article is only running tests at a limited course scale, and many functions are not fully open for the time being, leaving a lot of room for adjustment. In the subsequent iterations, we can add Monte Carlo Dropout or the integration method to integrate the relevant content of the uncertainty estimation into the system, so that the model output is more referenced. Images processed with a U-Net structure can also be directly fed into downstream jet-level regression, providing more appropriate input data for subsequent regression calculations. We can also test weakly supervised application scenarios, which can match the number of real labels in such scenarios, and verify the actual performance of the model when the data is insufficient. We can also try more complex model architectures, such as Swin Transformer, Graph Neural Networks, or GNN, or hybrid CNN-Transformer backbone networks, which may further improve the overall performance of the model. However, the premise of these attempts is that we must first split the baseline, pre-processing process, and residual inspection, and adjust them correctly in all models, the follow-up tests will not get accurate results.[2,7]

References

[1] Sahoo, N. R. (2026). Jet quenching in heavy-ion collisions at RHIC and the LHC experiments. *Journal of Subatomic Particles and Cosmology*, 100384. https://doi.org/10.1016/j.jspc.2026.100384

[2] Zheng, S., & Liu, J. (2024). Review of deep learning in high-energy heavy-ion collisions. *Symmetry, 16*(11), 1426. https://doi.org/10.3390/sym16111426

[3] Cacciari, M., Salam, G. P., & Soyez, G. (2008). Pileup subtraction using jet areas. *Physics Letters B, 659*(1–2), 119–126. https://doi.org/10.1016/j.physletb.2007.09.077

[4] Berta, P., Spousta, M., Miller, D. W., & Leitner, R. (2014). Particle-level pileup subtraction for jets and jet shapes. *Journal of High Energy Physics, 2014*, 92. https://doi.org/10.1007/JHEP06(2014)092

[5] Cacciari, M., Salam, G. P., & Soyez, G. (2015). SoftKiller, a particle-level pileup removal method. *The European Physical Journal C, 75*, 59. https://doi.org/10.1140/epjc/s10052-015-3267-2

[6] Haake, R., & Loizides, C. (2019). Machine-learning-based jet momentum reconstruction in heavy-ion collisions. *Physical Review C, 99*(6), 064904. https://doi.org/10.1103/PhysRevC.99.064904

[7] Qureshi, U. S., & Elayavalli, R. K. (2025). Deep image reconstruction for background subtraction in heavy-ion collisions. *Physical Review C*. Advance online publication. https://doi.org/10.1103/dqnx-1fqk

DOI: 10.54254/2753-8818/2026.35208

[8] Bierlich, C., Chakraborty, S., Desai, N., Gellersen, L., Helenius, I., Ilten, P., Lönnblad, L., Mrenna, S., Prestel, S., Preuss, C. T., Sjöstrand, T., Skands, P., Utheim, M., & Verheyen, R. (2022). A comprehensive guide to the physics and usage of PYTHIA 8.3. *SciPost Physics Codebases, 8*. https://doi.org/10.21468/SciPostPhysCodeb.8

[9] Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

[10] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). Association for Computing Machinery. https://doi.org/10.1145/2939672.2939785

[11] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015* (pp. 234–241). Springer. https://doi.org/10.1007/978-3-319-24574-4_28

[12] Oktay, O., Schlemper, J., Le Folgoc, L., Lee, M., Heinrich, M., Misawa, K., Mori, K., McDonagh, S., Hammerla, N. Y., Kainz, B., Glocker, B., & Rueckert, D. (2018). *Attention U-Net: Learning where to look for the pancreas*. arXiv. https://doi.org/10.48550/arXiv.1804.03999

[13] Cacciari, M., Salam, G. P., & Soyez, G. (2008). The anti-k_t jet clustering algorithm. *Journal of High Energy Physics, 2008*(04), 063. https://doi.org/10.1088/1126-6708/2008/04/063

Supplementary

A Supplementary scalar-regression figures

This appendix summarizes the scalar-regression diagnostic tool to support common comparative analysis without changing the output of the tool itself. Figure 5 records the CNN training curve, Figure 6 presents a CNN-only prediction diagnostic, and Figure 7 shows the optimized benchmark residual distribution.

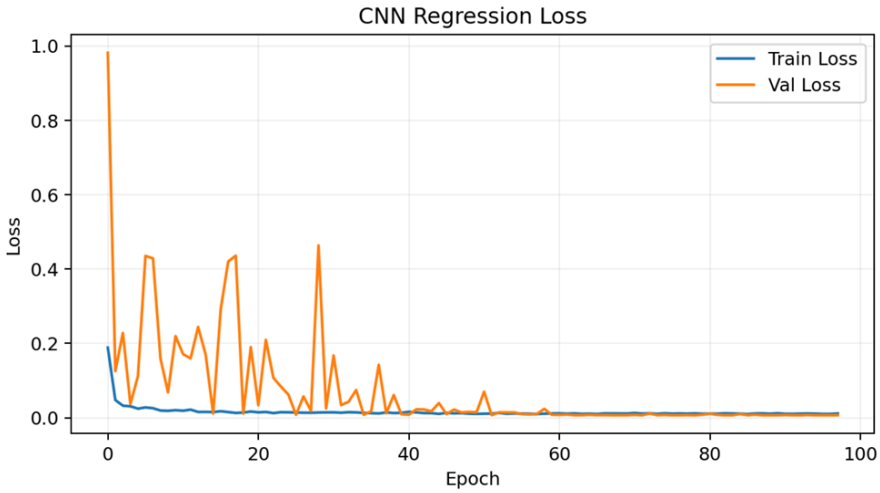

Figure 5. Scalar-regression CNN training curve.

B Supplementary image-reconstruction figures

The appendix content of this paper mainly includes a single-channel inspection process of organizing image reconstruction numbers. Figure 8 and Figure 9 show representative multiplicity and charged-p_T reconstructions, while Figure 10 gives the U-Net training curve. Figure 11, Figure 12, and Figure 13 show later-snapshot event-level reconstructions, and Figure 14, Figure 15, and Figure 16 summarize the corresponding multi-event channels.

DOI: 10.54254/2753-8818/2026.35208

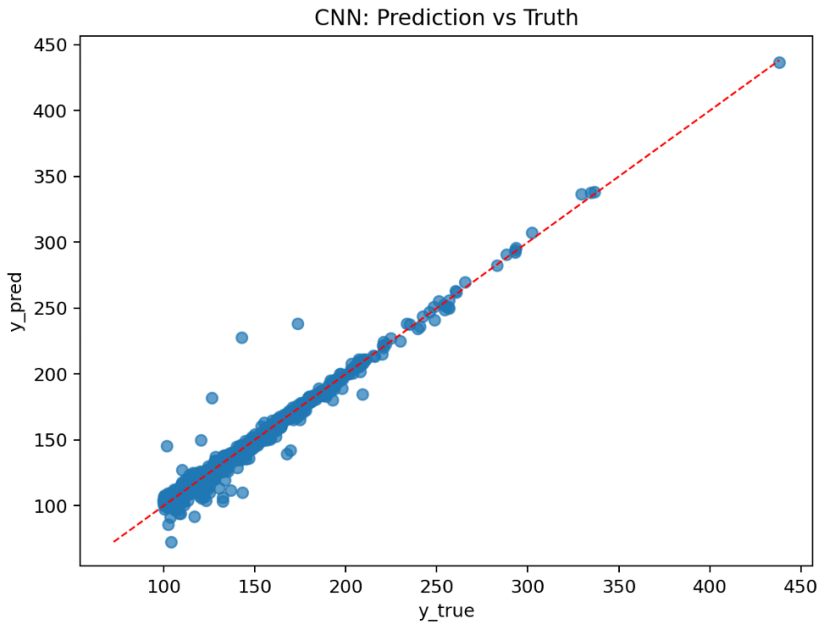

Figure 6. CNN-only prediction-versus-truth diagnostic on the shared test samples.

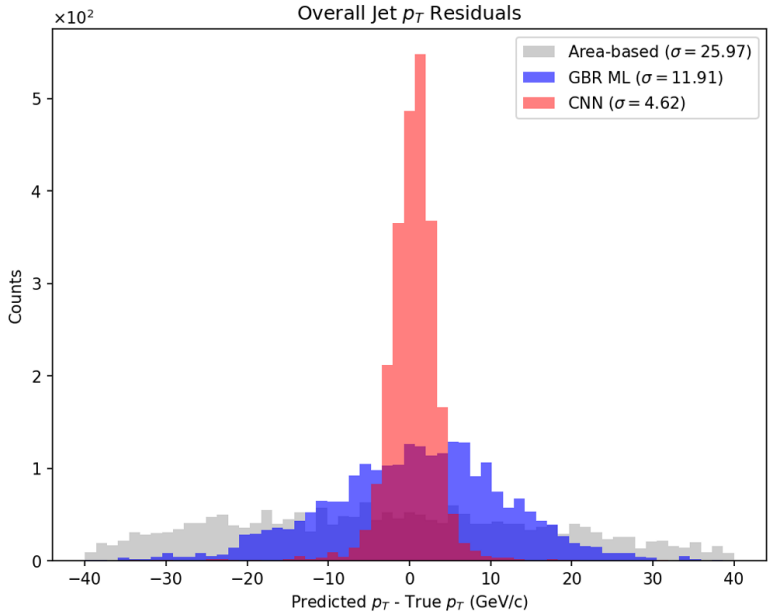

Figure 7. Optimized benchmark residual histogram after baseline and ML correction.

DOI: 10.54254/2753-8818/2026.35208

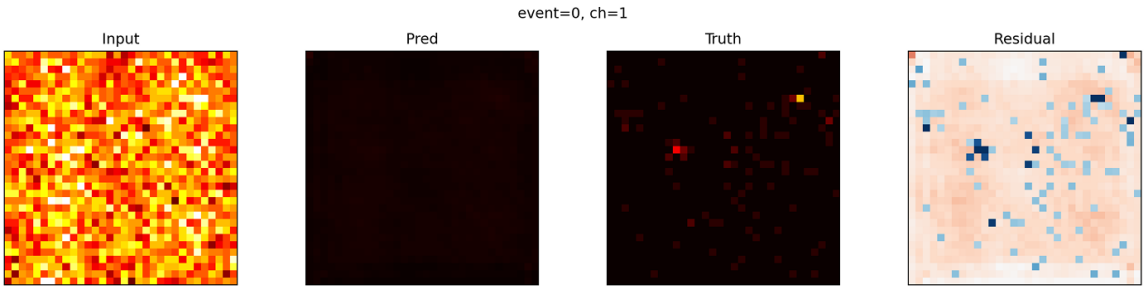

Figure 8. Multiplicity-channel event-level reconstruction example.

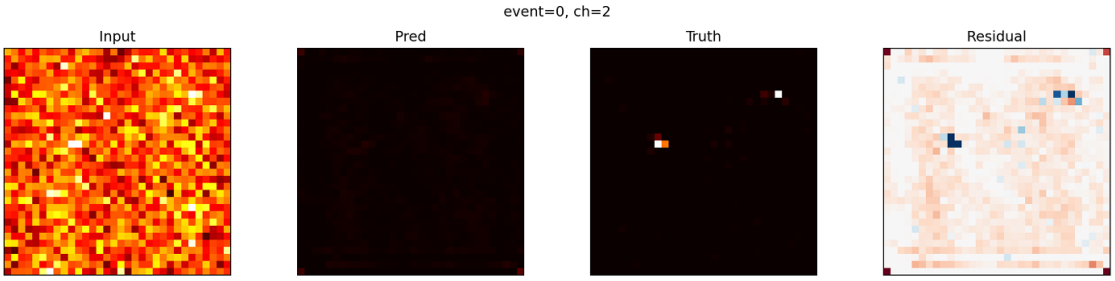

Figure 9. Charged-p_T event-level reconstruction example.

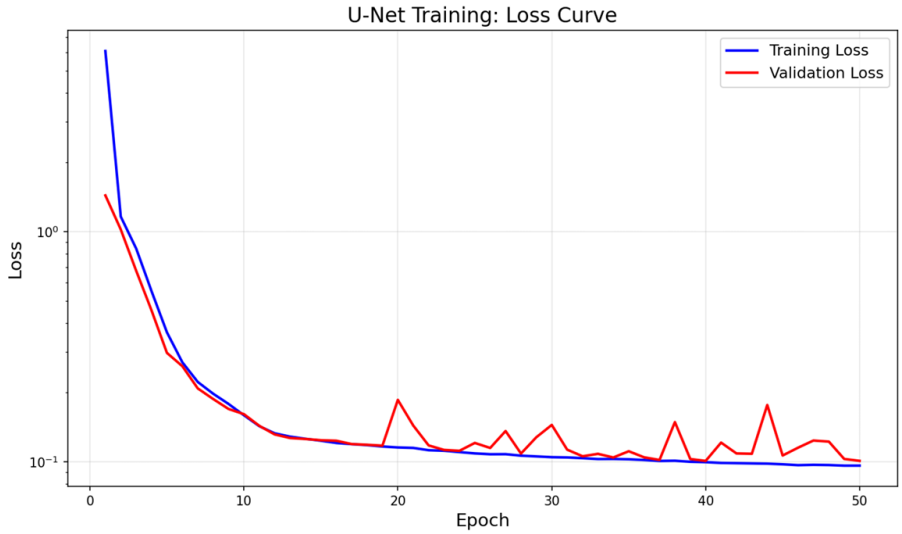

Figure 10. U-Net reconstruction training curve.

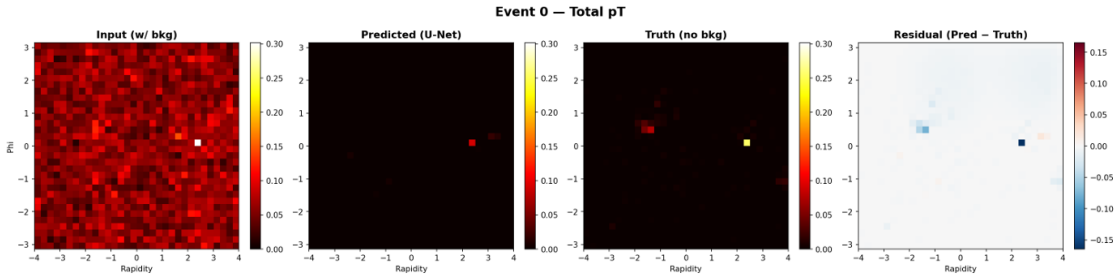

Figure 11. Later-snapshot U-Net event-level reconstruction in the Total-p_T channel.

DOI: 10.54254/2753-8818/2026.35208

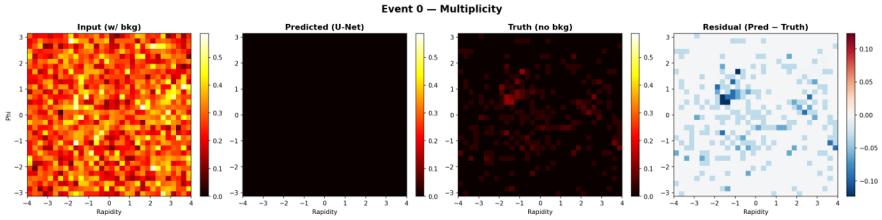

Figure 12. Later-snapshot U-Net event-level reconstruction in the Multiplicity channel.

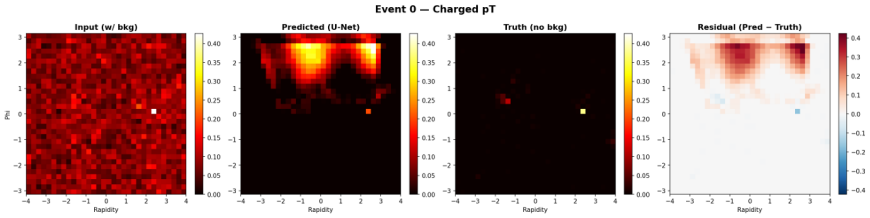

Figure 13. Later-snapshot U-Net event-level reconstruction in the Charged-p_T channel.

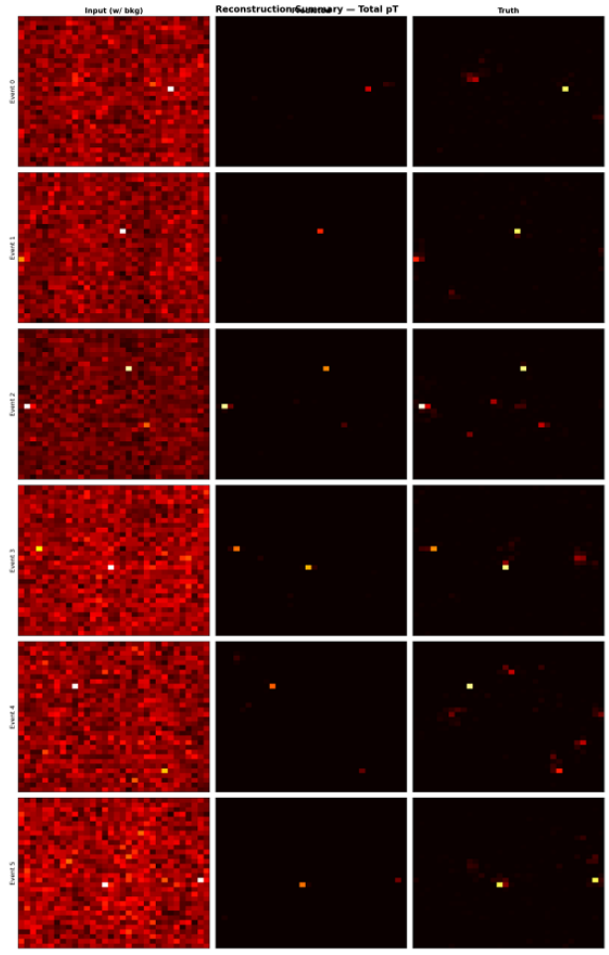

Figure 14. Later-snapshot U-Net multi-event summary in the Total-p_T channel.

DOI: 10.54254/2753-8818/2026.35208

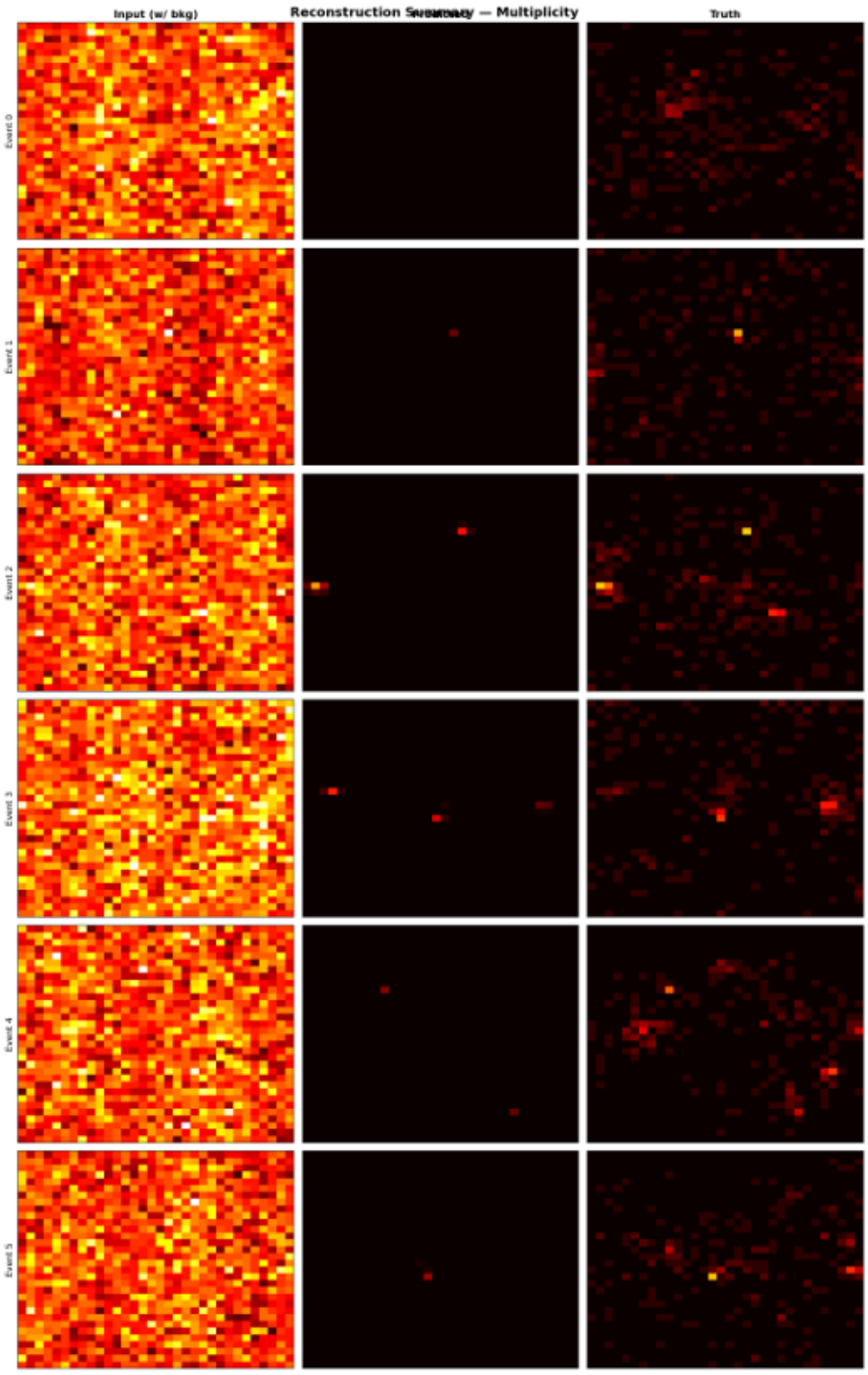

Figure 15. Later-snapshot U-Net multi-event summary in the Multiplicity channel.

DOI: 10.54254/2753-8818/2026.35208

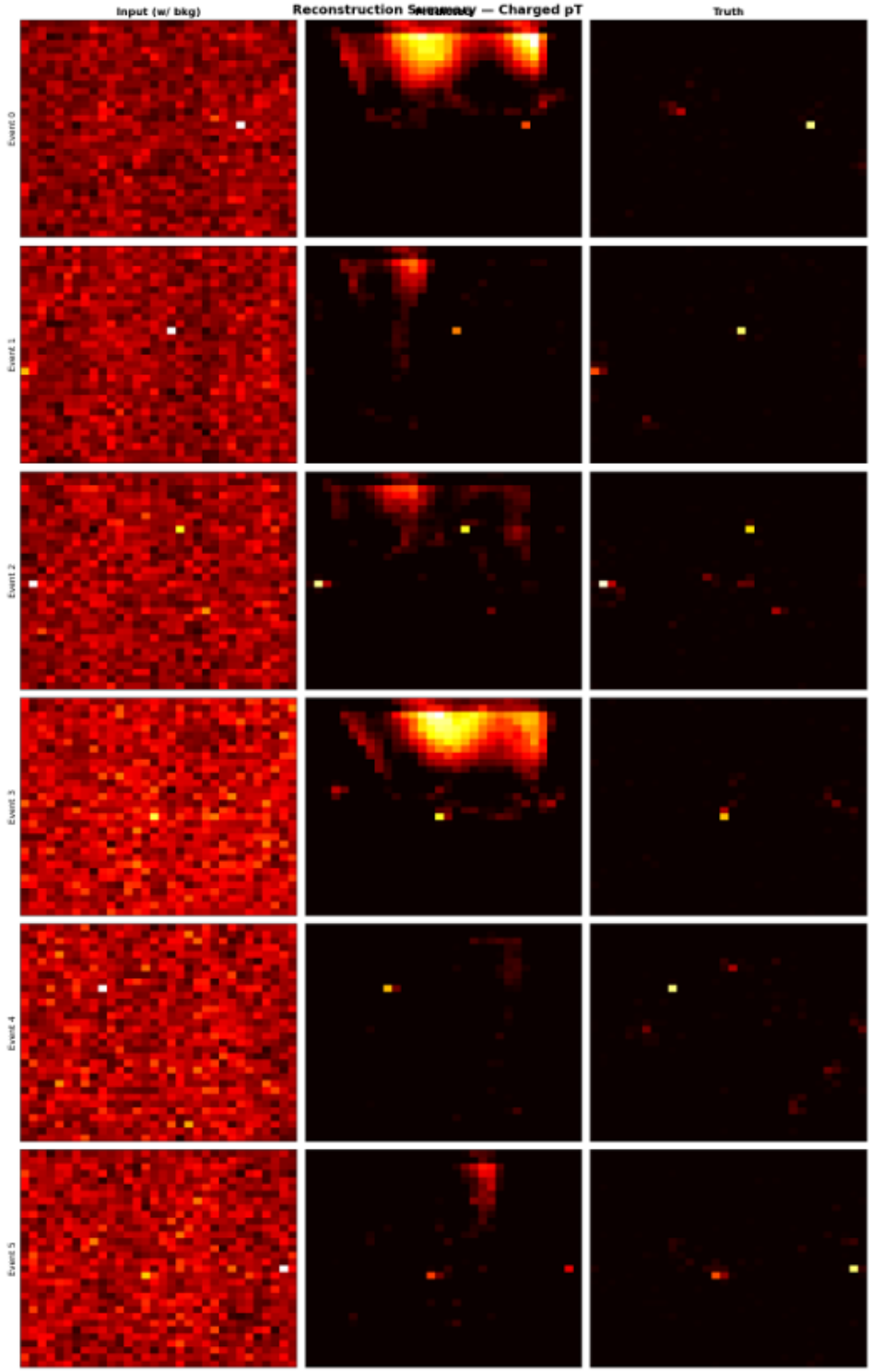

Figure 16. Later-snapshot U-Net multi-event summary in the Charged-p_T channel.