Transfer Learning with Convolutional Autoencoders for Pixel Detectors

Tomáš Čelko[1,2,*], František Mráz[1], Benedikt Bergmann[2] and Stefan Gohl[2]

[1]Department of Software and Computer Science Education, Charles University, Prague 1, Malostranské nám. 25, 11800, CZ
[2]Institute of Experimental and Applied Physics, Czech Technical University, Husova 240/5, 11000, CZ

Abstract

A standard workflow in particle physics data analysis involves developing processing pipelines on simulated datasets before applying them to experimental measurements. However, in most cases, there is a significant discrepancy between labeled simulations and real detector data, which is often noisy and unlabeled. This challenge is particularly pronounced for hybrid pixel detectors such as Timepix3, which capture tracks together with the deposited energy and trajectory of each particle. In the absence of sufficient experimental ground truth, machine learning models trained predominantly on simulation data often exhibit limited robustness. To address this issue, we propose a transfer learning framework based on convolutional autoencoders. First, a U-Net neural network is pretrained in a self-supervised manner on unlabeled experimental data to learn detector-specific noise characteristics and intrinsic structural features. The pretrained model is then fine-tuned on labeled simulation data for downstream classification and regression tasks. Analysis of latent representations via dimensionality reduction indicates that this pretraining stage accelerates convergence and enhances performance, particularly in low-label regimes. The proposed framework is validated using experimental data, including those from the Space Application of Timepix Radiation Monitor (SATRAM) mission, demonstrating its applicability to real-world radiation-monitoring scenarios.

Keywords

CNN, autoencoders, transfer learning, self-supervised learning, Timepix, pixel detectors

1. Introduction

Characterizing unknown radiation fields is a recurring challenge in space science, radiation protection, and particle physics: spacecraft must monitor radiation to protect hardware and crew, therapy facilities must monitor secondary radiation for safe treatment planning, and collider experiments must resolve particle type, energy, and trajectory.

Hybrid pixel detectors of the Timepix [1] family provide noise-free detection of ionizing radiation interacting with their semiconductor. These leave traces that are sampled with tens of μm granularity, providing 2D footprints whose characteristic patterns of per-pixel energy and time measurement allow the determination of particle type, energy, and trajectory (within some physics constraints), rather than purely providing integrated dose or particle hit rate.

The Timepix3 [2] semiconductor detector offers a 256 × 256 pixel matrix with 55 μm pixel pitch and 1.56 ns temporal binning, suited for applications from precise tracking to imaging. It records energy (time-over-threshold) and timing (time-of-arrival) per pixel, unlike Timepix [1], which lacks timing information. Pixels with similar spatial-temporal signatures are grouped into tracks, also known as clusters. These are characterized by attributes such as particle type, incidence angle, and primary energy. Such attributes are unknown during measurement and must be inferred from the shape, size, and energy profile of the cluster. Their values can be estimated analytically using expert knowledge; see, e.g., [3, 4], which sorts clusters into particle-type groups via a hand-crafted physics-based classification scheme. Such methods are interpretable and require no training data, but their thresholds must be manually re-derived whenever the sensor, particle mixture, or energy range changes, limiting generalization.

ITAT'26: Information Technologies – Applications and Theory, September 25–29, 2026, Hotel Vršatec, Biele Karpaty, Slovakia
*Corresponding author.
celko@ksvi.mff.cuni.cz (T. Čelko); frantisek.mraz@mff.cuni.cz (F. Mráz); benedikt.bergmann@utef.cvut.cz (B. Bergmann); Stefan.Gohl@utef.cvut.cz (S. Gohl)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-09

CEUR-WS.org/Vol-4323/paper9.pdf

A more recent alternative fits models directly to data. Still, these approaches typically rely on simulated experiments for ground-truth labels, since a measured track's true particle type, energy, and angle are rarely independently known. This ignores the far larger volume of unlabeled real data that every detector produces and relies on simulations that are costly to run at scale and may not fully capture detector-specific effects, like pixel noise, charge sharing, or radiation-induced degradation. A model trained purely on simulation risks transferring poorly to real data. Additionally, if there is no ground truth to check against, this gap can go unnoticed until deployment.

We use self-supervised pretraining with autoencoders to leverage unlabeled data, reducing reliance on simulation and accelerating convergence during training. A convolutional autoencoder is first trained to reconstruct unlabeled, experimentally measured Timepix3 clusters, learning detector-specific noise and structural patterns directly from real data before any labels are introduced. The pretrained encoder is then fine-tuned to predict the track properties described above. We validate this framework on data from the Space Application of Timepix Radiation Monitor (SATRAM) [5, 6] mission, showing that pretraining accelerates convergence and improves performance, particularly in low-label-count settings.

2. Related work

Convolutional neural networks (CNNs) have long dominated image processing [7, 8]: local receptive fields, weight sharing, and pooling yield translation-equivariant, hierarchical features with far fewer parameters than a fully connected network [7, 9], making CNNs the default for classification, detection, and segmentation [8, 10].

Vision Transformers (ViT) [11] have since matched or exceeded CNNs by modeling relationships between distant patches via self-attention, trading CNNs' built-in locality and translation-equivariance for biases learned from data. This flexibility only pays off after pretraining on large datasets with more parameters and higher compute costs [11, 12], a trade-off that favors CNNs in the label-scarce setting.

CNNs suit natural images because such images are (A) high-dimensional; (B) locally correlated, since smoothly varying surfaces and textures are what a local kernel exploits; and (C) geometrically invariant, as object identity is largely unaffected by position, orientation, or scale [9]. Timepix3 cluster data relate to these properties differently: (A) dimensionality is far lower, with clusters often spanning only a handful active pixels against an otherwise zero frame [3]; (B) correlation is weaker and shorter-range, reflecting real energy-deposition and charge-sharing physics, but sparser than dense photographic structure [3]; and (C) geometric invariance still holds, arguably more cleanly, since a track's type, energy, or incidence angle depends on the relative position of the sensor with respect to the interaction point. CNNs thus remain a natural fit even for small, sparse cluster data.

Previous Timepix3 attribute-prediction works exploit this fit, but, to our knowledge, remain shallow, with a few convolutional or fully connected layers rather than deep mainstream-vision architectures [13, 14]. This seems reasonable given the clusters' small extent, but it leaves open the question of whether deeper CNNs help with this lower-dimensional, sparser yet structured data.

This raises the question of supervision. Fully supervised training of a deep CNN is constrained by the scarcity of labeled experimental data. NLP faced an analogous problem, and the approach of pretraining models via self-supervision on unlabeled text is now considered standard in the field [15, 16]. Two analogous self-supervision objectives exist for images. Contrastive methods pull representations of augmented views of the same input together while pushing other views apart [17, 18]; generative methods, instead, reconstruct the input or a corrupted version of it, forcing the encoder to capture the underlying structure [19]. In this paper, we adopt the generative route.

This route is not new. Classic autoencoders have already been proposed for unsupervised dimensionality reduction [20], and denoising autoencoders for learning representations robust to input corruption by reconstructing a clean signal from a corrupted one [21]. The architecture we rely on for this task is a convolutional U-Net [22] rather than a fully connected network.

Autoencoders have already been used in detector physics, but predominantly for unsupervised

anomaly detection on high-level, low-dimensional event or jet features at collider experiments, where the reconstruction error itself, rather than the learned latent representation, is of interest [23]. Our use differs in level and purpose: we apply a U-Net autoencoder directly to raw, pixel-level Timepix3 cluster images and repurpose the learned encoder for downstream classification and regression via fine-tuning. To our knowledge, convolutional-autoencoder pretraining on unlabeled real detector data before fine-tuning on labeled simulation data has not previously been explored for Timepix-type detectors. In contrast, prior work has applied purely supervised CNNs [13, 14] rather than using self-supervision to bridge the simulation-to-real-data gap.

3. Methodology

3.1. Model overview

We adopt EfficientNetB4 [24] as the encoder backbone of our model, a well-established CNN known for achieving high accuracy on benchmarks like CIFAR-10 with significantly fewer parameters than alternatives such as ResNet [10] or Inception [25], thanks to its compound scaling strategy. Once the data passes through the encoder, it is forwarded to the decoder. The decoder is constructed by inverting the encoder structure – replacing convolutions with transposed convolutions while forming a U-shaped architecture [22]. For the supervised phase of training, we discard the decoder and replace it with a multilayer perceptron.

3.2. Inputs

To match CNN input requirements, pixel hit data are preprocessed into two channels: per-pixel deposited energy and a binary map of pixels exceeding a fixed energy threshold. While the first channel captures the essential energy information about the track, the second channel serves as an auxiliary mask, hinting to the model to pay more attention to high-energy pixels, which often indicate close proximity to the trajectory of the detected charged particle.

For the first channel, we use square-root normalization – linear normalization seems suboptimal due to the wide per-pixel energy range that a detected particle may span (from keV to MeV). In contrast, while logarithmic normalization is unable to resolve mid-energy-range pixels.

Given the fixed input size of our network (380×380 pixels, after upsampling) and the highly variable track sizes (1 to 10^4 pixels), we apply nearest-neighbor upsampling up to 16× to enhance signal strength and mitigate sparsity. A drawback of this method is the obstruction of information about the original track size (the scaling factor depends on the size of the bounding box of each track). Crucially, while this information is no longer evident from the scaled image, the model can infer the scaling factor from the pixelated/blocky behavior of the upsampling method. However, without upsampling, the sparsity of the data would make the training inefficient.

3.3. Targets

As the final step in the decoder part of the model, we attach three specialized heads: the first reconstructs the complete unmasked input, the second predicts the binary track mask, and the third identifies pixels above a fixed threshold. We address the importance of these binary masks on the output in Section 4.

During fine-tuning, the decoder is discarded and replaced with two fully connected layers. During this phase of the training, we aim to predict the chosen property of the track. For the particle category, the choice is relatively straightforward: we maximize the likelihood of the data and use the categorical cross-entropy loss with softmax activation. However, for angle prediction (see Figure 1), the situation is different. A simple option would be to predict the angle as a real number (potentially scaled to $\langle 0, 1 \rangle$) and ignore the modulo nature of the data. However, this linear mapping produces unstable gradients at the boundary of the predicted interval (predictions of 0 degrees for the 359-degree target should yield a low loss).

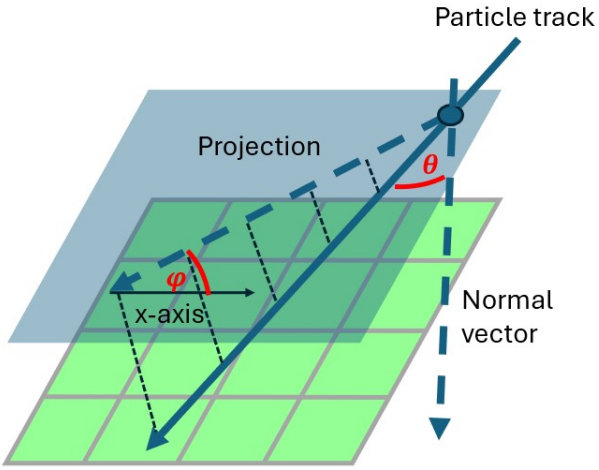


Figure 1: Scheme displaying the definition of the θ angle as the divergence of the linear track from the sensor normal vector. The angle ϕ is defined as the angle between the projection of the track onto the sensor plane and the x-axis.

Therefore, it is common to predict any angle α as a pair of $(\sin\alpha, \cos\alpha)$ instead. This captures the modulo nature of the angles and produces more stable gradients, albeit at the cost of adding a degree of freedom. When converted to polar coordinates (r, α), the new degree of freedom r can be used as a degree of certainty about the prediction.

3.4. Training

The training consists of the following steps.

1. Initialize the EfficientNet encoder with ImageNet-pretrained weights.
2. Train both the encoder and the decoder on unlabeled Timepix3 data to reconstruct its inputs. In order to prevent the model from blindly copying the data into its output, we also randomly zero out a fixed fraction of the nonzero pixels. The decoder is trained with a higher learning rate, while the encoder is updated more slowly using stepwise decay.
3. Finally, replace the decoder with a task-specific prediction head, and fine-tune the model on simulated data to predict track properties. Here, we again allow the encoder to update its weights slowly, while the prediction head weights are updated with a higher learning rate.

4. Experiments

Here, we first introduce the datasets used in the experiments. Then we analyze the model after pretraining, showing its latent-space visualization and the progress of its supervised fine-tuning. In the next section, we follow up with assessing the performance of the fine-tuned models on simulated data. And finally, for validation, we test the models on real-world measured experimental data.

4.1. Datasets

4.1.1. Pretraining dataset

The choice of training dataset was mainly driven by the effort of capturing a wide spectrum of track types, shapes, and sizes. The first part of the dataset consisted of data measured with the Timepix3 detector located within the ATLAS [26] detector at the Large Hadron Collider (LHC) at CERN. This dataset benefits from the diversity of the radiation field created during the collisions at the LHC. Despite

the wide variety of track types, one particular group seems to be underrepresented – heavy ions. These tracks often span thousands of pixels and exhibit a complex morphological structure.

This motivated the addition of the second part of the dataset, which consisted of lead ions and their fragments measured at SPS, CERN, during a test beam campaign. In order to balance the dataset, we use the pixel count to divide the clusters into exponentially-sized bins, from which we uniformly sample during training.

4.1.2. Fine-tuning dataset

For fine-tuning the model, we chose a dataset from an entirely different environment – Timepix orbiting the Earth as part of the SATRAM project [27]. This dataset was also chosen to demonstrate the generalization capabilities of the model, as the pretraining dataset was obtained using the Timepix3 chip, whereas the Timepix [1] chip was used in SATRAM experiments.

The Geant4 [28] software was used to simulate the response of protons and electrons with energies expected at the orbit. During the simulations, both the particle type and its direction (θ and ϕ angles) were recorded. Since the separation of protons and electrons has already been explored in related work [29], we consider it to be a reasonable benchmark for the performance of our model.

The simulation uses an omnidirectional field of particles. This has an important consequence on the distribution of the detected track angles. While the ϕ angle is uniformly distributed in such a field, the θ angle is not. If we picture an infinite sphere around the flat detector, only the particles originating from the poles can be recorded with $\theta = 0°$, despite having the maximum visible cross-section of the sensor from this angle. Conversely, for $\theta = 90°$, you would get the entire equatorial belt as a possible source, but then, you would get a minimal cross-section of the sensor from this region. Both of these cases turn out to be minima of the distribution, which is the product of $\sin\theta$ and $\cos\theta$ (one representing the source position and the other the visible detector cross-section). This yields $\frac{dN}{d\theta} \propto \sin 2\theta$, which implies almost no tracks present around 0° or 90°. Because of that, using an isotropic distribution, we simulated additional tracks at these angles to fill the missing data and balance the dataset.

Additionally, all tracks with a pixel on the border of the sensor are removed from the dataset to prevent the inclusion of truncated signals, which may violate the equivariance principles of our model.

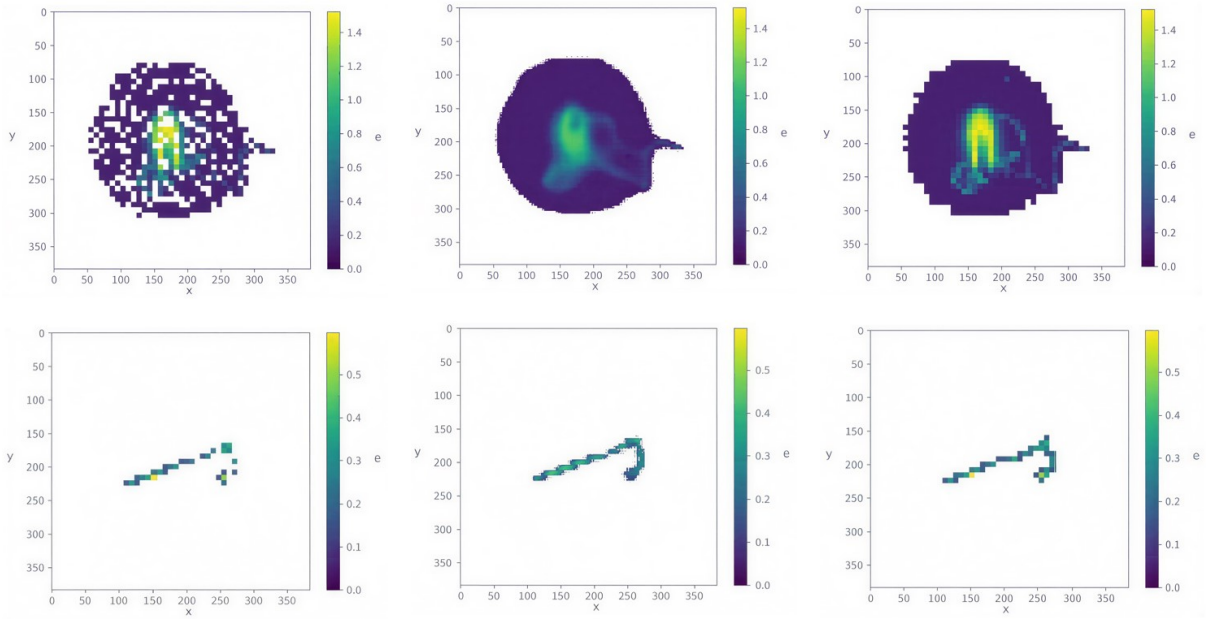
4.1.3. Real-world validation dataset

In addition to measuring performance on the simulated dataset, we assess the performance of the fine-tuned models on the real-world data from the experiments. These can be split into multiple groups as follows:

- *Proton identification.* Proton data originate from the Danish Centre for Particle Therapy in Aarhus, with beam data at 160 MeV. This dataset originally had two dominant peaks, the first consisting of gammas and the second consisting of protons. An energy cut between the peaks was applied (individually for each measurement angle) to remove most gamma rays from the dataset. However, a minor degree of impurity is likely still present.
- *Electron identification.* Electrons were measured at the National Radiation Protection Institute of the Czech Republic. These were obtained from measurements of Rn^{222} β decay and originally also contained α particle tracks, but these were filtered out because of their distinctive round shape as opposed to the curly shape of electron tracks.
- *Theta angle prediction.* Protons with known incidence angles for this experiment are the same protons as described above.
- *Proton-electron separation.* For this task, we used measured data from the SATRAM space instrument acquired over the first two months of 2015.

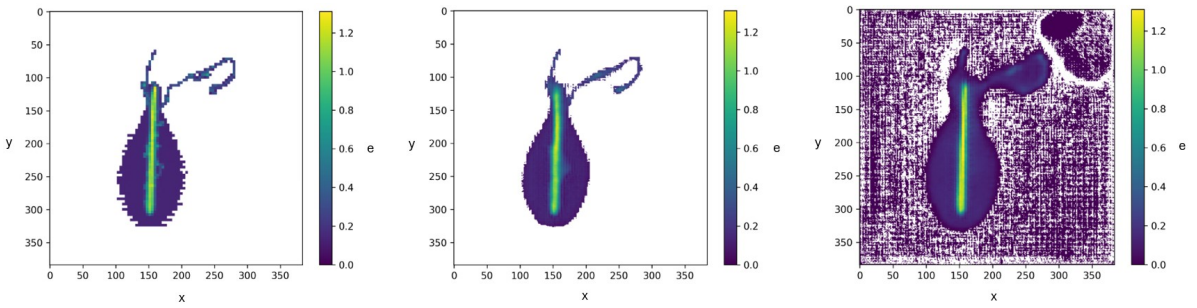
4.2. Pretrained model analysis

To gain a better insight into the state of the pretrained model, we conducted multiple experiments. Since the unlabeled pool of data is large, during pretraining it suffices to use a single training epoch



(a) (b) (c)

Figure 2: Visualization of the per-pixel scaled deposited energy of a track. (a) Partially masked input (25%) presented to the network. (b) Network output as generated by the autoencoder. (c) Target output.



(a) (b) (c)

Figure 3: (a) Heavy ion track energy deposition. (b) Reconstruction of the track by the autoencoder. (c) Reconstruction of the track without a binary output mask. The x and y dimensions are in pixels, and deposited energy is in logit form. White pixels have a value below 0.

with a batch size of 10 and 7×10^4 weight updates until convergence. A sample of the reconstructed tracks at the end of training updates is depicted in Figure 2.

4.2.1. High-frequency track details

One can see that the model reconstructs simple tracks relatively well, but it struggles with high-frequency details in more complex ions. This seems to be an architecture-specific feature of U-Nets without skip connections [30]. In segmentation tasks or other supervised learning scenarios, this is solved by simply adding residual connections between layers with matching resolution in the encoder and decoder. This would have caused the network to primarily learn identity via the skip connection, with only a small portion of the gradients flowing through the entire U-shape of the network. So residual connections would improve reconstruction quality, but they would also make learning the representation much harder.

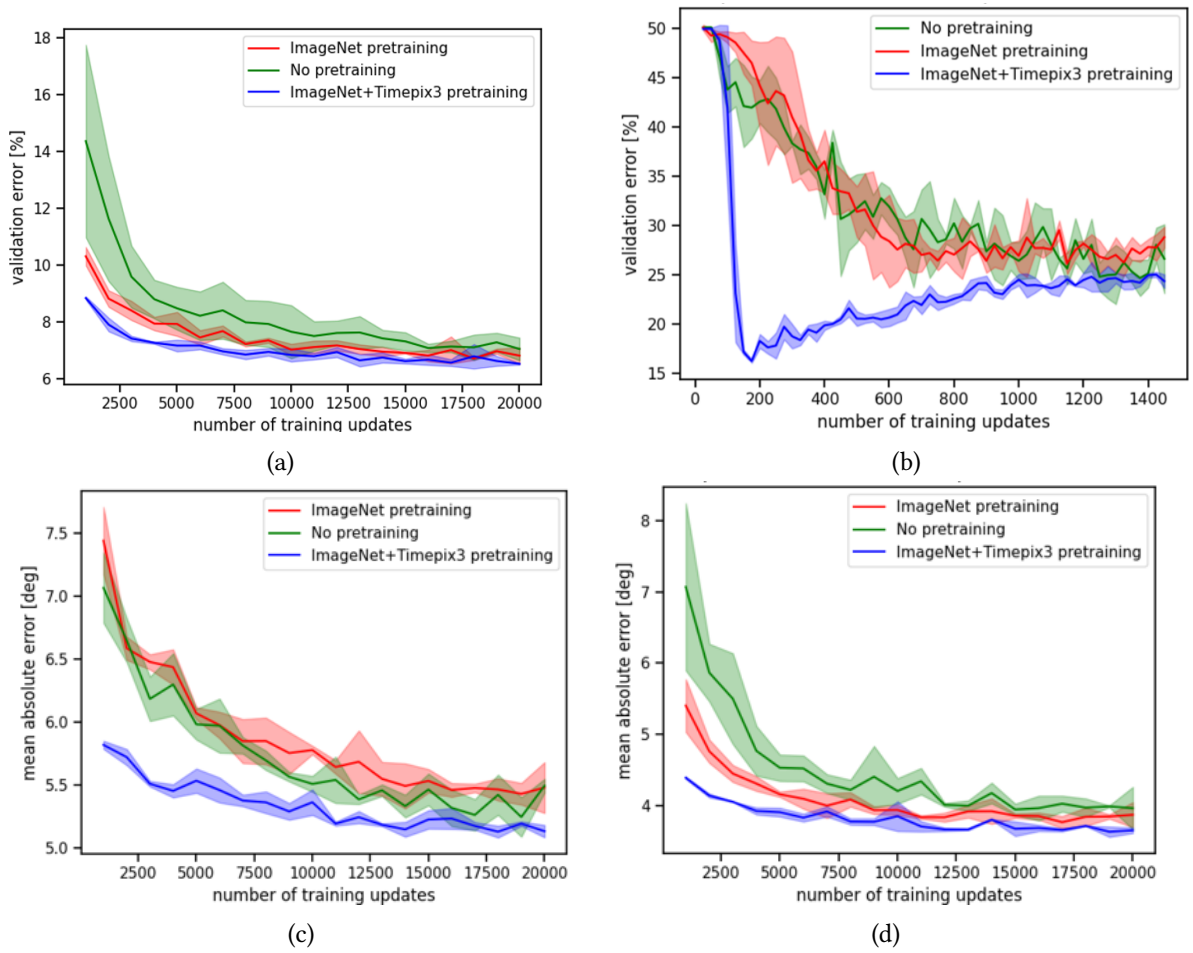


Figure 4: Error on validation set as a function of pretraining method, number of training updates, and downstream task as follows: (a) Proton-electron separation with a base dataset containing 10^5 tracks. (b) Proton-electron separation with a small dataset consisting of 50 tracks. (c) ϕ angle prediction (d) θ angle prediction. Each run was repeated $n_{rep} = 5$ times to estimate the variance.

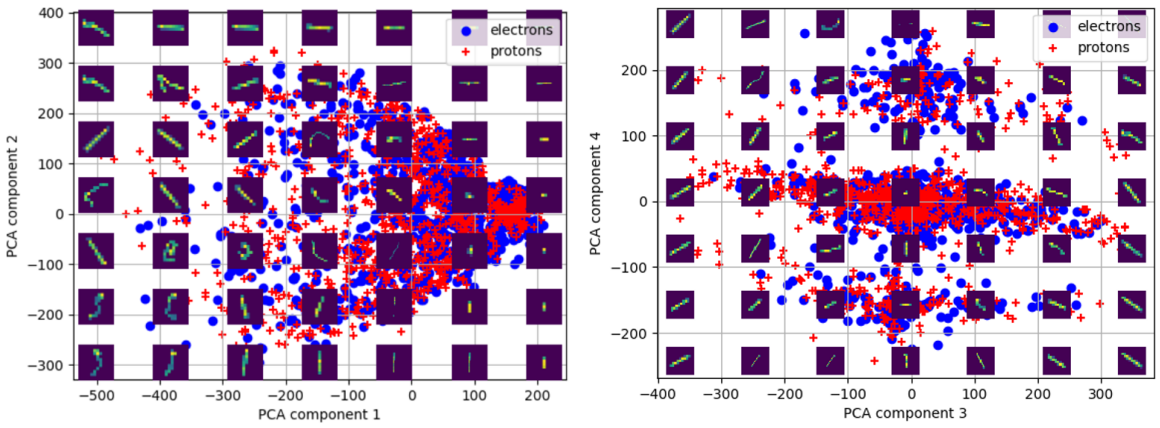
4.2.2. Importance of masking

As described in Section 3.3, at the end of the decoder, we have multiple heads: the first predicts the energy deposition (regression) and the other two predict the track bitmap of active pixels above a threshold. A comparison of the track reconstruction with and without these prediction heads is shown in Figure 3. First, the mask provides the model with additional information on the foreground-background separation, which can be used to suppress the background prediction noise seen in Figure 3c. More importantly, beyond the expected background noise suppression, we observe that binary mask prediction likely yields better gradients and more accurate representations overall. Notice the short piece of track in Figure 3a around position (160,240). This detail is not visible in Figure 3c, but it is visible in Figure 3b, showing how the masking head can impact the regression head.

Conversely, over the course of the experiments, it was not observed that input pixel masking would lead to better downstream task performance or better representations, so over the course of the experiments, the initial mask rate was reduced from 50% to 25% and later to 0%.

4.2.3. Representation quality

To assess the usefulness of the acquired latent representation, we examine how quickly it adapts to downstream tasks during fine-tuning. We compare three variants of the encoder: the first with random weights, the second with pretrained weights from the ImageNet database, and finally, a combination of



(a) (b)

Figure 5: Visualization of the latent space of the bottleneck layer of the model. (a)-(b) Principal component analysis applied over the bottleneck feature vector after pretraining, with the first four features being displayed. The first component correlates with the track length, the second with its vertical/horizontal orientation, and the third with left/right orientation. The small images displayed correspond to the representative tracks nearest to the image position within the plot. Note: Similar latent space visualization was also obtained with T-distributed Stochastic Neighbor Embedding (T-SNE).

ImageNet pretraining followed by pretraining on Timepix3 data.

A plot showing the validation error as a function of training iterations is depicted in Figure 4. We can see that our model consistently performs best and that pretraining helps it to understand the data. This difference is most pronounced when the dataset is small, where models tend to overfit quickly, as shown in Figure 4b.

Additionally, we used PCA and T-SNE to visualize the representations of proton and electron tracks. PCA representations are shown in Figure 5. One can notice that the model tends to pay the highest attention to the direction of the tracks and their length. In contrast, the particle type did not seem to be a distinguishing factor, as we can see that these tracks share a lot of morphological similarities.

4.3. Simulated data results

After the simulated fine-tuning converged, we measured the performance on downstream tasks, as described in the following subsections.

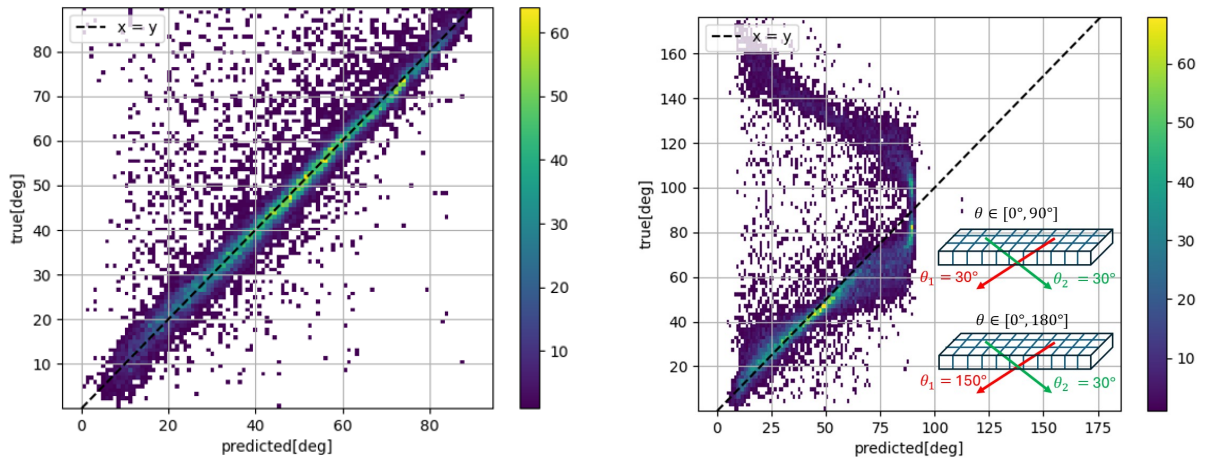
4.3.1. Proton-electron separation

For our first proton-electron separation, we used data simulated in Geant4 [28] to closely approximate the expected energy spectrum of particles at the orbit altitude where SATRAM operates. For the real spectrum [27], we achieved the accuracy of 93.4%. In order to be able to compare with related work [29] in the literature, we also recreated a simplified spectrum from the referenced paper. Compared to the previous results, we see a slight improvement in accuracy from 94.49% to 95.95%, albeit at the cost of a significant increase in model complexity.

4.3.2. Proton angle prediction

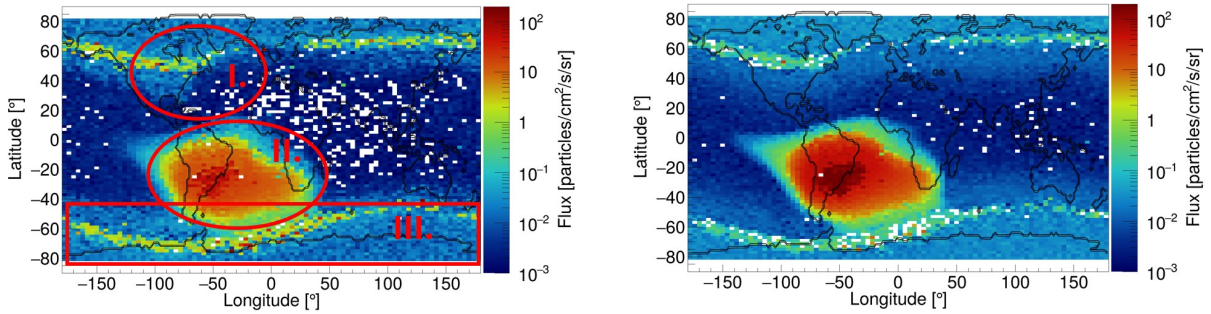
Regarding angle prediction, we only do it for certain particle types; in this case, only protons. This is because electrons undergo significant scattering in the sensor material, rendering the initial angular information hardly recoverable.

Additionally, due to the lack of timing information in the Timepix detector, we predict θ and ϕ angles modulo 180 degrees. In fact, for θ prediction, we mirror it around 90°, predicting $\theta' := \theta$ if $\theta \leq 90°$ and $\theta' := 180 - \theta$ otherwise. This is again motivated by the fact that the tracks with $\theta_1 = 90° - \alpha$ and



(a) (b)

Figure 6: Plot depicting the relation between predicted and true angles of the simulated tracks. (a) Prediction of θ angle with MAE = 3.68° and $R^2 = 0.82$. A similar plot was obtained for ϕ with MAE = 4.98° and $R^2 = 0.94$. (b) Prediction of θ angle without symmetry ($\theta \in [0°, 180°]$). The model struggles to correctly distinguish the angle because the tracks are too similar.



(a) (b)

Figure 7: (a) Proton flux estimated by the previous model [27]. (b) Proton flux estimated by the model presented in this paper. The key differences in the proton flux are illustrated in the figure: (I) our model produces a smoother proton field estimate; (II) the South Atlantic Anomaly shows a higher proton flux; and (III) there are fewer protons in the polar belts. All of these points are more consistent with the underlying physics theory.

$\theta_2 = 90° + \alpha$ for $0° \leq \alpha \leq 90°$ appear to generate very similar tracks, as also shown in the analysis of model performance in Figure 6b.

Moreover, we also remove tracks with fewer than three pixels, as they do not provide sufficient information about the angle and significantly contribute to the error of the model.

4.4. Experimental data results

Finally, in this subsection, we present the validation results for the trained and fine-tuned models.

4.4.1. Proton-electron separation

The separation accuracy for each proton and electron class is shown in Table 1. To achieve reasonable separation at 0° protons, dataset balancing was required; see Section 4.1. Even then, it remains the main source of error, with small tracks providing insufficient information to correctly distinguish the particle class.

As the second validation test, we used the experimental data from SATRAM. In Figure 7, we plot the estimated proton fluxes predicted by our model, compared to previous results. Since the field is

Table 1
Proton-electron separation accuracy across the measured experimental datasets.

	Protons 0°	Protons 30°	Protons 60°	Protons 75°	Electrons
Accuracy (%)	92.08	98.65	99.67	99.82	97.31

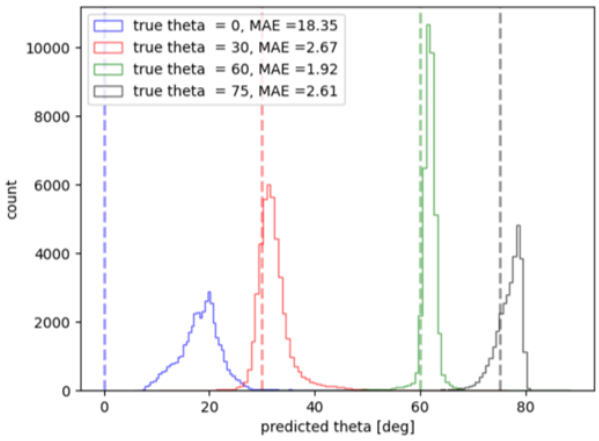


Figure 8: Distribution of the θ angle predictions vs. the expected angle. For each measured θ, the corresponding mean absolute error is displayed.

electron-dominated (in a ratio of approximately 1:10) and the predicted electron fluxes do not differ significantly, we do not display them.

4.4.2. Theta angle proton prediction

A final validation experiment was conducted to estimate the θ proton angle for known measurements. The color indicates the number of tracks in the given bin. In Figure 8, we show the true angle and the distribution of the angles predicted by the model. An important nuance for the prediction to work is that the simulated and measured sensors should be of the same thickness. The thicker the sensor, the longer the tracks appear, and the mismatch would create bias in the prediction. Overall, we see good agreement with the expected angle, except for the case with $\theta = 0°$, where we lack information to accurately determine the angle because the tracks have too few pixels.

5. Conclusion

In this paper, we present a method for pretraining deep CNNs on sparse images, primarily for detecting elementary particles. We visualize the model's latent space and find that track direction emerges as the dominant extracted feature. The proposed methods are validated across multiple downstream tasks, including the prediction of incidence angle and particle type. The models are further validated against experimental data from beamline and space experiments, showing good agreement with theoretical physics, except for a few-pixel tracks, which lack sufficient information to accuratelydetermine track properties.

Acknowledgments

The authors T.Č. and F.M. acknowledge funding from the Charles University Grant Agency (GA UK, project no. 142424). The authors B.B. and S.G. acknowledge funding from the Czech Science Foundation (GACR, reg. no. GM23-04869M).

Declaration on Generative AI

During the preparation of this work, the author(s) used Claude Sonnet 5 and Gemini 3.5 Flash in order to stylistically improve the text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] X. Llopart, R. Ballabriga, M. Campbell, L. Tlustos, W. Wong, Timepix, a 65k programmable pixel readout chip for arrival time, energy and/or photon counting measurements, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 581 (2007) 485–494. URL: https://www.sciencedirect.com/science/article/pii/S0168900207017020. doi:https://doi.org/10.1016/j.nima.2007.08.079, vCI 2007.

[2] T. Poikela, J. Plosila, T. Westerlund, M. Campbell, M. De Gaspari, X. Llopart, V. Gromov, R. Kluit, M. Van Beuzekom, F. Zappon, et al., Timepix3: a 65K channel hybrid pixel readout chip with simultaneous ToA/ToT and sparse readout, Journal of Instrumentation 9 (2014) C05013–C05013. URL: https://iopscience.iop.org/article/10.1088/1748-0221/9/05/C05013. doi:10.1088/1748-0221/9/05/C05013.

[3] C. Granja, J. Jakubek, S. Polansky, V. Zach, P. Krist, D. Chvatil, J. Stursa, M. Sommer, O. Ploc, S. Kodaira, M. Martisikova, Resolving power of pixel detector Timepix for wide-range electron, proton and ion detection, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 908 (2018) 60–71. URL: https://www.sciencedirect.com/science/article/pii/S0168900218309586. doi:https://doi.org/10.1016/j.nima.2018.08.014.

[4] S. Gohl, B. Bergmann, C. Granja, A. Owens, M. Pichotka, S. Polansky, S. Pospisil, Measurement of particle directions in low earth orbit with a timepix, Journal of Instrumentation 11 (2016) C11023. URL: https://doi.org/10.1088/1748-0221/11/11/C11023. doi:10.1088/1748-0221/11/11/C11023.

[5] C. Granja, S. Polansky, Z. Vykydal, S. Pospisil, A. Owens, Z. Kozacek, K. Mellab, M. Simcak, The SATRAM Timepix spacecraft payload in open space on board the Proba-V satellite for wide range radiation monitoring in LEO orbit, Planetary and Space Science 125 (2016) 114–129. doi:10.1016/j.pss.2016.03.009.

[6] B. Bergmann, S. Gohl, D. Garvey, J. Jelínek, P. Smolyanskiy, Results and perspectives of timepix detectors in space—from radiation monitoring in low earth orbit to astroparticle physics, Instruments 8 (2024). URL: https://www.mdpi.com/2410-390X/8/1/17. doi:10.3390/instruments8010017.

[7] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (1998) 2278–2324. doi:10.1109/5.726791.

[8] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: F. Pereira, C. Burges, L. Bottou, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, volume 25, Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (2015) 436–444. URL: https://doi.org/10.1038/nature14539. doi:10.1038/nature14539.

[10] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.

[11] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations (ICLR), 2021. arXiv:2010.11929.

[12] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, H. Jegou, Training data-efficient image transformers & distillation through attention, in: M. Meila, T. Zhang (Eds.), Proceedings of the 38th

International Conference on Machine Learning, volume 139 of *Proceedings of Machine Learning Research*, PMLR, 2021, pp. 10347–10357. URL: https://proceedings.mlr.press/v139/touvron21a.html.

[13] K. Sykorova, Z. Buk, M. Cepek, C. Granja, C. Oancea, L. Marek, Machine learning models for single-particle classification with Timepix3 detector, Journal of Instrumentation 20 (2025) C01028. doi:10.1088/1748-0221/20/01/C01028.

[14] C. Oancea, L. Marek, M. Vuolo, J. Jakubek, E. Soharova, J. Ingerle, D. Turecek, M. Andrlik, V. Vondracek, T. Baca, M. Sabia, R. Kaderabek, J. Gajewski, A. Rucinski, S. Stasica, C. Granja, Trax engine: Advanced processing of radiation data acquired by Timepix detectors in space, medical, educational and imaging applications, Journal of Instrumentation 20 (2025) C07065. doi:10.1088/1748-0221/20/07/C07065, arXiv:2410.10242.

[15] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[16] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving Language Understanding by Generative Pre-Training, Technical Report, OpenAI, 2018.

[17] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: H. D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, volume 119 of *Proceedings of Machine Learning Research*, PMLR, 2020, pp. 1597–1607. URL: https://proceedings.mlr.press/v119/chen20j.html.

[18] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 9726–9735. doi:10.1109/CVPR42600.2020.00975.

[19] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, R. Girshick, Masked autoencoders are scalable vision learners, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, pp. 15979–15988. doi:10.1109/CVPR52688.2022.01553.

[20] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (2006) 504–507. doi:10.1126/science.1127647.

[21] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features with denoising autoencoders, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, Association for Computing Machinery, New York, NY, USA, 2008, p. 1096–1103. URL: https://doi.org/10.1145/1390156.1390294. doi:10.1145/1390156.1390294.

[22] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[23] T. Finke, M. Krämer, A. Morandini, A. Mück, I. Oleksiyuk, Autoencoders for unsupervised anomaly detection in high energy physics, Journal of High Energy Physics 2021 (2021) 161. URL: https://doi.org/10.1007/JHEP06(2021)161. doi:10.1007/JHEP06(2021)161.

[24] M. Tan, Q. Le, EfficientNet: Rethinking model scaling for convolutional neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, volume 97 of *Proceedings of Machine Learning Research*, PMLR, 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015, IEEE Computer Society, 2015, pp. 1–9. URL: https://doi.org/10.1109/CVPR.2015.7298594. doi:10.1109/CVPR.2015.7298594.

[26] ATLAS Collaboration, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 (2008) S08003. doi:10.1088/1748-0221/3/08/S08003.

[27] S. Gohl, B. Bergmann, H. Evans, P. Nieminen, A. Owens, S. Posipsil, Study of the radiation fields in

leo with the space application of Timepix radiation monitor (SATRAM), Advances in Space Research 63 (2019) 1646–1660. URL: https://www.sciencedirect.com/science/article/pii/S0273117718308706. doi:https://doi.org/10.1016/j.asr.2018.11.016.

[28] S. Agostinelli, et al. (GEANT4), GEANT4: A Simulation toolkit, Nucl. Instrum. Meth. A 506 (2003) 250–303. doi:10.1016/S0168-9002(03)01368-8.

[29] M. Ruffenach, S. Bourdarie, B. Bergmann, S. Gohl, J. Mekki, J. Vaillé, A new technique based on convolutional neural networks to measure the energy of protons and electrons with a single Timepix detector, IEEE Transactions on Nuclear Science 68 (2021) 1746–1753. doi:10.1109/TNS.2021.3071583.

[30] P. Isola, J.-Y. Zhu, T. Zhou, A. A. Efros, Image-to-image translation with conditional adversarial networks, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5967–5976. doi:10.1109/CVPR.2017.632.