Learning 3D Structure in Irradiated Lithium Fluoride via Masked Autoencoders

Carolyn Hellerqvist Smith
Stanford University, Department of Physics
carsmith@stanford.edu

Piper Fleming
Stanford University, Department of Computer Science
piperf@stanford.edu

Abstract

Minerals have recently emerged as promising detection media for rare event searches in experimental particle physics. Nuclear recoils from particle interactions can leave micrometer-scale defect tracks in crystalline materials. However, identifying these interaction event signatures requires imaging large volumes of mineral at high resolution and accurately segmenting signal. Current segmentation techniques rely on expert annotation or transfer learning. These approaches are either unscalable or poorly matched to this domain.

To address this, we propose a self-supervised learning framework for pretraining a transformer encoder on irradiated Lithium Fluoride (LiF) samples imaged with light sheet fluorescence microscopy (LSFM) from the PALEOCCENE collaboration [1]. Using a masked autoencoding strategy, our goal is to learn robust representations that support future finetuning for semantic segmentation or object detection. Initial results show that while the model effectively reconstructs periodic background patterns, it struggles to recover true signal at masking ratios above 20%, likely due to the signal's spatial sparsity and strong local structure. These findings suggest that more targeted masking or patch selection strategies may be necessary for the model to capture features relevant to rare event signatures.

1. Introduction

This project addresses the challenge of learning meaningful representations of nuclear recoil-induced defects in LiF crystals imaged using light-sheet fluorescence microscopy (LSFM). Nuclear recoils occur when energetic particles collide with atoms in a solid, displacing them from their lattice positions. These vacancies in the lattice structure can be visualized using LSFM by selectively illuminating and imaging thin optical slices of the sample. Accurate identification of these defects is essential for solid-state detectors used in rare event searches (physics experiments that aim to detect very weakly interacting particles like neutrinos and hypothesized dark matter).

Rare event search experiments require scanning a lot of material because the expected signals are very sparse. The current state of research and development for minerals like LiF as a detection medium is very small-scale. Scans are collected and human scientists inspect and label defects. A previous attempt [1] to use machine learning for defect segmentation relies on a pretrained model trained using biological microscopy data. This model makes mistakes often enough that scientists still inspect and correct its output. Having a human expert hand-label data, as is currently done, is not a scalable approach and thus creates the need for a domain-specific segmentation model.

Given raw 3D fluorescence microscopy data of irradiated LiF crystals, our goal is to learn rich feature representations of localized defect structures using self-supervised learning. These learned representations can later support downstream tasks such as identifying which regions of the image contain nuclear recoil-induced tracks and distinguishing between tracks caused by different types of particles, such as alpha particles or neutrons.

Our raw data consists of 3D image volumes, which are stacks of 2D optical slices that form a volumetric representation of the crystal. For the purposes of visualization we present summed 2D projections of these volumes throughout the paper, but our entire pipeline operates on 3D data.

We begin by denoising the raw 3D data and extracting crops, algorithmically ensuring that each crop contains sufficient signal. We then tokenize each crop using a lightweight convolutional layer and incorporate spatial context through a learnable positional encoding network. These embedded tokens are then passed through a transformer-based encoder and decoder trained with masked autoencod-

ing. Finally, a reconstruction head projects the decoded tokens back into 3D image space, where we compute a loss with respect to the ground truth image. This pipeline allows the model to learn spatial correlations between tokens in order to perform accurate reconstruction.

2. Related Work

2.1. Physics Research

To motivate our project, we reference a whitepaper [2] that describes the concepts and early research and development efforts into using minerals as detectors in rare event searches. This approach is particularly promising because naturally occurring crystals can persist for billions of years. This increases the likelihood that they have recorded rare interactions, such as those caused by neutrinos and hypothesized dark matter particles, in their defect structures. Because the signals of interest are so rare, rare event detection would require scanning a huge amount of mineral volumes. Baum et al. [2] strongly articulate the need for automated methods to identify signals or regions of interest in this data.

We used this perspective to inform our analysis and understanding of Araujo et al. [1], of the PALEOCCENE Collaboration, who collect the microscopy data we use in this project and implement the code we use to interface with it. They also introduce a preliminary analysis framework to segment regions of signal.

While the use of LSFM to scan large crystal volumes is not novel, this paper pioneered its use as a candidate method for the high-throughput, variable resolution scanning that a mineral-based rare event search would require. A major strength of this paper is the successful demonstration of this imaging method.

However, a weakness is that the code supporting their preliminary signal segmentation method is not sufficiently robust or accurate for compatibility with production-scale scanning. They rely on an out-of-the-box, "no code solution" segmentation tool called Ilastik, which was pretrained using biological microscopy data. This tool accepts a small number of 2D hand-labeled slices and presumably uses transfer learning to finetune itself. The results are inaccurate enough that experts still review and correct its output before the segmented images are passed through the rest of the analysis pipeline. In other words, accurate signal segmentation is a largely unsolved problem in this domain.

2.2. Model Architecture Research

Given our lack of labeled data, we believe the best approach to creating a robust signal segmentation model is to separately pretrain an encoder and then finetune a segmentation head. This would allow us to pretrain the encoder using a self-supervised learning method, hopefully allowing it to develop a useful representations of our data without requiring extensive labels.

Two candidate architectures we consider are explored by Young et al. [17], who present a strong example of learning structural information about particle tracks with a masked autoencoder (MAE), and by Dominé et al. [7], who achieve the same task with a supervised CNN. The data used by both of these references comes from simulations of a different type of particle detector and consists of sparse pointclouds. While this is inherently different from our dense, noisy, and voxelized 3D images, they search for signals with the same geometry as us: line-like structures which extend for some characteristic length.

Young et al. [17] introduced a transformer-based MAE which approached the performance of supervised methods at semantic segmentation of this pointcloud data.

The algorithm in this work involves first tokenizing the particle trajectories, chiefly through a method known as volumetric point cloud grouping. The rationale for this method lies in its ability to bypass many of the traditional problems associated with KNN methods. These points are then grouped into patches, normalized, encoded into a single latent vector, and then run through an encoder with positional embeddings. However, this model struggles to separate uncorrelated events within small spatial regions.

The strength of Dominé et al. [7] lies in solving the problem of using sparse data in networks traditionally designed for dense visual data. This paper shows that a CNN could learn features of our sparsely distributed signal. While this paper relies on supervised learning, it was intriguing to see them use GPU and wall time as two evaluation metrics, particularly considering that those could apply in our model.

Lastly, considering that most state-of-the-art approaches in computer vision make use of transformers, we reference Li et al. [10] for an understanding of the current transformer landscape and a sense of how transformers compare to other methods like CNNs. A crucial insight from this paper was that transformer approaches are both simpler and more accurate, particularly when given enough data.

Other papers used to inform the approach from an architecture perspective included Marks et al. on alternative self-supervision methods, [11], Shah et al. on U-nets for electron microscopy images [16], Dionelis et al. for alternate error methods [6], Chitta et al. on exploring the idea of proxy labels, [5], and Mehta et al. [12] as a way to further narrow down our choice of segmentation method. We also referenced the Caron et al., or the DINO paper[4] to inform our self-attention maps.

3. Dataset and Features

3.1. Data Preprocessing

Preprocessing aims to retain enough signal for the model to learn meaningful structure. To achieve this, we improve the signal-to-noise ratio.

Each scan of the LiF is slightly different in terms of lighting, shape, and z-resolution. We standardize our data by normalizing intensity values within each scan and downsampling all scans to match the lowest resolution. It is notable that fluorescence intensity in LSFM is measured in photon counts.

The approach used by the PALEOCCENE Collaboration [1] involves selecting the crystal regions in each scan which were illuminated by the imaging tool and applying a gaussian blur. To replicate this, the off the shelf packages we used include Torch [13], NumPy [8], Pandas [15], and Scikit-Learn [14].

Our data consists of per-voxel intensity values, and regions of high intensity correspond to regions of high signal. Semantically, we look for regions of high intensity which have a geometry characteristic of particle tracks (elongated line-like structures).

The data consists of 3D volumetric scans of LiF samples. As a first preprocessing step, we removed regions of the raw scans that were not illuminated during imaging. Because of the light-sheet fluorescence microscopy setup, this cut was applied strictly along the y-axis. The dimensions of the non-luminous scan regions were recorded in a spreadsheet by the original data collectors [1], which we referenced in order to accurately crop the scan.

While the resolution of our data varies slightly across scans, they all have an approximate resolution of 3000 voxels (0.425 micrometers per voxel) along the X and Y axes. The resolution in Z is much coarser, around 30 voxels (7 micrometers per voxel). The result is that our 3D volumetric image has anisotropic voxel dimensions. The primary variation across scans in our dataset was the Z-axis resolution, which ranged from 5 to 10 micrometers. To standardize the Z-dimension, we downsampled the higher-resolution scans accordingly.

Next, 15 micrometers (around 60 voxels) from each edge of the scan along the X and Z axes was removed due to physical limitations of the microscopy beam. To exclude regions of the largest light sheet spread, data sections closer to the center are utilized. This is consistent with how the original authors in [1] prepared the scans for segmentation.

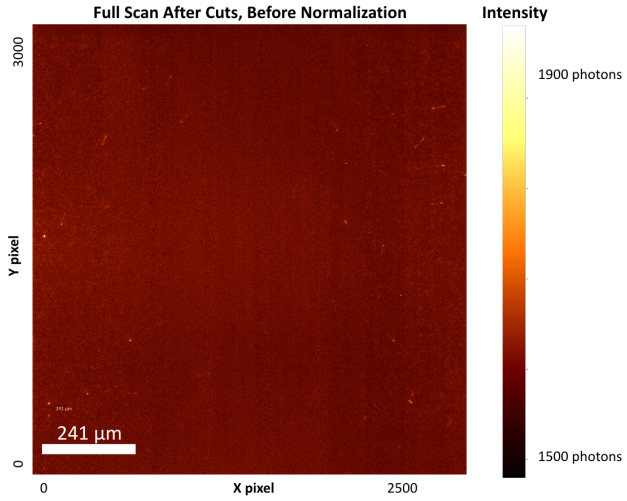


Figure 1: One of the full scans after cuts but before intensity normalization.

Once we apply these cuts to our data, we apply Gaussian blurring with a voxel-wise isotropic standard deviation of 2. We select the blur standard deviation through qualitative inspection of the results.

To increase the effective size of our dataset, we partition each volumetric scan into smaller sub-volumes, treating each as an independent data point. This is justified by the assumption that particle tracks are spatially uncorrelated, given the stochastic nature of the nuclear recoil events that produce them. While they are certainly temporally correlated, the resulting defect patterns are spatially unpredictable, as decay and scattering processes can occur at arbitrary locations within the crystal. Furthermore, for the purpose of a rare event search, we only care about whether we have recorded an event, not when it happened.

Our initial scans had total dimensions on the order of thousands of micrometers in X and Y, and hundreds of micrometers in Z. Defect tracks tend to be on the order of $<$50 micrometers long. We choose a kernel size of 5 x 250 x 250 voxels when extracting smaller data points from our full scans. This gives us around 5000 samples. While this step augments the size of our dataset, we apply no augmentations to the data itself (flips, rotations, etc.).

Then, the samples are normalized globally to ensure that discrepancies in lighting across different scans do not affect our model's ability to learn. This is particularly important because our model looks at intensity differences, and skipping this step would lead to different intensity scalings per data point.

Our nontrivial next step was selecting which 5 x 250 x 250 voxel samples to include in our final dataset. Because our data is very noisy, we want to avoid including samples that are exclusively noise. We utilize a method implemented by the PALEOCCENE group [1] which uses Scipy library

methods to identify boundaries and label the size of candidate signal regions. A major limitation of this method is that it is not compatible with our 3D data, meaning we must instead apply candidate signal region finding on summed 2D projections. By inspection, we observe that this method fails often. We choose to only include samples that have at least 2 candidate signal regions, ending up with a training set of 2,125 samples, a validation set of 250 samples, and a test set of 125.

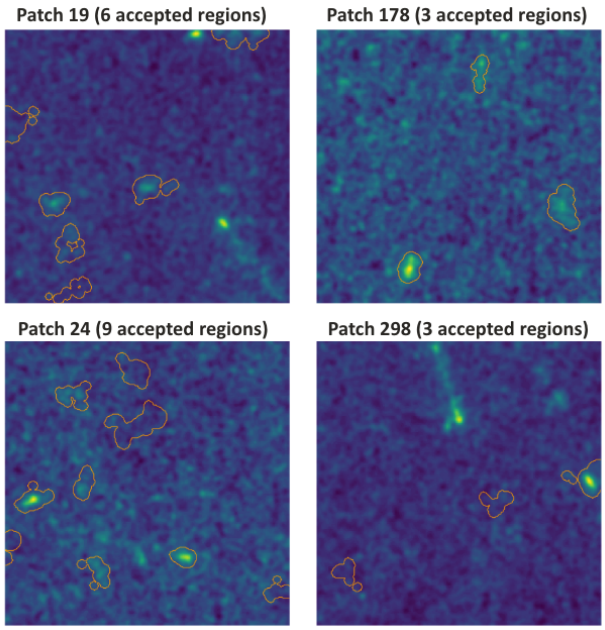


Figure 2: Examples of performance by the out of the box segmentation method. We see examples of where the segmentation method clearly got bright tracks, but also where it missed tracks visible to the human eye.

3.2. Model Architecture and Hyperparameters

Motivated by the lack of extensive labeled data in this domain, our goal is to pretrain an encoder that learns meaningful representations of our data in a self-supervised way. We do this in anticipation of future finetuning for semantic segmentation or object detection with very minimal labeled datasets. We choose to use a masked autoencoder (MAE) strategy to pretrain a transformer encoder.

For our input, one data point consists of a volumetric image of 5x250x250 voxels. We tokenize this input into non-overlapping 5x10x10 voxel patches.

We begin by applying a 3D convolutional layer to each input token to produce a 384-dimensional feature embedding that captures local intensity patterns. In parallel, we generate learnable positional embeddings using a lightweight two-layer multilayer perceptron (MLP). These embeddings are also of dimension 384. Incorporating positional information is essential for enabling the model to learn about spatial correlations.

Positional embeddings and patch embeddings are element-wise added and used as the inputs to our transformer encoder.

The transformer encoder includes six self-attention layers with four attention heads, operating on 384-dimensional embedding vectors. Multi-headed attention was chosen to help improve our model's ability to learn complex features. We also include a linear layer after each multi-headed attention layer to combine attention outputs from each head into a single latent vector per token.

After encoding, we introduce a learnable mask token for each token that was originally masked. These learnable tokens are introduced after the encoder stage due to a critical insight from the original MAE paper [9]. They demonstrate that early inclusion of learnable mask tokens disrupts the disrupts the learning of informative latent representations. Deferring mask token introduction until decoding ensures that the encoder - the backbone we anticipate using for downstream tasks - processes only real, uncorrupted image data, therefore preserving the quality of learned features.

Then, we use the same positional encoding model on the 384-dimensional learnable tokens and the latent vectors. Both the learnable tokens and the latent vectors, now containing positional embeddings, are fed into the decoder. Our transformer decoder has two self-attention layers and four heads with an output dimension of 1024. Since we implement a pretraining strategy, we use a lightweight decoder compared to our encoder in order to ensure the encoder is forced to learn the best possible representations.

Lastly, we send the decoder outputs through a reconstruction head in order to transform them back into the 3D image space (a 5x10x10 voxel token), where a MSE loss between ground truth and reconstructed masked data is calculated.

We selected the learning rate, token size, and masking ratio hyperparameters using a cross-validation strategy. We trained our model architecture for 200 epochs, and selected the learning rate and token size which yielded the lowest test loss. Our final learning rate scheme was a cosine annealing decay schedule with five epochs of linear warm-up, peaking at 1e-4.

We selected the masking ratio by qualitatively inspecting the reconstructions of our data and choosing a ratio that gives rise to a non-trivial learning task yet does not mask out entire neighborhoods of our patches.

The spatial correlations of signal in our data are hyperlocal, where different geometric regions of the same particle track are correlated, but regions corresponding to different particle tracks are not. This suggests that a conventional masking ratio of 60% is too strong. We select a masking ratio of 20% as our final value after testing 20%, 40%, and

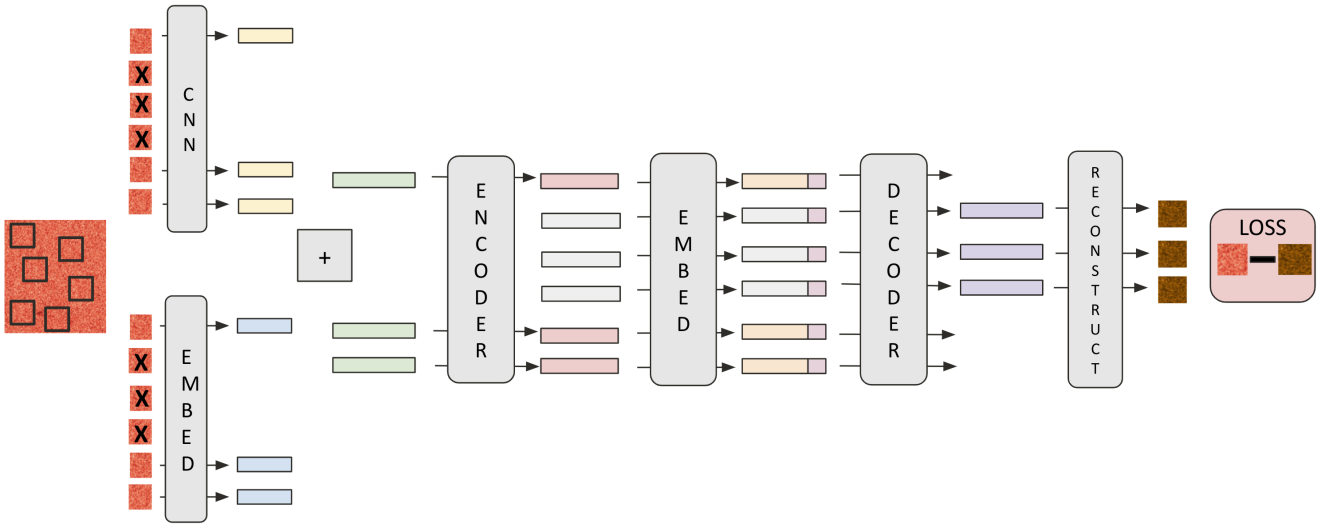


Figure 3: Our model architecture.

60%.

To evaluate our choices for embedding dimension, we attempted to overfit our model on a set of 256 images with a masking ratio of 0%. By testing its capacity as a simple autoencoder, we assessed whether we were introducing unwanted bottlenecks. We found that an embedding dimension of 128 did not give our model the capacity to reconstruct even unmasked inputs. Our final dimension choice of 384 does provide our model with the capacity to represent our data. This experiment is discussed more in Section 5.

Final training was conducted using the Adam optimizer for 400 epochs.

4. Evaluation

We are interested in self-supervised learning methods for this project due to the highly specialized and label-scarce nature of our data. In future physics experiments searching for rare signals, large volumes of data must be collected. Hand-labeling by experts is not a scalable signal identification strategy. Manual labeling is time-consuming, error-prone, and inefficient, particularly for our application where the data is inherently noisy and signal appearances are inconsistent. Even domain experts may struggle to provide accurate annotations, motivating the need for approaches that can learn meaningful representations without reliance on extensive labeled datasets.

To achieve a final goal of semantic segmentation, it is necessary to pretrain a backbone with self-supervised learning before finetuning on a very minimal labeled dataset. In this project, we explore pretraining the backbone to learn the general structure of our data.

We choose masked autoencoding as our self-supervised learning framework. This is motivated by the structure of our data: while the background signal exhibits spatially periodic and relatively smooth behavior, true signal features, like particle tracks, manifest as sharply peaked, localized structures. This spatial disparity suggests that a model trained to reconstruct masked regions of the data can learn to attend to the high-resolution, hyperlocal correlations indicative of true signal while modeling broader background regularities.

The pretext task used in masked autoencoding is 3D reconstruction of our original 3D image.

Mathematically, the masked autoencoding algorithm works as follows:

Let $x \in \mathbb{R}^{Z,Y,X}$ be a 3D volumetric data sample. We tokenize it into non-overlapping patches, yielding a set of N tokens $\{x_i\}_{i=1}^{N}$. Each token corresponds to a local 3D region.

We randomly mask a subset $\mathcal{M} \subset \{1, \ldots, N\}$ of these tokens. Let $\tilde{x}$ be a corrupted version of x (e.g., with masked tokens replaced by learnable embeddings).

Let f_θ denote our masked autoencoder model, parameterized by θ.

We learn $f_\theta : \mathbb{R}^{Z,Y,X} \to \mathbb{R}^{Z,Y,X}$, where we decode each token back into 3D image space after embedding it into a latent space.

The reconstruction loss is defined over the masked tokens only. Given that y_i denotes the ground truth token at location $i \subset \mathcal{M}$, and $\hat{y}_i = f_\theta(\tilde{x})_i$ is the reconstructed token, we optimize the following mean-squared error:

$$\min_\theta \ \mathbb{E}_{x\sim\mathcal{D}} \left[\frac{1}{|\mathcal{M}|} \sum_{i\in\mathcal{M}} ||\hat{y}_i - y_i||_2^2 \right] \tag{1}$$

Notably, we did not choose to use contrastive learning, despite it being a common self-supervised learning strategy.

Contrastive learning involves creating two views of the same data sample, and minimizing the contrastive loss over positive pairs:

$$\mathcal{L}_i = -\log \frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(z_i, z_k)/\tau)} \quad (2)$$

where $\text{sim}(u, v) = \frac{u^\top v}{\|u\|\|v\|}$ (cosine similarity), and τ is a temperature hyperparameter. The contrasting learning objective is to cluster semantically similar data points in the embedding space, essentially collapsing all variations of a concept into one representation. This is why class embeddings in contrastive learning models often appear as blurry averages of many views [3]. However, this is explicitly not what we want. Because our signals are highly localized, we want to teach our model to remain sensitive to localized features in our data.

This is why we chose to do masked autoencoding, since masking part of the data and asking the model to reconstruct it would ensure that our calculated loss was not misaligned with our objective of being sensitive to highly localized spatial correlations.

A main qualitative metric we used to evaluate our model were self attention maps, where the outputs of the attention layer were visually graphed. For each attention layer in the model, each head computes how relevant each patch is to every other patch, and these values are then fed through a softmax function. What we get out is a 625 x 625 matrix per head representing how highly correlated each pixel was with every other pixel. We use this method in order to visually identify where the model is looking for information, and how it values certain regions of the data.

5. Experiments/Results/Discussion

Initially, we trained our model for 150 epochs with an embedding dimension of 128, a token size of 5x25x25 voxels, and a masking ratio of 60%. However, with a masking ratio this high, we observed that our model learned to reconstruct the mean intensity of each token, rather than distinctive features within the token.

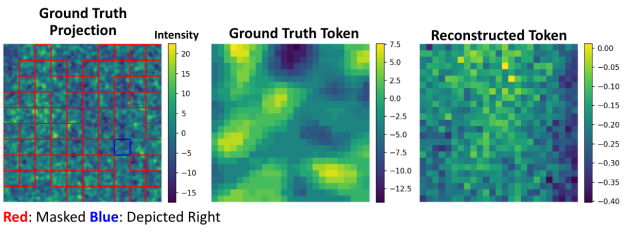
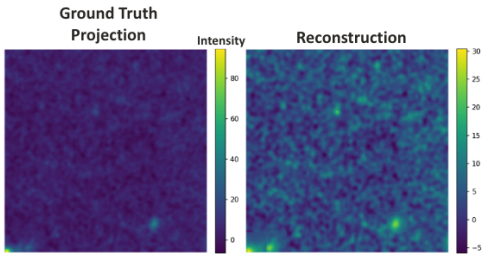


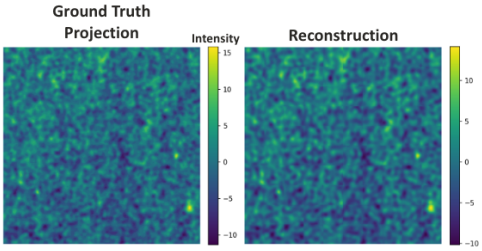
Figure 4: Initial model converges to token mean. Note intensity color bar scales on ground truth vs. reconstructed token.

This initial result led us to revisit our hyperparameter choices. To test model capacity, we trained it as a simple autoencoder (no masking) on a small set of 256 images. With an embedding dimension of 128, the model failed to reconstruct the data. However, after increasing to 384, the model was able to accurately reconstruct the overfitting dataset.

We trained the simple autoencoder with an embedding dimension of 384 for 150 epochs using our existing optimizer and learning rate schedule. This was sufficient to reconstruct the periodic background noise. In regions with sharp signal peaks, the model also produced peaked intensities, though training was not long enough to overfit these highly localized features.



(a) Periodic background with relatively narrow intensity range is reconstructed well.



(b) We did not completely overfit to regions of intense signal.

Figure 5: Results of overfitting a small autoencoder to a small training set.

Once we established that our model had sufficient capacity, we began to reintroduce masking. An additional change we introduced at this stage was a smaller token size of 5x10x10 voxels, aligned better with the characteristic spatial period of our background. This caused us to reduce our batch size from 256 to 64 due to GPU memory constraints.

We trained our model for 200 epochs with a masking ratio of 60% as used by [17].

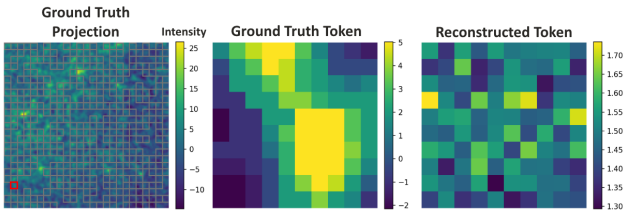


Figure 6: Validation token reconstruction after training with 60% masking ratio for 200 epochs.

However, we realized that a masking ratio of 60% was too high for our data, since the spatial correlations of signals are highly localized. Masking whole signal regions does not enable the model to learn about their structure, because there is simply not sufficient relevant information contained in other areas of the sample.

As a result of this, we dropped our masking ratio to 20% and retrained our model for 200 epochs. Figure 7 depicts a reasonable reconstruction of a masked token from the validation dataset by this model. The model was able to accurately predict both the relative intensity ranges and their approximate positions in the path.

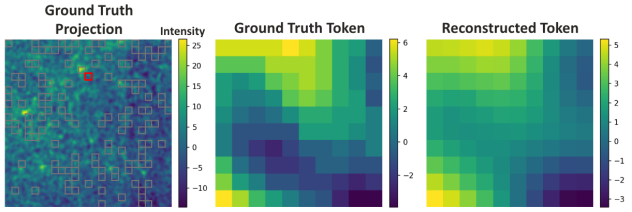
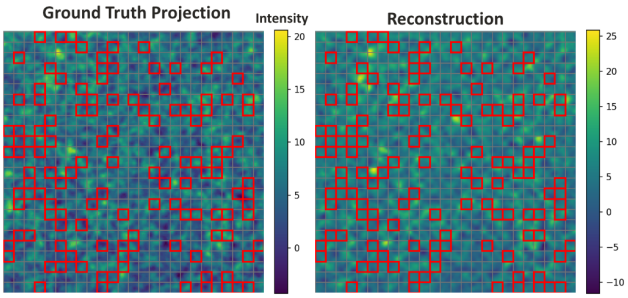


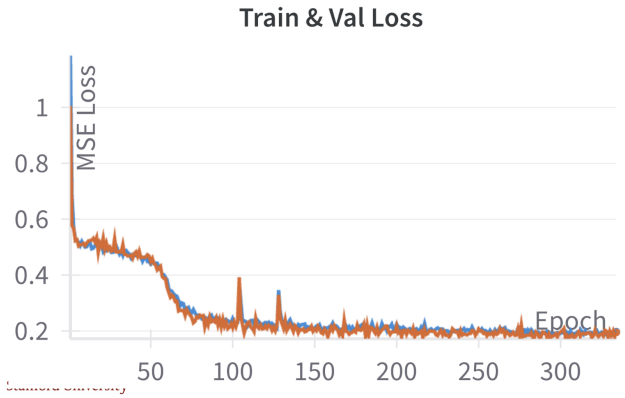
Figure 7: Token reconstruction on a test sample after training for 200 epochs with masking ratio of 20%.

From plotting training and validation loss curves, we find that that our model is not overfitting since we observe no differing asymptotic behavior.

Given that we observe no overfitting, we continue training our model to 330 epochs in order to extract better performance.



(a) When visualized for a whole image, we see the model continues to do well in predicting local structure.



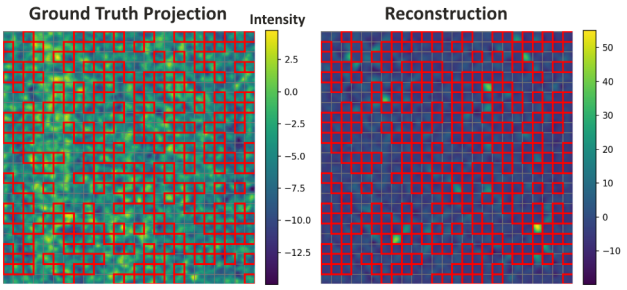
(b) We do not observe overfitting in our train and validation loss curves.

Figure 8: Results from training for 330 epochs with a masking ratio of 20%.

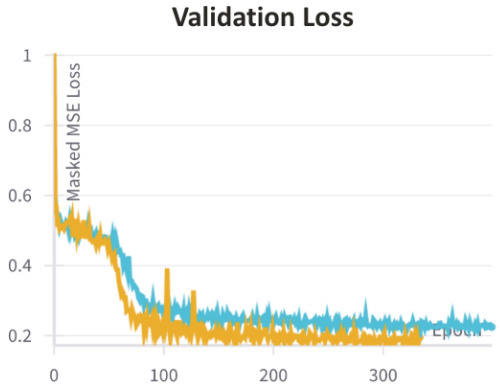
This is our best-performing model. It is trained for 330 epochs using a masking ratio of 20%, cosine annealing learning rate scheduler with five epochs of linear warm up peaking at 1e-4, and embedding dimension of 384.

This model has reasonable performance at patch reconstruction, especially in regions consisting of background. However, considering that our masking ratio is only 20%, we suspect that our model has just learned to copy and interpolate the existing data instead of actually learning complex features / spatial correlations. To test this hypothesis, we trained from scratch for 400 epochs with a stratified sampling strategy and a masking ratio of 40%. We then visualized self-attention maps from our transformer encoder.

We implement stratified sampling by masking 10 patches out of non-overlapping 5x5 voxel (25 square-voxel) regions. We choose this approach in order to avoid masking entire local neighborhoods when we increase the masking ratio. However, we observe that this model performs significantly worse at the reconstruction task than our model with 20% random masking.



(a) Ground truth vs. reconstructed test patch projection.



(b) Training with random 20% masking (yellow) vs. stratified 40% masking (blue). The blue curve is taking longer to converge.

Figure 9: Results from training for 400 epochs with stratified sampling and masking ratio of 40%.

Then, we consulted self attention visualizations in order to check whether or not there were actual regions to which our model was attending. These visualizations are representations of the importance between pairs of tokens.

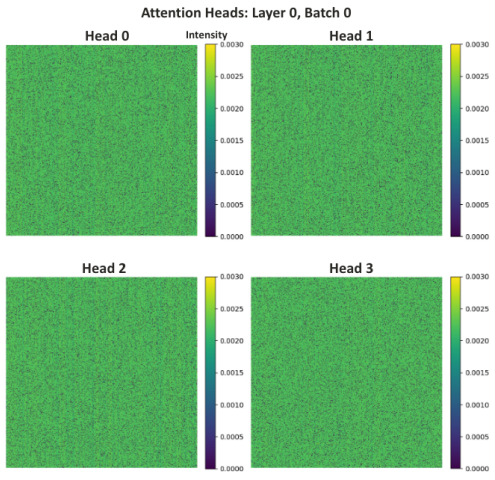


Figure 10: The visualization of self attention output from the first head, for the first transformer layer.

As we see in this figure, it turns out that our model is not learning the complex features, since we only see noise and not specific emphasis on certain regions. As such, while we conclude that our model is accurately able to represent and predict unseen data, we know that these abilities come from interpolation and not specifically learning the features themselves.

6. Conclusion/Future Work

Using masked autoencoding as a pretraining strategy, we attempt to train a transformer encoder to learn structure in 3D volumetric fluorescence scans. Training an encoder in a self-supervised way is a key step towards enabling downstream tasks like semantic segmentation in the domain of light-sheet fluorescence microscopy of crystals, which lacks extensive labeled data.

We find that our masked autoencoder is able to learn the structure of periodic noise within our data reasonably well, but struggles to accurately reconstruct bright signals. When we dropped our masking ratio to 0%, the resulting simple autoencoder was the most high-performing. Increasing the masking ratio led to overall worse performance, because the model can no longer just learn to 'copy' input data. While this is expected, making the learning task more challenging is essential to learning representations that are actually useful for downstream tasks.

The weakest component of our current pipeline is the patch selection strategy. The candidate region-of-interest (ROI) finder we use, which is based on basic SciPy heuristics, is highly inaccurate, resulting in the inclusion of input samples that are predominantly background noise. Even with a hypothetically perfect ROI detector, our current method lacks a mechanism for systematically sampling tokens along the full spatial extent of a signal. As a result, signal patches contain not just the track itself but also a large amount of surrounding noise. This imbalance causes the model to see orders of magnitude more data from the structured, periodic background than from the rare, localized signal events we actually aim to model. Consequently, the learned representations are biased toward the background, limiting the model's ability to effectively capture the structure of nuclear recoil-induced defects.

There are two follow-up steps we would suggest for this project. First, we would like to develop a more nuanced tokenization selection strategy that at least partially addresses the issue discussed above. Second, there are existing Self-Distillation with No Labels (DINO) models for feature representation of 3D medical microscopy data. It would be interesting to put our data through these models, visualize the self-attention maps, and compare them to those from our MAE.

7. Contributions and Acknowledgments

We thank the PALEOCCENE Collaboration [1] for sharing their LSFM data interfacing code and raw data.

All of the model-associated code contained in this project is our own, with the exception of the transformer layers, which were taken from the layers we implemented in class. We referenced the Github associated with the Young et al. paper [17]: https://github.com/DeepLearnPhysics/PoLAr-MAE. All of the work for this project was done via pair programming and pair discussion, and due to the sequential nature of our model, we each contributed to all models. This project is based on an extension of Carolyn's research, and she brought existing knowledge, code, and experience to this project. However, an approximate breakdown would be as follows:

Model Research and Background: Since this is Carolyn's research, she brought a lot of innate knowledge and ideas about what to do, and spearheaded much of the design. Piper did most of the research and literature review to back up many of the design decisions.

Data preprocessing: An even split. Piper worked more on the processing pipeline due to her experience with data, and Carolyn worked more on implementing the physics-specific decisions and functions.

Model creation: While both group members were present and contributing to the model design and creation, Carolyn took the lead on coding, again due to her prior familiarity with the material.

Paper/poster creation: Conversely, due to her infamiliarity with the material, Piper took the lead on writing and formatting the paper.

8. References/Bibliography

References

[1] Gabriela A. Araujo, Laura Baudis, Nathaniel Bowden, Jordan Chapman, Anna Erickson, Mariano Guerrero Perez, Adam A. Hecht, Samuel C. Hedges, Patrick Huber, Vsevolod Ivanov, Igor Jovanovic, Giti A. Khodaparast, Brenden A. Magill, Jose Maria Mateos, Maverick Morrison, Nicholas W. G. Smith, Patrick Stengel, Stuti Surani, Nikita Vladimirov, Keegan Walkup, Christian Wittweg, and Xianyi Zhang. Nuclear recoil detection with color centers in bulk lithium fluoride, 2025. 1, 2, 3, 9

[2] Sebastian Baum, Patrick Stengel, Natsue Abe, Javier F. Acevedo, Gabriela R. Araujo, Yoshihiro Asahara, Frank Avignone, Levente Balogh, Laura Baudis, Yilda Boukhtouchen, Joseph Bramante, Pieter Alexander Breur, Lorenzo Caccianiga, Francesco Capozzi, Juan I. Collar, Reza Ebadi, Thomas Edwards, Klaus Eitel, Alexey Elykov, Rodney C. Ewing, Katherine Freese, Audrey Fung, Claudio Galelli, Ulrich A. Glasmacher, Arianna Gleason, Noriko Hasebe, Shigenobu Hirose, Shunsaku Horiuchi, Yasushi Hoshino, Patrick Huber, Yuki Ido, Yohei Igami, Norito Ishikawa, Yoshitaka Itow, Takashi Kamiyama, Takenori Kato, Bradley J. Kavanagh, Yoji Kawamura, Shingo Kazama, Christopher J. Kenney, Ben Kilminster, Yui Kouketsu, Yukiko Kozaka, Noah A. Kurinsky, Matthew Leybourne, Thalles Lucas, William F. McDonough, Mason C. Marshall, Jose Maria Mateos, Anubhav Mathur, Katsuyoshi Michibayashi, Sharlotte Mkhonto, Kohta Murase, Tatsuhiro Naka, Kenji Oguni, Surjeet Rajendran, Hitoshi Sakane, Paola Sala, Kate Scholberg, Ingrida Semenec, Takuya Shiraishi, Joshua Spitz, Kai Sun, Katsuhiko Suzuki, Erwin H. Tanin, Aaron Vincent, Nikita Vladimirov, Ronald L. Walsworth, and Hiroko Watanabe. Mineral detection of neutrinos and dark matter. a whitepaper. *Physics of the Dark Universe*, 41:101245, 2023. 2

[3] Jan Niklas Böhm, Philipp Berens, and Dmitry Kobak. Unsupervised visualization of image datasets using contrastive learning, 2023. 6

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021. 2

[5] Kashyap Chitta, Jianwei Feng, and Martial Hebert. Adaptive semantic segmentation with a strategic curriculum of proxy labels, 2018. 2

[6] Nikolaos Dionelis and Nicolas Longepe. Fine-tuning foundation models with confidence assessment for enhanced semantic segmentation. *IEEE Geoscience and Remote Sensing Letters*, PP:1–1, 01 2024. 2

[7] Laura Dominé and Kazuhiro Terao. Scalable deep convolutional neural networks for sparse, locally dense liquid argon time projection chamber data. *Phys. Rev. D*, 102:012005, Jul 2020. 2

[8] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with numpy. *Nature*, 585(7825):357–362, September 2020. 3

[9] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners, 2021. 4

[10] Xiangtai Li, Henghui Ding, Haobo Yuan, Wenwei Zhang, Jiangmiao Pang, Guangliang Cheng, Kai Chen, Ziwei Liu, and Chen Change Loy. Transformer-based visual segmentation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10138–10163, 2024. 2

[11] Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, and Pietro Perona. A closer look at benchmarking self-supervised pre-training with image classification - international journal of computer vision, Apr 2025. 2

[12] Dushyant Mehta, Andrii Skliar, Haitam Ben Yahia, Shubhankar Borse, Fatih Porikli, Amirhossein Habibian, and Tijmen Blankevoort. Simple and efficient architectures for semantic segmentation, 2022. 2

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 3

[14] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 3

[15] Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, Simon Hawkins, Gfyoung, Sinhrks, Matthew Roeschke, et al. pandas-dev/pandas: Pandas. *Zenodo*, 2020. 3

[16] Aagam Shah, Joshua A. Schiller, Isiah Ramos, James Serrano, Darren K. Adams, Sameh Tawfick, and Elif Ertekin. Automated image segmentation of scanning electron microscopy images of graphene using u-net neural network. *Materials Today Communications*, 35:106127, 2023. 2

[17] Sam Young, Yeon jae Jwa, and Kazuhiro Terao. Particle trajectory representation learning with masked point modeling, 2025. 2, 6, 9