Title: *Protein to Image Transformer: Amino Acid "Translation" for Computer Science Concentrators*
Who: Kylash Ganesh (kganesh), Adeethyia Shankar (ashank10), Andy Le (andyhuyle)
Team Name: Neural Net-Workers

Introduction

From carrying oxygen to our brain to regulating how our DNA is expressed, proteins are an essential part of how living things operate. While this is the case, they still remain one of the great frontiers of biology that we yet to fully decode. After the introduction of AlphaFold in 2018, an abundance of protein structures have been proposed. A question from this stepping stone is where do these novel protein structures act within cells. Our project reimplements the research paper "CELL-E: A Text-to-Image Transformer for Protein Image Prediction" (Khwaja, E., Song, Y.S., Huang, B. (2024)) using TensorFlow instead of its original PyTorch implementation. The objective of this paper is to predict a refined representation of protein localization given an amino acid sequence and nucleus image, essentially allowing us to see where these novel proteins work.

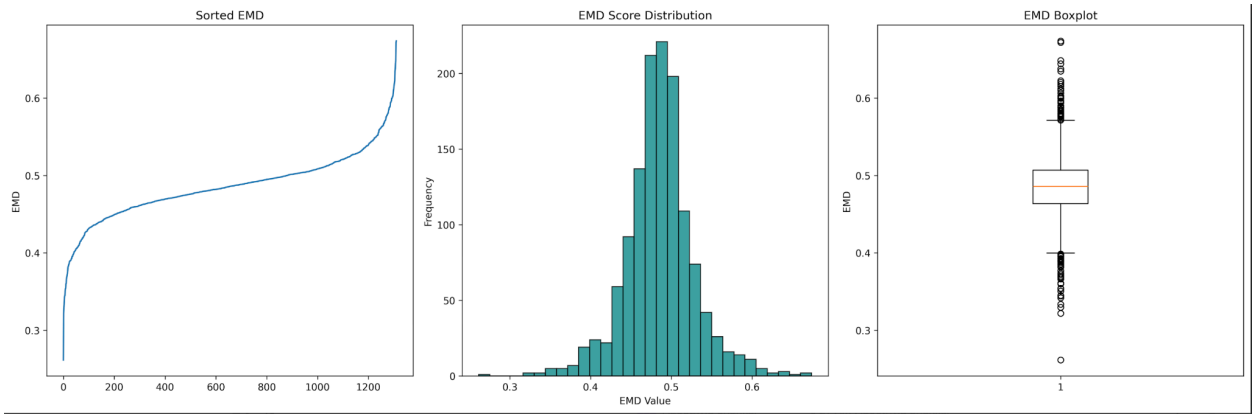
Methodology

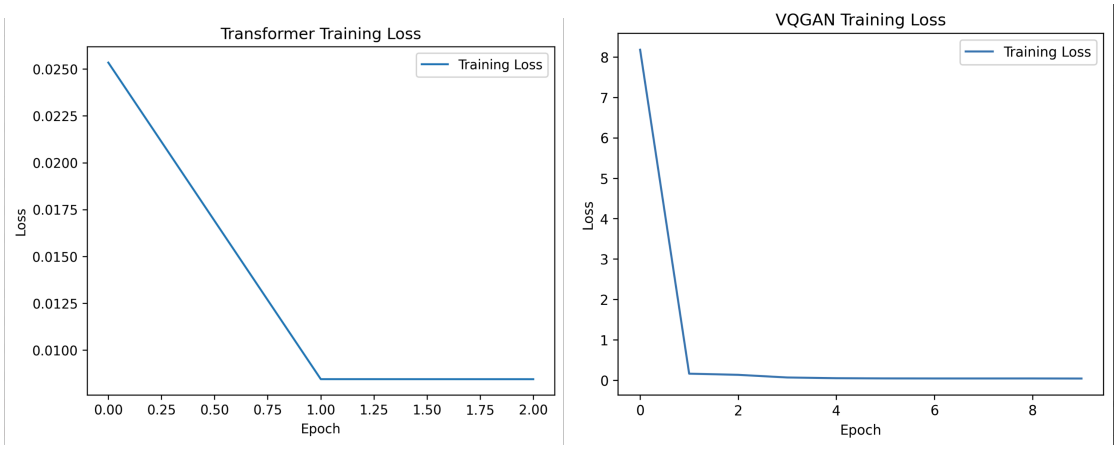
We split up our model creation workflow into three sections: data preprocessing, model architecture, and metric collection and analysis. We chose to use the same *OpenCell* cell image database as the original paper due to the nature of how specific biological databases are in terms of features they provide. Downloading the original (.tif) images from this dataset, we then converted them into a (.png) that was easier to process. These images were then paired with their corresponding amino acid sequence and then localized into a data.csv file for ease of access. The images from our data.csv were then passed into our custom VQGAN models, which tokenized the images to extract meaningful embeddings. These image embeddings were then concatenated with our amino acid embeddings obtained through a industry standard BERT amino acid tokenizer. This final concatenated tensor was then passed into our transformer and used to create our final density maps for cell images. Finally, we collected metrics such as VQGAN training loss and Earth Mover's Distance data to analyze our model better and better understand our outputs.

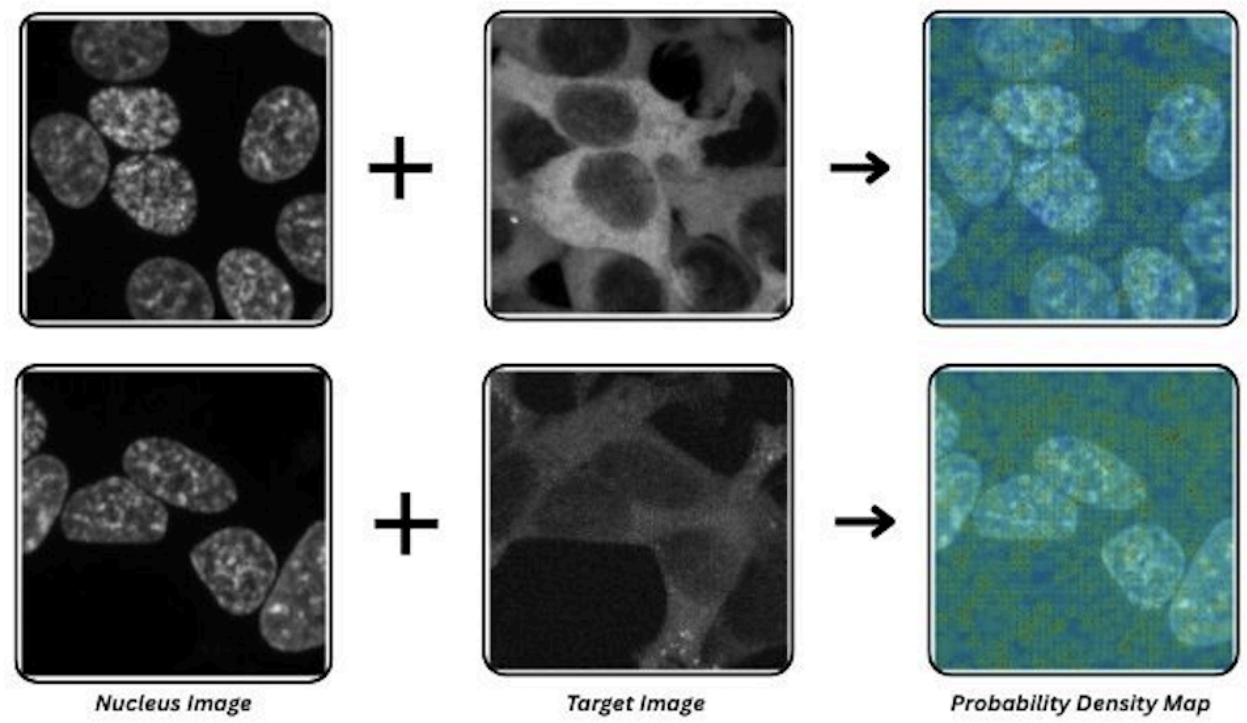
Results

Our model created a density map output for a given cell image and an amino acid sequence. While our model created an output, we are not sure that it is a fully morphologically accurate representation of the actual protein localization (as we cannot say that random chance does not play a full role in creating our model's final density maps).

Here our some of our metrics and results:


Sorted EMD
EMD
EMD Score Distribution
Frequency
EMD Value
EMD Boxplot
EMD


Transformer Training Loss
Training Loss
Loss
Epoch
VQGAN Training Loss
Training Loss
Loss
Epoch


Nucleus Image
Target Image
Probability Density Map

Challenges

The hardest part of the project we have encountered is processing the data. The model requires three distinct types of data: protein sequences, nucleus images, and protein images. In addition, the nucleus images and protein images need to be fed through a generative model (i.e. the VQGAN) to obtain the "true" inputs/outputs that our transformer will learn. In addition to downloading the data from OpenCell (the same dataset that the original paper used), we had to do preprocessing and rewrite the data loader. Moreover, we had to manually generate a data.csv file with three columns: nucleus_img_path, protein_img_path, and amino_acid_seq. The original "cell" images from the OpenCell dataset were in a .tiff image format with 2 frames in the file, one corresponding to "nucleus" cell image and the other corresponding to "protein" cell images. We needed to split the .tiff image file into 2 separate .png cell images (one for the nucleus and one for the protein). A challenge with this workflow is that we had to interface with the AWS server host for the OpenCell dataset. Furthermore, actually saving the images to the csv file was confusing. After saving these images, we looked up each image's corresponding ENSG id for the protein that acts on each cell. Using this id, we downloaded the amino acid sequence from the Ensembl gene database and put them with their corresponding pair of cell images in the csv.

Reflection

Overall, we feel great about our project. Before starting, we were doubtful about whether or not we would even get an output. Although we had the GitHub from the original paper, most of the code and functions the authors used were very complicated and were outside the scope of our knowledge. However, after breaking down the paper, we were able to extract the core components and reimplement the paper to great a working output, which we were very proud of. Although the probability density maps in the original paper were cleaner and more defined than our outputs, we are very satisfied that we were able to get an output with any probability density.

Our model worked in the way we expected. We got an output of a probability density map of where the protein is most likely to be. With that being said, our output is quite noisy. We would have preferred our output to be cleaner and more defined, similar to the original paper's output images

Initially, since we had the GitHub of the original paper, our approach was to re-implement each function of the code 1-to-1 in TensorFlow. However, there were two reasons why this was infeasible. Firstly, the original paper's code was extensive and complicated. We tried our best to understand the author's code, but we found it difficult because their coding and biology knowledge far outweighed our knowledge. Secondly, their code was written in PyTorch, which is more fine-tuned for researchers. This enabled the authors to add more customizability to their

implementation. However, TensorFlow is made for production code, so it is more streamlined, and thus made it difficult for us to copy each function exactly. Consequently, we pivoted to breaking down the paper into its core components and reimplementing those in TensorFlow. For instance, we broke the paper down into 5 main parts (1) Preprocessing, (2) VQGAN, (3) Concatenation, (4) Transformer, and (5) Density Map. We focused on re-implementing these components at a high level, which helped us avoid getting stuck at nuances in the original code, but still led us to achieving an output.

If given more time, we think we could have fine-tuned our data pre-processing pipeline to format our data in more biologically significant ways. For example, we think targeted cropping of cell images to actually crop the relevant morphological features of a cell would have improved our models predictive power for a given protein and cell image input. Furthermore, incorporating other more robust datasets in our main input dataset would further contribute to fine-tuning our model's training. Another improvement we could have tried to implement would have been incorporating the actual protein structure into our inputs rather than just their amino acid sequences. This would give more morphologically relevant embeddings that our model could have used to train with. Additionally, as mentioned previously, the authors of the original paper fine-tuned their models to get better outputs using their advanced knowledge of biology and coding. If given more time, we would have liked to understand these adjustments they made to help us get a better output

Our biggest takeaway from this project is that deep learning models are applicable to a wide range of fields. The only limitation is the availability of data within the field one is interested in. Furthermore, translating knowledge and features from one field to another requires a great deal of effort to make each work with the other. The majority of the time for our project was spent on preprocessing our data from the original cell microscopy images from the *OpenCell* dataset. Features that biologists might consider "important" might not necessarily align with what computer scientists might consider "important." The collaboration between different fields is a mutually beneficial relationship for science as a whole.

Overall, we learned a lot from this project! It challenged us—pushing us to better understand the concepts we learned in class as well as learn new concepts. We are very proud of the work we put into this project!