

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Autoregressive Text-to-Image Model for Protein Localization Prediction

Permalink

<https://escholarship.org/uc/item/97x8x780>

Author

Khwaja, Emaad

Publication Date

2023

Peer reviewed|Thesis/dissertation

Autoregressive Text-to-Image Model for Protein Localization Prediction

by

Emaad Khwaja

DISSERTATION

Submitted in partial satisfaction of the requirements for degree of

DOCTOR OF PHILOSOPHY

in

Bioengineering

in the

GRADUATE DIVISION

of the

UNIVERSITY OF CALIFORNIA, SAN FRANCISCO

AND

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved:

DocuSigned by:

Bo Huang

Bo Huang

1E7C3BA93EB54F3...

Chair

DocuSigned by:

Steven Altschuler

Steven Altschuler

DocuSigned by:

Michael Keiser

Michael Keiser

DocuSigned by:

Laura Waller

Laura Waller

ED8FDE5E6E3B4D6...

Committee Members

Copyright 2023

by

Emaad Khwaja

Acknowledgments

Very special thanks to my advisor Bo Huang for all the guidance and support. I also want to thank Yun Song for providing invaluable feedback.

Abstract

Autoregressive Text-to-Image Model for Protein Localization Prediction

Emaad Khwaja

Accurately predicting cellular activities of proteins based on their primary amino acid sequences would greatly improve our understanding of the proteome. we present CELL-E, a text-to-image transformer model that generates 2D probability density images describing the spatial distribution of proteins within cells. Given an amino acid sequence and a reference image for cell or nucleus morphology, CELL-E predicts a more refined representation of protein localization, as opposed to previous *in silico* methods that rely on pre-defined, discrete class annotations of protein localization to subcellular compartments.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Background | 1 |
| 1.2 | Related Work | 3 |
| 2 | The CELL-E Model | 7 |
| 2.1 | Dataset | 7 |
| 2.2 | Model Architecture | 9 |
| 2.3 | Modular Training Paradigm | 13 |
| 3 | Results | 26 |
| 3.1 | Image Reconstruction | 26 |
| 3.2 | 1D Prediction Comparison | 32 |
| 4 | Discussion | 35 |
| 4.1 | Extra-domain Prediction | 35 |
| 4.2 | Reproducing Experimental Results <i>in silico</i> | 37 |
| 4.3 | Ablation Studies | 41 |
| | Bibliography | 46 |

A Additional Tables

List of Figures

| | | |
|-----|--|----|
| 1.1 | Given an amino acid sequence and a reference nucleus image, CELL-E makes a prediction of protein localization with respect to the nucleus as a 2D probability density function, shown as heatmap, with color indicating relative confidence for each pixel. | 3 |
| 1.2 | Left: Images of ADP ribosylation factor 1 (green) from Human Protein Atlas [1]. Right: Left image with nucleus channel (blue) and ER channel (channel) visible. Although the HPA localization label is “Cytosol, Membrane,” protein signal is clearly present in the nucleus (red box) and ER (white box). | 5 |
| 2.1 | DALL-E-like model with only amino acid sequence as the input. The sequence (left column) is used as input. The middle 3 columns show separate predicted images from random initialization. The true protein image is shown in the right column. | 10 |
| 2.2 | CELL-E model with Protein Threshold VQGAN replaced with Protein Image VQGAN. We note morphological and localization similarity with the ground truth images in the generated output images. | 11 |
| 2.3 | 512 image patches extracted from the nucleus reference VQGAN | 18 |

| | | |
|-----|---|----|
| 2.4 | Example of thresholding process to convert protein image (left) to thresholded ground truth image (right) for CELL-E model | 19 |
| 2.5 | 512 image patches extracted from the protein threshold VQGAN | 20 |
| 2.6 | Graphical depiction of CELL-E. Solid lines correspond to pre-trained components. Gray dashed lines are learned in Phase 1 and 2 (Reference Image and Protein Threshold VQGANs). Black dashed lines correspond to components learned in Phase 3. A start token is prepended to the sequence and the final protein image token is removed. The amino acid sequence embedding from the model is preserved, and embedding spaces for the image tokens are cast in the same depth and concatenated with the amino acid sequence embedding. The transformer is tasked with reproducing the original sequence of tokens (e.g., the input sequence with start token shifted to the right one position). | 21 |
| 2.7 | Simplified example of probability map calculation. Each circle corresponds to an image token within the quantized VQGAN embedding space. Each PDF patch (yellow) is obtained as a weighted sum over all protein threshold image VQGAN codebook vectors. | 22 |
| 3.1 | Nucleus Image (left), Protein Image (middle), and Overlay (Right). The alpha value for the protein channel in the right column is set to .7. Overlay is used as the “Original Image” in Fig. 3.2 and Fig. 3.3. | 27 |

| | | |
|-----|---|----|
| 3.2 | Prediction results of several types of proteins from the validation set, unseen to the model during training. The nucleus channel is depicted in grayscale, and the protein channel is shown as an overlay in red (Fig. 3.1 for clarification). The thresholded image (Column 2) is designated “Ground Truth” because those are the types of images exposed to the model during training. The predicted probability map is obtained from a weighted sum of potential image patches and normalized to 1. | 28 |
| 3.3 | More prediction results from the validation set. We observe a high degree of spatial awareness from the model, notably in UDP-xylose-acetylglucosamine Transporter, which accurately predicts signal between cell nuclei with high confidence. | 29 |
| 3.4 | Masking procedure depicted. | 31 |
| 4.1 | Predicted localization of GFP and modified-GFP sequences. | 36 |
| 4.2 | UMAP of encodings. We observe a largely nuclear protein “neighborhood” (left) and a cytoplasmic neighborhood (right). Prepending the NLS to GFP causes a shift towards the nuclear protein neighborhood. | 37 |

| | | |
|-----|---|----|
| 4.3 | CELL-E’s predicted localization (images) of eGFP fusions from [2] and corresponding localization annotations (table) from the original paper. In the table on the right hand side, green indicates agreement between CELL-E and experimental results, while red indicates disagreement. <i>aa 1-199</i> contains the entire N-terminus region. <i>aa 1-146</i> only contains Motifs I and V. <i>aa 1-67</i> only contains Motif-I. <i>aa 148-199</i> contains Motif II, III, IV and V. | 38 |
| 4.4 | Attention weights for significant tokens when patches containing a large percentage of protein are selected (bottom-right figure). Previous computationally identified putative NLSs are boxed in black (top figure). These are <i>aa 59-65</i> (Motif I, KKHKEKE), <i>aa 150-156</i> (Motif II, KKIKTED), <i>aa 174-180</i> (Motif III, KKPKNKD), and <i>aa 192-198</i> (Motif IV, KKKPKKE). Additionally, the new NLS identified in Mo et. al.[2], Motif V (<i>aa 117-146</i>), is highlighted. | 40 |
| 4.5 | Ablation Plot. “Fixed” and “Unfixed” embedding refer exclusively to the amino acid embeddings. Image embeddings are always unfixed. Mean values and standard deviation are marked in black. \sim 150 points are randomly selected for display out of 1303 total predictions per model. | 42 |
| 4.6 | Cross-entropy loss, the metric used to evaluate performance, is shown after 130 epochs. | 45 |

List of Tables

| | | |
|-----|---|----|
| 3.1 | Image Accuracy | 30 |
| 3.2 | Nuclear Localization Prediction Accuracy | 33 |
| A.1 | Depicted Results Localization Annotations | 62 |
| A.2 | Full Results Table | 62 |

Chapter 1

Introduction

1.1 Background

In recent years, advancements in sequencing technologies have allowed for the comprehensive cataloging of proteins and their amino acid sequences across a wide range of organisms [3]. Despite this progress, the exact functions and cellular dynamics of many proteins remain unclear. In order to gain a deeper understanding of these proteins, researchers have sought ways to predict their properties, including structure, interactions, subcellular localization, and trafficking patterns, from their amino acid sequences. This type of computational analysis has the potential to shed light on the “dark matters” of the proteome and enable large-scale screening before expensive experimental validation. These tools have numerous applications in biomedical research, such as drug design and therapeutic target discovery [4].

In these studies, our focus is on predicting subcellular localization of proteins from their amino acid sequences, with nuclei images providing spatial context for their cellular functions. The localization of a protein to a specific subcellular compartment can be driven by either active transport or passive diffusion in conjunction with specific protein-protein in-

teractions, often involving localization “signals” in the amino acid sequence [5–7]. In many cases, however, the exact mechanisms for sequence recognition and trafficking are not yet fully understood [8]. For example, there is ongoing debate about the mechanism behind the import of proteins via the nuclear localization sequence (NLS) [9]. Given these challenges, machine learning utilizing existing knowledge of protein localization has become a particularly useful tool.

Although computational prediction of protein subcellular localization from primary amino acid sequences is an active area of research, most works train the model with class annotation of subcellular compartments (e.g., nucleus, plasma membrane, endoplasmic reticulum, etc.) [10] which are available from databases such as UniProt [11]. This approach has two major limitations. First, many proteins are present in different and variable amounts across multiple subcellular compartments. Second, protein localization could be highly heterogeneous and dynamic depending on the cell type and cell state (including cell cycle state). Neither of these two aspects have been captured by existing discrete class annotations. Consequently, machine-learning-based protein localization prediction still has limited applications. Furthermore, to assist mechanistic discoveries, it is highly desirable for the machine learning models to be explainable.

To investigate the relationship between sequence and subcellular localization, we present CELL-E, a text-to-image transformer model which predicts the probability of protein localization on a per-pixel level from a given amino acid sequence and a conditional reference image for the cell or nucleus morphology and location (Fig. 1.1). It relies on transfer learning

via amino acid embeddings from a pre-trained protein language model and two quantized image encoders trained from a live-cell imaging dataset. By generating a two-dimensional probability density function (2D PDF) atop the reference image, CELL-E naturally accounts for multi-compartment localization and the cell type/state information implicitly encoded by the cell morphology. We demonstrate the capability of CELL-E to predict localization of proteins, identify changes in localization due to mutations, and uncover sequence features correlated with the specification of subcellular protein localization.

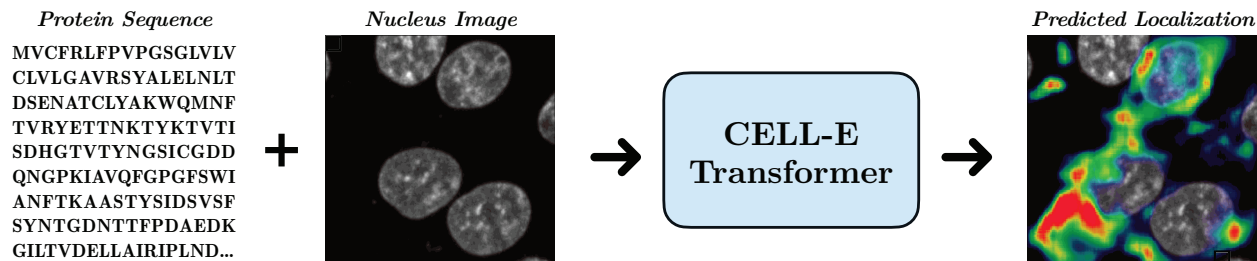


Figure 1.1: Given an amino acid sequence and a reference nucleus image, CELL-E makes a prediction of protein localization with respect to the nucleus as a 2D probability density function, shown as heatmap, with color indicating relative confidence for each pixel.

1.2 Related Work

Protein Localization Prediction

While traditional supervised approaches, such as stochastic modeling, have been limited by feature representation or computation time, deep learning has proven to be a powerful

tool in predicting localization [12]. With the ability to optimize millions of parameters, deep neural networks have shown the ability to represent complex patterns in a manner that traditional manual feature extraction cannot [13, 14]. The success of language models in protein prediction tasks suggests that patterns dictating these structures are buried within residue sequences [15, 16].

Protein localization is typically framed as a class prediction task. 1D localization predictors take the primary sequence as input and produce a fixed-length vector, with each entry corresponding to a subcellular location and the values being probability values. However, these methods have limitations [17, 18]. Discrete classifications for contiguous regions of the cell, such as the nuclear membrane, can be ambiguous and may have flawed annotations in established datasets. [19]. Additionally, these methods do not account for the influence of local cellular geometries [20–23] and cell states [24, 25] on transport dynamics. For example, one would expect significantly high amounts of transcription factors for DNA replication in the S-phase of the cell cycle, but not during cell separation in mitosis [26]). Furthermore, discrete classifications for regions of the cell which are contiguous like the nuclear membrane, endoplasmic reticulum, and parts of the Golgi apparatus can be spatially ambiguous and have flawed annotation within established datasets (Fig. 1.2)

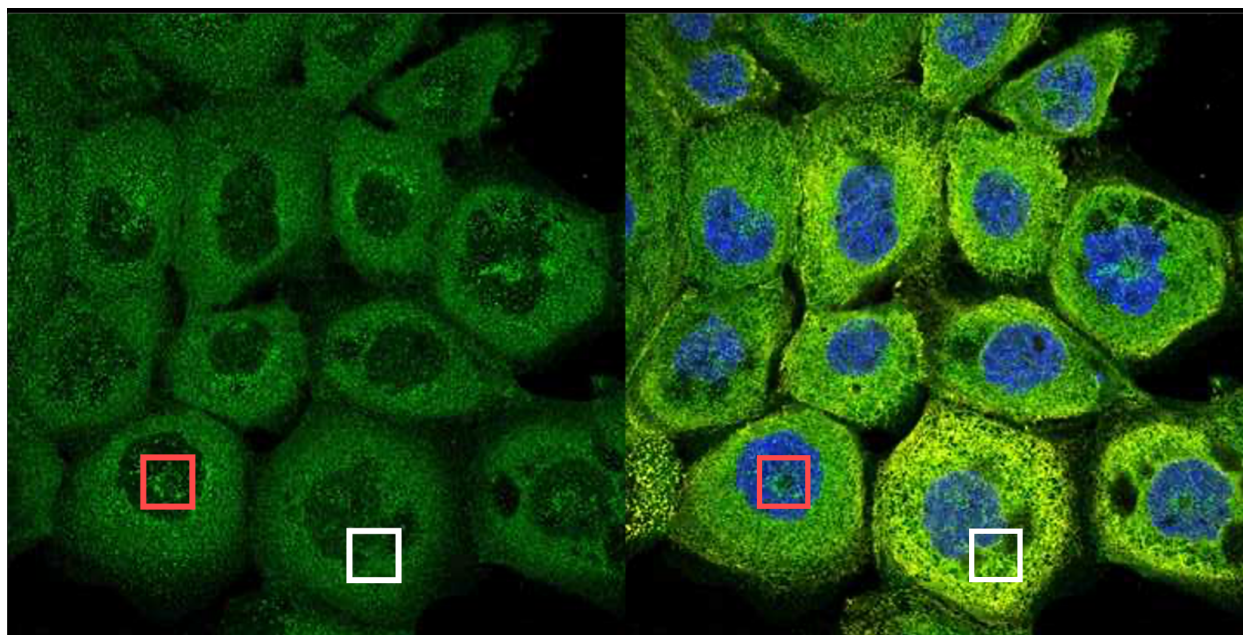


Figure 1.2: Left: Images of ADP ribosylation factor 1 (green) from Human Protein Atlas [1]. Right: Left image with nucleus channel (blue) and ER channel (channel) visible. Although the HPA localization label is “Cytosol, Membrane,” protein signal is clearly present in the nucleus (red box) and ER (white box).

Protein Language Models

Natural language processing (NLP) has found applications in amino acid sequence encoding, due to the long contextual dependencies of amino acids in a protein’s folded three-dimensional structure [27]. Self-supervised models from the NLP field have demonstrated excellent performance in predicting protein properties from amino acid sequence inputs [28–31]. These models are trained on millions of amino acid sequences from databases such as BFD [32], UniRef [33], Pfam [34], and Protein Data Bank [35]. The language models

have proven effective in downstream tasks like structure prediction, evolutionary analysis, and protein engineering [16], with LSTM and attention-based models achieving particularly impressive results [36]. UniRep is an LSTM model that predicts the next amino acid in a variable length sequence [37], while BERT uses bidirectional masked language modeling to predict the identity of masked tokens throughout the sequence [38]. Facebook’s Evolutionary Scale Model (ESM) is a state-of-the-art masked-language model model, pre-trained with 250 million amino acid sequences and over 700 million parameters [29].

Text-To-Image Generation

Ramesh et. al. [39] demonstrated true zero-shot text-to-image generation with their model, DALL-E. Unlike previous models, DALL-E utilized an autoregressive framework, which was trained on a joint distribution of text and image tokens, enabling it to make novel image predictions with high fidelity. In contrast, earlier models based on variational autoencoders (VAE) [40] or Generative Adversarial Networks (GAN) [41] performed poorly when generating images outside of the training data, resulting in distorted images and artifacts [42–44].

Chapter 2

The CELL-E Model

2.1 Dataset

Data availability in this domain is a large obstacle. DALL-E was trained on 250 million text-images pairs. amino acid sequence databases for which there are corresponding cellular images are comparatively smaller. For example, the largest publicly available dataset with annotated protein images in human cells, Human Protein Atlas (HPA), contains 12,003 unique proteins with just 82,000 immunofluorescence staining and confocal microscopy (ICC-IF) images [1]. Additionally, image captions are often much shorter than full amino acid sequences. CogView and DALL-E relied on text encoding which were only 256 word tokens in length. amino acid sequences can contain upwards of 600 amino acid residues, where any portion could have functional and localization-critical domains [45]. Another difference is NLP utilize dictionaries which contain thousands of possible words, whereas a protein language model only needs to encode at least 20 entries for the canonical amino acids.

In this work, we chose to use the OpenCell dataset [46] for training and validation because

of its strong image quality, uniformity of imaging and cell conditions, and availability of consistent morphology reference images. OpenCell contains a library of 1,311 CRISPR-edited HEK293T human cell lines, each with one target protein fluorescently tagged via the split-mNeonGreen2 system. For each cell line, the OpenCell imaging dataset contain 4-5 confocal images of the tagged protein with accompanying DNA staining as the reference for nuclei morphology. The cells were imaged alive, which collectively represents a more accurate depiction of protein distribution within the cell than immunofluorescence.

The OpenCell dataset was selected because the split-fluorescent protein fusion system allows for tagging endogenous genomic proteins, maintaining local genomic context, and the preservation of native expression regulation [46]. This last point is specifically important when compared to the previously mentioned HPA, which contains $\sim 10\times$ more proteins and images. ICC-IF, which is the technique used for obtaining HPA images, requires several rounds of fixation and washing [47]. This means the proteins are not observed in a live cell, are subject to signal loss, artifacts, and/or relocalization events, and therefore does not represent the true nature of protein expression and distribution within a cell [48].

Each protein entry in OpenCell is accompanied by multiple high-resolution 3D confocal images containing multiple cells [46]. Having multiple live cells enables the potential for protein distribution to be captured at several time points within a cell's lifetime. To reduce computational cost for our demonstration, we converted a 3D z-stack into a 2D maximum intensity projection [49], which still clearly depicts most subcellular structures and allow subtle subcellular protein localization differences to be distinguished from the OpenCell

images [46].

Training and validation sets were generated by randomly splitting the OpenCell dataset by protein 80%-20% training-validation. For every stage of training, models were blind to sequences, nuclei, and protein images contained within the validation set. We utilize data augmentation techniques such as random horizontal and vertical flips on images during training.

2.2 Model Architecture

Overview

CELL-E (Fig. 2.6) is inspired by the text-to-natural-image generation model of DALL-E [39]. We found that the method from (author?) [39] does similarly allow for truly zero-shot protein image prediction (Fig. 2.1), however our goal for image generation extends beyond visual fidelity and includes a degree of spatial accuracy. This is crucial for capturing the dynamic process of protein abundance in cells, which fluctuates with respect to cell state and environmental factors. To overcome this challenge, our proposed model includes conditioning on a nucleus image channel as a spatial reference (Fig. 2.2). Similar to DALL-E, our model autoregressively learns text and image tokens as a single stream of data. On the other hand, While the goal of general text-to-image models is to produce images with high perceptual strength, they do not necessarily aim for quantitative accuracy [39, 50, 51]. Therefore,

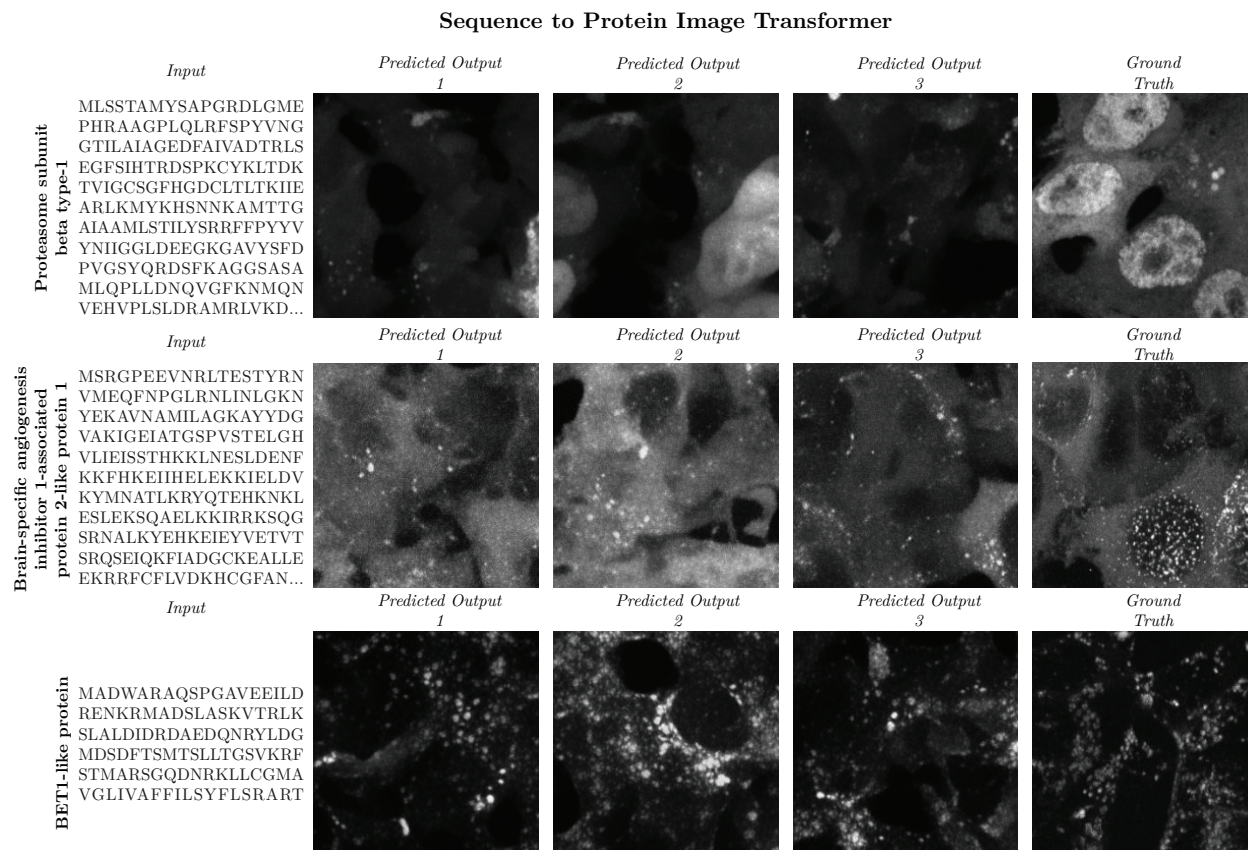


Figure 2.1: DALL-E-like model with only amino acid sequence as the input. The sequence (left column) is used as input. The middle 3 columns show separate predicted images from random initialization. The true protein image is shown in the right column.

CELL-E was designed with the following considerations:

1. **Transfer learning.** Training CELL-E requires a library of cellular images and corresponding morphological reference images for a large number of proteins. For this purpose, we utilized the recently established OpenCell library[46], which contains a library of 1,311 CRISPR-edited HEK293T human cell lines, each having one target

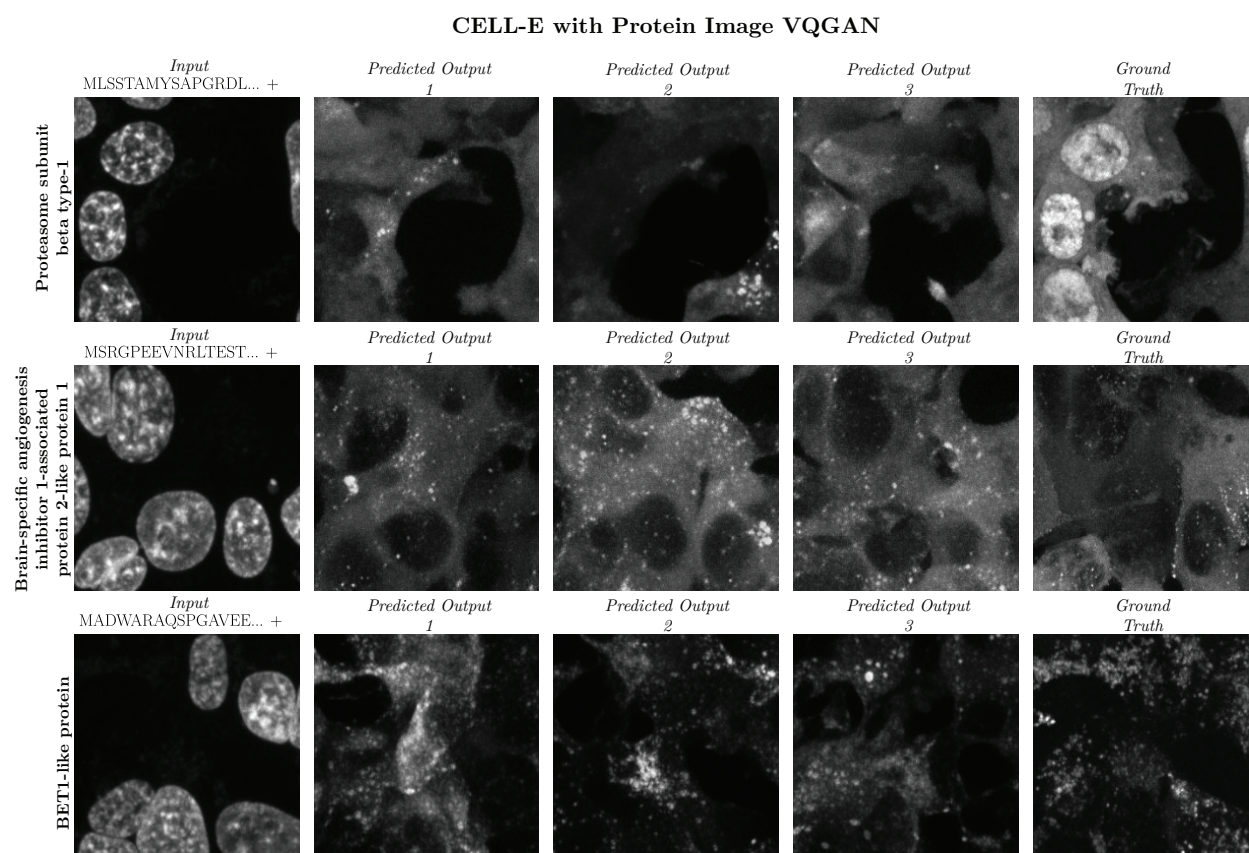


Figure 2.2: CELL-E model with Protein Threshold VQGAN replaced with Protein Image VQGAN. We note morphological and localization similarity with the ground truth images in the generated output images.

protein fluorescently tagged and imaged by confocal microscopy with accompanying DNA staining as the reference for nuclei morphology. The high image quality and consistency makes OpenCell a good choice as the training and validation dataset (See Section 2.1 for more information). Still, data availability in this domain remains a large obstacle. For example, DALL-E was trained on 250 million text-images pairs [39], orders of magnitude larger than the OpenCell dataset. We utilize transfer learning by incorporating frozen embeddings from a pre-trained protein language model as the input representation of the amino acid text sequence. This approach reduces the number of learned parameters, thereby alleviating the burden for CELL-E to also learn the amino acid sequence space. This allows training to be concentrated on the relationship between sequence and image tokens. We evaluated multiple protein language models (see Supplementary Information and Table A.2) and eventually chose the BERT-based model from Rao et. al. [16], which we refer to as the TAPE model, for subsequent work.

2. **Morphological reference.** In our initial efforts, we found that a transformer using just the amino acid tokens and protein image tokens is capable of generating cell-like images from the amino acid sequence alone (Fig. 2.1). However, quantifying protein localization information in the generated images is challenging. Furthermore, an estimation of a single snapshot of protein localization is not necessarily a quantifiable indication of global behavior. Therefore, in addition to amino acid tokens and protein

image tokens, we add a 3rd embedding space to include tokens representing the overall cell morphology from a reference image (See Fig. 2.2.). The reference image provides the model with information regarding the localization of subcellular structures and compartments. Moreover, cell morphology implicitly provides the cell type and cell state context for CELL-E predictions.

3. **Image model.** Instead of the Vector Quantized Variational Autoencoder (VQVAE) previously used to analyze OpenCell imaging data [52], we chose to use Vector Quantized Generative Adversarial Network (VQGAN) [53] which produces images with comparatively higher spatial frequency. To simplify the task of the protein image VQGAN, we let it predict per-pixel binary representations of protein localization (i.e., a thresholded image) (Fig. 2.4). This allows us to use the marginal probabilities predicted for each image token from CELL-E to create a weighted sum on the image tokens. This latent space linear combination is then used to generate a continuous 2D probability density function of protein localization, which resembles a gray-scale image (Fig. 2.7). We note that the same model can also be trained to output gray-scale images directly (See Section 1.2 and Fig. 2.2).

2.3 Modular Training Paradigm

We use a multi-phase training approach similar to DALL-E, but our model also uses pre-trained language-model input embeddings for the amino acid text sequences via TAPE:

- **Phase 1** A Vector Quantized-Generative Adversarial Network (VQGAN) [53] is trained to represent a single channel 256×256 nucleus image as a grid comprised of 16×16 image tokens (Fig. 2.3), each of which could be one of 512 tokens.
- **Phase 2** A similar VQGAN is trained on images corresponding to binarized versions (Fig. 2.4) of protein images. These tokens represent the spatial distribution of the protein (Fig. 2.5).
- **Phase 3** The VQGAN image tokens are concatenated to 1000 amino acid tokens for the autoregressive transformer which models a joint distribution over the amino acids, nucleus image, and protein threshold image tokens.

The optimization problem is modelled as maximizing the evidence lower bound (ELBO) [40, 54] on a joint likelihood distribution over protein threshold images u , nucleus images x , amino acids y , and tokens z for the protein threshold image:

Theorem 1.

$$p_{\theta, \psi}(u, x, y, z) = p_{\theta}(u \mid x, y, z)p_{\psi}(x, y, z)$$

This is bounded by:

Theorem 2.

$$\ln p_{\theta,\psi}(u, x, y) \geq \mathbb{E}_{z \sim q_{\phi}(z|u)}[\ln p_{\theta}(u | x, y, z)] - KL(q_{\phi}(x, y, z | u), p_{\psi}(x, y, z))$$

where q_{ϕ} is the distribution 16×16 image tokens from the VQGAN corresponding to the threshold protein image u , p_{θ} is the distribution over protein threshold generated by the VQGAN given the image tokens, and p_{ψ} indicates the joint distribution over the amino acid, nucleus, and protein threshold tokens within the transformer.

We utilized 4×NVIDIA RTX 3090 TURBO 24G GPUs for this study. 2 GPUs were utilized for training VQGANs via distributed training. Only a single GPU is ever used to train CELL-E models.

Our computer also contained 2×Intel Xeon Silver and 8×32768 mb 2933MHz DR×4 Registered ECC DDR4 RAM.

Amino Acid Embedding

For language transformers, it is necessary to learn both input embedding representations of a text vector as well as attention weights between embeddings [36]. In practice, this creates a need for very large datasets [55]. The OpenCell dataset contains 1,311 proteins, while the human body is estimated to contain upwards of 80,000 unique proteins [56]. It is unlikely that such a small slice could account for the large degrees of variability found in nature.

In order to overcome this obstacle, we opted for a transfer learning strategy, where fixed amino acid embeddings from a pre-trained language model exposed to a much larger dataset were utilized. We found the strongest performance came from TAPE embeddings [16]. Utilizing pre-trained embeddings had the two-fold benefit of giving our model a larger degree of protein sequence context, as well as reducing the number of trained model parameters, which allowed us to scale the depth of our network.

We tried training using random initialization for amino acid embeddings (See Section 4.3), however, we noted overfitting on the validation set image reconstruction and high loss on validation sequences. We also experimented with other types of protein embeddings, including UniRep [15] and ESM1-b [29].

Nucleus Image Encoder

Training both image VQGANs maximizes ELBO with respect to ϕ and θ . The VQGAN improves upon existing quantized autoencoders by introducing a learned discriminator borrowed from GAN architectures [53]. The Nucleus Image Encoder is a VQGAN which represents 256×256 nucleus reference images as $256 \cdot 16 \times 16$ image patches. The VQGAN codebook size was set to $n = 512$ image patches.

VQGAN code was obtained from Esser et. al. [53], which was available via MIT license (Copyright (c) 2020 Patrick Esser and Robin Rombach and Björn Ommer).

The model was trained on random 256×256 crops of 512×512 nuclei images. The model

was initially trained solely using mean-squared error reconstruction loss. After 50,000 steps, ~ 7 epochs, the discriminator loss term was introduced. This term helps with reducing the blurriness typically associated with VAEs. 512 discrete image codes were learned. Training occurred until the model reached convergence (at 344 epochs).

Protein Threshold Image Encoder

The protein threshold image encoder learns a dimension reduced representation of a discrete binary PDF of per-pixel protein location, represented as an image image. We adopt a VQGAN architecture identical to the Nucleus VQGAN. The VQGAN serves to approximate the total set of binarized image patches. While in theory a discrete lookup of each pixel arrangement is possible, this would require $\sim 1.16 \times 10^{77}$ entries, which is computationally infeasible. Furthermore, some distributions of pixels might be so improbable that having a discrete entry would be a waste of space.

Protein images are binarized with respect to a mean-threshold, via:

$$\bar{u}_{i,j} = \begin{cases} 1, & u_{i,j} \geq \mu, \\ 0, & u_{i,j} < \mu, \end{cases}$$

\forall pixels $u \in$ image U of size $i \times j$, where μ is the mean pixel intensity in the image (Fig. 2.4).

The 16×16 image patches learned within the VQGAN codebook therefore correspond to local protein distributions. In Section 2.3, we detail how a weighted sum over these binarized image patches is used to determine a final probability density map.

Nucleus Image Codebook

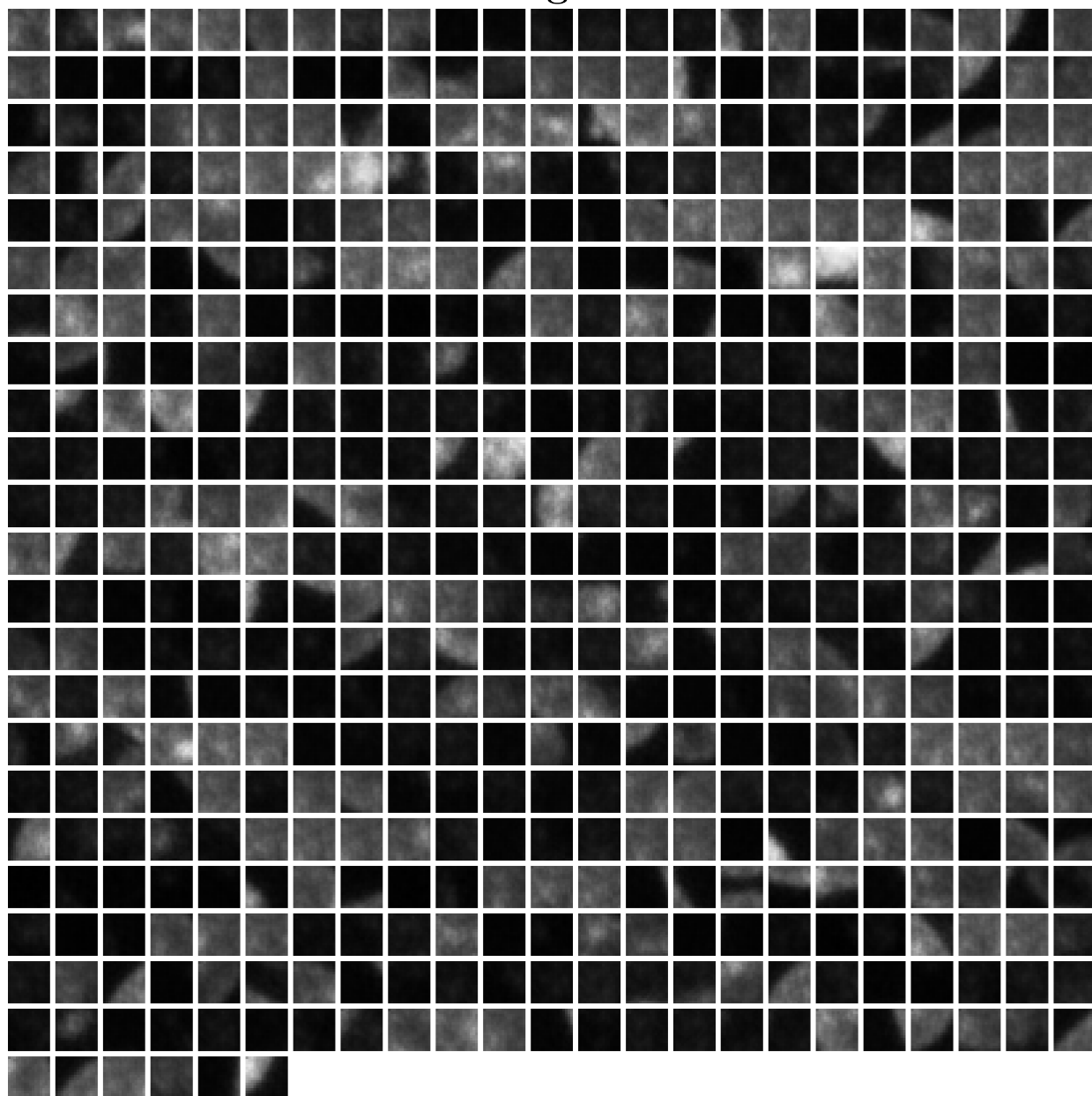


Figure 2.3: 512 image patches extracted from the nucleus reference VQGAN

Thresholding Process

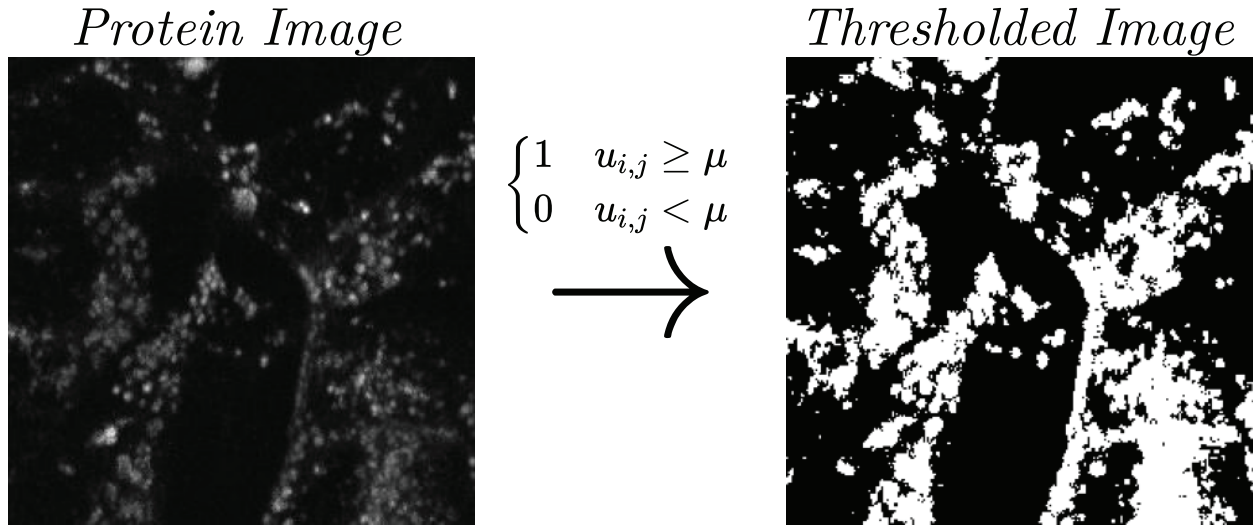


Figure 2.4: Example of thresholding process to convert protein image (left) to thresholded ground truth image (right) for CELL-E model

The model was trained on random 256×256 crops of 512×512 nuclei images. Adam Optimizer was used with learning rate set to 4.5×10^{-6} . The model was initially trained solely using mean-squared error reconstruction loss. After 50,000 steps, ~ 7 epochs, the discriminator loss term was introduced. This term helps with reducing the blurriness typically associated with VAEs. 512 discrete image codes were learned. Training occurred until the model reached convergence (at 371 epochs).

Protein Image Codebook

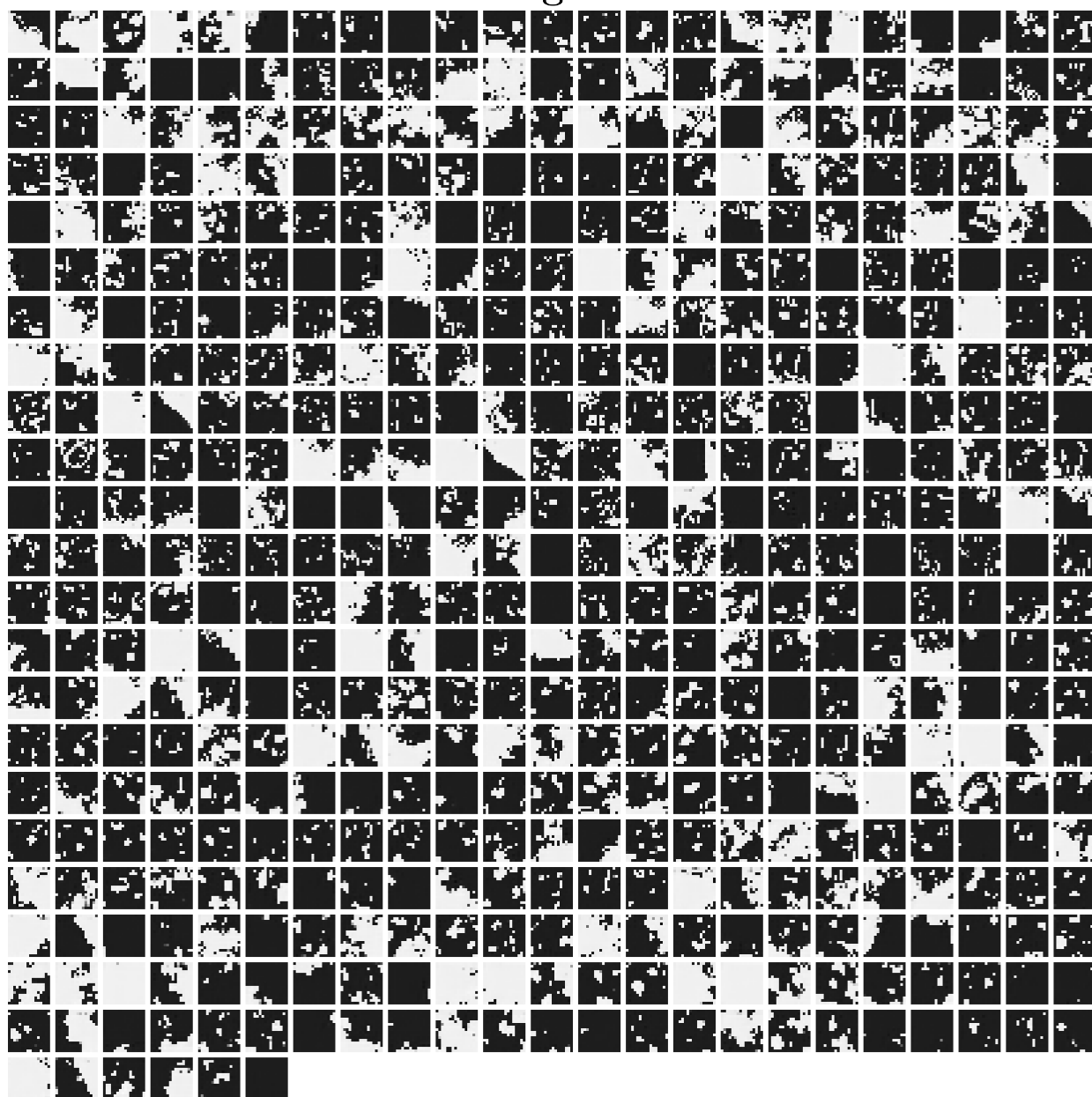


Figure 2.5: 512 image patches extracted from the protein threshold VQGAN

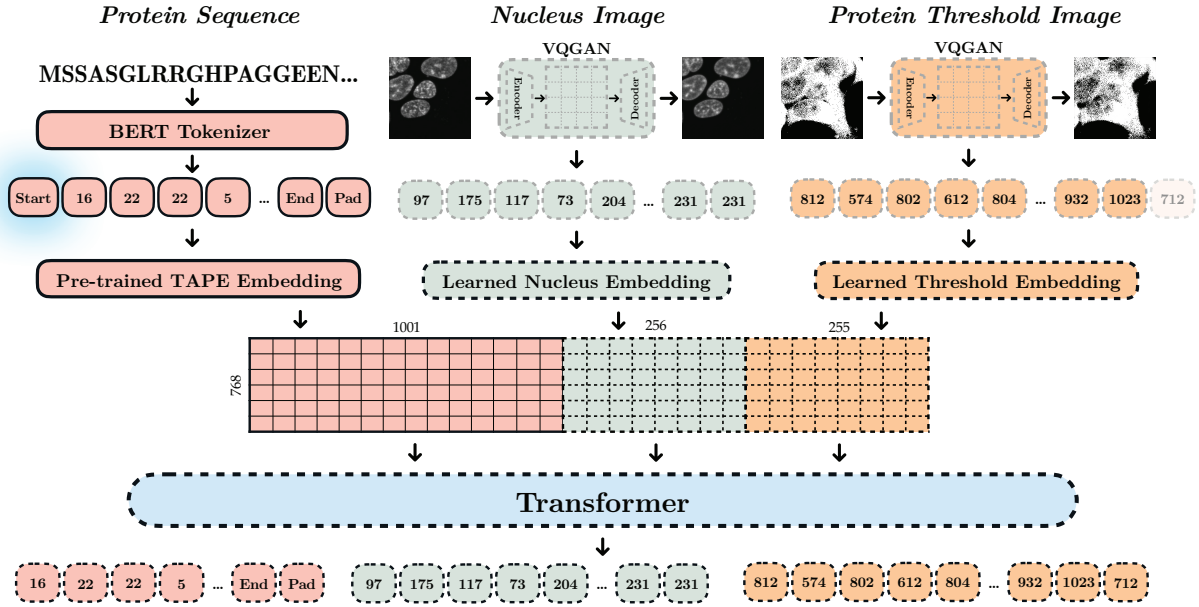


Figure 2.6: Graphical depiction of CELL-E. Solid lines correspond to pre-trained components. Gray dashed lines are learned in Phase 1 and 2 (Reference Image and Protein Threshold VQGANs). Black dashed lines correspond to components learned in Phase 3. A start token is prepended to the sequence and the final protein image token is removed. The amino acid sequence embedding from the model is preserved, and embedding spaces for the image tokens are cast in the same depth and concatenated with the amino acid sequence embedding. The transformer is tasked with reproducing the original sequence of tokens (e.g., the input sequence with start token shifted to the right one position).

CELL-E Base Transformer

The transformer (p_ϕ) utilizes an input comprised of amino acid tokens, a 256×256 nucleus image crop, and the 256×256 corresponding protein image threshold crop. In this phase, ϕ and θ are fixed, and a prior over all tokens is learned by maximizing ELBO with respect to ϕ . It is a decoder-only model [57].

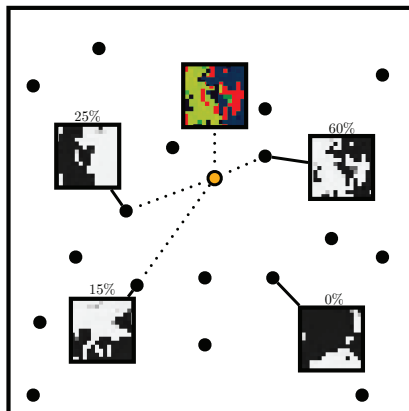


Figure 2.7: Simplified example of probability map calculation. Each circle corresponds to an image token within the quantized VQGAN embedding space. Each PDF patch (yellow) is obtained as a weighted sum over all protein threshold image VQGAN codebook vectors.

The model is trained on a concatenated sequence of text tokens, nucleus image tokens, and protein threshold image tokens, in order. Within the CELL-E transformer, image token embeddings were cast into the same dimensionality as the language model embedding to in order to maintain the larger protein context information, however the embeddings corresponding to the image tokens within this dimension are learned (See Fig. 2.6).

Amino acid sequences were converted to indices via the selected language tokenizers. Unless otherwise stated, all results in this work utilized the IUPAC tokens and TAPE language embeddings. CELL-E uses encodings from TAPE, There are 30 possible codebook values for amino acids within this model, with 25 corresponding to amino acids and 5 corresponding to special tokens (i.e. padding). amino acid sequence length was limited to 1000 amino acids, which is longer than 96% of sequences within the dataset. For amino acid se-

quences shorter than 1000 amino acids, an end token (if utilized by the language model) was appended, followed by padding tokens. For amino acid sequences longer than 1000 amino acids, we randomly cropped a 1000 length subsection. If the right end of the crop ended before the true end of the amino acid sequence, no end token was applied. A start token is then prepended to all 1000 length sequence. The TAPE model used represents input embeddings as vectors with dimension $n \times 768$, where n is the number of amino acids. The sequence embedding for the TAPE based models therefore had embedding vector sizes of 1001×768 . Input amino acids were tokenized and their embeddings were retrieved from the language models. This input embedding is fixed. We also explored other embeddings (UniRep, ESM1b, One-hot encoding, and chemical descriptors) in Table A.2.

The images were passed through the encoders of their respective VQGANs to obtain codebook tokens, and the final protein threshold image token is removed. We utilized data augmentation techniques including random cropping and random flips, just as was performed when training the VQGAN models. Within the CELL-E transformer, image token embeddings were cast into the same dimensionality as the language model embedding to in order to maintain the larger protein context information, however the embeddings corresponding to the image tokens within this dimension are learned. This ultimately creates a full sequence embeddings 1512×768 (1001×768 for text, 256×768 for nucleus images and 255×768 for protein threshold images) (Fig. 2.6). A rotary positional embedding [58] is then applied to the input embeddings.

We noted improved performance by shifting embeddings over by 1 (time-shifting [59]) in

the feature dimension, but only for image tokens. Image token embeddings were shifted one position from the top and one position from the left.

A full attention scheme [60] is used where future tokens are masked in order to retain full sequence and image context. The output of the attention layers is passed through a block consisting of a linear and softmax layer to produce logits for predicted tokens at each position. Selective masking is applied so the model is unable to select anything but amino acid tokens for amino acid positions, nucleus image tokens for the nucleus image positions, and protein image tokens for the protein image positions.

Cross-entropy loss is used to measure the model’s ability to reconstruct the original input vector (without the prepended start token and including the removed final protein threshold image token). The cross entropy is initially scaled by the length of the input, but further weighting is placed to emphasize the output protein image threshold tokens. We used weightings of $\frac{1}{9}$ for the amino acid tokens, $\frac{1}{9}$ for nucleus image tokens, and $\frac{8}{9}$ for the protein threshold image tokens.

The main CELL-E model had a depth of 32, indicating 32 consecutive attention and feed forward blocks, and 16 attention heads with dimension = 64. We used attention and feed forward attention dropout both = .1 during training. The language embedding was fixed. Model convergence occurred at 130 epochs and these weights were used for study.

Probability Density Maps

Within the training data, the corresponding protein image is merely a snapshot in time and could be markedly different if taken at another time point. For this reason, we do not believe a image prediction without confidence provides scientific utility. We utilize the learned distribution of sequence and image pairings to generate a continuous 2D probability density function of protein localization by having the model predict per-pixel binary probability representations of protein localization, which can then be linearly combined.

When generating images, the model is provided with the amino acid sequence and nucleus image. The transformer autoregressively predicts the protein-threshold image. In order to select a token, the model outputs logits which contain probability values corresponding to the codebook identity of the next token. The image patch v_i is selected by filtering for the top 25% of tokens and applying top-k sampling with gumbel noise [61].

Ordinarily, the final image is generated by converting the predicted codebook indices of the protein threshold image to the VQGANs decoder. However, to generate the probability density map \bar{v} , we include the full range of probability values corresponding to image patches, $p(v_i)$, obtained from the output logits. The values are clipped between 0 and 1 and multiplied by the embedding weights within the VQGAN’s decoder, w_i :

Theorem 3.

$$\bar{v} = w \cdot p(v) = \sum_{i=1}^n w_i p(v_i)$$

This output is normalized and displayed as a heatmap (Fig. 2.7).

Chapter 3

Results

3.1 Image Reconstruction

Fig. 3.2 and Fig. 3.3 show the CELL-E predictions for several proteins in the validation dataset. High similarities can be seen between the predictions and the ground truth. Even though the reference images only depict the nuclei, which is a limitation of the OpenCell training data, CELL-E can reasonably paint the shape of the cell for cytoplasmic proteins. Interestingly, the case of Mitogen-Activated Protein Kinase 9 (MAPK9) contains a cell in metaphase (top row of Fig. 3.2). CELL-E correctly predicts the round shape of its distribution around the mitotic chromosomes instead of the more expanded distribution for the adjacent interphase cell. This result suggests that CELL-E can indeed capture cell state information from the morphological reference images.

We used several metrics to evaluate the reconstruction performance of CELL-E, summarized in Table 3.1. Among the metrics, nucleus proportion accuracy measures how close the estimated proportion of pixel intensity within the nucleus is to the ground truth thresholded image. We believe this is the most relevant metric as it is not obscured by small spatial

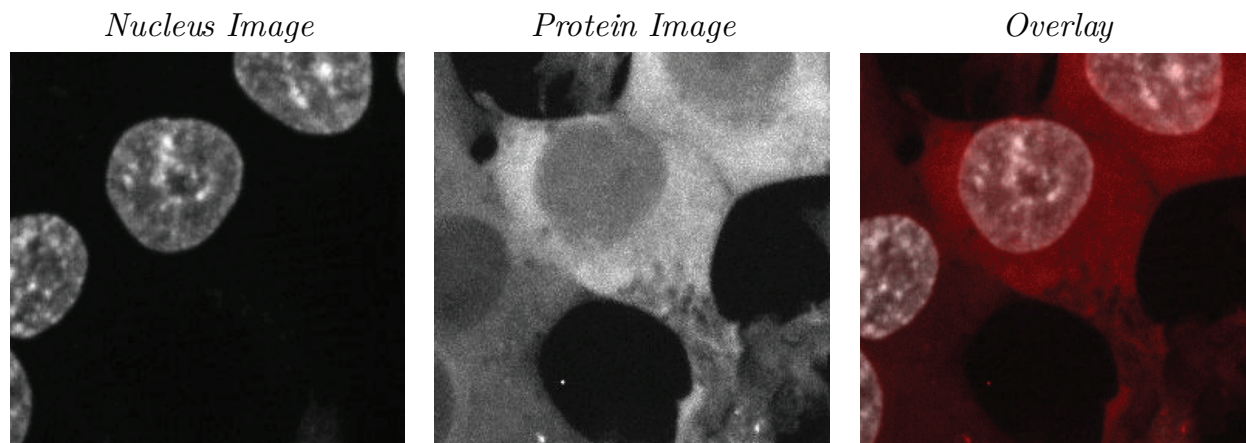


Figure 3.1: Nucleus Image (left), Protein Image (middle), and Overlay (Right). The alpha value for the protein channel in the right column is set to .7. Overlay is used as the “Original Image” in Fig. 3.2 and Fig. 3.3.

variations and nucleus boundaries can be obtained from the reference images. Description of other metrics and more information on the evaluation procedure can be found in Section 3.1. Using these metrics, we performed ablations studies to optimize our model architecture and choice of protein language embedding (see Section 4.3, Fig. 4.5 and Table A.2).

To assess performance, we generated a single prediction per image found in the OpenCell set. Each image was randomly cropped and flipped similar to training, but cropped regions and flips were maintained between models.

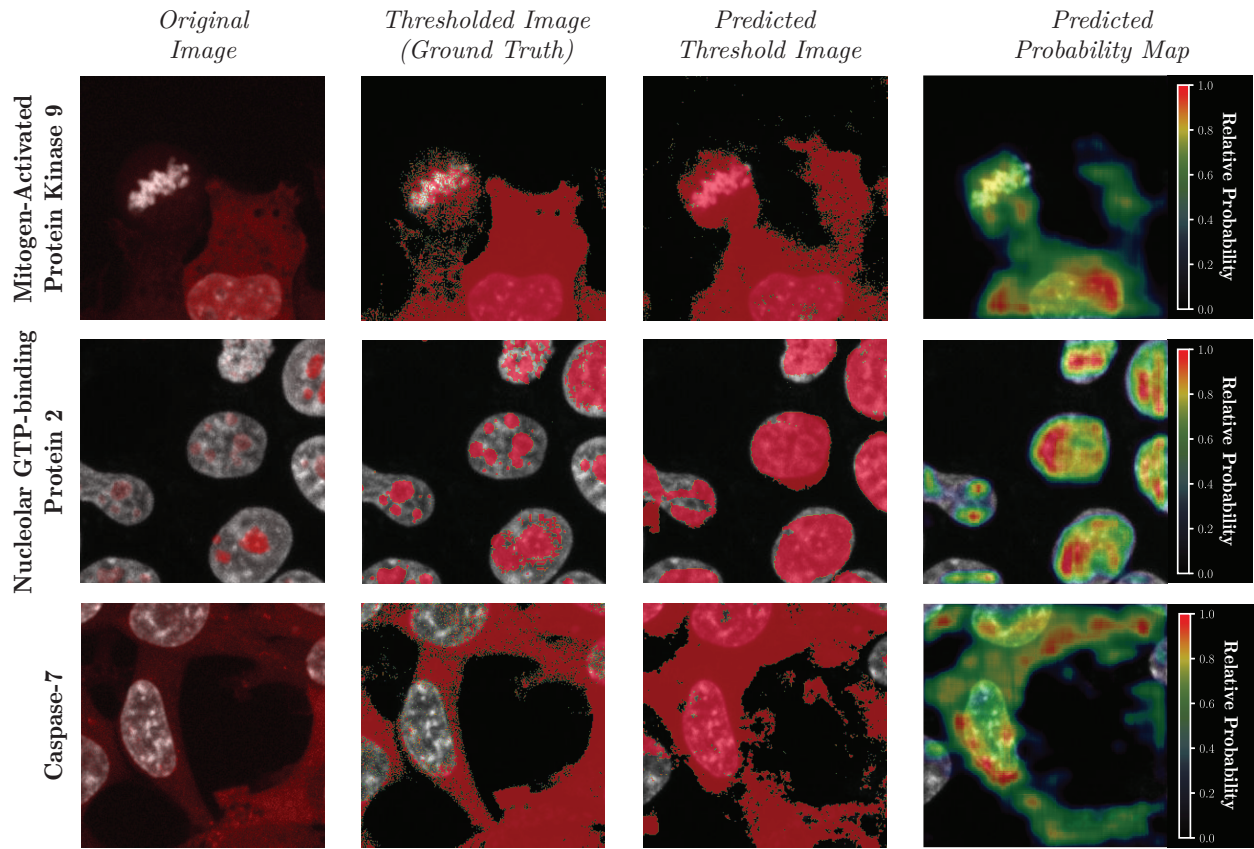


Figure 3.2: Prediction results of several types of proteins from the validation set, unseen to the model during training. The nucleus channel is depicted in grayscale, and the protein channel is shown as an overlay in red (Fig. 3.1 for clarification). The thresholded image (Column 2) is designated “Ground Truth” because those are the types of images exposed to the model during training. The predicted probability map is obtained from a weighted sum of potential image patches and normalized to 1.

Metrics

Nucleus Proportion Accuracy

To calculate the proportion of intensity in the nucleus, we first create a mask (Fig. 3.4 of the nucleus channel using Cellpose [62]. We take a sum over the predicted 2D PDF pixels

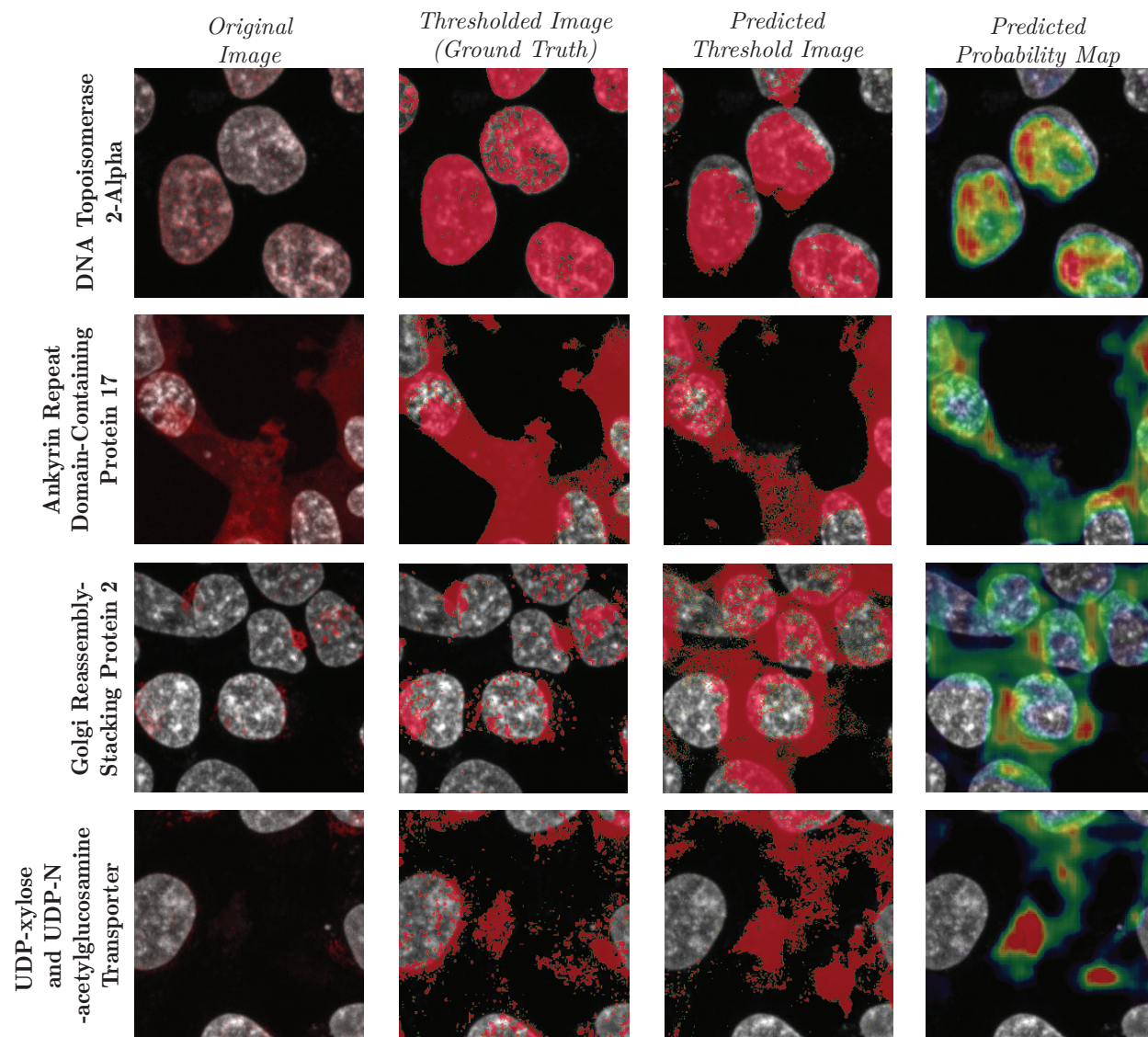


Figure 3.3: More prediction results from the validation set. We observe a high degree of spatial awareness from the model, notably in UDP-xylose-acetylglucosamine Transporter, which accurately predicts signal between cell nuclei with high confidence.

Table 3.1: Image Accuracy

| | Train | | Validation | |
|-------------------------------------|-----------------|-----------------|-----------------|-----------------|
| | CELL-E | VQGAN | CELL-E | VQGAN |
| Nucleus Proportion Accuracy | 0.94 ± 0.05 | 0.99 ± 0.01 | 0.81 ± 0.18 | 0.99 ± 0.01 |
| Predicted Threshold Pixel Accuracy | 0.77 ± 0.06 | 0.87 ± 0.08 | 0.77 ± 0.05 | 0.88 ± 0.08 |
| Predicted 2D PDF Pixel Accuracy | 0.68 ± 0.10 | | 0.63 ± 0.10 | |
| Structural Similarity Index Measure | 0.32 ± 0.21 | 0.55 ± 0.25 | 0.25 ± 0.16 | 0.56 ± 0.25 |
| Inception Score | 2.77 ± 0.07 | 4.17 ± 0.17 | 2.13 ± 0.07 | 3.87 ± 0.17 |
| Fréchet Inception Distance | 107 | 15 | 156 | 23 |

Performance is reported as mean \pm standard deviation where applicable. The VQGAN columns indicate the value of these metrics evaluated on the ground truth threshold image passed through the protein threshold VQGAN. As CELL-E selects tokens from this VQGAN to produce its outputs, these values represent the best possible performance for our model.

found within the nucleus mask, and divide this by the sum of pixels across the image.

For the the ground truth, we use a similar masking calculation, but consider the values of the ground truth protein image. These values are subtracted to calculate a mean-average error (MAE). Since the maximum possible value is 1 and minimum possible value is 0, we report accuracy as $1 - \text{MAE}$.

Predicted Threshold Pixel Accuracy

We simple calculate a pixel-wise MAE between the predicted protein threshold image of CELL-E and the ground truth protein threshold image.

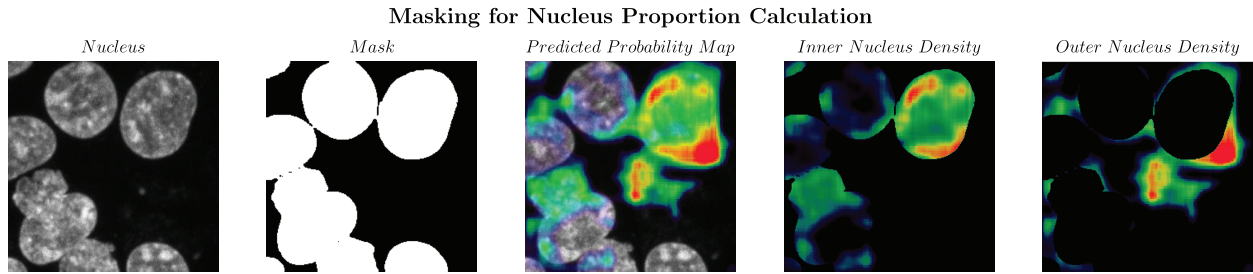


Figure 3.4: Masking procedure depicted.

Predicted 2D PDF Pixel Accuracy

This metric is similar to Predicted Threshold Pixel Accuracy, except we evaluate the difference using the predicted 2D PDF, rather than the predicted protein threshold image. We expect this number to be less accurate as tokens with less confidence will reduce the pixel value, while all values in the protein threshold image are 0 or 1.

SSIM

Structural similarity index measure (SSIM) is a measure of local perceptual similarity between images. It considers neighboring pixels to evaluate loss contextually by incorporating luminance and contrast information. SSIM values range between 0, indicating no similarity, and 1, indicating maximum similarity.

IS

Inception score (IS) is often used to evaluate the image outputs of GANs as a measure of “realisticness.” It rewards image variety and similarity to real-life data. Performance evaluation is based on the magnitude of the IS score.

FID

Fréchet Inception Distance (FID) is another popular metric for evaluating the quality of images from generative models. It compares the distributions between generated and ground truth images as the squared Wasserstein metric between two multidimensional Gaussian distributions. For this study FID was scored against the training or validation sets when applicable, rather than the entire OpenCell dataset.

3.2 1D Prediction Comparison

While CELL-E is not specifically trained as a discrete localization classifier, we also performed naive comparison between CELL-E model and 1D protein localization classifiers MuLoc [63] and Subcons [64] specifically trained with annotated protein localizations. We focused on nuclear classification using a simple classification criteria on CELL-E output (see Section 3.1), and the results are summarized in Table 3.2. We observed a relatively high degree of accuracy from this method compared to the task-specific models. CELL-E was a close second for validation set proteins despite not seeing localization annotations during

training.

Table 3.2: Nuclear Localization Prediction Accuracy

| | Train | Validation |
|---------|-----------------------------------|-----------------------------------|
| VQGAN | 0.99 ± 0.08 | 0.99 ± 0.09 |
| CELL-E | 0.89 ± 0.31 | 0.72 ± 0.45 |
| MuLoc | 0.71 ± 0.45 | 0.79 ± 0.41 |
| Subcons | 0.43 ± 0.49 | 0.69 ± 0.46 |

VQGAN indicates the accuracy evaluated on the ground truth threshold image passed through the VQGAN image encoder. As CELL-E selects tokens from this VQGAN to produce its outputs, these values represent the best possible performance for our model.

Nuclear Localization Prediction

The ground truth label for nuclear localization was designated by masking the nucleus, but computing the proportion of intensity on the ground truth thresholded protein image. If $> 50\%$ of this intensity was contained within the area of the nuclear mask, the assigned label would be positive for nuclear localization. Otherwise, the protein would be designated as non-nuclear. For the predicted label, we took a summation over the masked and unmasked regions of the predicted 2D PDF. If $> 50\%$ of pixel intensity for the 2D PDF was found in the nucleus, it was classified as a nuclear localizing protein. The protein localization prediction models were provided the amino acid sequence and were considered to predict nuclear if present in the localization prediction. These predictions were also compared against our naïve labels.

Overall we see CELL-E is a fairly strong performer in all categories. In Section 4.3, we performed ablation studies with respect to the nucleus proportion accuracy on the validation set to understand the performance improvement introduced from each model component. We noted that a completely sequence-blind model was able to predict nucleus proportion with accuracy = .7140. This is likely due to class imbalance in the OpenCell dataset, which contains $\sim 55\%$ nuclear proteins. We do still observe a $\sim 10\%$ increase prediction accuracy with the inclusion of sequence information.

Chapter 4

Discussion

4.1 Extra-domain Prediction

As a first test to show that CELL-E can recognize specific, functional sequence features, we let it predict the images for Green Fluorescent Protein (GFP), which is non-native to human and does not contain known localization signals, as well as GFP appended with two commonly used NLS's KRPAATKKAGQAKKKK from nucleoplasmin [65] and PAAKRVKLD from N-Myc [66]) that drive nuclear localization of a protein. We also appended a randomly generated sequence as a control. A randomly chosen nuclear image from the OpenCell dataset was used as the morphological reference. CELL-E does not localization of GFP (or random sequence + GFP) to a specific subcellular compartment with high confidence, whereas the two NLS-GFP fusions are clearly predicted to be localized within the nucleus (Fig. 4.1). Therefore, CELL-E has the potential to perform computational insertion screenings for the functional sufficiency of putative localization sequence features.

We visualize these results within a UMAP of joint CELL-E sequence-image encodings (Fig. 4.2) and observe a transition from a cytosolic region to a nuclear region upon appending

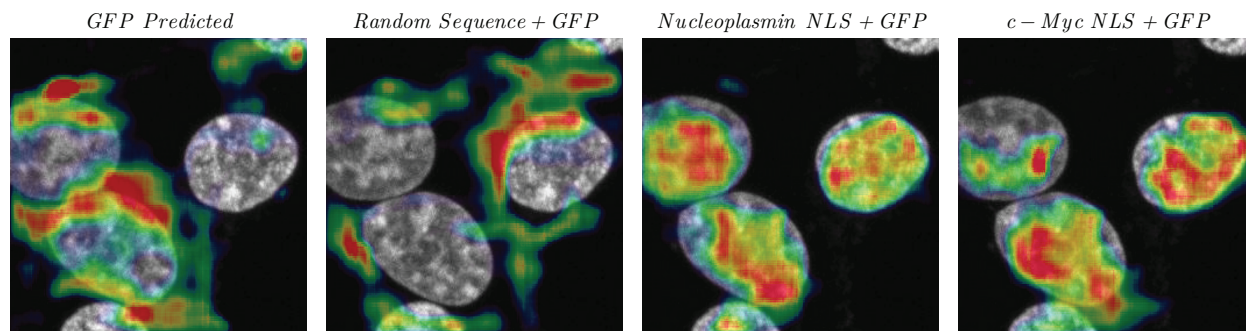


Figure 4.1: Predicted localization of GFP and modified-GFP sequences.

the NLSs.

CELL-E’s performance seems to be currently limited by the scope of the OpenCell dataset, which only accounts for a handful of proteins within a single cell type and imaging modality. As the OpenCell project is an active development, we expect stronger performance as more data become available. The availability of brightfield (e.g., phase-contrast) images as the morphological reference will also likely improve the prediction of cytoplasmic protein localization compared to using nuclei images. Furthermore, the utility of the model comes in terms of linking embedding spaces of dependent data. One could imagine follow up experiments where rather than images being the prediction, other signatures such as protein mass spec could be predicted. Additionally, other sources of information, such as structural embeddings could be incorporated to bolster CELL-E’s capabilities.

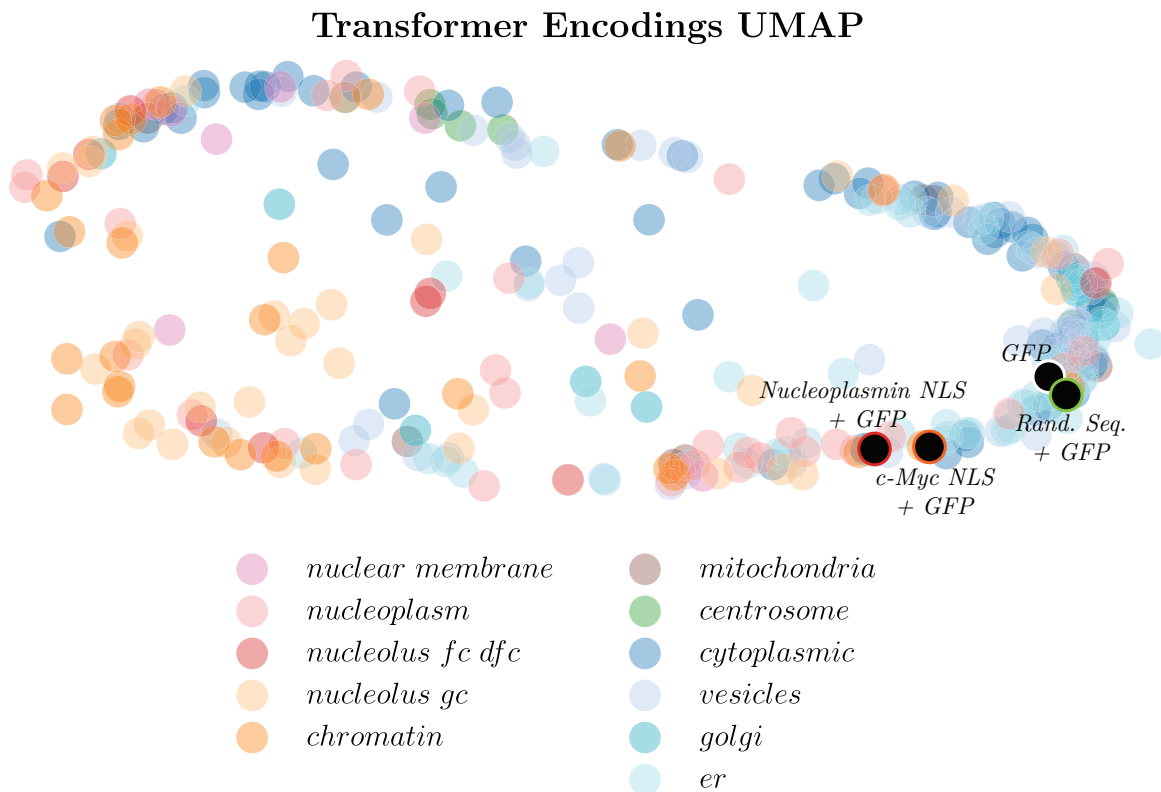


Figure 4.2: UMAP of encodings. We observe a largely nuclear protein “neighborhood” (left) and a cytoplasmic neighborhood (right). Prepending the NLS to GFP causes a shift towards the nuclear protein neighborhood.

4.2 Reproducing Experimental Results *in silico*

Next, we examined whether CELL-E can identify NLS in a protein by computationally performing truncation/deletion studies. For this purpose, we chose DNA Topoisomerase I (TOP1), whose N-terminal intrinsically disordered region (amino acid (aa) 1-199) is essential for its nuclear localization [67]. An experimental study generated a series of deletion mutants

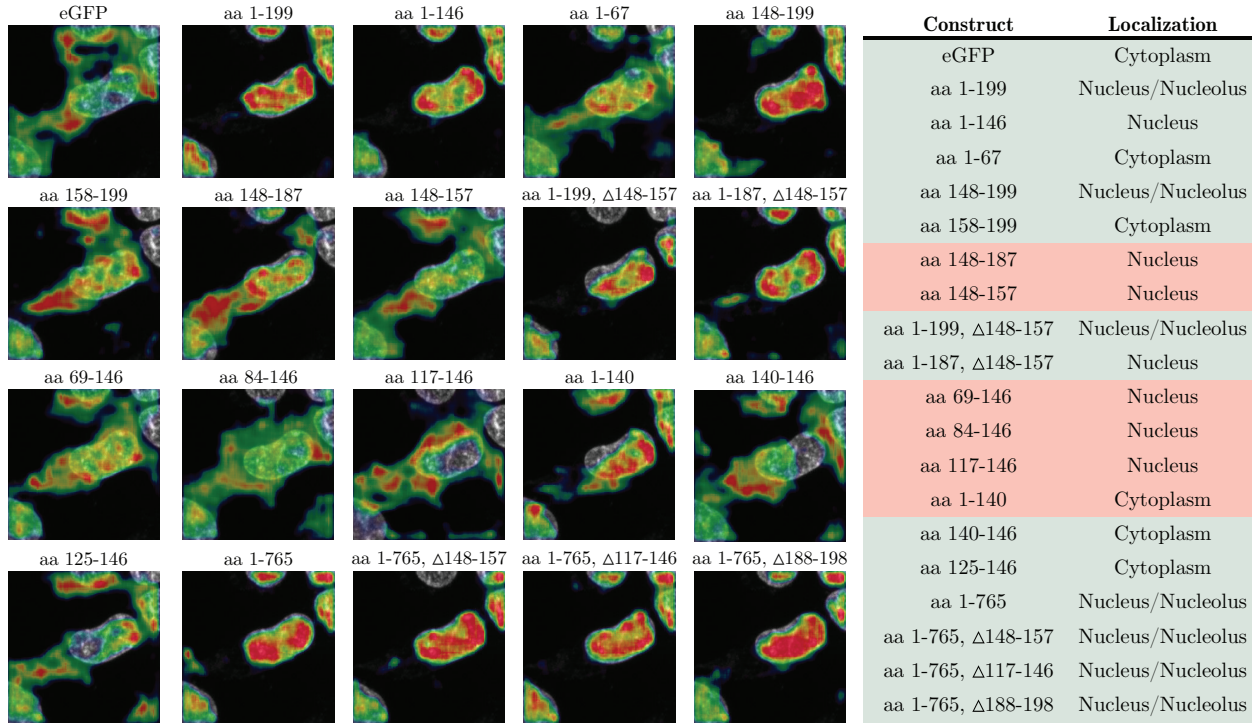


Figure 4.3: CELL-E’s predicted localization (images) of eGFP fusions from [2] and corresponding localization annotations (table) from the original paper. In the table on the right hand side, green indicates agreement between CELL-E and experimental results, while red indicates disagreement. *aa 1-199* contains the entire N-terminus region. *aa 1-146* only contains Motifs I and V. *aa 1-67* only contains Motif-I. *aa 148-199* contains Motif II, III, IV and V.

for this region and imaged the subcellular localization in HeLa cells when fused to eGFP [2]. To computationally reproduce this study, we fed the exact sequences of the deletion mutants to CELL-E. As shown in Fig. 4.3, the predictions were largely consistent with the experimental data, recapturing the inability for *aa 1-67* to drive nuclear localization despite containing a putative NLS, as well as the sufficiency of *aa 148-199* as an NLS.

Lastly, we demonstrate a more direct approach than computational insertion or deletion studies to identify putative sequence features responsible for protein localization. Specifically, we split the generated image patches into two groups, one with the target protein being present and the other being absent based on the average pixel intensity within the 16×16 image patch. Then, we calculated the difference of attention weights for each amino acid token to contribute to the two groups. Fig. 4.4 highlights the amino acids with higher weights for the “present” group. The highlighted amino acids include the three putative NLSs (Motifs II, III, and IV) in the experimentally verified *aa 148-199* range, as well as part of the new *aa 117-146* NLS identified in [2]. On the other hand, the putative NLS (Motif I) in the experimentally invalidated *aa 1-69* range are not activated. The attention map also suggest that *aa 89-107* (KIKKE) could be another NLS in this protein. We must point out that the calculation of attention map was simply based on a protein being “present” or “not present” in image patches and did not specify “nuclear localization” at all. Therefore, it should be capable of serving as a general approach to discover putative sequence features driving protein localization to a variety of subcellular compartments.

To obtain Fig. 4.4, we first split the 16×16 generated threshold image patches into 2 groups, one where protein is primarily determined to be present $\bar{u}_{i,j} > .75$ and another where background tokens are primarily selected $\bar{u}_{i,j} < .25$. For each respective group, we calculate a median of the attention matrices and used attention rollout [68] to recursively multiply across 32 layers. The final layers of both groups are then compared. We initially look at tokens with higher weightings for the present protein group, and discard the rest.

DNA Topoisomerase I Significant Tokens for Nuclear Localization

Sequence

N-Terminus

aa 1-67 Motif I
 MSGDHLHND~~SOIEAD~~FRLNDSHKHKDKHKDREHRHKEHKKEKDREKSKHSNSEHKDSE KHHKEKEKT

aa 68-133 Motif V
 KHKDGSSEKHKDKHKDRDKE EK RK EEKV RAS GDA KIK KE KENG FSS PPOI KDE EDD G Y F V P K E D I K

aa 134-199 Motif II Motif III Motif IV
PLKRPRDEDDADYKPKKIKTEDTKKEKKRKLEEEEDGKLKKPKNKDKDKKVPEPDNKKKKPKKEEE

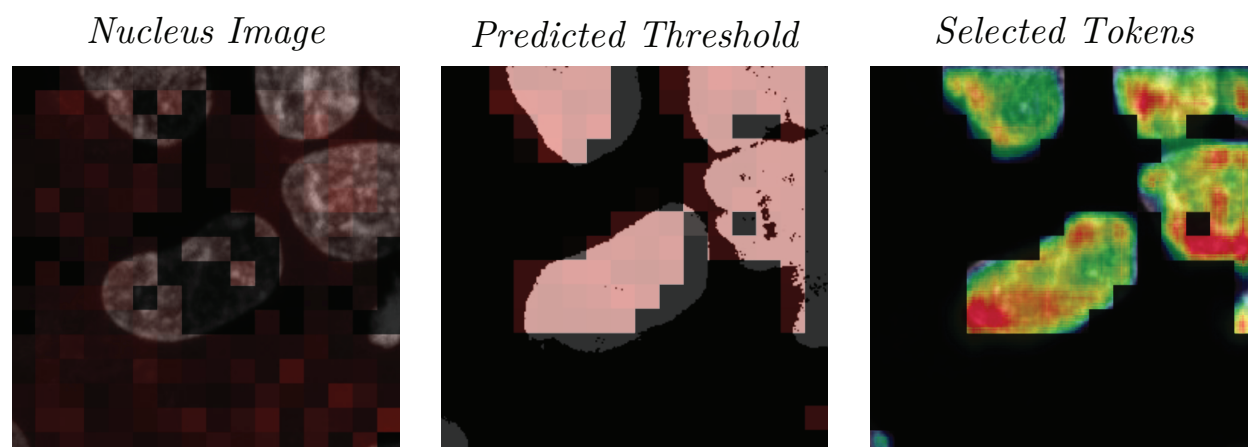


Figure 4.4: Attention weights for significant tokens when patches containing a large percentage of protein are selected (bottom-right figure). Previous computationally identified putative NLSs are boxed in black (top figure). These are *aa 59-65* (Motif I, KKHKEKE), *aa 150-156* (Motif II, KKIKTED), *aa 174-180* (Motif III, KKPKNKD), and *aa 192-198* (Motif IV, KKKPKKE). Additionally, the new NLS identified in Mo et. al.[2], Motif V (*aa 117-146*), is highlighted.

bo.huanf

4.3 Ablation Studies

Model Hyperparameters

In order to gauge the importance of each component of the model, we performed ablation by training several versions of the model with the same initialization. We specifically chose to look at performance in nuclear proportion accuracy over the validation set (Fig. 4.5).

On the top row, performance from the VQGAN is used as a reference of possible top performance, just as in Table 3.1. The second row depicts our main CELL-E (with depth = 32 and a fixed language embedding) used for this study. We found the performance in both cross-entropy and nuclear proportion accuracy increased with model depth when compared to similar models of smaller depth (third, fourth, and fifth rows).

To understand the effect of using fixed language models, we trained 2 versions of CELL-E of same depth. The first (second row, light blue) had a fixed language embedding, while the second (seventh row, red) was free to change during training. We also introduced a model with a randomly initialized language embedding (sixth row, light green). While we note fairly high performance from the unfixed models on the training data, they performed quite poorly on the validation, indicating severe overfitting. This is a result of the comparatively small number of proteins represented within the OpenCell dataset when compared to the large pFam database used to train TAPE.

We also trained a versions of the model which did not use a nucleus input (second to last row, pink) like DALL-E, and a model that only used a nucleus input and no sequence (last

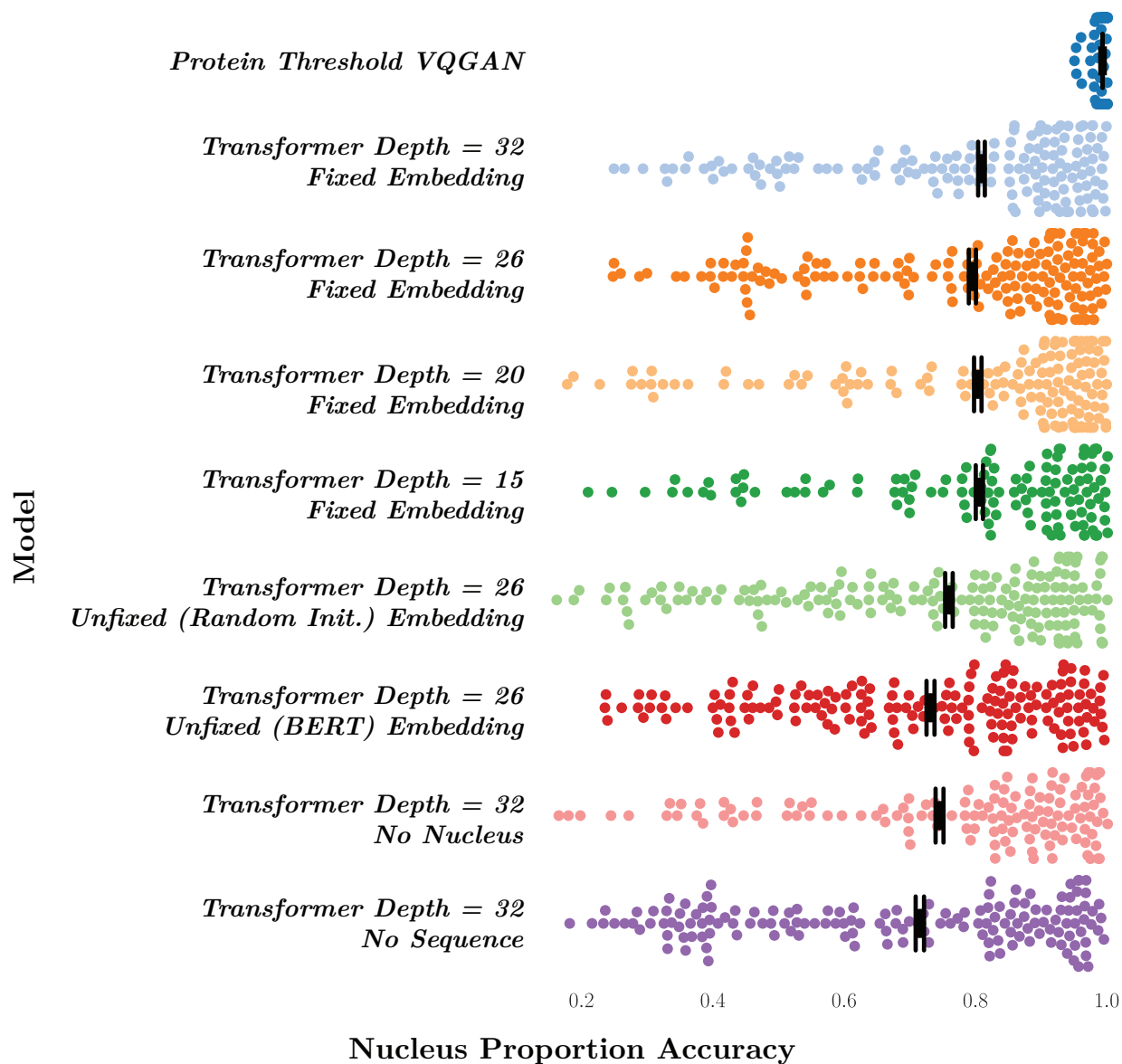


Figure 4.5: Ablation Plot. “Fixed” and “Unfixed” embedding refer exclusively to the amino acid embeddings. Image embeddings are always unfixed. Mean values and standard deviation are marked in black. ~ 150 points are randomly selected for display out of 1303 total predictions per model.

row, purple), although a start token was still prepended.

Overall, we observe a distributional shift to the right, indicating more accurate predictions, as the depth of the transformer is scaled. Full results for both training and validation sets can be seen in Table A.2. We also evaluated the performance of CELL-E using different protein embedding spaces. These were configured such that they were either at the same depth as the TAPE model, or the depth was scaled as deep as possible such that the GPU memory was saturated during training. All models were trained until convergence on the validation set.

Language Embedding

Alongside language embeddings, we also used one-hot and amino acid chemical descriptors as embedding features. The amino acid chemical descriptors come from Osorio et. al. [69], which contains amino numerical descriptions of amino acid properties from various literature sources [70–79]. Using one-hot and amino acid descriptors allowed us to scale to deeper model depths, but we did not see much improvement from doing so using these embeddings. These encodings likely do not contain sufficient information complexity about local environments that TAPE, UniRep, and ESM1b contain. While we do not see a consistent top performer on the training data, TAPE-based models generally performed the best across all metrics on the validation set, indicating a higher degree of generalizability.

Train-Validation Split Sequence Diversity

In machine learning applications which utilize amino acid sequence, it is recommended to cluster proteins based on similarity in order to create a distributional shift between a training and validation (and/or test) set. Oftentimes, redundancy in subsequences between both sets may result in memorization of training sequences and inflated performance metrics [80].

To investigate the effect of this on CELL-E, we performed a clustered split using a procedure identical to the one used by [10] to create a standard dataset used in benchmarking protein localization prediction. This model relies on PSI-CD-HIT [81]. In short, we clustered proteins based on a value cutoff of a designated percentage of identity for which the alignment must cover 80% of shorter sequences. We retrained CELL-E with train/validation splits with clustered with varied threshold percentages of sequence identity, ranging from 15% to 95% for 130 epochs. Our random split used for the main CELL-E effectively represents clustering based on 100% identity.

We did not observe any patterns in cross-entropy loss during training of the main transformer model in response to different cutoff values for sequence identity.

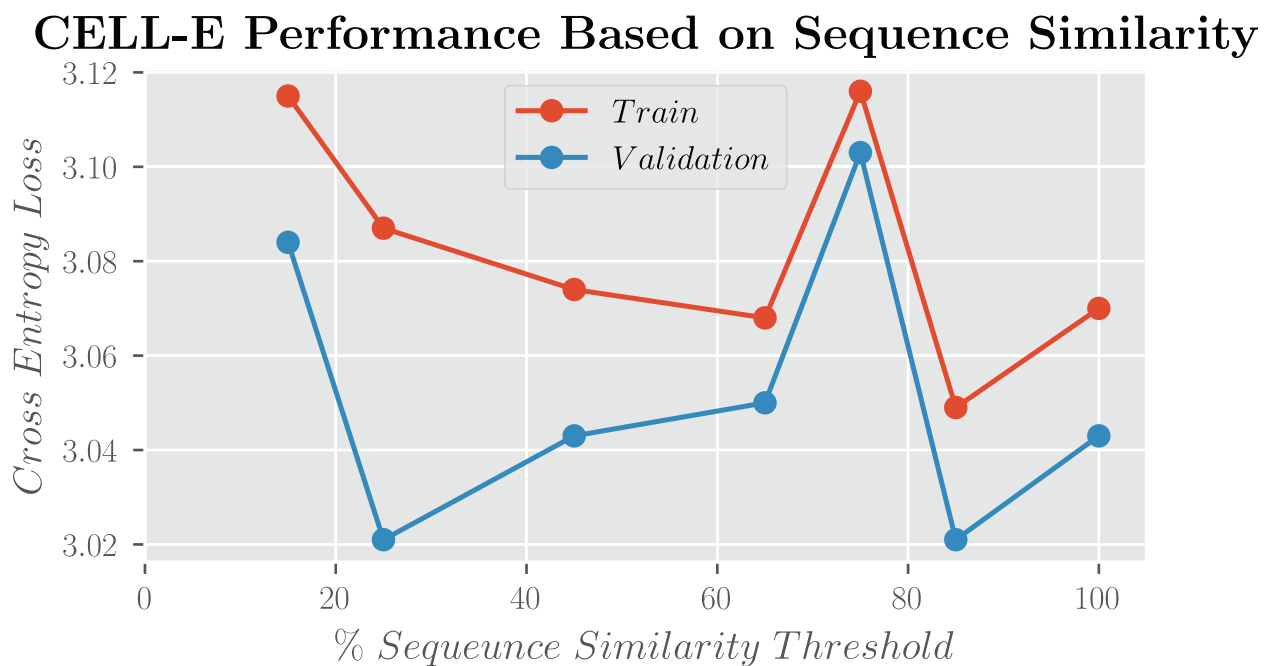


Figure 4.6: Cross-entropy loss, the metric used to evaluate performance, is shown after 130 epochs.

Bibliography

- [1] Thul, P. J. & Lindskog, C. The human protein atlas: A spatial map of the human proteome. *Protein Science: A Publication of the Protein Society* **27**, 233–244 (2018).
- [2] Mo, Y.-Y., Wang, C. & Beck, W. T. A Novel Nuclear Localization Signal in Human DNA Topoisomerase I*. *Journal of Biological Chemistry* **275**, 41107–41113 (2000). URL <https://www.sciencedirect.com/science/article/pii/S0021925819556435>.
- [3] Hu, T., Chitnis, N., Monos, D. & Dinh, A. Next-generation sequencing technologies: An overview. *Human Immunology* **82**, 801–811 (2021). URL <https://www.sciencedirect.com/science/article/pii/S0198885921000628>.
- [4] Palma, C.-A., Cecchini, M. & Samorì, P. Predicting self-assembly: from empirism to determinism. *Chemical Society Reviews* **41**, 3713–3730 (2012). URL <https://pubs.rsc.org/en/content/articlelanding/2012/cs/c2cs15302e>. Publisher: The Royal Society of Chemistry.
- [5] Chacinska, A., Koehler, C. M., Milenkovic, D., Lithgow, T. & Pfanner, N. Importing Mitochondrial Proteins: Machineries and Mechanisms. *Cell* **138**, 628–644 (2009). URL <https://www.sciencedirect.com/science/article/pii/S0092867409009672>.

- [6] Imai, K. & Nakai, K. Prediction of subcellular locations of proteins: where to proceed? *Proteomics* **10**, 3970–3983 (2010).
- [7] Ahmed, H. R. & Glasgow, J. Sokolova, M. & van Beek, P. (eds) *A Novel Particle Swarm-Based Approach for 3D Motif Matching and Protein Structure Classification*. (eds Sokolova, M. & van Beek, P.) *Advances in Artificial Intelligence*, Lecture Notes in Computer Science, 1–12 (Springer International Publishing, Cham, 2014).
- [8] Gardy, J. L. & Brinkman, F. S. L. Methods for predicting bacterial protein subcellular localization. *Nature Reviews Microbiology* **4**, 741–751 (2006). URL <https://www.nature.com/articles/nrmicro1494>. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 10 Primary_atype: Reviews Publisher: Nature Publishing Group.
- [9] Lu, J. *et al.* Types of nuclear localization signals and mechanisms of protein import into the nucleus. *Cell Communication and Signaling* **19**, 60 (2021). URL <https://doi.org/10.1186/s12964-021-00741-y>.
- [10] Almagro Armenteros, J. J., Sønderby, C. K., Sønderby, S. K., Nielsen, H. & Winther, O. DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics* **33**, 3387–3395 (2017). URL <https://doi.org/10.1093/bioinformatics/btx431>.
- [11] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic acids research* **45**, D158–D169 (2017). Place: England.

- [12] Pan, G., Sun, C., Liao, Z. & Tang, J. in *Machine and Deep Learning Deep learning (DL) for Prediction of Subcellular Localization* (ed.Cecconi, D.) *Proteomics Data Analysis Methods in Molecular Biology*, 249–261 (Springer US, New York, NY, 2021). URL https://doi.org/10.1007/978-1-0716-1641-3_5.
- [13] Zou, J. *et al.* A primer on deep learning in genomics. *Nature Genetics* **51**, 12–18 (2019). URL <https://www.nature.com/articles/s41588-018-0295-5>. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 1 Primary_atype: Reviews Publisher: Nature Publishing Group Subject_term: Computational biology and bioinformatics;Genetics;Genome informatics Subject_term_id: computational-biology-and-bioinformatics;genetics;genome-informatics.
- [14] Yun, K., Huyen, A. & Lu, T. Deep Neural Networks for Pattern Recognition. *arXiv:1809.09645 [cs]* (2018). URL <http://arxiv.org/abs/1809.09645>. ArXiv: 1809.09645.
- [15] Bepler, T. & Berger, B. Learning the protein language: Evolution, structure, and function. *Cell Systems* **12**, 654–669.e3 (2021). URL <https://linkinghub.elsevier.com/retrieve/pii/S2405471221002039>.
- [16] Rao, R. *et al.* Evaluating Protein Transfer Learning with TAPE. *arXiv:1906.08230 [cs, q-bio, stat]* (2019). URL <http://arxiv.org/abs/1906.08230>. ArXiv: 1906.08230.
- [17] Pang, L., Wang, J., Zhao, L., Wang, C. & Zhan, H. A Novel Protein Subcellular Local-

- ization Method With CNN-XGBoost Model for Alzheimer's Disease. *Frontiers in Genetics* **9**, 751 (2019). URL <https://www.frontiersin.org/article/10.3389/fgene.2018.00751>.
- [18] Yang, W.-Y., Lu, B.-L. & Yang, Y. *A Comparative Study on Feature Extraction from Protein Sequences for Subcellular Localization Prediction*, 1–8 (2006).
- [19] Hager, K. M., Striepen, B., Tilney, L. G. & Roos, D. S. The nuclear envelope serves as an intermediary between the ER and Golgi complex in the intracellular parasite *Toxoplasma gondii*. *Journal of Cell Science* **112** (Pt 16), 2631–2638 (1999).
- [20] Mim, C. & Unger, V. M. Membrane curvature and its generation by BAR proteins. *Trends in Biochemical Sciences* **37**, 526–533 (2012). URL <https://www.sciencedirect.com/science/article/pii/S0968000412001387>.
- [21] Ewing, G. W. pH is a Neurally Regulated Physiological System. Increased Acidity Alters Protein Conformation and Cell Morphology and is a Significant Factor in the Onset of Diabetes and Other Common Pathologies. *The Open Systems Biology Journal* **5** (2012). URL <https://benthamopen.com/ABSTRACT/TOSYSBJ-5-1>.
- [22] Martorana, A. *et al.* Probing Protein Conformation in Cells by EPR Distance Measurements using Gd³⁺ Spin Labeling. *Journal of the American Chemical Society* **136**, 13458–13465 (2014). URL <https://doi.org/10.1021/ja5079392>. Publisher: American Chemical Society.

- [23] Lou, H.-Y., Zhao, W., Zeng, Y. & Cui, B. The Role of Membrane Curvature in Nanoscale Topography-Induced Intracellular Signaling. *Accounts of Chemical Research* **51**, 1046–1053 (2018). URL <https://doi.org/10.1021/acs.accounts.7b00594>. Publisher: American Chemical Society.
- [24] Ohno, M., Karagiannis, P. & Taniguchi, Y. Protein Expression Analyses at the Single Cell Level. *Molecules* **19**, 13932–13947 (2014). URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6270791/>.
- [25] Grün, D. Revealing dynamics of gene expression variability in cell state space. *Nature Methods* **17**, 45–49 (2020). URL <https://www.nature.com/articles/s41592-019-0632-3>. Number: 1 Publisher: Nature Publishing Group.
- [26] Kotliar, D. *et al.* Identifying gene expression programs of cell-type identity and cellular activity with single-cell RNA-Seq. *eLife* **8**, e43803 (2019). URL <https://doi.org/10.7554/eLife.43803>. Publisher: eLife Sciences Publications, Ltd.
- [27] Vig, J. *et al.* BERTology Meets Biology: Interpreting Attention in Protein Language Models (2021). URL <http://arxiv.org/abs/2006.15222>. ArXiv:2006.15222 [cs, q-bio] version: 3.
- [28] Zaheer, M. *et al.* Big Bird: Transformers for Longer Sequences (2021). URL <http://arxiv.org/abs/2007.14062>. ArXiv:2007.14062 [cs, stat] version: 2.

- [29] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118**, e2016239118 (2021). URL <https://www.pnas.org/doi/abs/10.1073/pnas.2016239118>. [_eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.2016239118](https://www.pnas.org/doi/pdf/10.1073/pnas.2016239118).
- [30] Elnaggar, A. *et al.* ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. *IEEE transactions on pattern analysis and machine intelligence* **44**, 7112–7127 (2022).
- [31] Wang, Y. *et al.* A High Efficient Biological Language Model for Predicting Protein–Protein Interactions. *Cells* **8**, 122 (2019). URL <https://www.mdpi.com/2073-4409/8/2/122>. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.
- [32] Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods* **16**, 603–606 (2019). URL <https://doi.org/10.1038/s41592-019-0437-4>.
- [33] Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B. & Wu, C. H. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015). URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4375400/>.

- [34] Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic Acids Research* **49**, D412–D419 (2021). URL <https://doi.org/10.1093/nar/gkaa913>.
- [35] Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Research* **28**, 235–242 (2000). URL <https://doi.org/10.1093/nar/28.1.235>.
- [36] Vaswani, A. *et al.* Guyon, I. *et al.* (eds) *Attention is All you Need*. (eds Guyon, I. *et al.*) *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017). URL <https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>.
- [37] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods* **16**, 1315–1322 (2019). URL <https://www.nature.com/articles/s41592-019-0598-1>. Number: 12 Publisher: Nature Publishing Group.
- [38] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]* (2019). URL <http://arxiv.org/abs/1810.04805>. ArXiv: 1810.04805.
- [39] Ramesh, A. *et al.* Zero-Shot Text-to-Image Generation. *arXiv:2102.12092 [cs]* (2021). URL <http://arxiv.org/abs/2102.12092>. ArXiv: 2102.12092.

- [40] Kingma, D. P. & Welling, M. Auto-Encoding Variational Bayes. *arXiv:1312.6114 [cs, stat]* (2014). URL <http://arxiv.org/abs/1312.6114>. ArXiv: 1312.6114.
- [41] Goodfellow, I. *et al.* Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. Q. (eds) *Generative Adversarial Nets*. (eds Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. & Weinberger, K. Q.) *Advances in Neural Information Processing Systems*, Vol. 27 (Curran Associates, Inc., 2014). URL https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.
- [42] Mansimov, E., Parisotto, E., Ba, J. L. & Salakhutdinov, R. Generating Images from Captions with Attention. *arXiv:1511.02793 [cs]* (2016). URL <http://arxiv.org/abs/1511.02793>. ArXiv: 1511.02793.
- [43] Reed, S. *et al.* Balcan, M. F. & Weinberger, K. Q. (eds) *Generative Adversarial Text to Image Synthesis*. (eds Balcan, M. F. & Weinberger, K. Q.) *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48 of *Proceedings of Machine Learning Research*, 1060–1069 (PMLR, New York, New York, USA, 2016). URL <https://proceedings.mlr.press/v48/reed16.html>.
- [44] Xu, T. *et al.* AttnGAN: Fine-Grained Text to Image Generation with Attentional Generative Adversarial Networks. *arXiv:1711.10485 [cs]* (2017). URL <http://arxiv.org/abs/1711.10485>. ArXiv: 1711.10485.

- [45] Tiessen, A., Pérez-Rodríguez, P. & Delaye-Arredondo, L. J. Mathematical modeling and comparison of protein size distribution in different plant, animal, fungal and microbial species reveals a negative correlation between protein size and protein number, thus providing insight into the evolution of proteomes. *BMC Research Notes* **5**, 1–23 (2012). URL <https://bmcresearchnotes.biomedcentral.com/articles/10.1186/1756-0500-5-85>. Number: 1 Publisher: BioMed Central.
- [46] Cho, N. H. *et al.* OpenCell: Endogenous tagging for the cartography of human cellular organization. *Science (New York, N.Y.)* **375**, eabi6983 (2022). Place: United States.
- [47] Thul, P. J. *et al.* A subcellular map of the human proteome. *Science* **356**, eaal3321 (2017). URL <https://www.science.org/doi/10.1126/science.aal3321>. Publisher: American Association for the Advancement of Science.
- [48] Schnell, U., Dijk, F., Sjollem, K. A. & Giepmans, B. N. G. Immunolabeling artifacts and the need for live-cell imaging. *Nature Methods* **9**, 152–158 (2012). URL <https://doi.org/10.1038/nmeth.1855>.
- [49] Wallis, J., Miller, T., Lerner, C. & Kleerup, E. Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging* **8**, 297–230 (1989). Conference Name: IEEE Transactions on Medical Imaging.
- [50] Ding, M. *et al.* CogView: Mastering Text-to-Image Generation via Transform-

- ers. *arXiv:2105.13290 [cs]* (2021). URL <http://arxiv.org/abs/2105.13290>. ArXiv: 2105.13290.
- [51] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C. & Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents (2022). URL <http://arxiv.org/abs/2204.06125>. ArXiv:2204.06125 [cs].
- [52] Kobayashi, H., Cheveralls, K. C., Leonetti, M. D. & Royer, L. A. Self-Supervised Deep Learning Encodes High-Resolution Features of Protein Subcellular Localization. preprint, Cell Biology (2021). URL <http://biorxiv.org/lookup/doi/10.1101/2021.03.29.437595>.
- [53] Esser, P., Rombach, R. & Ommer, B. Taming Transformers for High-Resolution Image Synthesis. *arXiv:2012.09841 [cs]* (2021). URL <http://arxiv.org/abs/2012.09841>. ArXiv: 2012.09841.
- [54] Rezende, D. J., Mohamed, S. & Wierstra, D. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*, 1278–1286 (PMLR, 2014). URL <https://proceedings.mlr.press/v32/rezende14.html>. ISSN: 1938-7228.
- [55] Popel, M. & Bojar, O. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics* **110**, 43–70 (2018). URL <http://content.sciendo.com/view/journals/pralin/110/1/article-p43.xml>.

- [56] Schuler, G. D. *et al.* A gene map of the human genome. *Science (New York, N.Y.)* **274**, 540–546 (1996).
- [57] Liu, P. J. *et al.* *Generating Wikipedia by Summarizing Long Sequences* (2023). URL <https://openreview.net/forum?id=Hyg0vbWC->.
- [58] Su, J., Lu, Y., Pan, S., Wen, B. & Liu, Y. RoFormer: Enhanced Transformer with Rotary Position Embedding. *arXiv:2104.09864 [cs]* (2021). URL <http://arxiv.org/abs/2104.09864>. ArXiv: 2104.09864.
- [59] Bo, P. Improve the Transformer self-attention mechanism with just a few lines of code (almost no increase in computation). URL https://zhuanlan-zhuh.com.translate.google/p/191393788?_xt_rsl=en_xtrtl=zh-CN_xtrhl=en_xtrpto=wapp.
- [60] Child, R., Gray, S., Radford, A. & Sutskever, I. Generating Long Sequences with Sparse Transformers (2019). URL <http://arxiv.org/abs/1904.10509>. ArXiv:1904.10509 [cs, stat].
- [61] Jang, E., Gu, S. & Poole, B. Categorical Reparameterization with Gumbel-Softmax. *arXiv:1611.01144 [cs, stat]* (2017). URL <http://arxiv.org/abs/1611.01144>. ArXiv: 1611.01144.
- [62] Stringer, C., Wang, T., Michaelos, M. & Pachitariu, M. Cellpose: a generalist algorithm for cellular segmentation. *Nature Methods* **18**, 100–106 (2021). URL

- <https://www.nature.com/articles/s41592-020-01018-x>. Number: 1 Publisher: Nature Publishing Group.
- [63] Jiang, Y., Wang, D., Wang, W. & Xu, D. Computational methods for protein localization prediction. *Computational and Structural Biotechnology Journal* **19**, 5834–5844 (2021). URL <https://www.sciencedirect.com/science/article/pii/S2001037021004451>.
- [64] Salvatore, M., Warholm, P., Shu, N., Basile, W. & Elofsson, A. SubCons: a new ensemble method for improved human subcellular localization predictions. *Bioinformatics* **33**, 2464–2470 (2017). URL <https://doi.org/10.1093/bioinformatics/btx219>.
- [65] Dingwall, C., Robbins, J., Dilworth, S. M., Roberts, B. & Richardson, W. D. The Nucleoplasmin Nuclear Location Sequence Is Larger and MoreComplex than That of SV40 Large T Antigen. *The Journal of Cell Biology* **107**, 9 (1988).
- [66] Ray, M., Tang, R., Jiang, Z. & Rotello, V. M. Quantitative Tracking of Protein Trafficking to the Nucleus Using Cytosolic Protein Delivery by Nanoparticle-Stabilized Nanocapsules. *Bioconjugate Chemistry* **26**, 1004–1007 (2015). URL <https://doi.org/10.1021/acs.bioconjchem.5b00141>. Publisher: American Chemical Society.
- [67] Alsner, J., Svejstrup, J. Q., Kjeldsen, E., Sørensen, B. S. & Westergaard, O. Identification of an N-terminal domain of eukaryotic DNA topoisomerase I dispensable for

- catalytic activity but essential for in vivo function. *The Journal of Biological Chemistry* **267**, 12408–12411 (1992).
- [68] Abnar, S. & Zuidema, W. *Quantifying Attention Flow in Transformers*, 4190–4197 (Association for Computational Linguistics, Online, 2020). URL <https://aclanthology.org/2020.acl-main.385>.
- [69] Osorio, D., Rondón-Villarreal, P. & Torres Sáez, R. Peptides: A Package for Data Mining of Antimicrobial Peptides. *The R Journal* **7**, 4–14 (2015).
- [70] Kidera, A., Konishi, Y., Oka, M., Ooi, T. & Scheraga, H. A. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *Journal of Protein Chemistry* **4**, 23–55 (1985). URL <https://doi.org/10.1007/BF01025492>.
- [71] Sandberg, M., Eriksson, L., Jonsson, J., Sjöström, M. & Wold, S. New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *Journal of medicinal chemistry* **41**, 2481–2491 (1998). URL <https://doi.org/10.1021/jm9700575>.
- [72] Cruciani, G. *et al.* Peptide studies by means of principal properties of amino acids derived from MIF descriptors. *Journal of Chemometrics* **18**, 146–155 (2004). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cem.856>. [_eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.856](https://onlinelibrary.wiley.com/doi/pdf/10.1002/cem.856).

- [73] Liang, G. & Li, Z. Factor Analysis Scale of Generalized Amino Acid Information as the Source of a New Set of Descriptors for Elucidating the Structure and Activity Relationships of Cationic Antimicrobial Peptides. *QSAR & Combinatorial Science* **26**, 754–763 (2007). URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/qsar.200630145>.
_eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/qsar.200630145>.
- [74] Tian, F., Zhou, P. & Li, Z. T-scale as a novel vector of topological descriptors for amino acids and its application in QSARs of peptides. *Journal of Molecular Structure* **830**, 106–115 (2007). URL <https://www.sciencedirect.com/science/article/pii/S0022286006006314>.
- [75] Mei, H., Liao, Z. H., Zhou, Y. & Li, S. Z. A new set of amino acid descriptors and its application in peptide QSARs. *Biopolymers* **80**, 775–786 (2005).
- [76] van Westen, G. J. *et al.* Benchmarking of protein descriptor sets in proteochemometric modeling (part 1): comparative study of 13 amino acid descriptor sets. *Journal of Cheminformatics* **5**, 41 (2013). URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3848949/>.
- [77] Yang, L. *et al.* ST-scale as a novel amino acid descriptor and its application in QSAM of peptides and analogues. *Amino Acids* **38**, 805–816 (2010).
- [78] Georgiev, A. G. Interpretable numerical descriptors of amino acid space. *Journal of*

Computational Biology: A Journal of Computational Molecular Cell Biology **16**, 703–723 (2009).

- [79] Zaliani, A. & Gancia, E. MS-WHIM Scores for Amino Acids: A New 3D-Description for Peptide QSAR and QSPR Studies. *J. Chem. Inf. Comput. Sci.* (1999).
- [80] Walsh, I., Pollastri, G. & Tosatto, S. C. E. Correct machine learning on protein sequences: a peer-reviewing perspective. *Briefings in Bioinformatics* **17**, 831–840 (2016). URL <https://doi.org/10.1093/bib/bbv082>.
- [81] Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012). URL <https://doi.org/10.1093/bioinformatics/bts565>.

Appendix A

Additional Tables

Table A.1: Depicted Results Localization Annotations

| Figure | Protein Name | OpenCell Annotation |
|----------|--|---|
| Fig. 3.2 | Mitogen-Activated Protein Kinase 9 | Cytoplasmic, Nucleoplasm |
| Fig. 3.2 | Ribosome Biogenesis Protein NSA 2 Homolog | Nucleolus - GC |
| Fig. 3.2 | Caspase-7 | Cytoplasmic, Nucleoplasm, Aggregates |
| Fig. 3.3 | DNA Topoisomerase 2-Alpha | Chromatin, Nucleoplasm, Nuclear punctae, Nuclear Membrane |
| Fig. 3.3 | Ankyrin Repeat Domain-Containing Protein 17 | Cytoplasmic, Nucleoplasm |
| Fig. 3.3 | Golgi Reassembly-Stacking Protein 2 | Golgi |
| Fig. 3.3 | UDP-xylose and UDP-N-acetylglucosamine Transporter | Vesicles, Golgi |

Table A.2: Full Results Table

| Train | | | | | | | | | | | | |
|--------------------------|-------------|---------------------|-----------------------|--------------------|-----------------------------|---------------------|------------------|----------------|----------------|----------------|----------|--|
| Sequence Embedding Model | Model Depth | Embedding Dimension | Fixed Sequence Epochs | Embedding? Trained | Nucleus Proportion Accuracy | Predicted Threshold | | Pixel Accuracy | SSIM | IS | FID | Nucleus Localization Prediction Accuracy |
| | | | | | | Pixel Accuracy | Predicted 2D PDF | | | | | |
| N/A (VQGAN) | | | | | | | | | | | | |
| TAPE | 32 | 768 | ✓ | 371 | .9927 ± .0084 | .8749 ± .0817 | .7678 ± .0647 | .6758 ± .1047 | .5523 ± .2528 | 4.174 ± .1716 | 14.818 | .9933 ± .0810 |
| TAPE | 26 | 768 | ✓ | 136 | .9475 ± .0465 | .7551 ± .0726 | .6849 ± .1051 | .6131 ± .0928 | .3146 ± .2195 | 2.9021 ± .0667 | 81.0061 | .9059 ± .2920 |
| TAPE | 26 | 768 | ✗ | 49 | .9260 ± .0656 | .7568 ± .0664 | .6131 ± .0928 | .6131 ± .0928 | .2266 ± .1485 | 2.3797 ± .0742 | 140.339 | .8730 ± .3330 |
| TAPE | 20 | 768 | ✓ | 63 | .9433 ± .0501 | .7647 ± .0719 | .6815 ± .1081 | .6815 ± .1081 | .3138 ± .2234 | 2.8539 ± .0474 | 98.5444 | .9004 ± .2994 |
| TAPE | 15 | 768 | ✓ | 84 | .9456 ± .0469 | .7485 ± .0704 | .6730 ± .1041 | .6730 ± .1041 | .2955 ± .2122 | 2.687 ± .0742 | 112.4738 | .9095 ± .2869 |
| TAPE (No Nuc.) | 32 | 768 | ✓ | 91 | .9115 ± .0762 | .7609 ± .0762 | .5743 ± .0887 | .5743 ± .0887 | .2092 ± .1609 | 2.2793 ± .3975 | 137.8209 | .8516 ± .3554 |
| ESM1b | 20 | 1280 | ✓ | 54 | .9403 ± .0523 | .7610 ± .0664 | .6633 ± .1027 | .6633 ± .1027 | .2855 ± .2057 | 2.6059 ± .1153 | 111.8584 | .8902 ± .3127 |
| UniRep | 15 | 1900 | ✓ | 8 | .6591 ± .1547 | .7429 ± .0632 | .6011 ± .0803 | .6011 ± .0803 | .2108 ± .1410 | 2.1317 ± .0432 | 193.1455 | .8651 ± .3417 |
| AA Descriptors | 58 | 66 | ✓ | 60 | .9439 ± .0528 | .7674 ± .0690 | .6815 ± .1083 | .6815 ± .1083 | .3220 ± .2261 | 2.8121 ± .0529 | 94.2488 | .9044 ± .2941 |
| AA Descriptors | 32 | 66 | ✓ | 66 | .9472 ± .0499 | .7622 ± .0792 | .6825 ± .1114 | .6825 ± .1114 | .3293 ± .2420 | 3.144 ± .1315 | 65.5666 | .9082 ± .2888 |
| One-Hot | 59 | 25 | ✓ | 100 | .9469 ± .0487 | .7524 ± .0681 | .6791 ± .1077 | .6791 ± .1077 | .3141 ± .2212 | 2.7269 ± .0917 | 102.8544 | .9008 ± .2907 |
| One-Hot | 32 | 25 | ✓ | 70 | .9478 ± .0476 | .7475 ± .0705 | .6769 ± .1073 | .6769 ± .1073 | .03041 ± .2287 | 2.7469 ± .0905 | 90.0424 | .9057 ± .2922 |
| Random Initialization | 26 | 768 | ✗ | 71 | .9348 ± .0567 | .7693 ± .0668 | .6006 ± .1093 | .6006 ± .1093 | .2943 ± .2144 | 2.7593 ± .0852 | 110.9421 | .8885 ± .3147 |
| No Sequence | 32 | 768 | ✓ | 51 | .9235 ± .0678 | .7578 ± .0749 | .6113 ± .0930 | .6113 ± .0930 | .2196 ± .1517 | 2.2836 ± .0658 | 159.0451 | .8628 ± .3440 |
| Validation | | | | | | | | | | | | |
| N/A (VQGAN) | | | | | | | | | | | | |
| TAPE | 32 | 768 | ✓ | 371 | 0.9921 ± .0091 | .8756 ± .0824 | .7653 ± .0520 | .6342 ± .0964 | .5567 ± .2540 | 3.8718 ± .1662 | 25.9567 | .9923 ± .0872 |
| TAPE | 26 | 768 | ✓ | 136 | .7943 ± .1992 | .7574 ± .0607 | .6377 ± .1026 | .6377 ± .1026 | .2536 ± .1629 | 2.1300 ± .0704 | 155.7741 | .7155 ± .4511 |
| TAPE | 26 | 768 | ✗ | 49 | .7300 ± .2229 | .7505 ± .0648 | .6092 ± .0918 | .6092 ± .0918 | .2446 ± .1689 | 2.2708 ± .1303 | 136.294 | .6979 ± .4594 |
| TAPE | 20 | 768 | ✓ | 63 | .8024 ± .2071 | .7739 ± .0613 | .6388 ± .1088 | .6388 ± .1088 | .2547 ± .1821 | 2.2641 ± .1379 | 156.3681 | .5905 ± .4917 |
| TAPE | 15 | 768 | ✓ | 84 | .8044 ± .1927 | .7422 ± .0554 | .6267 ± .0914 | .6267 ± .0914 | .2226 ± .1546 | 2.0341 ± .0848 | 170.5773 | .7247 ± .4467 |
| TAPE (No Nuc.) | 32 | 768 | ✓ | 91 | .7742 ± .2239 | .7733 ± .0660 | .5688 ± .0739 | .5688 ± .0739 | .1956 ± .1311 | 3.3204 ± .1737 | 45.6887 | .6288 ± .4833 |
| ESM1b | 20 | 1280 | ✓ | 54 | .8044 ± .1851 | .7463 ± .0602 | .6235 ± .0916 | .6235 ± .0916 | .2207 ± .1604 | 2.1925 ± .0868 | 146.5215 | .7224 ± .4478 |
| UniRep | 15 | 1900 | ✓ | 8 | .7474 ± .2063 | .7393 ± .0616 | .5982 ± .0789 | .5982 ± .0789 | .2009 ± .1332 | 2.1287 ± .1231 | 206.3325 | .6173 ± .4862 |
| AA Descriptors | 58 | 66 | ✓ | 60 | .7854 ± .2234 | .7642 ± .0602 | .6340 ± .1020 | .6340 ± .1020 | .2487 ± .1736 | 2.2561 ± .1030 | 144.271 | .7002 ± .4582 |
| AA Descriptors | 32 | 66 | ✓ | 66 | .7688 ± .2435 | .7676 ± .0761 | .6350 ± .1163 | .6350 ± .1163 | .2662 ± .2016 | 2.8155 ± .1382 | 88.4847 | .6817 ± .4658 |
| One-Hot | 59 | 25 | ✓ | 100 | .7688 ± .1989 | .7347 ± .0535 | .6207 ± .0904 | .6207 ± .0904 | .2209 ± .1484 | 2.1214 ± .1182 | 159.35 | .6457 ± .4783 |
| One-Hot | 32 | 25 | ✓ | 70 | .7714 ± .2094 | .7383 ± .0599 | .6255 ± .1032 | .6255 ± .1032 | .2252 ± .1703 | 2.2802 ± .1170 | 129.3922 | .6419 ± .4795 |
| Random Initialization | 26 | 768 | ✗ | 71 | .7587 ± .2169 | .7605 ± .0517 | .6204 ± .0948 | .6204 ± .0948 | .2287 ± .1543 | 2.083 ± .1373 | 167.6092 | .6250 ± .4841 |
| No Sequence | 32 | 768 | ✓ | 51 | .7140 ± .2559 | .7528 ± .0733 | .6054 ± .0889 | .6054 ± .0889 | .2032 ± .1387 | 2.2473 ± .1400 | 169.204 | .5721 ± .4948 |

Publishing Agreement

It is the policy of the University to encourage open access and broad distribution of all theses, dissertations, and manuscripts. The Graduate Division will facilitate the distribution of UCSF theses, dissertations, and manuscripts to the UCSF Library for open access and distribution. UCSF will make such theses, dissertations, and manuscripts accessible to the public and will take reasonable steps to preserve these works in perpetuity.

I hereby grant the non-exclusive, perpetual right to The Regents of the University of California to reproduce, publicly display, distribute, preserve, and publish copies of my thesis, dissertation, or manuscript in any form or media, now existing or later derived, including access online for teaching, research, and public service purposes.

DocuSigned by:

Emaad Khwaja

618EAF7D24774B2...

Author Signature

7/15/2023

Date