# CELL-E: A Text-To-Image Transformer for Protein Localization Prediction

Emaad Khwaja[1,2*], Yun S. Song[2,3,4] and Bo Huang[4,5,6*]

[1*]UC Berkeley - UCSF Joint Graduate Program in Bioengineering, CA, USA.
[2]Computer Science Division, UC Berkeley, Berkeley, 94720, CA, USA.
[3]Department of Statistics, UC Berkeley, Berkeley, 94720, CA, USA.
[4]Chan Zuckerberg Biohub - San Francisco, San Francisco, 94158, CA, USA.
[5]Department of Pharmaceutical Chemistry, UCSF, San Francisco, 94143, CA, USA.
[6]Department of Biochemistry and Biophysics, UCSF, San Francisco, 94143, CA, USA.

*Corresponding author(s). E-mail(s): emaad@berkeley.edu; bo.huang@ucsf.edu;
Contributing authors: yss@berkeley.edu;

**Abstract**

Accurately predicting cellular activities of proteins based on their primary amino acid sequences would greatly improve our understanding of the proteome. In this paper, we present CELL-E, a text-to-image transformer architecture that generates a 2D probability density map of protein distribution within cells. Given a amino acid sequence and a reference image for cell or nucleus morphology, CELL-E offers a more direct 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.

**Keywords:** text-to-image synthesis, transformers, single-cell imaging, generative models

# 1 Introduction

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 [1]. 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 [2].

In this study, our focus is on predicting subcellular localization of proteins from their amino acid sequences, which serves as the 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 interactions, often involving localization "signals" in the amino acid sequence [3–5]. In many cases, however, the exact mechanisms for sequence recognition and trafficking are not yet fully understood [6]. For example, there is ongoing debate about the mechanism behind the import of proteins via the nuclear localization sequence (NLS) [7]. 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.) which are available from databases such as UniProt [8]. 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 have limited applications [9]. 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 based transformer 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 morphology and location (Fig. 1). It relies on transfer learning via amino acid encodings from a pretrained 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 from mutations, and uncover sequence features correlated with the specification of subcellular protein localization.

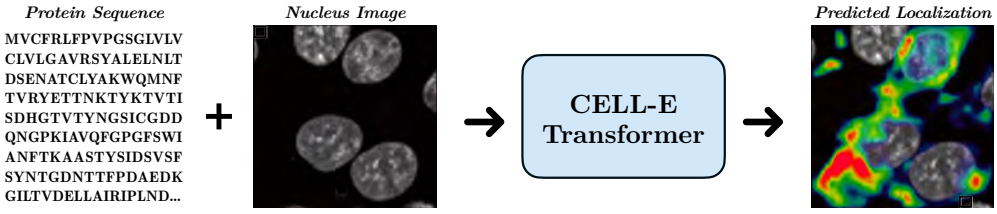



**Fig. 1** Given an input of amino acids and a reference nucleus image, CELL-E makes a prediction of protein localization with respect to the nucleus as a 2D probability density function, shown as heatmap, with color indicating relative confidence for each pixel.

# 2 Results

## 2.1 The CELL-E Model

CELL-E is inspired by the text-to-natural-image generation model of DALL-E [10] (See Section S.2.1) for a review of relevant work). Similar to DALL-E, our model autoregressively learns text and image tokens as a single stream of data. 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 [10–12]. Therefore, 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[13], which contains a library of 1,311 CRISPR-edited HEK293T human cell lines, each having one target 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 S.3.1 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 [10], whereas even the largest publicly available dataset with annotated protein images in human cells, Human Protein Atlas (HPA), only contains 12,003 unique proteins with just 82,000 images [14]. We found that utilizing transfer learning by incorporating a frozen embeddings from a language model pre-trained over thirty-one million protein domains from Pfam [15] as the input embedding for the amino acid text sequence. This approach reduces the number of learned paramaters, 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 S.1) 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 image tokens is capable of generating cell-like images from the amino acid sequence alone (Fig. S.3). However, quantifying

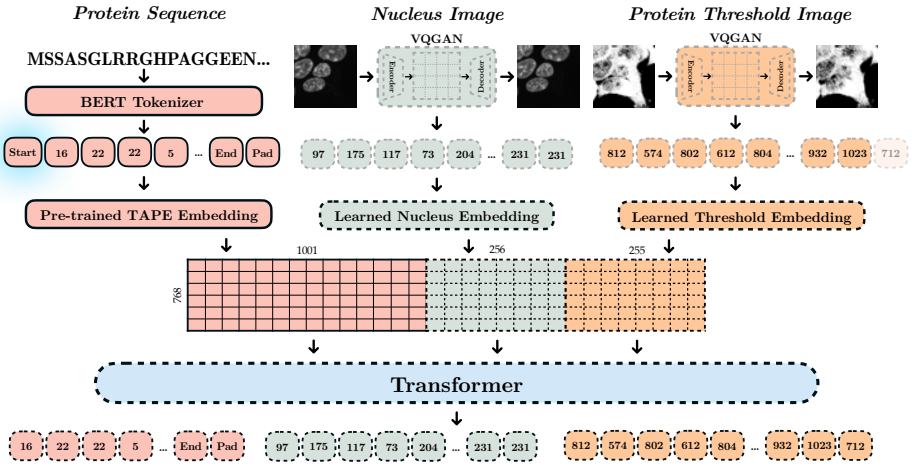



**Fig. 2** Graphical depiction of CELL-E. Solid lines correspond to pre-trained components. Gray dashed lines are learned in Phase 1 and 2 (Reference Image and Protein Threshold VQGANs). Black dashed lines correspond to components learned in Phase 3. A start token is prepended to the sequence and the 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).

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 utilize 3 separate embedding spaces that also include tokens representing the overall cell morphology from a reference image. The references 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 Varational Autoencoder (VQVAE) previously used to analyze OpenCell imaging data [17], we chose to use Vector Quantized Generative Adversarial Network (VQGAN) [18] 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). 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. 7). We note that the same model can also be trained to output gray-scale images directly (See Supplementary Notes S.2.2 and Fig. S.4).

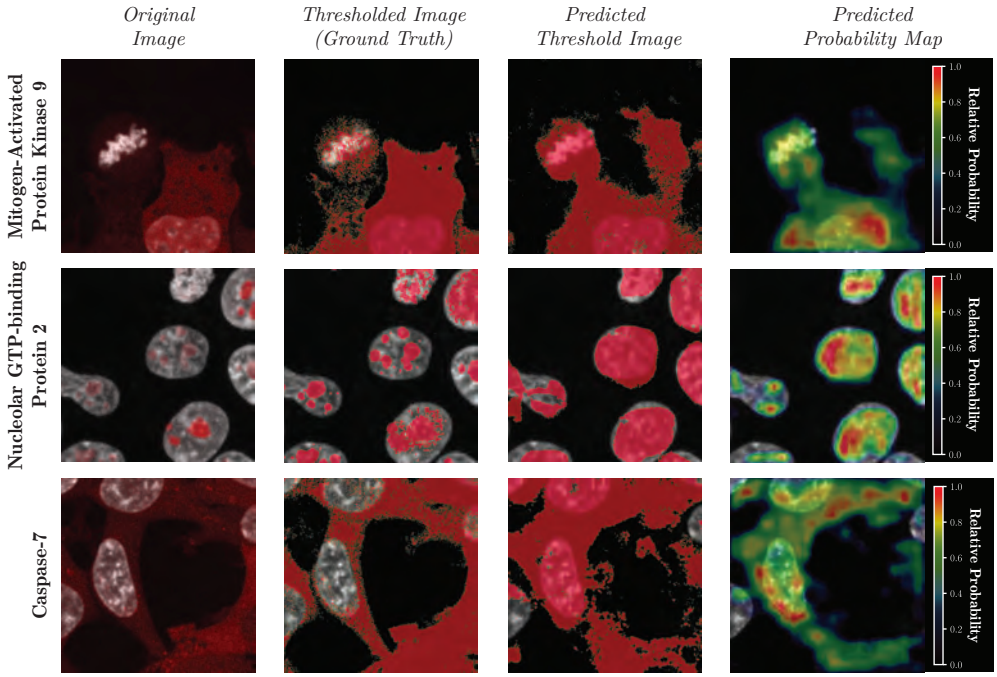



**Fig. 3** Prediction results of several types of proteins from the validation set, unseen to the model during training. The nucleus channel is depicted in grayscale, and the protein channel is shown as an overlay in red (Fig. S.1 for clarification). The thresholded image (Column 2) is designated "Ground 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.

## 2.2 Performance Evaluation

Fig. 3 and Fig. S.2 shows 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. 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 S.2. 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 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 S.3.7. Using these metrics, we performed ablations

**Table 1** Nuclear Localization Prediction Accuracy

| | Train | Validation |
|---|---|---|
| VQGAN | $0.99 \pm 0.08$ | $0.99 \pm 0.09$ |
| CELL-E | $\mathbf{0.89 \pm 0.31}$ | $0.72 \pm 0.45$ |
| MuLoc | $0.71 \pm 0.45$ | $\mathbf{0.79 \pm 0.41}$ |
| Subcons | $0.43 \pm 0.49$ | $0.69 \pm 0.46$ |

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.

studies to optimize our model architecture and choice of protein language embedding (see Section S.2.3, Fig. S.5 and Table S.1).

While not specifically trained as a discrete localization classifier, we also performed naive comparison between CELL-E model and 1D protein localization classifiers MuLoc [19] and Subcons [20] specifically trained with annotated protein localizations. We focused on nuclear classification using a simple classification criteria on CELL-E output (see Section S.3.7), and the results are summarized in Table 1. We observed a relatively high degree of accuracy from this method compared to the task-specific models. CELL-E outperformed on the training set, and it was a close second for validation set proteins despite not seeing localization annotations during training.

## 2.3 Analysis of NLS using CELL-E

As a first test to show 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, as well as GFP appended with two commonly used NLS's `KRPAATKKAGQAKKKK` from nucleoplasmin [21] and `PAAKRVKLD` from N-Myc [22]) 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 struggles to make any prediction with high confidence from the base GFP sequence or GFP appended with a short peptide containing a random sequence, whereas the two GFP-NLS fusions are clearly predicted to be localized within the nucleus. Therefore, CELL-E has the potential to perform computational insertion screenings for the functional sufficiency of putative localization sequence features.

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 [23]. An experimental study generated a series of deletion mutants for this region and imaged the subcellular localization in HeLa cells when fused to eGFP [24]. To computationally reproduce this study, we fed the exact sequences of the deletion mutants to CELL-E. As shown in Fig. 5, the

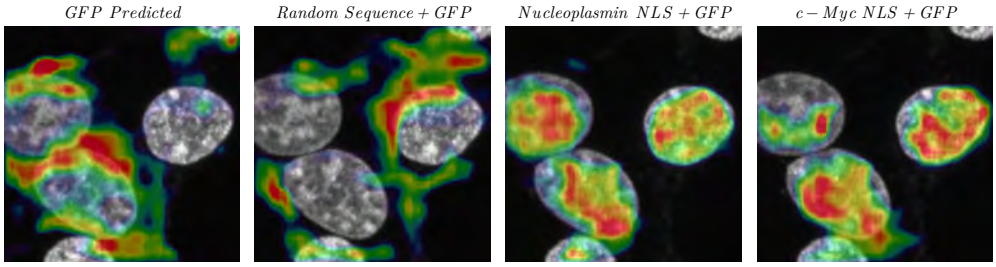
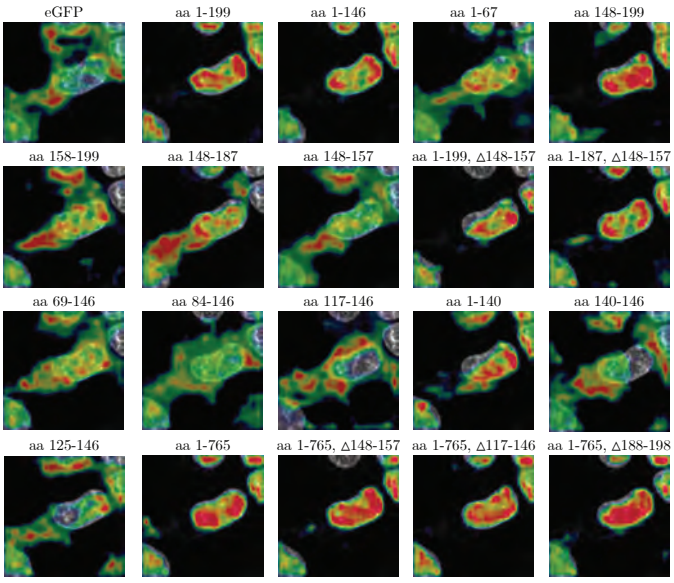



**Fig. 4** Predicted localization of GFP and modified-GFP sequences.

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.




| Construct | Localization |
|---|---|
| eGFP | Cytoplasm |
| aa 1-199 | Nucleus/Nucleolus |
| aa 1-146 | Nucleus |
| aa 1-67 | Cytoplasm |
| aa 148-199 | Nucleus/Nucleolus |
| aa 158-199 | Cytoplasm |
| aa 148-187 | Nucleus |
| aa 148-157 | Nucleus |
| aa 1-199, △148-157 | Nucleus/Nucleolus |
| aa 1-187, △148-157 | Nucleus |
| aa 69-146 | Nucleus |
| aa 84-146 | Nucleus |
| aa 117-146 | Nucleus |
| aa 1-140 | Cytoplasm |
| aa 140-146 | Cytoplasm |
| aa 125-146 | Cytoplasm |
| aa 1-765 | Nucleus/Nucleolus |
| aa 1-765, △148-157 | Nucleus/Nucleolus |
| aa 1-765, △117-146 | Nucleus/Nucleolus |
| aa 1-765, △188-198 | Nucleus/Nucleolus |

**Fig. 5** Predicted localization of eGFP fusions from [24] and corresponding figures from the original paper. *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.

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 $16 \times 16$ image patch. Then, we calculated the difference of attention weights for each amino acid token to contribute to the two groups. Fig. 6

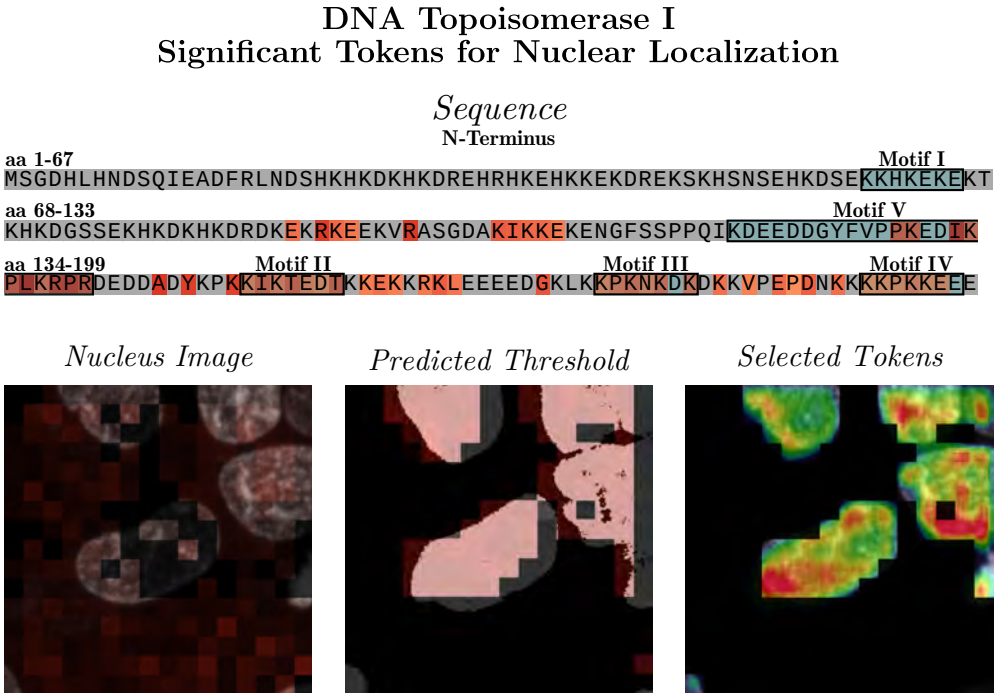



**Fig. 6** Attention weights for significant tokens when patches containing a large percentage of protein are selected (right column). Previously computationally identified putative NLSs are boxed in blue (left column). These are *aa 59-65* (Motif I, **KKHKEKE**), *aa 150–156* (Motif II, **KKIKTED**), *aa 174–180* (Motif III, **KKPKNKD**), and *aa 192–198* (Motif IV, **KKKPKKE**). Additionally, the new NLS identified in Mo et. al.[24], Motif V (*aa 117-146*), is highlighted.

highlights the amino acids with higher weights for the "present" group. The highlighted amino acids includes the three putative NLSs in the experimentally verified *aa 148-199* range, as well as part of the new *aa 117-146* NLS identified in [24]. On the other hand, the putative NLS 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 to serve as a general approach to discover putative sequence features driving protein localization to a variety of subcellular compartments.

## 3 Discussion

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 data becomes 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.

# 4 Methods

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) [18] is trained to represent a single channel 256×256nucleus image as a grid comprised of $16 \times 16$ image tokens (Fig. S.7), each of which could be one of 512 tokens.
- **Phase 2** A similar VQGAN is trained on images corresponding to binarized versions of protein images. These tokens represent the spatial distribution ofthe protein (Fig. S.9).
- **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.

## 4.1 Model Specifics

The optimization problem is modelled as maximizing the evidence lower bound (ELBO) [25, 26] 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.**

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

where $q_\phi$ is the distribution $16 \times 16$ image tokens from the VQGAN corresponding to the threshold protein image $u$, $p_\theta$ is the distribution over protein threshold generated by the VQGAN given the image tokens, and $p_\psi$ indicates the joint distribution over the amino acid, nucleus, and protein threshold tokens within the transformer.

## 4.2 Nucleus Image Encoder

Training both image VQGANs maximizes ELBO with respect to $\phi$ and $\theta$. The VQGAN improves upon existing quantized autoencoders by introducing a learned discriminator borrowed from GAN architectures [18]. The Nucleus Image Encoder is a VQGAN which represents $256 \times 256$ nucleus reference images as 256  $16 \times 16$ image patches. The VQGAN codebook size was set to $n = 512$ image patches. Further details can be found in Section S.3.4.

## 4.3 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 $\mu$ is the mean pixel intensity in the image (Fig. S.8).

The $16 \times 16$ image patches learned within the VQGAN codebook therefore correspond to local protein distributions. In Section 4.6, we detail how a weighted sum over these binarized image patches is used to determine a final probability density map. Hyperparameters and other training details can be found in Section S.3.5.

## 4.4 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 [27]. In practice, this creates a need for very large datasets [28]. The OpenCell dataset contains 1,311 proteins, while the human body is estimated to contain upwards of 80,000 unique proteins [29]. 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 pretrained language model exposed to a much larger dataset were utilized. We found the strongest performance came from TAPE embeddings [16]. Utilizing pretrained 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.

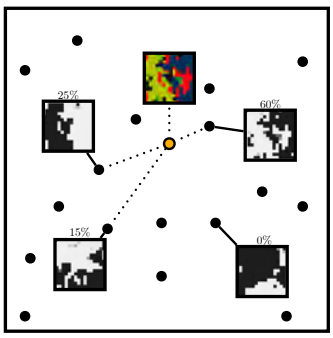



**Fig. 7** Simplified example of probability map calculation. Each circle corresponds to an image token within the quantized VQGAN embedding space. Each PDF patch (yellow) is obtained as a weighted sum over all protein threshold image VQGAN codebook vectors.

We tried training using random initialization for amino acid embeddings (See Section S.2.3), 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 [30] and ESM1-b [31].

## 4.5 CELL-E Transformer

The transformer ($p_\phi$) utilizes an input comprised of amino acid tokens, a $256 \times 256$ nucleus image crop, and the $256 \times 256$ corresponding protein image threshold crop. In this phase, $\phi$ and $\theta$ are fixed, and a prior over all tokens is learned by maximizing ELBO with respect to $\phi$. It is a decoder-only model [32].

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).

## 4.6 Probability Density Maps

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 [33].

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. 7).

# 5 Data and Code availability

Our model is a heavily modified version of an open source text-to-image transformer [34], available via the MIT license (Copyright (c) 2021 Phil Wang). Our code is available at https://github.com/BoHuangLab/Protein-Localization-Transformer via the MIT license (Copyright (c) 2022 Emaad Khwaja, Yun Song, & Bo Huang).

# 6 Acknowledgements

B.H. is supported by the National Institutes of Health (R01GM131641). Y.S.S. and B.H. are Chan Zuckerberg Biohub - San Francisco Investigators. Y.S.S. is supported by NIH grant R35-GM134922.

# 7 Author information

E.K. played a key role in the advancement of the approach, carrying out the majority of the coding, designing and conducting a significant number of the experiments, and producing an initial version of the manuscript. The remaining authors also offered consistent input on all aspects of the project, assessed the code, and helped with the final draft of the manuscript.

# Appendix S.1 Supplementary Figures

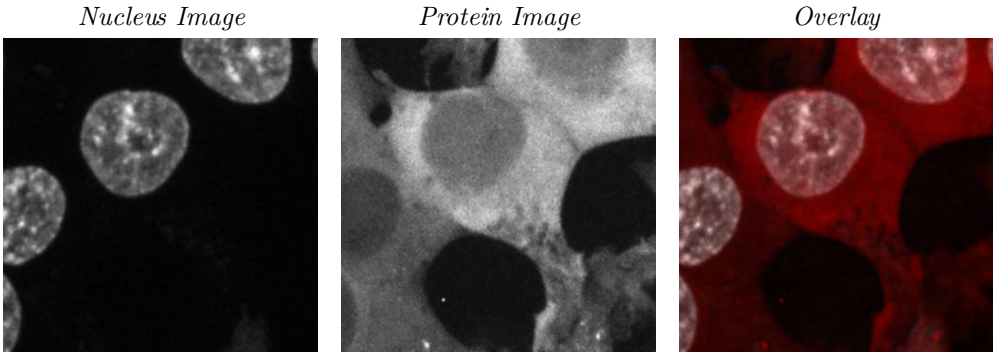



**Fig. S.1** Nucleus Image (left), Protein Image (middle), and Overlay (Right). The alpha value for the protein channel in the right column is set to .7. Overlay is used as the "Original Image" in Fig. 3 and Fig. S.2.

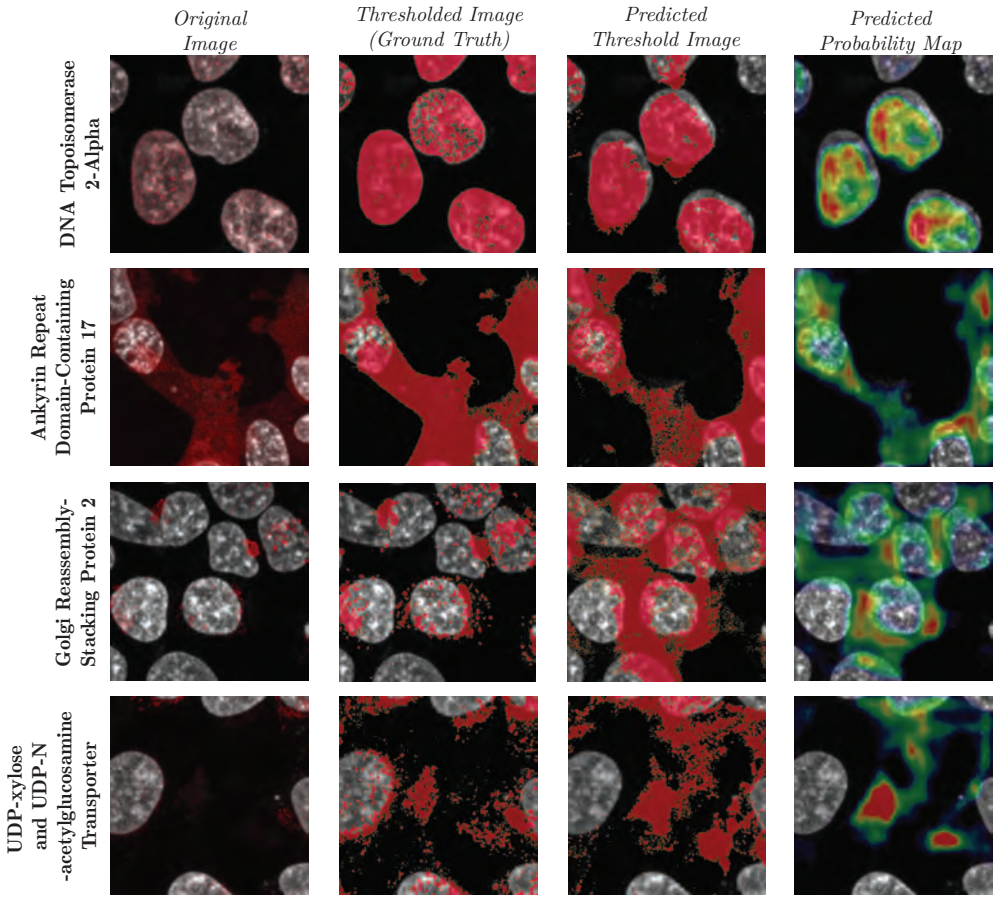



**Fig. S.2** More prediction results from the validation set. We observe a high degree of spatial awareness from the model, notably in UDP-xylose-acetylglucosamine Transporter, which accurately predicts signal between cell nuclei with high confidence.

# Appendix S.2 Supplementary Notes

## S.2.1 Related work

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 [35]. Self-supervised models from the NLP field have demonstrated excellent performance in predicting protein properties from amino acid sequence inputs [31, 36–38]. These models are trained on millions of amino acid sequences from databases such as BFD [39], UniRef [40], Pfam [15], and Protein Data Bank [41]. 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 [27]. UniRep is an

LSTM model that predicts the next amino acid in a variable length sequence [42], while BERT uses bidirectional masked language modeling to predict the identity of masked tokens throughout the sequence [43]. 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 [31].

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 [44]. 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 [45, 46].The success of language models in protein prediction tasks suggests that patterns dictating these structures are buried within residue sequences [16, 30].

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 [47, 48]. Discrete classifications for contiguous regions of the cell, such as the nuclear membrane, can be ambiguous and may have flawed annotations in established datasets. [49]. Additionally, these methods do not account for the influence of local cellular geometries [50–53] and cell states [54, 55] 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 [56]).

### S.2.2 Text-To-Image Generation

Ramesh et. al. [10] 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) [25] or Generative Adversarial Networks (GAN) [57] performed poorly when generating images outside of the training data, resulting in distorted images and artifacts [58–60].

While our method does similarly allow for truly zero-shot protein image prediction (Fig. S.3), 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, the model is tasked with predicting per-pixel binary probability representations of protein localization, which can then be linearly combined to generate a continuous 2D probability density function of protein localization.

The following images (Fig. S.3, Fig. S.4) are from text-to-image models architecturally similar to CELL-E, but replace the Protein Threshold VQGAN with a similarly trained Protein Image VQGAN.

Fig. S.4 shows model outputs model similar to Fig. S.3 above, but does include image synthesis conditioned on a nucleus image input (via Nucleus Image VQGAN). The predicted outputs are perceptually more similar to the ground truth protein

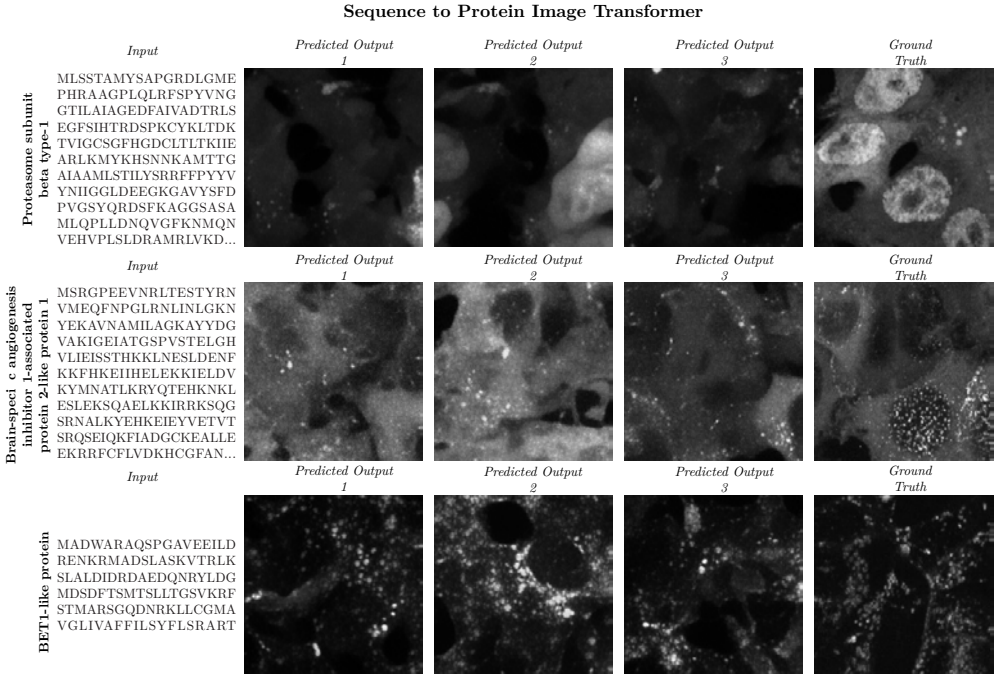



**Fig. S.3** DALL-E-like model with only amino acid sequence as the input. The sequence (left column) is used as input. The middle 3 columns show separate predicted images from random initialization. The true protein image is shown in the right column.

image, but the questions of accuracy and scientific utility mentioned previously continue to be a factor within this paradigm.

The sequence-to-image transformer is similar to CELL-E and does not take a nucleus image as input. This produces perceptually similar images (Fig. S.3). 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.

### S.2.3 Ablation Study

**Model Architecture** 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. S.5).

On the top row, performance from the VQGAN is used as a reference of possible top performance, just as in Table S.2. 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).

**Train**

| Sequence Embedding Model | Model Depth | Embedding Dimension | Fixed Sequence Embedding? | Epochs Trained | Nucleus Proportion Accuracy | Predicted Threshold Pixel Accuracy | Predicted 2D PDF Pixel Accuracy | SSIM | IS | FID | Nucleus Localization Prediction Accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N/A (VQGAN) | | | | 371 | .9927 ± .0084 | .8749 ± .0817 | | .5523 ± .2528 | 4.174 ± .1716 | 14.818 | .9933 ± .0810 |
| TAPE | 32 | 768 | ✓ | 130 | .9396 ± .0535 | .7678 ± .0647 | .6758 ± .1047 | .3154 ± .2100 | 2.7684 ± .0681 | 107.4147 | .8946 ± .3071 |
| TAPE | 26 | 768 | ✓ | 136 | .9475 ± .0465 | .7551 ± .0726 | **.6849 ± .1051** | .3146 ± .2195 | 2.9021 ± .0067 | 81.0061 | .9059 ± .2920 |
| TAPE | 26 | 768 | ✗ | 49 | .9260 ± .0656 | .7568 ± .0664 | .6131 ± .0928 | .2266 ± .1485 | 2.3797 ± .0742 | 140.339 | .8730 ± .3330 |
| TAPE | 20 | 768 | ✓ | 63 | .9433 ± .0501 | .7647 ± .0719 | .6815 ± .1081 | .3138 ± .2234 | 2.8539 ± .0474 | 98.5444 | .9004 ± .2994 |
| TAPE | 15 | 768 | ✓ | 84 | .9456 ± .0469 | .7485 ± .0704 | .6730 ± .1041 | .2955 ± .2122 | 2.687 ± .0742 | 112.4738 | **.9095 ± .2869** |
| TAPE (No Nuc.) | 32 | 768 | ✓ | 91 | .9115 ± .0762 | .7609 ± .0762 | .5743 ± .0887 | .2092 ± .1609 | 2.2793 ± .3975 | 137.8209 | .8516 ± .3554 |
| ESM1b | 20 | 1280 | ✓ | 54 | .9403 ± .0523 | .7610 ± .0664 | .6633 ± .1027 | .2855 ± .2057 | 2.6059 ± .1153 | 111.8584 | .8902 ± .3127 |
| UniRep | 15 | 1900 | ✓ | 8 | .6591 ± .1547 | .7420 ± .0632 | .6011 ± .0803 | .2108 ± .1410 | 2.1317 ± .0432 | 193.1455 | .8651 ± .3417 |
| AA Descriptors | 58 | 66 | ✓ | 60 | .9439 ± .0528 | .7674 ± .0690 | .6815 ± .1083 | .3220 ± .2261 | 2.8121 ± .0529 | 94.2488 | .9044 ± .2941 |
| AA Descriptors | 32 | 66 | ✓ | 66 | .9472 ± .0499 | .7622 ± .0792 | .6825 ± .1114 | **.3293 ± .2420** | **3.144 ± .1315** | **65.5666** | .9082 ± .2888 |
| One-Hot | 59 | 25 | ✓ | 100 | .9469 ± .0487 | .7524 ± .0681 | .6791 ± .1077 | .3141 ± .2212 | 2.7269 ± .0917 | 102.8544 | .9068 ± .2907 |
| One-Hot | 32 | 25 | ✓ | 70 | **.9478 ± .0476** | .7475 ± .0705 | .6769 ± .1073 | 0.3041 ± .2287 | 2.7469 ± .0905 | 90.0424 | .9057 ± .2922 |
| Random Initialization | 26 | 768 | ✗ | 71 | .9348 ± .0567 | **.7693 ± .0668** | .6606 ± .1093 | .2943 ± .2144 | 2.7503 ± .0852 | 110.9421 | ,8885 ± .3147 |
| No Sequence | 32 | 768 | | 51 | .9235 ± .0678 | .7578 ± .0749 | .6113 ± .0930 | .2196 ± .1517 | 2.2836 ± .0658 | 159.0451 | .8628 ± .3440 |

**Validation**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| N/A (VQGAN) | | | | 371 | 0.9921 ± .0091 | .8756 ± .0824 | | .5567 ± .2540 | 3.8718 ± .1662 | 25.9567 | .9923 ± .0872 |
| TAPE | 32 | 768 | ✓ | 130 | **.8078 ± .1837** | .7653 ± .0520 | .6342 ± .0964 | .2536 ± .1629 | 2.1300 ± .0704 | 155.7741 | .7155 ± .4511 |
| TAPE | 26 | 768 | ✓ | 136 | .7943 ± .1992 | .7574 ± .0607 | .6377 ± .1026 | .2446 ± .1689 | 2.2708 ± .1303 | 136.294 | .6979 ± .4594 |
| TAPE | 26 | 768 | ✗ | 49 | .7300 ± .2229 | .7505 ± .0648 | .6092 ± .0918 | .2148 ± .1436 | 2.2962 ± .1740 | 155.4038 | .5905 ± .4917 |
| TAPE | 20 | 768 | ✓ | 63 | .8024 ± .2071 | **.7739 ± .0613** | **.6388 ± .1088** | .2547 ± .1821 | 2.2641 ± .1379 | 156.3681 | .7017 ± .4575 |
| TAPE | 15 | 768 | ✓ | 84 | .8044 ± .1927 | .7422 ± .0554 | .6267 ± .0914 | .2226 ± .1546 | 2.0341 ± .0848 | 170.5773 | **.7247 ± .4467** |
| TAPE (No Nuc.) | 32 | 768 | ✓ | 91 | .7742 ± .2239 | .7733 ± .0660 | .5688 ± .0739 | .1956 ± .1311 | **3.3204 ± .1737** | **45.6887** | .6288 ± .4833 |
| ESM1b | 20 | 1280 | ✓ | 54 | .8044 ± .1851 | .7465 ± .0602 | .6235 ± .0916 | .2207 ± .1604 | 2.1925 ± .0868 | 146.5215 | .7224 ± .4478 |
| UniRep | 15 | 1900 | ✓ | 8 | .7474 ± .2063 | .7393 ± .0616 | .5982 ± .0789 | .2009 ± .1332 | 2.1287 ± .1231 | 206.3325 | .6173 ± .4862 |
| AA Descriptors | 58 | 66 | ✓ | 60 | .7854 ± .2234 | .7642 ± .0602 | .6340 ± .1020 | .2487 ± .1736 | 2.2561 ± .1030 | 144.271 | .7002 ± .4582 |
| AA Descriptors | 32 | 66 | ✓ | 66 | .7688 ± .2435 | .7676 ± .0761 | .6350 ± .1163 | **.2662 ± .2016** | 2.8155 ± .1382 | 88.4847 | .6817 ± .4658 |
| One-Hot | 59 | 25 | ✓ | 100 | .7688 ± .1989 | .7347 ± .0535 | .6207 ± .0904 | .2209 ± .1484 | 2.1214 ± .1182 | 159.35 | .6457 ± .4783 |
| One-Hot | 32 | 25 | ✓ | 70 | .7714 ± .2094 | .7383 ± .0599 | .6255 ± .1032 | .2252 ± .1703 | 2.2802 ± .1170 | 129.3922 | .6419 ± .4795 |
| Random Initialization | 26 | 768 | ✗ | 71 | .7587 ± .2169 | .7605 ± .0517 | .6204 ± .0948 | .2287 ± .1543 | 2.083 ± .1373 | 167.6092 | .6250 ± .4841 |
| No Sequence | 32 | 768 | | 51 | .7140 ± .2359 | .7528 ± .0733 | .6054 ± .0889 | .2032 ± .1387 | 2.2473 ± .1400 | 169.204 | .5721 ± .4948 |

**Table S.1** Full Results Table

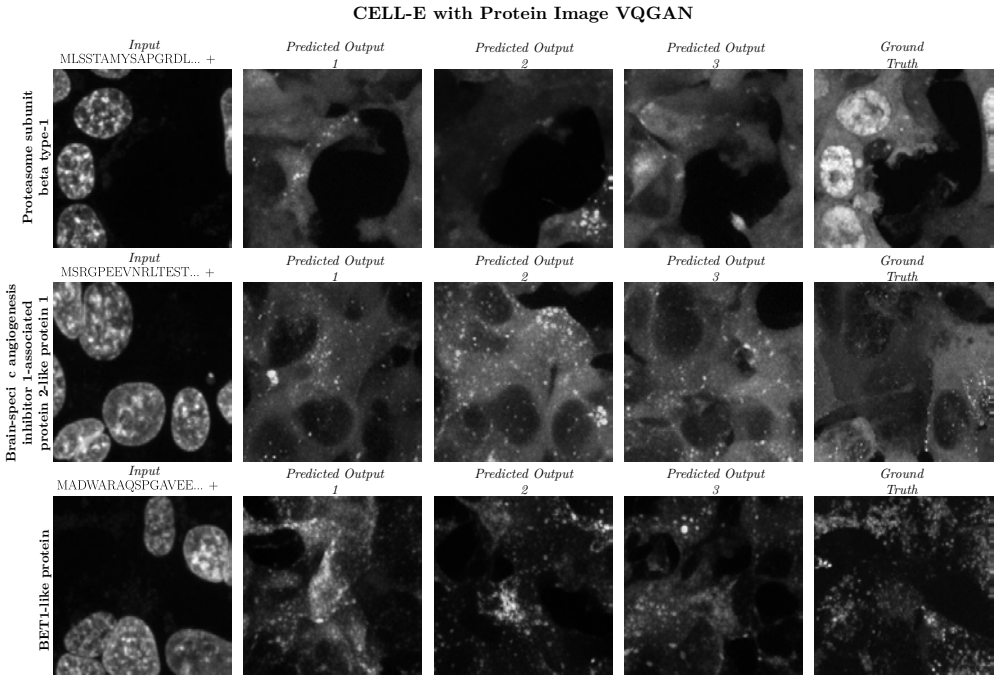



**Fig. S.4** CELL-E model with Protein Threshold VQGAN replaced with Protein Image VQGAN.

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 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 S.1. 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 model 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

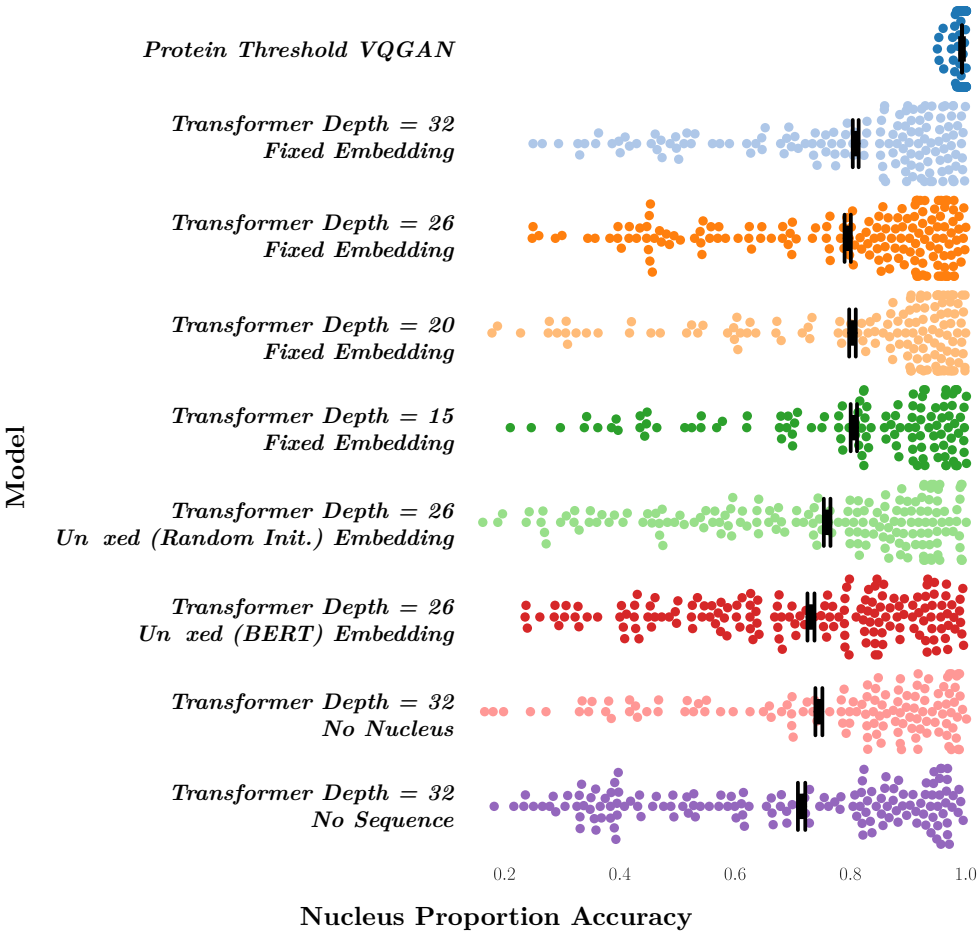



**Fig. S.5** Ablation Plot. "Fixed" and "Unfixed" embedding refer exclusively to the amino acid embeddings. Image embeddings are always unfixed. Mean values and standard deviation are marked in black. $\sim$ 150 points are randomly selected for display out of 1303 total predictions per model.

descriptors come from Osorio et. al. [61], which contains amino numerical descriptions of amino acid properties from various literature sources [62–71]. 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.

# Appendix S.3 Supplementary Methods

## S.3.1 Dataset

Each protein entry in OpenCell is accompanied by multiple high-resolution 3D confocal images containing multiple cells [13]. 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 out demonstration, we converted a 3D z-stack into a 2D maximum intensity projection [72], which still clearly depicts most subcellular structures and allow subtle subcellular protein localization differences to be distinguished from the OpenCell images [13].

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 [13]. 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 [73]. 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 [74].

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.

## S.3.2 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 results in memorization of training sequences and inflated performance metrics [75].

To investigate the effect of this on CELL-E, we performed a clustered split using a procedure identical to the one used by [76] to create, a standard dataset used in benchmarking protein localization prediction. This model relies on PSI-CD-HIT [77]. 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.

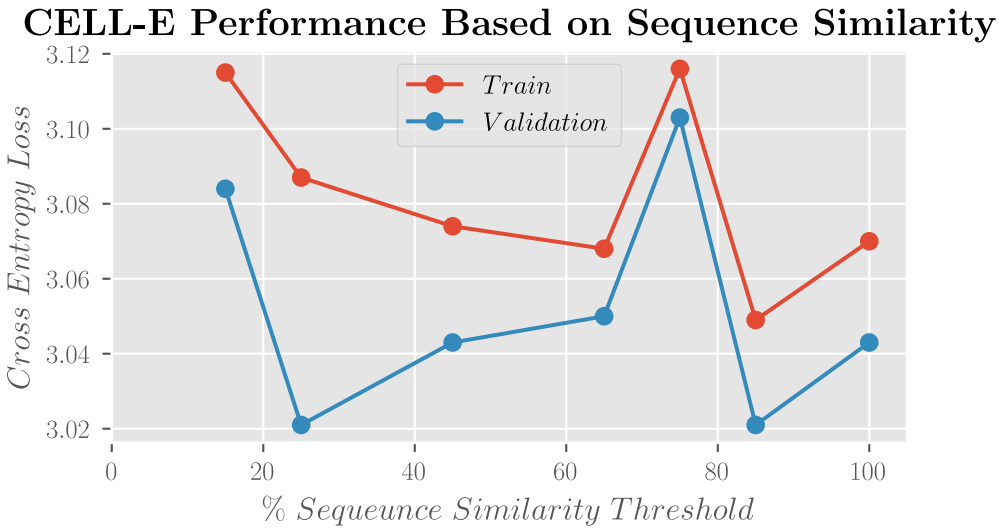



**Fig. S.6**

### S.3.3 Training

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.

### S.3.4 Nucleus Image Encoder

VQGAN code was obtained from Esser et. al. [18], 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. Adam Optimizer was used with learning rate set to $4.5 \times 10^{-6}$. The model was initially trained solely using mean-squared error reconstruction loss. After 50,000 steps, $\sim$ 7 epochs, the discriminator loss term was introduced. This terms 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).

### S.3.5 Protein Threshold Image Encoder

The model was trained on random 256 × 256 crops of 512 × 512 nuclei images. Adam Optimizer was used with learning rate set to $4.5 \times 10^{-6}$. The model was initially trained solely using mean-squared error reconstruction loss. After 50,000 steps, $\sim$ 7 epochs, the discriminator loss term was introduced. This terms 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).

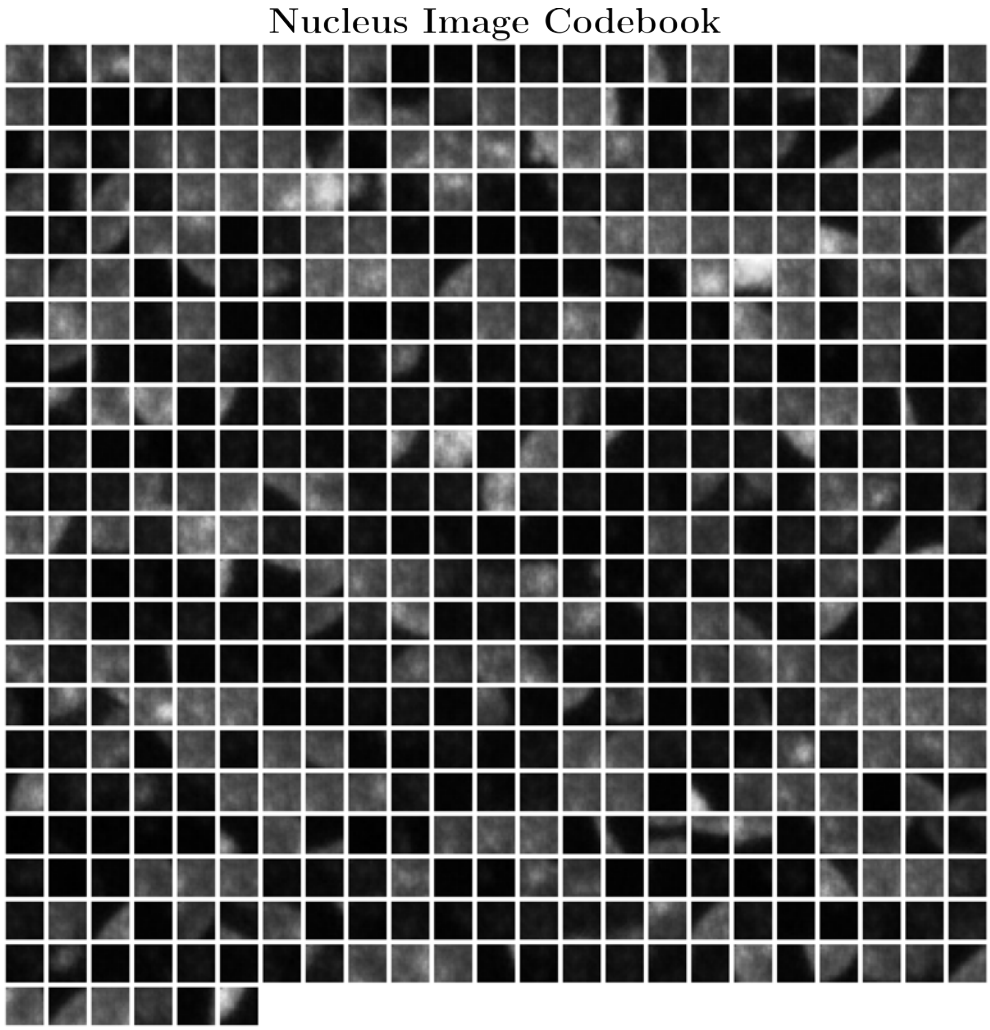



**Fig. S.7** 512 image patches extracted from the nucleus reference VQGAN

### S.3.6 CELL-E Transformer

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 sequences 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

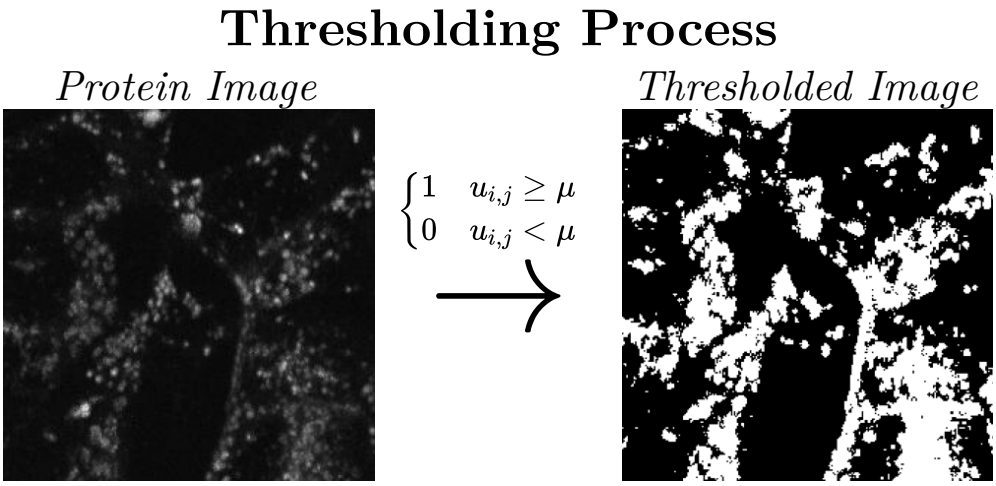



**Fig. S.8** Example of thresholding process to convert protein image (left) to thresholded ground truth image (right) for CELL-E model

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 $1001 \times 768$. 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 S.1.

CELL-E was trained with an Adam optimizer with learning rate set to $3 \times 10^{-4}$.

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 $1512 \times 768$ ($1001 \times 768$ for text, $256 \times 768$ for nucleus images and $255 \times 768$ for protein threshold images) (Fig. 2).

A rotary positional embedding [78] is then applied to the input embeddings.

We noted improved performance by shifting embeddings over by 1 (time-shifting [79]) 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 [80] 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

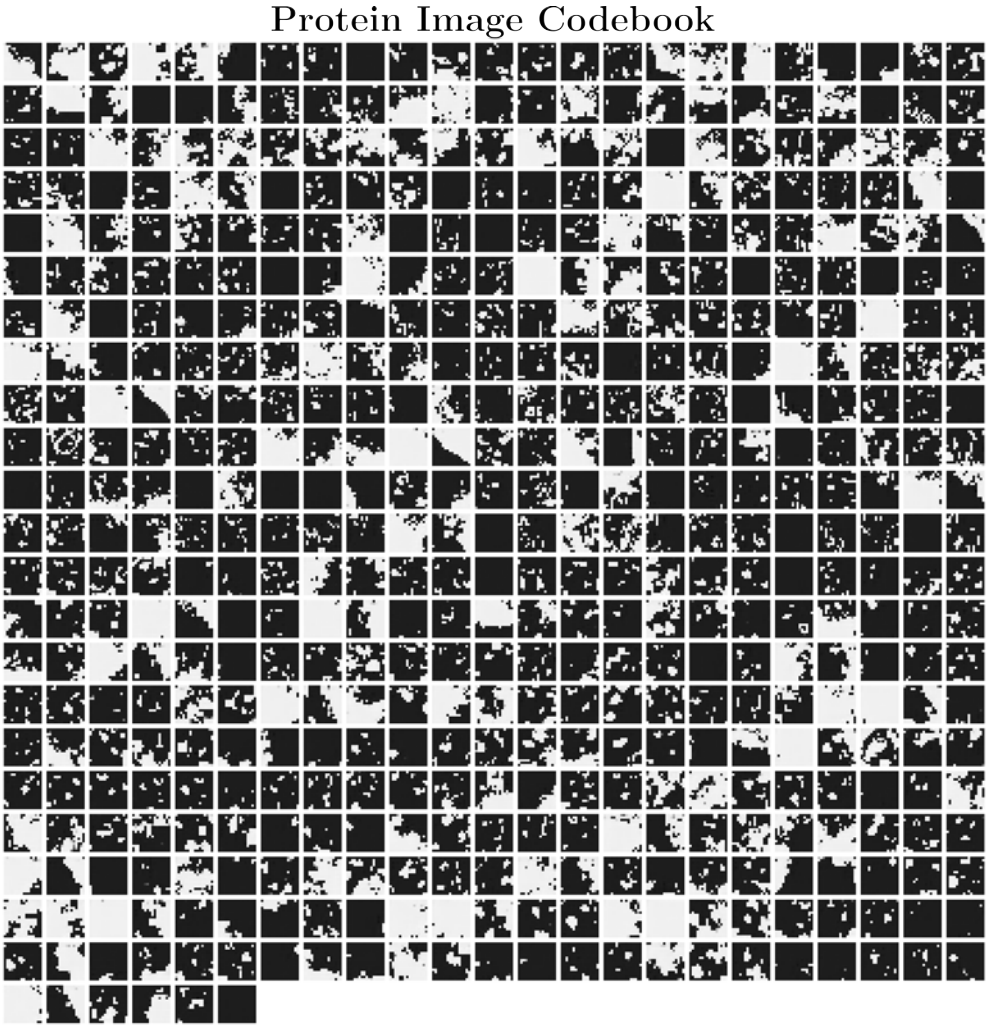



**Fig. S.9** 512 image patches extracted from the protein threshold VQGAN

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.

**Table S.2** Image Accuracy

| | Train | | Validation | |
|---|---|---|---|---|
| | CELL-E | VQGAN | CELL-E | VQGAN |
| Nucleus Proportion Accuracy | $0.94 \pm 0.05$ | $0.99 \pm 0.01$ | $0.81 \pm 0.18$ | $0.99 \pm 0.01$ |
| Predicted Threshold Pixel Accuracy | $0.77 \pm 0.06$ | $0.87 \pm 0.08$ | $0.77 \pm 0.05$ | $0.88 \pm 0.08$ |
| Predicted 2D PDF Pixel Accuracy | $0.68 \pm 0.10$ | | $0.63 \pm 0.10$ | |
| Structual Similarity Index Measure | $0.32 \pm 0.21$ | $0.55 \pm 0.25$ | $0.25 \pm 0.16$ | $0.56 \pm 0.25$ |
| Inception Score | $2.77 \pm 0.07$ | $4.17 \pm 0.17$ | $2.13 \pm 0.07$ | $3.87 \pm 0.17$ |
| 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.

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.

### S.3.7 Performance Evaluation

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.

#### Nucleus Proportion Accuracy

To calculate the proportion of intensity in the nucleus, we first create a mask (Fig. S.10 of the nucleus channel using Cellpose [81]. We take a sum over the predicted 2D PDF pixels found within the nucleus mask, and divide this by the sum of pixels across the image.

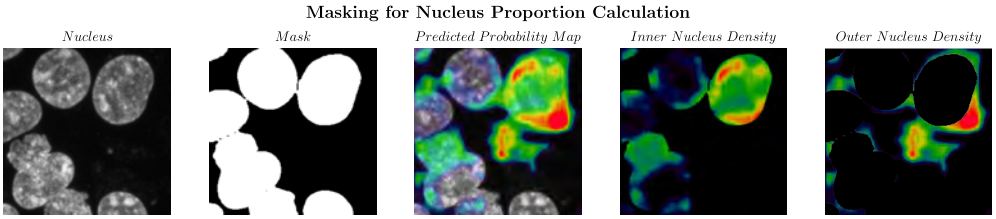



**Fig. S.10** Masking procedure depicted.

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 - 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.

### 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.

### 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.

### S.3.8 Visualizing Attention

To obtain Fig. 6, we first split the 16x16 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 [82] 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.

We show the entire image generation process, with frames corresponding to time-steps, in the attached video file: DNAtopoisomerase1.mp4.

## References

[1] 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.

[2] 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.

[3] 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.

[4] Imai, K. & Nakai, K. Prediction of subcellular locations of proteins: where to proceed? *Proteomics* **10**, 3970–3983 (2010).

[5] 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).

[6] 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.

[7] 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.

[8] The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic acids research* **45**, D158–D169 (2017). Place: England.

[9] Jha, S. K., Ramanathan, A., Ewetz, R., Velasquez, A. & Jha, S. Protein Folding Neural Networks Are Not Robust. *arXiv:2109.04460 [cs, q-bio]* (2021). URL http://arxiv.org/abs/2109.04460. ArXiv: 2109.04460.

[10] 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.

[11] Ding, M. *et al.* CogView: Mastering Text-to-Image Generation via Transformers. *arXiv:2105.13290 [cs]* (2021). URL http://arxiv.org/abs/2105.13290. ArXiv: 2105.13290.

[12] 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].

[13] 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.

[14] 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).

[15] 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.

[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] 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.

[18] 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.

[19] 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.

[20] 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.

[21] Dingwall, C., Robbins, J., Dilworth, S. M., Roberts, B. & Richardson, W. D. The Nucleoplasmin Nuclear Location Sequence Is Larger and MoreComplex than That of S¥-40 Large T Antigen. *The Journal of Cell Biology* **107**, 9 (1988).

[22] 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.

[23] 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).

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] Schuler, G. D. *et al.* A gene map of the human genome. *Science (New York, N.Y.)* **274**, 540–546 (1996).

[30] 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.

[31] 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.

[32] Liu, P. J. *et al. Generating Wikipedia by Summarizing Long Sequences* (2023). URL https://openreview.net/forum?id=Hyg0vbWC-.

[33] 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.

[34] Wang, P. DALL-E in Pytorch (2022). URL https://github.com/lucidrains/DALLE-pytorch. Original-date: 2021-01-05T20:35:16Z.

[35] 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.

[36] 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.

[37] 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).

[38] 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.

[39] 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.

[40] 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/.

[41] 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.

[42] 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.

[43] 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.

[44] Pan, G., Sun, C., Liao, Z. & Tang, J. in *Machine and Deep LearningDeep 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_15.

[45] 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.

[46] 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.

[47] Pang, L., Wang, J., Zhao, L., Wang, C. & Zhan, H. A Novel Protein Subcellular Localization 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.

[48] Yang, W.-Y., Lu, B.-L. & Yang, Y. *A Comparative Study on Feature Extraction from Protein Sequences for Subcellular Localization Prediction*, 1–8 (2006).

[49] 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).

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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/.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

[59] 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.

[60] 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.

[61] Osorio, D., Rondón-Villarreal, P. & Torres Sáez, R. Peptides: A Package for Data Mining of Antimicrobial Peptides. *The R Journal* **7**, 4–14 (2015).

[62] 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.

[63] 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.

[64] 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.

[65] 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.

[66] 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.

[67] 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).

[68] 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/.

[69] 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).

[70] 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).

[71] 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).

[72] 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.

[73] 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.

[74] Schnell, U., Dijk, F., Sjollema, 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.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] Bo, P. Improve the Transformer self-attention mechanism with just a few lines of code (almost no increase in computation). URL https://zhuanlan-zhihu-com.translate.goog/p/191393788?_x_tr_sl=en&_x_tr_tl=zh-CN&_x_tr_hl=en&_x_tr_pto=wapp.

[80] 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].

[81] 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.

[82] 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.