MAESTER: Masked Autoencoder Guided Segmentation at Pixel Resolution for Accurate, Self-Supervised Subcellular Structure Recognition Supplemental Material

Ronald Xie[1,2,3,4,*,†] Kuan Pang[1,4,*] Gary D. Bader[1,3,4,‡] Bo Wang[1,2,3,‡]

[1]University of Toronto, [2]Vector Institute, [3]University Health Network, [4]The Donnelly Centre

{ronald.xie, kuan.pang, gary.bader}@mail.utoronto.ca , bowang@vectorinstitute.ai

A. Implementation Details

A.1. Self-supervised Segmentation Experiments

We use a ViT encoder [1] to learn semantically informative patch level representations. We present the default training settings in Table S1. Different from other image segmentation tasks, volume electron microscopy (VEM) datasets are stored as 3-D arrays which requires sampling 2-D images from the volume, and thereby conventional epoch measure does not apply. For each reconstruction task, we sample 5.6M images at the size of FOV from the dataset with random flip and random crop data augmentations.

Our ViT encoder and decoder implementations are based on modified ViT-B architecture. Modifications include decreasing embedding dimension to overcome practical limitations of storage while maintaining sufficient expressivity of learned patch representations. We also adjusted the number of layers to enable asymmetric encoder and decoder design. Positional embeddings are added in both encoder and decoder. In encoder, the positional embeddings are weighted to avoid the positional bias in the clustering.

A.2. Supervised Segmentation Experiments

We evaluate our method against two strong supervised baselines, Segmenter [3] and Vanilla ViT [1]. Segmenter is a ViT backboned model for semantic segmentation, it consists of a ViT encoder for processing feature extraction and a Mask Transformer for matching embeddings with classes and mapping the embedding to pixel classification. Vanilla ViT baseline uses a vanilla ViT encoder for feature extraction and a vanilla ViT decoder for mapping embeddings to segmentation map. Both baselines are configured to have same or similar architecture to our default setting in self-supervised experiment. We expect these approaches to illustrate ViT's capability in achieving pixel-precision performance with supervised signal. We report the training settings for supervised baseline in Table S2, and model specific settings in Table S3 and Table S4. Since each cell in the dataset is partially labeled, we disable the loss calculation on the unlabeled regions.

config	value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	1.0e-5
optimizer momentum	β_1, β_2=0.9, 0.0.999
batch size	32
training samples	5.6M
augmentation	RandFlip, RandCrop

Table S1. Default training settings for self-supervised experiments.

config	value
optimizer	AdamW
base learning rate	1.5e-4
weight decay	1.0e-5
optimizer momentum	β_1, β_2=0.9, 0.0.999
batch size	64
training samples	5.6M
augmentation	RandFlip, RandCrop
loss function	Cross Entropy Loss

Table S2. Default training settings for supervised experiments.

config	ViT encoder	Mask Transformer decoder
embedding dimension	192	128
transformer layer	14	7
attention head	1	8
MLP ratio	2.0	2.0
positional embedding weight	1.0	1.0

Table S3. Segmenter [3] architecture for supervised experiments.

[*] Equal contribution

[†] Project lead

[‡] Co-senior author

config	ViT encoder	ViT decoder
embedding dimension	192	128
transformer layer	14	7
attention head	1	8
MLP ratio	2.0	2.0
positional embedding weight	1.0	1.0

Table S4. Vanilla ViT [1] architecture for supervised experiments.

B. Supplementary Figures

To achieve pixel resolution segmentation of the entire betaSeg testing dataset, we generated over 600 million patches and associated learnt token representations. To compute k-mean centers for label assignment, we randomly sample 500 thousand, or around 0.08% of the total number of patches for practicality. We then further subset 50 thousand token representations for visualization via UMAP [2]. The UMAP was colored based on the reference segmentation and the matching predicted classes from the result of k-means clustering Figure S1. As shown, we see clear semantic separation of putative classes generated by k-means clustering, in concordance with the reference segmentation. We also present a confusion matrix Figure S2 showing the robustness and accuracy of our generated segmentation.

3D renderings of our generated segmentation are shown in Figure S3 and Figure S4. Precise segmentation enables cell biologist to quickly gain an holistic overview of cells of interest and facilitate downstream analysis on the segmented subcellular structures.

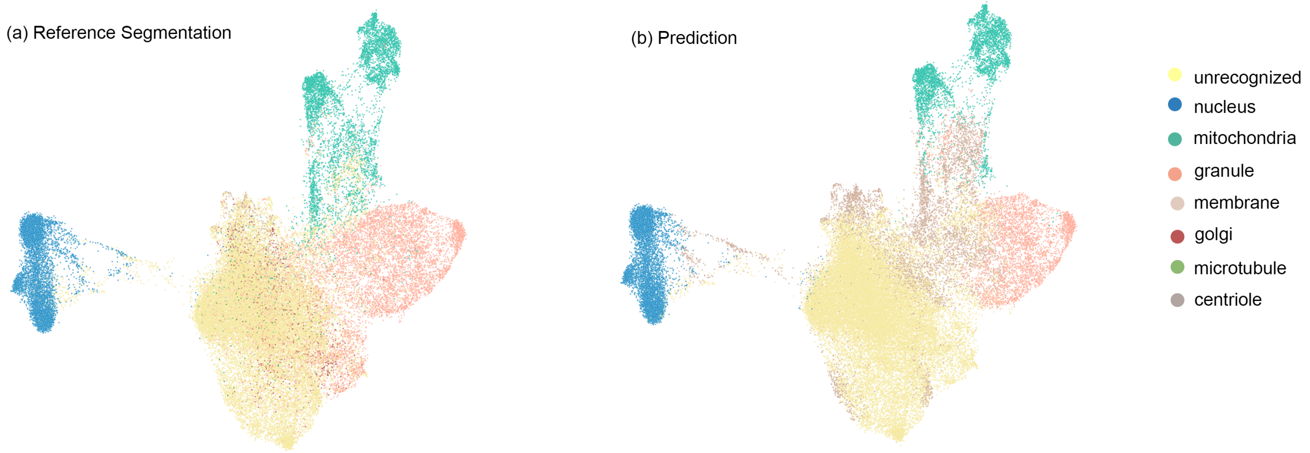

Figure S1. UMAP [2] visualizations of $50,000$ randomly selected token representations generated by the trained encoder, demonstrating clear semantic separation of the putative classes. (a) Color coded based on ground truth segmentation classes. (b) Color coded based on our predicted classes.

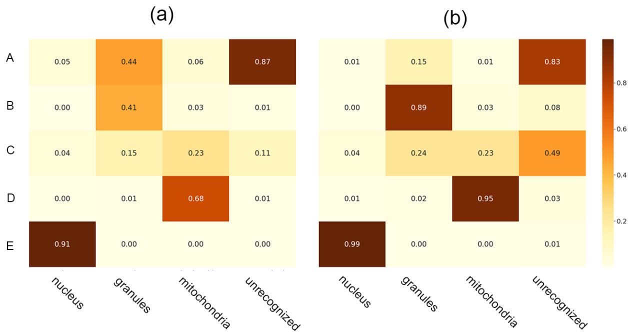

Figure S2. Confusion matrix showing $k = 6$ (with one class merged), (a) is normalized by column and (b) is normalized by row.

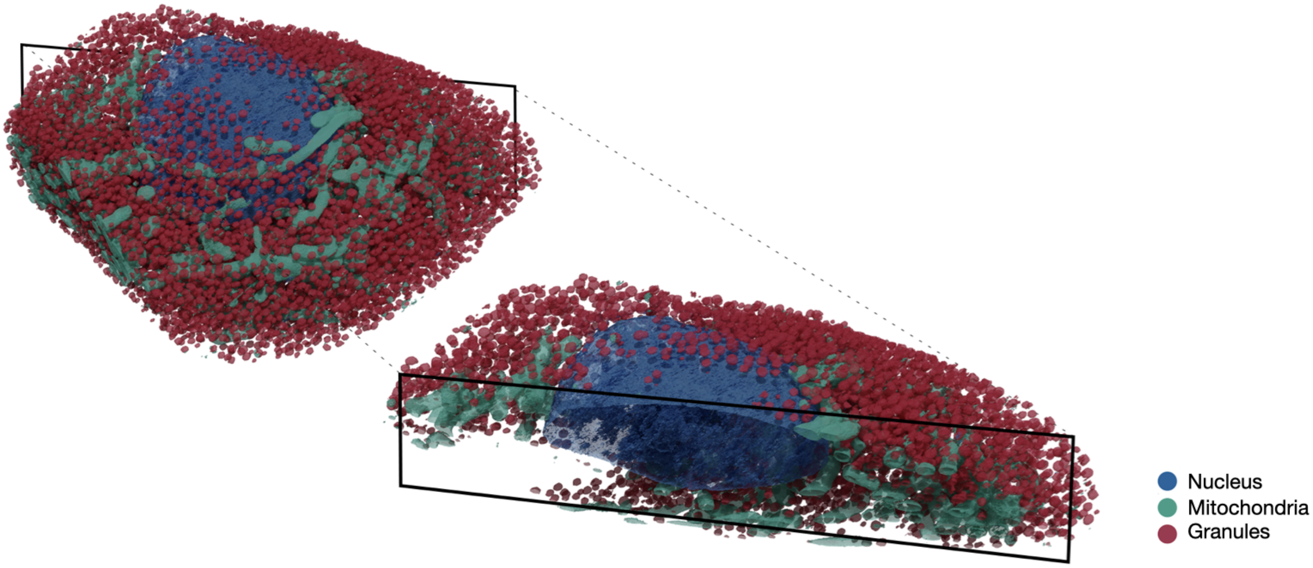

Figure S3. 3D-renderings of an overview for our segmentation result on testing cell, allowing biologists to quickly identify the cell ultrastructures.

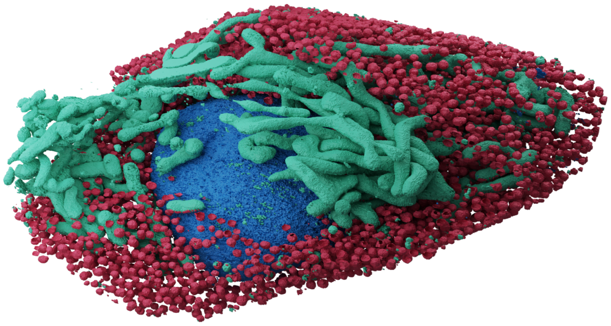

Figure S4. 3D-renderings of our segmentation result on testing cell with 1/4 granules hidden for subcellular structure demonstration.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2

[2] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. 2, 3

[3] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021. 1