

1 **CEM500K – A large-scale heterogeneous unlabeled cellular electron microscopy image**  
2 **dataset for deep learning.**

3  
4 **Ryan Conrad<sup>1,2</sup> and Kedar Narayan<sup>1,2</sup>**

5 <sup>1</sup>Center for Molecular Microscopy, Center for Cancer Research, National Cancer Institute,  
6 National Institutes of Health, Bethesda, Maryland, USA. <sup>2</sup>Cancer Research Technology Program,  
7 Frederick National Laboratory for Cancer Research, Frederick, Maryland, USA.

8  
9 **Keywords:** electron microscopy, volume electron microscopy, vEM, convolutional neural  
10 networks, deep learning, automated segmentation, pre-training dataset, reference dataset

11  
12 **Abstract**

13 Automated segmentation of cellular electron microscopy (EM) datasets remains a challenge.  
14 Supervised deep learning (DL) methods that rely on region-of-interest (ROI) annotations yield  
15 models that fail to generalize to unrelated datasets. Newer unsupervised DL algorithms require  
16 relevant pre-training images, however, pre-training on currently available EM datasets is  
17 computationally expensive and shows little value for unseen biological contexts, as these  
18 datasets are large and homogeneous. To address this issue, we present CEM500K, a nimble 25  
19 GB dataset of  $0.5 \times 10^6$  unique cellular EM images curated from nearly 600 three-dimensional  
20 (3D) and 10,000 two-dimensional (2D) images from >100 unrelated imaging projects. We show  
21 that models pre-trained on CEM500K learn features that are biologically relevant and resilient to  
22 meaningful image augmentations. Critically, we evaluate transfer learning from these pre-trained  
23 models on six publicly available and one newly derived benchmark segmentation task and report  
24 state-of-the-art results on each. We release the CEM500K dataset, pre-trained models and  
25 curation pipeline for model building and further expansion by the EM community. Data and code  
26 are available at <https://www.ebi.ac.uk/pdbe/emdb/empiar/entry/10592/> and <https://git.io/JLLTz>.

27  
28  
29  
30

## 31 **Introduction**

32

33 Accurate image segmentation is essential for analyzing the structure of organelles and cells in  
34 electron microscopy (EM) image datasets. Segmentation of volume EM (vEM) data has enabled  
35 researchers to address questions of fundamental biological interest, including the organization of  
36 neural circuits [1][2] and the structure of various organelles [3][4][5]. Truly automated EM  
37 image segmentation methods hold the promise of significantly accelerating the rate of discovery  
38 by enabling researchers to extract and analyze information from their datasets without months or  
39 years of tedious manual labeling. While supervised deep learning (DL) models are effective at  
40 the segmentation of objects in natural images (e.g. of people, cars, furniture, and landscapes)  
41 [6][7][8][9] they require significant human oversight and correction when applied to the  
42 organelles and cellular structures captured by EM [10][11].

43

44 Many of the limitations of supervised DL segmentation models for cellular EM data result from  
45 a lack of large and, importantly, diverse training datasets [12][13][14]. Although several  
46 annotated image datasets for cell and organelle segmentation are publicly available, these often  
47 exclusively consist of images from a single experiment or tissue type, and a single imaging  
48 approach [15][16][17][18][19]. The homogeneity of such datasets often means that they are  
49 ineffective for training DL models to accurately segment images from unseen experiments.  
50 Instead, when confronted with new data, the norm is to extract and annotate small regions-of-  
51 interest (ROIs) from the EM image, train a model on the ROIs, and then apply the model to infer  
52 segmentations for the remaining unlabeled data [15][16][17][18][19][20][21]. Often, not only are  
53 these models dataset-specialized, reducing their utility, they often fail to generalize even to parts  
54 of the same dataset that are spatially distant from the training ROIs [16][22].

55

56 Gathering more annotated data for model training from disparate sources could certainly improve  
57 a model's ability to generalize to unseen images, yet it is rarely feasible for typical research  
58 laboratories to generate truly novel datasets; most have expertise in a particular imaging  
59 technique, organism or tissue type. Beyond collecting the EM data, manual segmentation is time-  
60 consuming and, unlike for natural images, difficult to crowdsource because of the extensive  
61 domain knowledge required to identify objects in novel cellular contexts. Promising work is

62 being done in the area of citizen science as it pertains to EM data, but it is clear that there are  
63 limitations to the range of structures that can be accurately segmented by volunteers [23][24].  
64 Moreover, structure-specific annotations will not solve the generalization problem for all  
65 possible EM segmentation targets; for example, thousands of hours spent labeling neurites is  
66 unlikely to buy any gains for mitochondrial segmentation. An efficient alternative to collecting  
67 additional structure-specific data is to use transfer learning. In transfer learning, a DL model is  
68 pre-trained on a general task and its parameters are reused for more specialized downstream  
69 tasks. A well-known example is to transfer parameters learned from the ImageNet classification  
70 task [25] to other classification or object detection tasks which have fewer training examples  
71 [26]. Transfer learning, when relevant pre-trained parameters are available, is the default  
72 approach for extracting the best performance out of small training datasets [27][28]. While  
73 ImageNet pre-trained models are sometimes used for cellular EM segmentation tasks [29][30],  
74 high-level features learned from ImageNet may not be applicable to biological imaging domains  
75 [31]. Building a more domain-specific annotated dataset large enough for pre-training would be a  
76 significant bottleneck, and indeed, it required multiple years to annotate the  $3.2 \times 10^6$  images that  
77 form the basis of ImageNet. Fortunately, recent advances in unsupervised learning algorithms  
78 have now enabled effective pre-training and transfer learning without the need for any up front  
79 annotations; in fact, on many tested benchmarks, unsupervised pre-training leads to better  
80 transfer learning performance [32][33][34][35][36][37][38].

81

82 To provide a resource for the EM community to explore these exciting advances, we constructed  
83 an unlabeled cellular EM dataset which we call CEMraw, containing images from 101 unrelated  
84 biological projects. The image data superset, comprising 591 3D image volumes and 9,626 2D  
85 images are collated from a collection of experiments conducted in our own laboratory as well as  
86 data from publicly available sources. After gathering this set of heterogeneous images, we create  
87 a pipeline where we first remove many nearly identical images and then filter out low-quality  
88 and low-information images. This results in a highly information-rich, relevant, and non-  
89 redundant 25 GB image dataset comprising  $0.5 \times 10^6$  images. As a proof of concept for its  
90 potential applications, we pre-trained a DL model on CEM500K using an unsupervised  
91 algorithm, MoCoV2 [39], and evaluated the results for transfer learning on six publicly available  
92 benchmarks and one newly derived benchmark that we introduce in this work. CEM500K pre-

93 trained models significantly outperformed randomly initialized and ImageNet pre-trained  
94 models, as well as previous baseline results from benchmark-associated publications.

95

96

## 97 **Results**

98

### 99 *Creation of CEM500K*

100 In order to create an image dataset that is relevant to cellular EM and yet general enough to be  
101 applicable to a variety of biological studies and experimental approaches, we collected 2D and  
102 3D cellular EM images from both our own experiments and publicly available sources. These  
103 included images from a variety of imaging modalities and their corresponding sample  
104 preparation protocols, resolutions reported, and cell types imaged (**Fig. 1 a-c**). We selected “in-  
105 house” datasets corresponding to 251 reconstructed FIB-SEM volumes from 33 unrelated  
106 experiments and 2,975 TEM images from 35 additional experiments. Other data was sourced  
107 externally; as there is currently no central hub for accessing publicly available datasets, we  
108 manually searched through databases (Cell Image Library, Open Connectome Project [40],  
109 EMPIAR [41]), GitHub repositories, and publications. A complete accounting of the datasets  
110 with relevant attribution is detailed in the **Supplementary Materials**. Included in this batch of  
111 data were 340 EM image volumes (some derived from video data) from 26 experiments and  
112 9,792 2D images from 14 other experiments. Among the externally gathered datasets there were  
113 disparate file types (avi, mp4, png, tiff, jpeg, mrc, nii.gz) and pixel/voxel data types (signed and  
114 unsigned, 32-bit float, 8-bit and 16-bit integer) as well as a mixture of image volumes with  
115 isotropic or anisotropic voxels, and regular or inverted intensities. These data were standardized  
116 into 2D tiff images, or patches, of 224 x 224 8-bit unsigned pixels (see Materials and Methods);  
117 the resulting set of  $5.3 \times 10^6$  images constitutes what we term CEMraw (**Fig. 1 d, top**).

118

119 Within CEMraw, however, most images were redundant. Nearly identical patches existed  
120 because of the similarity between adjacent cross-sections in high-resolution 3D volumes as well  
121 as in patches cropped from uniform intensity regions like empty resin. Duplicates are not only  
122 memory and computationally inefficient, but they may also induce undesirable biases toward the  
123 most frequently sampled features in the dataset. Therefore, we aggressively removed duplicates  
124 using an automated algorithm: we calculated and compared image hashes for each patch in  
125 CEMraw and then kept a single, randomly chosen exemplar image from each group of near  
126 duplicates (see Materials and Methods). As a result of this operation, we obtained an 80%  
127 decrease in the number of patches when compared to CEMraw; this “deduplicated” subset of 1.1

128  $\times 10^6$  image patches we refer to as CEMdedup (**Fig. 1 d, middle**). Although it is currently  
129 impossible to determine *a priori* what data will be useful for a model, we expect that this  
130 removal of significant redundancies in the image dataset is unlikely to result in the loss of  
131 meaningful information for DL model training.

132  
133 Deduplication ensures that each image will make a unique contribution to our dataset, but it is  
134 agnostic to the content of the image, which may or may not be relevant to downstream tasks.  
135 Upon visual inspection, it was clear that many of the images in CEMdedup contained little  
136 information useful to the segmentation of organelles or cellular structures, e.g., images  
137 dominated by empty resin, background padding, or homogeneously stained interiors of nuclei or  
138 cytoplasm (**Supplementary Figure 1a**). However, while these images were uninformative for  
139 our purposes, they also represented a wide variety of image features, making them challenging to  
140 identify with simple image statistics. Instead, we separated an arbitrary subset of 12,000 images  
141 from CEMdedup into informative and uninformative classes and trained a DL model to perform  
142 binary classification on the entire dataset. Uninformative images were characterized by poor  
143 contrast, large areas of uniform intensity, artifacts, and the presence of non-cellular objects.  
144 Detailed criteria are given in Materials and Methods. The classifier achieved an area under the  
145 receiver operating characteristic (AUROC) score of 0.962 on a holdout test set of 2,000 images,  
146 as shown in **Supplementary Figure 1b**, suggesting that it could reliably distinguish between the  
147 informative and uninformative image classes. Classification of the remaining unlabeled images  
148 with this model yielded  $0.5 \times 10^6$  patches with a visibly higher density of views containing  
149 organelles and cellular structures. We refer to this final subset of uniquely informative 2D  
150 cellular EM images as CEM500K (**Fig. 1 d, bottom**). Representative patches from the three  
151 datasets (CEMraw, CEMdedup and CEM500K) are shown in **Supplementary Figure 2**.

152  
153 *Test of pre-training by CEM500K*

154 We then decided to test CEM500K for unsupervised pre-training of a DL model, using the  
155 MoCoV2 algorithm, a relatively new and computationally efficient approach [34]. The algorithm  
156 works by training a DL model to match differently augmented (e.g., cropped, rotated, zoomed in,  
157 brightened, etc.) pairs of images. The first batch of augmented images is called the query and the  
158 batch of their differently augmented counterparts is called the key. Before matching, the encoded

159 images in the key are added to a continuously updated queue containing tens of thousands of  
160 recently seen images (**Supplementary Figure 3a**). To be useful for other tasks, it is assumed  
161 that the model will learn features that correspond to relevant objects within the training images.  
162 Recently, models pre-trained on ImageNet with the MoCoV2 algorithm have shown superior  
163 transfer learning performance over supervised methods when applied to a variety of tasks  
164 including segmentation [39]. Before we were able to evaluate the MoCoV2 algorithm on  
165 CEM500K, it was necessary to define a set of downstream tasks to quantify and compare  
166 performance. We chose six publicly available benchmark datasets: CREMI Synaptic Clefts [42],  
167 Guay [15], Kasthuri++ and Lucchi++ [17], Perez [18] and UroCell [16]. The benchmarks  
168 included a total of eight organelles or subcellular structures for segmentation (mitochondria,  
169 lysosomes, nuclei, nucleoli, canalicular channels, alpha granules, dense granules, dense granule  
170 cores, and synaptic clefts). In **Fig. 2a** we show representative images and label maps from the  
171 benchmarks. Additional information about the benchmarks, including imaging techniques and  
172 sizes of the training and test sets, is given in **Supplementary Table 1**.

173  
174 Performance on each benchmark was measured using the standard Intersection-over-Union (IoU)  
175 score. Considered on their own, many of these benchmark datasets are not difficult enough to  
176 expose the gap in performance between different models: they only require the segmentation of a  
177 single organelle within a test set that is often from the same image volume as the training set. At  
178 the same time, they are an accurate reflection of common use cases for deep learning in EM  
179 laboratories where the goal is to segment data from a single experiment in order to support  
180 biological, not computational, research. To address the lack of variety within the benchmark  
181 training and test sets, we derived an additional benchmark that we call All Mitochondria, which  
182 is a combination of the training and test sets from each of the five benchmarks that contain label  
183 maps for mitochondria (Guay, Perez, UroCell, Lucchi++ and Kasthuri++; the labels for all other  
184 objects were removed). Although this benchmark is specific to a single organelle, it is  
185 challenging in that it requires a model to learn features that are general for mitochondria from  
186 image volumes generated independently and from unrelated experiments and imaging  
187 parameters.

188

189 Our overall pre-training, transfer, and evaluation workflow is shown in a schematic in **Fig. 2b**.  
190 Pre-training was performed by applying the MoCoV2 algorithm to learn parameters for a  
191 ResNet50 [43] before transferring the parameters into the encoder of a U-net [44]. A detailed  
192 schematic of the UNet-ResNet50 architecture is shown in **Supplementary Figure 3b**. For this  
193 section, once transferred, the parameters were frozen such that no updates were made during  
194 training on the benchmark tasks; this enabled us to isolate the effects of pre-training. As a simple  
195 baseline reference for calibrating later results, we started by measuring the performance of the  
196 proposed segmentation model with randomly initialized and frozen encoder parameters (i.e., we  
197 skipped the pre-training step in the workflow); the results for each benchmark are shown **Fig. 2c**.  
198 Given that in our architecture, the encoder includes approximately  $23 \times 10^6$  parameters and the  
199 decoder approximately  $9 \times 10^6$  parameters, some 70% of the model's parameters were never  
200 been updated during training. Still, some benchmarks permit strikingly good performance, with  
201 IoU scores of over 0.75 on both Lucchi++ and Kasthuri++. These results emphasize the necessity  
202 of evaluating deep learning algorithms and pre-training datasets on multiple benchmarks before  
203 drawing conclusions about their quality.

204  
205 We next tested the influence of our curation pipeline on the quality of pre-trained parameters.  
206 We pre-trained models on CEMraw, CEMdedup and CEM500K with an abbreviated training  
207 schedule (see Materials and Methods) and compare the IoU scores achieved on the benchmarks  
208 in **Fig. 2d** (the actual IoU scores are shown in **Table 1**). We observed that pre-training on  
209 CEM500K gave better or equivalent results than the CEMraw superset and CEMdedup subset  
210 for every benchmark. The average increase in performance of CEM500K over CEMraw was  
211 4.5%, and CEM500K over CEMdedup was 2.0%, with a maximum increase of 12.3% and 4.1%,  
212 respectively, on the UroCell benchmark (IoU scores increased from 0.652 and 0.699 to 0.729).  
213 These increases are significant. As a comparison, a 2% increase in model performance is similar  
214 in magnitude to what might be expected from using an ensemble of a few models [45]. Besides  
215 these gains, curation is valuable for reducing the computational cost of using CEM500K: the  
216 final filtered subset is 90% smaller than the raw superset (25 GB compared to 250 GB).  
217 Deduplication and filtering likely contributed to the performance gain by enabling both faster  
218 convergence and the learning of more relevant feature detectors. Duplicate images consume  
219 training iterations without presumably transmitting any new information, resulting in slower



220 learning. Uninformative images, on the other hand, may guide a model to discover discriminative  
221 features that are useless for most segmentation tasks. For example, a model must learn feature  
222 detectors that can distinguish between images of empty resin in order to succeed on the pre-  
223 training task, but those feature detectors are unlikely to help with a common task like  
224 mitochondrial segmentation. Therefore, eliminating uninformative images may reduce the  
225 learning of irrelevant details during pre-training.

226  
227 We also posited that, in addition to the benefits of curation, the heterogeneity of examples in  
228 CEM500K would be essential for achieving good segmentation performance across disparate  
229 biological contexts. To test this, we considered an alternative pre-training dataset consisting  
230 exclusively of  $1 \times 10^6$  images from a single large connectomics volume of mouse brain tissue  
231 (Bloss et al., 2018) [46]. Coming from a single volume of a highly homogeneous tissue type,  
232 images in this dataset show much less variation in cellular features than those in CEM500K (a  
233 random sampling of images is shown in **Supplementary Figure 4**). The size of the volume and  
234 the density of its content allowed us to sparsely sample patches without the need for  
235 deduplication and filtering.

236  
237 Compared to the Bloss pre-training dataset, CEMraw, CEMdedup and CEM500K all  
238 demonstrated significantly higher performance on four of the seven benchmarks, as shown in  
239 **Fig. 2e** (the actual IoU scores are shown in **Table 1**). The average increase in IoU scores from  
240 the Bloss baseline to CEM500K over these 4 benchmarks was 9.1%, with a maximum of 13.8%  
241 for the UroCell benchmark (increase in IoU score from 0.638 to 0.729). Tellingly, the 3  
242 benchmarks on which Bloss pre-trained models performed comparably well (Kasthuri++,  
243 Lucchi++ and Perez) were the only benchmarks that exclusively contained images from mouse  
244 brain tissue, like the Bloss dataset itself. This apparent specificity for images from the same  
245 organism and tissue type may indicate that the models learn to represent elements of the  
246 underlying biology or tissue architecture. Alternatively, it may reflect similarities in the image  
247 acquisition and sample preparation protocols, though the plausibility of this explanation is  
248 unlikely, given that each benchmark dataset was imaged with different, albeit broadly similar,  
249 technologies (Bloss with serial section TEM; Kasthuri++ with ATUM-SEM; Lucchi++ with  
250 FIB-SEM; Perez with SBF-SEM). It is clear that pre-training on large but biological narrow

251 datasets is insufficient for learning general-purpose features that apply equally well across a  
252 broad spectrum of contexts. To guard against potential biases our results instead suggest that the  
253 pre-training dataset ought to include image examples from as many different tissues, organisms,  
254 sample preparation protocols, and EM techniques as possible. Furthermore, a set of diverse  
255 benchmark datasets is essential for identifying such biases when they do arise.

256

257 *CEM500K models are largely impervious to meaningful image augmentations*

258 Having established CEM500K as the EM dataset for pre-training and transfer learning, we  
259 investigated the qualities of the model pre-trained by the MoCoV2 algorithm on CEM500K and  
260 compare it to a model pre-trained by the MoCoV2 algorithm on ImageNet (IN-moco). We note  
261 that unlike the abbreviated training used to evaluate pre-training on various subsets of CEM, here  
262 we trained the model for the complete schedule, and henceforth refer to the fully trained model  
263 as CEM500K-moco. In general, good DL models have neurons that are both robust to distortions  
264 and are selective for particular features [47]. In the context of EM images, for example, a good  
265 model must be able to recognize a mitochondrion as such irrespective of its orientation in space,  
266 its size, or some reasonable variation in resolution of its membrane. On the other hand, the same  
267 model must also be able to discern mitochondria, no matter how heterogeneous, from a variety of  
268 other organelles or cellular features. First, we attempted to evaluate the robustness of CEM500K-  
269 moco neurons by measuring their invariances to transformations of input images. Specifically,  
270 we considered the average activations of the 2,048 neurons in the last layer of the ResNet50s'  
271 encoders, pre-trained by either CEM500k-moco or IN-moco, to input images. Broadly following  
272 the approach detailed in Goodfellow et al [47] we defined invariance based on the mean firing  
273 rates of neurons in response to distortions of their inputs. Plots showing changes in mean firing  
274 rates with respect to rotation, Gaussian blur and noise, brightness, contrast and scale are shown  
275 in **Fig. 3a**. These six transforms that we choose account for much of the variation observed  
276 experimentally in cellular EM datasets, and we expect that models in which many neurons are  
277 invariant to these differences would be better suited to cellular EM segmentation tasks.

278

279 We observed that neurons in CEM500K-moco models had consistently stronger invariance to all  
280 tested transformations (**Fig. 3a**). The two exceptions were a reduction in invariance when  
281 contrast was very high and a smaller reduction when scale factors were very large (**Fig. 3a, v**

282 **and vi, respectively**). First, with regards to rotation, virtually all the neurons in the CEM500K-  
283 moco model were remarkably invariant to rotation compared to about 70% of the neurons in the  
284 IN-moco model, reflecting the fact that orientation matters for representing images in ImageNet  
285 but, appropriately, not for CEM500K. Next, neurons in the CEM500K-moco model fire more  
286 consistently when presented with increasingly blurry and noisy images, in both cases falling off  
287 significantly later as compared to IN-moco, when, presumably, meaningful information in the  
288 images has been lost. Further, while both of the tested pre-trained models responded comparably  
289 to increasing image brightness, the CEM500K-moco model had a noticeably greater invariance  
290 to both more brightened and more darkened images. For contrast adjustments, there was a similar  
291 robustness to decreased contrast. This was indicative of the distribution of images in CEM500K,  
292 and cellular EM data more broadly: very low-contrast images are common, very high-contrast  
293 images are not. On the other hand, the gap between CEM500K-moco and IN-moco pre-trained  
294 models in the high-contrast regime not only reinforce this observation but also suggest more  
295 relevant learning by the former. CEM500K-moco neurons show an invariance to a  
296 transformation only insofar as that transformation mimics real variance in the data distribution,  
297 and the firing rate decreases when the high contrast becomes no longer plausible. Similarly, there  
298 is some evidence that the results for scale invariance follow the same logic. In CEM500K, the  
299 most common reported image pixel sampling was 15-20 nm and the highest was 2 nm. Extreme  
300 scaling transformations (greater than 5x) would exceed the limits of features commonly sampled  
301 in CEM500K, rendering invariance to such transformations useless. We expect that the superior  
302 robustness to variations in cellular EM data baked into CEM500K-moco should simplify the  
303 process of adjusting to new tasks. For example, when training a U-Net on a segmentation task,  
304 the parameters in the decoder will receive a consistent signal from the pre-trained encoder  
305 regardless of the orientation and other typical variations of the input image, presumably easing  
306 the learning burden on the decoder. For the same reason, we expect models to gain robustness to  
307 rare and random events such as artifacts generated during image acquisition.

308

### 309 *CEM500K models learn biologically relevant features*

310 Next, we assessed selectivity for objects of interest, that is, do these models learn something  
311 meaningful from cellular EM images? We created feature maps by appropriately upsampling the  
312 activations of each of the 2,048 neurons in the last layer of the pre-trained ResNet50 and

313 correlated these maps to the ground truth segmentations for three different organelles. In **Fig. 3b**,  
314 activations of the 32 neurons most positively correlated with the presence of the corresponding  
315 organelle were averaged, scaled from 0-1 (displayed as a heatmap), and then binarized with a  
316 threshold of 0.3 (displayed as a binary mask). We observed that these derived heatmaps from the  
317 CEM500K-moco model shared a higher correlation with the presence of an organelle than  
318 features from the equivalent IN-moco model, irrespective of whether the organelle interrogated  
319 was ER, mitochondria, or nucleus. For the CEM500K-moco model, Point-Biserial correlation  
320 coefficients were 0.418, 0.680, and 0.888 for ER, mitochondria, and nucleus compared to 0.329,  
321 0.608, and 0.803 for the IN-moco model. The segmentations created by binarizing the mean  
322 responses also have a greater IoU with ground truth segmentations (CEM500K-moco: 0.284,  
323 0.517, and 0.887 for ER, mitochondria, and nucleus; IN-moco: 0.208, 0.325, and 0.790,  
324 respectively) for the model. Unexpectedly, features learned from ImageNet displayed some  
325 selectivity for mitochondria and nuclei, emphasizing the surprising transferability of features to  
326 domains that are seemingly unrelated to a model's training dataset. Nevertheless, it is clear that  
327 relevant pre-training, as is the case with CEM500K-moco, results in the model learning features  
328 that are meaningful in a cell biological context. The link between these results and the  
329 subsequent model's performance on downstream segmentation tasks is self-evident.

330  
331 Pre-training on CEM500K encouraged the learning of representations that encode information  
332 about organelles. We analyzed how the model completed the MoCoV2 training task of matching  
333 differently transformed views of the same image. We first generated two different views of the  
334 same image by taking random crops and then randomly rescaling them. Then we took one of the  
335 images in the pair and sequentially masked out small squares of data and measured the dot  
336 product similarity between the model's output on this occluded image and its output on the other  
337 image in the pair. Using this technique, called occlusion analysis, we were able to detect the  
338 areas in each image that were the most important for making a positive match [48]. Results are  
339 displayed as heatmaps overlaid on the occluded image (**Fig. 3c**), and show, importantly, that  
340 without any guidance, the model spontaneously learned to use organelles as "landmarks" in the  
341 images, visible as "hot spots" around such features. This behavior mirrors how a human  
342 annotator would likely approach the same problem: identify a prominent object in the first image  
343 and look for it in the second image. That these prominent objects should happen to be organelles

344 is not coincidental as sample preparation protocols for electron microscopy are explicitly  
345 designed to accentuate organelles and membranes relative to other content. Thus, representations  
346 learned by CEM500K-moco pre-training display robustness to EM-specific image variations and  
347 selectivity for objects of interest, demonstrating that they should be well-suited to any  
348 downstream segmentation tasks.

349  
350 With this understanding for how a model pre-trained with MoCoV2 on an EM-specific dataset  
351 might confer an advantage for EM segmentation tasks as compared to similar pre-training on a  
352 natural image dataset (ImageNet), we quantified this advantage by evaluating IoU improvements  
353 across the benchmark datasets. In addition to the CEM500K-moco and IN-moco pre-trained  
354 encoders we also considered two alternative parameter initializations: ImageNet Supervised (IN-  
355 super)[34] and, as a baseline, random initialization. In contrast to results in Fig. 2c, all encoder  
356 parameters for randomly initialized models were updated during training. Pre-trained models, as  
357 before, had their encoder parameters frozen to assess their transferability.

358  
359 *Fully trained CEM500K models achieve state-of-the-art results on EM benchmarks*

360 Results showing the measured percent difference in IoU scores against random initialization are  
361 shown in **Fig. 4a**. For each benchmark, we applied the number of training iterations that gave the  
362 best performance for CEM500K-moco pre-trained models and averaged the results from five  
363 independent runs (see **Table 2**). Across the board, CEM500K-moco was the best initialization  
364 method with performance increases over random initialization ranging from 0.2% on the  
365 Lucchi++ benchmark to a massive 73% on UroCell; the mean improvement (excluding CREMI  
366 Synaptic Clefts) was 30%. The baseline random initialization IoU score on the CREMI Synaptic  
367 Clefts benchmark was 0.000, making any % measurements of performance improvements  
368 meaningless. For ease of visualization, we assigned an IoU score of 0.2 for this dataset and  
369 calculated improvements based off of this score. Example 2D and 3D segmentations on the  
370 UroCell benchmark test set are shown in **Fig. 4b**; we also display representative segmentations  
371 for selected labelmaps from all of the 2D-only benchmarks in **Fig. 4c**. On the UroCell test set, all  
372 of the initialization methods except CEM500K-moco failed to accurately segment mitochondria  
373 in an anomalously bright and low-contrast region (example marked by a black arrow in Fig. 4b).  
374 Indeed, CEM500K-moco also correctly identified features that the human annotator appears to

375 have missed (example of missed mitochondrion, red arrow in Fig. 4c). On average, IN-super and  
376 IN-moco achieved 11% and 14% higher IoU scores than random initialization, respectively.  
377 Parameters pre-trained with the unsupervised MoCoV2 algorithm thus appear to transfer better to  
378 new tasks than parameters pre-trained on the ImageNet supervised classification task [34].  
379 Crucially, the 14% average increase in IoU scores from CEM500K-moco over IN-moco reveals  
380 the advantage of pre-training on a domain-specific dataset. Thus, while it is clear that some of  
381 CEM500K-moco's improvement over random initialization is explained by pre-training with the  
382 MoCoV2 algorithm in general, most of the improvement comes from the characteristics of the  
383 pre-training data.

384

385 In addition to better IoU performance, pre-trained models converged more quickly. We found  
386 that models pre-trained with the MoCoV2 algorithm converged the fastest (**Fig. 4d, top**). Within  
387 just 500 iterations, these models reach over 90% of their performance at 10,000 training  
388 iterations, and within only 100 iterations, they achieve over 80%. In some cases, 100 iterations  
389 required less than 45 seconds of training on our hardware, which included an Nvidia P100 GPU,  
390 making this approach more feasible for resource limited work. We posit that the faster training  
391 associated with the MoCoV2 algorithm stems from the much lower magnitudes of feature  
392 activations, as observed in [32], which facilitates training with higher learning rates. CEM500K-  
393 moco models trained marginally faster than IN-moco models. This speedup may have stemmed  
394 from CEM500K-moco's better robustness to the chosen data augmentations, reducing variance  
395 in the feature maps received by the trainable U-Net decoder. Overall, these results suggest a  
396 suitability of CEM500K-moco models for applications where rapid turnarounds for, say, a  
397 roughly accurate segmentation may be desired. In cases where more accurate segmentations are  
398 required, faster training as we see in **Fig. 4d** reduces the amount of time needed for  
399 hyperparameter optimization.

400

401 Finally, the plot of average IoU scores over a range of training iterations showed that the  
402 performance of randomly initialized models leveled off after 5,000 iterations, **Fig. 4d, bottom**.  
403 Previously, it has been observed that granted enough time to converge, randomly initialized  
404 models can often achieve comparable results to pre-trained models [49], and we did observe this  
405 for the easiest benchmarks (Perez, Lucchi++, and Kasthuri++, data not shown). After 30,000

406 iterations of training on these benchmarks, the performance of randomly initialized models  
407 effectively reached parity with CEM500K-moco models. However, for the hard benchmarks,  
408 randomly initialized models never reached the average IoU scores measured at even just 500  
409 training iterations for CEM500K-moco models. ImageNet pre-trained models, on the other hand,  
410 had the lowest average IoUs on easy benchmarks, but were better than random initialization for  
411 hard benchmarks. All of these observations align with expectations. Pre-trained models with  
412 frozen encoders only have  $9 \times 10^6$  parameters to fit to the data. On easy benchmarks where  
413 overfitting is not a concern, this reduction in trainable parameters hurt ImageNet pre-trained  
414 models, but not CEM500K-moco models, since the latter were already pre-trained to EM data.  
415 On hard benchmarks, the regularization effects of having fewer trainable parameters are an  
416 advantage. Randomly initialized models continued to decrease training loss on hard benchmarks,  
417 yet those gains did not translate to increases in test set IoU, a signature of overfitting (data not  
418 shown). Overfitting may be avoided by smaller models with fewer trainable parameters, similar  
419 to the pre-trained models, however this would require costly and slow additional engineering and  
420 hyperparameter optimization for each benchmark. Our results show that regardless of whether  
421 benchmarks are easy or hard, CEM500K-moco pre-trained models trained the fastest and  
422 achieved the best IoU scores. Indeed, these models outperformed the customized algorithms and  
423 training schemes presented as baselines for 4 of the benchmarks that we tested (by 3.0% on  
424 Guay, 8.6% on Kasthuri++, 1.2% on Lucchi++, and 10% on Perez), see **Table 2**. The All  
425 Mitochondria benchmark is a newly derived dataset and therefore has not been previously  
426 evaluated, but we show that it is a relatively challenging benchmark and suggest its use as a  
427 baseline for future comparisons. The remaining two benchmarks (CREMI Synaptic Clefts and  
428 UroCell) used special evaluation methods that were incompatible with our work (see Materials  
429 and Methods); instead, we present a representative visual comparison of our best results with  
430 those from the UroCell publication (**Supplementary Figure 5**) showing a marked improvement  
431 in mitochondria (blue) and lysosome (red) 3D reconstructions. While ImageNet pre-trained  
432 models are broadly useful, our results show that for some EM segmentation tasks they perform  
433 worse than random initialization. For all the available benchmarks and the newly derived All  
434 Mitochondria benchmark, CEM500K-moco pre-training uniformly performed better than the  
435 current alternatives and we demonstrate here its reliability and effectiveness for EM-specific  
436 transfer learning.

437

## 438 **Discussion**

439

440 CEM500K is a diverse, relevant, information-rich, and non-redundant dataset of unlabeled  
441 cellular EM images designed expressly to aid in the development of more robust and general DL  
442 models. Above all, two features distinguish CEM500K from other larger, publicly available EM  
443 datasets that make it superior for DL applications. First, it is derived from a far greater variety of  
444 tissue types, experimental conditions and imaging techniques, resulting in models with less bias  
445 toward such specific variables. Second, it is condensed by aggressively deleting redundant and  
446 uninformative images; this improves model performance and renders CEM500K more accessible  
447 to users. By evaluating on seven benchmarks that represent different segmentation tasks and  
448 biological contexts, we demonstrate that, on average, models pre-trained on CEM500K  
449 performed better than those pre-trained on a dataset extracted from a single large EM volume  
450 (Bloss). Remarkably, the targeted removal of 90% of the images from the original corpus of data  
451 to generate CEM500K returned a significant increase in the quality of pre-trained parameters as  
452 measured by segmentation IoU scores.

453

454 This raises the question of what the nature and extent of dataset curation should be: If a target  
455 segmentation task contains data from a particular biological context, should the pre-training  
456 dataset be curated specifically for that context? And would pre-training on the task data alone  
457 result in adequate models? Our results suggest that the benefits from curating the pre-training  
458 dataset for a particular context are minimal. Pre-training exclusively on images of mouse brain  
459 tissue (Bloss) did not improve performance over CEM500K on benchmarks from that same  
460 tissue (see **Fig. 2e**). The effect of pre-training exclusively on images from a target dataset (say,  
461 for a segmentation task) is unclear – in our case, it was impossible to fairly measure pre-training  
462 on any of the individual benchmark datasets. The MoCoV2 algorithm requires a training dataset  
463 with tens of thousands of images (65,536 in our experiments), many more than any of the  
464 benchmark datasets at our disposal. We speculate that as dataset size decreases, it becomes more  
465 likely that a model will overfit to the pre-training task and learn image features that are irrelevant  
466 for other downstream tasks [50][51]. Other unsupervised pre-training algorithms that work for



467 smaller datasets and/or larger benchmark datasets would be needed to determine the appropriate  
468 curation approach.

469

470 Regardless, we have shown here that parameters trained on CEM500K are a strong and general-  
471 purpose starting point for improving downstream segmentation models. U-Nets pre-trained on  
472 CEM500K significantly outperformed randomly initialized U-Nets on all of the segmentation  
473 benchmarks that we tested, with the largest improvements corresponding to the most difficult  
474 benchmarks. Impressively, such pre-trained models achieved state-of-the-art IoU scores on all  
475 benchmarks for which comparison with previous results was possible. The only variables tuned  
476 were the number of training iterations and data augmentations. Use of CEM500K pre-trained  
477 models by the EM community may reveal that further tuning of hyperparameters or unfreezing  
478 of the U-Net's encoder parameters could further boost performance.

479

480 Our work focused on the application of CEM500K for transfer learning. This decision was  
481 informed by the current status of DL research for cellular EM, where, typically, segmentation  
482 tasks are performed by models trained on a few labeled examples [21][15][18][16][17][42]. In  
483 general, pre-trained parameters have been shown to guide downstream models to converge to  
484 more general optima than they would from random initialization [27][52][53]. As the number of  
485 examples in the training dataset increases the generalization benefits from transfer learning start  
486 to diminish (gains in training speed are retained)[54][49]. Therefore, while unsupervised pre-  
487 training on CEM500K for transfer learning has demonstrably high utility for the common  
488 paradigm of “train on labeled ROIs / infer labels for the whole dataset”, currently it cannot solve  
489 the problem of creating general segmentation models that reliably segment features of interest  
490 for data generated by novel experiments. However, using CEM500K as seed data provides a path  
491 forward for tackling this much more difficult challenge. With  $0.5 \times 10^6$  uniquely informative  
492 images representing approximately six hundred 3D and ten thousand 2D images corresponding  
493 to more than 100 completely unrelated biological projects, CEM500K is to our knowledge the  
494 most comprehensive and diversified resource of cellular EM images. Annotating images from  
495 CEM500K (or identifying them as negative examples) will enable the creation of new task-  
496 specific training datasets with substantially more variety than previously available. Models  
497 trained on such datasets will likely be better equipped to handle data from new microscopes,

498 biological contexts, and sample preparation protocols. Moreover, each image chosen for  
499 annotation from CEM500K is likely to be uniquely informative for a model because of the  
500 extensive deduplication and filtering pipeline that we have created and used here, and which we  
501 share for future work by the community.

502  
503 The available benchmark datasets that we chose are a reflection of common applications of DL  
504 to cellular EM data, but they do not cover the full scope of possible segmentation tasks. In  
505 particular, all but one of the benchmarks involved the annotation of mitochondria and three of  
506 the seven were from mouse brain tissue. We observed that benchmark variety is essential to  
507 identify biases in pre-trained parameters and that difficult tasks are a necessary and stringent test  
508 of pre-training algorithms or datasets. For example, visual inspection of the label maps in **Fig. 4c**  
509 makes it obvious that our results leave little room for improvement on relatively easy (and 2D  
510 only) benchmarks like Lucchi++, Kasthuri++, and Perez, suggesting that going forward, new and  
511 more challenging benchmarks will be required.

512  
513 Additionally, we only tested semantic and not instance segmentation (i.e. all objects from one  
514 class share one label). We made this decision in order to avoid the more complex model  
515 architectures, postprocessing and hyperparameters that usually accompany instance segmentation  
516 [55][56][20]. Focusing on simple end-to-end semantic segmentation tasks emphasizes the effects  
517 of pre-training and eliminates the possibility that non-DL algorithms could confound the  
518 interpretation of our results. Applying pre-training for instance segmentation, an important and  
519 common task in cellular EM connectomics research, would require extension to 3D models. We  
520 chose to operate in 2D for practical reasons. 2D models work well for semantic segmentation in  
521 both 2D and 3D (our 2D models beat the state-of-the-art results set by 3D models on some of the  
522 benchmarks, see **Table 2**), whereas 3D models cannot be applied to 2D images. From a  
523 computational standpoint, 2D models have far fewer parameters than their 3D counterparts and  
524 run efficiently on a single GPU; these savings are particularly valuable for laboratories with  
525 limited access to high performance computing resources. Therefore, at this current moment, we  
526 believe that 2D pre-trained parameters are the most broadly useful for cellular EM researchers.  
527 Unsupervised pre-training on 3D data is currently an underexplored research area, although in

528 principle, there is no reason why an algorithm like MoCoV2 should not work in 3D if a  
529 sufficiently large dataset can be constructed.

530

531 The goal of this work is to begin the process of creating a data ecosystem for cellular EM images  
532 and datasets. CEM500K will be a valuable resource for experimenting with and taking advantage  
533 of the latest developments in DL research, where access to troves of image data is usually taken  
534 for granted. To further increase its utility, more data from uncommon organisms, tissue and cell  
535 types, sample preparation protocols and acquisition parameters will be needed. In the current  
536 state, the dataset is still heavily skewed to a few common organisms like mice and tissues like  
537 brain, and it is clear that there is much room for greater sampling and heterogeneity  
538 (**Supplementary Figure 6**). We hope that other researchers will consider using the curation tools  
539 that we developed in this work to contribute to CEM500K. The dataset has been made available  
540 for download on EMPIAR (ID 10592). The massive reduction in dataset size from curation  
541 makes the sharing of data relatively quick and easy; moreover, the elimination of 3D context  
542 from volume EM datasets ensures that the shared data can only reasonably be used for DL  
543 applications. Similar to pre-training on natural images, we expect that the quality of the pre-  
544 trained parameters for transfer learning will improve logarithmically as CEM500K grows [57].  
545 In the meantime, the pre-trained parameters that we release here can serve as the foundation for  
546 rapidly prototyping and building more general segmentation models for cellular EM data.

547

## 548 **Acknowledgments**

549

550 We thank the creators of the benchmarks and other datasets for making the image data freely  
551 available to the community. We thank Patrick Friday for help with running some of the models  
552 described herein, members of the CMM for contributing EM images, and FNL and NCI  
553 colleagues for critical comments on this manuscript. This work utilized the computational  
554 resources of the NIH HPC Biowulf cluster (<http://hpc.nih.gov>). This project has been funded in  
555 whole or in part with Federal funds from the National Cancer Institute, National Institutes of  
556 Health, under Contract No. 75N91019D00024. The content of this publication does not  
557 necessarily reflect the views or policies of the Department of Health and Human Services, nor

558 does mention of trade names, commercial products, or organizations imply endorsement by the  
559 U.S. Government.

560

561

## 562 **Methods**

563

### 564 Dataset standardization

565

566 Datasets generated from microscopes in our lab were already in the desired standardized format:

567 8-bit unsigned volumes or 2D tiff images. Publicly available EM data are in a variety of file

568 formats and data types; these datasets were individually reformatted as needed to match the

569 formatting of our internal datasets. Importantly, data from each of the seven benchmarks we

570 tested were included as well but comprised less than 0.1% of the dataset. To reduce the memory

571 requirements of large 3D volumes, datasets were downsampled such that no individual dataset

572 was larger than 5GB (affecting only 7 of the total 591 image volumes). The majority of 3D

573 datasets included metadata of their image resolutions; isotropic and anisotropic volumes were

574 thus automatically identified and processed differently. For all isotropic voxel data and for any

575 anisotropic voxel data in which the z resolution was less than 20% different from the x and y

576 resolutions, 2D cross-sections from the xy, xz, and yz planes were sequentially extracted.

577 Anisotropic voxel data with a greater than 20% difference in axial versus lateral resolutions were

578 only sliced into cross-sections in the xy plane. At this point, all of the gathered image data was in

579 the format of 2D tiff images, though with variable heights and widths. These images were

580 cropped into 224x224 patches without any overlap. If the image's width or height was not a

581 multiple of 224, then crops from the remaining area were discarded if either of their dimensions

582 were less than 112.

583

584 Separately, additional 2D images available through the Open Connectome Project were

585 collected. As these volumes were too large to reasonably download and store (tens of TB), Cloud

586 Volume API was used to randomly sample 1,000 2D patches from the xy planes of each

587 available dataset. These extracted patches were already of the correct size and format, therefore

588 no further processing was required. This corpus of  $5.3 \times 10^6$  2D patches constitutes "CEMraw".

589 Certain datasets were not accessible with this method and were therefore not included in the final  
590 version of CEMraw (see Supplementary Materials). The “Bloss baseline” dataset [46] was also  
591 extracted and generated with this method; however,  $1 \times 10^6$  patches were collected from that  
592 single data volume to roughly match the number of images in CEMraw (Supplementary Figure  
593 4).

594

#### 595 Deduplication

596

597 To remove duplicate patches, image hashes for all  $5.3 \times 10^6$  images in CEMraw were calculated.  
598 Difference hashes gave the best results of all the hashing algorithms tested [58]. A hash size of 8  
599 results in a 64-bit array to encode each  $224 \times 224$  image. The similarity of two images was then  
600 measured by the Hamming distance between their hashes. A pairwise comparison of all  $5.3 \times 10^6$   
601 hashes was not computationally feasible or meaningful. Instead, hashes belonging to the same  
602 2D or 3D source dataset were compared. For a 64-bit hash, distances range from 0 to 64. Sets of  
603 hashes with a distance  $< 12$  (distance cutoff chosen by visual inspection of groups) between them  
604 were considered a group of near-duplicates. All but one randomly chosen image from each group  
605 were dropped (Fig. 1b). Together, the resulting  $1.1 \times 10^6$  images constitute a deduplicated dataset  
606 or “CEMdedup”.

607

#### 608 Uninformative Patch Filtering

609

610 A random subset of 14,000 images from CEMdedup were manually labeled either informative or  
611 uninformative. The criteria for this classification process were informed by the hyperparameters  
612 of the MoCoV2 pre-training algorithm, which takes random crops as small as 20% of an area of  
613 an image. For an image that is only 20% informative, there is a 30% chance that such a randomly  
614 drawn crop will be completely uninformative, and this fraction increases exponentially for  
615 images less than 20% informative (**Supplementary Figure 7**). Therefore 20% was chosen as the  
616 cutoff for manual labeling. Concretely, this means that images with 80% or more of their area  
617 occupied by uniform intensity structures like nuclei, cytoplasm, or resin are classified as  
618 uninformative. Other criteria included whether the image was low-contrast, displayed many  
619 artifacts, or contained non-cellular objects as determined by a human annotator. A breakdown of

620 the frequency of traits present in a subset of uninformative patches is shown in **Supplementary**  
621 **Figure 1a**.

622  
623 2,000 labeled images were set aside as a test set and the remaining 12,000 were used as training  
624 data for a model classifier: a ResNet34 pre-trained on ImageNet. The fourth layer of residual  
625 blocks and the classification head of the model were fine-tuned for 30 epochs on a P100 GPU  
626 with the Adam optimizer and a learning rate of 0.001. A Random Forest classifier trained on four  
627 image-level statistics (the standard deviations of the local binary pattern [59] and image entropy,  
628 the median of the geometric mean, and the mean value of a canny edge detector [60]) was also  
629 tested. These features were chosen from a larger superset based on their measured importance.  
630 The performance for the two classifiers is shown in **Supplementary Figure 1b**. The DL model  
631 was used to create CEM500K with a confidence threshold set at 0.5.

632  
633 Momentum Contrast Pre-training

634  
635 For unsupervised pre-training, the Momentum Contrast (MoCoV2) algorithm [31, 32] was used.  
636 A schematic of a single step in the algorithm is shown in Supplementary Figure 3a. Pre-training  
637 was completed on a machine with 4 Nvidia V100 GPUs using a batch size of 128 and queue  
638 length of 65,536. The initial learning rate was set to 0.015 and divided by 10 at epochs 120 and  
639 160. In addition, 360° rotations and Gaussian noise with a standard deviation range of  $1 \times 10^{-5}$  to  
640  $1 \times 10^{-4}$  were added to the data augmentations. All other hyperparameters and data augmentations  
641 were left as the defaults presented in [32]. For pre-training comparisons between different EM  
642 datasets, i.e. the three subsets of CEM plus Bloss (**Fig. 2d, e**),  $4.5 \times 10^5$  total parameter updates  
643 (iterations) were run for each model, which is equivalent to 120 passes (epochs) through all the  
644 images in CEM500K. The average training time for each of these models was 2.5 days. The final  
645 pre-trained parameters generated for results shown in **Fig. 4b, c** were trained on CEM500K for  
646 an additional 80 epochs: a total of 200 epochs and 4 days of training.

647  
648 U-Net Segmentation Architecture

649

650 Our implementation was similar to the original implementation of the U-Net, except that the  
651 encoder was replaced with a ResNet50 model (**Supplementary Figure 3b**). When using pre-  
652 trained models in these experiments all parameters in the encoder were frozen such that no  
653 updates were made during training. Randomly initialized encoders were tested with both frozen  
654 and unfrozen parameters. The random number generator seed was fixed at 42 such that any  
655 randomly initialized parameters in either the U-Net encoder or decoder would be the same in  
656 every experiment.

657

### 658 Benchmark Segmentation Tasks

659

660 The One Cycle Policy and AdamW optimizer with maximum learning rate 0.003, weight decay  
661 0.1, batch size 16, and (binary) cross entropy loss were used for all benchmarks [61][62]. For the  
662 Guay and Urocell benchmarks, which required multiclass segmentation, the cross-entropy loss  
663 was weighted by the prevalence of each class; this yielded better IoU scores. Classes that  
664 accounted for less than 10% of all pixels in the dataset were given a weight of 3, those that  
665 accounted for more than 10% were given a weight of 1, and all background classes were given a  
666 weight of 0.1. Data augmentations included randomly resized crops with scaling from 0.08 to 1  
667 and aspect ratio from 0.5 to 1.5, 360° rotations, random 30% brightness and contrast  
668 adjustments, and horizontal and vertical flips. For the Guay benchmark, and consequently the All  
669 Mitochondria benchmark, Gaussian Noise with a variance limit of 400 to 1200 and Gaussian  
670 Blur with a maximum standard deviation of 7 were also added. The decision to add more data  
671 augmentations for these benchmarks was made in response to observed overfitting on the Guay  
672 benchmark validation dataset. Lastly, different crop sizes were used for each benchmark: 512 x  
673 512 for Guay, CREMI, Synaptic Cleft, Kasthuri++ and Lucchi++, 480 x 480 for Perez, and 224 x  
674 224 for UroCell and All Mitochondria.

675

676 To create 3D segmentations for the UroCell, Guay, and CREMI Synaptic Cleft test sets we used  
677 either orthoplane or 2D stack inference following [63]. Briefly, in 2D stack inference the model  
678 only makes predictions on xy cross-sections; in orthoplane inference, the model makes  
679 predictions on xy, yz, and xz cross-sections and the confidence scores are averaged together.  
680 Orthoplane inference was used for the UroCell test set because its test volume has isotropic

681 voxels. Because both the Guay and CREMI Synaptic Cleft test volumes are anisotropic we used  
682 2D stack inference instead.

683

684 Evaluation generally followed the details given in the publication that accompanied the  
685 benchmark. First, test images in the Perez datasets did not have labels for all instances of an  
686 object e.g. only 1 nucleus was labeled in an image containing 2 nuclei. To circumvent this  
687 problem, we ignored areas in the predicted segmentations that did not coincide with a labeled  
688 instance in the ground truth. Second, the UroCell benchmark was evaluated in previous work by  
689 averaging K-Fold cross-validation results on 5 unique splits of the 5 training volumes such that  
690 each training volume was used as the test set once. The authors also excluded pixels on the  
691 boundary of object instances both when training and when calculating the prediction's IoU with  
692 ground truth. Here, a simpler evaluation was run on a single split of the data with 4 volumes used  
693 for training and 1 volume used for testing. To eliminate small regions of missing data we  
694 cropped 2 of the 5 volumes along the y axis (fib1-0-0-0.nii.gz, the test volume, by 12 pixels and  
695 fib1-1-0-3.nii.gz by 54 pixels). Third, for the CREMI Synaptic Cleft benchmark the training and  
696 test datasets did not have an official evaluation metric, and the ground truth segmentations were  
697 not publicly available. Therefore, volumes A and B were used exclusively for training and IoU  
698 scores were evaluated on volume C.

699

700 Mean Firing Rate

701

702 Following [47] neuron firing thresholds were determined by passing 1,000 images of randomly  
703 sampled noise through each pre-trained ResNet50 model and calculating the 99<sup>th</sup> percentile of  
704 responses. In our experiments, only the neurons in the output of the global average pooling layer  
705 were considered such that there were 2,048. Responses to 100 randomly selected images from  
706 CEM500K were then recorded over a range of distortion strengths. For each neuron, the set of  
707 undistorted images that activated the neuron near maximally (over the 90<sup>th</sup> percentile), called Z,  
708 was determined. A set containing versions of all images in Z with a particular distortion applied  
709 is called Z'. Any neuron that responded to images in Z less strongly than the neuron's firing  
710 threshold were ignored as they are not selective for features observed in the test images.  
711 However, for all remaining neurons, the firing rate at a particular distortion strength is calculated



712 as the number of images in  $Z'$  that activate the neuron over its firing threshold divided by the  
713 number of images in  $Z$ . The mean firing rate to a particular distortion is then the average of firing  
714 rates for any of the 2,048 neurons that were selective enough to be considered.

715

716

717 Feature selectivity

718

719 To measure feature selectivity, we first manually segmented 3 organelles (ER, mitochondria,  
720 nucleus) in 3 images. By construction, the ResNet50 architecture downsamples an input image  
721 by 32. For thin and small organelles like ER, the final feature maps were too coarse to accurately  
722 show the localization of responses. Therefore, we eliminated the last 4 downsampling operations  
723 such that the output feature map was only 2x smaller than the input. Following similar logic, we  
724 eliminated the last 2 downsampling operations for mitochondria and the last downsampling  
725 operation for nuclei -- 8x and 16x smaller than the input images, respectively. For all organelles,  
726 these differently downsampled feature maps were resized to match the dimensions of the input  
727 image (224x224) and then each feature map was compared against the ground truth labelmap by  
728 Point Biserial correlation. A simple average of the 32 most correlated feature maps was then  
729 overlaid on the original image as the mean response. Drawing a threshold at 0.3 yielded the  
730 binary segmentations.

731

732 Occlusion Analysis

733

734 Typically, occlusion analysis measures the importance of regions in an image to a classification  
735 task [48]. In our experiments, importance was measured as a function of the dot product  
736 similarity between the feature vectors output by the global average pooling layer of a ResNet50  
737 for an image and its occluded copy. Sequential regions of 61x61 pixels spaced every 30 pixels  
738 (in both x and y dimensions) were zeroed out in each image. Region importance to the similarity  
739 measurement was then normalized to fall in the range 0 to 1 and overlaid on the original image.

740

741

742 **References**

- 743 [1] S. Y. Takemura *et al.*, “Synaptic circuits and their variations within different columns in  
744 the visual system of *Drosophila*,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 112, no. 44, pp.  
745 13711–13716, Nov. 2015.
- 746 [2] N. Kasthuri *et al.*, “Saturated Reconstruction of a Volume of Neocortex,” *Cell*, vol. 162,  
747 no. 3, pp. 648–661, Aug. 2015.
- 748 [3] A. E. Vincent, D. M. Turnbull, V. Eisner, G. Hajnóczky, and M. Picard, “Mitochondrial  
749 Nanotunnels,” *Trends Cell Biol.*, vol. 27, no. 11, pp. 787–799, Nov. 2017.
- 750 [4] A. E. Vincent, K. White, T. Davey, R. W. Taylor, D. M. Turnbull, and M. Picard,  
751 “Quantitative 3D Mapping of the Human Skeletal Muscle Mitochondrial Network,”  
752 *CellReports*, vol. 26, pp. 996-1009.e4, 2019.
- 753 [5] D. P. Hoffman *et al.*, “Correlative three-dimensional super-resolution and block-face  
754 electron microscopy of whole vitreously frozen cells,” *Science (80-. )*, vol. 367, no. 6475,  
755 Jan. 2020.
- 756 [6] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh,  
757 “CSPNet: A new backbone that can enhance learning capability of CNN,” in *IEEE*  
758 *Computer Society Conference on Computer Vision and Pattern Recognition Workshops*,  
759 2020, vol. 2020-June, pp. 1571–1580.
- 760 [7] A. Tao, K. Sapra, and B. Catanzaro, “Hierarchical Multi-Scale Attention for Semantic  
761 Segmentation,” *arXiv2005.10821 [cs]*, May 2020.
- 762 [8] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-  
763 End Object Detection with Transformers,” *arXiv2005.12872 [cs]*, May 2020.
- 764 [9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” *IEEE Trans. Pattern*  
765 *Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.
- 766 [10] J. W. Lichtman, H. Pfister, and N. Shavit, “The big data challenges of connectomics,”  
767 *Nat. Neurosci.*, vol. 17, no. 11, pp. 1448–1454, Oct. 2014.
- 768 [11] S. M. Plaza and J. Funke, “Analyzing Image Segmentation for Connectomics,” *Front.*  
769 *Neural Circuits*, vol. 12, p. 102, Nov. 2018.
- 770 [12] A. Goodfellow, Ian; Bengio, Yoshua; Courville, *Deep Learning*. MIT Press, 2016.
- 771 [13] F. Pereira, P. Norvig, and A. Halev, “The Unreasonable Effectiveness of Data,” *IEEE*  
772 *Intell. Syst.*, 2009.

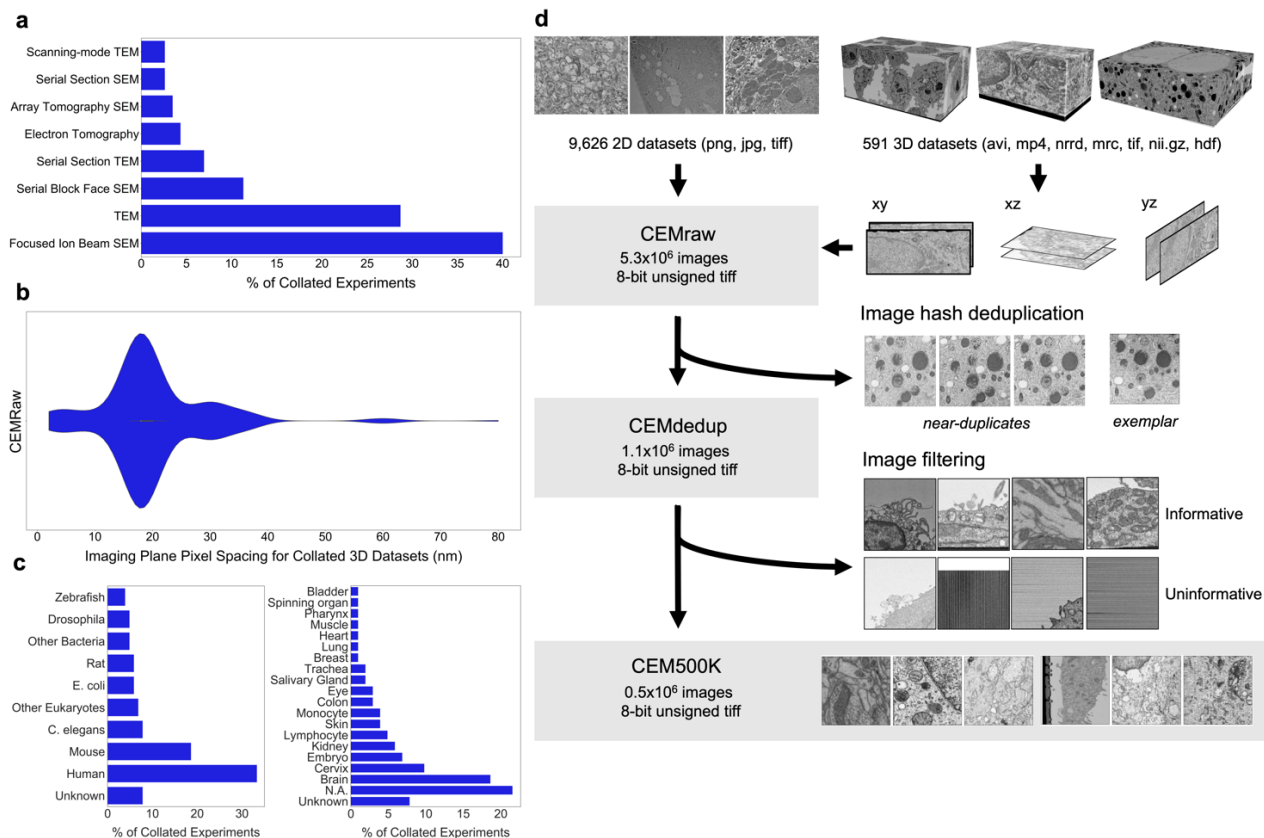
- 773 [14] C. Sun, A. Shrivastava, S. Singh, and A. Gupta, “Revisiting Unreasonable Effectiveness  
774 of Data in Deep Learning Era,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-October, pp.  
775 843–852, Jul. 2017.
- 776 [15] M. Guay, Z. Emam, A. Anderson, M. Aronova, and R. Leapman, “Dense cellular  
777 segmentation using 2D-3D neural network ensembles for electron microscopy,” *bioRxiv*  
778 *2020.01.05.895003*, 2020.
- 779 [16] M. Žerovnik Mekuč *et al.*, “Automatic segmentation of mitochondria and endolysosomes  
780 in volumetric electron microscopy data,” *Comput. Biol. Med.*, vol. 119, p. 103693, 2020.
- 781 [17] V. Casser, K. Kang, H. Pfister, and D. Haehn, “Fast Mitochondria Segmentation for  
782 Connectomics,” *arXiv1812.06024 [cs]*, Dec. 2018.
- 783 [18] A. J. Perez *et al.*, “A workflow for the automatic segmentation of organelles in electron  
784 microscopy image stacks,” *Front. Neuroanat.*, vol. 8, no. November, p. 126, Nov. 2014.
- 785 [19] M. Berning, K. M. Boergens, and M. Helmstaedter, “SegEM: Efficient Image Analysis for  
786 High-Resolution Connectomics,” *Neuron*, vol. 87, pp. 1193–1206, 2015.
- 787 [20] M. Januszewski *et al.*, “High-precision automated reconstruction of neurons with flood-  
788 filling networks,” *Nat. Methods*, vol. 15, no. 8, pp. 605–610, 2018.
- 789 [21] J. Funke *et al.*, “Large Scale Image Segmentation with Structured Loss Based Deep  
790 Learning for Connectome Reconstruction,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.  
791 41, no. 7, pp. 1669–1680, Jul. 2019.
- 792 [22] J. Buhmann *et al.*, “Automatic Detection of Synaptic Partners in a Whole-Brain  
793 Drosophila EM Dataset,” *bioRxiv*, p. 2019.12.12.874172, Mar. 2019.
- 794 [23] H. Spiers *et al.*, “Citizen science, cells and CNNs – deep learning for automatic  
795 segmentation of the nuclear envelope in electron microscopy data, trained with volunteer  
796 segmentations,” *bioRxiv*, p. 2020.07.28.223024, Jul. 2020.
- 797 [24] J. S. Kim *et al.*, “Space-time wiring specificity supports direction selectivity in the retina,”  
798 *Nature*, vol. 509, no. 7500, pp. 331–336, May 2014.
- 799 [25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale  
800 Hierarchical Image Database,” *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- 801 [26] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object  
802 Detection with Region Proposal Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol.  
803 39, no. 6, pp. 1137–1149, Jun. 2017.

- 804 [27] M. Huh, P. Agrawal, and A. A. Efros, “What makes ImageNet good for transfer  
805 learning?,” *arXiv1608.08614 [cs]*, 2016.
- 806 [28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep  
807 Bidirectional Transformers for Language Understanding,” *NAACL HLT 2019 - 2019 Conf.  
808 North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, vol. 1,  
809 pp. 4171–4186, Oct. 2018.
- 810 [29] C. Karabağ, M. L. Jones, C. J. Peddie, A. E. Weston, L. M. Collinson, and C. C. Reyes-  
811 Aldasoro, “Semantic segmentation of HeLa cells: An objective comparison between one  
812 traditional algorithm and four deep-learning architectures,” *PLoS One*, vol. 15, no. 10, p.  
813 e0230605, Oct. 2020.
- 814 [30] K. S. Devan, P. Walther, J. von Einem, T. Ropinski, H. A. Kestler, and C. Read,  
815 “Detection of herpesvirus capsids in transmission electron microscopy images using  
816 transfer learning,” *Histochem. Cell Biol.*, vol. 151, no. 2, pp. 101–114, Feb. 2019.
- 817 [31] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, “Transfusion: Understanding transfer  
818 learning for medical imaging,” in *Advances in Neural Information Processing Systems*,  
819 2019, vol. 32.
- 820 [32] Y. Tian, D. Krishnan, and P. Isola, “Contrastive Multiview Coding,” *arXiv1906.05849*  
821 *[cs]*, Jun. 2019.
- 822 [33] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive  
823 Learning of Visual Representations,” *arXiv2002.05709 [cs]*, 2020.
- 824 [34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum Contrast for Unsupervised  
825 Visual Representation Learning,” *arXiv1911.05722 [cs]*, Nov. 2019.
- 826 [35] J. Donahue and K. Simonyan, “Large Scale Adversarial Representation Learning,”  
827 *arXiv1907.02544 [cs]*, Jul. 2019.
- 828 [36] X. Ji, J. F. Henriques, and A. Vedaldi, “Invariant Information Clustering for Unsupervised  
829 Image Classification and Segmentation,” *arxiv1807.06653 [cs]*, Jul. 2018.
- 830 [37] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised Feature Learning via Non-  
831 Parametric Instance Discrimination,” *arxiv1805.01978 [cs]*, 2018.
- 832 [38] A. Kolesnikov *et al.*, “Large Scale Learning of General Visual Representations for  
833 Transfer,” *arxiv1912.11370 [cs]*, Dec. 2019.
- 834 [39] X. Chen, H. Fan, R. Girshick, and K. He, “Improved Baselines with Momentum

- 835 Contrastive Learning,” *arxiv2003.04297 [cs]*, 2020.
- 836 [40] J. T. Vogelstein *et al.*, “A community-developed open-source computational ecosystem  
837 for big neuro data,” *Nat. Methods*, vol. 15, no. 11, pp. 846–847, Nov. 2018.
- 838 [41] A. Iudin, P. K. Korir, J. Salavert-Torres, G. J. Kleywegt, and A. Patwardhan, “EMPIAR:  
839 A public archive for raw electron microscopy image data,” *Nat. Methods*, vol. 13, no. 5,  
840 pp. 387–388, May 2016.
- 841 [42] “CREMI,” *Miccai Challenge on Circuit Reconstruction From Electron Microscopy*  
842 *Images (CREMI)*, 2016. [Online]. Available: <https://cremi.org/>. [Accessed: 27-Oct-2020].
- 843 [43] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in  
844 *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern*  
845 *Recognition*, 2016, vol. 2016-Decem, pp. 770–778.
- 846 [44] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical  
847 image segmentation,” in *Lecture Notes in Computer Science (including subseries Lecture*  
848 *Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2015, vol. 9351, pp.  
849 234–241.
- 850 [45] C. Ju, A. Bibaut, and M. J. Van Der Laan, “The Relative Performance of Ensemble  
851 Methods with Deep Convolutional Neural Networks for Image Classification,”  
852 *arxiv1704.01664 [cs]*, 2017.
- 853 [46] E. B. Bloss, M. S. Cembrowski, B. Karsh, J. Colonell, R. D. Fetter, and N. Spruston,  
854 “Single excitatory axons form clustered synapses onto CA1 pyramidal cell dendrites,”  
855 *Nat. Neurosci.*, vol. 21, no. 3, pp. 353–363, Mar. 2018.
- 856 [47] I. J. Goodfellow, Q. V Le, A. M. Saxe, H. Lee, and A. Y. Ng, “Measuring Invariances in  
857 Deep Networks,” in *Advances in Neural Information Processing Systems*, 2009, pp. 646–  
858 654.
- 859 [48] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in  
860 *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial*  
861 *Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8689 LNCS, no. PART 1,  
862 pp. 818–833.
- 863 [49] K. He, R. Girshick, and P. Dollár, “Rethinking ImageNet Pre-training,” *Proc. IEEE Int.*  
864 *Conf. Comput. Vis.*, vol. 2019-October, pp. 4917–4926, Nov. 2018.
- 865 [50] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, “What makes for good

- 866 views for contrastive learning,” *arxiv2005.10243 [cs]*, May 2020.
- 867 [51] M. Minderer, O. Bachem, N. Houlsby, and M. Tschannen, “Automatic Shortcut Removal  
868 for Self-Supervised Representation Learning,” *arxiv2002.08822 [cs]*, 2020.
- 869 [52] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, “How transferable are features in deep  
870 neural networks?,” in *Advances in Neural Information Processing Systems*, 2014, pp.  
871 3320–3328.
- 872 [53] B. Neyshabur, H. Sedghi, and C. Zhang, “What is being transferred in transfer learning?,”  
873 *arxiv2008.11687 [cs]*, 2020.
- 874 [54] B. Zoph *et al.*, “Rethinking Pre-training and Self-training,” *arxiv2006.06882 [cs]*, Jun.  
875 2020.
- 876 [55] L. Heinrich, J. Funke, C. Pape, J. Nunez-Iglesias, and S. Saalfeld, “Synaptic Cleft  
877 Segmentation in Non-Isotropic Volume Electron Microscopy of the Complete *Drosophila*  
878 Brain,” *arxiv1805.02718 [cs]*, 2018.
- 879 [56] J. Funke *et al.*, “Large Scale Image Segmentation with Structured Loss based Deep  
880 Learning for Connectome Reconstruction,” *arXiv1709.02974 [cs]*, 2020.
- 881 [57] D. Mahajan *et al.*, “Exploring the Limits of Weakly Supervised Pretraining,”  
882 *arXiv1805.00932 [cs]*, 2018.
- 883 [58] “Kind of Like That,” *The Hacker Factor Blog*, 2013. [Online]. Available:  
884 <http://www.hackerfactor.com/blog/index.php?/archives/529-Kind-of-Like-That.html>.  
885 [Accessed: 28-Oct-2020].
- 886 [59] T. Ojala, M. Pietikäinen, and T. Mäenpää, “Multiresolution gray-scale and rotation  
887 invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal.*  
888 *Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
- 889 [60] J. Canny, “A Computational Approach to Edge Detection,” *IEEE Trans. Pattern Anal.*  
890 *Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- 891 [61] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” *7th Int. Conf.*  
892 *Learn. Represent. ICLR 2019*, Nov. 2017.
- 893 [62] L. N. Smith, “A disciplined approach to neural network hyper-parameters: Part 1 --  
894 learning rate, batch size, momentum, and weight decay,” *arxiv1803.09820 [cs]*, Mar.  
895 2018.
- 896 [63] R. Conrad, H. Lee, and K. Narayan, “Enforcing Prediction Consistency Across

897           Orthogonal Planes Significantly Improves Segmentation of FIB-SEM Image Volumes by  
898           2D Neural Networks.,” *Microsc. Microanal.*, pp. 1–4, Jul. 2020.  
899  
900



901

902

903 **Figure 1: Preparation of a deep learning appropriate 2D EM image dataset rich with**

904 **relevant and unique features.** (a) Percent distribution of collated experiments grouped by

905 imaging technique TEM, transmission EM; SEM, scanning EM. (b) Distribution of imaging

906 plane pixel spacings in nm for volumes in the 3D corpus. (c) Percent distribution of collated

907 experiments by organism and tissue origin. (d) Schematic of our workflow: 2D EM image stacks

908 (top left) or 3D EM image volumes sliced into 2D cross-sections (top right) were cropped into

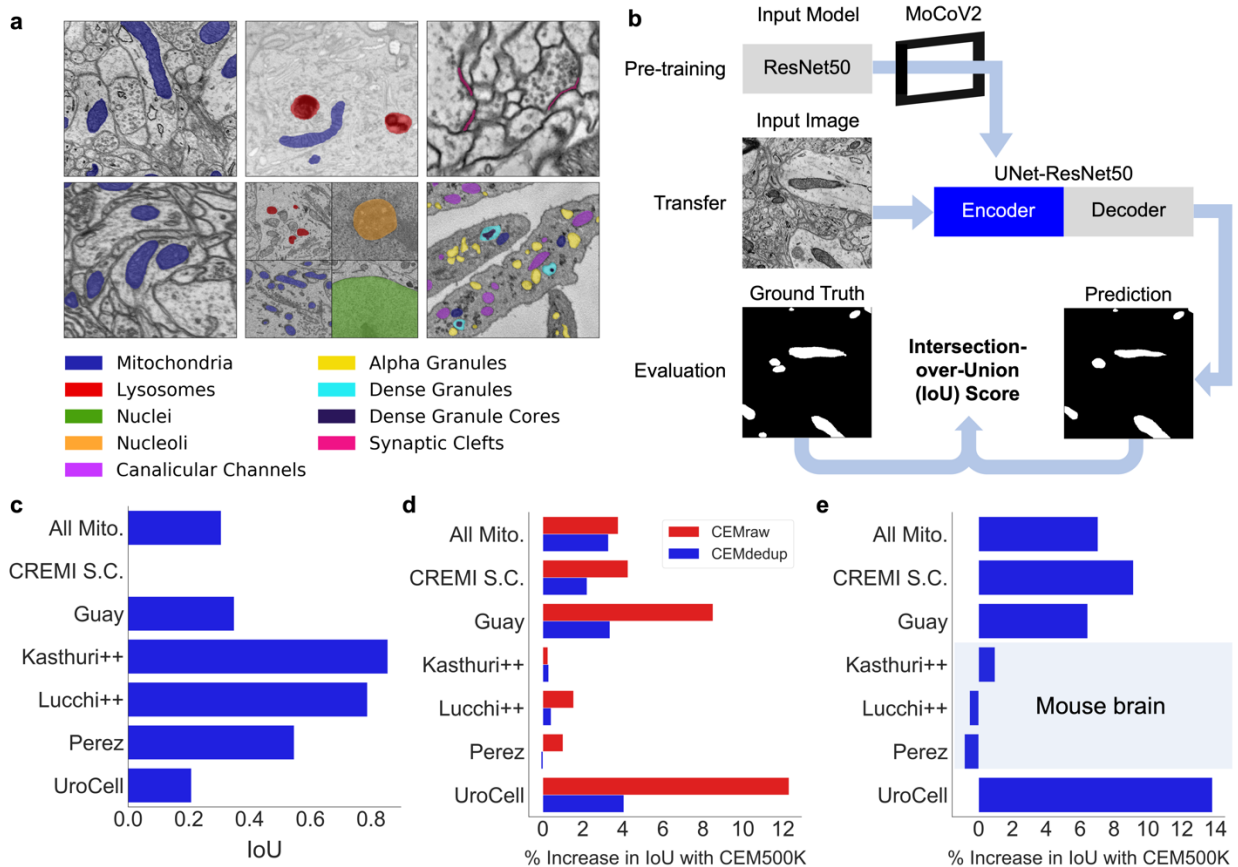
909 patches of 224 x 224 pixels, comprising CEMraw. Nearly identical patches excepting a single

910 exemplar were eliminated to generate CEMdedup. Uninformative patches were culled to form

911 CEM500K.

912



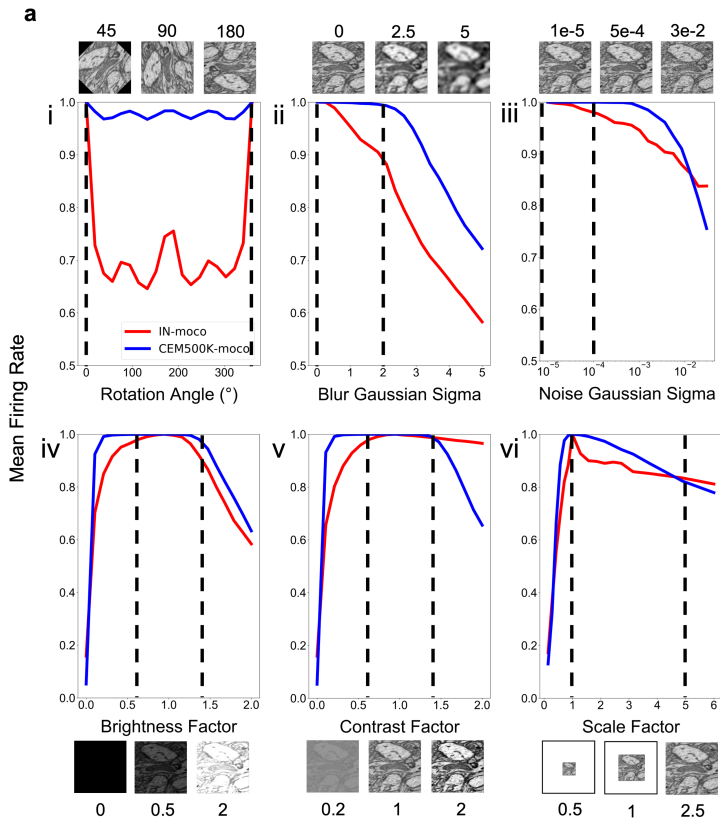


913

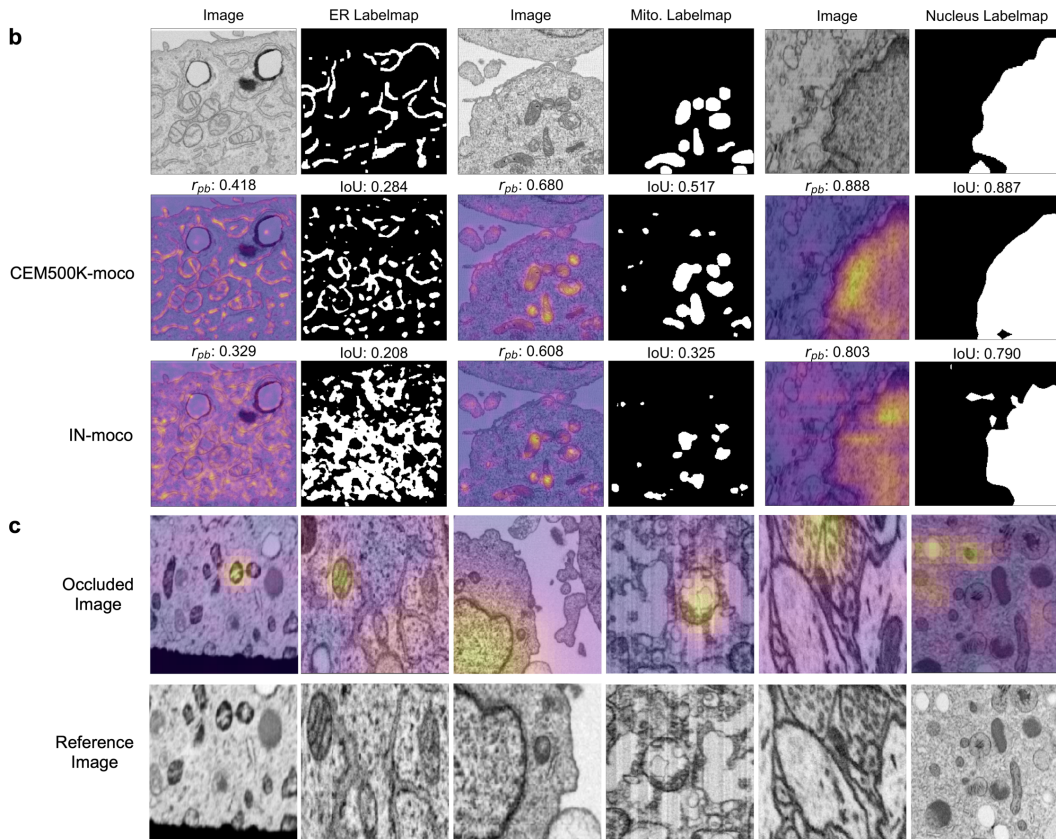
914 **Figure 2: CEM500K pre-training improves the transferability of learned features.**

915 (a) Example images and colored label maps from each of the six publicly available benchmark  
 916 datasets: clockwise from top left: Kasthuri++, UroCell, CREMI Synaptic Clefts, Guay, Perez,  
 917 and Lucchi++. The All Mitochondria benchmark is a superset of these benchmarks and is not  
 918 depicted. (b) Schematic of our pre-training, transfer and evaluation workflow. Gray blocks  
 919 denote trainable models with randomly initialized parameters; blue block denotes a model with  
 920 frozen pre-trained parameters. (c) Baseline IoU scores for each benchmark achieved by skipping  
 921 MoCoV2 pre-training. Randomly initialized parameters in ResNet50 layers were transferred  
 922 directly to UNet-ResNet50 and frozen during training. (d) Measured percent difference in IoU  
 923 scores between models pre-trained on CEMraw vs CEM500K (red) and on CEMdedup vs  
 924 CEM500K (blue). (e) Measured percent difference in IoU scores between a model pre-trained on  
 925 CEM500K over the mouse brain (Bloss) pre-training dataset. Benchmark datasets comprised  
 926 exclusively of EM images of mouse brain tissue are highlighted.

927



928



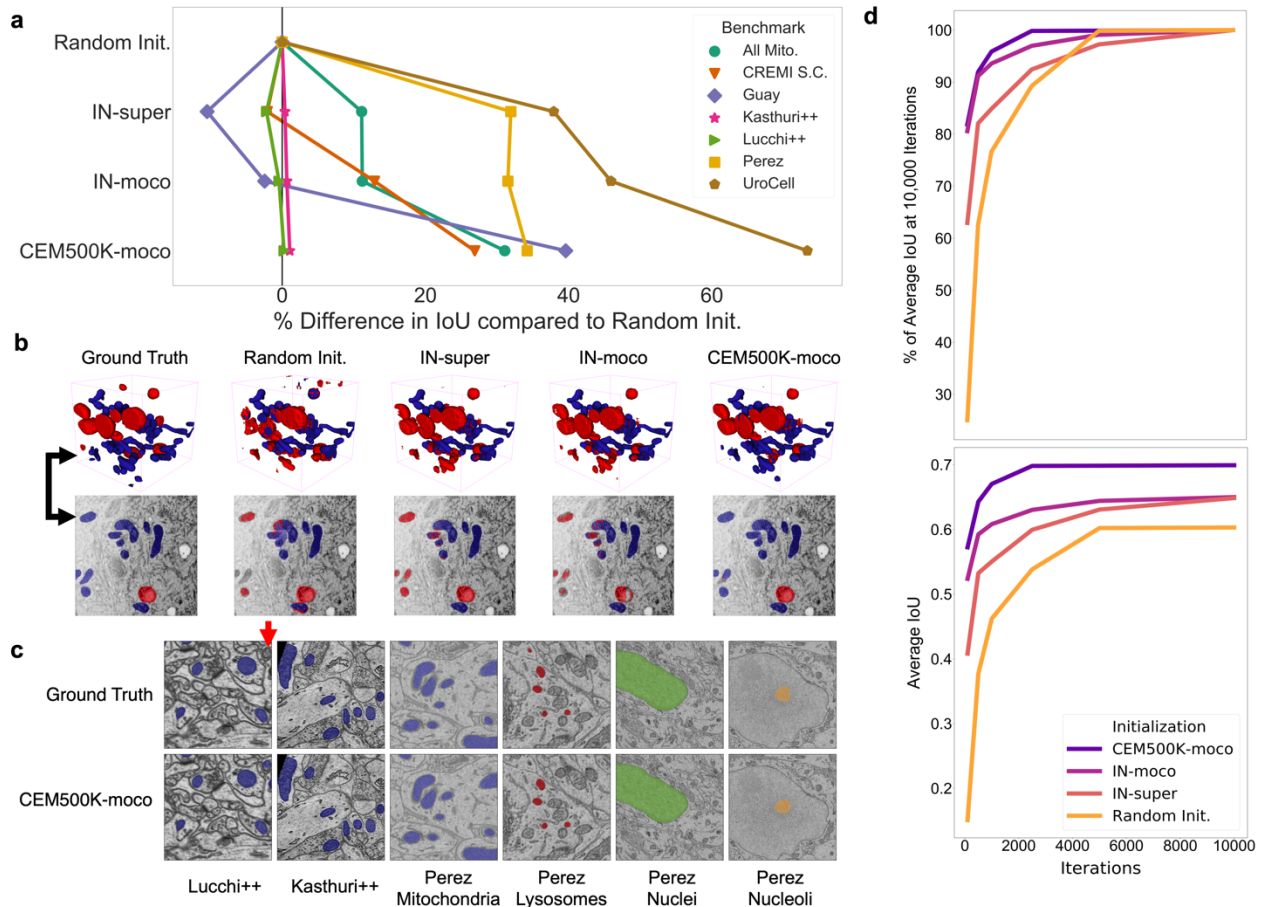
929

930

931 **Figure 3: Features learned from CEM500K pre-training are more robust to image**  
932 **transformations and encode for semantically meaningful objects with greater selectivity.** (a)

933 Mean firing rates calculated between feature vectors of images distorted by i. Rotation, ii.  
934 Gaussian blur, iii. Gaussian noise, iv. Brightness, v. Contrast, vi. Scale. Dashed black lines show  
935 the range of augmentations used for CEM500K + MoCoV2 during pre-training. For transforms  
936 in the top row, the undistorted images occur at  $x=0$ ; bottom row, at  $x=1$ . (b) Evaluation of  
937 features corresponding to ER (left), mitochondria (middle) and nucleus (right). For each  
938 organelle, the panels show: input image and ground truth label map (top row), heatmap of  
939 CEM500K-moco activations of the 32 filters most correlated with the organelle and CEM500K-  
940 moco binary mask created by thresholding the mean response at 0.3 (middle row), IN-moco  
941 activations and IN-moco binary mask (bottom row). Also included are Point-Biserial correlation  
942 coefficients ( $r_{pb}$ ) values and IoUs for each response and segmentation. All feature responses are  
943 rescaled to range  $[0, 1]$ . (c) Heatmap of occlusion analysis showing the region in each occluded  
944 image most important for forming a match with a corresponding reference image. All  
945 magnitudes are rescaled to range  $[0, 1]$ .

946



947

948 **Figure 4: Models pre-trained on CEM500K yield superior segmentation quality and**

949 **training speed on all segmentation benchmarks.** (a) Plot of percent difference in segmentation

950 performance between pre-trained models and a randomly initialized model. (b) Example

951 segmentations on the UroCell benchmark in 3D (top) and 2D (bottom). The black arrows shows

952 the location of the same mitochondrion in 2D and in 3D. (c) Example segmentations from all

953 2D-only benchmark datasets. The red arrow marks a false negative in ground truth segmentation

954 detected by the CEM500K-moco pre-trained model. (d) Top, average IoU scores as a percent of

955 the average IoU after 10,000 training iterations (ii); bottom, absolute average IoU scores over a

956 range of training iteration lengths.

957

Benchmark	Random Init. (No Pretraining)	Bloss et al. 2018	CEMraw	CEMdedup	CEM500K
All Mitochondria	0.306	0.694	0.719	0.722	<b><u>0.745</u></b>
CREMI Synaptic Clefts	0.000	0.242	0.254	0.259	<b><u>0.265</u></b>
Guay	0.349	0.380	0.372	0.391	<b><u>0.404</u></b>
*Kasthuri++	0.855	0.907	0.913	0.913	<b><u>0.915</u></b>
*Lucchi++	0.788	<b><u>0.899</u></b>	0.880	0.890	0.894
*Perez	0.547	<b><u>0.874</u></b>	0.854	0.866	0.869
UroCell	0.208	0.638	0.652	0.699	<b><u>0.729</u></b>
<b>*Average Mouse Brain</b>	0.730	<b><u>0.893</u></b>	0.883	0.890	<b><u>0.893</u></b>
<b>Average Other</b>	0.216	0.489	0.499	0.518	<b><u>0.536</u></b>

958  
959 **Table 1:** Comparison of segmentation IoU results for benchmark datasets from models randomly  
960 initialized and pre-trained with MoCoV2 on the Bloss dataset, and CEMraw, CEMdedup and  
961 CEM500K. \* denotes benchmarks that exclusively contain EM images from mouse brain tissue.  
962 The best result for each benchmark is highlighted in bold and underlined.

963

Benchmark	Training Iterations	Random Init.	IN-super	IN-moco	CEM500K-moco	Reported
All Mitochondria	10000	0.587	0.653	0.653	<b><u>0.770</u></b>	--
CREMI Synaptic Clefts	5000	0.000	0.196	0.226	<b><u>0.254</u></b>	--
Guay [15]	1000	0.308	0.275	0.300	<b><u>0.429</u></b>	0.417
Kasthuri++ [17]	10000	0.905	0.908	0.911	<b><u>0.915</u></b>	0.845
Lucchi++ [17]	10000	0.894	0.865	0.892	<b><u>0.895</u></b>	0.888
Perez [18]	2500	0.672	0.886	0.883	<b><u>0.901</u></b>	0.821
Lysosomes	--	0.842	0.838	0.816	<b><u>0.849</u></b>	0.726
Mitochondria	--	0.130	0.860	0.866	<b><u>0.884</u></b>	0.780
Nuclei	--	0.984	0.987	0.986	<b><u>0.988</u></b>	0.942
Nucleoli	--	0.731	0.859	0.865	<b><u>0.885</u></b>	0.835
UroCell	2500	0.424	0.584	0.618	<b><u>0.734</u></b>	--

964  
965 **Table 2:** Comparison of segmentation IoU scores for different weight initialization methods  
966 versus the best results on each benchmark as reported in the publication presenting the  
967 segmentation task. All IoU scores are the average of five independent runs.

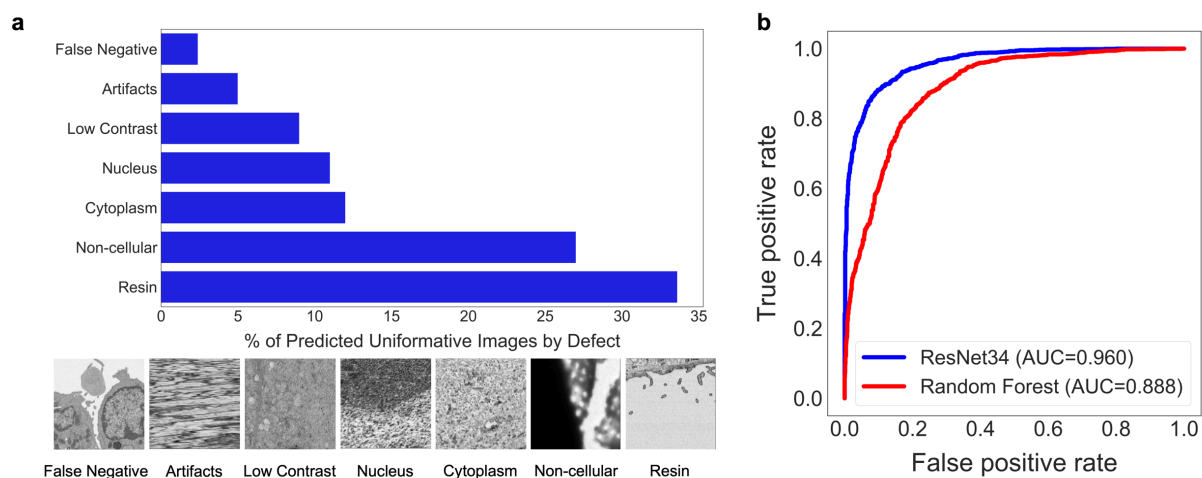
968

Benchmark	Biological Context(s)	Imaging Technique(s)	Voxel Size (nm)	Dimensionality	Training Set Pixels/Voxels	Test Set Pixels/Voxels	Segmentation Class(es)
All Mitochondria	Mouse Bladder, Mouse Brain & Human Platelets	SBF-SEM, FIB-SEM, ssSEM	10x10x50, 5x5x5, 3x3x29, 30x30x30, 16x16x15	2D and 3D	4.42E+08	3.71E+08	Mitochondria
CREMI Synaptic Clefts	Drosophila Brain	ssTEM	4x4x40	3D	3.91E+08	1.95E+08	Synaptic Clefts
Guay	Human Platelets	SBF-SEM	10x10x50	3D	3.20E+07	2.95E+07	Mitochondria, Canalicular Channels, Alpha Granules, Dense Granules, Dense Granule Cores
Lucchi++	Mouse Brain	FIB-SEM	5x5x5	2D	1.30E+08	1.30E+08	Mitochondria
Kasthuri++	Mouse Brain	ssSEM	3x3x29	2D	2.01E+08	1.55E+08	Mitochondria
Perez	Mouse Brain	SBF-SEM	30x30x30	2D	1.25E+07	4.00E+07	Mitochondria, Lysosomes, Nuclei, Nucleoli
UroCell	Mouse Bladder	FIB-SEM	16x16x15	3D	6.71E+07	1.68E+07	Mitochondria, Lysosomes

969

970 **Supplementary Table 1: Characteristics of the benchmark datasets.**

971



972

973 **Supplementary Figure 1: Deduplication and image filtering.** (a) Breakdown of fractions(top)

974 and representative examples (bottom) of patches labeled “uninformative” by a trained DL model

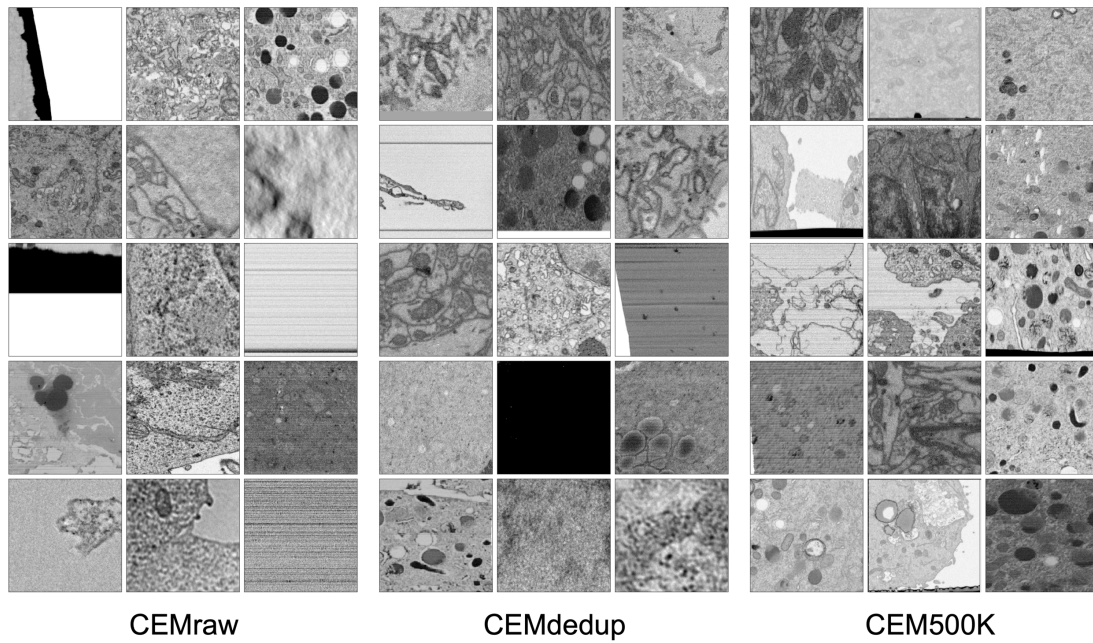
975 based on defect (as determined by a human annotator) (b) Receiver operating characteristic curve

976 for the DL model classifier and a Random Forest classifier evaluated on a holdout test set of

977 2,000 manually labeled patches (1,000 informative and 1,000 uninformative).

978

979



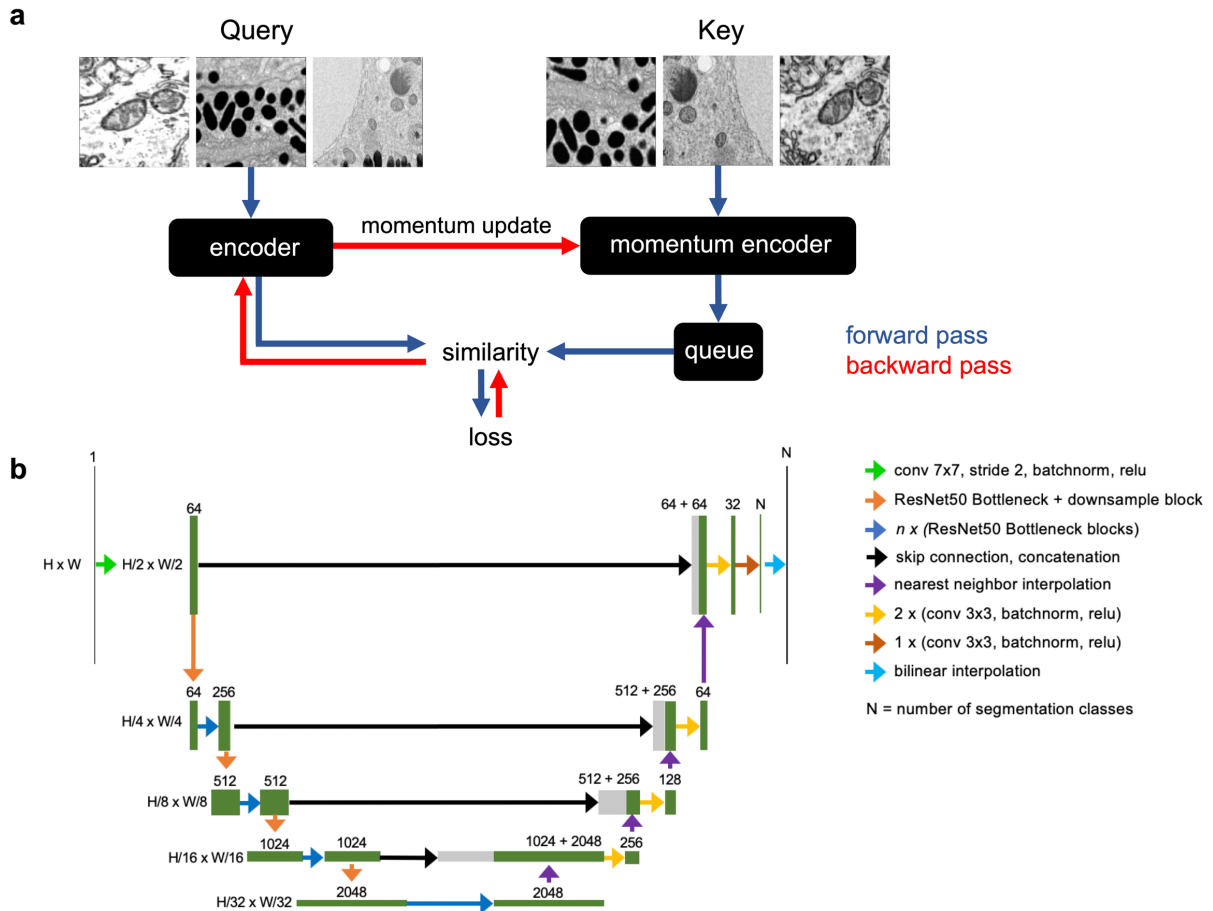
980

981 **Supplementary Figure 2: Randomly selected images from CEMraw, CEMdedup and**

982 **CEM500K.**

983

984



985

986 **Supplementary Figure 3: Schematics of the MoCoV2 algorithm and UNet-ResNet50 model**

987 **architecture.** (a) Shows a single step in the MoCoV2 algorithm. A batch of images is copied;

988 images in each copy of the batch are independently and randomly transformed and then shuffled

989 into a random order (the first batch is called the *query* and the second is called the *key*). *Query*

990 and *key* are encoded by two different models, the *encoder* and *momentum encoder*, respectively.

991 The encoded *key* is appended to the *queue*. Dot products of every image in the *query* with every

992 image in the *queue* measure similarity. The similarity between an image in the *query* and its

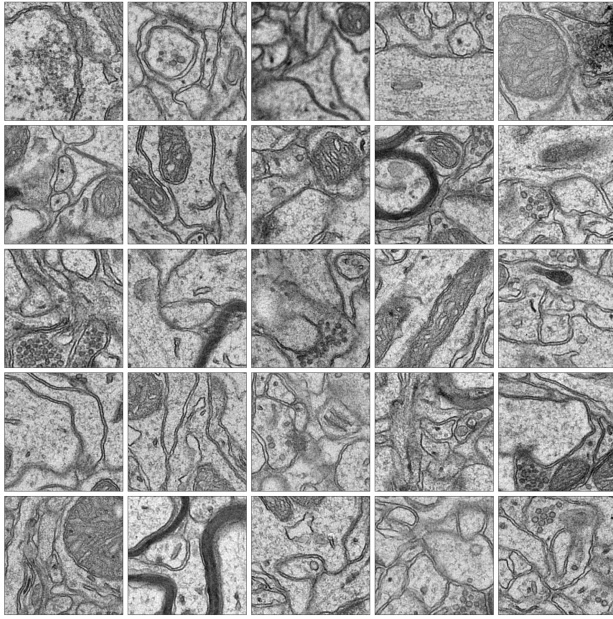
993 match from the *key* is the signal that informs parameter updates. More details in [34]. (b)

994 Detailed schematic of the UNet-ResNet50 architecture.

995

996



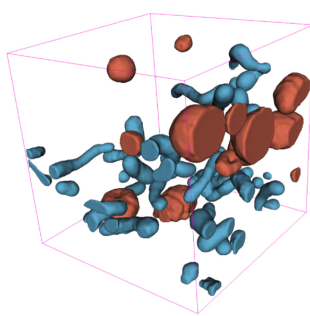


997

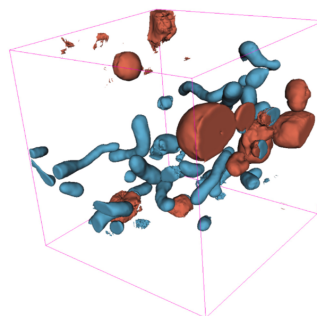
998 **Supplementary Figure 4: Randomly selected images from the Bloss et al. 2018 pre-training**

999 **dataset.**

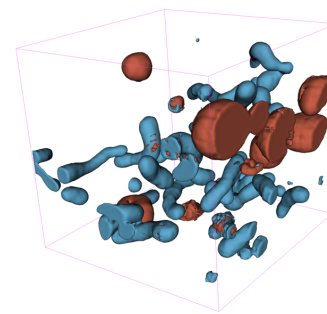
1000



UroCell Ground Truth [16]



UroCell Publication  
Presented Results [16]



CEM500K-moco Results

1001

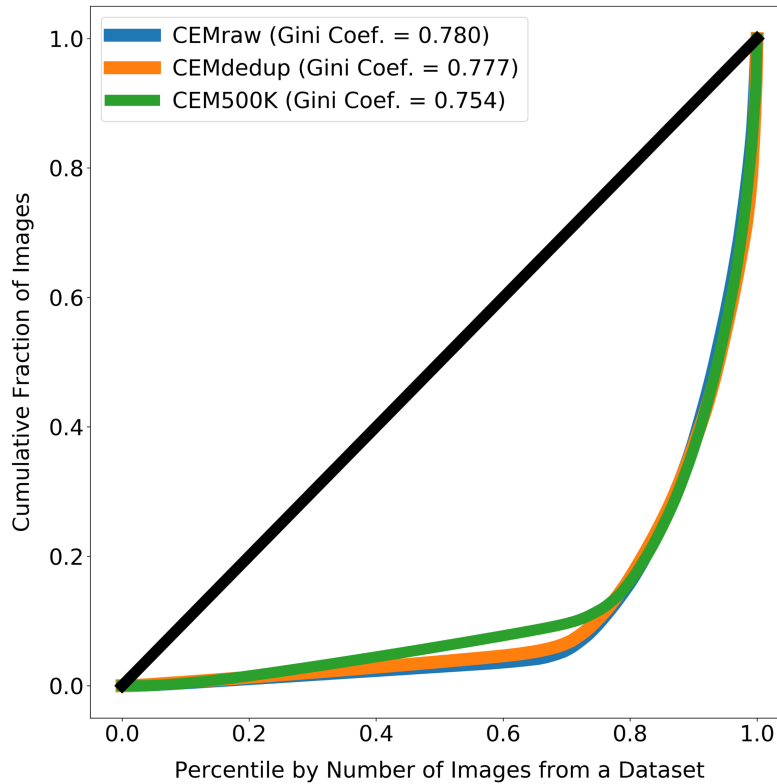
1002 **Supplementary Figure 5: Visual comparison of results on the UroCell benchmark. The**

1003 ground truth and Authors' Best Results are taken from the original UroCell publication [16]. The

1004 results from the CEM500K-moco pre-trained model have been colorized to approximately match

1005 the originals; 2D label maps were not included in the UroCell paper.

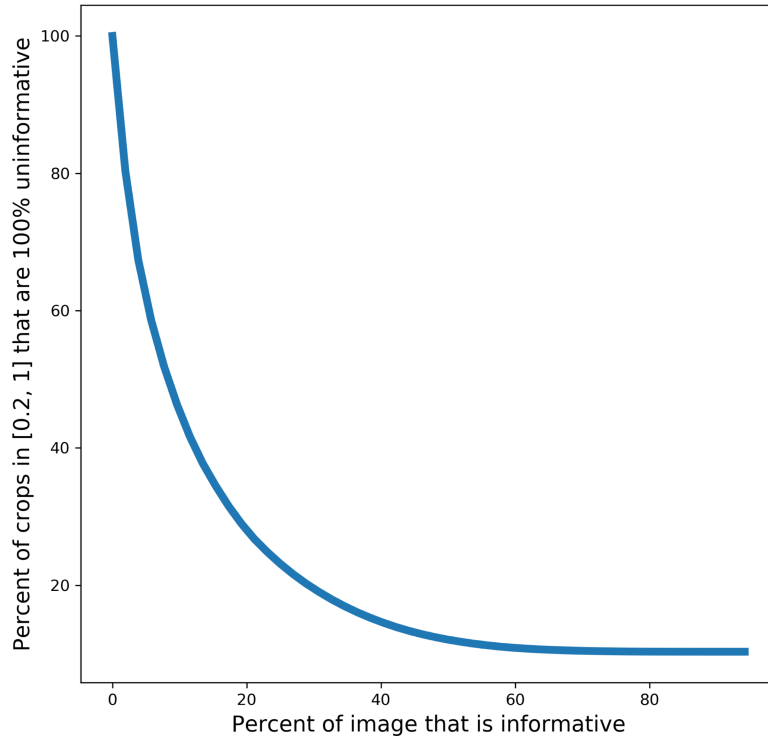
1006



1007

1008 **Supplementary Figure 6: Images from source EM volumes are unequally represented in**  
1009 **the subsets of CEM.** The line at 45° shows the expected curve for perfect equality between all  
1010 source volumes (i.e. each volume would contribute the same number of images to CEMraw,  
1011 CEM deup or CEM500K). Gini coefficients measure the area between the Lorenz Curves and the  
1012 line of perfect equality, with 0 meaning perfect equality and 1 meaning perfect inequality. For  
1013 each subset of CEM, approximately 20% of the source 3D volumes account for 80% of all the  
1014 2D patches.

1015



1016

1017 **Supplementary Figure 7: Plot showing the percent of random crops from an image that**

1018 **will be 100% uninformative based on the percent of the image that is informative.**