CellTranspose: Few-shot Domain Adaptation for Cellular Instance Segmentation

Matthew R. Keaton       Ram J. Zaveri       Gianfranco Doretto
West Virginia University  Morgantown, WV 26506
{mrkeaton, rz0012, gidoretto}@mix.wvu.edu

Abstract

Automated cellular instance segmentation is a process utilized for accelerating biological research for the past two decades, and recent advancements have produced higher quality results with less effort from the biologist. Most current endeavors focus on completely cutting the researcher out of the picture by generating highly generalized models. However, these models invariably fail when faced with novel data, distributed differently than the ones used for training. Rather than approaching the problem with methods that presume the availability of large amounts of target data and computing power for retraining, in this work we address the even greater challenge of designing an approach that requires minimal amounts of new annotated data as well as training time. We do so by designing specialized contrastive losses that leverage the few annotated samples very efficiently. A large set of results show that 3 to 5 annotations lead to models with accuracy that: 1) significantly mitigate the covariate shift effects; 2) matches or surpasses other adaptation methods; 3) even approaches methods that have been fully retrained on the target distribution. The adaptation training is only a few minutes, paving a path towards a balance between model performance, computing requirements and expert-level annotation needs.

1. Introduction

Automating the analysis of scientific imaging data via computer vision techniques is becoming increasingly convincing as our field matures. In order to accelerate scientific discovery, neural network-based methods have recently been developed to automatically segment and count individual instances of cells in laboratory-produced imaging data [44, 51, 14]. Such type of data acquisitions exhibit a remarkable variability, which is due to the large variety of imaging modalities being used, the different types of tissues and how they are processed.

Current approaches for the specific task of cell instance segmentation are mainly based on supervised learning. They are trained on large datasets in an attempt to compensate for the diversity of the new data they are meant to be used on. However, new data to process will very likely not be distributed in the same way as the data used for training the models, so, they will perform the task, often with disappointing accuracy. To address this covariate shift problem [42] the obvious solution is retraining the models, which is costly and time consuming because it requires manual annotation of large amounts of the new target data we are seeking to automatically process. An alternative is to use domain adaptation methods, which attempt to adapt the model to the target data distribution. Current domain adaptation methods for segmentation are by and large tailored to imaging modalities, or specific tasks or applications that are very different than cell instance segmentation [15, 53]. Some promising work has addressed the problem in an unsupervised manner [26, 27]. But these approaches assume that a large fraction and amount of target data is available to undergo a relatively intense training to adapt the model.

In this work, we advocate for a more practical and scalable solution to address the need to generalize well out of distribution. We assume that a model for segmenting instances, such as cell bodies, membranes, or nuclei, has already been trained on a source dataset. Then, by annotating only a handful of samples of the target dataset we adapt the model, with a low training budget, to generalize well on the new distribution. We introduce CellTranspose, a new approach that implements the paradigm just described, for the few-shot supervised adaption of cell instance segmentation. The approach builds on a state-of-the-art model, and we introduce new losses and a training procedure for the
quick adaptation of the model to new data.

Our framework allows for a broad range of data to be properly segmented beyond the capabilities of current generalist approaches. We show that only a small number of annotations on the target dataset are required for the model to learn to produce high-fidelity segmentations and demonstrate this both on 2-D and 3-D data. In particular, few annotated samples are sufficient also to reach adaptation levels comparable to the unsupervised adaptation models. Additionally, CellTranspose affords a much faster training scheme as compared to training a model with similar accuracy from scratch.

2. Related Work

2.1. Cellular Instance Segmentation

The current top approach for cell instance segmentation is Cellpose [44]. Like many recent segmentation algorithms, Cellpose’s underlying model is a variant of U-Net [38], outputing a pixel-wise mask prediction. By itself, this only produces a semantic segmentation: the output mask merely determines the class of each pixel (foreground/cell or background), making it impossible to delineate between individual cells when they are clustered together. Deriving inspiration from the traditional watershed algorithm [3] and a gradient-based deep pose recognition algorithm called OpenPose [5], Cellpose contains two additional outputs corresponding to predicted gradients towards each pixel’s associated cell source location, one in the $x$-direction and one in the $y$-direction. Through an iterative process, the gradient of each pixel is followed to neighboring pixels until a “source” pixel representing a cell center is encountered. Each pixel directly linked to the source is then considered a part of the cell instance, in this way constructing segmentations of individual cells.

Other cellular segmentation methods have been proposed recently. Similar to Cellpose, Mesmer [14] represents another variant of “deep watershed” algorithms, also producing a pixel-wise mask prediction, but instead of the two flow outputs it generates an “inner distance transform” to predict the distance of each pixel to a cell’s center. StarDist [40] and StarDist-3D [51] predict object centers and then approximate the distance from the given location to cell boundaries at fixed angles from the center. The points defined by these predictions are then connected to each neighboring point to produce the outline of the cell’s mask. Other approaches include NuSeT [57], which also builds on the watershed algorithm, and DenoiSeg [4], which utilizes a joint learning strategy in concert with a denoising task to produce better accuracy on noisy samples.

2.2. Domain Adaptation for Medical Imaging

A common issue in deep learning practice is the scarcity of labeled data for many tasks and the often unrealistic requirement to produce vast amounts of costly annotations in order to appropriately train a model. Although many efforts have been made to produce “generalist” models which are invariant across different datasets for a specific task, doing so is intractable in real-world applications where data has intrinsically high variability. Domain adaptation allows for taking a high-performing model trained on a large dataset and adapting it to work on new data representing some target domain. This enables the model to take advantage of learned low-level features acquired from the larger source dataset while tuning to the specific features of the target dataset, requiring less target data.

Due to this lower dependence on manual annotation from experts, a vast amount of research on domain adaptation-based approaches has been unleashed on the medical imaging field [15, 53]. In addition to a few domain adaptation approaches on semantic segmentation for various imaging modalities including brain tumors [10, 47, 12], whole tissue [65], and organelles [2], several unsupervised mechanisms for semantic segmentation [55, 22, 19, 49, 36, 9, 11, 21, 20, 63, 62, 7, 54, 56, 37, 41, 35, 23] and the more challenging instance segmentation task [26, 27, 58, 24] have been proposed. Generally, these unsupervised methods aim to learn representations of a given target domain without the use of any annotations. This greatly reduces the need for effort from medical experts while producing better segmentations than non-adapted models. However, this is accompanied by a clear trade-off with model accuracy, and the training scheme, in addition to more complex model architectures, causing training time to increase heavily. There is also an intrinsic expectation for the shape and relative size of segmentations from the target data to match closely with those of the source dataset, constraining such approaches to nuclei segmentation rather than more heterogeneous full-cell segmentations. These methods are also notably not “generalist,” and tend to focus on and perform best on a particular cell or imaging type.

2.3. Few-shot Domain Adaptation

In order to accommodate many real-world settings, one set of techniques known as “few-shot” domain adaptation aims to balance the demand for annotations with overall model performance. Since at least one large dataset is often available for a given task, many of the weaknesses posed by few-shot learning on its own can be avoided. Some popular techniques for addressing domain adaptation with a limited number of
target samples includes utilizing adversarial techniques to increase the confusion between the source and target domain [32], using meta-learning approaches to use tasks from one domain for adversarially learning tasks in a smaller, novel domain [39], and generating prototypes based on learned embeddings from the few target samples [64], a technique common to few-shot learning.

One which has proven effective across multiple paradigms is the use of contrastive losses [16]. A contrastive loss function encourages each sample from the target domain to be represented similarly to one or more samples of the same class (positive samples) in the source domain while simultaneously pushing its representation away from samples from other classes (negative samples) within the source domain. This approach has proven successful in a number of scenarios including domain adaptation [25, 50, 28], unsupervised learning [48, 66, 30, 45, 18, 6], self-supervised learning [8, 59, 31], meta-learning [39], and even few-shot domain adaptation [33].

3. Problem Definition

We are interested in the segmentation of instances such as cell bodies, membranes, or nuclei which are present in cell images. Such image structures have a remarkable variability because they are acquired with a variety of microscopy techniques, come from a diverse range of tissues, and present very different shapes. See Figure 1. Therefore, even state-of-the-art supervised instance-based segmentation approaches [44, 14], which claim to offer a generalist solution, unfortunately experience rapid performance degradation when facing the challenge of data distribution shift [42]. In essence, when the so-called generalist model is tasked to operate on data distributed differently than the data it was trained on, expectations are unmet.

Given how frequent the need to generalize out-of-distribution is for the task at hand, and given how costly and time consuming, if not completely impractical, the process of collecting and annotating sufficient target data for retraining the model can be, we reframe the problem as one of few-shot learning. The intent is to significantly widen the range of image variability handled by current generalist solutions. Specifically, given a source dataset $D^s$ made of labeled images with which a generalist model could be trained, the learning task is to adapt such a model to generalize well on a target dataset $D^t$, which is distributed differently than $D^s$. To perform such adaptation, the user is required to label only a minimal amount of target data, let us say $K$ data instances for a $K$-shot learning. In Section 4.2.3 we clarify the meaning of a single shot.

Note that simple fine-tuning of a generalist model on very few target samples is not a feasible solution due to the obvious overfitting challenges [13], which is why we introduce a method that aims at a very low budget in terms of the amount of target data to annotate, as well as the time for adaptation training. Given the specific setting of our problem, to the best of our knowledge, this is the first work to propose a method for few-shot supervised domain adaptation for the task of cell instance segmentation.

4. Approach

Our approach, shown in Figure 2, assumes that some large labeled source dataset $D^s$ can be used to initially train the model. However, regardless of the variety of cell images within $D^s$, there will always conceivably be new data samples outside of the source distribution. This is due to limitations in both model and dataset size. We also assume that given some out-of-distribution set of target samples, annotation is costly, and therefore only a few samples will be labeled. In the following subsections, we first describe the pretrained method we rely on, and then introduce the proposed few-shot supervised adaptation model.

4.1. Pretrained Model

To provide a robust starting point, we use the state-of-the-art cell instance segmentation approach Cellpose [44], which we summarize here to introduce notation and make the paper self-contained. Given an image $I$, the model uses a network $f$ to produce a dense, pixel-wise feature $Z = f(I)$, where $Z = [Z_1, Z_2, Z_3] \in \mathbb{R}^{h \times w \times 3}$. Given the feature $\mathbf{z} = [z_1, z_2, z_3] \in Z$ for some pixel $i$, then $\mathbf{z} \in (z_1, z_2)$ has the meaning of gradient pointing towards the center of the cell structure to which pixel $i$ belongs. Collectively, $(Z_1, Z_2)$ form a gradient flow. Instead, $z \in z_3$ represents the unnormalized score indicating the probability of pixel $i$ to belong to a cell structure. Note that with this notation, the feature $\mathbf{z}$ can be written as $\mathbf{z} = [z, z]$. The network $f$ is trained in a supervised manner with pixel-wise instance
The segmentation loss is

$$L_i^{IS} = (z_1 - g_x)^2 + (z_2 - g_y)^2 + \nu H(m, \sigma(z)),$$

where, for pixel $i$, $(g_x, g_y)$ represents the ground-truth gradient label with unit $\ell_2$-norm, $m \in \{0, 1\}$ is the binary mask label indicating absence/presence of a cell structure, $\sigma(z) = 1/(1 + \exp(-z))$, $H$ represents the binary cross-entropy, and $\nu$ is a hyperparameter set to 0.04. The pixel-wise loss contributions are then aggregated into a final loss $L^{IS} = \sum L_i^{IS}$ for the image $I$.

Given the feature $Z$, the cell instance segmentation head $g$ produces the mask $Y = g(Z)$, where for pixel $i$, the predicted label $y$ is a number in the set $\{0, 1, \cdots, N\}$, with $N$ being the total number of cell instances segmented. Pixels belonging to the same cell instance have same label, and $y = 0$ indicates absence of a cell instance. For details on how $g$ is implemented the reader can consult [44]. See also Figure 2.

4.2. Adaptation Model

Consider now a target image $I \in D^t$. If $I$ was drawn from the same distribution from which source images composing $D^s$ were drawn, we would expect the following to be true: For a target pixel $i$ with label $(g_x^t, g_y^t, m^t)$, its feature $z^t$ should be very close to the features of pixels in the source dataset that have the same label. However, in presence of a domain shift this is generally untrue, which leads to performance deterioration of the cell instance segmentation process, when we use the feature network $f$ on $I$, followed by the instance segmentation head $g$. Therefore, provided that the labels of some target image pixels are available, we design a method for reversing the effects of domain shift, by adapting $f$ to generalize well on the target dataset $D^t$. Given the distinct predictive nature (continuous vs. discrete) of the gradient flow features $z = (z_1, z_2)$, from the mask feature $z$, we proceed by designing adaptation losses for each case.

4.2.1 Contrastive Flow Loss

In order to align $z^t$ with the gradient flow features of source pixels with same label, we set up a contrastive prediction task [8]. We identify a positive source pixel with binary label $m^t = m^i = 1$, and with gradient flow features $z^s$ that best matches the label $(g_x^i, g_y^i)$, according to a similarity measure, like cosine similarity $s(u, v) = u^T v/\|u\|\|v\|$, where $\| \cdot \|$ denotes $\ell_2$-norm. Then, we compose a set of negative source gradient flow features $N_i = \{z^s \mid s(z^s, z^t) < \delta, m^s = 1\}$, where $\delta$ is a suitable constant hyperparameter threshold that we choose. Now we can use a contrastive loss function for pixel $i$ that attempts to pull the positive pair $(z^t, z^s)$ closer, while pushing apart every negative pair $(z^t, z^s)$, for $z^s \in N_i$. The loss function we used, which we name contrastive flow loss, is

$$L_i^{CF} = -\log \frac{\exp(s(z^t, z^s) / \tau)}{\exp(s(z^t, z^s) / \tau) + \sum_{z^s \in N_i} \exp(s(z^t, z^s) / \tau)}$$

where $\tau$ denotes a temperature parameter. Note that (2) addresses only the directional alignment of the gradient flow features. This is also what matters most, because the segmentation head $g$ uses only that information for assigning pixels to instances, not the gradient magnitude. However, in Section 4.2.3 we will see that (1) is also applied on target data, which does encourage gradients to have unit magnitude.

An important component of the loss is how we mine the negative source features. Specifically, $N_i$ is composed by selecting the features closest to the positive source feature $z^s$. This hard-mining strategy removes the need to consider large amounts of negative source features, because it makes them less informative. This leads to faster training, and usually more rapid convergence and better model performance, as it was shown also in [48]. Moreover, for a given target pixel $i$ we extract one positive feature, and all the $|N_i|$ negative features from one source image. We set $|N_i|$ to be the same for every target pixel, and generally, there
is a sufficient number of negative features such that
the similarity between \( z^t_+ \) and each negative feature is
approximately equal to \( \delta \). Therefore, the \( N_t \) features
are roughly equally divided between those at \( \cos^{-1} \delta \)
 radians clockwise and those at \( \cos^{-1} \delta \) radians counter-
clockwise from \( z^t_+ \). Thus, when \( z^t \) is off, the negative
samples nearer to \( z^t \) will produce a greater gradient
than those further away.

Given a target image patch and a source image patch
in the minibatch, for this pair we compute a loss contribu-
tion by aggregating all the components coming from the
target pixels that belong to cell structures, so they
have label \( m^t \). Let us indicate with \( M \) this set of
pixels, then the loss for the given pair will be
\[
L^{CF} = \frac{1}{|M|} \sum_{M} L^{CF}_i .
\] (3)

4.2.2 Contrastive Mask Loss

The contrastive flow loss (2) is a specialized version of
a loss used in [43, 46, 8, 48], and when \( \tau \to +\infty \) it converges to
\[
\lim_{\tau \to +\infty} L_i^{CF} = -s(z^t, z^s_+) + \lambda \sum_{z^s_+ \in N_t} s(z^t, z^s_+) ,
\] (4)

where \( \lambda \) is a hyperparameter. The loss (4) was originally proposed in [17], and a specialized form of it was introduced in [33] for supervised domain adaptation and generalization for multi-class visual classification.

Since we are interested in aligning the unnormalized
different classes
binary classification score \( z^t \) with the scores of the
source pixels with same label, we derive a contrastive
loss for segmentation, inspired by (4) and [33]. This
is also motivated by the fact that losses like (2) have
been proven effective with cosine similarity, which we
cannot use for this task and we need to replace.

When (4) is optimized, the first term aims at max-
imizing similarity. In our case this would be the sim-
ilarity between \( z^t \) and the unnormalized score of the
pixel of a source image \( z^s_+ \) with label \( m^s = m \). We can
replace that term with an alignment term based on a
distance, leading to a squared loss
\[
d(t^t, z^s_+) = \frac{1}{2}(z^t - z^s_+)^2 .
\] (5)
The second term of (4), instead, aims at minimizing
similarity. In our case this would be the similarity be-
 tween \( z^t \) and the unnormalized score of pixels in source
images with label different than \( m^t \). This would lead to
separation between scores with opposite labels. Given
one such pixel with score \( z^s_- \), we can measure the sim-
ilarity with \( z^t \) with
\[
k(t^t, z^s_-) = \frac{1}{2} \max(0, m - |z^t - z^s_-|)^2 ,
\] (6)

Figure 3. Contrastive Mask Loss. Representation of
the Contrastive Mask Loss for a given source-target sample
pair. The mask of a source and target sample are
overlapped to compare representations on a per-pixel ba-
sis. Pixels belonging to the same class are encouraged to
be represented similarly while pixels from different classes
have their representations separated.

where \( m \) is a margin, and the loss provides gradient
contributions when \( z^t \) is within the margin \( m \) of \( z^s_+ \).

Given a target image patch and a source image patch
in a minibatch, we aggregate the losses (5) and (6) as
follows. Let \( P \) be the set of positive pairs of scores
\((z^t, z^s_+)\) corresponding to pixels in the same relative
positions in the target and source patches, and with
same label, i.e., \( m^s = m \); and let \( N \) be the set of neg-
active pairs of scores \((z^t, z^s_-)\) corresponding to pixels in
the same relative positions in the target and source
patches but with different label, i.e., \( m^s \neq m \). Then,
the loss for a pair of target and source patches is
\[
L^{CM} = \frac{1}{|P|} \sum_{P} d(t^t, z^s_+) + \lambda \frac{1}{|N|} \sum_{N} k(t^t, z^s_-) ,
\] (7)
and we refer to this as the contrastive mask loss. Note
that each term of the loss is normalized with respect
the area covered by the positive and negative pairs,
respectively. Also, the formation of the sets \( P \) and
\( N \) is based on the comparison of the mask labels of the
target and source patches, as in Figure 3, and the
pairs originate from pixels with same relative position.
This is not a strict requirement, but it is convenient
because it allows forming sufficiently large sets \( P \) and
\( N \), while implementing (7) is easier and faster since it
takes advantage of the parallel architecture of GPUs.

4.2.3 Few-shot Adaptation

Assuming that \( D^t_K \) is the target dataset portion of \( D^t \)
with \( K \) labeled samples, the \( K \)-shot adaptation learning
aims at minimizing the loss
\[
L^{ISA} = \sum_{D^t_K} \left( L^{IS} + \frac{\gamma_1}{|D^s|} \sum_{D^s} L^{CM} + \frac{\gamma_2}{|D^s|} \sum_{D^s} L^{CF} \right) .
\] (8)
The training assumes that the generalist model has al-
ready been pretrained. The Cellpose model we used
5. Experiments

We evaluate the proposed approach, which we name CellTranspose, with several datasets and multiple settings. In absence of additional directions, the following setup is used for each experiment. The model in Section 4.1 is pretrained with the “generalized” dataset of [44] as source data $D^s$. Adaptation is done as described in Section 4.2.3. In general, we follow the data splitting guidelines laid out by previous works to ensure fair comparisons. In order to fit our approach to these guidelines, we draw our $K$ sample patches, to produce $D^s_k$, from the training split of the dataset of the target distribution. Instead, the testing split of the target distribution dataset is used as our remaining, unlabeled portion of the target dataset $D^t$. We use SGD with initial learning rate $10^{-2}$, momentum 0.9, weight decay $10^{-5}$, and batch size of 2 provides optimal results. For the first five epochs, the learning rate decreases by a factor of 10 each epoch, and is kept constant for the remaining five. We take square patches with side length $h = w = 112$, use a minimum overlap of 84 during evaluation, and enforce a nominal cell size $m_n = 30$. Additional hyperparameters are set as: $|N_i| = 20, \tau = 0.1, m = 10, \lambda = 1, \gamma_1 = 0.05, \gamma_2 = 2$, and $\delta = 0.05$. Because the source dataset used is always significantly larger than the target data, adaptation takes roughly the same amount of time regardless of the size of $K$ or the target data to which the model is adapted. Using a singular NVIDIA TITAN Xp GPU, adaptation takes approximately 5 minutes to complete for each experiment.

**Broad Bioimage Benchmark Collection – 006**

We evaluated CellTranspose on the target dataset BBBC006 [29], hosted by the Broad Institute. This dataset is composed of human U2OS cells which are fairly homogeneous and easy to segment in ideal settings. However, the same tissue samples have been imaged with different focus settings, generating different images, which allow us to observe the effect of the associated covariate shift on generalist and our approaches. We consider images taken with five focal planes, specifically at $z=0.00, 0.08, 0.16, 0.24$, and 0.32. The optimal focal plane is at $z=0.16$. Figure 4 shows qualitative segmentation results. As we move away from the optimal focal plane the generalist Cellpose model exposes greater performance deterioration than CellTranspose.
high variability of the background. The Triple Negative Breast Cancer (TNBC) dataset [34], gathered by the Curie Institute, is comprised of 50 images obtained from 11 distinct tissue types, furthering the inherent difficulty of the dataset.

We compare CellTranspose with the top-performing unsupervised domain adaptive cellular instance segmentation approach, CyC-PDAM [26]. This is the closest approach to ours we found in terms of problem settings and data, since the only other supervised domain adaptive instance segmentation approaches [61, 52] were tuned to very specific and different applications. Following the lead of CyC-PDAM, we divert from the protocol used above and pretrain CellTranspose on the BBBC039 dataset. We then adapt the model with 3, 5, and 10 shot samples selected from the target dataset of 40 images and 8 tissue types from TNBC. Each model is then tested on the 10 images from the remaining 3 tissue types. Results, using the same metrics as CyC-PDAM, are shown in Table 2. It can be noted that a 5-shot adaptation leads to performance metrics comparable with those of CyC-PDAM, that has used all the training target data available to adapt the model in an unsupervised manner. Note that the settings of this experiment is still different and disadvantageous from the protocol for which CellTranspose was designed, because the target testing data distribution is different from the target training data distribution. We also tested Cellpose-UB in this settings, which appears to be affected more than CellTranspose by the domain shift still present in the evaluation protocol.

### Table 2. TNBC dataset

<table>
<thead>
<tr>
<th>TissueNet</th>
<th>Cellpose</th>
<th>Cellpose-UB</th>
<th>CyC-PDAM</th>
<th>CellTranspose</th>
</tr>
</thead>
<tbody>
<tr>
<td>AJI</td>
<td>0.5819 ± 0.0114</td>
<td>0.6072 ± 0.0146</td>
<td>0.7092 ± 0.0188</td>
<td>0.7476 ± 0.0133</td>
</tr>
<tr>
<td>Pixel-F1</td>
<td>0.5415 ± 0.0187</td>
<td>0.6972 ± 0.0278</td>
<td>0.7568 ± 0.0293</td>
<td>0.7879 ± 0.0287</td>
</tr>
<tr>
<td>Object-F1</td>
<td>0.5498 ± 0.0180</td>
<td>0.6456 ± 0.0290</td>
<td>0.7216 ± 0.0204</td>
<td>0.7760 ± 0.0140</td>
</tr>
</tbody>
</table>

### 3D Segmentation

As biological applications often produce and require the analysis of 3-D data, following the approach in [44], we extend CellTranspose to operate in 3-D, named CellTranspose3D, by making adaptation and combining the predictions along the x, y, and z axis volume sections. We test our approach on two

---

Figure 5. BBBC006 dataset. Average precision of K-shot adaptation with K = 1, 2, 3, 5, 10 target samples. Experiments were completed using five BBBC006 datasets.

Table 1. TissueNet dataset. F-1 score from state-of-the-art “generalist” approaches and CellTranspose.

<table>
<thead>
<tr>
<th>TissueNet Results</th>
<th>Platform-specific</th>
<th>Tissue-specific</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cellpose</td>
<td>CODEX</td>
<td>Breast</td>
</tr>
<tr>
<td></td>
<td>CyCIF</td>
<td>GI</td>
</tr>
<tr>
<td></td>
<td>MIBH</td>
<td>Imm.</td>
</tr>
<tr>
<td></td>
<td>Vectra</td>
<td>Panc.</td>
</tr>
<tr>
<td>Mesmer</td>
<td>0.88</td>
<td>0.74</td>
</tr>
<tr>
<td></td>
<td>0.80</td>
<td>0.82</td>
</tr>
<tr>
<td></td>
<td>0.76</td>
<td>0.92</td>
</tr>
<tr>
<td></td>
<td>0.72</td>
<td>0.92</td>
</tr>
<tr>
<td>CellTranspose</td>
<td>0.940</td>
<td>0.911</td>
</tr>
<tr>
<td></td>
<td>0.940</td>
<td>0.906</td>
</tr>
<tr>
<td></td>
<td>0.932</td>
<td>0.934</td>
</tr>
<tr>
<td></td>
<td>0.918</td>
<td>0.955</td>
</tr>
<tr>
<td>Cellpose-UB</td>
<td>0.962</td>
<td>0.950</td>
</tr>
<tr>
<td></td>
<td>0.967</td>
<td>0.940</td>
</tr>
<tr>
<td></td>
<td>0.945</td>
<td>0.953</td>
</tr>
<tr>
<td></td>
<td>0.960</td>
<td>0.958</td>
</tr>
</tbody>
</table>

We also test CellTranspose with different numbers K = 1, 2, 3, 5, 10, of few-shot adaptation. Figure 5 plots the average precision (AP) against the intersection over union (IoU) for the different focal settings. CellTranspose consistently achieves high performance levels with as few as three annotated shots. Beyond this, results begin to exhibit diminished returns. So, unless otherwise specified, other results have been obtained with a 3-shot adaptation, given the balance it gives between performance and annotation needs.

TissueNet. We evaluated CellTranspose on TissueNet [14], a dataset developed alongside a generalist method called Mesmer. TissueNet is comprised of samples from various imaging platforms and tissue types, providing a wide-spanning array of cellular images. In [14] one set of experiments split TissueNet into subsets of the four most common imaging types, each of which was further divided into different tissue types. Similarly, four other subsets were composed of the four most common tissue types, each being further split into the imaging types that make up the samples for that tissue type. Table 1 shows the F-1 scores on these eight data splits, computed for the generalist Cellpose, Mesmer trained on each of the splits, and CellTranspose 3-shot adapted to each of the splits. Again, CellTranspose consistently shows improved performance. Since CellTranspose essentially adapts Cellpose to the target domain, we can interpret it as a lower bound of performance. On the other hand, we also re-trained Cellpose with each target training dataset, obtaining what could be interpreted as an upper bound for CellTranspose, and we indicated that as Cellpose-UB.

### Triple Negative Breast Cancer

Among the most challenging types of cellular data to segment is that of hematoxylin and cosin-stained (H&E) images. This is in part due to the fact that multiple cell types often appear within an individual sample, in addition to the high variability of the background. The Triple Negative Breast Cancer is most common tissue types, each being further split into different tissue types.
Figure 6. 3D Segmentation. Qualitative segmentation results on samples from the BBBC024 dataset. Note that Cellpose3D tends to oversample; this is likely due to the fact that the particular characteristics of the data Cellpose3D was trained on necessitated pixels of even lower intensities to be segmented. More importantly, on the bottom right of the results obtained with Cellpose3D, it is possible to observe some over-segmentation effects, which are not present in the results of CellTranspose3D.

Table 3. 3D datasets. Average Precision at IoU threshold 0.5.

<table>
<thead>
<tr>
<th>3-D Results</th>
<th>Worm</th>
<th>BBBC024</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cellpose3D</td>
<td>0.575</td>
<td>0.822</td>
</tr>
<tr>
<td>CellTranspose3D</td>
<td>0.648</td>
<td>0.994</td>
</tr>
<tr>
<td>Cellpose3D-UB</td>
<td>0.675</td>
<td>1.0</td>
</tr>
<tr>
<td>StarDist-3D</td>
<td>0.765</td>
<td></td>
</tr>
</tbody>
</table>

3-D datasets. The BBBC024 is composed of synthetic annotated cells from the Broad Institute. The Worm dataset is a series of nuclei images from larval stage C. elegans. Similar to the 2-D experiments, only three 2-D sample patches, taken from the XY-plane from volumes in each training set, are used as target data for the model.

StarDist-3D [51] and Cellpose3D-UB have been trained and tested on Worm, and Cellpose3D-UB on BBBC024 also, and serve here as upper bounds. Table 3 shows the average precision results, highlighting the improvement of CellTranspose3D over the generalist Cellpose3D. Figure 6 provides a qualitative comparison between 3D segmentations of the BBBC024 dataset.

Ablation Study. Table 4 shows ablation results computed on the BBBC006 dataset with $z=0$ and 3-shot adaptation. The addition of both contrastive losses improves the overall AP by more than 15%. Interestingly, however, the removal of only one adaptation loss tends to decrease the performance to below that of removing both, which follows the same training scheme outlined in Section 4.2.3, but without either adaptation loss. This indicates that the flow and mask losses are intrinsically tied to one another, which is consistent with the fact that weights are shared between both outputs until the final layer.

Additionally, the cell size calculation method seems to play an important role in accurate segmentation. Cellpose computes the cell size based upon the number of pixels corresponding to a particular cell, but non-spherical cells have the potential to provide a similar diameter while spanning a much larger area. Thus, a more robust cell size calculation could be based on computing the total rectangular area enclosing a cell, which is how it is done in CellTranspose. The second row from bottom in Table 4 shows that when this second strategy is replaced with the one used by Cellpose, performance further deteriorates.

6. Conclusions

In this work, we have exemplified the need for adaptation of cell instance segmentation methods when working in new domains. Our approach is effective at achieving this with very few samples, thanks to the proposed new losses and training procedures we implemented. CellTranspose shows a level of performance that effectively tackles the effects of the covariate shift, and that can be comparable to models fully retrained on the target distribution. We have also shown that few samples are sufficient to reach performance similar to unsupervised approaches that rely on larger datasets and training simulations. We found that 3 to 5 annotated samples usually allowed to match or surpass the state of the art. The adaptation procedure takes a few minutes on a single GPU, striking an attractive balance in terms of time and resources used for training and annotation.

Acknowledgements

Research reported in this publication was supported by the National Institute Of Mental Health of the National Institutes of Health under Award Number R44MH125238. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health. This material is also based upon work supported by the National Science Foundation under Grant No. 1920920.
References


