

---

# Stateless actor-critic for instance segmentation with high-level priors

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

1 Instance segmentation is an important computer vision problem which remains  
2 challenging despite impressive recent advances due to deep learning-based meth-  
3 ods. Given sufficient training data, fully supervised methods can yield excellent  
4 performance, but annotation of ground-truth data remains a major bottleneck, espe-  
5 cially for biomedical applications where it has to be performed by domain experts.  
6 The amount of labels required can be drastically reduced by using rules derived  
7 from prior knowledge to guide the segmentation. However, these rules are in  
8 general not differentiable and thus cannot be used with existing methods. Here,  
9 we relax this requirement by using stateless actor critic reinforcement learning,  
10 which enables non-differentiable rewards. We formulate the instance segmentation  
11 problem as graph partitioning and the actor critic predicts the edge weights driven  
12 by the rewards, which are based on the conformity of segmented instances to  
13 high-level priors on object shape, position or size. The experiments on toy and real  
14 datasets demonstrate that we can achieve excellent performance without any direct  
15 supervision based only on a rich set of priors.

## 16 1 Introduction

17 Instance segmentation is the task of segmenting all objects in an image and assigning each of them  
18 a different label. It forms the necessary first step to the analysis of individual objects in a scene  
19 and is thus of paramount importance in many practical applications of computer vision. Over the  
20 recent years, fully supervised instance segmentation methods have made tremendous progress both  
21 in natural image applications and in scientific imaging, achieving excellent segmentations for very  
22 difficult tasks [1, 2].

23 A large corpus of training images is hard to avoid when the segmentation method needs to take  
24 into account the full variability of the natural world. However, in many practical segmentation  
25 tasks the appearance of the objects can be expected to conform to certain rules which are known *a*  
26 *priori*. Examples include surveillance, industrial quality control and especially medical and biological  
27 imaging applications where full exploitation of such prior knowledge is particularly important as the  
28 training data is sparse and difficult to acquire: pixelwise annotation of the necessary instance-level  
29 groundtruth for a microscopy experiment can take weeks or even months of expert time. The use of  
30 shape priors has a strong history in this domain [3, 4], but the most powerful learned shape models  
31 still require groundtruth [5] and generic shapes are hard to combine with the CNN losses and other,  
32 non-shape, priors. For many high-level priors it has already been demonstrated that integration of  
33 the prior directly into the CNN loss can lead to superior segmentations while significantly reducing  
34 the necessary amounts of training data [6]. However, the requirement of formulating the prior as  
35 a differentiable function poses a severe limitation on the kinds of high-level knowledge that can  
36 be exploited with such an approach. The aim of our contribution is to address this limitation and

37 establish a framework in which a rich set of non-differentiable rules and expectations can be used to  
38 steer the network training.

39 To circumvent the requirement of a differentiable loss function, we turn to the reinforcement learning  
40 paradigm, where the rewards can be computed from a non-differentiable cost function. We base  
41 our framework on a stateless actor-critic setup [7], providing one of the first practical applications  
42 of this important theoretical construct. In more detail, we solve the instance segmentation problem  
43 as agglomeration of image superpixels, with the agent predicting the weights of the edges in the  
44 superpixel region adjacency graph. Based on the predicted weights, the segmentation is obtained  
45 through (non-differentiable) graph partitioning and the segmented objects are then evaluated by the  
46 critic, which learns to approximate the rewards based on the object- and image-level reasoning (see  
47 Fig. 1).

48 The main contributions of this work can be summarized as follows: (i) we formulate instance segmen-  
49 tation as a RL problem based on a stateless actor-critic setup, encapsulating the non-differentiable step  
50 of instance extraction into the environment and thus achieving end-to-end learning; (ii) we exploit  
51 prior knowledge on instance appearance and morphology by tying the rewards to the conformity of  
52 the predicted objects to pre-defined rules and learning to approximate the (non-differentiable) reward  
53 function with the critic; (iii) we introduce a strategy for spatial decomposition of rewards based on  
54 fixed-sized subgraphs to enable localized supervision from combinations of object- and image-level  
55 rules. (iv) we demonstrate the feasibility of our approach on synthetic and real images and show  
56 an application to an important segmentation task in developmental biology, where our framework  
57 delivers an excellent segmentation with no supervision other than high-level rules.

## 58 **2 Related work**

59 Reinforcement learning has so far not found significant adoption in the segmentation domain. The  
60 closest to our work are two methods in which RL has been introduced to learn a sequence of  
61 segmentation decision steps as a Markov Decision Process. In the actor critic framework of [8], the  
62 actor recurrently predicts one instance mask at a time based on the gradient provided by the critic.  
63 The training needs fully segmented images as supervision and the overall system, including an LSTM  
64 sub-network between the encoder and the decoder, is fairly complex. In [9], the individual decision  
65 steps correspond to merges of clusters while their sequence defines a hierarchical agglomeration  
66 process on a superpixel graph. The reward function is based on Rand index and thus not differentiable,  
67 but the overall framework requires full (super)pixelwise supervision for training.

68 Reward decomposition was introduced for multi agent RL by [10] where a global reward is decom-  
69 posed into a per agent reward. [11] proves that a stateless RL setup with decomposed rewards requires  
70 far less training samples than a RL setup with a global reward. In [12] reward decomposition is  
71 applied both temporally and spatially for zero-shot inference on unseen environments by training on  
72 locally selected samples to learn the underlying physics of the environment.

73 The restriction to differentiable losses is present in all application domains of deep learning. Common  
74 ways to address it are usually based on a soft relaxation of the loss that can be differentiated. The  
75 relaxation can be designed specifically for the loss, such as, for example, Area-under-Curve [13] for  
76 classification or Jaccard Index [14] for semantic segmentation. These approaches are not directly  
77 applicable to our use case as we aim to enable the use of a variety of object- and image-level priors  
78 which can easily be combined without handcrafting an approximate loss for each case. More generally,  
79 but still for a concrete task loss, Direct Loss Minimization has been proposed for CNN training in  
80 [15]. For semi-supervised learning of a classification or ranking task, Discriminative Adversarial  
81 Networks have been proposed as a means to learn an approximation to the loss [16]. Most generally,  
82 Grabocka et al. in [17] propose to train a surrogate neural network which will serve as a smooth  
83 approximation of the true loss. In our setup, the critic can informally be viewed as a surrogate network  
84 as it learns to approximate the priors through the rewards by Q-learning.

85 Incorporation of rules and priors is particularly important in biomedical imaging applications, where  
86 such knowledge can be exploited to augment or even substitute scarce groundtruth annotations.  
87 For example, the shape prior is explicitly encoded in the popular nuclear [18] and cellular [19]  
88 segmentation algorithms based on spatial embedding learning. Learned non-linear representations  
89 of the shape are used in [5], while in [20] the loss for object boundary prediction is made topology-  
90 aware. Domain-specific priors can also be exploited in post-processing by graph partitioning [21].  
91 Interestingly, the energy minimization procedure underlying the graph partitioning can also be  
92 incorporated into the learning step [22, 23].

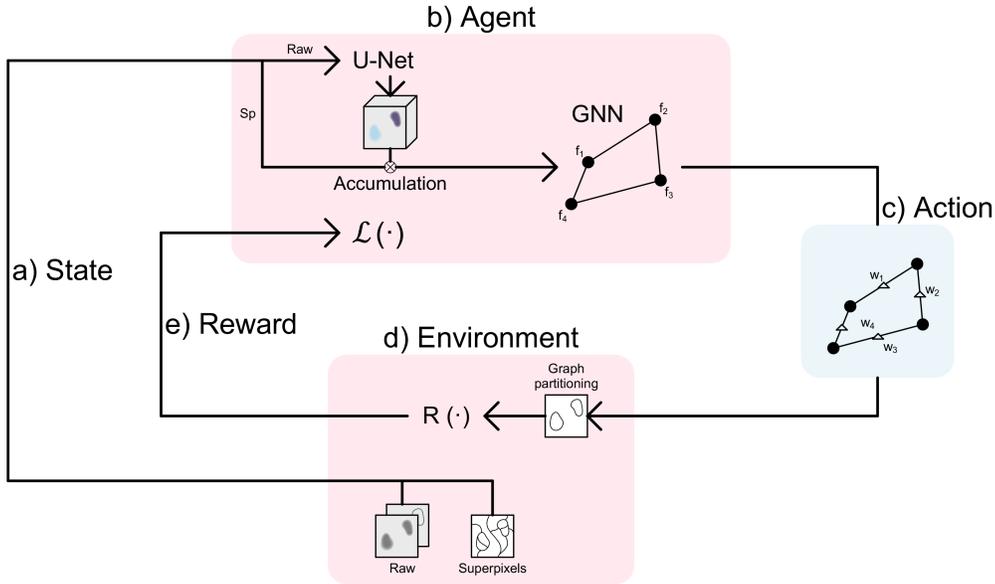


Figure 1: Interaction of the agent with the environment: (a) shows the state which is composed of the raw image and the superpixel over-segmentation; (b) depicts the agent and the superpixel graph, which accumulates the features for nodes of the GNN from pixels which belong to the corresponding superpixels; (c) given the state, the agent performs the actions by predicting edge weights on the superpixel graph; (d) the environment, which includes the graph partitioning built from the weights predicted through agent actions; (e) rewards are obtained by evaluating the segmentation arising from the graph partitioning, based on pre-defined and data dependent rules. The rewards are given back to the agent where they are used for training.

### 93 3 Methods

94 The task of instance segmentation can be formalized as transforming an image  $x$  into a labeling  $y$ ,  
 95 where  $y$  maps each pixel to a label value. An instance corresponds to the maximal set of pixels with  
 96 the same label value. Typically, the instance segmentation problem is solved via supervised learning,  
 97 i.e. using a training set with ground-truth labels  $\hat{y}$ . Note that  $y$  is invariant under the permutation  
 98 of label values. In general, it is difficult to formulate instance segmentation in a fully differentiable  
 99 manner. Most approaches first predict a "soft" representation with a CNN, e.g. affinities [1, 24, 25],  
 100 boundaries [26, 27] or embeddings [28, 29] and apply non-differentiable post-processing, such as  
 101 agglomeration [27, 30], clustering [31, 32] or partitioning [33], to obtain the instance segmentation.  
 102 Alternatively, proposal-based methods predict a bounding-box per instance and then predict the  
 103 instance mask for each bounding-box [34]. Furthermore, the common evaluation metrics for instance  
 104 segmentation [35, 36] are also not differentiable.

105 Our main motivation to explore RL for the instance segmentation task is to circumvent the restriction  
 106 to differentiable losses and - regardless of the loss - to make the whole pipeline differentiable end-to-  
 107 end even in presence of non-differentiable steps which transform pixelwise CNN predictions into  
 108 individual instances.

109 We formulate the instance segmentation problem using a region adjacency graph  $G = (V, E)$ ,  
 110 where the nodes  $V$  correspond to superpixels (homogeneous clusters of pixels) and the edges  $E$   
 111 connect nodes which belong to spatially adjacent superpixels. Given edge weights  $W$ , an instance  
 112 segmentation can be obtained by partitioning the graph, here using an approximate multicut solver  
 113 [37]. Together, the image data, superpixels, graph and the graph partitioning make up the environment  
 114  $\mathcal{E}$  of our RL setup. Based on the state  $s$  of  $\mathcal{E}$ , the agent  $\mathcal{A}$  predicts actions  $a$ , which are used to  
 115 compute the partitioning. The reward  $r$  is then computed based on this partitioning. Our agent  $\mathcal{A}$  is a  
 116 stateless actor-critic [38], represented by two graph neural networks (GNN) [39]. The actor predicts  
 117 the actions  $a$  based on the graph and its node features  $F$ . The node(superpixel) features are computed  
 118 by pooling together the corresponding pixel features based on the raw image data.

119 Here, we make use of two different setups: Method 1, where the per-pixel features are computed  
 120 based on the image data with the feature extractor being part of the agent  $\mathcal{A}$  and Method 2 where the  
 121 feature extractor is part of the environment  $\mathcal{E}$ . The feature extractor is trained end-to-end in Method  
 122 1, whereas it is fixed and thus needs to be pre-trained in Method 2. We use a U-Net [40] as feature  
 123 extractor and can use hand-crafted features in addition to the learned features. More details about  
 124 the pre- training can be found in the Appendix. The agent - environment interaction for Method 1 is  
 125 depicted in Figure 1. For Method 2 we refer to the Appendix.

126 Importantly, this setup enables us to use both a non-differentiable instance segmentation step and  
 127 reward function, by encapsulation of the “pixels to instances” step in the environment and learning a  
 128 policy based on the rewards with a stateless actor critic.

### 129 3.1 Stateless Reinforcement Learning Setup

130 Unlike most RL settings [41], our approach does not require an explicitly time dependent state: the  
 131 actions returned by the agent correspond to the real-valued edge weights in  $[0, 1]$ , which are used to  
 132 compute the graph partitioning. Any state can be reached by a single step from the initial state and  
 133 there exists no time dependency in the state transition. Unlike [9], we predict all edge values at once  
 134 which allows us to avoid the iterative strategy of [8] and deliver and evaluate a complete segmentation  
 135 in every step. We implement a stateless actor critic formulation with episodes of length 1.

136 To the best of our knowledge, stateless RL was introduced in [7] to study the connection between  
 137 generative adversarial networks and actor critics and our method is one of the first practical applica-  
 138 tions of this concept. Here, the agent consists of an actor, which predicts the actions  $a$  and a critic,  
 139 which predicts the action value  $Q$  (expected future discounted reward) given the actions. The stateless  
 140 approach simplifies the action value function: the action value has to estimate the reward for a single  
 141 step instead of estimating the expected sum of discounted future rewards for many steps. We have  
 142 explored a multi-step setup as well, but found that it yields inferior results for our application; details  
 143 can be found in the Appendix. As described in detail in 3.2, we compute localized sub-graph rewards  
 144 instead of relying on a single global reward.

145 The actor corresponds to a single GNN, which predicts the mean and variance of a normal distribution  
 146 for each edge. The actions  $a$  are determined by sampling from this distribution and applying a  
 147 sigmoid to the result to obtain continuous edge weights in the value range  $[0, 1]$ . The GNN takes the  
 148 state  $s = (G, F)$  as input arguments and its graph convolution for the  $i^{th}$  node is defined as in [39]:

$$f_i = \gamma_\pi \left( f_i, \frac{1}{|N(i)|} \sum_{j \in N(i)} \phi_\pi(f_i, f_j) \right) \quad (1)$$

149 where  $\gamma_\pi$  as well as  $\phi_\pi$  are MLPs,  $(\cdot, \cdot)$  is the concatenation of vectors and  $N(i)$  is the set of neighbors  
 150 of node  $i$ . The gradient of the loss for the actor is given by:

$$\nabla_\theta \mathcal{L}_{actor} = \nabla_\theta \frac{1}{|SG|} \sum_{sg \in G} \left[ \alpha \sum_{\hat{a} \in sg} \log(\pi^\theta(\hat{a}|s)) - Q_{sg}(s, a) \right] \quad (2)$$

151 This loss gradient is derived following [38]. We adapt it to the sub-graph reward structure by  
 152 calculating the joint action probability of the policy  $\pi^\theta$  over each sub-graph  $sg$  in the set of all  
 153 sub-graphs  $SG$ . Using this loss to optimize the policy parameters  $\theta$  minimizes the Kullback-Leibler  
 154 divergence between the Gibbs distribution of action values for each sub-graph  $Q_{sg}(s, a)$  and the  
 155 policy with respect to the parameters  $\theta$  of the policy.  $\alpha$  is a trainable temperature parameter which is  
 156 optimized following the method introduced by [38].  
 157

158 The critic predicts the action value  $Q_{sg}$  for each sub-graph  $sg \in SG$ . It consists of a GNN  $Q_{sg}(s, a)$   
 159 that takes the state  $s = (G, F)$  as well as the actions  $a$  predicted by the actor as input and predicts a  
 160 feature vector for each edge. The graph convolution from Equation 2 is slightly modified:

$$f_i = \gamma_Q \left( f_i, \frac{1}{|N(i)|} \sum_{j \in N(i)} \phi_Q(f_i, f_j, a_{(i,j)}) \right) \quad (3)$$

161 again  $\gamma_Q$  and  $\phi_Q$  are MLPs. Based on these edge features  $Q_{sg}$  is predicted for each sub-graph via an  
 162 MLP. Here, we use a set of subgraph sizes (typically, 6, 12, 32, 128) to generate a supervision signal

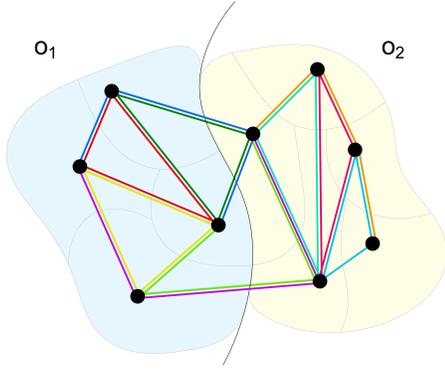


Figure 2: The graph is subdivided into sub-graphs, each sub-graph is highlighted by a different color. All sub-graphs have the same number of edges (here 3). Overall, we use a variety of sizes covering different notions of locality.

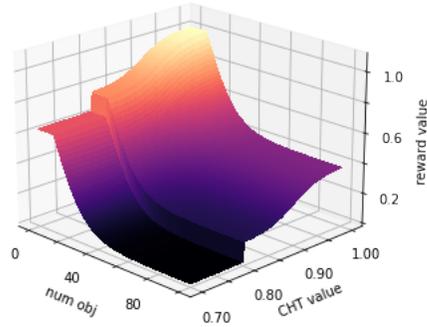


Figure 3: An example reward landscape Circle Hough Transform (CHT) rewards. High rewards are given if the overall number of predicted objects is not too high and if the respective object has a large CHT value. We found an exponential gradient of the reward landscape to work best.

163 for different neighborhood scales. A given MLP is only valid for a fixed graph size, so we employ a  
 164 different MLP for each size. The loss for the critic is given by:

$$\mathcal{L}_{critic} = \frac{1}{|SG|} \sum_{sg \in G} \frac{1}{2} (Q_{sg}^{\delta}(s, a) - r)^2 \quad (4)$$

165 Minimizing this loss with respect to the action value function’s parameters  $\delta$  minimizes the difference  
 166 between the expected reward and action values  $Q_{sg}^{\delta}(s, a)$ .

### 167 3.2 Localized Supervision Signals

168 The RL paradigm is to provide a global reward for a given state transition [41]. However, we find  
 169 that for our application it is possible and desirable to instead provide several more localized rewards  
 170 per state transition: Given a large action space with a policy represented by a complex multivariate  
 171 probability distribution, it is beneficial to learn from rewards for the specific actions rather than from  
 172 a scalar global reward for the union of all actions. Of course then requirement arises that the union of  
 173 local rewards must resemble to the global reward. E.g. the optimal policy is the same for local as for  
 174 the global reward.

175 Our actor critic setup (Section 3.1) expects rewards per sub-graph. A good set of sub-graphs should  
 176 fulfill the following requirements: each sub-graph should be connected so that the information  
 177 presented to the MLP computing the action value for this sub-graph is correlated. The size of  
 178 the sub-graphs, given by the number of edges, should be a parameter and all sub-graphs should  
 179 be extracted with exactly that size to serve as valid input for one of the MLPs. The union of all  
 180 sub-graphs should cover the complete graph so that each edge contributes to at least one action  
 181 value  $Q_{sg}$ . The sub-graphs should overlap to provide a smooth sum of action values. We introduce  
 182 Algorithm 1 to extract such a set of sub-graphs (see Appendix). Figure 2 shows the sub-graphs for a  
 183 small example graph.

184 While some of the rewards used in our experiments can be directly defined for the sub-graphs, most  
 185 are instead defined per object (see Appendix for details on reward design). We use the following  
 186 general procedure to map object-level rewards to sub-graphs: first assign to each superpixel the  
 187 reward of its corresponding object, then determine the reward per edge as the maximum value of its  
 188 two incident superpixels’ rewards and average the edge rewards to obtain the reward per sub-graph.  
 189 Here, we use the maximum because high object scores indicate that all actions contributing to the  
 190 respective object should get a high reward. However, for low object scores it is not possible to localize  
 191 the specific action responsible for the low score. Hence, by taking the maximum we assign the  
 192 higher score to edges whose incident superpixels belong to different objects, because they probably  
 193 correspond to a correct split. Note that the uncertainty in the assignment of low rewards can lead to  
 194 a noisy reward signal, but the averaging of the edge rewards over the sub-graphs and the overlaps

195 between the sub-graphs smooth and partially denoise the rewards. We have also explored a different  
 196 actor critic setup that can use object level rewards directly, eliminating the need for the sub-graph  
 197 extraction and mapping. However, this approach yields inferior results, see the Appendix for details.

## 198 4 Experiments

199 The agent of our setup acts on the superpixel graph and thus depends on the features assigned to the  
 200 nodes of the graph. We introduced two variants of our algorithm: in the base variant (Method 1)  
 201 we start from random features and make them part of the agent, allowing them to change through  
 202 back-propagation (Fig. 1). In contrast, Method 2 acts on predefined features which are provided  
 203 as part of the environment and are computed before training, e.g. through unsupervised clustering.  
 204 A very accurate clustering in the features produces an easy problem for the agent to solve where  
 205 even a global reward for all actions might be sufficient. However, in a real-world setting with no  
 206 supervision, the noisier the features become the more local the reward has to be. We evaluate Method  
 207 2 on synthetic data where self-supervised pretraining can deliver noisy, but meaningful node features.  
 208 Our full setup with Method 1 is evaluated on a dataset from a light microscopy experiment, where  
 209 highly regular object shapes are to be expected, but no good feature pre-training is possible.

210 To transform the edge weight predictions of the agent into an instance segmentation we use the  
 211 Multicut [42] algorithm. Here, other options are also possible such as hierarchical clustering used in  
 212 [9], but we choose the Multicut for its global optimality property. Hyperparameters of the pipeline  
 213 were found by cross-validation (see Appendix).

### 214 4.1 Synthetic dataset: circles on structured ground

215 To evaluate the feasibility of our approach, we create a synthetic dataset with prominent structured  
 216 background. Our aim is to segment irregular disks on such background using only rule-based  
 217 supervision. We generate the superpixels by the mutex watershed algorithm [25] which we run on  
 218 the Gaussian gradient image. The node features of the superpixel graph were computed through  
 219 self-supervised pretraining with contrastive loss as described in Appendix and fixed as part of the  
 220 environment.

221 As we aim to segment disks, we compute the circularity of the segmented objects for the rewards  
 222 using the Circle Hough Transform [43]. This object-level reward is combined with the global rough  
 223 estimate of the number of objects in the image to create the reward surface depicted in Fig. 3. The  
 224 reward for the number of objects provides useful gradient during early training stages: for example,  
 225 when too few potential objects are found in the prediction, a low reward can be given to what is  
 226 thought to be the background object. On the other hand, if too many potential objects are found, a  
 227 low reward can be given to all the foreground objects with a low CHT value.

228 In more detail, the object rewards  $r_{fg}$  are composed as follows. We define a threshold  $\gamma$  on the CHT  
 229 value ( $\gamma = 0.8$  in the reward surface shown in Fig. 3). Let  $c \in [0, 1]$  be the CHT value corresponding  
 230 to the object and let  $k$  be the total number of objects that we expect and  $n$  be the number of predicted  
 231 objects. Then

$$r_{local} = \begin{cases} \sigma\left(\left(\frac{c-\gamma}{1-\gamma} - 0.5\right)6\right) 0.4, & \text{if } c \geq \gamma \\ 0, & \text{otw} \end{cases} \quad (5)$$

$$r_{global} = \begin{cases} r_{exp}\left(\frac{k}{n}\right), & \text{if } n \geq k \\ 0.6, & \text{otw} \end{cases} \quad (6)$$

$$r_{fg} = r_{local} + r_{global} \quad (7)$$

232 Here  $\sigma(\cdot)$  is the sigmoid function. The input to the sigmoid function is normalized to the interval  
 233  $[-3, 3]$  which was empirically found to be a good range. The rewards are always in  $[0, 1]$  here this is  
 234 split up into  $[0, 0.5]$  for the local reward as well as for the global reward.

235 For the largest predicted object we strongly suspect the background object. For this object background  
 236 rewards  $r_{bg}$  are calculated by

$$r_{bg} = \begin{cases} r_{exp}\left(\frac{n}{k}\right), & \text{if } n \leq k \\ 1, & \text{otw} \end{cases} \quad (8)$$

237 Note that this rewards have a large globally calculated part which makes this setup not fit for Method  
 238 1. It needs some feature representation that already gives a good idea for the clustering. The only

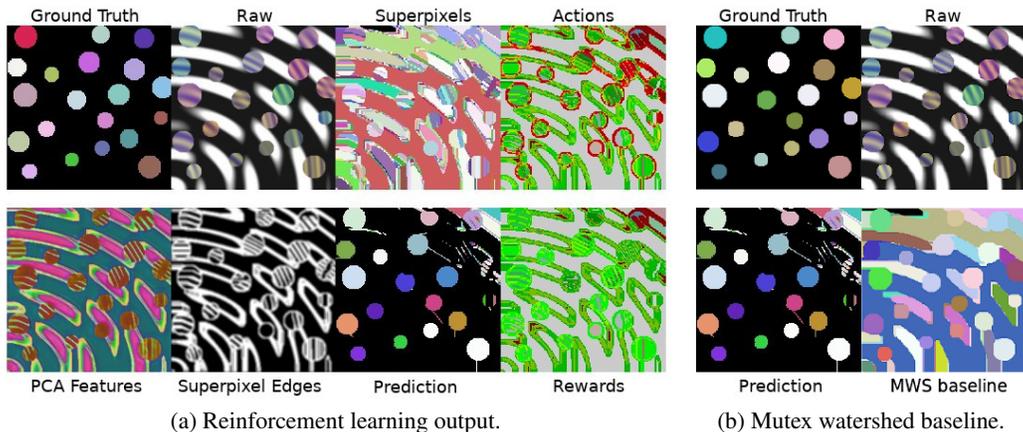


Figure 4: The “Circles” dataset. Top left to right: ground truth segmentation, raw data, superpixel over-segmentation and a visualization for the actions on every edge, where a merge action is displayed in green and a split action in red. Bottom left to right: the pre-trained pixel embeddings projected to their first 3 PCA components shown as RGB, an edge image of the superpixels, the segmentation resulting from the graph agglomeration on the predicted edge weights and a visualization of the rewards based on the CHT, where light green shows high rewards and dark red low rewards.

239 useful local information in the reward is the CHT value. Therefore, if the features have a fairly  
 240 distinct structure for circles, the agent should be able to find and to correctly cluster them.  
 241 Fig. 4 shows the output of all algorithm components on a sample image. For comparison, we also  
 242 computed mutex watershed [25] predictions. Texture within objects and structured background are  
 243 inherently difficult for region-growing algorithms, but our approach can exploit higher-level reasoning  
 244 along with low-level information and achieve a good segmentation.

## 245 4.2 Real dataset: light microscopy imaging

246 Biomedical applications often require segmentation of objects of known morphology which are  
 247 positioned in regular patterns, while extensive prior knowledge is available on variability of both  
 248 under normal experimental conditions [44]. Such data presents the best use case for our algorithm as  
 249 the reward function can leverage the known characteristics of individual object shape and texture and  
 250 the overall similarity of the objects.

251 The dataset used for this experiment contains 317 2D images extracted from a video of a developing  
 252 fruitfly embryo acquired with a light-sheet microscope [45] (Fig. 5). The image shows boundaries  
 253 (plasma membranes) of the embryo cells. Across the dataset, 10 images were fully segmented by an  
 254 expert, we use those for validation.

255 Fruitfly embryo is a well-studied system for which we can exploit the prior knowledge on the expected  
 256 cell shape and the radial pattern of cells. Furthermore, as the analysis of cell shape dynamics is  
 257 a paramount part of many biological experiments, multiple pre-trained networks are available for  
 258 the cell segmentation task [18, 19, 46, 47]. Due to the differences in sample preparation and image  
 259 acquisition settings, none of these would work out-of-the-box for our data. However, the CNNs in  
 260 [47] which are trained to predict boundaries in confocal microscope images of plant tissue, can serve  
 261 as a strong edge detector to create superpixels in our images. The superpixels are obtained using the  
 262 seeded watershed algorithm on seeds at the local minima of the predicted edge map.

263 The rewards for this experiment are designed as follows: we set a high reward for merging the  
 264 superpixels which are certain to lie in the background (close to the image boundary or the image  
 265 center). For the background edges near the foreground area we modulate the reward by the circularity  
 266 of the overall foreground contour. Finally, for the edges which are likely to be in the foreground  
 267 we compute object-level rewards by fitting a rotated bounding box to each object and comparing its  
 268 side lengths as well as its orientation to predefined template values. We do not perform semantic  
 269 segmentation to define precise foreground/background boundaries, but instead use a soft weighting  
 270 scheme with Gaussian weights to combine object and background rewards based on the prior

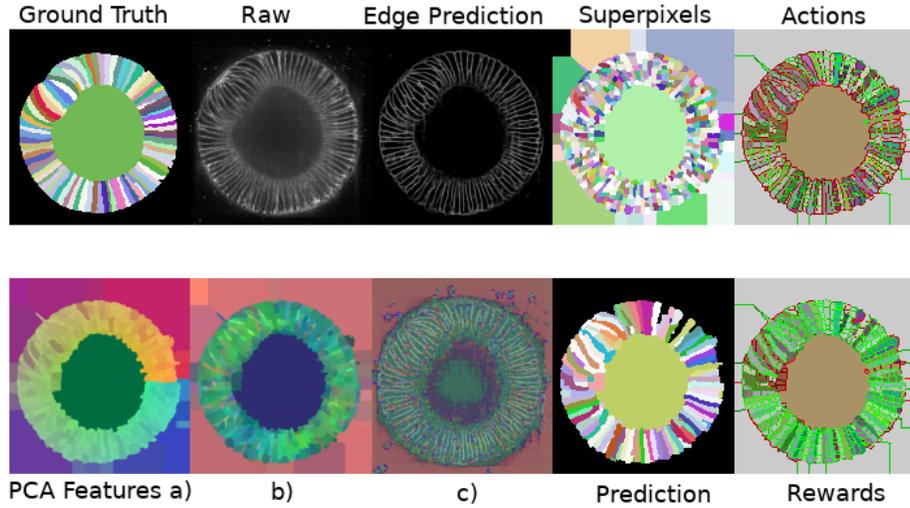


Figure 5: Microscopy dataset experiment. Top left to right: ground truth segmentation; raw data; edge map; superpixel over-segmentation; visualization for the actions on every edge, where a merge action is displayed in green and a split action in red. Bottom left to right: a) handcrafted features; b) learned features accumulated on superpixels; c) learned features projected to their first 3 PCA components shown as RGB; the segmentation resulting from the Multicut on the predicted edge weights; visualization of the rewards, where light green shows high rewards and dark red low rewards.

271 knowledge of the embryo width. An image of the weights for different locations in the image can be  
 272 found in the appendix.

273 More formally the edge rewards  $r_{edge}$  are calculated as follows. For each edge, we define the distance  
 274  $h$  between the edge and the center of the image as the average distance of the incident objects' center  
 275 of mass and the center  $c$  of the image.  $j$  is the approximate radius of the circle that lies within the  
 276 foreground and  $m$  is the maximal distance between  $c$  and the image boarder. Let further  $\mathcal{K}(\cdot)$  be the  
 277 Gaussian kernel function. Then  $r_{edge}$  yields

$$r_{bg} = \begin{cases} \mathcal{K}\left(\frac{\|h-c\|}{\gamma}\right) (1-a), & \text{if } h \leq j \\ \mathcal{K}\left(\frac{\|m-h\|}{\eta}\right) (1-a), & \text{otw} \end{cases} \quad (9)$$

$$r_{fg} = \mathcal{K}\left(\frac{\|h-j\|}{\delta}\right) \max(r_{o1}, r_{o2}) \quad (10)$$

$$r_{edge} = r_{fg} + r_{bg} \quad (11)$$

278 Here  $\gamma, \eta, \delta$  are normalization constants. Equation 9 first determines the background probability for  
 279 an edge by the kernel values.  $1-a$  constitutes a reward that directly favors merges which is scaled  
 280 by the background probability. For each edge,  $r_{o1}$  and  $r_{o2}$  are the rewards corresponding to the two  
 281 objects connected to that edge. The object rewards are given by fitting a rotated bounding box to the  
 282 object and then compare rotation and dimensions to template values.

283 Note that in this experiment no self-supervised pretraining is used for the node features in the  
 284 agent's GNNs. Unlike the "Circles" dataset, all objects in these images have very similar intensity  
 285 distributions and can only be separated through the detection of boundaries between them. Instead  
 286 of the pretraining, we experiment with using a few hand-crafted features like the polar coordinate  
 287 of the node's respective superpixel's center of mass with respect to the coordinate system sitting  
 288 at the center of the image as well as the superpixel's mass, and with learning other features by  
 289 back-propagation from the agent. The handcrafted features are normalized, concatenated to the  
 290 learned features and used as input to the GNN. The projection of the first 3 PCA components of these  
 291 features into RGB space is shown in Fig. 5 respectively for learned feature maps, their projection  
 292 to node features through the accumulation procedure and finally the concatenation of those and the

Table 1: Quantitative evaluation on the microscopy dataset. Note that the projection of superpixels to the ground truth (sp gt) sets an upper (lower for VI) bound for our method. We use Symmetric Best Dice as well as the Variation of Information metric to compare all results on the validation set.

Method	SBD	VI merge	VI split
sp gt	$0.656 \pm 0.019$	$0.672 \pm 0.061$	$0.594 \pm 0.028$
ours + augmentation noise	$0.508 \pm 0.031$	$1.233 \pm 0.156$	$1.060 \pm 0.258$
ours	$0.482 \pm 0.020$	$0.839 \pm 0.118$	$1.374 \pm 0.357$
ours without edges	$0.446 \pm 0.041$	$0.953 \pm 0.212$	$0.994 \pm 0.200$
ours only handcrafted	$0.408 \pm 0.087$	$0.987 \pm 0.101$	$1.536 \pm 0.410$
edge + mc [47]	$0.283 \pm 0.023$	$3.019 \pm 0.040$	$0.342 \pm 0.045$
contrastive [28]	$0.215 \pm 0.009$	$1.155 \pm 0.037$	$3.285 \pm 0.084$
contrastive + edge [28]	$0.248 \pm 0.014$	$1.229 \pm 0.048$	$3.336 \pm 0.073$

293 handcrafted features. Note that the learned features converge to a representation which resembles a  
 294 semantic segmentation of boundaries in the image.

295 We train the complete setup for Method 1 end-to-end on a Nvidia GeForce RTX 3090 GPU for 4  
 296 days. For comparison we keep the model which achieved the highest reward on the test set. This  
 297 makes training as well as the validation independent from ground truth annotations. The evolution  
 298 of the rewards on the validation set for different random seeds is shown in the Appendix. All of the  
 299 conducted trainings show a stride for high rewards regardless of different random seeding.

300 For the validation scores we use the variation of information (VI) for both input combinations (merge  
 301 and split) and the Symmetric Best Dice score. To show the influence of the imperfect superpixels on  
 302 the final clustering, we project the superpixels to their respective ground truth clustering ("sp gt" in  
 303 Table 1) which sets an upper (lower in case of VI) bound for our method. In this study we use several  
 304 versions of our approach. In Table 1 (ours) refers to method 1 as described in section 4.2, (ours +  
 305 augmentation noise) is the same method but add some noise to the input data during training, (ours  
 306 without edges) is our method but without the additional edge prediction as an input and (ours only  
 307 handcrafted) is our method where we only use the handcrafted features as described in section 4.2.  
 308 We find that learned features significantly contribute to the performance of our method.

309 We compare to the following baseline approaches: *edge + mc*, which solves the Multicut graph  
 310 partitioning based on edge weights derived from boundary predictions used for superpixel creation,  
 311 *contrastive*, which predicts a pixel-wise embedding space that is clustered into instances using  
 312 k-means and for which the embeddings are trained using the discriminative loss function of [28] on  
 313 the ovules dataset from [47] and *contrastive + edge*, which is similar to *contrastive*, but receives the  
 314 [47] boundary predictions as additional input channel.

## 315 5 Discussion

316 We introduced an end-to-end instance segmentation algorithm which can exploit non-differentiable  
 317 loss functions and high-level prior information. Our RL approach is based on stateless actor-critic  
 318 and predicts the full segmentation at every step, allowing us to assign rewards to all objects and  
 319 reach stable convergence. The segmentation problem is formulated as graph partitioning; we design  
 320 a reward decomposition algorithm which maps object- and image-level rewards to sub-graphs for  
 321 localized supervision.

322 We performed proof-of-concept experiments to demonstrate the feasibility of our approach on  
 323 synthetic and real data and showed in particular that our setup can segment microscopy images  
 324 with no direct supervision other than high-level reasoning. In the future, we plan to explore other  
 325 problems and reward functions as well as a semi-supervised setup (briefly introduced in Appendix)  
 326 where we think our approach can be very beneficial. Furthermore, even in case of full supervision  
 327 with ample groundtruth, our RL-based formulation enables end-to-end instance segmentation with  
 328 direct object-level reasoning, which will allow for post-processing-aware training of the CNN which  
 329 predicts object boundaries or embeddings.

## 330 References

- 331 [1] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebastian Seung. Superhuman accuracy  
332 on the snemi3d connectomics challenge. *arXiv preprint arXiv:1706.00120*, 2017.
- 333 [2] Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic  
334 segmentation, 2021.
- 335 [3] S. Osher and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*.  
336 Springer New York, 2007. ISBN 9780387218106. URL [https://books.google.de/books?](https://books.google.de/books?id=ZWzrBwAAQBAJ)  
337 [id=ZWzrBwAAQBAJ](https://books.google.de/books?id=ZWzrBwAAQBAJ).
- 338 [4] Ricard Delgado-Gonzalo, Virginie Uhlmann, Daniel Schmitter, and Michael Unser. Snakes on  
339 a plane: A perfect snap for bioimage analysis. *IEEE Signal Processing Magazine*, 32(1):41–48,  
340 2014.
- 341 [5] Ozan Oktay, Enzo Ferrante, Konstantinos Kamnitsas, Mattias Heinrich, Wenjia Bai, Jose  
342 Caballero, Stuart A. Cook, Antonio de Marvao, Timothy Dawes, Declan P. O’Regan, Bernhard  
343 Kainz, Ben Glocker, and Daniel Rueckert. Anatomically constrained neural networks (acnns):  
344 Application to cardiac image enhancement and segmentation. *IEEE Transactions on Medical*  
345 *Imaging*, 37(2):384–395, 2018. doi: 10.1109/TMI.2017.2743464.
- 346 [6] Hoel Kervadec, Jose Dolz, Meng Tang, Eric Granger, Yuri Boykov, and Ismail Ben Ayed.  
347 Constrained-cnn losses for weakly supervised segmentation. *Medical Image Analysis*, 54:88–99,  
348 2019. ISSN 1361-8415. doi: <https://doi.org/10.1016/j.media.2019.02.009>.
- 349 [7] David Pfau and Oriol Vinyals. Connecting generative adversarial networks and actor-critic  
350 methods. *CoRR*, abs/1610.01945, 2016. URL <http://arxiv.org/abs/1610.01945>.
- 351 [8] Nikita Araslanov, Constantin Rothkopf, and Stefan Roth. Actor-critic instance segmentation.  
352 In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,  
353 2019.
- 354 [9] Viren Jain, Srinivas Turaga, Kevin Briggman, Moritz Helmstaedter, Winfried Denk, and Hyun-  
355 june Seung. Learning to agglomerate superpixel hierarchies. *Advances in Neural Information*  
356 *Processing Systems*, 24, 01 2011.
- 357 [10] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi,  
358 Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel.  
359 Value-decomposition networks for cooperative multi-agent learning, 2017.
- 360 [11] Drew Bagnell and Andrew Ng. On local rewards and scaling distributed reinforcement learning.  
361 In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing*  
362 *Systems*, volume 18. MIT Press, 2006. URL [https://proceedings.neurips.cc/paper/](https://proceedings.neurips.cc/paper/2005/file/02180771a9b609a26dcea07f272e141f-Paper.pdf)  
363 [2005/file/02180771a9b609a26dcea07f272e141f-Paper.pdf](https://proceedings.neurips.cc/paper/2005/file/02180771a9b609a26dcea07f272e141f-Paper.pdf).
- 364 [12] Huazhe Xu, Boyuan Chen, Yang Gao, and Trevor Darrell. Scoring-aggregating-planning:  
365 Learning task-agnostic priors from interactions and sparse rewards for zero-shot generalization.  
366 *CoRR*, abs/1910.08143, 2019. URL <http://arxiv.org/abs/1910.08143>.
- 367 [13] Elad Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Ryan Rifkin, and Gal Elidan. Scalable  
368 learning of non-decomposable objectives. In *Artificial intelligence and statistics*, pages 832–840.  
369 PMLR, 2017.
- 370 [14] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss:  
371 A tractable surrogate for the optimization of the intersection-over-union measure in neural  
372 networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*,  
373 pages 4413–4421, 2018.
- 374 [15] Yang Song, Alexander G. Schwing, Richard S. Zemel, and Raquel Urtasun. Training deep  
375 neural networks via direct loss minimization. *International Conference on Machine Learning*,  
376 2016.

- 377 [16] Cicero Nogueira dos Santos, Kahini Wadhawan, and Bowen Zhou. Learning loss functions for  
378 semi-supervised learning via discriminative adversarial networks, 2017.
- 379 [17] Josif Grabocka, Randolph Scholz, and Lars Schmidt-Thieme. Learning surrogate losses. *arXiv*  
380 *preprint arXiv:1905.10108*, 2019.
- 381 [18] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with  
382 star-convex polygons. In *International Conference on Medical Image Computing and Computer-*  
383 *Assisted Intervention*, pages 265–273. Springer, 2018.
- 384 [19] Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist  
385 algorithm for cellular segmentation. *Nature Methods*, 18(1):100–106, 2021.
- 386 [20] Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving  
387 deep image segmentation. In *Advances in Neural Information Processing Systems*,  
388 volume 32, 2019. URL [https://proceedings.neurips.cc/paper/2019/file/](https://proceedings.neurips.cc/paper/2019/file/2d95666e2649fc6e3af75e09f5adb9-Paper.pdf)  
389 [2d95666e2649fc6e3af75e09f5adb9-Paper.pdf](https://proceedings.neurips.cc/paper/2019/file/2d95666e2649fc6e3af75e09f5adb9-Paper.pdf).
- 390 [21] Constantin Pape, Alex Matskevych, Adrian Wolny, Julian Hennies, Giulia Mizzon, Marion  
391 Louveaux, Jacob Musser, Alexis Maizel, Detlev Arendt, and Anna Kreshuk. Leveraging domain  
392 knowledge to improve microscopy image segmentation with lifted multicuts. *Frontiers in*  
393 *Computer Science*, 1:6, 2019.
- 394 [22] Jeremy B Maitin-Shepard, Viren Jain, Michal Januszewski, Peter Li, and Pieter Abbeel. Combi-  
395 natorial energy learning for image segmentation. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon,  
396 and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Cur-  
397 ran Associates, Inc., 2016. URL [https://proceedings.neurips.cc/paper/2016/file/](https://proceedings.neurips.cc/paper/2016/file/31857b449c407203749ae32dd0e7d64a-Paper.pdf)  
398 [31857b449c407203749ae32dd0e7d64a-Paper.pdf](https://proceedings.neurips.cc/paper/2016/file/31857b449c407203749ae32dd0e7d64a-Paper.pdf).
- 399 [23] Jie Song, Bjoern Andres, Michael J Black, Otmar Hilliges, and Siyu Tang. End-to-end learning  
400 for graph decomposition. In *Proceedings of the IEEE/CVF International Conference on*  
401 *Computer Vision*, pages 10093–10102, 2019.
- 402 [24] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang.  
403 Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE/CVF*  
404 *International Conference on Computer Vision*, pages 642–651, 2019.
- 405 [25] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Rahaman, Anna Kreshuk, Ullrich Köthe,  
406 and Fred A Hamprecht. The mutex watershed and its objective: Efficient, parameter-free graph  
407 partitioning. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- 408 [26] Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D Bock,  
409 Albert Cardona, Graham W Knott, Stephen M Plaza, Louis K Scheffer, et al. Multicut brings  
410 automated neurite segmentation closer to human performance. *Nature methods*, 14(2):101–102,  
411 2017.
- 412 [27] Jan Funke, Fabian Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld,  
413 and Srinivas C Turaga. Large scale image segmentation with structured loss based deep learning  
414 for connectome reconstruction. *IEEE transactions on pattern analysis and machine intelligence*,  
415 41(7):1669–1680, 2018.
- 416 [28] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a  
417 discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.
- 418 [29] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation  
419 by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the*  
420 *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8837–8845, 2019.
- 421 [30] Alberto Bailoni, Constantin Pape, Steffen Wolf, Thorsten Beier, Anna Kreshuk, and Fred A  
422 Hamprecht. A generalized framework for agglomerative clustering of signed graphs applied to  
423 instance segmentation. *arXiv preprint arXiv:1906.11713*, 2019.

- 424 [31] Leland McInnes and John Healy. Accelerated hierarchical density based clustering. In *2017*  
425 *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 33–42. IEEE,  
426 2017.
- 427 [32] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis.  
428 *IEEE Transactions on pattern analysis and machine intelligence*, 24(5):603–619, 2002.
- 429 [33] Bjoern Andres, Thorben Kroeger, Kevin L Briggman, Winfried Denk, Natalya Korogod, Graham  
430 Knott, Ullrich Koethe, and Fred A Hamprecht. Globally optimal closed-surface segmentation  
431 for connectomics. In *European Conference on Computer Vision*, pages 778–791. Springer,  
432 2012.
- 433 [34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of*  
434 *the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- 435 [35] Marina Meilă. Comparing clusterings by the variation of information. In *Learning theory and*  
436 *kernel machines*, pages 173–187. Springer, 2003.
- 437 [36] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the*  
438 *American Statistical association*, 66(336):846–850, 1971.
- 439 [37] Brian W Kernighan and Shen Lin. An efficient heuristic procedure for partitioning graphs. *The*  
440 *Bell system technical journal*, 49(2):291–307, 1970.
- 441 [38] Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan,  
442 Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Soft actor-critic  
443 algorithms and applications. *CoRR*, abs/1812.05905, 2018. URL [http://arxiv.org/abs/](http://arxiv.org/abs/1812.05905)  
444 [1812.05905](http://arxiv.org/abs/1812.05905).
- 445 [39] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl.  
446 Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL [http://](http://arxiv.org/abs/1704.01212)  
447 [arxiv.org/abs/1704.01212](http://arxiv.org/abs/1704.01212).
- 448 [40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for  
449 biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL [http://arxiv.org/](http://arxiv.org/abs/1505.04597)  
450 [abs/1505.04597](http://arxiv.org/abs/1505.04597).
- 451 [41] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT  
452 Press, second edition, 2018. URL [http://incompleteideas.net/book/the-book-2nd.](http://incompleteideas.net/book/the-book-2nd.html)  
453 [html](http://incompleteideas.net/book/the-book-2nd.html).
- 454 [42] Jörg Hendrik Kappes, Markus Speth, Björn Andres, Gerhard Reinelt, and Christoph Schn.  
455 Globally optimal image partitioning by multicuts. In Yuri Boykov, Fredrik Kahl, Victor  
456 Lempitsky, and Frank R. Schmidt, editors, *Energy Minimization Methods in Computer Vision*  
457 *and Pattern Recognition*, pages 31–44, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.  
458 ISBN 978-3-642-23094-3.
- 459 [43] Allam Shehata Hassanein, Sherien Mohammad, Mohamed Sameer, and Mohammad Ehab Ragab.  
460 A survey on hough transform, theory, techniques and applications. *CoRR*, abs/1502.02160,  
461 2015. URL <http://arxiv.org/abs/1502.02160>.
- 462 [44] D’Arcy Wentworth Thompson. *On Growth and Form*. Canto. Cambridge University Press,  
463 1992. doi: 10.1017/CBO9781107325852.
- 464 [45] Sourabh Bhide, Ralf Mikut, Maria Leptin, and Johannes Stegmaier. Semi-automatic generation  
465 of tight binary masks and non-convex isosurfaces for quantitative analysis of 3d biological  
466 samples, 2020.
- 467 [46] Lucas von Chamier, Romain F Laine, Johanna Jukkala, Christoph Spahn, Daniel Krentzel, Elias  
468 Nehme, Martina Lerche, Sara Hernández-Pérez, Pieta K Mattila, Eleni Karinou, Séamus Holden,  
469 Ahmet Can Solak, Alexander Krull, Tim-Oliver Buchholz, Martin L Jones, Loïc A Royer,  
470 Christophe Leterrier, Yoav Shechtman, Florian Jug, Mike Heilemann, Guillaume Jacquemet,  
471 and Ricardo Henriques. Democratizing deep learning for microscopy with ZeroCostDL4Mic.  
472 *Nature Communications*, 4 2021.

- 473 [47] Adrian Wolny, Lorenzo Cerrone, Athul Vijayan, Rachele Tofanelli, Amaya Vilches Barro,  
 474 Marion Louveaux, Christian Wenzl, Sören Strauss, David Wilson-Sánchez, Rena Lymbouridou,  
 475 et al. Accurate and versatile 3d segmentation of plant tissues at cellular resolution. *Elife*, 9:  
 476 e57613, 2020.
- 477 [48] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a  
 478 discriminative loss function, 2017.
- 479 [49] Reuben R. Shamir, Yuval Duchin, Jinyoung Kim, Guillermo Sapiro, and Noam Harel.  
 480 Continuous dice coefficient: a method for evaluating probabilistic segmentations. *CoRR*,  
 481 abs/1906.11031, 2019. URL <http://arxiv.org/abs/1906.11031>.

## 482 Checklist

- 483 1. For all authors...
- 484 (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s  
 485 contributions and scope? [Yes]
- 486 (b) Did you describe the limitations of your work? [Yes]
- 487 (c) Did you discuss any potential negative societal impacts of your work? [No] It does not  
 488 have any negative societal impacts.
- 489 (d) Have you read the ethics review guidelines and ensured that your paper conforms to  
 490 them? [Yes]
- 491 2. If you are including theoretical results...
- 492 (a) Did you state the full set of assumptions of all theoretical results? [N/A] We do not  
 493 claim any theoretical assumptions.
- 494 (b) Did you include complete proofs of all theoretical results? [N/A] See above.
- 495 3. If you ran experiments...
- 496 (a) Did you include the code, data, and instructions needed to reproduce the main experi-  
 497 mental results (either in the supplemental material or as a URL)? [Yes] Yes, see the  
 498 Supplementary.
- 499 (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they  
 500 were chosen)? [Yes] See the Supplementary
- 501 (c) Did you report error bars (e.g., with respect to the random seed after running experi-  
 502 ments multiple times)? [Yes] The bars are reported w.r.t. to the samples within the best  
 503 seed, see Supplementary section for further details
- 504 (d) Did you include the total amount of compute and the type of resources used (e.g., type  
 505 of GPUs, internal cluster, or cloud provider)? [Yes]
- 506 4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
- 507 (a) If your work uses existing assets, did you cite the creators? [Yes]
- 508 (b) Did you mention the license of the assets? [Yes]
- 509 (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]  
 510 The link to the biological dataset is given in the Supplementary
- 511 (d) Did you discuss whether and how consent was obtained from people whose data  
 512 you’re using/curating? The dataset was provided from the other laboratory as a part of  
 513 collaboration.
- 514 (e) Did you discuss whether the data you are using/curating contains personally identifiable  
 515 information or offensive content? [No] No human data is used
- 516 5. If you used crowdsourcing or conducted research with human subjects...
- 517 (a) Did you include the full text of instructions given to participants and screenshots, if  
 518 applicable? [N/A]
- 519 (b) Did you describe any potential participant risks, with links to Institutional Review  
 520 Board (IRB) approvals, if applicable? [N/A]
- 521 (c) Did you include the estimated hourly wage paid to participants and the total amount  
 522 spent on participant compensation? [N/A]