
A Generalized Framework for Agglomerative Clustering of Signed Graphs applied to Instance Segmentation

Anonymous Author(s)

Affiliation

Address

email

Abstract

We propose a novel theoretical framework that generalizes algorithms for hierarchical agglomerative clustering to weighted graphs with both attractive and repulsive interactions between the nodes. This framework defines GASP, a Generalized Algorithm for Signed graph Partitioning, and allows us to explore many combinations of different linkage criteria and cannot-link constraints. We prove the equivalence of existing clustering methods to some of those combinations, and introduce new algorithms for combinations which have not been studied. An extensive comparison is performed to evaluate properties of the clustering algorithms in the context of instance segmentation in images, including robustness to noise and efficiency. We show how one of the new algorithms proposed in our framework outperforms all previously known agglomerative methods for signed graphs, both on the competitive CREMI 2016 EM segmentation benchmark and on the CityScapes dataset.

1 Introduction

In computer vision, the partitioning of weighted graphs has been successfully applied to such tasks as image segmentation, object tracking and pose estimation. Most graph clustering methods work with positive edge weights only, which can be interpreted as similarities or distances between the nodes. These methods are parameter-based and require users to specify the desired numbers of clusters or a termination criterion (e.g. spectral clustering or iterated normalized cuts) or even to add a seed for each object (e.g. seeded watershed or random walker).

Other graph clustering methods work with so-called *signed graphs*, which include both positive and negative edge weights corresponding to attraction and repulsion between nodes. The advantage of signed graphs over positive-weighted graphs is that balancing attraction and repulsion allows us to perform the clustering without defining additional parameters. This can be done optimally by solving the so-called *multicut optimization* or *correlation clustering* problem [31, 10]. This problem is NP-hard, but approximate solvers have already been proposed [7]. Besides, the general problem of graph partitioning can be solved approximately by greedy agglomerative clustering [35, 48, 79, 32].

Agglomerative clustering algorithms for signed graphs have clear advantages: they are parameter-free and efficient. Despite the fact that there exists a variety of these algorithms, no overarching study has so far been made to compare their robustness and efficiency or to provide guidelines for matching an algorithm to the partitioning problem at hand.

In this paper, we propose a novel theoretical framework that generalizes over agglomerative algorithms for signed graphs by linking them to hierarchical agglomerative clustering on positive-weighted

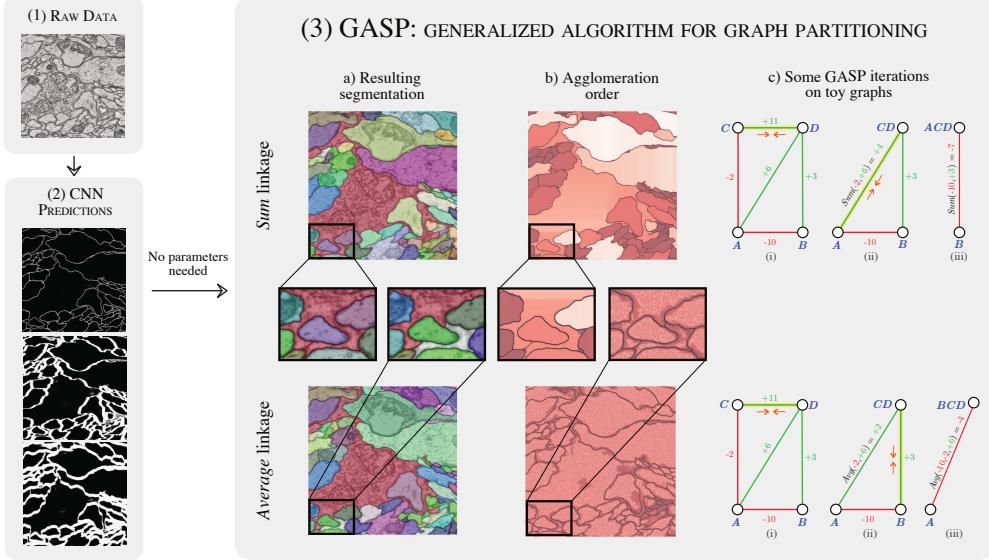


Figure 1: Pipeline description: (1) Raw data from the CREMI 2016 neuron-segmentation challenge. (2) Some short- and long-range predictions of our CNN model, where white pixels represent boundary evidence. (3) Outputs of two agglomerative algorithms included in our proposed generalized clustering framework, with *Sum* and *Average* linkage criteria. The final clustering / instance segmentation is shown in 3a, overlaid with the raw image. The agglomeration order in 3b shows which pairs of neighboring pixels were merged first (white), later on (brown), or never (black). In 3c, we also illustrate some iterations of the algorithms on toy graph examples with attractive/positive (green) and repulsive/negative (red) interactions. At each iteration, the yellow edge with highest interaction is contracted (orange arrows), until only negative edges are left in the graph.

34 graphs [44]. This framework defines an underlying basic algorithm and allows us to explore its
 35 combinations with different linkage criteria and *cannot-link constraints*. We then formally prove that
 36 some of the combinations correspond to existing clustering algorithms and introduce new algorithms
 37 for combinations which have not been explored yet.

38 We evaluate and compare these algorithms on *instance segmentation* - a computer vision task of
 39 assigning each pixel of an image to an object instance. We use a CNN to predict the edge weights of
 40 a graph such that each node represents a pixel of the image, similarly to [56, 47, 79], and provide
 41 these weights as input to the algorithms in our framework (see Fig. 1).

42 With our comparison experiments, performed both on 2D urban scenes from the CityScapes dataset
 43 and 3D electron microscopy image volumes of neurons, we evaluate the properties of the algorithms
 44 in our framework, focusing on their efficiency, robustness and tendency to over- or under-cluster.
 45 We show that one of the new algorithms derived from our framework, based on an average linkage
 46 criterion, outperforms the previously known agglomeration methods expressed in the framework. It
 47 also achieves competitive performance on the challenging CREMI 2016 segmentation benchmark.

48 2 Related work

49 **Proposal-based methods** have been highly successful in instance segmentation competitions like MS
 50 COCO [51], Pascal VOC2012 [19] and CityScapes [14]. They decompose the instance segmentation
 51 task into two steps that consists in generating object proposals and assigning to each bounding box
 52 a class and a binary segmentation mask [29, 81, 49, 43, 28, 9, 15, 50]. They commonly rely on
 53 Faster-RCNN [66] and can be trained end-to-end using non-maximum suppression. Other methods
 54 use instead recurrent models to sequentially generate instances one-by-one [68, 65].

55 **Proposal-free methods** adopt a bottom-up approach by directly grouping pixels into instances.
 56 Recently, there has been a growing interest for such methods that do not involve object detection,
 57 since, in certain types of data, object instances cannot be approximated by bounding boxes. For

example, the approach proposed in [37] uses a combinatorial framework for instance segmentation; SGN [52] sequentially group pixels into lines and then instances; a watershed transform is learned in [5] by also predicting its gradient direction, whereas the template matching [76] deploys scene depth information. Others use metric learning to predict high-dimensional associative pixel embeddings that map pixels of the same instance close to each other, while mapping pixels belonging to different instances further apart [20, 59, 16, 42]. Final instances are then retrieved by applying a clustering algorithm, like in the end-to-end trainable mean-shift pipeline of [40].

Edge detection also experienced recent progress thanks to deep learning, both on natural images [80, 39] and biological data [47, 73, 58, 13]. In neuron segmentation for connectomics, a field of neuroscience we also address in our experiments, boundaries are converted to final instances with subsequent postprocessing and superpixel-merging: some use loopy graphs [33, 41] or trees [58, 55, 53, 23, 77] to represent the region merging hierarchy; the lifted multicut [8] formulates the problem in a combinatorial framework, while flood-filling networks [30] eliminate superpixels by training a recurrent CNN to perform region growing one region at the time. A structured learning approach was also proposed in [25, 75].

Agglomerative graph clustering has often been applied to instance segmentation [67, 54, 71], because of its efficiency as compared to other top-down approaches like graph cuts. Novel termination criteria and merging strategies have often been proposed: the agglomeration in [57] deploys fixed sets of merge constraints; ultrametric contour maps [3] combine an oriented watershed transform with an edge detector, so that superpixels are merged until the ultrametric distance exceeds a learned threshold; the popular graph-based method [21] stops the agglomeration when the merge costs exceed a measure of quality for the current clusters. The optimization approach in [36] performs greedy merge decisions that minimize a certain energy, while other pipelines use classical HAC linkage criteria, e.g. average linkage [56, 47], median [25] or a linkage learned by a random forest classifier [60, 38].

Clustering of signed graphs has the goal of partitioning a graph with both attractive and repulsive cues. Finding an optimally balanced partitioning has a long history in combinatorial optimization [26, 27, 11]. NP-hardness of the *correlation clustering* problem was shown in [6], while the connection with graph multicuts was made by [17]. Modern integer linear programming solvers can tackle problems of considerable size [2], but accurate approximations [61, 7, 82], greedy agglomerative algorithms [48, 78, 35, 32] and persistence criteria [46, 45] have been proposed for even larger graphs.

This work reformulates the clustering algorithms of [48, 79, 35] in a generalized framework and adopt ideas from the proposal-free methods [56, 79, 47] to predict long-range relationships between pixels.

3 Generalized framework for agglomerative clustering of signed graphs

In this section, we first define notation and then introduce one of our main contributions: a signed graph partitioning algorithm (Sec. 3.2) that can be seen as a generalization of several existing and new clustering algorithms (Sec. 3.3).

3.1 Notation and graph formalism

We consider an undirected simple edge-weighted graph $\mathcal{G}(V, E, w^+, w^-)$ with both attractive and repulsive edge attributes. In computer vision applications, the nodes can represent either pixels, superpixels or voxels. We call the set Π a *clustering* or *partitioning* with K clusters if $V = \cup_{S \in \Pi} S$, $S \cap S' = \emptyset$ for different clusters $S, S' \in \Pi$ and every cluster $S \in \Pi$ induces a connected subgraph of \mathcal{G} . We also denote as S_u the cluster associated with node u . The weight function $w^+ : E \rightarrow \mathbb{R}^+$ associates to every edge a positive scalar attribute $w_e^+ \in \mathbb{R}^+$ representing a merge affinity or a similarity measure: the higher this number, the higher the inclination of the two incident vertices to be assigned to the same cluster¹. On the other hand, $w^- : E \rightarrow \mathbb{R}^+$ associates to each edge a split tendency $w_e^- \in \mathbb{R}^+$: the higher this weight, the more the incident vertices would like to be in different clusters. Graphs of the type $\mathcal{G}(V, E, w^+, w^-)$ are also often defined as *signed graphs* $\mathcal{G}(V, E, w)$,

¹Note that other formalisms for positively weighted graphs associate distances to the edges, thus, the *lower* the edge weight, the higher the attraction between the two linked nodes, contrary to our definition of w^+ .

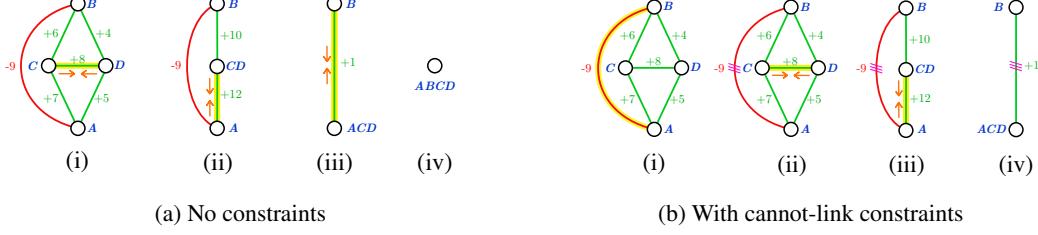


Figure 2: Some iterations of the generalized algorithm (using *Sum* linkage criteria) with and without adding cannot-link constraints. The graph has both attractive (green) and repulsive (red) edges and cannot-link constraints are shown with triple violet bars on the edges. We note that when constraints are enforced, the final clustering is given by two clusters instead of only one.

Algorithm 1 GASP: generalized algorithm for signed graph partitioning

Input: Graph $\mathcal{G}(V, E, w^+, w^-)$; linkage criterion \mathcal{W} ; boolean `addCannotLinkConstraints`
Output: Final clustering Π

```

1: Initialize clustering  $\Pi = \{\{v_1\}, \dots, \{v_{|V|}\}\}$  with each node in its own cluster
2: Initial interactions between nodes given by  $w_e = w_e^+ - w_e^-$ 
3: repeat
4:   Select pair of clusters  $S_u, S_v \in \Pi$  with highest absolute interaction  $|\mathcal{W}(S_u, S_v)|$ 
5:   if  $[\mathcal{W}(S_u, S_v) > 0]$  and  $[S_u, S_v \text{ are not constrained}]$  then
6:     Merge cluster  $S_u$  with  $S_v$ : update interactions and cannot-link constraints with all their neighbors
7:   else if  $[\mathcal{W}(S_u, S_v) \leq 0]$  and addCannotLinkConstraints then
8:     Add CannotLink Constraint between clusters  $S_u$  and  $S_v$ 
9:   until  $[\text{all interactions between clusters are repulsive}]$  or  $[\text{all adjacent clusters have cannot-link constraints}]$ 
10:  return  $\Pi$ 

```

108 featuring positive and negative edge weights $w_e \in \mathbb{R}$. Following the theoretical considerations in
109 [46], we define these signed weights as $w_e = w_e^+ - w_e^-$. Some approaches directly compute w_e ,
110 whereas others compute w_e^+ and w_e^- separately. In this formalism, graphs with purely attractive
111 interactions are a special case of $\mathcal{G}(V, E, w)$ with $w_e \geq 0, \forall e \in E$.

112 **Inter-cluster interaction** We call two clusters S_u, S_v *adjacent* if there exists at least one edge
113 $e_{ts} \in E$ connecting a node $t \in S_u$ to a node $s \in S_v$. In hierarchical agglomerative clustering, the
114 interaction $\mathcal{W}(S_u, S_v)$ between the two clusters is usually defined as a function $\mathcal{W} : \Pi \times \Pi \rightarrow \mathbb{R}$,
115 named *linkage criterion*, depending on the weights of *all* edges connecting clusters S_u and S_v , i.e.
116 $(S_u \times S_v) \cap E$. All the linkage criteria tested in this article are listed and defined in Table 1.

117 **3.2 GASP: generalized algorithm for signed graph partitioning**

118 In Algorithm 1, we provide simplified pseudo-code for the proposed GASP algorithm. GASP
119 implements a bottom-up approach that starts by assigning each node to its own cluster and then
120 iteratively merges pairs of adjacent clusters. The algorithm has two variants. The first one, with
121 `addCannotLinkConstraints=False`, starts by merging clusters with the strongest attractive in-
122 teraction and stops once the remaining clusters share only mutual by repulsive interactions (see
123 iterations on toy graphs in block 3 of Fig. 1). After each merging iteration, the interaction between
124 the merged cluster and its neighbors is updated according to one of the linkage criteria $\mathcal{W}(S_u, S_v)$
125 listed in Table 1.

126 In the second variant, when `addCannotLinkConstraints=True`, Algorithm 1 also introduces
127 *cannot-link constraints*, which represent mutual exclusion relationships between pairs of nodes that
128 cannot be associated with the same cluster in the final clustering. This variant selects the pair of
129 clusters with the highest absolute interaction $|\mathcal{W}(S_u, S_v)|$, so that the most attractive and the most
130 repulsive pairs are analyzed first (see example in Fig. 2(b)). If the interaction is repulsive, then the
131 two clusters are constrained and its members can never merge in subsequent steps. If the interaction
132 is attractive, then the clusters are merged, provided that they were not previously constrained. The
133 algorithm terminates when all the remaining clusters are constrained.

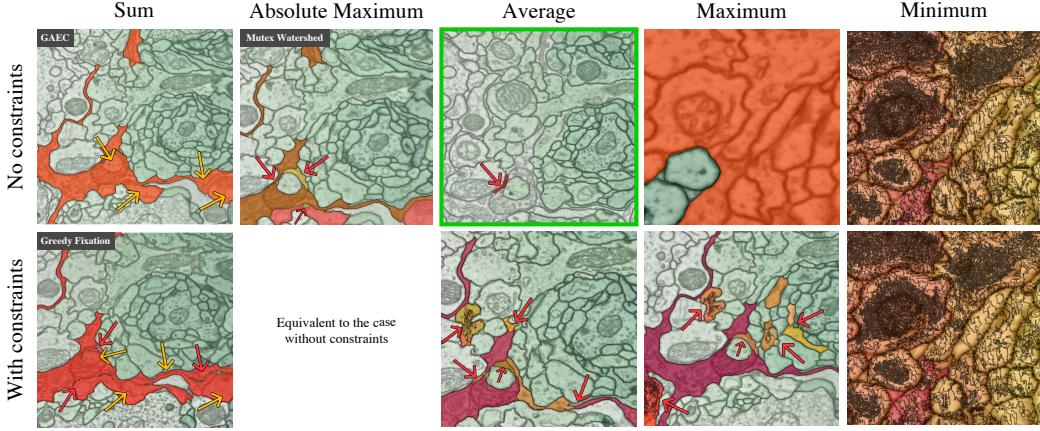


Figure 3: Failure cases of GASP with different linkage criteria highlighted on some difficult parts of the CREMI Challenge data. The main *wrongly* segmented regions are highlighted in different warm colors. Note that the data is 3D, hence the same color could be assigned to parts of segments that appear disconnected in 2D. Red arrows point to wrongly split regions. Yellow arrows point out merge errors. The *Average* linkage without cannot-link constraints returned the best segmentation.

134 In Appendix 7.1, we comment on the algorithm’s computational complexity $\mathcal{O}(N^2 \log N)$ and
 135 present our efficient implementation given by the edge contraction Algorithm 2 using a priority
 136 queue.

137 3.3 GASP with different linkage criteria: new and existing algorithms

138 Our main contribution is the generalized algorithm for signed graph partitioning, short GASP, that
 139 encompasses several known and novel agglomerative algorithms on display in Table 1. In our
 140 framework, individual algorithms are differentiated by the linkage criterion employed. We review
 141 them in the following paragraphs.

142 In the special case of an unsigned graph with only positive interactions, i.e. $w_e^- = 0$ and $w_e \geq 0$
 143 $\forall e \in E$, the algorithm performs a standard agglomerative hierarchical clustering by returning only a
 144 single cluster and a hierarchy of clusters defined by the order in which the clusters are merged (see
 145 Table 1, unsigned graphs).

GASP linkage criteria $\mathcal{W}(S_u, S_v)$	Unsigned Graphs	Signed Graphs	
		No Constraints	With Constraints
Sum: $\sum_{e \in E_{uv}} w_e$	Sum Linkage Hier. Aggl. Clust.	GAEC [35]	Greedy Fixation [48]
Absolute Max: w_e with $e = \arg \max_{t \in E_{uv}} w_t $	Single Linkage Hier. Aggl. Clust.	Mutex Watershed [79]	Mutex Watershed [79]
Average: $\sum_{e \in E_{uv}} w_e / E_{uv} $	Average Linkage Hier. Aggl. Clust.	NEW	NEW
Max: $\max_{e \in E_{uv}} w_e$	Single Linkage Hier. Aggl. Clust.	NEW	NEW
Min: $\min_{e \in E_{uv}} w_e$	Complete Linkage Hier. Aggl. Clust.	NEW	NEW

Table 1: Existing and new clustering algorithms that can be reformulated as special cases of the proposed generalized algorithm for signed graph partitioning GASP, given a linkage criterium, a type of graph (signed or unsigned) and the optional use of cannot-link constraints. The set E_{uv} is defined as the set of all edges connecting cluster S_u to cluster S_v , i.e. $E_{uv} = (S_u \times S_{v \neq u}) \cap E$.

146 Given a graph with both attractive and repulsive cues, an edge contraction algorithm with a sum
147 update rule was pioneered in [48, 35] (Table 1, *Sum* linkage). The authors present both a version
148 with cannot-link constraints and one without, and then compare them with other greedy local-search
149 algorithms approximating the multicut optimization problem. The Mutex Watershed [79] is another
150 signed graph partitioning algorithm that introduces dynamical cannot-link constraints. In Appendix
151 7.2 we prove that, surprisingly, it can also be seen as an efficient implementation of GASP with
152 *Absolute maximum* linkage (see def. in Table 1). We observe that in this case GASP returns the same
153 clustering with or without enforcing cannot-link constraints. On the other hand, to our knowledge,
154 *Average*, *Max* or *Min* linkage criteria have never been used for signed graph agglomerative algorithms
155 or been combined with cannot-link constraints.

156 Apart from the linkage criteria defined in Table 1, additional ones were proposed in the literature: [60]
157 for example uses a learned approach where a random forest classifier updates the cluster interactions
158 depending on predefined edge and node features; other approaches introduce a weight regularization
159 depending on the size of the clusters [21, 32], whereas [25] uses a *quantile* linkage criteria by
160 populating a histogram for each inter-cluster interaction. In our experiments, we decided to focus on
161 the linkage criteria listed in Table 1, since they represent the most common options.

162 4 Experiments on neuron segmentation

163 We first evaluate and compare the agglomerative clustering algorithms described in the generalized
164 framework on the task of neuron segmentation in electron microscopy (EM) image volumes. This
165 application is of key interest in connectomics, a field of neuro-science with the goal of reconstructing
166 neural wiring diagrams spanning complete central nervous systems. Currently, only proof-reading or
167 manual tracing yields sufficient accuracy for correct circuit reconstruction [72], thus further progress
168 is required in automated reconstruction methods.

169 EM segmentation is commonly performed by first predicting boundary pixels [8, 13] or undirected
170 affinities [79, 47, 25], which represent how likely it is for a pair of pixels to belong to the same neuron
171 segment. The affinities do not have to be limited to immediately adjacent pixels. Thus, similarly to
172 [47], we train a CNN to predict both short- and long-range affinities and use them as edge weights of
173 a 3D grid graph, where each node represents a pixel/voxel of the volume image.

174 4.1 Data: CREMI challenge

175 We evaluate all algorithms in the proposed framework on the competitive CREMI 2016 EM Segmen-
176 tation Challenge [24] that is currently the neuron segmentation challenge with the largest amount
177 of training data available. The dataset comes from serial section EM of *Drosophila* fruit-fly tissue
178 and consists of 6 volumes of 1250x1250x125 voxels at resolution 4x4x40nm, three of which present
179 publicly available training ground truth. The results submitted to the leaderboard are evaluated
180 using the CREMI score (<https://cremi.org/leaderboard/>), based on the Adapted Rand-Score
181 (Rand-Score) and the Variation of Information Score [4]. In Appendix 7.4, we provide more details
182 about the training of our CNN model, inspired by work of [47, 25].

183 4.2 Results and discussion

184 **Comparison of linkage criteria** Table 2 shows how the agglomerative algorithms derived from our
185 framework compare to each other. For a simple baseline, we also include a segmentation produced
186 by thresholding the affinity predictions (THRESH). GASP with *Average* linkage, representing one of
187 the new algorithms derived from our generalized framework, significantly outperformed all other
188 previously proposed agglomerative methods like GAEC (GASP Sum) [35], Greedy Fixation (GASP
189 Sum + Constraints) [48] or Mutex Watershed (GASP Abs. Max.) [79]. The competitive performance
190 of this simple parameter-free algorithm is also reported in Table 3, showing the current leader-board
191 of the challenge: all entries, apart from GASP, employ superpixel-based post-processing pipelines,
192 several of which rely on the lifted multicut formulation of [8] that uses several random forests to
193 predict graph edge weights, relying not only on information derived from affinity maps but also
194 raw data and shape information. Note that the test volumes contain several imaging artifacts that
195 make segmentation particularly challenging and might profit from more robust edge statistics of
196 super-pixel based approaches. On the other hand, the fact that our algorithm can operate on pixels

	CREMI-Score (lower is better)		CREMI-Score (lower is better)
GASP Average	0.226	Our CNN + DTWS + LMC	0.221
GASP Sum + Constraints [48]	0.282	PNI CNN [47]	0.228
GASP Abs. Max. [79]	0.322	Our CNN + GASP Average	0.241
GASP Max. + Constraints	0.324	MALA CNN + MC [25]	0.276
GASP Sum [35]	0.334	CRU-Net [83]	0.566
GASP Average + Constraints	0.563	LFC [62]	0.616
THRESH	1.521		

Table 2: CREMI-Scores achieved by different linkage criteria and thresholding. All methods use the affinity predictions from our CNN as input. Scores are averages over the three CREMI training datasets.

Table 3: Current leading entries in the CREMI challenge leaderboard [24] (May 2019). The scores are averages of the three test datasets.

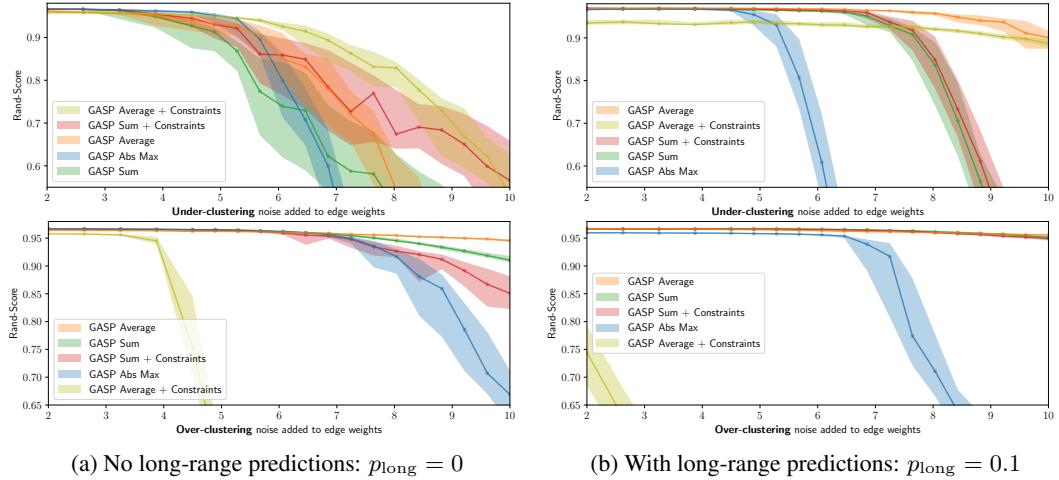


Figure 4: GASP sensitivity to noise: *Average* linkage proved to be the most robust. Performances are given by Rand-Score (higher is better) depending on the amount of noise added to the CNN predictions. Solid lines represent median values over 30 experiments. Values between the 25th and the 75th percentile are shown in shaded areas. The two sets of experiments using under- and over-clustering noise are summarized in the plots at the top and at the bottom, respectively (see Appendix 7.6 for more details). For each experiment, some of the long-range CNN predictions were randomly selected with probability p_{long} and added as long-range edges to the pixel grid-graph.

197 directly removes the parameter tuning necessary to obtain good super-pixels and can also avoid errors
 198 that result from wrong superpixels that cannot be fixed during later agglomeration. In Appendix 7.5,
 199 we provide more details about how we scaled up GASP to the full datasets. Appendix Table 5 lists
 200 the performances and the run-times for all tested GASP linkage.

201 **Noise experiments** Additionally, we conduct a set of experiments where the CNN predictions
 202 are perturbed by structured noise, in order to highlight the properties of each GASP variant and
 203 perform an in-depth comparison that is as quantitative as possible. Appendix 7.6 introduces the type
 204 of spatially correlated noise that allowed us to perturb the CNN outputs by introducing simulated
 205 additional artifacts like missing or false positive boundary evidence. Fig. 4 summarizes our 12000
 206 noise experiments: we focus on the best performing linkage criteria, i.e. *Average*, *Sum* and *Abs
 207 Max*, and test them with different amount of noise. In these experiments, we also want to assess
 208 how beneficial it is to use long-range CNN predictions in the agglomeration. Thus, we perform a set
 209 of simulations without adding long-range connections to the grid-graph and another set where we
 210 introduce them with a 10% probability².

211 **Average and Abs Max linkage** Our findings confirm that GASP with *Average* linkage criterion
 212 represents the most robust algorithm tested and the one that benefits the most from using the long-

²We also performed experiments adding all the long-range predictions given by the CNN model, but we did not note major differences when using only 10% of them. Adding this fraction is usually sufficient to improve the scores.

Pipeline	Agglomeration method	Use constraints:	
		No	YES
DWT [5]	-	21.2	-
SGN [52]	-	29.2	-
Mask RCNN [29]	-	31.5	-
	GASP Average	34.3	33.9
GMIS [56]	MultiStepHAC [56]	33.0	-
	GASP Abs. Max. [79]	32.1	32.1
	GASP Sum [35, 48]	31.3	31.9

Table 4: Average Precision (AP) scores (higher is better) on the CityScapes validation set.

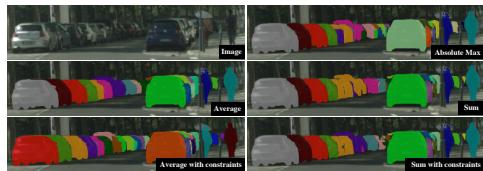


Figure 5: Instance segmentations given by different GASP linkage criteria on CityScapes

range CNN predictions. On the other hand, it is not a surprise that the *Abs Max* statistic proposed by [79] is less robust to noise than the *Average* linkage, but, as we show in the Appendix Table 5, *Abs Max* represents a valid and considerably faster option. Adding long-range connections to the graph is generally helpful, but when many of them carry repulsive weights, then GASP with cannot-link constraints shows a clear tendency to over-cluster.

Sum linkage All our experiments show that GASP with *Sum* linkage is the algorithm with the highest tendency to under-cluster and incorrectly merge segments (see Fig. 3 for an example). This property is related to the empirical observation that a *Sum* statistic tends to grow clusters one after the other, as shown in Fig. 1 by the quite unique agglomeration order of the *Sum* statistic. An intuitive explanation of this fact is the following: initially, most of the intra-cluster nodes present similar attractive interactions between each others; when the two nodes sharing the most attractive interaction are merged, there is a high chance that they both share an attractive interaction with a common neighboring node, so the new interaction with this common neighbor will be immediately assigned to a high priority in the agglomeration, given by the sum of two high weights; this usually starts a “chain reaction”, where only a single cluster is aggregated at the beginning. On the other hand, as we also see in Fig. 1, other linkage criteria like *Average* or *Abs Max* grow clusters of similar sizes in parallel and accumulate in this way much more reliable inter-cluster statistics.

5 Experiments on CityScapes

We also evaluate the performances of GASP on the CityScapes dataset [14], which consists of 5000 street-scene images: 2975 for training, 500 for validation and 1525 for testing. See Appendix 7.7 for more details on how we fine-tuned the state-of-the-art proposal-free pipeline proposed in GMIS [56] by using a *Sørensen-Dice* loss, similarly to [79]. Results are summarized in Table 4 and Fig. 5: similarly to the previous experiments, GASP with *Average* linkage achieves the best scores, whereas other linkage criteria tend to over-cluster, like *Abs Max*, or under-cluster and merge instances, like *Sum*. The graph-merging algorithm proposed by [56] (MultiStepHAC) requires the user to tune several threshold parameters and it was probably tailored to the original affinities predicted by them, so it did not generalize well to our fine-tuned model and it achieved lower scores compared to the original AP value of 34.1 reported in [56]. Appendix - Table 6 includes the scores of all other tested GASP algorithms.

6 Conclusion

We have presented a novel unifying framework for agglomerative clustering of graphs with both positive and negative edge weights and we have shown that several existing clustering algorithms, e.g. the Mutex Watershed [79], can be reformulated as special cases of one underlying agglomerative algorithm. This framework also allowed us to introduce new algorithms, one of which, based on an *Average* linkage criterion, outperformed all the others: It proved to be a simple and remarkably robust approach to process short- and long-range predictions of a CNN applied to an instance segmentation task. On biological images, this simple average agglomeration algorithm can represent a valuable choice for user who is not willing to spend much time tuning complex task-dependent pipelines based on superpixels. In future work we plan to extend the comparison to other types of non-image graphs and explore common theoretical properties of the algorithms included in the framework.

253 **References**

- 254 [1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: ranking and clustering.
255 *Journal of the ACM (JACM)*, 55(5):23, 2008.
- 256 [2] B. Andres, T. Kroeger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Koethe, and F. A. Hamprecht.
257 Globally optimal closed-surface segmentation for connectomics. In *European Conference on Computer
258 Vision*, pages 778–791. Springer, 2012.
- 259 [3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation.
260 *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- 261 [4] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber,
262 D. Laptev, S. Dwivedi, J. M. Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms
263 for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.
- 264 [5] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE
265 Conference on Computer Vision and Pattern Recognition*, pages 5221–5229, 2017.
- 266 [6] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.
- 267 [7] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum
268 cost lifted multicut problem. In *European Conference on Computer Vision*, pages 715–730. Springer, 2016.
- 269 [8] T. Beier, C. Pape, N. Rahaman, T. Prange, S. Berg, D. D. Bock, A. Cardona, G. W. Knott, S. M. Plaza,
270 L. K. Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature
271 Methods*, 14(2):101, 2017.
- 272 [9] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In
273 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3470–3478,
274 2015.
- 275 [10] S. Chopra and M. R. Rao. On the multiway cut polyhedron. *Networks*, 21(1):51–89, 1991.
- 276 [11] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, 1993.
- 277 [12] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger. 3d u-net: learning dense
278 volumetric segmentation from sparse annotation. In *International conference on medical image computing
279 and computer-assisted intervention*, pages 424–432. Springer, 2016.
- 280 [13] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal
281 membranes in electron microscopy images. In *Advances in neural information processing systems*, pages
282 2843–2851, 2012.
- 283 [14] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and
284 B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE
285 conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- 286 [15] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In
287 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158,
288 2016.
- 289 [16] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss
290 function. *arXiv preprint arXiv:1708.02551*, 2017.
- 291 [17] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs.
292 *Theoretical Computer Science*, 361(2-3):172–187, 2006.
- 293 [18] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302,
294 1945.
- 295 [19] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes
296 (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- 297 [20] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance
298 segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017.
- 299 [21] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal
300 of computer vision*, 59(2):167–181, 2004.

- 301 [22] J. R. Finkel and C. D. Manning. Enforcing transitivity in coreference resolution. In *Proceedings of the*
 302 *46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 45–48. Association for Computational Linguistics, 2008.
- 304 [23] J. Funke, F. A. Hamprecht, and C. Zhang. Learning to segment: training hierarchical segmentation under
 305 a topological loss. In *International Conference on Medical Image Computing and Computer-Assisted
 306 Intervention*, pages 268–275. Springer, 2015.
- 307 [24] J. Funke, S. Saalfeld, D. Bock, S. Turaga, and E. Perlman. Cremi challenge. <https://cremi.org>., 2016.
 308 Accessed: 2019-05-15.
- 309 [25] J. Funke, F. D. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga. Large
 310 scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE
 311 transactions on pattern analysis and machine intelligence*, 2018.
- 312 [26] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical
 313 Programming*, 45(1-3):59–96, 1989.
- 314 [27] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*,
 315 47(1-3):367–387, 1990.
- 316 [28] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In
 317 *European Conference on Computer Vision*, pages 297–312. Springer, 2014.
- 318 [29] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international
 319 conference on computer vision*, pages 2961–2969, 2017.
- 320 [30] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka,
 321 W. Denk, and V. Jain. High-precision automated reconstruction of neurons with flood-filling networks.
Nature methods, 15(8):605, 2018.
- 323 [31] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schn. Globally optimal image partitioning by
 324 multicuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern
 325 Recognition*, pages 31–44. Springer, 2011.
- 326 [32] A. Kardoost and M. Keuper. Solving minimum cost lifted multicut problems by node agglomeration. In
 327 *ACCV 2018, 14th Asian Conference on Computer Vision*, Perth, Australia, 2018.
- 328 [33] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller,
 329 J. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron
 330 microscopy images. *Medical image analysis*, 22(1):77–88, 2015.
- 331 [34] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell system technical
 332 journal*, 49(2):291–307, 1970.
- 333 [35] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image
 334 and mesh graphs by lifted multicuts. In *Proceedings of the IEEE International Conference on Computer
 335 Vision*, pages 1751–1759, 2015.
- 336 [36] B. R. Kiran and J. Serra. Global-local optimizations by hierarchical cuts and climbing energies. *Pattern
 337 Recognition*, 47(1):12–24, 2014.
- 338 [37] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances
 339 with multicut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
 340 5008–5017, 2017.
- 341 [38] S. Knowles-Barley, V. Kaynig, T. R. Jones, A. Wilson, J. Morgan, D. Lee, D. Berger, N. Kasthuri,
 342 J. W. Lichtman, and H. Pfister. Rhoanet pipeline: Dense automatic neural annotation. *arXiv preprint
 343 arXiv:1611.06973*, 2016.
- 344 [39] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint
 345 arXiv:1511.07386*, 2015.
- 346 [40] S. Kong and C. C. Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE
 347 Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018.
- 348 [41] N. Krasowski, T. Beier, G. W. Knott, U. Koethe, F. A. Hamprecht, and A. Kreshuk. Improving 3d em
 349 data segmentation by joint optimization over boundary evidence and biological priors. In *2015 IEEE 12th
 350 International Symposium on Biomedical Imaging (ISBI)*, pages 536–539. IEEE, 2015.

- 351 [42] V. Kulikov, V. Yurchenko, and V. Lempitsky. Instance segmentation by deep coloring. *arXiv preprint*
 352 *arXiv:1807.10007*, 2018.
- 353 [43] L. Ladický, P. Sturges, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining
 354 object detectors and crfs. In *European conference on computer vision*, pages 424–437. Springer, 2010.
- 355 [44] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: 1. hierarchical
 356 systems. *The computer journal*, 9(4):373–380, 1967.
- 357 [45] J.-H. Lange, B. Andres, and P. Swoboda. Combinatorial persistency criteria for multicut and max-cut.
 358 *arXiv preprint arXiv:1812.01426*, 2018.
- 359 [46] J.-H. Lange, A. Karrenbauer, and B. Andres. Partial optimality and fast lower bounds for weighted
 360 correlation clustering. In *International Conference on Machine Learning*, pages 2898–2907, 2018.
- 361 [47] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung. Superhuman accuracy on the snemi3d connectomics
 362 challenge. *arXiv preprint arXiv:1706.00120*, 2017.
- 363 [48] E. Levinkov, A. Kirillov, and B. Andres. A comparative study of local search algorithms for correlation
 364 clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer, 2017.
- 365 [49] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In
 366 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367,
 367 2017.
- 368 [50] X. Liang, Y. Wei, X. Shen, Z. Jie, J. Feng, L. Lin, and S. Yan. Reversible recursive instance-level object
 369 segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
 370 633–641, 2016.
- 371 [51] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft
 372 coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer,
 373 2014.
- 374 [52] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In
 375 *Proceedings of the IEEE International Conference on Computer Vision*, pages 3496–3504, 2017.
- 376 [53] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen. A modular hierarchical approach to 3d electron
 377 microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, 2014.
- 378 [54] T. Liu, M. Seyedhosseini, and T. Tasdizen. Image segmentation using hierarchical merge tree. *IEEE*
 379 *transactions on image processing*, 25(10):4596–4607, 2016.
- 380 [55] T. Liu, M. Zhang, M. Javanmardi, N. Ramesh, and T. Tasdizen. Sshmt: Semi-supervised hierarchical
 381 merge tree for electron microscopy image segmentation. In *European Conference on Computer Vision*,
 382 pages 144–159. Springer, 2016.
- 383 [56] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu. Affinity derivation and graph merge for instance
 384 segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–703,
 385 2018.
- 386 [57] F. Malmberg, R. Strand, and I. Nyström. Generalized hard constraints for graph segmentation. In
 387 *Scandinavian Conference on Image Analysis*, pages 36–47. Springer, 2011.
- 388 [58] Y. Meirovitch, A. Matveev, H. Saribekyan, D. Budden, D. Rolnick, G. Odor, S. Knowles-Barley, T. R.
 389 Jones, H. Pfister, J. W. Lichtman, et al. A multi-pass approach to large-scale connectomics. *arXiv preprint*
 390 *arXiv:1612.02120*, 2016.
- 391 [59] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and
 392 grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.
- 393 [60] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical
 394 clustering to segment 2d and 3d images. *PloS one*, 8(8):e71715, 2013.
- 395 [61] C. Pape, T. Beier, P. Li, V. Jain, D. D. Bock, and A. Kreshuk. Solving large multicut problems for
 396 connectomics via domain decomposition. In *Proceedings of the IEEE International Conference on*
 397 *Computer Vision*, pages 1–10, 2017.
- 398 [62] T. Parag, F. Tschopp, W. Grisaitis, S. C. Turaga, X. Zhang, B. Matejek, L. Kamentsky, J. W. Lichtman,
 399 and H. Pfister. Anisotropic em segmentation by 3d affinity learning and agglomeration. *arXiv preprint*
 400 *arXiv:1707.08935*, 2017.

- 401 [63] K. Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985.
- 402 [64] K. Perlin. Noise hardware. *Real-Time Shading SIGGRAPH Course Notes*, 2001.
- 403 [65] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6656–6664, 2017.
- 405 [66] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- 407 [67] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2013.
- 409 [68] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European conference on computer vision*, pages 312–329. Springer, 2016.
- 411 [69] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- 414 [70] S. Saalfeld, R. Fetter, A. Cardona, and P. Tomancak. Elastic volume reconstruction from series of ultra-thin microscopy sections. *Nature methods*, 9(7):717, 2012.
- 416 [71] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE transactions on Image Processing*, 9(4):561–576, 2000.
- 418 [72] P. Schlegel, M. Costa, and G. S. Jefferis. Learning from connectomics on the fly. *Current opinion in insect science*, 24:96–105, 2017.
- 420 [73] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell detection with star-convex polygons. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–273. Springer, 2018.
- 423 [74] T. Sørensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species and its application to analyses of the vegetation on danish commons. *Biol. Skr.*, 5:1–34, 1948.
- 425 [75] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung. Maximin affinity learning of image segmentation. pages 1865–1873, 2009.
- 427 [76] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level semantic labeling. In *German Conference on Pattern Recognition*, pages 14–25. Springer, 2016.
- 429 [77] M. G. Uzunbas, C. Chen, and D. Metaxas. An efficient conditional random field approach for automatic and interactive neuron segmentation. *Medical image analysis*, 27:31–44, 2016.
- 431 [78] S. Wolf, A. Bailoni, C. Pape, N. Rahaman, A. Kreshuk, U. Köthe, and F. A. Hamprecht. The mutex watershed and its objective: Efficient, parameter-free image partitioning. *arXiv preprint arXiv:1904.12654*, 2019.
- 434 [79] S. Wolf, C. Pape, A. Bailoni, N. Rahaman, A. Kreshuk, U. Kothe, and F. Hamprecht. The mutex watershed: Efficient, parameter-free image partitioning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 546–562, 2018.
- 437 [80] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proc. ICCV'15*, pages 1395–1403, 2015.
- 438 [81] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1731–1743, 2012.
- 440 [82] J. Yarkony, A. Ihler, and C. C. Fowlkes. Fast planar correlation clustering for image segmentation. In *European Conference on Computer Vision*, pages 568–581. Springer, 2012.
- 442 [83] T. Zeng, B. Wu, and S. Ji. Deepem3d: approaching human-level performance on 3d anisotropic em image segmentation. *Bioinformatics*, 33(16):2555–2562, 2017.