
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 an underlying basic algorithm and allows us to explore its combinations with different linkage criteria and cannot-link constraints, i.e. mutual exclusion between clusters. We then prove how these different combinations correspond to existing clustering methods as well as new algorithms that we introduce and analyze. An extensive comparison study is performed to evaluate properties of the clustering algorithms, e.g. robustness and efficiency, by applying them to instance segmentation, which is the task of partitioning an image into distinct segments. We then 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 the CityScapes dataset.

1 Introduction

In computer vision, the clustering 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 a supervision in terms of seeds (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 using 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 [27, 9]. However, this problem is NP-hard, so instead of finding optimal solutions, several greedy agglomerative clustering algorithms were proposed.

Thus, agglomerative clustering algorithms for signed graphs have clear advantages: they are parameter free and efficient. Despite the fact that there exist a variety of these algorithms [30, 43, 69, 28], there have been no overarching studies comparing them and making it possible to choose the most appropriate algorithm for particular applications and assess their properties, e.g. robustness and efficiency.

In this paper, we propose a novel theoretical framework for generalizing over agglomerative algorithms for signed graphs by linking them to hierarchical agglomerative clustering, which is a popular

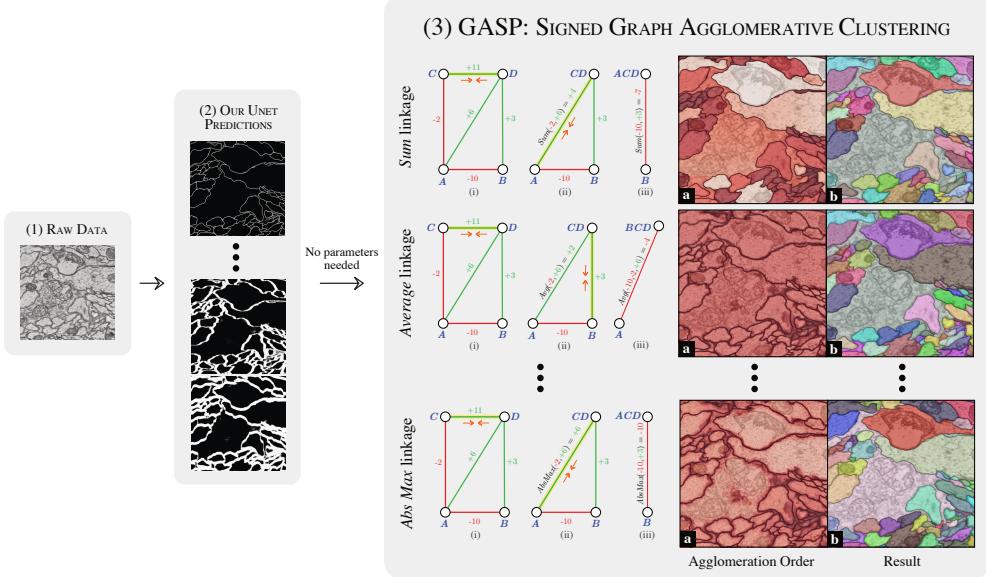


Figure 1: Pipeline description: **(1)** Raw data from the CREMI 2016 neuron-segmentation challenge. **(2)** Some short- and long-range predictions of our UNet model. **(3)** Outputs of three different agglomerative algorithms included in the proposed generalized framework. The agglomeration order in (a) shows which pairs of neighboring pixels were merged first (white), later on (brown), or never (black). The final clustering / instance segmentation is shown in (b). We also illustrate some iterations of the algorithms on toy graph examples with attractive (green) and repulsive (red) interactions.

35 bottom-up approach defining an hierarchy of clusters on positive-weighted graphs [39]. This frame-
 36 work defines an underlying basic algorithm and allows us to explore its combinations with different
 37 linkage criteria and *cannot-link constraints*. We then theoretically prove that different combinations
 38 correspond to existing clustering algorithms as well as new algorithms that we introduce and explore.

39 We evaluate and compare these algorithms on *instance segmentation*, which is a computer vision task
 40 consisting in assigning each pixel of an image to an object instance. We use a CNN to predict the
 41 edge weights of a graph such that each node represents a pixel of the image, similarly to [51, 42, 69],
 42 and provide these weights as input to the algorithms in our framework.

43 With our comparison experiments, performed both on 2D urban scenes and 3D electron microscopy
 44 image volumes of neurons, we evaluate the properties of the algorithms in our framework, focusing
 45 on their efficiency, robustness and tendency to over- or under-cluster. Our findings show that one of
 46 the new agglomerative algorithms introduced by our generalized framework, based on an average
 47 linkage criterion, consistently outperforms all other previously known agglomerative algorithms
 48 for signed graphs and achieves competitive scores on the challenging CREMI 2016 segmentation
 49 benchmark.

50 2 Related work

51 **Proposal-based methods** have been highly successful in instance segmentation competitions like MS
 52 COCO [46], Pascal VOC2012 [16] and CityScapes [12]. They decompose the instance segmentation
 53 task into two steps that consists in generating object proposals and assigning to each bounding box
 54 a class and a binary segmentation mask [25, 71, 44, 38, 24, 8, 13, 45]. They commonly rely on
 55 Faster-RCNN [59] and can be trained end-to-end using non-maximum suppression. Other methods
 56 use instead recurrent models to sequentially generate instances one-by-one [61, 58].

57 **Proposal-free methods** adopt a bottom-up approach by directly grouping pixels into instances.
 58 Recently, there has been a growing interest for such methods that do not involve object detection,
 59 since, in certain types of data, object instances cannot be approximated by bounding boxes. For
 60 example, the approach proposed in [32] uses a combinatorial framework for instance segmentation;

61 SGN [47] sequentially group pixels into lines and then instances; a watershed transform is learned in
62 [4] by also predicting its gradient direction, whereas the template matching [66] deploys scene depth
63 information. Others use metric learning to predict high-dimensional associative pixel embeddings
64 that map pixels of the same instance close to each other, while mapping pixels belonging to different
65 instances further apart [17, 54, 14, 37]. Final instances are then retrieved by applying a clustering
66 algorithm, like in the end-to-end trainable mean-shift pipeline of [35].

67 **Edge detection** also experienced recent progress thanks to deep learning, both on natural images
68 [70, 34] and biological data [42, 64, 53, 11]. In neuron segmentation for connectomics, a field
69 of neuroscience we also address in our experiments, boundaries are converted to final instances
70 with subsequent postprocessing and superpixel-merging: some use loopy graphs [29, 36] or trees
71 [53, 50, 48, 19, 67] to represent the region merging hierarchy; the lifted multicut [7] formulates the
72 problem in a combinatorial framework, while flood-filling networks [26] eliminate superpixels by
73 training a recurrent CNN to perform region growing one region at the time. A structured learning
74 approach was also proposed in [21, 65].

75 **Agglomerative graph clustering** has often been applied to instance segmentation [60, 49, 62],
76 because of its efficiency as compared to other top-down approaches like graph cuts. Novel termination
77 criteria and merging strategies have often been proposed: the agglomeration in [52] deploys fixed
78 sets of merge constraints; ultrametric contour maps [2] combine an oriented watershed transform
79 with an edge detector, so that superpixels are merged until the ultrametric distance exceeds a learned
80 threshold; the popular graph-based method [18] stops the agglomeration when the merge costs exceed
81 a measure of quality for the current clusters. The optimization approach in [31] performs greedy
82 merge decisions that minimize a certain energy, while other pipelines use classical HAC linkage
83 criteria, e.g. average linkage [51, 42], median [21] or a linkage learned by a random forest classifier
84 [55, 33].

85 **Clustering of signed graphs** has the goal of partitioning a graph with both attractive and repulsive
86 cues. Finding an optimally balanced partitioning has a long history in combinatorial optimization [22,
87 23, 10]. NP-hardness of the *correlation clustering* problem was shown in [5], while the connection
88 with graph multicuts was made by [15]. Modern integer linear programming solvers can tackle
89 problems of considerable size [1], but accurate approximations [56, 6, 72], greedy agglomerative
90 algorithms [43, 68, 30, 28] and persistence criteria [41, 40] have been proposed for even larger
91 graphs.

92 This work reformulates the clustering algorithms of [43, 69, 30] in a generalized framework and
93 adopt ideas from the proposal-free methods [51, 69, 42] to predict long-range relationships between
94 pixels.

95 3 Generalized Framework for Agglomerative Clustering of Signed Graphs

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

99 3.1 Notation and graph formalism

100 We consider an undirected simple edge-weighted graph $\mathcal{G}(V, E, w^+, w^-)$ with both attractive and
101 repulsive edge attributes. In computer vision applications, the nodes can represent either pixels,
102 superpixels or voxels. We call the set Π a *clustering* or *partitioning* with K clusters if $V = \cup_{S \in \Pi} S$,
103 $S \cap S' = \emptyset$ for different clusters $S, S' \in \Pi$ and every cluster $S \in \Pi$ induces a connected subgraph
104 of \mathcal{G} . We also denote as S_u the cluster associated with node u . The weight function $w^+ : E \rightarrow \mathbb{R}^+$
105 associates to every edge a positive scalar attribute $w_e^+ \in \mathbb{R}^+$ representing a merge affinity or a
106 similarity measure: the higher this number, the higher the inclination of the two incident vertices to
107 be assigned to the same cluster¹. On the other hand, $w^- : E \rightarrow \mathbb{R}^+$ associates to each edge a split
108 tendency $w_e^- \in \mathbb{R}^+$: the higher this weight, the more the incident vertices would like to be in different
109 clusters. Graphs of the type $\mathcal{G}(V, E, w^+, w^-)$ are also often defined as *signed graphs* $\mathcal{G}(V, E, w)$,
110 featuring positive and negative edge weights $w_e \in \mathbb{R}$. Following the theoretical considerations in

¹Note that other formalisms [REFS](#) 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^+ .

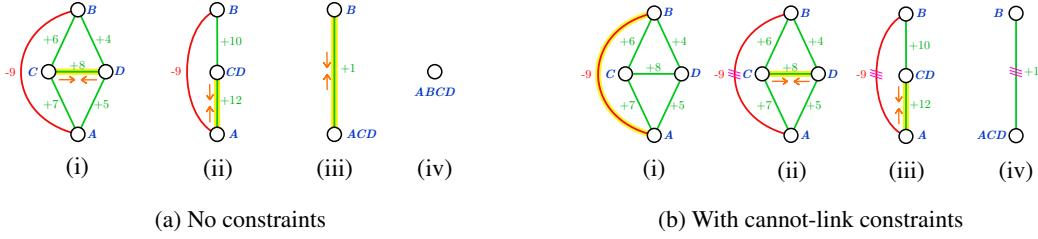


Figure 2: Some iterations of the generalized algorithm (using *Sum* linkage criteria) with and without adding cannot-link constraints. The graph has both attractive (green) and repulsive (red) edges and cannot-link constraints are shown with triple violet bars on the edges. We note that when constraints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

Algorithm 1 GASP: Generalized Algorithm for Agglomerative Signed Graph Partitioning

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

```

1: Initialize clustering  $\Pi = \{\{v_1\}, \dots, \{v_N\}\}$  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 [all interactions between clusters are repulsive] or [all adjacent clusters have cannot-link constraints]
10: return  $\Pi$ 

```

[41], we define these signed weights as $w_e = w_e^+ - w_e^-$. Some approaches directly compute w_e , whereas others compute w_e^+ and w_e^- separately. In this formalism, graphs with purely attractive interactions are a special case of $\mathcal{G}(\bar{V}, E, w)$ with $w_e \geq 0, \forall e \in E$.

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

3.2 GASP: Generalized Algorithm for Agglomerative Signed Graph Partitioning

In Algorithm 1, we provide a simplified pseudo-code for the proposed GASP algorithm that is based on a bottom-up approach starting with each node assigned to its own cluster and iteratively merging pairs of adjacent clusters. The algorithm has two variants, depending on the boolean value of the input option `addCannotLinkConstraints`. The first one, with `addCannotLinkConstraints=False`, starts by merging clusters with the strongest attractive interaction and it stops when the remaining clusters share only mutual repulsive interactions (see Fig. 1). After each merging iteration, the interaction between the merged cluster and its neighbors is updated according to one of the linkage criteria $\mathcal{W}(S_u, S_v)$ listed in Table 1.

In the second variant, when `addCannotLinkConstraints=True`, Algorithm 1 also introduces *cannot-link constraints*, which represent mutual exclusion relationships between pairs of nodes that cannot be associated with the same cluster in the final clustering. This variant selects the pair of clusters with the highest absolute interaction $|\mathcal{W}(S_u, S_v)|$, so that the most attractive and the most repulsive pairs are analyzed first (see example in Fig. 2(b)). If the interaction is repulsive, then the two clusters are constrained. If it is attractive, then they are merged, provided that they were not already previously constrained. The algorithm stops when all the remaining clusters are constrained.

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

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 [30]	Greedy Fixation [43]
Absolute Max: $w_e \text{ with } e = \arg \max_{t \in E_{uv}} w_t $	Single Linkage Hier. Aggl. Clust.	Mutex Watershed [69]	Mutex Watershed [69]
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: The table lists the existing clustering algorithms that can be reformulated as special cases of the proposed generalized algorithm GASP, given a linkage criteria, 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. $E_{uv} = (S_u \times S_v) \cap E$.

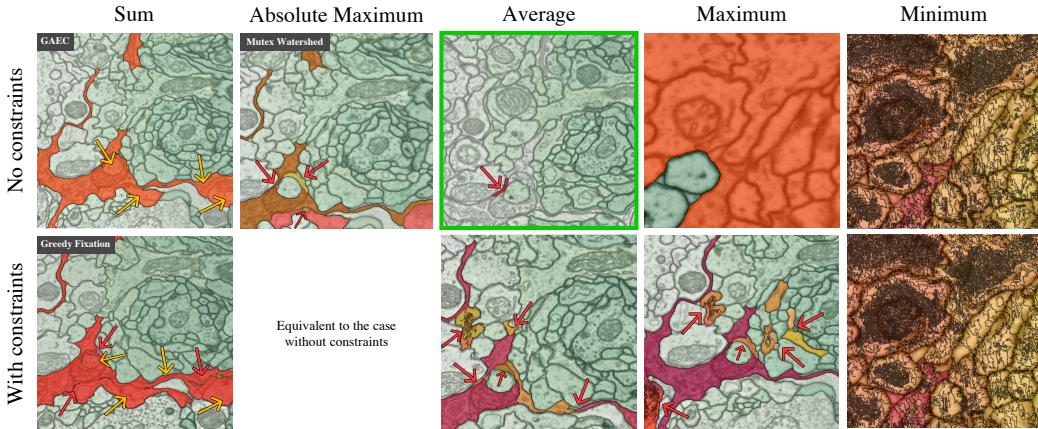


Figure 3: Failure cases of GASP with different linkage criteria highlighted on some difficult parts of the CREMI Challenge data. The main *wrongly* segmented regions are highlighted in different warm colors. Note that the data is 3D, hence the same color could be assigned to parts of segments that appear 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.

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

138 In the special case of an unsigned graph with only positive interactions, i.e. $w_e^- = 0$ and $w_e \geq 0$
 139 $\forall e \in E$, the algorithm performs a standard agglomerative hierarchical clustering by returning only a
 140 single cluster and a hierarchy of clusters defined by the order in which the clusters are merged (see
 141 bottom panel of Fig. 1 and Table 1, unsigned graphs).

142 Given a graph with both attractive and repulsive cues, an edge contraction algorithm with a sum
 143 update rule was already proposed in [43, 30] (Table 1, *Sum* linkage). They present both a version
 144 with cannot-link constraints and one without, and then compare them with other greedy local-search
 145 algorithms solving the multicut optimization problem. The Mutex Watershed [69] is another signed
 146 graph partitioning algorithm that introduces dynamical cannot-link constraints. In Appendix ?? we
 147 prove that it can be seen as an efficient implementation of GASP with *Absolute maximum* linkage
 148 (see def. in Table 1) and observe that in this case GASP returns the same clustering with or without
 149 enforcing cannot-link constraints. On the other hand, to our knowledge, *Average*, *Max* or *Min* linkage
 150 criteria have never been used for signed graphs agglomerative algorithms or been combined with
 151 cannot-link constraints.

152 Apart from the linkage criteria defined in Table 1, additional ones were proposed in the literature: [55]
153 for example uses a learned approach where a random forest classifier updates the cluster interactions
154 depending on predefined edge and node features; other approaches introduce a weight regularization
155 depending on the size of the clusters [18, 28], whereas [21] uses a *quantile* linkage criteria by
156 populating a histogram for each inter-cluster interaction. In our experiments, we decided to focus on
157 the linkage criteria listed in Table 1, since they represent the most common options.

158 4 Experiments on neuron segmentation

159 We first evaluate and compare the agglomerative clustering algorithms described in the generalized
160 framework on the task of neuron segmentation in electron microscopy (EM) image volumes. This
161 application is of key interest in connectomics, a field of neuro-science with the goal of reconstructing
162 neural wiring diagrams spanning complete central nervous systems. Currently, only proof-reading or
163 manual tracing yields sufficient accuracy for correct circuit reconstruction [63], thus further progress
164 is required in automated reconstruction methods.

165 EM segmentation is commonly performed by first predicting boundary pixels [7, 11] or undirected
166 affinities [69, 42, 21], which represent how likely it is for a pair of pixels to belong to the same neuron
167 segment. The affinities do not have to be limited to direct neighboring pixels. Thus, similarly to [42],
168 we train a CNN to predict both short- and long-range affinities and use them as edge weights of a 3D
169 grid graph, where each node represents a pixel/voxel of the volume image.

170 4.1 Data: CREMI Challenge

171 We evaluate the algorithms in our framework on the competitive CREMI 2016 EM Segmentation
172 Challenge [20] that is currently the neuron segmentation challenge with the largest amount of training
173 data available. The dataset comes from serial section EM of *Drosophila* fruit-fly tissue and consists
174 of 6 volumes of 1250x1250x125 voxels at resolution 4x4x40nm, three of which present publicly
175 available training ground truth. The results submitted to the leaderboard are evaluated using the
176 CREMI score², based on the Adapted Rand-Score (Rand-Score) and the Variation of Information
177 Score [3]. In Appendix ??, we provide more details about the training of our CNN model, inspired
178 by work of [42, 21].

179 **Additional methods tested** We compare the performances of GASP with other basic and state-
180 of-the-art post-processing methods. To ensure a fair comparison, we test all methods on the same
181 predictions of our CNN model. As basic method, we perform a simple thresholding (THRESH) by
182 running connected components on a boundary map generated from the CNN affinities (see Appendix
183 ?? for more details on this and the following methods). On the other hand, most of state-of-the-art
184 methods for neuron-segmentation first generates 2D superpixels and then apply a graph partitioning
185 algorithm, since this approach so far proved to be the most reliable that could scale up to the size of
186 the problem. Superpixels are usually computed with a watershed algorithm seeded at the maxima of
187 a boundary distance transform (WSDT). The algorithms employed in our comparison to partition
188 the superpixel graph were given by approximations of the multicut (WSDT+MC) and lifted multicut
189 (WSDT+LMC) problems.

190 4.2 Results and discussion

191 **Comparison results** Table 2 shows how GASP compares to other tested post-processing methods
192 applied to the predictions of our model. GASP with *Average* linkage, representing one of the
193 new algorithms introduced by our generalized framework, significantly outperformed all other
194 previously proposed agglomerative methods like GAEC, Greedy Fixation (using *Sum* linkage) or
195 Mutex Watershed (using *Abs Max* linkage). The competitive performances of this simple parameter-
196 free agglomerative algorithm are also reported by Table 3 that represents the current leader-board
197 of the challenge: all entries, apart from GASP, employ superpixel-based post-processing pipelines,
198 several of which rely on the complex multicut problem. In the comparison shown in Table 2,
199 GASP with *Average* linkage even achieved superior scores compared to other methods based on
200 WSDT superpixels. This shows that this method offers a simple and successful approach growing
clusters starting directly from pixels and had the advantage of avoiding potential errors in the

²<https://cremi.org/leaderboard/>

	GASP Linkage	Cremi-Score (lower is better)	Cremi-Score (lower is better)
Our UNet + GASP Avg-Link	Average	0.226	0.221
Our UNet + Greedy Fixation [43]	Sum + Constr.	0.282	0.228
Our UNet + DTWS + MC	-	0.310	0.241
Our UNet + DTWS + LMC	-	0.317	0.276
Our UNet + Mutex Watershed [69]	Abs. Max.	0.322	0.566
Our UNet + GAEC [30]	Sum	0.334	0.616
Our UNet + THRESH	-	1.521	

Table 2: Cremi-Scores achieved by different post-processing methods. Scores are averaged over the three CREMI training datasets

Table 3: Current published entries of the CREMI challenge leaderboard [20] on the three test datasets

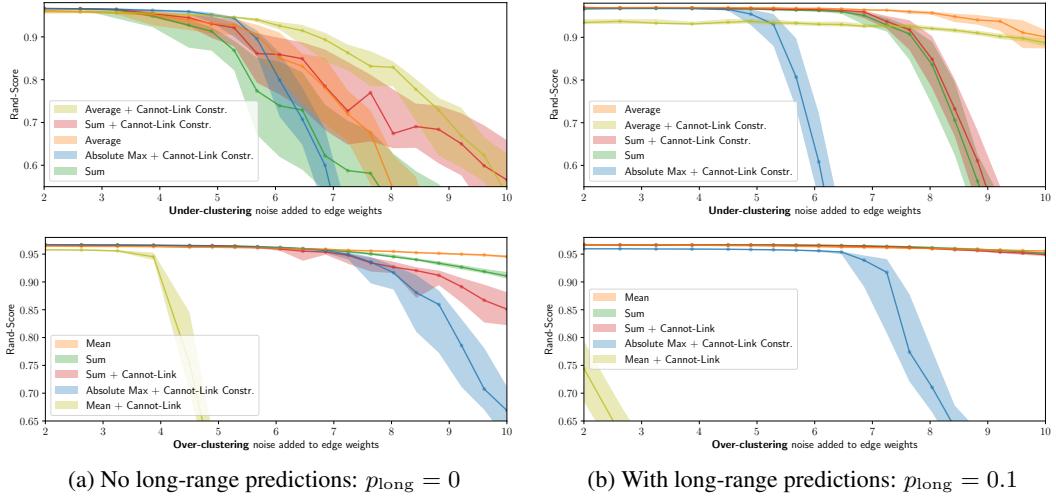


Figure 4: GASP sensitivity to noise: average linkage proved to be the most robust. Performances are given by Rand-Score (higher is better) depending on the amount of noise added to the CNN predictions. Solid lines represent median values over 30 experiments. Values between the 25th and the 75th percentile are shown in shade areas. The two set of experiments using under- and over-clustering noise are summarized in the plots at the top and at the bottom, respectively (see Appendix ?? 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 in the pixel grid-graph.

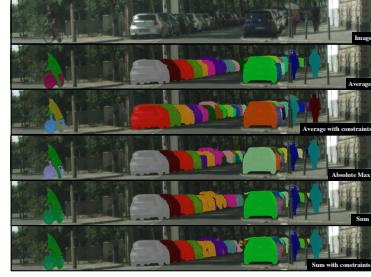
superpixel generation. In fact, generating for example good quality WSDT superpixels requires to tune several parameters. Moreover, superpixel generation does not usually make use of the long-range predictions of the CNN and, in our experiments, using them to compute the weights of the edges in the superpixel graph did not change the final scores. Fig. 3 highlights some failure cases of the different agglomerative algorithms included in our framework. In Appendix ??, we provide more details about how we scaled up GASP to the full datasets. Appendix Table ?? lists the performances and the run-times for all tested GASP linkage.

Noise experiments Additionally, we run a set of experiments where the CNN predictions are perturbed by noise, in order to highlight the properties of each GASP algorithm and perform an in-depth comparison that is as quantitative as possible. Appendix ?? introduces the type of spatially correlated noise that allowed us to perturb the CNN outputs by introducing simulated additional artifacts like missing or false boundary evidences. Fig. 4 summarize our 12000 noise experiments: we focus on the best performing linkage criteria, i.e. *Average*, *Sum* and *Abs Max*, and test them with different values of noise. In these experiments, we also want to assess how beneficial it is to use long-range CNN predictions in the agglomeration. Thus, we perform a set of simulations without adding long-range connections to the grid-graph and another set where we introduce them with a 10% probability³.

³We also performed simulations adding all the long-range predictions given by the CNN model, but we did not note major differences with the case using only 10% of them. Adding this small portion is usually sufficient to improve the scores.

Pipeline	Agglomeration method	Use constraints:	
		No	YES
DWT [4]	-	21.2	-
SGN [47]	-	29.2	-
Mask RCNN [25]	-	31.5	-
	GASP Average	34.3	33.9
GMIS [51]	MultiStepHAC [51]	33.0	-
	GASP Abs. Max. [69]	32.1	32.1
	GASP Sum [30, 43]	31.3	31.9

Table 4: Average Precision (AP) scores (higher is better) on the cityscapes validation set achieved by the tested pipelines.



219 **Average and Abs Max linkage** Our findings confirm that GASP with *Average* linkage criterion
 220 represents the most robust algorithm tested and the one that benefits the most from using the long-
 221 range CNN predictions. On the other hand, it is not a surprise that the *Abs Max* statistic proposed by
 222 [69] is less robust to noise than the *Average* linkage, but, as we show in the Appendix Table ??, *Abs*
 223 *Max* represents a valid and considerably faster option. Adding long-range connections to the graph
 224 is generally helpful, but when many of them carry repulsive weights, then GASP with cannot-link
 225 constraints shows a clear tendency to over-cluster.

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

238 5 Experiments on CityScapes

239 We also evaluate the performances of GASP on the CityScapes dataset [12], which consists of 5000
 240 street-scene images: 2975 for training, 500 for validation and 1525 for testing. See Appendix ??
 241 for more details on how we fine-tuned the state-of-the-art proposal-free pipeline proposed in GMIS
 242 [51] by using a *Søresen-Dice* loss, similarly to [69]. Results are summarized in Table 4 and confirm
 243 our findings on neuron-segmentation: GASP with average linkage achieves the best scores, whereas
 244 other linkage tend to over-cluster, like *Abs. Max.*, or under-cluster and merge instances, like *Sum*.
 245 In Appendix, Table ?? includes the scores of all other tested GASP algorithms. The graph-merging
 246 algorithm proposed by [51] (MultiStepHAC) requires the user to tune several threshold parameters
 247 and it was probably tailored to the original affinities predicted by them, so it did not generalize well to
 248 our fine-tuned model and it achieved lower scores compared to the original AP value of 34.1 reported
 249 in [51].

250 6 Conclusion

251 We have presented a novel theoretical framework for agglomerative clustering of graphs with both
 252 positive and negative weights and we have shown that several existing clustering algorithms, e.g.
 253 the Mutex Watershed [69], can be reformulated as special cases of one underlying agglomerative
 254 algorithm. This framework also allowed us to introduce new algorithms, one of which, based on
 255 an average linkage, outperformed all the others and proved to be a simple and remarkably robust
 256 approach to process short- and long-range predictions of a CNN applied to an instance segmentation
 257 task. On biological images, this simple average agglomeration algorithm can represent a valuable
 258 choice for an user that is not willing to spend much time tuning complex task-dependent pipelines
 259 based on superpixels. In future work we plan to extend the comparison to other types of graphs and
 260 explore common theoretical properties of the algorithms included in the framework.

261 **References**

- 262 [1] B. Andres, T. Kroeger, K. L. Briggman, W. Denk, N. Korogod, G. Knott, U. Koethe, and F. A. Hamprecht.
263 Globally optimal closed-surface segmentation for connectomics. In *European Conference on Computer
264 Vision*, pages 778–791. Springer, 2012.
- 265 [2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation.
266 *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- 267 [3] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber,
268 D. Laptev, S. Dwivedi, J. M. Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms
269 for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.
- 270 [4] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE
271 Conference on Computer Vision and Pattern Recognition*, pages 5221–5229, 2017.
- 272 [5] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.
- 273 [6] T. Beier, B. Andres, U. Köthe, and F. A. Hamprecht. An efficient fusion move algorithm for the minimum
274 cost lifted multicut problem. In *European Conference on Computer Vision*, pages 715–730. Springer, 2016.
- 275 [7] T. Beier, C. Pape, N. Rahaman, T. Prange, S. Berg, D. D. Bock, A. Cardona, G. W. Knott, S. M. Plaza,
276 L. K. Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature
277 Methods*, 14(2):101, 2017.
- 278 [8] Y.-T. Chen, X. Liu, and M.-H. Yang. Multi-instance object segmentation with occlusion handling. In
279 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3470–3478,
280 2015.
- 281 [9] S. Chopra and M. R. Rao. On the multiway cut polyhedron. *Networks*, 21(1):51–89, 1991.
- 282 [10] S. Chopra and M. R. Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, 1993.
- 283 [11] D. Ciresan, A. Giusti, L. M. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal
284 membranes in electron microscopy images. In *Advances in neural information processing systems*, pages
285 2843–2851, 2012.
- 286 [12] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and
287 B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE
288 conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- 289 [13] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multi-task network cascades. In
290 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158,
291 2016.
- 292 [14] B. De Brabandere, D. Neven, and L. Van Gool. Semantic instance segmentation with a discriminative loss
293 function. *arXiv preprint arXiv:1708.02551*, 2017.
- 294 [15] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs.
295 *Theoretical Computer Science*, 361(2-3):172–187, 2006.
- 296 [16] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes
297 (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- 298 [17] A. Fathi, Z. Wojna, V. Rathod, P. Wang, H. O. Song, S. Guadarrama, and K. P. Murphy. Semantic instance
299 segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017.
- 300 [18] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International journal
301 of computer vision*, 59(2):167–181, 2004.
- 302 [19] J. Funke, F. A. Hamprecht, and C. Zhang. Learning to segment: training hierarchical segmentation under
303 a topological loss. In *International Conference on Medical Image Computing and Computer-Assisted
304 Intervention*, pages 268–275. Springer, 2015.
- 305 [20] J. Funke, S. Saalfeld, D. Bock, S. Turaga, and E. Perlman. Cremi challenge. <https://cremi.org.>, 2016.
306 Accessed: 2019-05-15.
- 307 [21] J. Funke, F. D. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga. Large
308 scale image segmentation with structured loss based deep learning for connectome reconstruction. *IEEE
309 transactions on pattern analysis and machine intelligence*, 2018.

- 310 [22] M. Grötschel and Y. Wakabayashi. A cutting plane algorithm for a clustering problem. *Mathematical*
 311 *Programming*, 45(1-3):59–96, 1989.
- 312 [23] M. Grötschel and Y. Wakabayashi. Facets of the clique partitioning polytope. *Mathematical Programming*,
 313 47(1-3):367–387, 1990.
- 314 [24] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In
 315 *European Conference on Computer Vision*, pages 297–312. Springer, 2014.
- 316 [25] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international*
 317 *conference on computer vision*, pages 2961–2969, 2017.
- 318 [26] M. Januszewski, J. Kornfeld, P. H. Li, A. Pope, T. Blakely, L. Lindsey, J. Maitin-Shepard, M. Tyka,
 319 W. Denk, and V. Jain. High-precision automated reconstruction of neurons with flood-filling networks.
 320 *Nature methods*, 15(8):605, 2018.
- 321 [27] J. H. Kappes, M. Speth, B. Andres, G. Reinelt, and C. Schn. Globally optimal image partitioning by
 322 multicuts. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern*
 323 *Recognition*, pages 31–44. Springer, 2011.
- 324 [28] A. Kardoost and M. Keuper. Solving minimum cost lifted multicut problems by node agglomeration. In
 325 *ACCV 2018, 14th Asian Conference on Computer Vision*, Perth, Australia, 2018.
- 326 [29] V. Kaynig, A. Vazquez-Reina, S. Knowles-Barley, M. Roberts, T. R. Jones, N. Kasthuri, E. Miller,
 327 J. Lichtman, and H. Pfister. Large-scale automatic reconstruction of neuronal processes from electron
 328 microscopy images. *Medical image analysis*, 22(1):77–88, 2015.
- 329 [30] M. Keuper, E. Levinkov, N. Bonneel, G. Lavoué, T. Brox, and B. Andres. Efficient decomposition of image
 330 and mesh graphs by lifted multicut. In *Proceedings of the IEEE International Conference on Computer*
 331 *Vision*, pages 1751–1759, 2015.
- 332 [31] B. R. Kiran and J. Serra. Global-local optimizations by hierarchical cuts and climbing energies. *Pattern*
 333 *Recognition*, 47(1):12–24, 2014.
- 334 [32] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. Instancecut: from edges to instances
 335 with multicut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
 336 5008–5017, 2017.
- 337 [33] S. Knowles-Barley, V. Kaynig, T. R. Jones, A. Wilson, J. Morgan, D. Lee, D. Berger, N. Kasthuri,
 338 J. W. Lichtman, and H. Pfister. Rhoanet pipeline: Dense automatic neural annotation. *arXiv preprint*
 339 *arXiv:1611.06973*, 2016.
- 340 [34] I. Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint*
 341 *arXiv:1511.07386*, 2015.
- 342 [35] S. Kong and C. C. Fowlkes. Recurrent pixel embedding for instance grouping. In *Proceedings of the IEEE*
 343 *Conference on Computer Vision and Pattern Recognition*, pages 9018–9028, 2018.
- 344 [36] N. Krasowski, T. Beier, G. W. Knott, U. Koethe, F. A. Hamprecht, and A. Kreshuk. Improving 3d em
 345 data segmentation by joint optimization over boundary evidence and biological priors. In *2015 IEEE 12th*
 346 *International Symposium on Biomedical Imaging (ISBI)*, pages 536–539. IEEE, 2015.
- 347 [37] V. Kulikov, V. Yurchenko, and V. Lempitsky. Instance segmentation by deep coloring. *arXiv preprint*
 348 *arXiv:1807.10007*, 2018.
- 349 [38] L. Ladický, P. Sturges, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining
 350 object detectors and crfs. In *European conference on computer vision*, pages 424–437. Springer, 2010.
- 351 [39] G. N. Lance and W. T. Williams. A general theory of classificatory sorting strategies: 1. hierarchical
 352 systems. *The computer journal*, 9(4):373–380, 1967.
- 353 [40] J.-H. Lange, B. Andres, and P. Swoboda. Combinatorial persistency criteria for multicut and max-cut.
 354 *arXiv preprint arXiv:1812.01426*, 2018.
- 355 [41] J.-H. Lange, A. Karrenbauer, and B. Andres. Partial optimality and fast lower bounds for weighted
 356 correlation clustering. In *International Conference on Machine Learning*, pages 2898–2907, 2018.
- 357 [42] K. Lee, J. Zung, P. Li, V. Jain, and H. S. Seung. Superhuman accuracy on the snemi3d connectomics
 358 challenge. *arXiv preprint arXiv:1706.00120*, 2017.

- 359 [43] E. Levinkov, A. Kirillov, and B. Andres. A comparative study of local search algorithms for correlation
 360 clustering. In *German Conference on Pattern Recognition*, pages 103–114. Springer, 2017.
- 361 [44] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In
 362 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367,
 363 2017.
- 364 [45] X. Liang, Y. Wei, X. Shen, Z. Jie, J. Feng, L. Lin, and S. Yan. Reversible recursive instance-level object
 365 segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages
 366 633–641, 2016.
- 367 [46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft
 368 coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer,
 369 2014.
- 370 [47] S. Liu, J. Jia, S. Fidler, and R. Urtasun. Sgn: Sequential grouping networks for instance segmentation. In
 371 *Proceedings of the IEEE International Conference on Computer Vision*, pages 3496–3504, 2017.
- 372 [48] T. Liu, C. Jones, M. Seyedhosseini, and T. Tasdizen. A modular hierarchical approach to 3d electron
 373 microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, 2014.
- 374 [49] T. Liu, M. Seyedhosseini, and T. Tasdizen. Image segmentation using hierarchical merge tree. *IEEE
 375 transactions on image processing*, 25(10):4596–4607, 2016.
- 376 [50] T. Liu, M. Zhang, M. Javanmardi, N. Ramesh, and T. Tasdizen. Sshmt: Semi-supervised hierarchical
 377 merge tree for electron microscopy image segmentation. In *European Conference on Computer Vision*,
 378 pages 144–159. Springer, 2016.
- 379 [51] Y. Liu, S. Yang, B. Li, W. Zhou, J. Xu, H. Li, and Y. Lu. Affinity derivation and graph merge for instance
 380 segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 686–703,
 381 2018.
- 382 [52] F. Malmberg, R. Strand, and I. Nyström. Generalized hard constraints for graph segmentation. In
 383 *Scandinavian Conference on Image Analysis*, pages 36–47. Springer, 2011.
- 384 [53] Y. Meirovitch, A. Matveev, H. Saribekyan, D. Budden, D. Rolnick, G. Odor, S. Knowles-Barley, T. R.
 385 Jones, H. Pfister, J. W. Lichtman, et al. A multi-pass approach to large-scale connectomics. *arXiv preprint
 386 arXiv:1612.02120*, 2016.
- 387 [54] A. Newell, Z. Huang, and J. Deng. Associative embedding: End-to-end learning for joint detection and
 388 grouping. In *Advances in Neural Information Processing Systems*, pages 2277–2287, 2017.
- 389 [55] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii. Machine learning of hierarchical
 390 clustering to segment 2d and 3d images. *PloS one*, 8(8):e71715, 2013.
- 391 [56] C. Pape, T. Beier, P. Li, V. Jain, D. D. Bock, and A. Kreshuk. Solving large multicut problems for
 392 connectomics via domain decomposition. In *Proceedings of the IEEE International Conference on
 393 Computer Vision*, pages 1–10, 2017.
- 394 [57] T. Parag, F. Tschopp, W. Grisaitis, S. C. Turaga, X. Zhang, B. Matejek, L. Kamentsky, J. W. Lichtman,
 395 and H. Pfister. Anisotropic em segmentation by 3d affinity learning and agglomeration. *arXiv preprint
 396 arXiv:1707.08935*, 2017.
- 397 [58] M. Ren and R. S. Zemel. End-to-end instance segmentation with recurrent attention. In *Proceedings of the
 398 IEEE Conference on Computer Vision and Pattern Recognition*, pages 6656–6664, 2017.
- 399 [59] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region
 400 proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.
- 401 [60] Z. Ren and G. Shakhnarovich. Image segmentation by cascaded region agglomeration. In *Proceedings of
 402 the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2013.
- 403 [61] B. Romera-Paredes and P. H. S. Torr. Recurrent instance segmentation. In *European conference on
 404 computer vision*, pages 312–329. Springer, 2016.
- 405 [62] P. Salembier and L. Garrido. Binary partition tree as an efficient representation for image processing,
 406 segmentation, and information retrieval. *IEEE transactions on Image Processing*, 9(4):561–576, 2000.
- 407 [63] P. Schlegel, M. Costa, and G. S. Jefferis. Learning from connectomics on the fly. *Current opinion in insect
 408 science*, 24:96–105, 2017.

- 409 [64] U. Schmidt, M. Weigert, C. Broaddus, and G. Myers. Cell detection with star-convex polygons. In
410 *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 265–
411 273. Springer, 2018.
- 412 [65] S. C. Turaga, K. L. Briggman, M. Helmstaedter, W. Denk, and H. S. Seung. Maximin affinity learning of
413 image segmentation. pages 1865–1873, 2009.
- 414 [66] J. Uhrig, M. Cordts, U. Franke, and T. Brox. Pixel-level encoding and depth layering for instance-level
415 semantic labeling. In *German Conference on Pattern Recognition*, pages 14–25. Springer, 2016.
- 416 [67] M. G. Uzunbas, C. Chen, and D. Metaxas. An efficient conditional random field approach for automatic
417 and interactive neuron segmentation. *Medical image analysis*, 27:31–44, 2016.
- 418 [68] S. Wolf, A. Bailoni, C. Pape, N. Rahaman, A. Kreshuk, U. Köthe, and F. A. Hamprecht. The mutex
419 watershed and its objective: Efficient, parameter-free image partitioning. *arXiv preprint arXiv:1904.12654*,
420 2019.
- 421 [69] S. Wolf, C. Pape, A. Bailoni, N. Rahaman, A. Kreshuk, U. Kothe, and F. Hamprecht. The mutex watershed:
422 Efficient, parameter-free image partitioning. In *Proceedings of the European Conference on Computer
423 Vision (ECCV)*, pages 546–562, 2018.
- 424 [70] S. Xie and Z. Tu. Holistically-nested edge detection. In *Proc. ICCV'15*, pages 1395–1403, 2015.
- 425 [71] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes. Layered object models for image segmentation.
426 *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1731–1743, 2012.
- 427 [72] J. Yarkony, A. Ihler, and C. C. Fowlkes. Fast planar correlation clustering for image segmentation. In
428 *European Conference on Computer Vision*, pages 568–581. Springer, 2012.
- 429 [73] T. Zeng, B. Wu, and S. Ji. Deepem3d: approaching human-level performance on 3d anisotropic em image
430 segmentation. *Bioinformatics*, 33(16):2555–2562, 2017.