

000
001
002
003
004
005
006
007
008
009
010
011

GASP, a generalized framework for agglomerative clustering of signed graphs and its application to Instance Segmentation

Anonymous CVPR submission

Paper ID 5813

Abstract

We propose a new 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 before. On stochastic block model problems, GASP compares favorably to spectral clustering. More importantly, we conduct a systematic comparison of various instantiations of GASP in image instance segmentation problems, in terms of accuracy but also efficiency and robustness to noise. We find that 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 tasks as diverse 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 require users to specify the desired numbers of clusters (as in spectral clustering) or a termination criterion (e.g. in 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 obtain a clustering without defining additional parameters. A canonical formulation of the signed graph partitioning problem is the *multicut* or *correlation clustering* problem [36, 12]. This problem is NP-hard, though many approximate solvers have been proposed [50, 70, 7, 94]. The general problem of graph partitioning can also be solved approximately by greedy agglomerative clustering [40, 54, 90, 37]. Agglomerative clustering algorithms for signed graphs have clear advantages: they are parameter-free and efficient. Despite the fact that a variety of these algorithms exist, no overarching study has so far been conducted 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 new theoretical framework that generalizes over agglomerative algorithms for signed graphs by linking them to hierarchical agglomerative clustering on positive-weighted graphs [49]. This framework defines an underlying basic algorithm and allows us to explore its combinations with different linkage criteria and *cannot-link constraints*. We then formally prove that some of the combinations correspond to existing clustering algorithms and introduce new algorithms for combinations which have not been explored before.

We evaluate and compare these algorithms on *instance segmentation* – a computer vision task of assigning each pixel of an image to an object instance. We use a CNN to predict the edge weights of a graph such that each node represents a pixel of the image, similarly to [63, 53, 90], and provide these weights as input to the algorithms in our framework (see Fig. 1).

With our comparison experiments, performed both on 2D urban scenes from the CityScapes dataset and 3D electron microscopy image volumes of neurons, we benchmark all algorithms in our framework, focusing on their efficiency, robustness and tendency to over- or under-cluster. We show that one of the new algorithms derived from our framework, based on an average linkage criterion, outperforms all previously known agglomeration methods expressed in the framework and that it achieves competitive

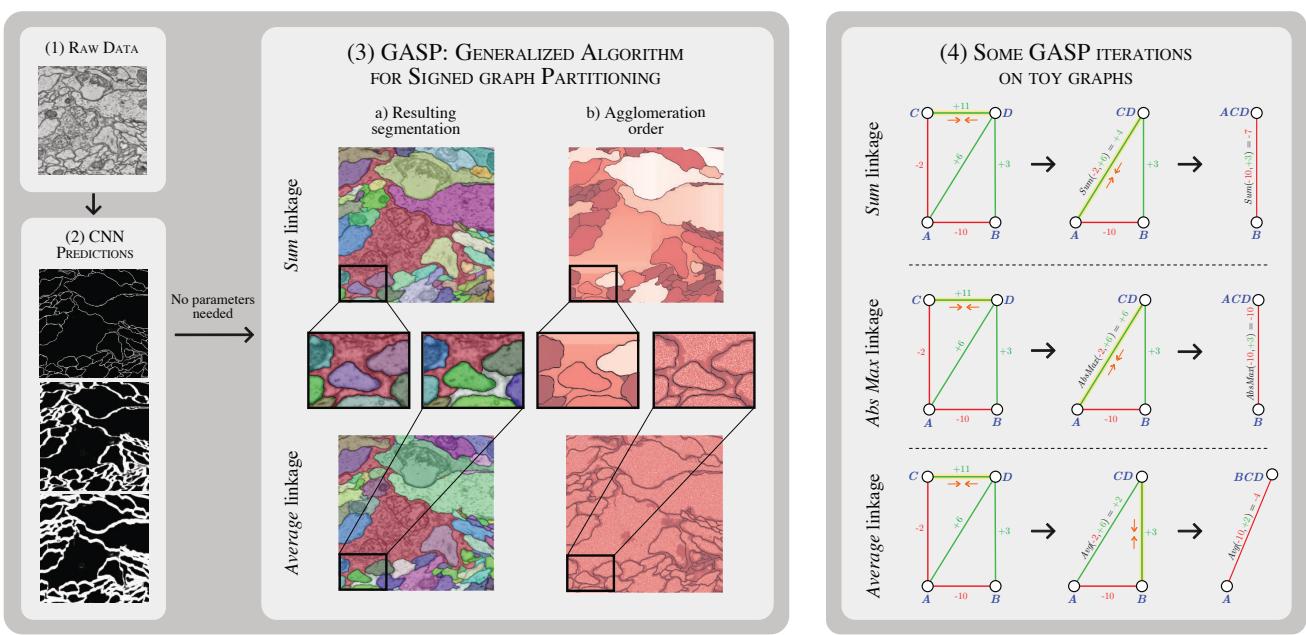


Figure 1: GASP example. (1) Raw data from the CREMI 2016 neuron-segmentation challenge. (2) Some short- and long-range predictions of our CNN model, where white pixels represent boundary evidence. (3) Outputs of two agglomerative algorithms included in our proposed generalized clustering framework, with *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/red), or never (black). (4) Some iterations of GASP 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.

performance on CityScapes and the challenging CREMI 2016 segmentation benchmark.

In Sec. 6, we also show how GASP outperforms spectral clustering methods on the task of neuron segmentation and how on synthetic graphs it achieves similar scores to a recently proposed spectral method for signed graphs.

2. Related work

Proposal-based methods have been highly successful in instance segmentation competitions like MS COCO [57], Pascal VOC2012 [23] and CityScapes [16]. They decompose the instance segmentation task into two steps that consists in generating object proposals and assigning to each bounding box a class and a binary segmentation mask [34, 74, 59, 93, 55, 48, 33, 9, 19, 56]. Other methods use instead recurrent models to sequentially generate instances one-by-one [78, 76].

Proposal-free methods adopt a bottom-up approach by directly grouping pixels into instances. Recently, there has been a growing interest in such methods that do not involve object detection, since, in certain types of data, object instances cannot be approximated by bounding boxes. For example, [42] uses a combinatorial framework for instance

segmentation, whereas a watershed transform is learned in [5] by also predicting its gradient direction. 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 [52, 24, 68, 20]. Final instances are then retrieved by applying a clustering algorithm, like the end-to-end trainable mean-shift pipeline of [45]. Other recent successful methods simply let the model predict the relative coordinates of the instance center [67, 10] or, given a point (x, y) in the image, the model generates the mask of the instance located at (x, y) [84].

Edge detection also experienced recent progress thanks to deep learning, both on natural images [30, 63, 91, 44] and biological data [53, 83, 65, 15]. 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 [38, 46] or trees [65, 62, 60, 27, 87] to represent the region merging hierarchy; the lifted multi-cut [8] formulates the problem in a combinatorial framework, while flood-filling networks [35] and MaskExtend [65] use a CNN to iteratively grow one region/neuron at

216 a time; recently, the work of [66] made the process more
 217 efficient by employing a combinatorial encoding of the seg-
 218 mentation. A structured learning approach was also pro-
 219 posed in [29, 86].

220 **Agglomerative graph clustering** has often been applied
 221 to instance segmentation [3, 77, 61, 81], because of its ef-
 222 ficiency as compared to other top-down approaches. Novel
 223 termination criteria and merging strategies have often been
 224 proposed: the agglomeration in [64] deploys fixed sets of
 225 merge constraints; the popular graph-based method [25]
 226 stops the agglomeration when the merge costs exceed a
 227 measure of quality for the current clusters. The optimization
 228 approach in [41] performs greedy merge decisions that mini-
 229 mize a certain energy, while other pipelines use classical
 230 linkage criteria, e.g. average linkage [63, 53], median [29]
 231 or a linkage learned by a random forest classifier [62, 43].

232 **Clustering of signed graphs** has the goal of partitioning
 233 a graph encoding both attractive and repulsive cues. Find-
 234 ing an optimally balanced partitioning has a long history
 235 in combinatorial optimization [31, 32, 13]. NP-hardness
 236 of the *correlation clustering* problem was shown in [6],
 237 while the connection with graph multcuts was made by
 238 [21]. Modern integer linear programming solvers can
 239 tackle problems of considerable size [2], but accurate ap-
 240 proximations [70, 7, 94], greedy agglomerative algorithms
 241 [54, 89, 40, 37] and persistence criteria [51, 50] have been
 242 proposed for even larger graphs. Another line of research
 243 is given by spectral clustering methods that do however re-
 244 quire the user to specify the number of clusters in advance.
 245 Recently, some of these methods have been generalized to
 246 graphs with signed weights [17, 11, 47], whereas others let
 247 the user specify must-link and cannot-link constraints be-
 248 tween clusters [75, 88, 18].

249 This work reformulates the clustering algorithms of [54,
 250 90, 40] in a generalized framework and adopts ideas from
 251 the proposal-free methods [63, 90, 53] to predict long-range
 252 relationships between pixels.

254 3. Generalized framework for agglomerative 255 clustering of signed graphs

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

262 3.1. Notation and graph formalism

263 We consider an undirected simple edge-weighted graph
 264 $\mathcal{G}(V, E, w^+, w^-)$ with both attractive and repulsive edge
 265 attributes. In computer vision applications, the nodes can
 266 represent either pixels, superpixels or voxels. We call the set Π
 267 a *clustering* or *partitioning* with K clusters if $V = \cup_{S \in \Pi} S$,
 268 $S \cap S' = \emptyset$ for different clusters $S, S' \in \Pi$ and every cluster

269 $S \in \Pi$ induces a connected subgraph of \mathcal{G} . We also denote
 270 as S_u the cluster associated with node u . The weight func-
 271 tion $w^+ : E \rightarrow \mathbb{R}^+$ associates to every edge a positive
 272 scalar attribute $w_e^+ \in \mathbb{R}^+$ representing a merge affinity or
 273 a similarity measure: the higher this number, the higher the
 274 inclination of the two incident vertices to be assigned to the
 275 same cluster¹. On the other hand, $w^- : E \rightarrow \mathbb{R}^+$ asso-
 276 ciates to each edge a split tendency $w_e^- \in \mathbb{R}^+$: the higher
 277 this weight, the more the incident vertices would like to be
 278 in different clusters. Graphs of the type $\mathcal{G}(V, E, w^+, w^-)$
 279 are also often defined as *signed graphs* $\mathcal{G}(V, E, w)$, fea-
 280 turing positive and negative edge weights $w_e \in \mathbb{R}$. Follow-
 281 ing the theoretical considerations in [51], we define these
 282 signed weights as $w_e = w_e^+ - w_e^-$. Some approaches di-
 283 rectly compute w_e , whereas others compute w_e^+ and w_e^-
 284 separately. In this formalism, graphs with purely attrac-
 285 tive interactions are a special case of $\mathcal{G}(V, E, w)$ with $w_e \geq$
 286 0, $\forall e \in E$.

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

296 3.2. GASP: generalized algorithm for signed graph 297 partitioning

298 In Algorithm I we provide simplified pseudo-code
 299 for the proposed GASP algorithm. GASP implements a
 300 bottom-up approach that starts by assigning each node to
 301 its own cluster and then iteratively merges pairs of adja-
 302 cent clusters. The algorithm has two variants. The first
 303 one (option *addCannotLinkConstraints* is False) starts by
 304 merging clusters with the strongest attractive interaction and
 305 stops once the remaining clusters share only mutual repul-
 306 sive interactions (see iterations on toy graphs in block 4 of
 307 Fig. I). After each merging iteration, the interaction be-
 308 tween the merged cluster and its neighbors is updated ac-
 309 cording to one of the linkage criteria $\mathcal{W}(S_u, S_v)$ listed in
 310 Table I

311 In the second variant (option *addCannotLinkConstraints*
 312 is True), Algorithm I also introduces cannot-link con-
 313 straints, which represent mutual exclusion relationships be-
 314 tween pairs of nodes that cannot be associated with the same
 315 cluster in the final clustering. This variant selects the pair of
 316 clusters with the highest absolute interaction $|\mathcal{W}(S_u, S_v)|$,
 317 so that the most attractive and the most repulsive pairs are

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

324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356

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: Initial clustering:  $\Pi = \{\{v_1\}, \dots, \{v_{|V|}\}\}$ 
2: Initialize interactions between clusters with  $w_e = w_e^+ - w_e^-$ 
3: repeat
4:   Get  $S_u, S_v \in \Pi$  with highest interaction  $|\mathcal{W}(S_u, S_v)|$ 
5:   if  $[\mathcal{W}(S_u, S_v) > 0]$  and  $[S_u, S_v \text{ not constrained}]$  then
6:     Merge cluster  $S_u$  with  $S_v$ 
7:     Update interactions & constraints with neighboring clusters
8:   else if addCannotLinkConstr and  $[\mathcal{W}(S_u, S_v) \leq 0]$  then
9:     Add CannotLink Constraint between  $S_u$  and  $S_v$ 
10:  until [all interactions between clusters are repulsive] or
11:    [all adjacent clusters have cannot-link constraints]
12:  return  $\Pi$ 

```

analyzed first (see example in Fig. 2(b)). If the interaction is repulsive, then the two clusters are constrained and its members can never merge in subsequent steps. If the interaction is attractive, then the clusters are merged, provided that they were not previously constrained. The algorithm terminates when all the remaining clusters are constrained.

In Appendix 8.1 we comment on the algorithm's computational complexity $\mathcal{O}(N^2 \log N)$ and present our implementation given by the edge contraction Algorithm 2 based on a *disjoint set data structure* and a *priority queue*.

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

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

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

Given a graph with both attractive and repulsive cues, an edge contraction algorithm with a sum update rule was pioneered in [54, 40] (Table I, *Sum* linkage). The authors present both a version with cannot-link constraints and one without, and then compare them with other greedy local-search algorithms approximating the multicut optimization problem. The Mutex Watershed [90] is another signed graph partitioning algorithm that introduces dynamical cannot-link constraints. In Proposition 8.1 (see Ap-

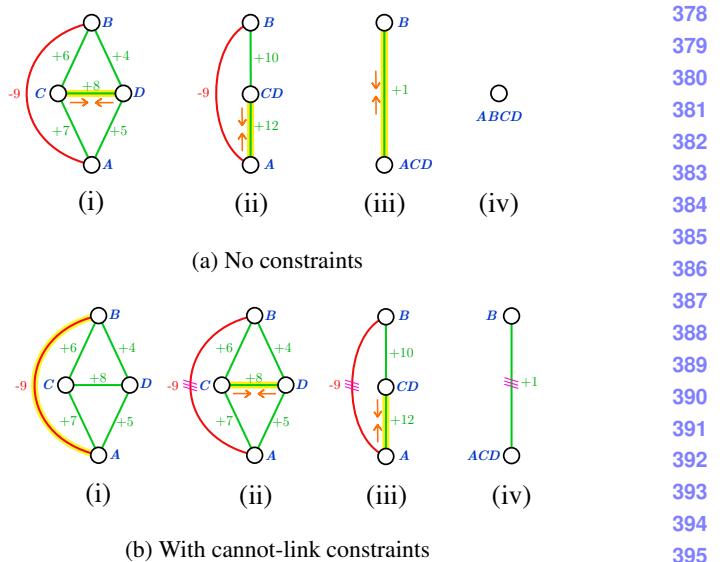


Figure 2: Some iterations of the generalized algorithm (using *Sum* linkage criteria) with and without adding cannot-link constraints. The graph has both attractive (green) and repulsive (red) edges and cannot-link constraints are shown with triple violet bars on the edges. The edge selected at each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

pendix 8.2) we prove that, surprisingly, it can also be seen as an efficient implementation of GASP with *Absolute maximum* linkage (def. in Table I). Moreover, in Proposition 8.2 we also prove that GASP with *Abs Max* linkage returns the same clustering with or without enforcing cannot-link constraints. On the other hand, to our knowledge, *Average*, *Max* or *Min* linkage criteria have never been used for signed graph agglomerative algorithms or been combined with cannot-link constraints.

Apart from the linkage criteria defined in Table I, additional ones were proposed in the literature: [62] for example uses a learned approach where a random forest classifier updates the cluster interactions depending on predefined edge and node features; other approaches introduce a weight regularization depending on the size of the clusters [25, 37], whereas [29] uses a *quantile* linkage criterion by populating a histogram for each inter-cluster interaction. In our experiments, we decided to focus on the linkage criteria listed in Table I, since they represent the most common options.

4. Experiments on neuron segmentation

We first evaluate and compare the agglomerative clustering algorithms described in the generalized framework on the task of neuron segmentation in electron microscopy (EM) image volumes. This application is of key interest

	GASP linkage criteria $\mathcal{W}(S_u, S_v)$	Sum $\sum_{e \in E_{uv}} w_e$	Absolute Maximum w_e with $e = \arg \max_{t \in E_{uv}} w_t $	Average $\sum_{e \in E_{uv}} w_e / E_{uv} $	Maximum $\max_{e \in E_{uv}} w_e$	Minimum $\min_{e \in E_{uv}} w_e$	486
432							487
433							488
434							489
435							490
436	Unsigned Graphs	Sum Linkage Hier. Aggl. Clust.	Single Linkage Hier. Aggl. Clust.	Average Linkage Hier. Aggl. Clust.	Single Linkage Hier. Aggl. Clust.	Complete Linkage Hier. Aggl. Clust.	491
437							492
438	Signed Graphs Without Constraints	GAEC [40]	Mutex Watershed [90]	NEW	NEW	NEW	493
439							494
440	Signed Graphs With Constraints	Greedy Fixation [54]	Mutex Watershed [90]	NEW	NEW	NEW	495
441							496
442							497
443							498
444							499
445							500
446							501
447							502
448		Sum	Absolute Maximum	Average	Maximum	Minimum	503
449	No constraints						504
450							505
451							506
452							507
453							508
454							509
455							510
456							511
457	With constraints		Equivalent to the case without constraints				512
458			Equivalent to the case without constraints				513
459			Equivalent to the case without constraints				514
460			Equivalent to the case without constraints				515
461			Equivalent to the case without constraints				516
462			Equivalent to the case without constraints				517
463			Equivalent to the case without constraints				518
464			Equivalent to the case without constraints				519
465			Equivalent to the case without constraints				520
466			Equivalent to the case without constraints				521
467			Equivalent to the case without constraints				522
468			Equivalent to the case without constraints				523
469			Equivalent to the case without constraints				524
470			Equivalent to the case without constraints				525
471			Equivalent to the case without constraints				526
472			Equivalent to the case without constraints				527
473			Equivalent to the case without constraints				528
474			Equivalent to the case without constraints				529
475			Equivalent to the case without constraints				530
476			Equivalent to the case without constraints				531
477			Equivalent to the case without constraints				532
478			Equivalent to the case without constraints				533
479			Equivalent to the case without constraints				534
480			Equivalent to the case without constraints				535
481			Equivalent to the case without constraints				536
482			Equivalent to the case without constraints				537
483			Equivalent to the case without constraints				538
484			Equivalent to the case without constraints				539
485			Equivalent to the case without constraints				

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 criterion, 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. $E_{uv} = (S_u \times S_{v \neq u}) \cap E$.

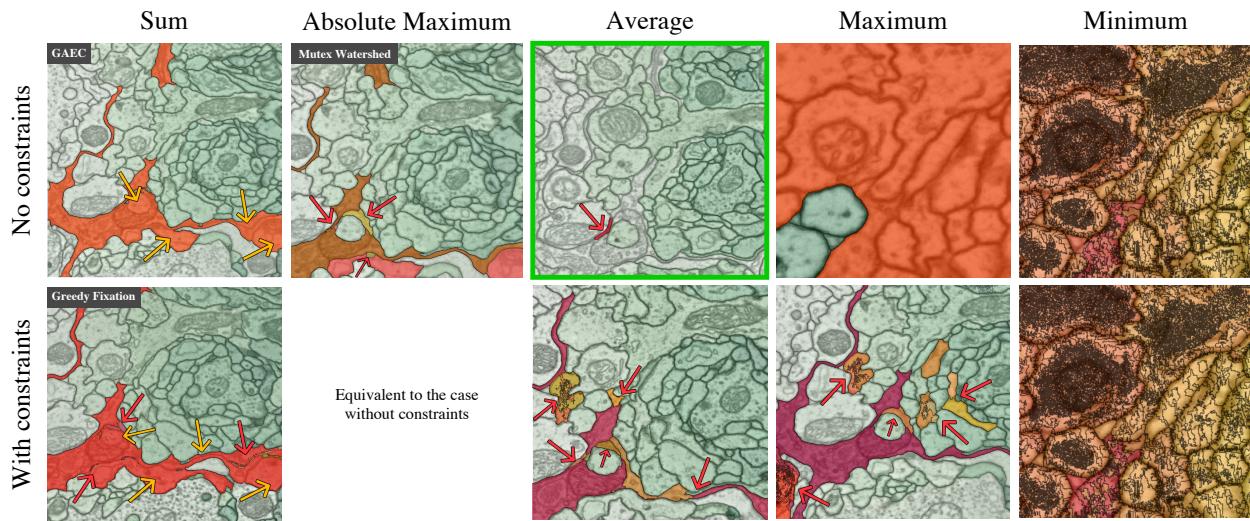


Figure 3: Failure cases of GASP with different linkage criteria highlighted on some difficult parts of the CREMI Challenge data. Only the *wrongly* segmented regions are highlighted in different warm colors. Note that the data is 3D, hence the same color could be assigned to parts of segments that appear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

in connectomics, a field of neuro-science with the goal of reconstructing neural wiring diagrams spanning complete central nervous systems. Currently, only proof-reading or manual tracing yields sufficient accuracy for correct circuit reconstruction [82], thus further progress is required in automated reconstruction methods.

EM segmentation is commonly performed by first predicting boundary pixels [8, 15] or undirected affinities [90, 53, 29], which represent how likely it is for a pair of pixels to belong to the same neuron segment. The affinities do not have to be limited to immediately adjacent pixels. Thus, similarly to [53], we train a CNN to predict both short- and long-range affinities and use them as edge weights of a 3D grid graph, where each node represents a pixel/voxel of the volume image.

4.1. Data: CREMI challenge

We evaluate all algorithms in the proposed framework on the competitive CREMI 2016 EM Segmentation Challenge [28] that is currently the neuron segmentation challenge with the largest amount of training data available. The dataset comes from serial section EM of *Drosophila* fruit-fly tissue and consists of 6 volumes of 1250x1250x125 voxels at resolution 4x4x40nm, three of which come with publicly available training ground truth. The results submitted to the leaderboard are evaluated using the CREMI score, based on the Adapted Rand-Score (Rand-Score) and the Variation of Information Score [4]. In Appendix 8.4, we provide more details about the training of our CNN model, inspired by work of [53, 29].

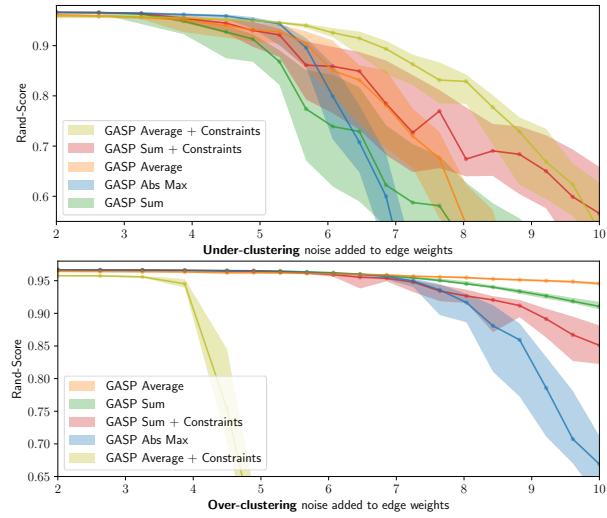
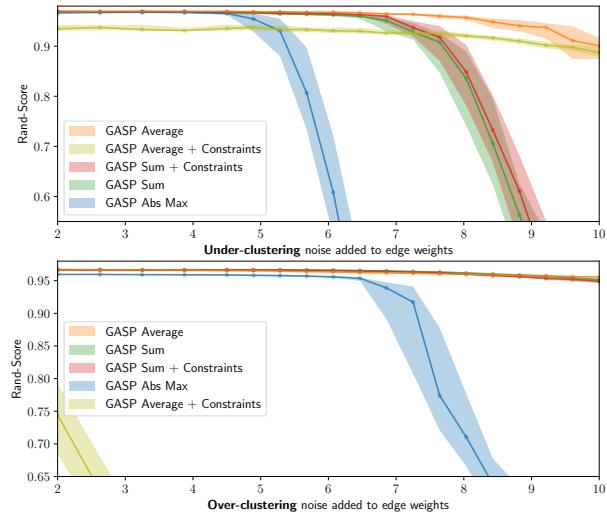
(a) No long-range predictions: $p_{\text{long}} = 0$ (b) With long-range predictions: $p_{\text{long}} = 0.1$

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 8.6 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. Experiments are performed on a crop of CREMI training sample B.

Method	CREMI-Score (lower is better)
GASP Average	0.226
GASP Sum + Constraints [54]	0.282
GASP Abs. Max. [90]	0.322
GASP Max. + Constraints	0.324
GASP Sum [40]	0.334
GASP Average + Constraints	0.563
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 averaged over the three CREMI training datasets.

Method	CREMI-Score (lower is better)
Our CNN + DTWS + LMC	0.221
PNI CNN [53]	0.228
Our CNN + GASP Average	0.241
MALA CNN + MC [29]	0.276
CRU-Net [95]	0.566
LFC [71]	0.616

Table 3: Current leading entries in the CREMI challenge leaderboard [28] (November 2019). All entries, apart from our using GASP, employ superpixel-based post-processing pipelines.

Method	AP	AP 50% (higher is better)
Panoptic-DeepLab [10]	34.6	57.3
UPSNNet [92] †	33.0	59.6
SSAP [30]	32.7	51.8
AdaptIS [84]	32.5	52.5
PANet [59] †	31.8	57.1
GMIS Model [63] + GASP Average	28.3	47.0
JOSECB [67]	27.7	50.9
GMIS [63]	27.3	45.6
Mask R-CNN [34] †	26.2	49.9
SGN [58]	25.0	44.9

Table 4: Results on CityScapes test. Methods marked with † are *proposal-based*. Only methods that do not use external training data (such as MS COCO) are shown.

4.2. Results and discussion

Comparison of linkage criteria Table 2 shows how the agglomerative algorithms derived from our framework compare to each other. For a simple baseline, we also include a segmentation produced by thresholding the affinity predictions (THRESH). GASP with *Average* linkage, representing one of the new algorithms derived from our generalized framework, significantly outperformed all other previously proposed agglomerative methods like GAEC [40] (GASP Sum), Greedy Fixation [54] (GASP Sum + Constraints) or Mutex Watershed [90] (GASP Abs. Max.). The competitive performance of this simple parameter-free

algorithm is also reported in Table 3, showing the current leader-board of the challenge: all entries, apart from GASP, employ superpixel-based post-processing pipelines, several of which rely on the lifted multicut formulation of [8] that uses several random forests to predict graph edge weights, relying not only on information derived from affinity maps but also raw data and superpixel shape information. Note that the test volumes contain several imaging artifacts that make segmentation particularly challenging and might profit from more robust edge statistics of super-pixel based approaches. On the other hand, the fact that our algorithm can operate on pixels directly removes the parameter

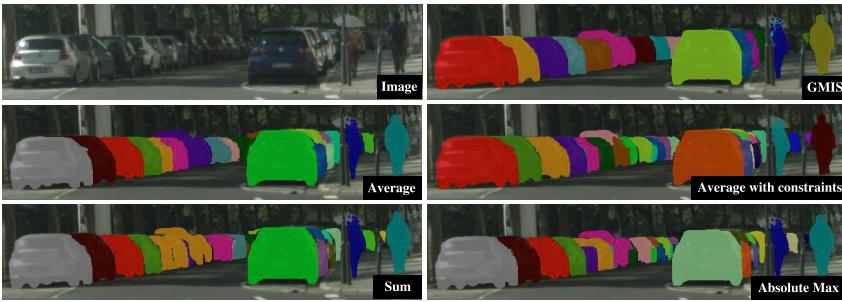


Figure 5: Visual results given by different GASP linkage criteria on a crop of a CityScapes image from the *validation* set.

tuning necessary to obtain good super-pixels and can also avoid errors that result from wrong superpixels that cannot be fixed during later agglomeration. In Appendix 8.5, we provide more details about how we scaled up GASP to the full datasets. Appendix Table 7 lists the performances and the run-times for all tested GASP linkage criteria.

Noise experiments Additionally, we conduct a set of experiments where the CNN predictions are perturbed by structured noise, in order to highlight the properties of each GASP variant and perform an in-depth comparison that is as quantitative as possible. Appendix 8.6 introduces the type of spatially correlated noise that allowed us to perturb the CNN outputs by introducing simulated additional artifacts like missing or false positive boundary evidence. Fig. 4 summarizes our 12000 noise experiments: we focus on the best performing linkage criteria, i.e. *Average*, *Sum* and *Abs Max*, and test them with different amount 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% probability².

Average and Abs Max linkage Our findings confirm that GASP with *Average* linkage criterion represents the most robust algorithm tested and the one that benefits the most from using the long-range CNN predictions. On the other hand, it is not a surprise that the *Abs Max* statistic proposed by [90] is less robust to noise than the *Average* linkage, but, as we show in the Appendix Table 7, *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

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.

Agglomeration method	AP	
GASP Average	34.3	702
GASP Average + Constraints	33.9	703
MultiStepHAC [63]	33.0	704
GASP Abs. Max. [90]	32.1	705
GASP Sum + Constraints [54]	31.9	706
GASP Sum [40]	31.3	707
		708
		709
		710
		711
		712
		713
		714
		715
		716
		717
		718
		719
		720
		721
		722
		723
		724
		725
		726
		727
		728
		729
		730
		731
		732
		733
		734
		735
		736
		737
		738
		739
		740
		741
		742
		743
		744
		745
		746
		747
		748
		749
		750
		751
		752
		753
		754
		755

Table 5: Average Precision scores on CityScapes *val* achieved by the GMIS Model trained in [63] and different graph agglomeration methods.

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 agglomerated 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

The segmentation performance of GASP is evaluated on the CityScapes dataset [16], which consists of 5000 street-scene images. Several of the current top instance segmentation methods on cityscapes predict affinities, e.g. SSAP [30] and GMIS [63]. Since [63] made the code and model publicly available, we used their pipeline consisting of two CNNs with similar architectures (one predicting semantic scores and the other pixel affinities between instances) and applied all the post-processing methods proposed by them, e.g. excluding background and resizing regions of interest. We then provided the output instance-affinities of the model as input to GASP. In [63] the instance-branch of the model was trained with a Binary Cross-Entropy loss, but we noticed how strong short-range boundary evidence was never predicted by the model. In Appendix 8.7 we present how we solved this problem by fine-tuning the model with *Sørensen-Dice* loss, similarly to [90]. Finally, the semantic categories were assigned to each instance by a majority vote based on

756	the semantic output.	810
757	Results on the <i>test</i> set are summarized in Table 4. The	811
758	best scores are achieved by Panoptic-DeepLab [10] that pro-	812
759	poses a more powerful CNN model to regress the centers	813
760	of the instances. The second best <i>proposal-free</i> method	814
761	is SSAP [30], which predicts long-range instance-affinities	815
762	similarly to GMIS [63], but trains the model with extra side-	816
763	losses. GASP with <i>Average</i> linkage also achieves competi-	817
764	tive results and outperforms the previous agglomeration	818
765	method proposed in GMIS [63]. Similarly to the experi-	819
766	ments on neuron segmentation, results on the <i>val</i> set (see	820
767	Table 5 and Fig. 5) show that other GASP linkage crite-	821
768	ria tend to over-cluster, e.g. <i>Abs Max</i> , or under-cluster and	822
769	merge instances, e.g. <i>Sum</i> . The graph-merging algorithm	823
770	proposed by [63] (MultiStepHAC) requires the user to tune	824
771	several threshold parameters and when we applied it to the	825
772	affinities predicted by our fine-tuned model it achieved an	826
773	AP score of 33.0 on the <i>val</i> set, which is worse than the	827
774	original value 34.1 reported in [63]. This is probably due to	828
775	the fact that MultiStepHAC was tailored to the output affinities	829
776	of the original model. Table 8 in Appendix includes the	830
777	scores of all other tested GASP algorithms.	831
778		832
779	<h2>6. Comparison with spectral clustering</h2>	833
780		834
781	In this section we compare GASP to spectral clustering	835
782	(SC) methods for signed graphs both on synthetic graphs	836
783	and real ones from neuron segmentation. These methods	837
784	require the user to specify the number of clusters in advance,	838
785	in contrast to our proposed agglomeration method that de-	839
786	termines the cluster number based on the signed weights	840
787	of the graph. In the following comparison, we make these	841
788	baselines as strong as possible by specifying the true num-	842
789	ber of clusters for the spectral methods.	843
790	Synthetic graphs – First, we compare the clustering	844
791	performance on synthetic graphs generated by a signed	845
792	stochastic block model (SSBM), where SC performs well.	846
793	In particular, we used an Erdős-Rényi random graph model	847
794	$\mathcal{G}(N, p)$ with $N = 10^5$ vertices and edge probability $p =$	848
795	0.1. Following the approach in [17], we partitioned the	849
796	graph into $k = 100$ equally-sized clusters, such that edges	850
797	connecting vertices belonging to the same cluster (different	851
798	clusters, respectively) had Gaussian distributed edge	852
799	weights centered at $\mu = 1$ ($\mu = -1$, respectively) and with	853
800	standard deviation 0.1. To model noise, we flipped the sign	854
801	of each edge independently with probability $\eta \in [0, 0.4]$.	855
802	Median scores, 25th and 75th percentiles over 30 repeti-	856
803	tions are shown in Fig. 6. GASP achieved scores comparab-	857
804	le to SPONGE _{sym} [17], a recently proposed SC method.	858
805	The specific design choice of a <i>sign flipping</i> noise used	859
806	in the SSBM experiments turned out to favor GASP with	860
807	<i>Sum</i> linkage that, according to the experiments presented in	861
808	Sec. 4.2 is the one with the lowest tendency to over-cluster	862
809	and grows one cluster at the time.	863
	<h2>7. Conclusion</h2>	
	We have presented a novel unifying framework for ag-	
	glomerative clustering of graphs with both positive and neg-	
	ative edge weights; and we have shown that several existing	
	clustering algorithms, e.g. the Mutex Watershed, 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 <i>Average</i> linkage	
	criterion, outperformed all the others: it proved to be a sim-	
	ple and remarkably robust approach to process short- and	
	long-range predictions of a CNN applied to an instance seg-	
	mentation task. On biological images, this simple average	
	agglomeration algorithm can represent a valuable choice for	
	a user who is not willing to spend much time tuning com-	
	plex task-dependent pipelines based on superpixels.	

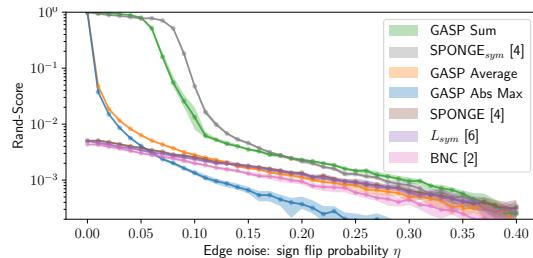


Figure 6: GASP performances compared to spectral methods on synthetic graphs. The spectral methods were given the true number of clusters as input, in contrast to GASP.

Method	Rand-Score
GASP Average	0.8966
GASP Sum	0.8965
GASP Abs Max	0.8932
SPONGE _{sym} [17]	0.5839
<i>L_{sym}</i> [47]	0.1931
SPONGE [4]	0.0789
BNC [11]	0.0074

Table 6: GASP compared to spectral clustering methods on a small crop of the CREMI dataset (sample B).

Neuron segmentation – We extended the comparison with spectral methods to the task of neuron segmentation. Since SC cannot scale to the full CREMI dataset, we evaluated GASP and SC on a smaller $10 \times 100 \times 100$ voxels volume resulting in a graph with 10^5 nodes and $\sim 10^6$ edges. Scores are summarized in Table 6. Despite the fact that the true number of ground truth clusters was given as an input to the SC methods, GASP significantly outperformed them on neuro-data. SC methods seem to have more difficulties when the graph is sparse. Moreover, they did not handle well pixels on the boundaries between segments and tended to cluster them together. Increasing k did not improve their scores either.

7. 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, 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 a user who is not willing to spend much time tuning complex task-dependent pipelines based on superpixels.

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

References

- [1] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23, 2008.
- [2] Bjoern Andres, Thorben Kroeger, Kevin L Briggman, Winfried Denk, Natalya Korogod, Graham Knott, Ullrich Koethe, and Fred A Hamprecht. Globally optimal closed-surface segmentation for connectomics. In *European Conference on Computer Vision*, pages 778–791. Springer, 2012.
- [3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- [4] Ignacio Arganda-Carreras, Srinivas C Turaga, Daniel R Berger, Dan Cireşan, Alessandro Giusti, Luca M Gambardella, Jürgen Schmidhuber, Dmitry Laptev, Sarvesh Dwivedi, Joachim M Buhmann, et al. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in neuroanatomy*, 9:142, 2015.
- [5] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5221–5229, 2017.
- [6] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine learning*, 56(1-3):89–113, 2004.
- [7] Thorsten Beier, Björn Andres, Ullrich Köthe, and Fred A Hamprecht. An efficient fusion move algorithm for the minimum cost lifted multicut problem. In *European Conference on Computer Vision*, pages 715–730. Springer, 2016.
- [8] Thorsten Beier, Constantin Pape, Nasim Rahaman, Timo Prange, Stuart Berg, Davi D Bock, Albert Cardona, Graham W Knott, Stephen M Plaza, Louis K Scheffer, et al. Multicut brings automated neurite segmentation closer to human performance. *Nature Methods*, 14(2):101, 2017.
- [9] Yi-Ting Chen, Xiaokai Liu, and Ming-Hsuan Yang. Multi-instance object segmentation with occlusion handling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3470–3478, 2015.
- [10] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab. *arXiv preprint arXiv:1910.04751*, 2019.
- [11] Kai-Yang Chiang, Joyce Jiyoung Whang, and Inderjit S Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 615–624. ACM, 2012.
- [12] Sunil Chopra and Mendum R Rao. On the multiway cut polyhedron. *Networks*, 21(1):51–89, 1991.
- [13] Sunil Chopra and Mendum R Rao. The partition problem. *Mathematical Programming*, 59(1-3):87–115, 1993.
- [14] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3D U-Net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.
- [15] Dan Ciresan, Alessandro Giusti, Luca M Gambardella, and Jürgen Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In *Advances in neural information processing systems*, pages 2843–2851, 2012.
- [16] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.
- [17] Mihai Cucuringu, Peter Davies, Aldo Glielmo, and Hemant Tyagi. SPONGE: A generalized eigenproblem for clustering signed networks. In *AISTATS*, 2019.
- [18] Mihai Cucuringu, Ioannis Koutis, Sanjay Chawla, Gary Miller, and Richard Peng. Simple and scalable constrained clustering: a generalized spectral method. In *Artificial Intelligence and Statistics*, pages 445–454, 2016.
- [19] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3150–3158, 2016.
- [20] Bert De Brabandere, Davy Neven, and Luc Van Gool. Semantic instance segmentation with a discriminative loss function. *arXiv preprint arXiv:1708.02551*, 2017.
- [21] Erik D Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theoretical Computer Science*, 361(2-3):172–187, 2006.
- [22] Lee R Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [23] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [24] Alireza Fathi, Zbigniew Wojna, Vivek Rathod, Peng Wang, Hyun Oh Song, Sergio Guadarrama, and Kevin P Murphy. Semantic instance segmentation via deep metric learning. *arXiv preprint arXiv:1703.10277*, 2017.
- [25] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International journal of computer vision*, 59(2):167–181, 2004.
- [26] Jenny Rose Finkel and Christopher D Manning. Enforcing transitivity in coreference resolution. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 45–48. Association for Computational Linguistics, 2008.
- [27] Jan Funke, Fred A Hamprecht, and Chong Zhang. Learning to segment: training hierarchical segmentation under a topological loss. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 268–275. Springer, 2015.
- [28] Jan Funke, Stephan Saalfeld, Davi Bock, Srinivasa Turaga, and Eric Perlman. Cremi challenge. <https://cremi.org/>. Accessed: 2019-11-15.
- [29] Jan Funke, Fabian David Tschopp, William Grisaitis, Arlo Sheridan, Chandan Singh, Stephan Saalfeld, and Srinivas C
- 918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971

- 972 Turaga. Large scale image segmentation with structured loss
973 based deep learning for connectome reconstruction. *IEEE*
974 *transactions on pattern analysis and machine intelligence*,
975 2018.
976 [30] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan
977 Yu, Ming Yang, and Kaiqi Huang. SSAP: Single-shot in-
978 stance segmentation with affinity pyramid. In *The IEEE In-*
979 *ternational Conference on Computer Vision (ICCV)*, October
980 2019.
981 [31] Martin Grötschel and Yoshiko Wakabayashi. A cutting plane
982 algorithm for a clustering problem. *Mathematical Program-*
983 *ming*, 45(1-3):59–96, 1989.
984 [32] Martin Grötschel and Yoshiko Wakabayashi. Facets of the
985 clique partitioning polytope. *Mathematical Programming*,
986 47(1-3):367–387, 1990.
987 [33] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Ji-
988 tendra Malik. Simultaneous detection and segmentation. In
989 *European Conference on Computer Vision*, pages 297–312.
990 Springer, 2014.
991 [34] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Gir-
992 shick. Mask r-cnn. In *Proceedings of the IEEE international*
993 *conference on computer vision*, pages 2961–2969, 2017.
994 [35] Michał Januszewski, Jörgen Kornfeld, Peter H Li, Art Pope,
995 Tim Blakely, Larry Lindsey, Jeremy Maitin-Shepard, Mike
996 Tyka, Winfried Denk, and Viren Jain. High-precision auto-
997 mated reconstruction of neurons with flood-filling networks.
998 *Nature methods*, 15(8):605, 2018.
999 [36] Jörg Hendrik Kappes, Markus Speth, Björn Andres, Gerhard
1000 Reinelt, and Christoph Schnörr. Globally optimal image par-
1001 titioning by multicut. In *International Workshop on En-*
1002 *ergy Minimization Methods in Computer Vision and Pattern*
1003 *Recognition*, pages 31–44. Springer, 2011.
1004 [37] Amirhossein Karrooost and Margret Keuper. Solving mini-
1005 mum cost lifted multicut problems by node agglomeration.
1006 In *ACCV 2018, 14th Asian Conference on Computer Vision*,
1007 Perth, Australia, 2018.
1008 [38] Verena Kaynig, Amelio Vazquez-Reina, Seymour Knowles-
1009 Barley, Mike Roberts, Thouis R Jones, Narayanan Kasthuri,
1010 Eric Miller, Jeff Lichtman, and Hanspeter Pfister. Large-
1011 scale automatic reconstruction of neuronal processes from
1012 electron microscopy images. *Medical image analysis*,
1013 22(1):77–88, 2015.
1014 [39] Brian W Kernighan and Shen Lin. An efficient heuristic pro-
1015 cedure for partitioning graphs. *Bell system technical journal*,
1016 49(2):291–307, 1970.
1017 [40] Margret Keuper, Evgeny Levinkov, Nicolas Bonneel, Guilla-
1018 me Lavoué, Thomas Brox, and Björn Andres. Efficient
1019 decomposition of image and mesh graphs by lifted multi-
1020 cuts. In *Proceedings of the IEEE International Conference*
1021 *on Computer Vision*, pages 1751–1759, 2015.
1022 [41] B Ravi Kiran and Jean Serra. Global-local optimizations by
1023 hierarchical cuts and climbing energies. *Pattern Recognition*,
1024 47(1):12–24, 2014.
1025 [42] Alexander Kirillov, Evgeny Levinkov, Bjoern Andres, Bog-
1026 dan Savchynsky, and Carsten Rother. Instancecut: from
1027 edges to instances with multicut. In *Proceedings of the IEEE*
1028 *Conference on Computer Vision and Pattern Recognition*,
1029 pages 5008–5017, 2017.
1030 [43] Seymour Knowles-Barley, Verena Kaynig, Thouis Ray
1031 Jones, Alyssa Wilson, Joshua Morgan, Dongil Lee,
1032 Daniel Berger, Narayanan Kasthuri, Jeff W Lichtman, and
1033 Hanspeter Pfister. RhoanaNet pipeline: Dense automatic
1034 neural annotation. *arXiv preprint arXiv:1611.06973*, 2016.
1035 [44] Iasonas Kokkinos. Pushing the boundaries of bound-
1036 ary detection using deep learning. *arXiv preprint*
1037 *arXiv:1511.07386*, 2015.
1038 [45] Shu Kong and Charles C Fowlkes. Recurrent pixel embed-
1039 ding for instance grouping. In *Proceedings of the IEEE Con-
1040 ference on Computer Vision and Pattern Recognition*, pages
1041 9018–9028, 2018.
1042 [46] Nikola Krasowski, Thorsten Beier, Graham W Knott, Ullrich
1043 Koethe, Fred A Hamprecht, and Anna Kreshuk. Improving
1044 3D EM data segmentation by joint optimization over bound-
1045 ary evidence and biological priors. In *2015 IEEE 12th Inter-
1046 national Symposium on Biomedical Imaging (ISBI)*, pages
1047 536–539. IEEE, 2015.
1048 [47] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch,
1049 Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak.
1050 Spectral analysis of signed graphs for clustering, prediction
1051 and visualization. SIAM, 2010.
1052 [48] L'ubor Ladický, Paul Sturges, Karteek Alahari, Chris Rus-
1053 sell, and Philip HS Torr. What, where and how many? com-
1054 bining object detectors and CRFS. In *European conference*
1055 *on computer vision*, pages 424–437. Springer, 2010.
1056 [49] Godfrey N Lance and William Thomas Williams. A gen-
1057 eral theory of classificatory sorting strategies: 1. Hierarchical
1058 systems. *The computer journal*, 9(4):373–380, 1967.
1059 [50] Jan-Hendrik Lange, Bjoern Andres, and Paul Swoboda.
1060 Combinatorial persistency criteria for multicut and max-cut.
1061 In *Proceedings of the IEEE Conference on Computer Vision*
1062 *and Pattern Recognition*, pages 6093–6102, 2019.
1063 [51] Jan-Hendrik Lange, Andreas Karrenbauer, and Bjoern And-
1064 res. Partial optimality and fast lower bounds for weighted
1065 correlation clustering. In *International Conference on Ma-*
1066 *chine Learning*, pages 2898–2907, 2018.
1067 [52] Kisuk Lee, Ran Lu, Kyle Luther, and H Sebastian Seung.
1068 Learning dense voxel embeddings for 3D neuron reconstruc-
1069 tion. *arXiv preprint arXiv:1909.09872*, 2019.
1070 [53] Kisuk Lee, Jonathan Zung, Peter Li, Viren Jain, and H Sebas-
1071 tian Seung. Superhuman accuracy on the SNEMI3D connec-
1072 toomics challenge. *arXiv preprint arXiv:1706.00120*, 2017.
1073 [54] Evgeny Levinkov, Alexander Kirillov, and Bjoern Andres. A
1074 comparative study of local search algorithms for correlation
1075 clustering. In *German Conference on Pattern Recognition*,
1076 pages 103–114. Springer, 2017.
1077 [55] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei.
1078 Fully convolutional instance-aware semantic segmentation.
1079 In *Proceedings of the IEEE Conference on Computer Vision*
1080 *and Pattern Recognition*, pages 2359–2367, 2017.
1081 [56] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Zequn Jie, Ji-
1082 aoshi Feng, Liang Lin, and Shuicheng Yan. Reversible re-
1083 cursive instance-level object segmentation. In *Proceedings*
1084 *of the IEEE Conference on Computer Vision and Pattern*
1085 *Recognition*, pages 633–641, 2016.
1086 [57] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays,
1087 Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence

- 1080 Zitnick. Microsoft coco: Common objects in context. In European conference on computer vision, pages 740–755. Springer, 2014. 1134
- 1081 [58] Shu Liu, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. Sgn: Sequential grouping networks for instance segmentation. In Proceedings of the IEEE International Conference on Computer Vision, pages 3496–3504, 2017. 1135
- 1082 [59] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8759–8768, 2018. 1136
- 1083 [60] Ting Liu, Cory Jones, Mojtaba Seyedhosseini, and Tolga Tasdizen. A modular hierarchical approach to 3D electron microscopy image segmentation. *Journal of neuroscience methods*, 226:88–102, 2014. 1137
- 1084 [61] Ting Liu, Mojtaba Seyedhosseini, and Tolga Tasdizen. Image segmentation using hierarchical merge tree. *IEEE transactions on image processing*, 25(10):4596–4607, 2016. 1138
- 1085 [62] Ting Liu, Miaomiao Zhang, Mehran Javanmardi, Nisha Ramesh, and Tolga Tasdizen. SSHMT: Semi-supervised hierarchical merge tree for electron microscopy image segmentation. In European Conference on Computer Vision, pages 144–159. Springer, 2016. 1139
- 1086 [63] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), pages 686–703, 2018. 1140
- 1087 [64] Filip Malmberg, Robin Strand, and Ingela Nyström. Generalized hard constraints for graph segmentation. In Scandinavian Conference on Image Analysis, pages 36–47. Springer, 2011. 1141
- 1088 [65] Yaron Meirovitch, Alexander Matveev, Hayk Saribekyan, David Budden, David Rolnick, Gergely Odor, Seymour Knowles-Barley, Thouis Raymond Jones, Hanspeter Pfister, Jeff William Lichtman, et al. A multi-pass approach to large-scale connectomics. *arXiv preprint arXiv:1612.02120*, 2016. 1142
- 1089 [66] Yaron Meirovitch, Lu Mi, Hayk Saribekyan, Alexander Matveev, David Rolnick, and Nir Shavit. Cross-classification clustering: An efficient multi-object tracking technique for 3-D instance segmentation in connectomics. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8425–8435, 2019. 1143
- 1090 [67] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8837–8845, 2019. 1144
- 1091 [68] Alejandro Newell, Zhiao Huang, and Jia Deng. Associative embedding: End-to-end learning for joint detection and grouping. In Advances in Neural Information Processing Systems, pages 2277–2287, 2017. 1145
- 1092 [69] Juan Nunez-Iglesias, Ryan Kennedy, Toufiq Parag, Jianbo Shi, and Dmitri B Chklovskii. Machine learning of hierarchical clustering to segment 2D and 3D images. *PloS one*, 8(8):e71715, 2013. 1146
- 1093 [70] Constantin Pape, Thorsten Beier, Peter Li, Viren Jain, Davi D Bock, and Anna Kreshuk. Solving large multicut problems for connectomics via domain decomposition. In Proceedings of the IEEE International Conference on Computer Vision, pages 1–10, 2017. 1147
- 1094 [71] Toufiq Parag, Fabian Tschopp, William Grisaitis, Srinivas C Turaga, Xuewen Zhang, Brian Matejek, Lee Kamentsky, Jeff W Lichtman, and Hanspeter Pfister. Anisotropic EM segmentation by 3D affinity learning and agglomeration. *arXiv preprint arXiv:1707.08935*, 2017. 1148
- 1095 [72] Ken Perlin. An image synthesizer. *ACM Siggraph Computer Graphics*, 19(3):287–296, 1985. 1149
- 1096 [73] Ken Perlin. Noise hardware. *Real-Time Shading SIGGRAPH Course Notes*, 2001. 1150
- 1097 [74] Lorenzo Porzi, Samuel Rota Bulo, Aleksander Colovic, and Peter Kontschieder. Seamless scene segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8277–8286, 2019. 1151
- 1098 [75] Syama Sundar Rangapuram and Matthias Hein. Constrained 1-spectral clustering. In AISTATS, volume 30, page 90, 2012. 1152
- 1099 [76] Mengye Ren and Richard 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. 1153
- 1100 [77] Zhile Ren and Gregory Shakhnarovich. Image segmentation by cascaded region agglomeration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2011–2018, 2013. 1154
- 1101 [78] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In European conference on computer vision, pages 312–329. Springer, 2016. 1155
- 1102 [79] Olaf Ronneberger, Philipp Fischer, and Thomas 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. 1156
- 1103 [80] Stephan Saalfeld, Richard Fetter, Albert Cardona, and Pavel Tomancak. Elastic volume reconstruction from series of ultra-thin microscopy sections. *Nature methods*, 9(7):717, 2012. 1157
- 1104 [81] Philippe Salembier and Luis 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. 1158
- 1105 [82] Philipp Schlegel, Marta Costa, and Gregory SXE Jefferis. Learning from connectomics on the fly. *Current opinion in insect science*, 24:96–105, 2017. 1159
- 1106 [83] Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In International Conference on Medical Image Computing and Computer-Assisted Intervention, pages 265–273. Springer, 2018. 1160
- 1107 [84] Konstantin Sofiuk, Olga Baranova, and Anton Konushin. AdaptIS: Adaptive instance selection network. In Proceedings of the IEEE International Conference on Computer Vision, pages 7355–7363, 2019. 1161
- 1108 [85] Thorvald 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. 1162
- 1109 [1134] 1135
- 1110 [1136] 1137
- 1111 [1138] 1139
- 1112 [1140] 1141
- 1113 [1142] 1143
- 1114 [1144] 1145
- 1115 [1146] 1147
- 1116 [1148] 1149
- 1117 [1150] 1151
- 1118 [1152] 1153
- 1119 [1154] 1155
- 1120 [1156] 1157
- 1121 [1158] 1159
- 1122 [1160] 1161
- 1123 [1162] 1163
- 1124 [1164] 1165
- 1125 [1166] 1167
- 1126 [1168] 1169
- 1127 [1170] 1171
- 1128 [1172] 1173
- 1129 [1174] 1175
- 1130 [1176] 1177
- 1131 [1178] 1179
- 1132 [1180] 1181
- 1133 [1182] 1183
- 1134 [1184] 1185
- 1135 [1186] 1187

- 1188 [86] Srinivas C Turaga, Kevin L Briggman, Moritz Helmstaedter, 1242
1189 Winfried Denk, and H Sebastian Seung. Maximin affinity 1243
1190 learning of image segmentation. pages 1865–1873, 2009. 1244
1191 [87] Mustafa Gokhan Uzunbas, Chao Chen, and Dimitris 1245
1192 Metaxas. An efficient conditional random field approach for 1246
1193 automatic and interactive neuron segmentation. *Medical im-* 1247
1194 *age analysis*, 27:31–44, 2016. 1248
1195 [88] Xiang Wang, Buyue Qian, and Ian Davidson. On constrained 1249
1196 spectral clustering and its applications. *Data Mining and* 1250
1197 *Knowledge Discovery*, 28(1):1–30, 2014. 1251
1198 [89] Steffen Wolf, Alberto Bailoni, Constantin Pape, Nasim Ra- 1252
1199 haman, Anna Kreshuk, Ullrich Köthe, and Fred A Ham- 1253
1200 precht. The mutex watershed and its objective: Efficient, 1254
1201 parameter-free image partitioning. *arXiv preprint arXiv:1904.12654*, 2019. 1255
1202 [90] Steffen Wolf, Constantin Pape, Alberto Bailoni, Nasim Ra- 1256
1203 haman, Anna Kreshuk, Ullrich Kothe, and FredA Ham- 1257
1204 precht. The mutex watershed: Efficient, parameter-free im- 1258
1205 age partitioning. In *Proceedings of the European Conference* 1259
1206 *on Computer Vision (ECCV)*, pages 546–562, 2018. 1260
1207 [91] Saining Xie and Zhuowen Tu. Holistically-nested edge de- 1261
1208 tection. In *Proc. ICCV'15*, pages 1395–1403, 2015. 1262
1209 [92] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min 1263
1210 Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified 1264
1211 panoptic segmentation network. In *Proceedings of the IEEE* 1265
1212 *Conference on Computer Vision and Pattern Recognition*, 1266
1213 pages 8818–8826, 2019. 1267
1214 [93] Yi Yang, Sam Hallman, Deva Ramanan, and Charless C 1268
1215 Fowlkes. Layered object models for image segmentation. 1269
1216 *IEEE Transactions on Pattern Analysis and Machine Intelli-* 1270
1217 *gence*, 34(9):1731–1743, 2012. 1271
1218 [94] Julian Yarkony, Alexander Ihler, and Charless C Fowlkes. 1272
1219 Fast planar correlation clustering for image segmentation. In 1273
1220 *European Conference on Computer Vision*, pages 568–581. 1274
1221 Springer, 2012. 1275
1222 [95] Tao Zeng, Bian Wu, and Shuiwang Ji. DeepEM3D: ap- 1276
1223 proaching human-level performance on 3D anisotropic EM 1277
1224 image segmentation. *Bioinformatics*, 33(16):2555–2562, 1278
1225 2017. 1279
1226 1280
1227 1281
1228 1282
1229 1283
1230 1284
1231 1285
1232 1286
1233 1287
1234 1288
1235 1289
1236 1290
1237 1291
1238 1292
1239 1293
1240 1294
1241 1295