

Rebuttal for ‘Guided Proofreading of Automatic Segmentations for Connectomics’

Thank you for your time and constructive comments. We will fix all minor issues.

R2: Quantitative Evaluation R2 requests an objective quantitative evaluation. Fortunately, our ‘automatic’ and ‘oracle’ experiments are exactly this. We define such experiments in lines 573–590 and report the results in Fig. 6, Fig. 7, and lines 792–818 (also in supplemental Sec. 2 and 3). These evaluations use the quantitative VI metric against a ground truth segmentation, with no user in the loop.

R2: Faster than State of the Art? ‘Faster’ here considers both how long each correction takes, and the likely VI reduction from that correction. Our approach has comparable correction time to the state-of-the-art Focused Proofreading approach, but a significantly greater VI reduction per correction (7.5 \times). Our presentation of this discovery could have been improved. We will add Table 1 to make this clearer (previously reported in lines 756–765, slopes in Fig. 6, and column 3 in Fig. 7).

Table 1: Average proofreading speed for users of Dojo, Focused Proofreading (FP) and our Guided Proofreading (GP). For comparable correction time, our system achieves significantly higher VI reduction per minute (7.5 \times) over state-of-the-art FP.

Approach	Time Per Correction (s)	VI Reduction Per Minute
<i>Dojo</i>	30.5	-0.002
<i>FP</i>	4.9	0.00023
<i>GP</i>	6.2	0.00173

R2: Reproducibility R2 expresses concerns regarding reproducibility. As per line 847, we have promised to release all our code and data, and we disclose all parameters.

R2: How were Optimal Parameters chosen? The threshold $p_t = 0.95$ was observed to be stable when evaluating on previously-unseen test data (lines 585–586, supplemental Sec. 1.3). The **input border is dilated by 5 pixels** to consider slight edge ambiguities and to cover extracellular space between segments in high-resolution electron microscopy data (lines 308–310). During merge error detection, **labels are dilated by 20 pixels** prior to finding potential borders (line 323) with border-seeded watershed—this way the borders tend to attach to real membrane boundaries (lines 364–366). As we use a CNN, there are many additional parameters; we consider choosing these effectively to

be an open problem, and for this we used learned experience and local brute-force searches.

R3: Training Datasets—U-net vs. GP? R3 raises the question of whether our GP approach was trained on the same data as membrane detection (U-net). There was no overlap (Tab. 2).

Table 2: Training data of membrane detection vs. training data of GP (for supplemental material).

Dataset	Training Set U-Net	Training Set GP
<i>L. Cylinder</i>	AC3+AC4 (1024 \times 1024 \times 175vx)	L. Cylinder (2048 \times 2048 \times 250vx)
<i>AC4 subvolume</i>	AC4 excl. test (1000 \times 1000 \times 90vx)	L. Cylinder (2048 \times 2048 \times 250vx)
<i>CREMI A/B/C</i>	AC3+AC4 (1024 \times 1024 \times 175vx)	CREMI A/B/C (1250 \times 1250 \times 300vx)

R3: Merge Error Detection R3 requests a better explanation of the merge error detection. We have updated Figure 4 in the main paper to include the watershed seeds (Fig. 1). We will also add a pseudo code version of the algorithm to the supplemental material to promote understanding (Alg. 1).

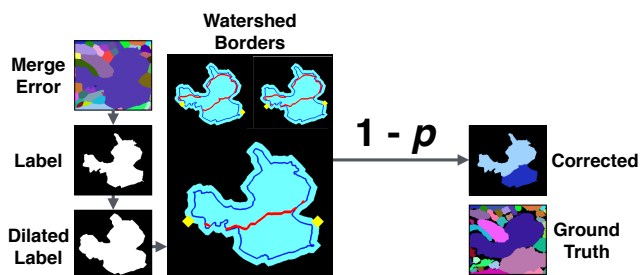


Figure 1: Updated Figure 4 including the random watershed seeds (yellow).

Algorithm 1 Merge Error Detection for label l

- 1: $l_d = \text{dilate}(l, 20)$
- 2: $\text{border} = \text{watershed}(l_d, \text{seeds}_{\text{random}})$
- 3: $p_{\text{split}} = \text{rank}(\text{border})$
- 4: $p_{\text{merge}} = 1 - p_{\text{split}}$

R3: GALA Active Learning Classifier In our automatic segmentation pipeline (line 499), GALA uses a random forest classifier to agglomerate segments. While it does not

108		162
109	Algorithm 2 Merge Error Detection for one label	163
110	1: dilate label	164
111	2: for N iterations do	165
112	3: watershed (randomly seeded)	166
113	4: p = rank border using split error CNN	167
114	5: $p_{\text{merge}} = 1 - p$	168
115	6: find max p_{merge}	169
116		170
117		171
118	require user interaction, it does require parameters. We will	172
119	add a section to the supplemental material containing a full	173
120	description of the approach and our use of it.	174
121		175
122		176
123		177
124		178
125		179
126		180
127		181
128		182
129		183
130		184
131		185
132		186
133		187
134		188
135		189
136		190
137		191
138		192
139		193
140		194
141		195
142		196
143		197
144		198
145		199
146		200
147		201
148		202
149		203
150		204
151		205
152		206
153		207
154		208
155		209
156		210
157		211
158		212
159		213
160		214
161		215