Guided Proofreading of Automatic Segmentations for Connectomics

Thank you for your constructive comments. We will fix all minor issues. We would like to clarify and correct the following major remarks.

1. Quantitative Evaluation

Reviewer 2 requests an objective quantitative evaluation. 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). The evaluation is fully numeric and we report VI scores. We will change the wording in the manuscript to emphasize this.

2. Reproducibility

Reviewer 2 expresses concerns regarding reproducibility. However, we define all parameters in the manuscript and promise to release code and data (line 847).

3. Optimal Parameters

Reviewer 2

4. Training Datasets U-net vs. GP

Reviewer 3 raises the question if GP was trained on the same data as membrane detection (U-net). There was no overlap (Tab. 1).

Table 1: Training data of membrane detection vs. training data of GP (for supplemental material).

Dataset	Training Set Membrane Detection (U-Net)	Training Set Guided Proofreading
L. Cylinder	AC3+AC4 ($1024 \times 1024 \times 175$vx)	L. Cylinder ($2048 \times 2048 \times 250$vx)
AC4 subvolume	AC4 excl. test ($1024 \times 1024 \times 90$vx)	L. Cylinder ($2048 \times 2048 \times 250$vx)
CREMI A/B/C	AC3+AC4 ($1024 \times 1024 \times 175$vx)	CREMI A/B/C ($1250 \times 1250 \times 300$vx)

5. Faster Proofreading

We agree with reviewer 2 that our claim that GP enables faster proofreading is not clearly presented. We report the average correction times for novice users in lines 756-765. Figure 7 (column 3) shows VI reduction after 30 minutes and the slopes in figure 6 also indicate better performance for GP. However, this presentation is not ideal and we will add Tab. 2 to the paper.

Table 2: Average proofreading speed for novice users of Dojo, FP and GP. Higher VI reduction per minute shows better performance of GP.

	Correction Time [s]	VI Reduction per minute
Dojo	30.5	-0.002
FP	4.9	0.00023
GP	6.2	0.00173

6. Merge Error Detection

Reviewer 3 suggests a better explanation of the merge error detection. We updated figure 4 in the paper to include the watershed seeds (Fig. 1). We will also add a pseudo code description of the algorithm to the supplemental material. We hope that this will make this part easier to understand.

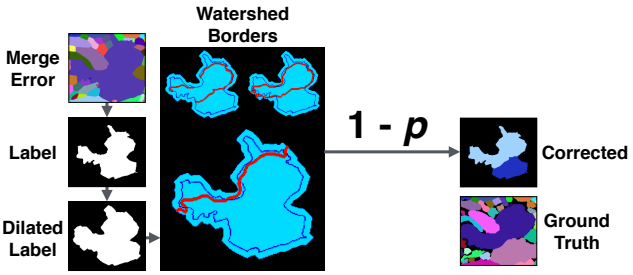

Figure 1: Updated figure 4, including the random watershed seeds. (TODO)

7. GALA Active Learning Classifier

We use GALA in our automatic segmentation pipeline (line 499). GALA uses a random forest classifier to agglomerate segments. While it does not require user interaction, it requires parameters. We will either add a reference to our yet unpublished segmentation pipeline or add a section to the supplemental material describing it in more detail as requested by reviewer 3.