1. Error Measures for Neuron Reconstruction

In the following, we present a brief survey of commonly used error measures for neuron reconstruction. For that, we use the label notation introduced in Section 2 of the main manuscript. Furthermore, we say that a labeling \( y \) imposes a partition \( Y \) of \( \Omega \) into non-empty subsets

\[
Y_k = \{ i \in \Omega \mid y(i) = k \},
\]

such that

\[
Y = \bigcup_{k \in K_y} Y_k, \quad \forall k \neq j : Y_k \cap Y_j = \emptyset .
\]

1.1. Rand index

The Rand index (RI) directly compares two partitions \( X \) and \( Y \) in terms of the agreement of pairs of elements in both partitions. Two partitions are said to agree on a pair \((i, j) \in \Omega \times \Omega\), if both partitions assign the pair to the same subset, \( i.e.\),

\[
x(i) = x(j) \quad \text{and} \quad y(i) = y(j),
\]

or both partitions assign the pair to two different subsets, \( i.e.\),

\[
x(i) \neq x(j) \quad \text{and} \quad y(i) \neq y(j).
\]

Let \( a \) be the number of agreeing pairs and \( n = \binom{\lvert \Omega \rvert}{2} \) the number of all pairs in \( \Omega \). The Rand index is then given by

\[
RI(X, Y) = \frac{a}{n},
\]

\( i.e.\), the fraction of agreeing pairs on all possible pairs. Hence, the Rand index is 1 for two equivalent partitions and 0 in the worst case where one partition contains only one subset of size \( \lvert \Omega \rvert \) and the other one \( \lvert \Omega \rvert \) partitions of size one\(^1\). In practice, where we have one of the partitions fixed as the ground truth, a Rand index of zero is therefore never reached.

\(^1\)Proof by contradiction: Assume that one of the worst case partitions contains equal and unequal pairs, then we can find a triple \((i, j, l)\) such that \( i = j, i \neq l, j \neq l \). In the other partition, to find no agreement, we would need to have \( i \neq j, i = l, \) and \( j = l \), which is not possible. Hence, the partitions can contain either only unequal pairs or only equal pairs.

1.2. Variation of Information

The variation of information (VOI) measures the similarity of two partitions \( X \) and \( Y \) indirectly by considering two random variables \( A \) and \( B \) and their conditional probabilities. \( A \) and \( B \) represent the chance of obtaining a certain label \( k \) in \( X \) and a label \( l \) in \( Y \) when randomly and uniformly selecting a location \( i \in \Omega \). Thus,

\[
p(A = k) = \frac{|X_k|}{|\Omega|} \quad \text{and} \quad p(B = l) = \frac{|Y_l|}{|\Omega|},
\]

and their conditional distributions are

\[
p(A = k|B = l) = \frac{|X_k \cap Y_l|}{|Y_l|} \quad \text{and} \quad p(B = l|A = k) = \frac{|X_k \cap Y_l|}{|X_k|}.
\]

In the following, we will use \( p(l) \overset{\text{def}}{=} p(B = l) \) and \( p(k) \overset{\text{def}}{=} p(A = k) \) as a shorthand. Now, the VOI of these two random variables is the sum of the two conditional entropies of one variable given the other:

\[
\text{VOI}(A, B) = H(A|B) + H(B|A).
\]

The conditional entropy \( H(A|B) \) gives the number of bits needed to determine the value of \( A \), if \( B \) is already known; and vice versa for \( H(B|A) \). More formally,

\[
H(A|B) = \sum_l p(l) H(A|l) = \sum_l p(l) \sum_k -p(k|l) \log(p(k|l)).
\]
Figure 1: VOI shown on an example of splitting one region $X_1$ into two regions $Y_1$ and $Y_2$ (a) and three regions $Y_1$, $Y_2$, and $Y_3$ (b). The plots show the effect of moving (a) the split position and (b) one of the split positions along the whole $Y$. It can clearly be seen that the VOI does depend on the split position. Furthermore, there are several situations where splitting into three regions gives a lower VOI than the worst case VOI for splitting into two regions (dotted line).

Intuitively, the two summands of the VOI give a measure for the split or merge error of a reconstruction in comparison to the ground truth. Assume that $A$ corresponds to the labels of the ground truth, and $B$ of the reconstruction. If a ground truth neuron with label $k$ was accidentally split into two neurons in the reconstruction, the number of bits needed to determine $B$ for all locations where $A = k$ will be bigger than zero and at most one. Conversely, if two ground truth neurons got accidentally merged into a single neuron with label $l$, the number of bits needed to determine $A$ for all locations where $B = l$ is bigger than zero. An example of this measure for the split of a single region can be seen in Figure 1.

1.3. Warping Error

The warping error by Jain et al. aims to measure the difference between a ground truth and a reconstruction in terms of their topological differences [2]. As such, it is the first error measure for neuron reconstruction that deals with the delicate question up to which point boundary shifts are not considered as errors.

The ground truth and the reconstruction are assumed to be given as foreground vs. background labelings, such that each connected component of the foreground color represents a neuron. This implies that two neurons have to be separated by at least one pixel width background sheet. This is in contrast to our definition of a labeling (see Section 2 in the main manuscript), where the identity of a neuron is maintained by the label instead of the connectedness.

The warping error is the minimal Hamming distance between two binary labelings $x$ and $y$ over a set of possible warpings of $y$. A warping is a sequence of topology-preserving relabelings of single locations $i \in M \subseteq \Omega$. This set $M$ is called the mask and encodes the tolerable deviation between the ground truth labeling and the reconstruction: Labels within $M$ can be changed, as long as the topology of the involved connected components does not change. In particular, this allows the shift of object boundaries within $M$, so that a slightly mislabeled boundary does not contribute to the final error measure. More formally, the warping error between two labelings $x$ and $y$ is

$$ W(x, y) = \min_{y' \in \mathcal{O}} \sum_{i \in \Omega} (x(i) - y'(i))^2, $$

where $y' \prec y$ denotes a warping $y'$ of a labeling $y$.

The mask $M$ is defined as the set of all image locations that are within a threshold Euclidean distance to the background label of the ground truth. Thus, reconstructed neurons are allowed to grow and shrink by a certain amount, unless they touch or split.

Since finding the exact warping error by solving Eq. (13) is not trivial, Jain et al. propose to follow an efficient greedy optimization strategy. Of all possible relabelings of single locations in $y$ that would reduce the Hamming distance, one of them is performed randomly until no further improvements can be made.

1.4. Anisotropic Edit Distance

The anisotropic edit distance (AED) that was introduced in [1] counts the errors of a reconstruction in terms of the edit distance to the ground truth. Four different types of errors that are specific to anisotropic neuron reconstruction are counted: missed and spurious neuron slices within a section, and missed and spurious links between the neuron slices across sections (see Figure 2 for an illustration).

The central concept of this error measure is the matching of neuron slices (i.e., 2D neuron cross-sections) in each section between the reconstruction and the ground truth. Neuron slices can either be matched one-to-one (if they are similar), or not at all. Let $S_R$ be the set of all neuron slices of the reconstruction in all sections, and $S_G$ the corresponding set of slices in the ground truth. A matching $m = (V, E)$ of the neuron slices is a bipartite graph with

$$ V = S_R \cup S_G, $$

$$ E \subseteq S_R \times S_G, $$

where the order of nodes is at most one to ensure that each neuron slice is matched at most once.

For any given matching $m$ of neuron slices, the AED is evaluated in the following way: Every not matched neuron slice in the reconstruction is a false positive (FP), every not matched neuron slice in the ground truth is a false negative (FN). Let $d_m : V \mapsto \{0, 1\}$ be the degree of a slice $v \in V$ under a mapping $m$, i.e., it is one if the slice $v$ was matched
Further, the links (i.e., 2D neuron cross-sections) are matched with the ground truth based on a threshold on their overlap ratio. Ground truth slices that have not been matched are counted as false negatives (FN), reconstruction slices that have not been matched are counted as false positives (FP). Links connecting slices across sections are matched between the reconstruction and ground truth, if the involved slices match. Ground truth links that have not been matched are counted as false splits (FS), reconstruction links that have not been matched are counted as false merges (FM). If there are multiple ways to match slices between the reconstruction and ground truth, the matching that minimizes the total number of errors is chosen.

and zero otherwise. The FP and FN errors can now be expressed as

\[
FP(m) = \sum_{v \in S_R} 1 - d_m(v) \quad (16)
\]
\[
FN(m) = \sum_{v \in S_G} 1 - d_m(v). \quad (17)
\]

Further, the links (i.e., the connections between the neuron slices of two subsequent sections) are matched between the ground truth and the reconstruction. Two links are said to match, if both of the involved slices match. Every not matched link in the reconstruction is a false merge (FM), every not matched link in the ground truth is a false split (FS). Let \( L_R \subset S_R \times S_R \) be the set of links in the reconstruction, and \( L_G \) the corresponding set of links in the ground truth. The FS and FM errors can now be expressed as

\[
FS(m) = \sum_{(s,s') \in L_R} 1 - \sum_{(s',t') \in L_G} \mathbb{1}_{\{(s,s') \in E\}} \mathbb{1}_{\{(t,t') \in E\}} \quad (18)
\]
\[
FM(m) = \sum_{(s,t) \in L_G} 1 - \sum_{(s',t') \in L_R} \mathbb{1}_{\{(s,s') \in E\}} \mathbb{1}_{\{(t,t') \in E\}}. \quad (19)
\]

where \( \mathbb{1}_x \) returns 1 if \( x \) is true, otherwise 0.

Since this procedure is sensitive to the matching \( m \) of the neuron slices, an optimal matching that minimizes the total number of errors is selected in the following way: First, all possible neuron slice matchings between the ground truth and the reconstruction are enumerated for each section individually. For that, only neuron slices that share at least a certain percentage \( k \) of the union of their pixels are considered as match candidates. This gives a set \( M_k \) of possible matchings for each section \( z \). Each matching \( m^z \in M_k^z \) proposes a number of FP in \( m^z \) and FN in \( m^z \) errors for this particular section. Similarly, each pair

\[
(m^z, m^{z+1}) \in M_k^z \times M_k^{z+1}
\]

of matchings of consecutive sections \( z \) and \( z + 1 \) proposes a number of FS in \( m^z \cup m^{z+1} \) and FM in \( m^z \cup m^{z+1} \) errors.

Let \( n \) be the number of sections in the stack, and \( x \) and \( y \) the reconstruction and ground truth labeling, from which the sets of neuron slices \( S_R, S_G \) and the sets of links \( L_R, L_G \) are extracted. Let the set of all possible matchings for the whole stack be given as

\[
\mathcal{M}_k = M_k^1 \times \ldots \times M_k^n.
\]

The AED is now the minimal number of total errors for a similarity threshold \( k \), and can be found by the following minimization:

\[
\text{AED}_k(x, y) = \min_{m \in \mathcal{M}_k} \sum_{z=1}^n \text{FP}(m^z) + \text{FN}(m^z) + \sum_{z=2}^n \text{FS}(m^{z-1} \cup m^z) + \text{FM}(m^{z-1} \cup m^z). \quad (20)
\]

The matching minimizing the AED is found in a forward-backward like scheme, similar to inference on a Markov chain, where each node represents the possible matchings of a section. If the similarity threshold \( k \) is smaller or equal to 50\%, there is only one possible matching per section. In this case, there is no need to perform an optimization.

2. A Bundle Method for Regularized Structured Risk Minimization

In the following, we describe our implementation of a bundle method to minimize the structured risk

\[
L(w) = \Omega(w) + M(w), \quad (22)
\]

where \( \Omega(w) = \lambda ||w||^2 \) is a quadratic regularizer with weight \( \lambda \) and

\[
M(w) = \max_{z \in Z} E(d', z) - E(d', z) + \Delta(z', z) \quad (23)
\]
further that the energy $E_\phi$ or equal to $a$ where we write consistency between the indicator variables.

$Z$ is the structured loss of the margin rescaling variant introduced in Section 3 of the main manuscript.

For that, we assume without loss of generality\footnote{Every discrete energy model can be transformed into an equivalent model with binary indicator variables and linear constraints that ensure consistency between the indicator variables.}, that the set $Z$ of responses is the set of all binary vectors of length $n$ subject to linear constraints

$$Z = \{ z \in \{0, 1\}^n | Az \leq b \},$$

(24)

where we write $a \preceq b$ to say that $a$ is element-wise smaller or equal to $b$. Again without loss of generality, we assume further that the energy $E(d, z)$ is defined in terms of feature vectors $\phi_i(d) : \mathcal{X} \rightarrow \mathbb{R}^m$ for each component $z_i$ of

$z$, where the binary $z_i$ enable or disable the contribution of feature vector $\phi_i(d)$ to the energy. More formally, we define the energy to be of the form

$$E(d, z) = \sum_{i=1}^{n} \langle w, \phi_i(d) \rangle z_i = \langle w, \phi(d)z \rangle,$$

(25)

where we write $\phi(d)$ to denote the matrix of all feature vectors $\phi_i(d)$ as columns. We refer to models satisfying Eq. (24) and Eq. (25) as linear constrained binary models.

In our implementation, we restrict $\Delta(z', z)$ to be linear in $z$ as well, i.e.,

$$\Delta(z', z) = (\Delta_1, z) + \Delta_c,$$

(26)
where $\Delta_1 \in \mathbb{R}^n$ and $\Delta_c \in \mathbb{R}$ depend on $z'$. This restriction is not sacrificing expressiveness of the cost function, since arbitrarily complex cost functions can be implemented by augmenting the vector $z$ and introducing new constraints. The resulting model will still be a linear constrained binary model.

Put together, the restrictions in Eq. (24), Eq. (25), and Eq. (26) allow us to perform the maximization of the margin rescaling variant of the loss $M(w)$ by solving an integer linear program (ILP), without losing generality of the models that can be handled. As we will see soon, this maximization has to be carried out repeatedly to find the optimal weights.

Our implementation uses a quadratic regularizer, i.e., $\Omega(w) = \lambda |w|^2$. In this case, the empirical risk $L(w)$ is piecewise quadratic and convex in $w$, despite the possible flexibility that $\Delta(z', z)$ allows. As we see in Figure 3, the reason for that is that $M(w)$ is the maximum over a set of hyperplanes. Thus, $M(w)$ is convex in $w$ and so is $L(w)$, which is just the sum of a quadratic regularizer and $M(w)$. Theoretically, the optimal set of weights could thus be found by solving the quadratic program

$$
\min_{w, \xi} \lambda |w|^2 + \xi, \quad \text{s.t. } \forall z \in \mathcal{Z} : \langle a_z, w \rangle + b_z \leq \xi, \quad (27)
$$

where $\langle a_z, w \rangle + b_z = 0$ defines the hyperplane corresponding to configuration $z$. In practice, however, the enumeration of the exponentially many constraints (one for each configuration $z \in \mathcal{Z}$) is intractable.

To avoid enumeration of all those hyperplanes, we exploit the fact that around the minimum, $M(w)$ can be described by only a few hyperplanes. We use a cutting plane method to discover those hyperplanes. For that, we repeatedly solve $M(w)$ for different weight vectors. Each found hyperplane is used to refine a lower bound $L(w)$ of $L(w)$, which we minimize to obtain the next weight vector to sample.

Our implementation follows the framework of Bundle Methods for Regularized Risk Minimization described in [3].

In Algorithm 1, Algorithm 2, and Algorithm 3 we provide details about the cutting plane technique we use.

### 3. Experiment Features

A complete list of all features of the assignment model that we used for our experiments is given in Table 1. The features are extracted for three different assignment types: continuations (C) assign one 2D slice candidate from one section to one in an adjacent section, branches (B) assign one candidate to two, and ends (E) state that a candidate does not have a counterpart.

For C and B, the features are extracted by comparing the involved candidates across sections. For example, center distance (CD) measures the distance between the center of mass of candidates in one section to the center of mass of the candidates in the next section. The feature slice distance (SD) and it variances measure for every pixel of a candidate its minimal distance to any pixel of another candidate. The aligned versions of some features are obtained by first aligning the candidates, such that their centers of mass line up.

The type features are binary indicators for each of the assignment types, such that priors can be learnt.

For the Mouse Cortex dataset, where more training data is available and therefore the risk of over-fitting is lower, we added the square of each feature to the feature vector as well.

### References

1. Jan Funke, Bjoern Andres, Fred A. Hamprecht, Albert Cardona, and Matthew Cook, Efficient Automatic 3D-Reconstruction of Branching Neurons from EM Data, CVPR, 2012, pp. 1004–1011.

2. Viren Jain, Benjamin Bollmann, Mark Richardson, Daniel R. Berger, Moritz N. Helmstaedter, Kevin L. Briggman, Winfried Denk, Jared B. Bowden, John M. Mendenhall, Wickliffe C. Abraham, Kristen M. Harris, Narayanan Kasthuri, Ken J. Hayworth, Richard Schalek, Juan Carlos Tapia, Jeff W. Lichtman, and H. Sebastian Seung, Boundary Learning by Optimization with Topological Constraints, CVPR, 2010.

3. Choon H. Teo, S. V. N. Vishwanathan, Alex Smola, and Quoc Le, Bundle Methods for Regularized Risk Minimization, Journal of Machine Learning Research 11 (2010), 311–365.
Figure 3: Visualization of the margin rescaling loss function $M(w)$ on a toy problem with a single real-valued feature for each $z \in Z$ (shown in Figure (c)). Figure (a) shows the loss for a given weight vector $w = 1$. $M(w)$ is the maximum of the individual losses over all $z \in Z$ (green line). The individual losses are composed of an energy difference $\delta E(z', z) = E(d', z') - E(d', z)$ to the ground truth (shown as black dots) and the cost for misclassification $\Delta(z', z)$ (shown as blue vertical lines). The goal of the learning procedure is to find feature weights $w$ that minimize $M(w)$, i.e., the loss of the currently worst response $\tilde{z}$. For that, responses with high costs have to be pushed down by increasing their energy and thus decreasing the difference $\delta E(z', z)$, regardless whether their energy is already higher than the ground truth energy. The dependency between $w$ and the energy difference $\delta E(z', z)$ is linear, with the coefficients being the feature differences between $z'$ and $z$ (see Eq. (25)). Thus, $M(w)$ is piecewise linear, as shown in Figure (b). Although $M(w)$ is also convex, minimizing it is difficult because of the large number of possible responses.
| Category   | Feature | Description                  | Assignment Type |
|------------|---------|------------------------------|-----------------|
| Geometry   | CD      | center distance              | C,B             |
|            | SE      | set difference               | C,B             |
|            | SR      | set difference ratio         | C,B             |
|            | ASE     | aligned set difference       | C,B             |
|            | ASR     | aligned set difference ratio | C,B             |
|            | SI      | size                         | E              |
|            | SD      | size difference               | C,B             |
|            | O       | overlap                      | C,B             |
|            | OR      | overlap ratio                 | C,B             |
|            | AD      | average slice distance        | C,B             |
|            | MD      | max slice distance            | C,B             |
|            | AAD     | aligned average slice distance| C,B             |
|            | AMD     | aligned max slice distance    | C,B             |
| Texture    | H0      | bin 0                         | C,B,E           |
|            | H1      | bin 1                         | C,B,E           |
|            | H2      | bin 2                         | C,B,E           |
|            | H3      | bin 3                         | C,B,E           |
|            | H4      | bin 4                         | C,B,E           |
|            | H5      | bin 5                         | C,B,E           |
|            | H6      | bin 6                         | C,B,E           |
|            | H7      | bin 7                         | C,B,E           |
|            | H8      | bin 8                         | C,B,E           |
|            | H9      | bin 9                         | C,B,E           |
| histogram  | N0      | bin 0                         | C,B,E           |
|            | N1      | bin 1                         | C,B,E           |
|            | N2      | bin 2                         | C,B,E           |
|            | N3      | bin 3                         | C,B,E           |
|            | N4      | bin 4                         | C,B,E           |
|            | N5      | bin 5                         | C,B,E           |
|            | N6      | bin 6                         | C,B,E           |
|            | N7      | bin 7                         | C,B,E           |
|            | N8      | bin 8                         | C,B,E           |
|            | N9      | bin 9                         | C,B,E           |
| Type       | IC      | is continuation assignment    | C,B,E           |
|            | IB      | is branch assignment          | C,B,E           |
|            | IE      | is end assignment             | C,B,E           |

Table 1: List of all features and assignment cases they are used in: C: continuation, B: branching, E: end assignments.