A From Leaf-Labeled to Labeled Merge Trees

As consequences of the algorithms in Section 4, we arrive at an ensemble of (updated) leaf-labeled merge trees together with their 1-center, which are in full agreement with respect to a shared leaf label set S. For visual embeddings and animations (Sec. 6), we need to further infer a 1-to-1 correspondence between internal vertices between $\mathscr{T}^i$ and $\mathscr{T}$ for each i. That is, we infer a complete labeling for internal vertices. Our algorithm is as follows:

S1. Transform $\mathscr{T}$ into a pivot tree $\mathscr{T}^p$. Let S_p be its label set.

S2. For each element $\mathscr{T}^i$, update its labeling using S_p.

We give a toy example in Figure 15.

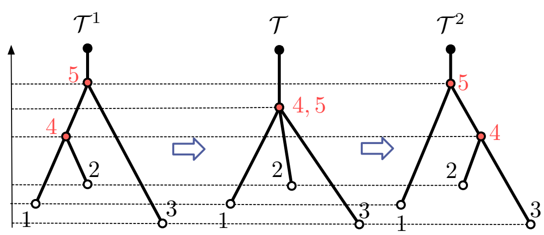

Fig. 15. Inferring a complete labeling for internal vertices based on leaf labels and minimum cost matching.

In Step S1, we select a tree $\mathscr{T}^j$ with the largest number of vertices. If $\mathscr{T}^j = \mathscr{T}$, then $\mathscr{T}^p := \mathscr{T}$. Otherwise, add dummy vertices to T with respect to T^j so that it becomes a tree T' with the largest number of vertices, therefore transforming $\mathscr{T} = (T, f, \omega)$ into a pivot tree $\mathscr{T}^p = (T_p, f_p, \omega_p)$, where $\omega_p : S_p \to V$ is an extension of $\omega : S \to L$ that is surjective on the vertices.

In Step S2, we run the algorithm for trees in partial agreement (Sec. 4.2) so that each $\mathscr{T}^i$ is updated to be $\mathscr{T}^{i'} = (T_i, f_i, \omega_i')$, where $\omega_i' : S_p \to V_i$ is surjective on its vertices.

B Uncertainty Visualization: Edge Consistency

Edge consistency. The vertex consistency can be extended to edge consistency. For a vertex consistency function $\alpha : V \to \mathbb{R}$ defined on the vertex set V of $\mathscr{T}$, it can be extended to be an edge consistency function using piecewise-linear (PL) $\beta^{PL} : |E| \to \mathbb{R}$ or piecewise-constant (PC) $\beta^{PC} : |E| \to \mathbb{R}$ interpolations in the usual way. For instance, the PC interpolation of α on an edge takes the minimum value of the two vertex consistencies.

Vertex and edge consistencies for an ensemble member. We encode edge consistency for an ensemble member $\mathscr{T}^i$ (with respect to the 1-center $\mathscr{T}$) using glyphs, as in Fig. 16.

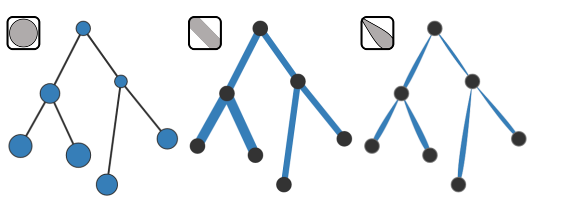

Fig. 16. From left to right: circular, line and ribbon glyphs are used to encode vertex and edge consistencies for an ensemble member.

The width of each line (resp. ribbon) glyph at a location $x \in e$ for $e \in E_i$ scales proportionally with the PC (resp. PL) edge consistency at x, $\beta^{PC}(x)$ (resp. $\beta^{PL}(x)$).

Variational consistencies for the 1-center tree. We encode variations in edge consistencies for the 1-center tree $\mathscr{T}$ using visual primitives inspired by [65, 66], as in Fig. 17, similarly to variations in vertex consistencies.

Statistical consistency for the 1-center tree. Inspired by box plots, we visualize the distribution of edge consistencies at the 1-center $\mathscr{T}$ similarly to the statistical vertex consistencies, see Fig. 18.

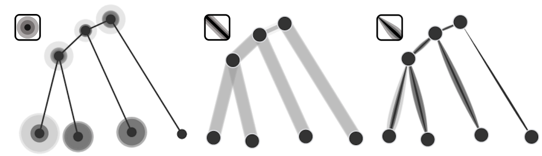

Fig. 17. From left to right: graduated circular glyphs, graduated lines and graduated ribbons are used to encode variational vertex, PC and PL edge consistencies for the 1-center tree.

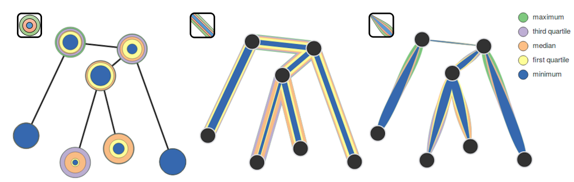

Fig. 18. From left to right: graduated circular glyphs, graduated lines and graduated ribbons are used to encode statistical vertex , PC and PL edge consistencies for the 1-center.

C Interactive Visualization System Design Details

We provide design details of our interactive visualization system. Its user interface is shown in Fig. 8.

Drawing Panel and Ensemble Panel. The drawing panel allows the user to draw individual merge trees using node-link diagrams and assign initial labels to the vertices. Each tree is created with an embedding onto the drawing panel, where each vertex v is equipped with a coordinate (v_x, v_y) and a height function value $f(v)$ according to the panel's underlying grid structure. Using various hot keys (see **?** for a user manual), an embedding can be geometrically reconfigured via insertion, deletion and movement of vertices and edges. At the moment, the system focuses on leaf-labeled merge trees, therefore only the labels on the leaves are used in the computation. Each tree is then added to the ensemble panel, where ensemble members can be selected, deleted and reconfigured/edited.

Control Panel and 1-center computation. With an input ensemble of leaf-labeled merge trees, we compute its 1-center using various options above the control panel. Using the **Enforce label** option, we compute a 1-center of trees in full or partial agreement, whereas the **Ignore label** option enables one to deal with trees in disagreement.

In terms of the parameter setting, we can choose between **Tree distance** d_T and **Euclidean distance** d_E, or a linear combination of the two, using the $\boldsymbol{\lambda}$ parameter for our heuristic labeling strategies. $\boldsymbol{\delta}$ is the locality parameter in consistency measures and **#steps** indicates the number of steps used in the animation.

The control panel also visualizes the relation between the 1-center (denoted as a red node labeled **AMT**) and the ensemble members in a star-shaped summary plot, where the 1-center lies in the center of the star, and the color and length of each link scales proportionally with the interleaving distance between an input tree and the 1-center. By clicking on a link in the summary plot, we enable an animated sequence between an input tree and the 1-center.

Animation. We compute and visualize an animated sequence between an input tree (as the source) and the 1-center (as the target) using two strategies. The **geodesic strategy** follows Theorem 3.2, where intermediate trees follow a geodesic connecting the source and the target pair. The **linear strategy** linearly interpolates between geometric embeddings of the source and the target.

During the animation, the intermediate trees can be displayed with **labels**, which highlight correspondences between leaves. The intermediate trees can also use **colored labels** according to vertex consistencies, for which the animated sequence highlights the changes in vertex consistency as a source tree is moved towards a target tree. In summary, the animation highlights structural variations between an input tree and the 1-center, and the evolution in vertex consistency during such a process.

Consistency Visualization. We visualize various consistency measures in the rightmost panel. First, we visualize vertex and edge consistency for each ensemble member with respect to the 1-center, using circular, linear and ribbon glyphs. Second, we visualize variational consistencies for the 1-center using graduated glyphs with sequential colormap of a single hue. We use graduated circular glyphs for vertices, graduated lines and graduated ribbons for PC and PL edges, respectively. Finally, we highlight statistical consistencies for vertices, PC edges, and PL edges that capture the distribution of consistency measures across ensemble members. In addition, all trees can be visualized with **labels** to indicate leaf correspondences between input and output.

1-Center tree visualization. Since each input tree $\mathscr{T}^i$ is drawing with a geometric embedding $\iota_i : |T_i| \to \mathbb{R}^2$, we compute an embedding ι of the 1-center $\mathscr{T}$ using information from ι_i for all $1 \leq i \leq k$. First we apply the algorithm in Sec. 4 and Appx. A to infer a complete correspondence between the internal vertices of $\mathscr{T}^i$ and $\mathscr{T}$ for each i. Then, for each vertex $v \in V$ with a label l, we compute its embedding $\iota(v) \mapsto (x_v, y_v)$. y_v comes naturally as $y_v = f(v)$. x_v is the 1-center of the x-coordinates of $\{\iota_1(\omega_1(l)), \cdots, \iota_k(\omega_k(l))\}$.

D Local-Global Tradeoff

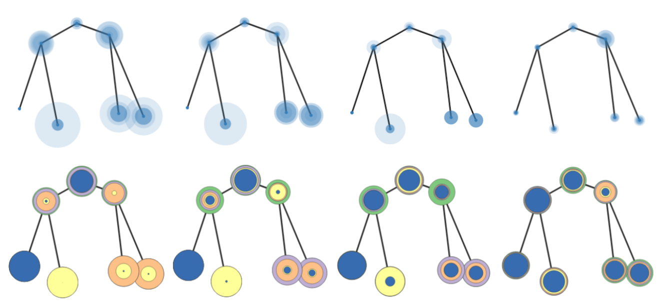

Fig. 19. The changes in variation and distribution of consistency as we increase the smoothing parameter $\delta \in \{0.05, 0.07, 0.10, 0.15\}$.

We could also use our system to study the tradeoff between local and global consistency measures in capturing structural similarities between ensemble members and their structural average. As we increase the locality parameter δ in the Gaussian-weighted cosine similarity measure, we could observe the change in structural variations on vertices; see Fig. 19 for an example. Using the same input ensemble as Fig. 10, we see that as δ increases, the variational vertex consistency decreases (top) as we pay more attention to global structural similarities among the ensemble members. Meanwhile, the distribution of consistency measures becomes increasingly concentrated (bottom). See the supplementary video for a demo.

E Intrinsic-Extrinsic Tradeoff

The input merge trees can have natural (function-induced) intrinsic metrics associated to them. Sometimes, these trees are geometric (i.e., embedded in Euclidean spaces), and thus also have natural extrinsic (ambient) metrics defined on them, e.g., in the case of neuron trees modeling neuron cells. Our tool supports a combination of both metrics. See the supplementary video for a demo.

Fig. 20 shows that using purely Euclidean distance vs. purely tree distance gives different new labels for *Tree 2* and *Tree 3* and thus affects the resulting 1-center as well as its statistical consistency. The results using a linear combination of both Euclidean and tree distance for $\lambda = 0.5$ are very similar to those based on pure tree distance, with minor differences visible for the 1-center trees and their statistical consistency plots.

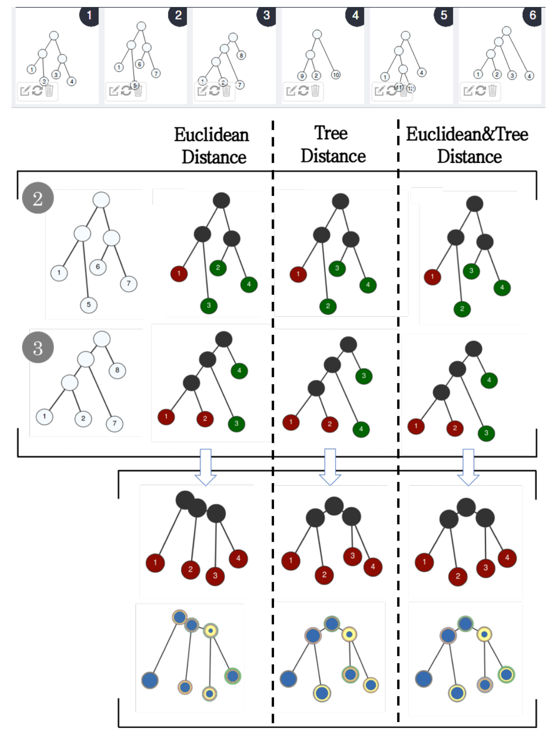

Fig. 20. Investigating the tradeoff between intrinsic and extrinsic metrics in labeling strategies. 1st row: input ensemble. 2nd row, from left to right: *Tree 2*; *Tree 2* with updated labels using Euclidean distance, tree distance, and a combination of both using $\lambda = 0.5$. 3rd row, similar to 2nd row, for *Tree 3*. 4th and 5th row: 1-center tree and its statistical consistency plot; from left to right: using Euclidean distance, tree distance, and a combination of both using $\lambda = 0.5$.

F Applications in Neuron Morphology

Neuron cells have tree morphology, and a rapidly increasing amount of neuroanatomical data are now publicly available (e.g., `NeuroMorpho.org` and `flycircuit.org`). Our proposed consistency measure can be used to understand structural variations among an ensemble of *neuron cell induced merge trees* with respect to their 1-center.

As a case study, we use our proposed methodologies to help study differences/variations among different reconstructions of the same neuron cell. In particular, in the past 15 years, a large number of algorithms have been developed to reconstruct a tree structure for neuron cells (referred to as *neuron trees*) from 2D or 3D images (e.g., the dozens of methods incorporated in the visualization software Vaa3D [53]).

Input trees. We use one of the olfactory projection fibers datasets, referred to as **OP_6**, from the DIADEM challenge [9]. We create a set of neuron trees reconstructed for **OP_6** using different reconstruction methods. Each neuron tree comes with a 3D embedding in the form of a 3D image. Given an image of a neuron tree, we extract its corresponding (unlabeled) merge tree representation as a pair $\mathscr{T}^i = (T_i, f_i)$. First, the vertex set V_i is obtained by extracting the leaves and branching points from the 3D image; each vertex is equipped with a geometric location via the 3D embedding of the neuron tree. The function $f_i : V_i \to \mathbb{R}$ is the geodesic distance of a vertex $x \in V_i$ to a base point $o \in V_i$ (chosen as the physical root of the neuron cell). This extracted merge tree is referred to as the *neuron cell induced merge tree*.

Analysis. Fig. 21 shows three neuron cell induced merge trees for dataset **OP_6** via reconstruction algorithms APP2 [79] (h), SmartTracing [18] (f) and NeuroGPS-Tree [63] (c), as well as their 1-center tree (a). The vertices for each input tree are colored by their consistencies. The input trees are reasonably similar; thus, to see the structural variation, we use a relatively small $\delta = 0.05$ value when computing consistency.

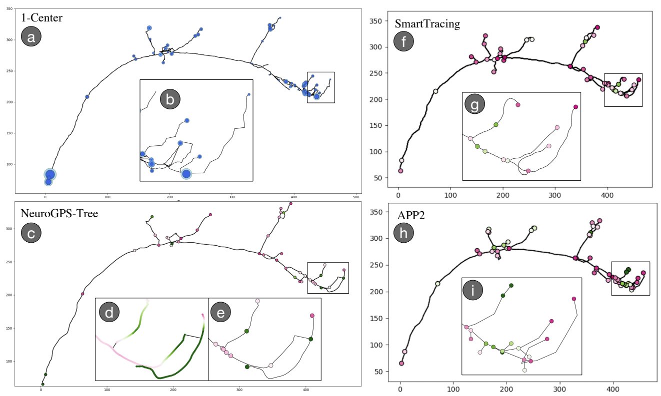

Fig. 21. 2D projections of vertex consistency visualization of an ensemble of neuron trees reconstructed from **OP_6**.

Meanwhile, each vertex of the 1-center tree is visualized with a variational consistency using graduated circular glyphs; see Fig. 21 (c,f,h). It is therefore easy to spot which vertices in the 1-center tree have high variance. Upon close inspection, each high variance vertex in the 1-center tree indeed corresponds to locally different reconstructions in input trees (e,g,i). Similarly, from each individual input tree, it is easy to see how each vertex deviates from the 1-center tree locally. For example, vertices from the region in the black rectangle of NeuroGPS-Tree have low consistency (white to green colors) with respect to the 1-center tree (e). Indeed, as the inset zoomed-in view shows (e), NeuroGPS-Tree produces a different local tree configuration (one with a severely bended branch) as the output of APP2 (i) and SmartTracing (g). We also remark that as we increase the δ value, the consistency measures similarity at a more global level and thus the structural variation becomes less visible. Finally, we can also show consistency along edges of input trees, which could help to make low-consistency regions more prominent to spot than the vertex consistency visualization. See Fig. 21(d) for an example. A detailed methodological development for neuron trees is left for future work.