GIVF: GEOMETRY-GUIDED WHOLE-TREE RECONSTRUCTION FROM CROP-WISE PREDICTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Whole-tree instance segmentation requires both assigning points to coherent objects and determining which reconstructed instances are valid. This becomes particularly challenging when large forest scenes are processed through overlapping crops, since reconciling local instance masks does not necessarily correct point membership errors within those predictions. We therefore formulate scene-level whole-tree decoding as two related problems: reconstructing point membership and assessing the validity of the resulting instances. To address them, we propose GIVF, a modular framework that integrates geometry-guided instance assembly and learned validity assessment into a single decoding pipeline. GIVF aggregates crop-wise offsets into spatial anchors and combines them with fine-scale geometric tree-membership and trunk-support evidence. Seeded geodesic propagation constructs disjoint whole-tree candidates on a scene-level graph using spatial connectivity and anchor consistency. A lightweight validity model subsequently determines whether each assembled candidate constitutes a valid tree instance, using its geometric structure, vertical profile, trunk support, and native prediction confidence. On four test plots of the Cherlet TLS benchmark, GIVF achieves a macro detection F1 of 0.7254 and a matched-tree mF1 of 0.8865.

1 INTRODUCTION

Accurate individual-tree segmentation is a key step in converting forest LiDAR observations into tree-level measurements. Forest inventories depend on information about stem dimensions, tree height, and crown structure, while geometric reconstructions of individual trees support non-destructive biomass estimation (Liang et al., 2016; Calders et al., 2015; Xiang et al., 2024). Obtaining these measurements from plot-level point clouds requires assigning points to individual trees. This association is particularly difficult in structurally complex forests, where neighboring crowns overlap, branches from different trees come into close contact, and occlusion creates gaps within the observed structure of a tree (Tao et al., 2015; Henrich et al., 2024). Spatial proximity is therefore ambiguous: nearby points may belong to different trees, while spatially separated points may belong to the same tree. Reliable whole-tree segmentation must consequently recover coherent stem–crown associations while separating neighboring structures and preventing erroneous assignments.

Recent 3D instance segmentation methods construct objects through pointwise grouping or query-based mask prediction (Jiang et al., 2020; Schult et al., 2023). In large forest scenes, however, inference is commonly performed over overlapping spatial crops, introducing an additional scene-level association problem. Crop-level predictions must be reconciled into a consistent plot-level segmentation. ForestFormer3D, for example, uses global mask ranking, overlap suppression, and crop-boundary filtering to reconcile predictions from overlapping crops (Xiang et al., 2025). Such operations resolve competition among candidate masks, but candidate selection or suppression is not equivalent to recovering the most appropriate point membership for an entire tree. Different crops may contain complementary fragments of the same tree, while an individual candidate may mix observations from neighboring trees. Retaining or merging existing masks can therefore preserve membership errors that originate within the crop-level predictions themselves. Moreover, imperfect point evidence or seed initialization can still produce weakly supported or spurious candidates after scene-level reconstruction, making candidate acceptance a separate problem from point assignment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

We treat whole-tree decoding as scene-level membership reconstruction followed by instance validity assessment. Aggregated point evidence determines which observations belong together, and the resulting candidates are evaluated for retention. This formulation builds on prediction-guided grouping (Henrich et al., 2024; Du et al., 2026) and assesses validity on the reconstructed point sets.

We present GIVF, a modular framework that integrates geometry-guided instance assembly and learned validity assessment within a single decoding pipeline. GIVF retains crop-wise neural prediction but does not treat native crop masks as final tree instances. Learned horizontal offsets are aggregated across overlapping crops to provide spatial anchors for cross-crop association, while lightweight geometric classifiers provide complementary tree-membership and trunk-support evidence from fine-scale local descriptors. Tree-supported seeds initialize multi-source geodesic propagation on a scene-level voxel graph, where assignment costs combine spatial connectivity with consistency between offset-derived anchors. The resulting disjoint point sets are subsequently evaluated by an instance-level validity model using geometric structure, vertical profiles, trunk support, and prediction confidence. In this way, GIVF first reconstructs candidate whole-tree masks from accumulated scene-level evidence and then determines which reconstructed instances are sufficiently well supported to retain.

We evaluate GIVF on the four test plots of the TLS benchmark introduced by Cherlet et al. (2026), measuring both tree detection and matched-tree delineation. Using the same neural checkpoint, we evaluate the pipeline at three stages native prediction, geometry-guided whole-tree assembly, and validity filtering to separate the effects of membership reconstruction and candidate assessment. We further study supervised ground-domain adaptation in which the neural predictor remains frozen while only the lightweight geometric evidence models are fitted to target-domain training data. Our contributions are:

- We introduce an explicit geometric evidence pathway that provides tree-membership and near-ground trunk-support estimates for whole-tree decoding, complementing crop-wise learned spatial association cues without relying on native crop-level instance identities for final membership.
- We formulate whole-tree decoding as scene-level point-membership reconstruction followed by instance-level validity assessment. Spatial connectivity, geometric propagation costs, and crop-aggregated neural anchors are combined to assemble tree candidates, which are then evaluated on the structure of the reconstructed masks.
- We demonstrate the framework across four forest benchmark plots through detection, matched-tree delineation, and stage-wise evaluation, and show that its explicit geometric evidence pathway can be adapted to ground-domain point clouds while keeping the neural predictor frozen.

2 Related Work

2.1 3D Point Cloud Instance Segmentation

Instance segmentation extends semantic point labeling by assigning distinct identities to objects that belong to the same category. Existing methods can be broadly divided into grouping-based approaches, which form instances from point-level predictions, and query-based approaches, which directly decode instance masks.

Grouping-based methods typically rely on learned semantic and spatial cues to assemble points into object instances. PointGroup clusters points in both the original and offset-shifted coordinate spaces and then scores the resulting proposals (Jiang et al., 2020). HAIS progressively aggregates point sets and refines candidate masks (Chen et al., 2021), whereas SSTNet constructs instances through a learned hierarchy of semantic superpoints (Liang et al., 2021). SoftGroup replaces hard semantic decisions with soft assignments, reducing the effect of semantic errors on the grouping stage, and further refines proposals to suppress false positives (Vu et al., 2022). Despite differences in grouping granularity and proposal refinement, these methods all require an explicit association step after point-level prediction.

Query-based methods take a different approach by representing candidate instances with learned object queries. Mask3D predicts instance masks through multi-scale interactions between queries and

point features (Schult et al., 2023). SPFormer performs query-based decoding on superpoint features (Sun et al., 2023). QueryFormer further improves query initialization and filters noisy background queries (Lu et al., 2023). OneFormer3D uses a shared Transformer decoder for semantic, instance, and panoptic segmentation (Kolodiazhnyi et al., 2024). These methods avoid much of the hand-designed grouping used in earlier pipelines. However, when large scenes are processed in separate spatial blocks, predictions from different blocks still need to be reconciled.

This issue has also motivated work on block-wise inference. Denis et al. (2023), for example, improve cross-block instance merging by allowing labels in previously processed blocks to be revised through label propagation. Such approaches aim to maintain consistent object identities across independently processed regions after local instance prediction. GIVF is related to this line of work in addressing scene-level association, but differs in reconstructing point membership from accumulated neural and geometric evidence rather than only reconciling existing block labels or masks.

2.2 Individual Tree Segmentation in Forest Point Clouds

Geometry-based tree construction. Forest-specific segmentation methods often exploit the geometric organization of stems, branches, and crowns. Comparative shortest-path methods associate crown observations with detected trunks (Tao et al., 2015), while treeseg combines clustering, local shape analysis, and region growing (Burt et al., 2019). Treeiso uses hierarchical graph partitioning to separate individual trees (Xi & Hopkinson, 2022). These methods are effective when tree structure can be recovered reliably, but their performance may degrade when stems, branches, or crown boundaries are ambiguous.

Learned semantic and spatial predictions provide additional cues for forest segmentation. ForAINet jointly predicts semantic and instance information from forest point clouds (Xiang et al., 2024). SegmentAnyTree adapts a PointGroup-style instance segmentation framework to different laser-scanning settings (Wielgosz et al., 2024). TreeLearn aggregates semantic and offset predictions from overlapping tiles before performing scene-level grouping, and uses geometric verticality to support tree identification (Henrich et al., 2024). TLS2trees combines semantic classes with structural shortest-path reasoning for woody components and foliage (Wilkes et al., 2023). More recently, SATree integrates structural semantics, stem-localization heatmaps, and offset predictions with graph-based shortest-path grouping (Du et al., 2026).

ForestFormer3D instead adopts a query-based formulation. It uses instance- and semantic-aware query selection together with one-to-many mask supervision (Xiang et al., 2025). At plot level, predictions from overlapping cylindrical crops are reconciled to form larger instances. Recent benchmark results also show that tree-segmentation performance can vary substantially across forest plots and evaluation settings (Cherlet et al., 2026).

3 Method

3.1 Overview and Point Evidence

Given a forest point cloud $\mathcal{X} = \{\mathbf{p}_i\}_{i=1}^{N}$, with $\mathbf{p}_i \in \mathbb{R}^3$, GIVF recovers disjoint point sets corresponding to individual trees. The framework combines crop-wise point-evidence prediction with scene-level whole-tree decoding. Learned spatial anchors and geometric tree/trunk estimates support instance construction, while a subsequent validity model determines which assembled candidates are retained. Figure 1 summarizes the pipeline.

Local geometric representation. We compute covariance-based descriptors at three spatial scales, following geometric point-cloud representations (Demantké et al., 2011; Weinmann et al., 2015; Landrieu & Simonovsky, 2018). Each scale contributes linearity, planarity, scattering, vertical alignment, and logarithmic local support count, producing $\mathbf{g}_i \in \mathbb{R}^{15}$. These descriptors are combined with crop-centered coordinates $\widetilde{\mathbf{p}}_i^{(b)}$ to form the neural input:

$$\mathbf{x}_i^{(b)} = [\mathbf{g}_i; \widetilde{\mathbf{p}}_i^{(b)}] \in \mathbb{R}^{18}. \tag{1}$$

The local predictor uses the Oneformer3D implementation (Kolodiazhnyi et al., 2024). A sparse U-Net extracts features from overlapping cylindrical crops, and separate prediction branches provide horizontal offsets and native query-mask confidence.

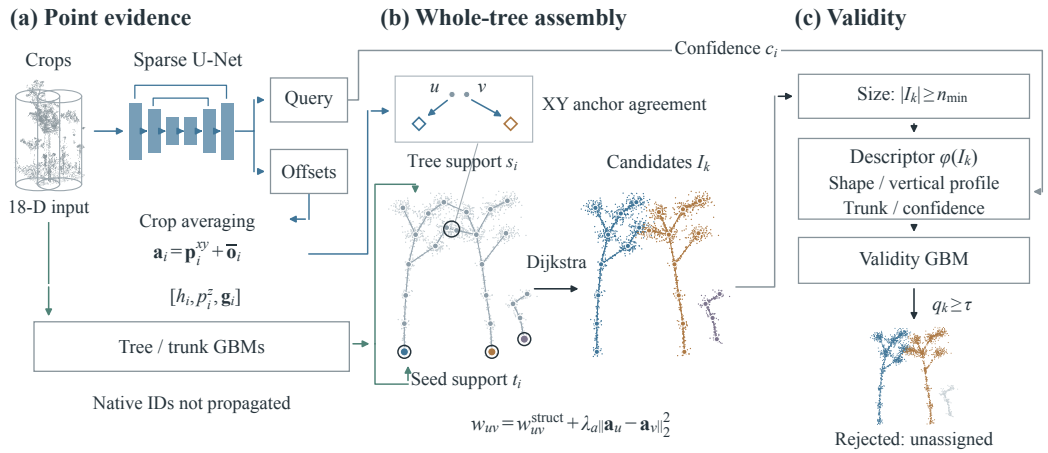

Figure 1: Overview of GIVF. (a) Crop-wise neural prediction supplies horizontal offsets and native confidence, while geometric GBMs estimate tree membership and trunk support. (b) Tree scores define graph support, trunk evidence supports seed construction, and predicted anchors modulate geodesic assignment. Rings indicate seeds and diamonds represent predicted XY anchors. (c) Size-eligible candidates are assessed using their geometry, vertical structure, trunk support, and native confidence. Rejected candidates become unassigned. The input thumbnail shows real points; the graph and candidate examples illustrate the decoding mechanism schematically.

Crop-aggregated spatial predictions. The offset head predicts horizontal displacements toward a tree-specific horizontal reference anchor. The reference is derived from its lower structure using the mean position of robust line-fitting inliers, with a lower-band mean as fallback. It is computed before crop sampling and transformed consistently with crop coordinates, providing a common reference across overlapping observations of the same tree. Foreground voxels receive L1 offset supervision.

Let $\mathcal{B}_i$ denote the crops contributing predictions for scene point i. After mapping predictions into the common scene frame, offsets are averaged to obtain a horizontal anchor:

$$\bar{\mathbf{o}}_i = \frac{1}{|\mathcal{B}_i|} \sum_{b \in \mathcal{B}_i} \mathbf{o}_i^{(b)}, \qquad \mathbf{a}_i = \mathbf{p}_i^{xy} + \bar{\mathbf{o}}_i. \tag{2}$$

The anchor provides a continuous spatial association cue across crops. Native query confidence is also mapped to scene points and retained as c_i for instance assessment. Final graph memberships are reconstructed from scene-level evidence.

3.2 Geometry-Guided Whole-Tree Decoding

Scene graph and structural seeds. Geometric GBMs provide tree-membership scores s_i and trunk-support scores t_i from local height, elevation, and multi-scale descriptors (Appendix A.3). Points satisfying $s_i \geq \theta_{\text{sem}}$ are voxelized into a spatial graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Each node stores its mean position, point count, mean predicted anchor, and maximum trunk score. Edges are formed by symmetrized spatial k-nearest-neighbor connections.

Near-ground nodes with sufficient trunk support are clustered into primary seed sets. Vertically persistent structures provide supplementary seeds away from existing trunk seeds. Sufficiently large connected components containing no seed receive a fallback seed at their lowest node. Let $\mathcal{S}_k \subseteq \mathcal{V}$ denote the source nodes associated with seed k.

Anchor-aware geodesic assignment. For neighboring nodes u and v, the propagation cost is

$$w_{uv} = d_{uv}^2 (1 + \beta_{uv}) \left[1 + \gamma \left(\frac{1}{n_u + 1} + \frac{1}{n_v + 1} \right) \right] + \lambda_a \|\mathbf{a}_u - \mathbf{a}_v\|_2^2, \tag{3}$$

where d_{uv} is the distance between node positions, n_u and n_v are voxel occupancies, and $\mathbf{a}_u$ and $\mathbf{a}_v$ are aggregated predicted anchors. The transition penalty β_{uv} modulates propagation according to edge orientation, height, and local anchor dispersion. Its complete definition is provided in Appendix A.4.

Spatial distance favors short connections, and the occupancy term increases the cost of transitions supported by fewer points. Anchor consistency penalizes transitions between neighboring nodes whose predictions indicate different spatial destinations. The cost therefore combines observed geometric connectivity with learned association cues.

For seed k, the accumulated path distance and voxel assignment are

$$\begin{aligned} D_k(v) &= \min_{\substack{s \in \mathcal{S}_k \\ \pi: s \rightsquigarrow v}} \sum_{(u,w) \in \pi} w_{uw}, \\ \ell(v) &= \arg\min_k D_k(v). \end{aligned} \tag{4}$$

Dijkstra shortest-path computation evaluates distances from the seed sets, and each reachable voxel is assigned to the set with minimum accumulated cost. Voxels without a finite path to any seed remain unassigned. Mapping voxel labels back to points produces disjoint candidate instances $\{I_k\}$. Candidates containing fewer than $n_{\min}$ points are removed before instance descriptors are computed.

For each size-eligible candidate, an instance descriptor $\boldsymbol{\phi}(I_k)$ summarizes point support, vertical extent, radial and shape statistics, vertical profiles, trunk support, and mean native confidence. A gradient-boosted validity model predicts

$$q_k = f_{\text{valid}}\big(\boldsymbol{\phi}(I_k)\big), \qquad \widehat{\mathcal{I}} = \{I_k : |I_k| \geq n_{\min},\ q_k \geq \tau\}. \tag{5}$$

Descriptors characterize the assembled point sets. Validity filtering preserves accepted memberships and leaves points in rejected candidates unassigned. Graph propagation therefore determines candidate membership, while the validity model determines candidate retention.

3.3 Learning and Fitting

The neural predictor retains its prediction objectives and auxiliary superpoint-contrastive supervision. Backbone features are mapped from neural voxels to points and pooled within geometric superpoints; whole-tree assignment uses a separate scene-level voxel graph. Loss definitions and sampling details are provided in Appendix A.2.

For each source benchmark plot, the geometric classifiers are fitted on its training subdivision. The validity model is fitted on candidates assembled from its validation subdivision. Candidate targets are matching-based IoU proxies computed on the fitting point support.

The operating threshold is selected from out-of-fold validation predictions subject to retaining a specified fraction of proxy-positive validation candidates. The retention setting and its development history are documented in Appendix A.5. Descriptor scales, graph parameters, classifier settings, and minimum-support thresholds are provided in the appendix.

4 Experiments

We evaluate detection and matched-tree delineation, supervised adaptation of geometric point evidence with a frozen neural predictor, and the incremental effects of the three stages of GIVF.

4.1 Experimental Setup

Datasets and supervision. Our primary evaluation follows the TLS benchmark of Cherlet et al. (2026), using the Litchfield, Wytham, Ofental, and Robson Creek test subdivisions. Each plot contains spatially separated training, validation, and test regions. The geometric tree and trunk classifiers are fitted on the corresponding training subdivision, while the instance-validity model is fitted using validation predictions. The development history of the validity retention setting is described in Appendix A.

Table 1: Tree detection on the four Cherlet test plots. Baseline values are reproduced from the standardized evaluation of Cherlet et al. (2026); TreeLearn and ForAINet refer to Henrich et al. (2024) and Xiang et al. (2024), respectively. Baseline columns report F1, while precision and recall are additionally shown for GIVF. Macro values are unweighted means over the four plots.

	1-plot FT		All-plot FT		GIVF		
Plot	TreeLearn	ForAINet	TreeLearn	ForAINet	Precision	Recall	F1
Litchfield	0.922	0.883	0.930	0.905	0.9839	0.9531	**0.968**
Wytham	0.526	0.334	0.540	0.363	0.8188	0.6243	**0.708**
Ofental	0.678	0.601	0.497	0.441	0.7778	0.7865	**0.782**
Robson Creek	0.204	0.288	0.178	0.247	0.4604	0.4267	**0.443**

We additionally evaluate supervised adaptation to four ground-based point-cloud scenes: CULS, RMIT, TUWIEN, and NIBIO_MLS. In this experiment, the neural predictor is kept frozen and only the lightweight geometric tree and trunk classifiers are fitted using ground-domain training point clouds.

Evaluation metrics. For the Cherlet benchmark, we use the benchmark evaluation implementation and report plot-level precision, recall, and F1. We additionally report matched-tree mF1, mRecall, and mPrecision. The ground-domain experiment uses raw-point membership IoU with one-to-one Hungarian matching at IoU > 0.5, and macro F1 is averaged equally over the four evaluated scenes. Because the source and adaptation evaluations use different protocols, their absolute scores are not directly compared.

Implementation. The local predictor uses 18-channel input consisting of XYZ coordinates and multi-scale geometric descriptors, and processes overlapping cylindrical crops of radius $16\,\mathrm{m}$. The neural model uses a $0.20\,\mathrm{m}$ voxel representation, while whole-tree decoding operates on a finer $0.10\,\mathrm{m}$ scene graph.

The graph configuration is shared across the four source plots. Geometric tree and trunk classifiers are fitted separately on each plot's training subdivision, and validity models are fitted on candidates assembled from the corresponding validation subdivision. Additional architecture, training, graph parameters, and validity retention details are provided in Appendix A.

4.2 Comparison with Existing Methods

We compare GIVF with the learning-based configurations reported by Cherlet et al. (2026). For completeness, we report both single-plot and all-plot fine-tuning configurations of TreeLearn and ForAINet, rather than selecting a different baseline configuration for each test plot. Table 1 reports their per-plot detection F1 together with the precision, recall, and F1 of GIVF.

GIVF achieves higher detection F1 than each of the listed learning-based baseline configurations on all four test plots. Its macro F1 is 0.7254, compared with 0.5825 for the strongest baseline configuration in terms of macro F1, TreeLearn with single-plot fine-tuning, corresponding to a difference of 14.3 percentage points. The margin is smallest on Litchfield, where the baseline methods already achieve relatively high detection performance, and is substantially larger on Wytham, Ofental, and Robson Creek.

The precision–recall breakdown shows that the remaining detection errors differ across plots. Litchfield has both high precision (0.9839) and recall (0.9531). Wytham is more recall-limited: GIVF matches 113 trees, with 25 false positives and 68 false negatives, yielding precision 0.8188 and recall 0.6243. Ofental exhibits a more balanced precision–recall profile. Robson Creek remains the most challenging plot: GIVF matches 64 of 150 evaluated ground-truth trees and produces 75 false positives, corresponding to 86 false negatives, precision 0.4604, and recall 0.4267. Thus, although the detection improvement is observed across all four plots, substantial errors remain in the most structurally difficult scenes.

Table 2: Matched-tree delineation on the Cherlet benchmark. Baseline values are reproduced from Cherlet et al. (2026); TreeLearn and ForAINet denote the methods of Henrich et al. (2024) and Xiang et al. (2024), respectively. Metrics are computed only over successfully matched trees and therefore do not penalize missed instances.

Method	mF1 ↑	mRecall ↑	mPrecision ↑
TreeLearn (1-plot FT)	0.8748	0.9125	0.8540
ForAINet (1-plot FT)	0.8638	0.9100	0.8403
TreeLearn (all-plot FT)	0.8610	0.9053	0.8383
ForAINet (all-plot FT)	0.8533	0.8990	0.8320
GIVF	**0.8865**	**0.9190**	**0.8708**

Table 3: Supervised adaptation of geometric evidence with a frozen neural predictor. Both configurations use the same graph-assembly procedure. Target classifiers are fitted after test-support exclusion. Values are raw-point detection F1; Macro is the unweighted mean over the four scenes.

Point evidence	CULS	RMIT	TUWIEN	NIBIO_MLS
Source neural cues	.4286	.1892	.0889	.8293
Target-fitted geometric fusion	**.9756**	**.4565**	**.6364**	**.9565**

GIVF obtains matched-tree mF1 0.8865, mRecall 0.9190, and mPrecision 0.8708. These gains are smaller than the detection improvement. Because matched subsets differ across methods, the scores do not measure refinement on a fixed tree set.

4.3 Qualitative Results

Figure 2 examines three local examples from Litchfield, Ofental, and Wytham. Small overview panels locate the enlarged regions. Within each row, ground truth, native predictions, and GIVF use identical point samples, viewing parameters, and spatial bounds. The same annotated target-tree points are emphasized across columns, while surrounding points are faded. Full-window comparisons for all four plots are provided in Appendix A.

In example A, many target-tree points are unassigned in the native output and are recovered by GIVF. In example B, native predictions distribute the target points across multiple instance labels, whereas GIVF assigns a larger fraction to one candidate. Example C illustrates a limitation of inspecting target points alone: similar assignments on the highlighted observations do not establish the purity of the complete predicted instance. The full-window comparisons retain neighboring-tree context for examining such errors.

4.4 Supervised Adaptation of Geometric Evidence

We examine whether the explicit geometric evidence can adapt segmentation to additional TLS and MLS scenes while keeping the neural predictor frozen. Two shared tree-membership and trunk-support classifiers are fitted on seven target-domain training scenes and evaluated on CULS, RMIT, TUWIEN, and NIBIO_MLS. Training observations near the test-point support and affected training instances are excluded before fitting. The exclusion rule, retained training support, and fitting procedure are detailed in Appendix A.

Semantic fusion retains native tree predictions and adds points whose geometric tree probability is at least 0.30. Trunk support is supplied by the target-fitted geometric classifier within the near-ground height band. Both conditions are evaluated at the size-filtered graph-assembly stage, using the same neural checkpoint, predicted offsets, graph parameters, and minimum candidate size of 1000 points. Instance-validity filtering is not applied in this experiment.

The fitted geometric fusion increases macro F1 from 0.3840 to 0.7563, with improvements on all four scenes. The largest absolute gains occur on CULS and TUWIEN, where tree-point recall within

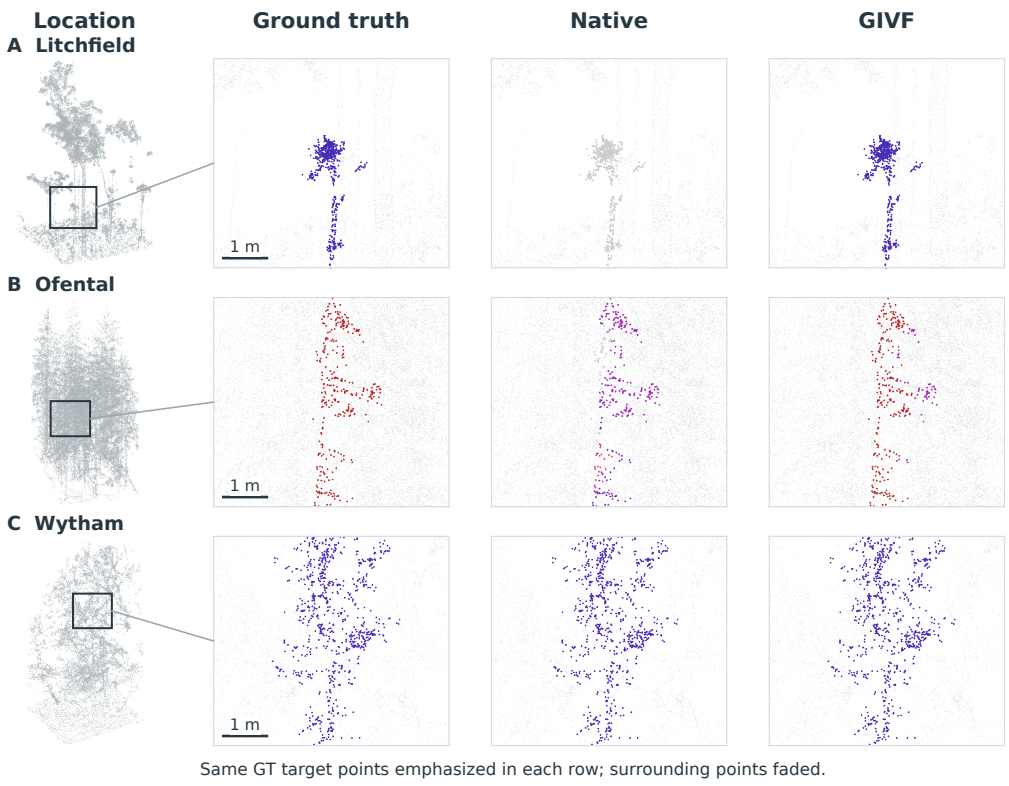

Figure 2: Local target-tree assignments in three forest scenes. The left column locates each enlarged region. The same GT target-tree points are emphasized in all three comparison columns, retaining their original instance colors or unassigned status. Surrounding points are uniformly faded. This diagnostic view shows how target-tree observations are assigned; the full-window comparisons in the appendix provide the surrounding instance context.

the 2 m local-height band increases from 0.0006 to 0.6751 and from 0.0124 to 0.4364, respectively. These changes recover tree-labeled support in the region used for seed construction.

The results demonstrate that adapting the geometric evidence interface can substantially improve segmentation while preserving the neural predictor. RMIT remains challenging, with F1 0.4565, indicating that improved point evidence alone does not resolve all detection and assignment errors.

4.5 Ablation Studies

Stage-wise evaluation of GIVF. We evaluate three inference configurations using the same neural checkpoint and benchmark evaluation protocol.

Native denotes the scene-level predictions obtained by aggregating crop-wise mask predictions through the original cross-crop merging and filtering procedure.

Geometric assembly combines geometric tree-membership and trunk-support estimates with crop-aggregated neural offsets. Scene-level graph assignment constructs candidate instances, followed by minimum-size filtering.

Validity (GIVF) applies the learned validity model to the same assembled candidates. Accepted candidates retain their point memberships, while rejected candidates become unassigned.

The transition from Native to Geometric assembly measures the combined effect of geometric point evidence and scene-level reconstruction. The subsequent transition measures the effect of candidate validity assessment.

Table 4: Three-stage evaluation on four Cherlet test plots. ΔF1 is measured relative to the preceding stage.

Stage	Macro P	Macro R	Macro F1	ΔF1	mF1
Native	.2575	.4750	.3170	–	.8778
+ Geometric assembly	.6007	**.7063**	.6413	+.3243	.8856
+ Validity (GIVF)	**.7602**	.6977	**.7254**	+.0841	**.8865**

Geometry-guided assembly increases macro F1 from 0.3170 to 0.6413, with improvements in both precision and recall. Instance validity assessment further increases F1 to 0.7254, mainly through a precision increase from 0.6007 to 0.7602, while recall decreases from 0.7063 to 0.6977. Matched-tree mF1 changes from 0.8778 to 0.8865. These results are consistent with the main gains arising from forming and retaining a better set of tree instances. Because matched subsets differ between stages, mF1 does not directly measure refinement on a fixed set of trees.

5 Discussion

GIVF shows that scene-level post-processing can substantially improve whole-tree segmentation from crop-wise neural predictions. Geometric tree and trunk evidence supports instance construction, while anchor-aware graph assignment reconstructs memberships across crops. Assessing validity on the assembled candidates further improves precision. These results highlight the importance of adapting the decoding stage to forest structure alongside improving local prediction.

The lightweight geometric and validity classifiers are fitted separately from the neural predictor, and their reliability depends on the available training data. Errors in tree-point selection or seed construction can propagate into the assembled instances, while validity filtering can reject candidates but cannot repair incorrect memberships within them. The ground-domain results further rely on target-supervised fitting, leaving reliable zero-shot transfer an open challenge.

6 Conclusion

We presented GIVF, a framework that combines explicit geometric tree and trunk evidence with scene-level whole-tree reconstruction. Crop-aggregated neural offsets provide spatial anchors, while geometric classifiers determine tree-point support and contribute to seed construction. Anchor-aware geodesic assignment reconstructs candidate instances, and a subsequent validity model determines which candidates are retained. Evaluation on four forest plots shows improved detection relative to the native output of the same checkpoint, together with strong matched-tree delineation scores. A supervised ground-domain experiment further demonstrates improved segmentation through adaptation of the geometric classifiers while keeping the neural predictor frozen. The three-stage evaluation supports the combined benefit of geometric evidence and scene-level assembly, followed by an additional precision gain from candidate validity assessment.

AI use statement

Generative AI tools were used solely to assist with language polishing and improving the clarity and readability of the manuscript. The author reviewed and revised all AI-assisted edits and takes full responsibility for the final content of the manuscript.

References

Andrew Burt, Mathias Disney, and Kim Calders. Extracting individual trees from lidar point clouds using treeseg. *Methods in Ecology and Evolution*, 10(3):438–445, 2019. doi: 10.1111/2041-210X.13121.

Kim Calders, Glenn Newnham, Andrew Burt, Simon Murphy, Pasi Raumonen, Martin Herold, Darius Culvenor, Valerio Avitabile, Mathias Disney, John Armston, and Mikko Kaasalainen. Nonde-

structive estimates of above-ground biomass using terrestrial laser scanning. *Methods in Ecology and Evolution*, 6(2):198–208, 2015. doi: 10.1111/2041-210X.12301.

Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15467–15476, 2021.

Wout Cherlet, Karun Dayal, Shilin Chen, Zane Cooper, Mathias Disney, Andreas Hanzl, Shaun Levick, Joanne Nightingale, Niall Origo, Cornelius Senf, Luna Soenens, Louise Terryn, Wouter A. J. Van den Broeck, and Kim Calders. Benchmarking tree instance segmentation of terrestrial laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 231:230–247, 2026. doi: 10.1016/j.isprsjprs.2025.10.033.

J. Demantké, C. Mallet, N. David, and B. Vallet. Dimensionality based scale selection in 3d lidar point clouds. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XXXVIII-5/W12, pp. 97–102, 2011. doi: 10.5194/isprsarchives-XXXVIII-5-W12-97-2011.

Leon Denis, Remco Donovan Royen, and Adrian Munteanu. Improved block merging for 3d point cloud instance segmentation. In *2023 24th International Conference on Digital Signal Processing (DSP)*, pp. 1–5. IEEE, 2023. doi: 10.1109/DSP58604.2023.10167976.

Shenglan Du, Jantien Stoter, Julian F. P. Kooij, and Liangliang Nan. SATree: Structure-aware tree instance segmentation from 3d LiDAR point clouds. *Urban Forestry & Urban Greening*, 120:129414, 2026. doi: 10.1016/j.ufug.2026.129414.

Jonathan Henrich, Jan van Delden, Dominik Seidel, Thomas Kneib, and Alexander Ecker. TreeLearn: A deep learning method for segmenting individual trees from ground-based LiDAR forest point clouds. *Ecological Informatics*, 84:102888, 2024. doi: 10.1016/j.ecoinf.2024.102888.

Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4867–4876, 2020. doi: 10.1109/CVPR42600.2020.00492.

Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. OneFormer3D: One transformer for unified point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20943–20953, 2024.

Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4558–4567, 2018. doi: 10.1109/CVPR.2018.00479.

Xinlian Liang, Ville Kankare, Juha Hyyppä, Yunsheng Wang, Antero Kukko, Henrik Haggrén, Xiaowei Yu, Harri Kaartinen, Anttoni Jaakkola, Fengying Guan, Markus Holopainen, and Mikko Vastaranta. Terrestrial laser scanning in forest inventories. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:63–77, 2016. doi: 10.1016/j.isprsjprs.2016.01.006.

Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2783–2792, 2021.

Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18516–18526, 2023. doi: 10.1109/ICCV51070.2023.01697.

Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8216–8223, 2023. doi: 10.1109/ICRA48891.2023.10160590.

Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 2393–2401, 2023. doi: 10.1609/aaai.v37i2.25335.

Shengli Tao, Fangfang Wu, Qinghua Guo, Yongcai Wang, Wenkai Li, Baolin Xue, Xueyang Hu, Peng Li, Di Tian, Chao Li, Hui Yao, Yumei Li, Guangcai Xu, and Jingyun Fang. Segmenting tree crowns from terrestrial and mobile LiDAR data by exploring ecological theories. *ISPRS Journal of Photogrammetry and Remote Sensing*, 110:66–76, 2015. doi: 10.1016/j.isprsjprs.2015.10.007.

Thang Vu, Kookhoi Kim, Tung M. Luu, Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2708–2717, 2022. doi: 10.1109/CVPR52688.2022.00273.

Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304, 2015. doi: 10.1016/j.isprsjprs.2015.01.016.

Maciej Wielgosz, Stefano Puliti, Binbin Xiang, Konrad Schindler, and Rasmus Astrup. SegmentAnyTree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data. *Remote Sensing of Environment*, 313:114367, 2024. doi: 10.1016/j.rse.2024.114367.

Phil Wilkes, Mathias Disney, John Armston, Harm Bartholomeus, Lisa Bentley, Benjamin Brede, Andrew Burt, Kim Calders, Cecilia Chavana-Bryant, Daniel Clewley, Laura Duncanson, Brieanne Forbes, Sean Krisanski, Yadvinder Malhi, David Moffat, Niall Origo, Alexander Shenkin, and Wanxin Yang. TLS2trees: A scalable tree segmentation pipeline for TLS data. *Methods in Ecology and Evolution*, 14(12):3083–3099, 2023. doi: 10.1111/2041-210X.14233.

Zhouxin Xi and Chris Hopkinson. 3d graph-based individual-tree isolation (Treeiso) from terrestrial laser scanning point clouds. *Remote Sensing*, 14(23):6116, 2022. doi: 10.3390/rs14236116.

Binbin Xiang, Maciej Wielgosz, Theodora Kontogianni, Torben Peters, Stefano Puliti, Rasmus Astrup, and Konrad Schindler. Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3d deep learning. *Remote Sensing of Environment*, 305:114078, 2024. doi: 10.1016/j.rse.2024.114078.

Binbin Xiang, Maciej Wielgosz, Stefano Puliti, Kamil Král, Martin Krůček, Azim Missarov, and Rasmus Astrup. ForestFormer3D: A unified framework for end-to-end segmentation of forest LiDAR 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24717–24727, 2025. doi: 10.1109/ICCV51701.2025.02292.

A Additional implementation and visualization details

A.1 Neural architecture and training

Architecture and inference outputs. The local predictor uses a sparse U-Net with channel widths $[32, 64, 96, 128, 160]$ and 32-dimensional output point features. The query decoder has six layers, model dimension 256, eight attention heads, feed-forward dimension 1024, GELU activation, and zero dropout. Its configuration uses 300 query points and two semantic queries. A separate head attached to the backbone predicts two-dimensional horizontal offsets. The native mask branch supplies confidence scores, which are mapped to scene points during inference. Figure 1 distinguishes the two downstream paths: aggregated offsets provide anchors for graph assignment, while native confidence contributes to instance-validity descriptors.

Data and optimization. Final-stage neural training uses the training subdivisions of Litchfield, Wytham, Ofental, and Robson Creek. Each epoch consists of 500 sampled cylindrical crops with batch size one. Table 5 lists the launch configuration for this stage. The saved checkpoint is the final scheduled epoch; the checkpoint hook does not perform automatic best-validation selection.

Augmentation applies random vertical-axis rotation, independent horizontal flips with probability 0.5, uniform scaling in $[0.9, 1.1]$, vertical scaling in $[0.85, 1.15]$, and small rotations about horizontal axes with angles in $[-5^\circ, 5^\circ]$. Coordinate jitter has standard deviation 0.01 m, and point dropout probability is 0.15. Geometric transforms are applied consistently to coordinates and their axis/anchor supervision. The additional canopy-density dropout is disabled in the final stage. The archived environment records PyTorch 2.7.0, CUDA 12.8, and MMEngine 0.10.7.

Table 5: Configuration of the final joint neural-training stage.

Setting	Value
Epochs / updates	20 / 10,000
Crops per epoch / batch size	500 / 1
Crop radius	16 m
Cached point resolution	0.02 m
Crop sampling grid / neural voxel size	0.20 m / 0.20 m
Maximum training crop points	400,000
Validation/inference point cap	640,000
Optimizer	AdamW
Initial learning rate / weight decay	5×10^{-5} / 0.05
Learning-rate schedule	Polynomial decay, power 0.9
Gradient clipping	L2 norm, maximum 10
Seed	1030761292
Checkpoint interval	Every 5 epochs
Training device	One NVIDIA RTX 6000D

A.2 Neural training objectives

Prediction losses. Final-stage training retains the native instance and semantic objectives, together with offset, trunk, tree-top, discriminative embedding, and superpoint-contrastive objectives. The instance criterion combines mask binary cross-entropy, Dice, and score losses with weights 1, 1, and 0.5, respectively, under one-to-many matching and intermediate decoder supervision. Decoder semantic supervision has weight 0.2; the additional binary tree/non-tree head is also supervised. The offset target is the horizontal displacement from a foreground voxel to its annotated tree anchor, and the offset head uses L1 loss with weight 1. Trunk logits use weighted binary cross-entropy against the cached trunk proxy; the positive-class weight is clipped to $[1, 30]$. The trunk, tree-top, discriminative, and superpoint-contrastive terms have outer weights of 1 in the final configuration. The optional pairwise-margin term is disabled. These retained neural training objectives do not imply that all their outputs enter the final graph: the source decoder replaces neural tree/trunk cues with geometric classifiers and disables embedding and tree-top edge terms.

Horizontal-offset loss. For foreground voxels $\mathcal{V}_{\mathrm{fg}}$, the target $\mathbf{o}_v^\star$ is the annotated horizontal anchor minus the voxel position. Foreground voxels contain a strict majority of tree points. The implementation averages the absolute error over voxels and both horizontal coordinates:

$$\mathcal{L}_{\mathrm{off}} = \frac{1}{2|\mathcal{V}_{\mathrm{fg}}|} \sum_{v \in \mathcal{V}_{\mathrm{fg}}} \|\widehat{\mathbf{o}}_v - \mathbf{o}_v^\star\|_1.$$

Crops without foreground voxels contribute zero offset loss.

Superpoint contrastive objective. Mean and maximum pooling of the 32-dimensional backbone features produce a 64-dimensional superpoint feature. A two-layer projection head, $\mathrm{Linear}(64, 64)$–ReLU–$\mathrm{Linear}(64, 64)$, followed by L2 normalization, produces the contrastive representation. Superpoints use majority instance labels to define same-tree positives and different-tree negatives. At most 512 contrastive anchors are sampled per crop. Each has four positive and 64 negative samples; half of the positives are sampled with a distance bias and half uniformly within the same tree. Half of the negatives are preferentially sampled within 0.4 m, with global fallback when local negatives are unavailable. Temperature is 0.1 and the loss weight is 1. The projection embeddings are used for auxiliary supervision and are not passed directly to the final graph decoder.

Sampled superpoint contrastive loss. Let $\mathcal{A}$ denote the sampled anchor superpoints. Each anchor a has $P = 4$ sampled positives $\mathcal{P}_a$ and 64 sampled negatives $\mathcal{N}_a$. For normalized embeddings $\mathbf{z}$ and temperature $T = 0.1$, the loss is

$$\mathcal{L}_{\mathrm{sp}} = -\frac{1}{|\mathcal{A}|P} \sum_{a \in \mathcal{A}} \sum_{p \in \mathcal{P}_a} \log \frac{\exp(\mathbf{z}_a^\top \mathbf{z}_p / T)}{\exp(\mathbf{z}_a^\top \mathbf{z}_p / T) + \sum_{n \in \mathcal{N}_a} \exp(\mathbf{z}_a^\top \mathbf{z}_n / T)}.$$

Table 6: Spatial scales used for geometric descriptor computation. The neighbor limit counts support representatives; their weights preserve the number of represented cached points.

Radius (m)	Support resolution (m)	Maximum neighbors
0.05	0.02	64
0.10	0.04	64
0.20	0.08	64

Each positive is compared separately against the sampled negatives. Sampling is with replacement, so repeated samples contribute with their sampling multiplicity. The geometric and validity classifiers are fitted separately from these neural objectives.

Separation from downstream fitting. Graph assignment and gradient-boosted geometric/validity classifiers are not optimized by backpropagation through the neural network. They follow the separate fitting procedures below. In the reported ground-domain adaptation experiment, the neural checkpoint remains fixed while the geometric tree and trunk classifiers are fitted on target-domain training data.

A.3 Geometric descriptors and point-cloud density

Sampling resolution and local support. We distinguish point-cloud sampling density, neighborhood support, and voxel occupancy. The cached source point clouds use a nominal 0.02 m resolution. The neural predictor uses 0.20 m voxels, whereas scene-level assembly uses 0.10 m voxels. These resolutions specify spatial representations; they do not imply a constant number of points per square or cubic meter. Local support is measured explicitly from the cached points used by the descriptor builder.

Multi-scale computation. Descriptors are evaluated at 0.04 m query-cell centers and mapped back to points. For each radius r, the builder retrieves at most 64 nearest support representatives and retains those within r. Coarser support representatives are cell means with weights equal to their numbers of represented cached points. At the finest scale, cached points are used directly with unit weights.

For query location $\mathbf{q}$, let $\mathcal{N}_r(\mathbf{q})$ be the retained representatives, with positions $\mathbf{y}_j$ and weights ω_j. Define

$$N_r(\mathbf{q}) = \sum_{j \in \mathcal{N}_r(\mathbf{q})} \omega_j, \qquad \boldsymbol{\mu}_r = \frac{\sum_j \omega_j \mathbf{y}_j}{N_r(\mathbf{q})},$$

$$\mathbf{C}_r = \frac{1}{N_r(\mathbf{q})} \sum_j \omega_j (\mathbf{y}_j - \boldsymbol{\mu}_r)(\mathbf{y}_j - \boldsymbol{\mu}_r)^\top.$$

For eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and principal unit eigenvector $\mathbf{v}_1$, the five descriptors are

$$\mathbf{g}^{(r)} = \left[\frac{\lambda_1 - \lambda_2}{\lambda_1}, \frac{\lambda_2 - \lambda_3}{\lambda_1}, \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3}, |\mathbf{v}_1^\top \mathbf{e}_z|, \log(1 + N_r) \right].$$

Eigenvalues and denominators are stabilized numerically. All five features are set to zero when the represented support count is below four. Concatenating the three scales gives 15 geometric channels.

Meaning of the density feature. The density channel $\log(1 + N_r)$ measures local sampling support at a fixed radius. It counts represented cached points, rather than simply counting the retrieved representatives, and is therefore not capped at $\log(65)$. It is a support-count descriptor rather than a calibrated volumetric density. The separate graph occupancy n_u counts the retained tree points assigned to voxel u on the decoder support. Neither quantity measures the number of trees per unit ground area.

Table 7: Shared graph parameters in the source evaluation.

Parameter	Value
Graph voxel size / spatial neighbors	0.10 m / 6
Anchor weight λ_a	5
Occupancy coefficient γ	40
Lateral penalty λ_{lat}	8
Height gain η_h / reference H_{ref}	6 / 10 m
Dispersion gain η_m / reference	6 / 0.6 m
Primary-seed height band / trunk threshold	2 m / 0.30
DBSCAN radius / minimum support	0.15 m / 4 nodes
Persistence minimum occupied height bins	5
Persistence non-maximum-suppression distance	0.7 m
Minimum distance of supplementary seeds from trunk seeds	2 m
Minimum size of unseeded components for fallback	50 nodes
Minimum candidate size $n_{\min}$	1000 points

Tree and trunk classifiers. The classifiers use 17-dimensional inputs $\mathbf{r}_i = [h_i, p_i^z, \mathbf{g}_i]$. Local height h_i is the point elevation relative to the minimum elevation in its 2 m horizontal grid cell. Semantic fitting computes this height proxy after training point sampling; trunk fitting computes it before sampling. Both classifiers use histogram gradient boosting with 250 iterations, learning rate 0.08, at most 63 leaves, and random seed 0. Tree-membership fitting uses at most three million training points. Trunk fitting uses at most four million points within the 4 m local-height band. Trunk scores outside that band are set to zero. Source tree membership uses a probability threshold of 0.30.

Trunk targets use the cached robust axis-based proxy when available, with the legacy proxy as fallback. The proxies identify points near an axis estimated from the annotated lower tree structure; they are not independently annotated stem labels. Source classifiers are fitted on the corresponding training subdivision of each plot.

A.4 Scene-level graph decoding

Evidence aggregation and graph construction. Crop offsets are mapped to scene points and averaged. Adding the mean horizontal offset to a point's XY coordinates gives its anchor $\mathbf{a}_i$. Tree-labeled points are grouped into 0.10 m voxels. Each node stores a mean position $\mathbf{c}_u$, mean anchor $\mathbf{a}_u$, point count n_u, and maximum trunk score. Symmetrized spatial six-nearest-neighbor connections define the graph. Native confidence is retained for instance descriptors.

Edge costs and sampling support. For adjacent nodes, the source configuration uses

$$w_{uv} = d_{uv}^2(1+\beta_{uv})\left[1+\gamma\left(\frac{1}{n_u+1}+\frac{1}{n_v+1}\right)\right]+\lambda_a\|\mathbf{a}_u-\mathbf{a}_v\|_2^2,$$

where $d_{uv} = \|\mathbf{c}_u - \mathbf{c}_v\|_2$. The occupancy factor increases the cost of edges supported by fewer tree points. The transition modulation is

$$\beta_{uv} = \left[\lambda_{\mathrm{lat}}\left(1+\eta_h\frac{\bar{H}_{uv}}{H_{\mathrm{ref}}}\right)+\eta_m\bar{m}_{uv}\right]\frac{\|\mathbf{c}_u^{xy}-\mathbf{c}_v^{xy}\|_2}{d_{uv}+\epsilon}.$$

Here, $\bar{H}_{uv}$ averages the nonnegative heights of the endpoints above the graph's local ground estimate, and $\bar{m}_{uv}$ averages their normalized anchor dispersion. For a horizontal column C,

$$\boldsymbol{\mu}_C = \frac{\sum_{u\in C} n_u\mathbf{a}_u}{\sum_{u\in C} n_u}, \quad \sigma_C = \sqrt{\frac{\sum_{u\in C} n_u\|\mathbf{a}_u-\boldsymbol{\mu}_C\|_2^2}{\sum_{u\in C} n_u}}, \quad m_u = \min(1, \sigma_{C(u)}/0.6).$$

Columns have 1 m horizontal width and spread is measured in meters. Neural anchors therefore contribute both through explicit anchor agreement and through the dispersion modulation.

Seeds and point assignment. Near-ground nodes with sufficient trunk support are clustered in XY to form primary seed sets S_k. Vertically persistent structures provide supplementary seeds where nearby trunk seeds are absent. Sufficiently large connected components without any seed receive a fallback seed at their lowest node. Multi-source Dijkstra assigns each reachable node to the seed set with minimum accumulated path cost:

$$\ell(v) = \arg\min_{k} \min_{s \in S_k} \mathrm{dist}_{\mathcal{G}}(s, v).$$

Unreachable nodes remain unassigned. Voxel labels are mapped back to points, and candidates below $n_{\min}$ are removed before validity feature extraction. This size refers to the decoder point support, not the number of points after raw-cloud export. Embedding-distance and tree-top edge terms are disabled in the reported source configuration.

A.5 Instance-validity fitting

Candidate descriptors. For candidate I_k, let $N_k = |I_k|$, $z_{\min}$ and $z_{\max}$ be its minimum and maximum elevations, and $H_k = z_{\max} - z_{\min}$. Horizontal radii are measured from the mean XY position of all candidate points. The 21-dimensional descriptor contains:

- $\log(1 + N_k)$, vertical extent, and minimum/maximum local height;
- mean, median, and 95th-percentile horizontal radius, and aspect ratio $H_k / \max(2r_{95}, \epsilon)$;
- mean, maximum, and standard deviation of trunk scores, and the fraction of points with trunk score above 0.5;
- the fraction of points within 2 m of $z_{\min}$;
- a stemness proxy: among points within 3 m of $z_{\min}$, the fraction within 0.3 m of the mean XY position, set to zero when that lower band contains at most ten points;
- mean native confidence and the binary tree-label fraction;
- five normalized vertical-profile bins over $(z_i - z_{\min}) / \max(H_k, \epsilon)$.

The profile bins sum to one. The binary tree-label fraction is constant for candidates composed exclusively of tree-labeled points.

Fitting and proxy labels. The validity model uses histogram gradient boosting with 80 iterations, learning rate 0.06, at most seven leaves, minimum leaf size 15, L2 regularization 5, and random seed 0. Per-plot fitting uses assembled validation candidates. Proxy labels are obtained by one-to-one Hungarian matching of point-membership IoU. Matches at IoU ≥ 0.5 are positives, and eligible GT objects contain at least 100 points on the fitting support. These proxies differ from the benchmark evaluation protocol used for reporting source detection results.

Threshold selection and retention. Out-of-fold probabilities use stratified shuffled folds, up to five folds, with random seed 0. The threshold is selected from 45 values between 0.02 and 0.90 to maximize validation proxy F1 while retaining at least 0.98 of the positive validation candidates. A zero threshold is used if no tested threshold meets the constraint. The final model is fitted on all validation candidates. Accepted candidates retain their memberships; rejected points receive background ID -1. Prediction IDs greater than or equal to zero are valid instance IDs.

The retention requirement of 0.98 was inherited from exploratory development in which test results had been inspected. Per-plot thresholds were subsequently selected from out-of-fold validation predictions under this fixed requirement.

A.6 Evaluation settings

Source benchmark. The source evaluation uses the spatially separated training, validation, and test subdivisions of Litchfield, Wytham, Ofental, and Robson Creek. Geometric classifiers use training subdivisions, and validity fitting uses validation candidates. The three-stage comparison uses the same neural checkpoint. Native denotes the original scene-level mask merging and filtering procedure; geometric assembly combines geometric point evidence, graph assignment, and size filtering; complete GIVF adds validity filtering on those same assembled candidates.

Ground-domain adaptation. Two shared geometric classifiers are fitted using CULS plot 1, RMIT train, TUWIEN train, and NIBIO MLS plots 107, 56, 57, and 82. Within each acquisition family, training cache points within 0.05 m of raw or cached test observations are excluded. If an excluded foreground point belongs to an annotated training instance, all points of that instance are removed.

The procedure excludes 371,451 points and six training tree instances from CULS, 68,108 points and 16 instances from RMIT, and 751,890 points and 40 instances from TUWIEN. The four NIBIO MLS training supports are unchanged under this rule. The exclusion uses test coordinates and training instance labels, but no test labels.

Local heights and geometric descriptors are recomputed on retained training points. Geometric channels retain the original float16 cache precision before conversion to float32 classifier inputs. Each scene contributes 300,000 semantic samples and 300,000 trunk samples, giving 2.1 million samples per classifier. Both classifiers use the original histogram-gradient-boosting configuration with at most 250 iterations and random seed 0.

Semantic fusion uses the union of native tree predictions and geometric predictions at threshold 0.30. Geometric trunk scores are applied within the 4 m local-height band. The neural predictor, graph parameters, and 1000-point candidate threshold remain fixed. Evaluation uses size-filtered graph candidates and does not apply the source validity model.

Predictions are mapped to original points by nearest-neighbor assignment within 0.05 m and evaluated using Hungarian matching at membership IoU > 0.5. All positive GT instance IDs are evaluated, and nonnegative prediction IDs are valid instances. The support-exclusion rule does not certify complete geographical separation between training and test regions.

Metric aggregation. Source detection scores follow the benchmark implementation. Macro precision, recall, and F1 are computed by averaging the corresponding per-plot scores separately. Matched-tree metrics summarize successfully matched prediction–GT pairs. The ground-domain raw-point matching protocol is reported separately from the source benchmark protocol.

A.7 Visualization protocol and additional results

Full-window comparisons. For each plot, 850,000 rows are sampled from the saved prediction point support using a fixed seed. Coordinates and GT labels are checked to match exactly across native, graph, and GIVF stages. Candidate windows have 12 m side length, with centers on a 6 m grid. They are screened using point support and GT occupancy, then ordered by distance to the median XY location rather than prediction quality. At most 120,000 points are displayed per selected window.

Instance colors use a one-to-one correspondence derived from sample IoU above 0.1. This correspondence only aligns colors and does not define a reported detection score. Unmatched predicted instances receive separate colors. In full-color panels, gray indicates background or unassigned points. All columns in a row share the same points and camera.

Local target-tree views. Figure 2 enlarges the original A–C locations in Litchfield, Ofental, and Wytham. Each zoom covers 5.4×4.8 m in the shared orthographic projection and retains all depths in the original sample. The same annotated target-tree points are emphasized across columns, retaining their original instance colors or unassigned status. Other points are uniformly faded and drawn beneath the target points. This diagnostic display illustrates target-point assignments; the full-window panels provide the neighboring-instance context needed to assess mixed memberships.

Whole-plot and method figures. Whole-plot overviews display 160,000 sampled points per plot without spatial cropping. These display samples are separate from the point supports used for quantitative evaluation. The method figure uses 9,170 observed Litchfield points in its input thumbnail. Crop outlines, graph propagation, candidate instances, and validity filtering are mechanism schematics rather than experimental segmentation results.

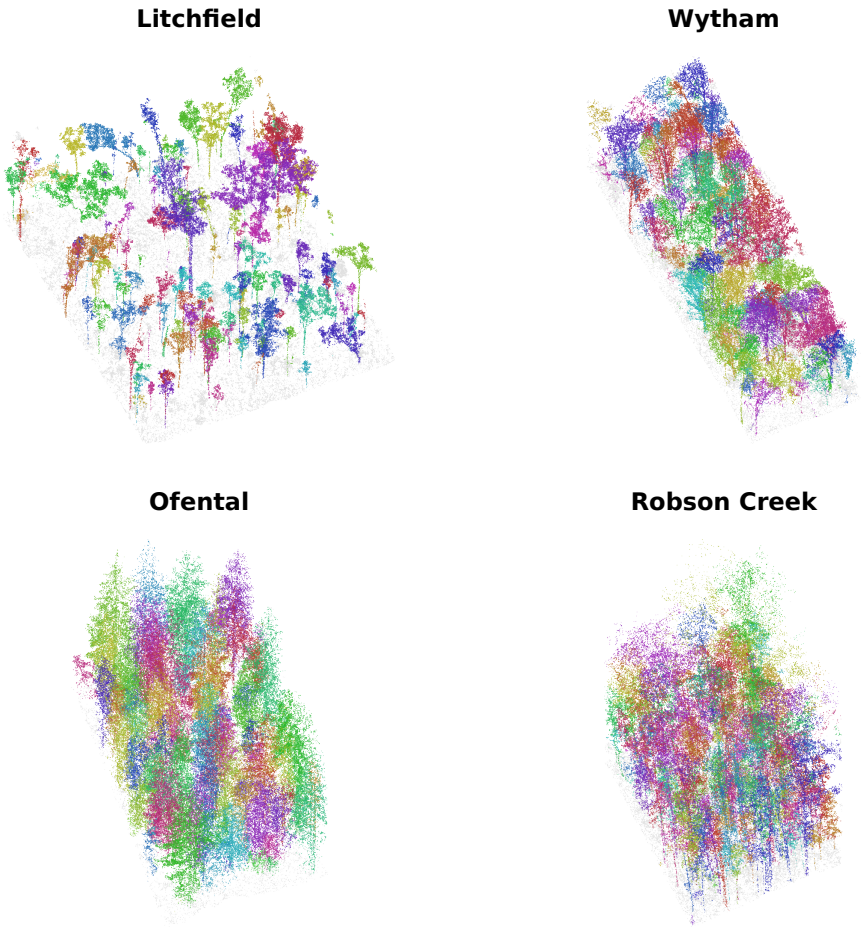

Figure 3: Whole-plot GIVF assignments for the four source test plots. Colors distinguish predicted instances, and gray denotes unassigned points. Each panel uses a fixed sample of the complete saved scene.

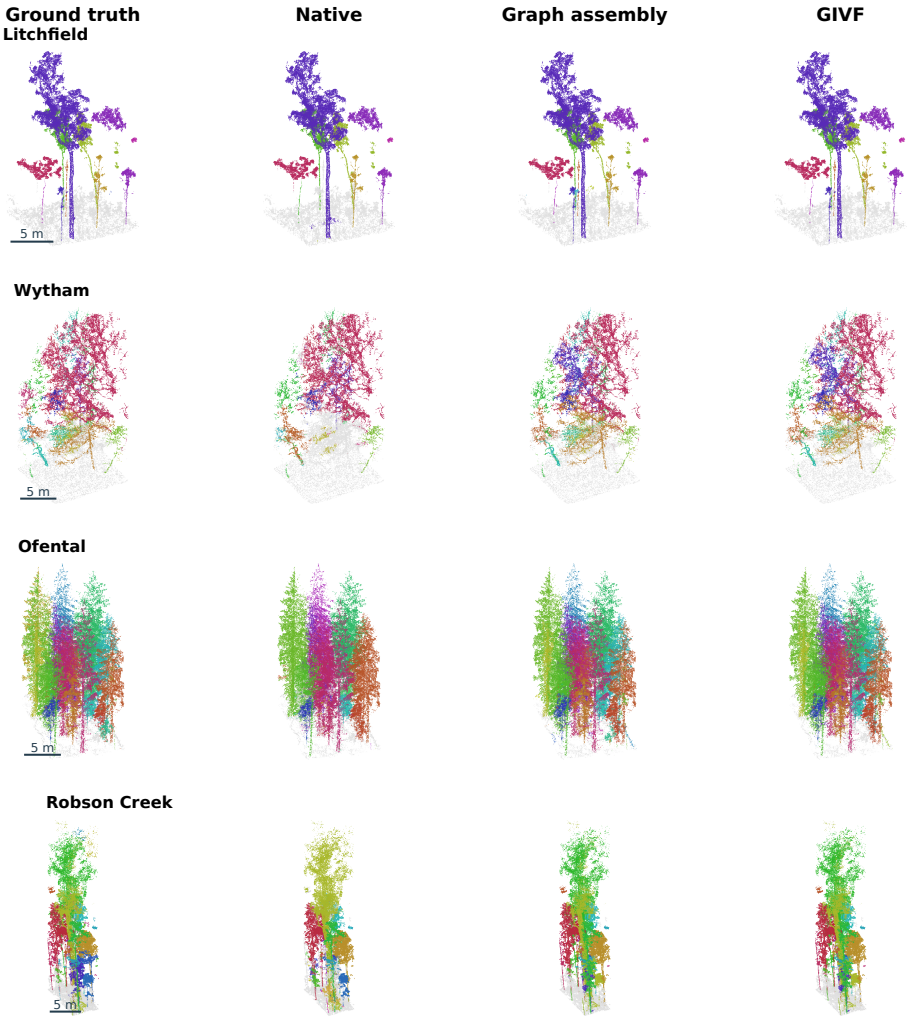

Figure 4: Full-window comparisons providing context for the local examples in Figure 2, together with Robson Creek. Columns show ground truth, native predictions, geometric assembly, and GIVF. Within each row, point samples, viewing parameters, and spatial bounds are identical across stages.