

RefTr: Recurrent Refinement of Confluent Trajectories for 3D Tubular Tree Centerlines

Roman Naeem^{*[0009-0001-2625-125X]}, David Hagerman^[0000-0002-5193-3789],
Jennifer Alvé^{n[0000-0003-4195-9325]}, and Fredrik Kahl^[0000-0001-9835-3020]

Chalmers University of Technology, 412 96 Gothenburg, Sweden
{nroman,david.hagerman,alven,fredrik.kahl}@chalmers.se

Abstract. Tubular tree structures such as blood vessels and lung airways are central to many clinical tasks, including diagnosis, treatment planning, and surgical navigation. Accurate centerline extraction with correct topology is essential, as missing small branches can lead to incomplete assessments or overlooked abnormalities. We propose RefTr, a 3D image-to-graph framework that generates tubular tree centerlines via recurrent refinement of confluent trajectories. RefTr adopts a Transformer-based Producer-Refiner architecture in which the Producer predicts candidate trajectories and a shared Refiner iteratively aligns with target branches. The confluent trajectory representation enables whole-branch refinement while explicitly enforcing valid topology. This recurrent scheme improves precision and reduces decoder parameters by $2.4\times$ compared to the state-of-the-art. We further introduce an efficient non-maximum suppression algorithm for spatial tree graphs to merge duplicate branches and extend evaluation metrics to be radius-aware for robust comparison. Experiments on multiple public datasets demonstrate higher recall and precision, faster inference, and substantially fewer parameters, highlighting the effectiveness of RefTr for 3D tubular tree analysis. Code is available at <https://github.com/RomStriker/RefTr>.

Keywords: Image-to-graph · Centerline Extraction · Tree Topology

1 Introduction

Tubular trees such as blood vessels and lung airways are essential for material transport in the human body, and abnormalities in these networks are linked to many diseases. Accurate, topologically correct centerline extraction is essential for diagnosis and treatment planning [8, 11], surgical navigation [5], hemodynamic analysis [10], and vascular morphometry [3, 6]. Missing small branches can lead to incomplete assessments and clinical risk, making high recall crucial. Centerline graphs provide a compact, interpretable representation, easier to annotate than dense masks and amenable to enforcing valid tree topology.

Segmentation-based methods, including topology-preserving and skeletonization losses [22, 20, 19, 16], require costly voxel-level labels and post-processing,

^{*} Corresponding author.

and lack end-to-end graph prediction, addressing a different problem setting than ours. Graph-based methods [15, 18] predict nodes and edges but do not guarantee valid tree topology and scale poorly. Sequential tracking methods [14, 13] iteratively generate centerlines but suffer from low recall due to imbalanced bifurcation/termination classification and cannot correct early errors. Classical multiple-hypothesis tracking [17] retains several candidate trees during reconstruction but ultimately prunes all but one, discarding potentially useful structures.

To address these limitations, we propose RefTr, a parameter-efficient Transformer based [1, 21] 3D image-to-graph model that generates centerline trees via recurrent refinement of confluent trajectories. Prior work [24] defines confluence as a geometric property of overlapping oriented smooth curves and enforces it to reconstruct vessel trees in an unsupervised setting; we instead learn a discrete confluent trajectory representation in which trajectories share a prefix and split at a divergence index, as illustrated in Fig. 1a. A Producer predicts multiple trajectory candidates rooted at the patch center and a shared Refiner recurrently aligns them with the target, reducing decoder parameters by $2.4\times$ versus the state-of-the-art Trexplorer Super [13].

To maximize recall, RefTr predicts multiple candidates per target via many-to-one matching, creating an ensembling effect. The resulting duplicate trajectories are suppressed at the patch level using pairwise divergence positions (duplicates do not diverge), and globally via Tree Non-Maximum Suppression (TNMS), a novel, efficient algorithm that merges overlapping branches while preserving topology. For robust comparison, we extend the Trexplorer Super [13] evaluation framework with radius-aware thresholds that scale with local branch radius.

Our contributions include: (1) RefTr, a parameter-efficient 3D image-to-graph framework predicting confluent trajectories with the highest recall and precision among compared methods; (2) Tree Non-Maximum Suppression, an efficient algorithm for removing duplicate branches while preserving tree topology; and (3) radius-aware evaluation metrics for robust comparison across datasets with varying branch radii.

2 Method

RefTr generates centerline graphs of tubular trees from CT volumes using a Transformer-based Producer–Refiner architecture and a confluent trajectory representation. The model architecture is shown in Fig. 1.

Problem Formulation and Inference. The goal is to infer a centerline tree graph $G = (V, E)$ from a 3D image patch. Each node $\mathbf{v} \in V$ represents a spatial position and the branch radius at that position. RefTr represents the tree as a set of confluent trajectories $\mathcal{T} = \{T_i\}_{i=1}^n$, as shown in Fig. 1a. Each trajectory $T_i = (\mathbf{v}_{i,1}, \dots, \mathbf{v}_{i,L})$ is a node sequence rooted at the patch center. The set is confluent in that any pair T_i, T_j shares an identical prefix and separates only at a predicted pairwise divergence index d_{ij} , i.e. $\mathbf{v}_{i,k} = \mathbf{v}_{j,k}$ for $k \leq d_{ij}$.

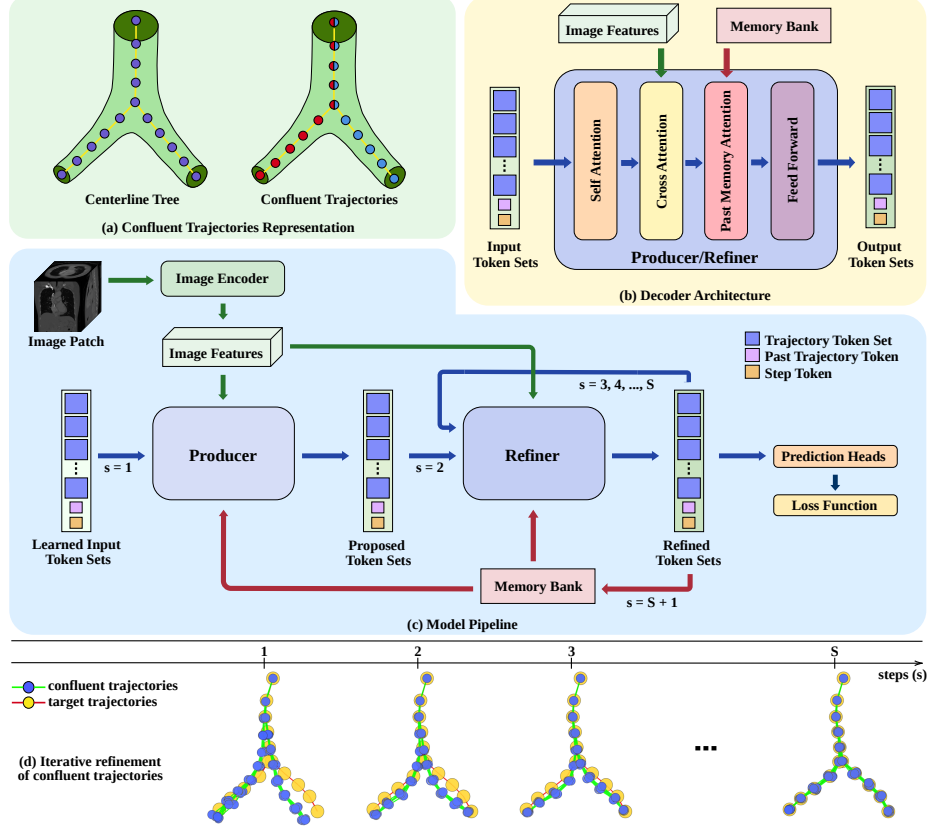


Fig. 1. Overview of the RefTr architecture. **(a)** A tubular tree centerline graph with corresponding confluent trajectories and divergence position at the fifth node. **(b)** Producer/Refiner architecture with Self-, Cross-, and Past-Memory attention. **(c)** Model pipeline including Image Encoder, Producer, Refiner, and prediction heads. The Producer uses learned token sets and image features to propose multiple trajectory embeddings, which the Refiner iteratively refines over S steps. Loss is computed at every step, including the Producer step (omitted for clarity). One refined token set per continuing branch is stored in the Memory Bank for the next patch. **(d)** Recurrent refinement of multiple confluent trajectories toward two target trajectories over S steps.

Unlike per node tracking, this representation refines whole branches jointly and maps to a tree by construction: shared prefixes become shared paths, divergence indices become bifurcations, and predicted end positions become terminations. Topology is therefore valid independently of TNMS, which only merges duplicate branches. During inference, branches extending beyond trajectory length are explored recursively using new patches.

RefTr Architecture. RefTr consists of an image encoder and a customized Transformer decoder with two modules, a Producer and a Refiner. For a fair comparison, we adopt the same encoder as Trexplorer Super [13], a modified SwinUNETR [4]. The Producer generates n initial trajectory proposals from learned token sets, and the Refiner recurrently updates them over $S - 1$ steps using shared parameters. The learned token sets are fixed, trainable embeddings optimized end-to-end (analogous to DETR object queries [1]); they are image-agnostic until cross-attention injects patch features. Both modules use a single decoder block. This refines alignment with the m target trajectories ($n > m$) while remaining parameter-efficient. The full pipeline is shown in Fig. 1c.

Each trajectory token set includes L node tokens that predict node positions and radii, an end token that predicts the termination position, and a divergence token that is pairwise concatenated with those of other trajectories to predict branching positions between them. End and divergence positions lie along the trajectory, ranging from 0 to L .

Self-attention models inter-trajectory interactions, enabling many-to-one matching that evenly distributes predictions across targets for an ensembling effect. Cross-attention aggregates image features. Past-Memory attention incorporates context from a memory bank that stores token sets of ancestor trajectories from up to five previously processed patches. This maintains trajectory direction for large-radius branches with limited field of view and reduces cross-patch duplicates. As in Trexplorer Super, each training sample contains six patches along a single trajectory, so the memory bank holds up to five ancestor token sets. The detailed decoder architecture is shown in Fig. 1b.

We further use a past trajectory token encoding up to the last ten positions and radii from the patch center, helping determine centerline direction, especially when the memory bank is empty. We also include a step token that indicates the current refinement step and regulates the Refiner’s update magnitude.

Matching and Loss. Predicted trajectories are matched to targets using a combined L1 (Manhattan) cost over position and radius differences. To allow multiple predictions per branch, we employ many-to-one matching by replicating targets and solving a bipartite assignment with the Hungarian algorithm [7]. Unlike one-to-many DETR variants [2, 25] that discard surplus predictions at inference, RefTr keeps and merges them, yielding an inference-time ensemble.

All components use L1 loss. The total loss aggregates position and radius errors across steps, with end and divergence position losses at the final step, where trajectories are most refined:

$$\mathcal{L} = \sum_s (\alpha_{\text{pos}} \mathcal{L}_{\text{pos}}^{(s)} + \alpha_{\text{rad}} \mathcal{L}_{\text{rad}}^{(s)}) + \alpha_{\text{end}} \mathcal{L}_{\text{end}} + \alpha_{\text{div}} \mathcal{L}_{\text{div}}. \quad (1)$$

To ensure consistent supervision, the matching is computed only once using the predictions from the first step and reused for the remaining $S - 1$ steps. We set $\alpha_{\text{pos}}=4.2$, $\alpha_{\text{rad}}=1.2$, $\alpha_{\text{end}}=0.9$, $\alpha_{\text{div}}=0.3$, tuned on the validation split. Re-matching at every step destabilizes training, as reassignment disrupts the progressive alignment of each trajectory with its target.

Tree Non-Maximum Suppression. Many-to-one matching yields an ensembling effect at the cost of duplicate trajectories, which are first filtered at the patch level using the pairwise divergence position. To remove residual duplicates globally, we introduce Tree Non-Maximum Suppression (TNMS), an efficient algorithm that merges overlapping branches while preserving tree topology.

Given a predicted tree with N nodes, TNMS performs a pre-order traversal and flags duplicate nodes using a radius-adaptive spatial threshold $\tau = \max(\tau_{\min}, \tau_{\text{pos}} \cdot r)$, with $\tau_{\text{pos}}=0.3$ and $\tau_{\min}=2.0$. Branches whose duplicate fraction exceeds $\rho=0.2$ are merged, and any cycles are resolved by retaining edges closest to the root. Using KD-tree queries, the algorithm runs in $\mathcal{O}(N \log N)$ time, processing graphs with thousands of nodes in under one second on a CPU.

3 Experiments and Results

Datasets. We evaluate on the benchmark centerline datasets [12, 13]: a synthetic dataset, the ATM’22 airway dataset [23], and the Parse 2022 pulmonary artery dataset [9].

Evaluation Metrics. Following Trexplorer Super [13], we report metrics at the node, branch, and graph levels. To enable a more robust evaluation, we introduce radius-aware matching thresholds that scale with the ground truth branch radius. A predicted node is considered a true positive if it lies within $\max(1.5, \tau^{\text{rad}} \cdot r)$ of a previously unmatched target node. Duplicate predictions are treated as false positives. At the node-level, we report radius-aware Average Precision (rAP), Average Recall (rAR), and Average F1-score (rF1), averaged over $\tau^{\text{rad}} = 0.25:0.05:0.75$ to reduce threshold sensitivity. For matched node pairs, we additionally report the mean absolute difference between predicted and ground truth radii.

For branch-level evaluation, a predicted branch is counted as a true positive if it overlaps with at least a fraction τ^{match} of nodes from an unmatched target branch. We report branch-level counterparts rBAP, rBAR, and rBF1, averaged over $\tau^{\text{match}} = 0.4:0.05:0.8$, with τ^{rad} fixed at 0.5. For graph-level evaluation, we report topological metrics, specifically the mean absolute error of Betti-0 (connected components) and Betti-1 (cycles), following Trexplorer Super. We retain the same dataset splits and protocol as prior work, upgrading only the matching thresholds; conclusions are unchanged under the legacy fixed-threshold metrics.

Implementation Details. Experiments were conducted on a single node with four NVIDIA A100 GPUs using mixed precision. RefTr was trained for 1.9 million iterations with $S = 8$ refinement steps, trajectory length $L = 10$, and $n = 20$ predicted branches. To ensure fair comparison with Trexplorer and Trexplorer Super [13, 14], we used the same dataset and train-test split, adopting the authors’ reported hyperparameters (26 tokens per bifurcation node, maximum 196

Table 1. Comparison of various models using point-level metrics.

Model	Point-level@ $\tau^{\text{rad}} = [0.25:0.05:0.75]$			
	rAP(%) $\uparrow$	rAR(%) $\uparrow$	rF1(%) $\uparrow$	Radius (MAE) $\downarrow$
Synthetic Dataset				
Vesselformer	48.53 $\pm$ 10.35	70.05 $\pm$ 2.76	53.15 $\pm$ 8.19	0.43 $\pm$ 0.011
Trexplorer	31.10 $\pm$ 9.52	78.64 $\pm$ 4.10	39.64 $\pm$ 8.68	0.23 $\pm$ 0.032
Trexplorer Super	92.10 $\pm$ 3.29	70.58 $\pm$ 3.03	77.98 $\pm$ 1.90	0.10 $\pm$ 0.006
RefTr (ours)	94.88 $\pm$ 0.77	79.39 $\pm$ 1.03	85.69 $\pm$ 1.30	0.45 $\pm$ 0.093
ATM'22 Dataset				
Vesselformer	24.85 $\pm$ 1.47	38.48 $\pm$ 1.00	29.84 $\pm$ 1.37	0.79 $\pm$ 0.020
Trexplorer	4.54 $\pm$ 1.44	5.90 $\pm$ 0.70	4.49 $\pm$ 0.29	0.98 $\pm$ 0.086
Trexplorer Super	71.32 $\pm$ 1.22	60.84 $\pm$ 3.38	61.45 $\pm$ 1.23	0.40 $\pm$ 0.021
RefTr (ours)	74.92 $\pm$ 1.17	66.73 $\pm$ 1.48	68.15 $\pm$ 0.59	0.60 $\pm$ 0.036
Parse 2022 Dataset				
Vesselformer	21.93 $\pm$ 2.06	18.21 $\pm$ 1.00	19.54 $\pm$ 2.69	1.11 $\pm$ 0.027
Trexplorer	11.37 $\pm$ 3.73	13.42 $\pm$ 7.80	11.36 $\pm$ 5.17	1.22 $\pm$ 0.304
Trexplorer Super	53.65 $\pm$ 5.91	35.91 $\pm$ 3.22	40.01 $\pm$ 3.86	0.58 $\pm$ 0.025
RefTr (ours)	53.68 $\pm$ 1.09	36.69 $\pm$ 1.12	42.37 $\pm$ 1.13	0.75 $\pm$ 0.054

Table 2. Comparison of various models using branch-level and graph-level metrics.

Model	Branch-level@ $\tau^{\text{match}} = [0.4:0.05:0.8]$			Graph-level (MAE)	
	rBAP(%)↑	rBAR(%)↑	rBF1(%)↑	Betti-0↓	Betti-1↓
Synthetic Dataset					
Vesselformer	20.43 ± 4.22	43.63 ± 4.72	25.43 ± 3.75	81.7 ± 16.8	653.5 ± 138.7
Trexplorer	30.58 ± 8.66	77.47 ± 4.11	39.40 ± 7.82	0.00 ± 0.0	0.00 ± 0.0
Trexplorer Super	92.35 ± 3.54	70.82 ± 3.06	78.22 ± 1.78	0.00 ± 0.0	0.00 ± 0.0
RefTr (ours)	96.75 ± 0.72	79.96 ± 1.15	86.76 ± 1.38	0.00 ± 0.0	0.00 ± 0.0
ATM'22 Dataset					
Vesselformer	11.70 ± 0.73	16.77 ± 1.07	13.46 ± 0.81	312.5 ± 25.1	180.4 ± 35.9
Trexplorer	4.02 ± 1.71	2.06 ± 0.46	2.33 ± 0.31	0.00 ± 0.0	0.00 ± 0.0
Trexplorer Super	68.25 ± 1.75	60.91 ± 3.64	59.87 ± 1.39	0.00 ± 0.0	0.00 ± 0.0
RefTr (ours)	68.40 ± 1.97	66.41 ± 1.53	64.23 ± 0.89	0.00 ± 0.0	0.00 ± 0.0
Parse 2022 Dataset					
Vesselformer	9.23 ± 0.80	6.31 ± 0.85	7.30 ± 0.45	410.1 ± 23.9	246.7 ± 78.1
Trexplorer	11.32 ± 3.56	11.77 ± 7.22	10.45 ± 4.98	0.00 ± 0.0	0.00 ± 0.0
Trexplorer Super	51.42 ± 5.58	34.41 ± 1.09	38.17 ± 3.63	0.00 ± 0.0	0.00 ± 0.0
RefTr (ours)	47.17 ± 1.19	32.31 ± 1.13	37.09 ± 1.16	0.00 ± 0.0	0.00 ± 0.0

tokens) and trained for 2 million iterations. For Vesselformer [15], we report the best performance obtained using the published hyperparameters, with 80 object tokens and 12 million training iterations. All models were trained from scratch under identical data splits.

Results. Tables 1 and 2 report mean and standard deviation over five runs across point-, branch-, and graph-level metrics. At the point level, RefTr achieves the highest recall and precision across all datasets. On Parse 2022, improvements

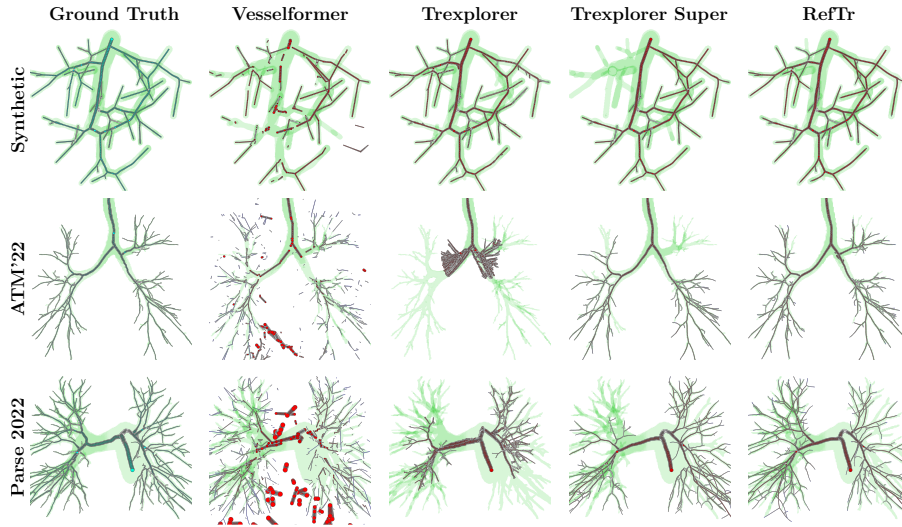


Fig. 2. Visual comparison of ground truth and predictions from RefTr and baseline methods on one sample per dataset. Marker sizes are proportional to local branch radii at node positions. The tubular structure mask (green) is for visualization only and not used during training or evaluation.

are smaller due to missing small branches in the ground truth, which RefTr correctly predicts (as verified by a radiologist review) but is penalized for, and encoder limitations from limited training data (72 training examples). RefTr’s radius MAE, while higher than Trexplorer Super’s, stays sub-voxel and has limited practical impact.

RefTr achieves superior branch-level metrics on the Synthetic and ATM’22 datasets. On Parse 2022, performance is comparable; here the missing small branches in the ground truth have a larger impact, since branches are weighted equally regardless of size. Graph-level metrics confirm topologically correct trees.

Table 3 shows RefTr reduces decoder parameters by $2.4\times$ versus Trexplorer Super, with faster inference and substantially less training time and memory.

Figure 2 provides a qualitative comparison for one example from each dataset. Compared to the previous state-of-the-art Trexplorer Super, RefTr shows visibly higher recall, capturing branches that Trexplorer Super misses.

Ablation Study. Table 4 presents an extensive ablation study of our method’s key components, including Recurrent Refinement, Many-to-One matching, Tree Non-Maximum Suppression (TNMS), Past-Attention, and Producer/Refiner design choices. We report the mean and standard deviation over three runs.

The study shows that Recurrent Refinement, Many-to-One matching, and Past-Memory attention are essential for achieving high recall. Removing TNMS instead leaves recall essentially unchanged but sharply lowers precision (Table 4),

Table 3. Number of parameters, runtime, training time, and memory usage for the compared models. Note: Vesselformer has a much higher effective runtime due to a multi-hour post-processing. Experiments conducted on four Nvidia A100 80GB GPUs.

Model	Parameters (M)			Runtime (ms/patch)	Training (hrs)	GPU Mem. (GBs)
	Encoder	Decoder	Total			
Vesselformer	82.83	13.59	98.73	23*	37.4	13.8
Trexplorer	23.31	25.22	55.46	222	18.6	51.34
Trexplorer Super	23.12	25.22	55.27	189	37.2	73.9
RefTr (ours)	23.12	10.50	42.32	156	18.2	39.5

Table 4. Ablation study evaluating the impact of the key components and design choices for the RefTr model on the ATM’22 dataset.

Description	Point-level@ $\tau^{\text{rad}} = [0.25:0.05:0.75]$		
	rAP(%) $\uparrow$	rAR(%) $\uparrow$	rF1(%) $\uparrow$
Key High-level Components			
No Recurrent Refinement ($S = 2$)	61.96 $\pm$ 2.17	58.08 $\pm$ 1.17	56.70 $\pm$ 1.66
No Many-to-one Matching	66.70 $\pm$ 1.27	63.25 $\pm$ 3.49	61.34 $\pm$ 1.68
No Tree Non-max Suppression	37.98 $\pm$ 2.42	69.96 $\pm$ 1.12	37.98 $\pm$ 3.87
Recurrent Refinement Steps			
Refinement Steps: $S = 4$	65.10 $\pm$ 6.32	55.72 $\pm$ 2.83	58.17 $\pm$ 3.42
Refinement Steps: $S = 6$	64.39 $\pm$ 5.17	60.93 $\pm$ 2.57	59.42 $\pm$ 3.19
Refinement Steps: $S = 10$	71.43 $\pm$ 3.12	65.16 $\pm$ 1.58	65.60 $\pm$ 1.91
Past Trajectory and Memory Attention			
No Past Trajectory Token	78.85 $\pm$ 3.24	37.07 $\pm$ 1.21	48.34 $\pm$ 2.78
No Past-Memory Attention	61.27 $\pm$ 2.12	66.38 $\pm$ 1.87	60.46 $\pm$ 2.24
Producer and Refiner Design Configurations			
Shared Producer and Refiner	65.18 $\pm$ 1.44	61.76 $\pm$ 1.32	60.27 $\pm$ 1.78
Increase Producer Blocks: 2	68.80 $\pm$ 2.24	62.88 $\pm$ 1.11	63.19 $\pm$ 1.83
Increase Refiner Blocks: 2	71.81 $\pm$ 2.07	64.67 $\pm$ 1.57	64.46 $\pm$ 1.98
Proposed Model	74.16 $\pm$ 1.45	67.72 $\pm$ 1.49	68.35 $\pm$ 1.27

confirming that the core generation mechanism recovers the structures, while TNMS only reclaims precision lost to residual cross-patch duplicates.

Sharing one decoder block between Producer and Refiner is suboptimal, as it must process two input types: learned and already refined token sets. Increasing the number of Producer blocks makes the initial proposals overly rigid, which limits the Refiner’s ability to correct early errors. Conversely, adding Refiner blocks makes each step apply larger updates that can push predictions off track and hinder recovery. Overall, the best performance is achieved with separate Producer and Refiner modules, each using a single decoder block and $S = 8$ refinement steps.

4 Conclusion

We present RefTr, a parameter-efficient framework for generating topologically accurate centerline graphs of tubular trees from 3D medical images. By predicting confluent trajectories and refining them recurrently, RefTr delivers state-of-the-art performance with substantially improved efficiency across multiple datasets, showing strong potential for clinical analysis. Future work will focus on improving the image encoder to further enhance feature quality and performance on challenging cases.

Acknowledgments. Compute and storage resources were provided by NAISS (grant 2022-06725, Swedish Research Council) and the Berzelius system at the National Supercomputer Centre, funded by the Knut and Alice Wallenberg Foundation.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
2. Chen, Q., Chen, X., Wang, J., Zhang, S., Yao, K., Feng, H., Han, J., Ding, E., Zeng, G., Wang, J.: Group detr: Fast detr training with group-wise one-to-many assignment. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6633–6642 (2023)
3. Choi, A.D., Marques, H., Kumar, V., Griffin, W.F., Rahban, H., Karlsberg, R.P., Zeman, R.K., Katz, R.J., Earls, J.P.: Ct evaluation by artificial intelligence for atherosclerosis, stenosis and vascular morphology(clarify): A multi-center, international study. *Journal of Cardiovascular Computed Tomography* **15**(6), 470–476 (2021)
4. Hatamizadeh, A., Nath, V., Tang, Y., Yang, D., Roth, H.R., Xu, D.: Swin unetr: Swin transformers for semantic segmentation of brain tumors in mri images. In: International MICCAI brainlesion workshop. pp. 272–284. Springer (2021)
5. Huang, D., Tang, W., Ding, Y., Wan, T., Chen, Y.: An interactive 3d preoperative planning and training system for minimally invasive vascular surgery. In: 2011 12th International Conference on Computer-Aided Design and Computer Graphics. pp. 443–449. IEEE (2011)
6. Khan, Z., Ngo, J.P., Le, B., Evans, R.G., Pearson, J.T., Gardiner, B.S., Smith, D.W.: Three-dimensional morphometric analysis of the renal vasculature. *American Journal of Physiology-Renal Physiology* **314**(5), F715–F725 (2018)
7. Kuhn, H.W.: The hungarian method for the assignment problem. *Naval research logistics quarterly* **2**(1-2), 83–97 (1955)
8. Li, H., Tang, Z., Nan, Y., Yang, G.: Human treelike tubular structure segmentation: A comprehensive review and future perspectives. *Computers in biology and medicine* **151**, 106241 (2022)
9. Luo, G., Wang, K., Liu, J., Li, S., Liang, X., Li, X., Gan, S., Wang, W., Dong, S., Wang, W., et al.: Efficient automatic segmentation for multi-level pulmonary arteries: The parse challenge. *arXiv preprint arXiv:2304.03708* (2023)

10. Miracourt, O., Salmon, S., Szopos, M., Thiriet, M.: Blood flow in the cerebral venous system: modeling and simulation. *Computer methods in biomechanics and biomedical engineering* **20**(5), 471–482 (2017)
11. Moccia, S., De Momi, E., El Hadji, S., Mattos, L.S.: Blood vessel segmentation algorithms—review of methods, datasets and evaluation metrics. *Computer methods and programs in biomedicine* **158**, 71–91 (2018)
12. Naeem, R.: Trexplorer super datasets (1.0). <https://doi.org/10.5281/zenodo.15888958> (2025). <https://doi.org/10.5281/zenodo.15888958>
13. Naeem, R., Hagerman, D., Alvéen, J., Svensson, L., Kahl, F.: Trexplorer super: Topologically correct centerline tree tracking of tubular objects in ct volumes. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 595–605. Springer (2025)
14. Naeem, R., Hagerman, D., Svensson, L., Kahl, F.: Trexplorer: Recurrent detr for topologically correct tree centerline tracking. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. pp. 744–754. Springer (2024)
15. Prabhakar, C., Shit, S., Paetzold, J.C., Ezhov, I., Koner, R., Li, H., Kofler, F.S., Menze, B.: Vesselformer: Towards complete 3d vessel graph generation from images. In: *Medical Imaging with Deep Learning*. pp. 320–331. PMLR (2024)
16. Qiu, Y., Shan, D., Wang, Y., Dong, P., Wu, D., Yang, X., Hong, Q., Shen, D.: A topology-preserving three-stage framework for fully-connected coronary artery extraction. *Medical Image Analysis* **103**, 103578 (2025)
17. Selvan, R., Petersen, J., Pedersen, J.H., de Bruijne, M.: Extracting tree structures in ct data by tracking multiple statistically ranked hypotheses. *Medical Physics* **46**(10), 4431–4440 (2019)
18. Shit, S., Koner, R., Wittmann, B., Paetzold, J., Ezhov, I., Li, H., Pan, J., Sharifzadeh, S., Kaissis, G., Tresp, V., et al.: Relationformer: A unified framework for image-to-graph generation. In: *European Conference on Computer Vision*. pp. 422–439. Springer (2022)
19. Shit, S., Paetzold, J.C., Sekuboyina, A., Ezhov, I., Unger, A., Zhylka, A., Plum, J.P., Bauer, U., Menze, B.H.: cldice-a novel topology-preserving loss function for tubular structure segmentation. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 16560–16569 (2021)
20. Tetteh, G., Efremov, V., Forkert, N.D., Schneider, M., Kirschke, J., Weber, B., Zimmer, C., Piraud, M., Menze, B.H.: Deepvesselnet: Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Frontiers in Neuroscience* **14**, 1285 (2020)
21. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. *Advances in neural information processing systems* **30** (2017)
22. Wittmann, B., Wattenberg, Y., Amiranashvili, T., Shit, S., Menze, B.: vesselfm: A foundation model for universal 3d blood vessel segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 20874–20884 (2025)
23. Zhang, M., Wu, Y., Zhang, H., Qin, Y., Zheng, H., Tang, W., Arnold, C., Pei, C., Yu, P., Nan, Y., et al.: Multi-site, multi-domain airway tree modeling. *Medical Image Analysis* **90**, 102957 (2023)
24. Zhang, Z., Marin, D., Drangova, M., Boykov, Y.: Confluent vessel trees with accurate bifurcations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9573–9582 (2021)

25. Zong, Z., Song, G., Liu, Y.: Detrs with collaborative hybrid assignments training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6748–6758 (2023)