

Uncertainty Visualization for Approximated Barycenter of Contour Trees Based on Partial Optimal Transport

Mingzhe Li*
University of Notre Dame

Fangfei Lan†
University of Lausanne

Gunther Weber‡
Lawrence Berkeley National Laboratory

Bei Wang §
University of Utah

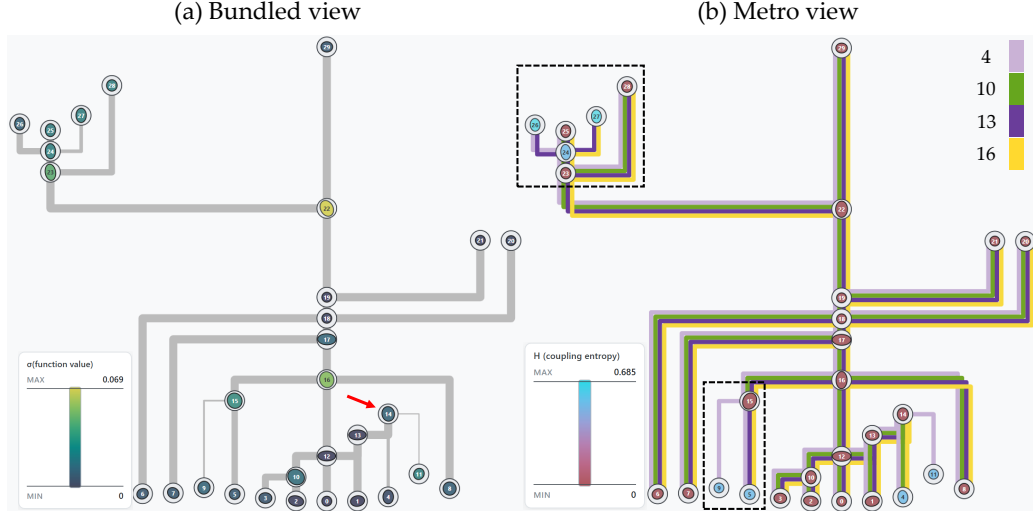


Fig. 1: Barycenter tree visualization for the Heated Cylinder ensemble using our framework. (a) Bundled view of the reconstructed barycenter tree using all 23 instances. Edge thickness encodes how many input trees contain each edge, revealing dominant structure across the ensemble. (b) Metro view showing per-input edge usage for a selected subset of four inputs (indices 4, 10, 13, and 16). Each colored line corresponds to one input tree, enabling direct comparison of structural variations and shared paths. Node colors encode function value uncertainty in (a) and coupling uncertainty in (b), while in-node ellipse shapes encode coordinate uncertainty in both views. Dotted regions highlight subtrees where inputs follow divergent paths, indicating localized structural uncertainty. The red arrow in (a) highlights saddle node 14, which violates the contour tree topology.

ABSTRACT

Analyzing ensembles of contour trees is challenging due to variability in both topology and node attributes across instances. We present a framework for summarizing and visualizing such ensembles via approximate barycenter contour trees computed using the Fused Partial Gromov-Wasserstein (FPGW) distance. These barycenters provide representative summaries that capture shared structural patterns while supporting partial correspondences for unmatched or transient features. Our approach first estimates a representative barycenter matrix under the FPGW formulation and then reconstructs an approximate contour tree from this matrix. We further introduce a visualization that integrates the structural summary with transport-induced uncertainty. Specifically, we encode variability in function values, spatial embedding, and node correspondences, and propose a MetroSets-inspired view to expose structural differences among ensemble members. We demonstrate the effectiveness of our method on flow simulation ensembles, where barycenter contour trees yield interpretable summaries of topological variability and highlight regions of structural and scalar uncertainty grounded in the underlying data.

Index Terms: Optimal transport, contour tree, uncertainty visualization, topological data analysis.

1 INTRODUCTION

Ensemble and time-varying scalar fields arise in many scientific domains, where analysts seek to identify shared structures across in-

stances and characterize variability. Contour trees provide a compact representation of such data by encoding the topological structure of scalar fields, and have been widely used for feature extraction and uncertainty analysis [6, 17, 32, 35]. In ensemble and time-varying settings, however, analysis must operate on *collections* of contour trees, requiring methods to summarize and compare multiple structures.

Recent optimal transport formulations offer flexible tools for comparing structured data and computing representative structures such as barycenters. However, applying these techniques to contour trees presents two key challenges. First, contour trees may exhibit unmatched or transient features across instances, requiring support for partial correspondences. Second, barycenters are typically expressed as matrices rather than valid contour trees, limiting their interpretability and utility for visualization.

In this work, we adopt the Fused Partial Gromov-Wasserstein (FPGW) framework [4] to analyze ensembles of contour trees. We represent each contour tree as an attributed measure network, where the graph structure encodes connectivity and node attributes capture function values and spatial locations. We reconstruct the barycenter network into a tree-like structure and visualize it using node uncertainty encodings for function values, spatial positions, and transport correspondences. A MetroSets-inspired design [16] further reveals structural differences across the ensemble.

We make the following contributions:

- A representation of contour trees as attributed measure networks for FPGW-based comparison, jointly capturing geometric, scalar, and topological information.
- A reconstruction pipeline that converts barycenter networks into contour-tree-like structures for interpretation and visualization.
- An uncertainty-aware visualization framework for barycenter contour trees that encodes structural consensus and variability in

*e-mail: mli33@nd.edu

†e-mail: fangfei.lan@gmail.com

‡e-mail: ghweber@lbl.gov

§e-mail: beiwang@sci.utah.edu

function values, spatial positions, and correspondences.

2 RELATED WORK AND BACKGROUND

Contour Trees. Given a scalar field $f : \mathbb{M} \rightarrow \mathbb{R}$ on a manifold $\mathbb{M}$, the *level set* of an isovalue $h \in \mathbb{R}$ defines a subset of points on $\mathbb{M}$ such as $f^{-1}(h) = \{x \in \mathbb{M} \mid f(x) = h\}$. The connected components of a level set are called contours. An equivalence relation for $x, y \in M$ is defined by $x \sim y$ if and only if they belong to the same contour of f . The quotient space of $\mathbb{M}$ under this relation, $\mathbb{M}/\sim$, collapses the data to a skeletal structure, called the Reeb graph [29]. A Reeb graph without cycles (i.e., when the domain $\mathbb{M}$ is simply connected) is called a contour tree [6].

Computing a barycenter of contour trees requires a meaningful comparative measure. Defining such a measure is challenging due to the complex branching structure of contour trees. Prior approaches include edit-distance-based tree alignment, yielding a common super-tree [24], and Reeb graph distances such as the interleaving distance [11], which are often difficult to compute. We adopt an optimal-transport-based method because it provides flexible and interpretable correspondences between contour trees.

Partial optimal transport for structured data. Optimal transport computes a soft correspondence between the nodes of two contour trees by transporting mass while minimizing transportation cost. Partial optimal transport further allows nodes without good counterparts to remain unmatched, incurring a penalty [3].

Following previous works on merge trees [22], we represent a contour tree as a *measure network* $G = (V, p, W)$, where V is a finite node set, p is a probability measure on V , and $W : V \times V \rightarrow \mathbb{R}$ encodes pairwise relations between nodes. Given two networks $G_1 = (V_1, p_1, W_1)$ and $G_2 = (V_2, p_2, W_2)$, optimal transport allows mass from one node to be distributed across multiple nodes. This soft correspondence is represented by a coupling matrix $\gamma \in \mathbb{R}_+^{|V_1| \times |V_2|}$.

Classical optimal transport compares distributions based on node attributes (Wasserstein distance), and Gromov-Wasserstein (GW) compares relational structures between nodes [25]. Combining both leads to an attributed measure network $G = (V, p, W, F)$, where $F : V \rightarrow A$ maps nodes to an attribute space. Partial optimal transport allows comparison between structures with unequal total mass, accommodating unmatched or instance-specific features [3].

Existing optimal transport distances typically consider attributes, structure, or partial matching separately [25, 31, 7, 3]. To jointly account for all three, we adopt the *Fused Partial Gromov-Wasserstein* (FPGW) distance [4]. Given attributed measure networks G_1 and G_2 with node attributes a_i and b_j , FPGW is defined as

$$\begin{aligned} \text{FPGW}(G_1, G_2) = & \min_{\gamma \in \Gamma_{\leq}(p_1, p_2)} \sum_{i,j,k,l} \left[(1 - \alpha) \ell(a_i, b_j) \right. \\ & \left. + \alpha L(W_1(i, k), W_2(j, l)) \right] \gamma_{i,j} \gamma_{k,l} \\ & + \lambda \left(\|p_1^{\otimes 2} - (\gamma \mathbf{1})^{\otimes 2}\|_1 + \|p_2^{\otimes 2} - (\gamma^\top \mathbf{1})^{\otimes 2}\|_1 \right). \end{aligned}$$

where $\Gamma_{\leq}(p_1, p_2)$ denotes couplings whose row and column sums do not exceed p_1 and p_2 , allowing some mass to remain unmatched. The first term $\ell(a_i, b_j) = |a_i - b_j|^2$ measures the attribute dissimilarity. The second term $L(W_1(i, k), W_2(j, l)) = |W_1(i, k) - W_2(j, l)|^2$ captures structural discrepancy, and the last term penalizes unmatched mass. FPGW admits minimizers and defines a (semi-)metric under suitable conditions [4].

The barycenter perspective has been widely explored in optimal transport. Wasserstein barycenters were introduced as Fréchet means of probability measures [1] and later extended to structured objects through Gromov-Wasserstein (GW) barycenters, enabling averages and statistical analysis of graphs, shapes, and relational data of different sizes [26, 8]. More recently, the GW framework has been successfully applied to topological descriptors for merge

tree and Morse complex summarization [20, 21], feature tracking [22], and comparison of merge trees and Reeb graphs endowed with additional topological attributes [9, 10]. Our work builds on this line by applying FPGW barycenters to contour trees for studying ensemble variability.

Ensemble Analysis of Topological Descriptors. For a broader overview, we refer the reader to the survey by Yan et al. [33]. Research on ensemble analysis and uncertainty visualization of contour trees is relatively limited. Early work visualizes scalar field uncertainty by attaching glyphs to contour tree layouts [32]. More recently, fuzzy contour trees were proposed to align multiple contour trees into a common layout and were later extended to time-varying data [24, 23]. In contrast, merge trees have seen broader development, including structural averages with uncertainty visualization [34], Wasserstein geodesics and barycenters [27], and principal geodesic analysis [28].

Recent work [19] uses MetroSets to study collections of Morse complex skeletons by visualizing structural similarities and differences. MetroSets [16] is a set visualization technique based on a metro-map metaphor, where nodes are shown as stations and sets as colored lines.

Fuzzy contour trees align multiple contour trees into a common supertree using edit-distance-based heuristics and visualize all aligned input trees within a shared layout. In contrast, our approach computes a single representative barycenter tree by optimizing the FPGW barycenter objective, yielding a principled statistical summary of the ensemble. Rather than embedding all input trees into a common structure, we summarize the ensemble by encoding uncertainty on the nodes and edges of the barycenter tree. Moreover, the FPGW formulation naturally supports partial correspondences, allowing instance-specific features to remain unmatched instead of forcing explicit alignments within a shared topology. Finally, we adopt a MetroSets-inspired visualization to reveal structural similarities and variations across the ensemble.

3 METHOD

Encoding contour trees as attributed measure networks. To adopt the FPGW formulation and compute the barycenter for an ensemble of contour trees, we need to encode each contour tree as an attributed measure network $G = (V, p, W, F)$. We let V be the set of contour tree nodes, and $p = \mathbf{1}_{|V|} = (1, 1, \dots, 1)^\top \in \mathbb{R}^{|V|}$ so that each node has equal weight in the absence of prior knowledge of importance, following Bai et al. [4]. W encodes the shortest-path distance between nodes on contour trees, where the edge weight is the absolute value of scalar difference between adjacent nodes. The attribute function $F : V \rightarrow A$ associates each node $v_i \in V$ with a vector $a_i \in A$ as the node attribute, following

$$a_i = F(v_i) = [x_i, y_i, z_i, f_i, \text{type}_i].$$

Here, (x_i, y_i, z_i) are spatial coordinates and f_i is the scalar value of the node v_i . We design type_i as a one-hot encoding of minimum, saddle, or maximum. For example, $\text{type}_i = (1, 0, 0)$ if v_i is a minimum, or $(0, 1, 0)$ if v_i is a saddle.

We normalize both node attributes and relation matrices for all inputs to make the first two cost terms in the FPGW distance comparable. Normalization details are discussed in the supplement.

FPGW barycenter computation. We initialize a network with random pairwise relations and attributes as a seed for the barycenter network, in which the number of nodes is initialized to the largest number of nodes across all inputs. Then, we update the barycenter network by minimizing the weighted sum of FPGW distances from the barycenter to all input trees. The optimization alternates between updating couplings $\{\gamma_s\}$ and updating barycenter structure and attributes, following the barycenter solvers for FPGW [4].

Barycenter tree reconstruction. The obtained barycenter is an attributed network consisting of a relation matrix $\bar{W}$ and node at-

tributes containing coordinates, scalar values, and critical types. We consider the barycenter network as a fully connected graph, where $\bar{W}$ are the edge weights. To construct an approximated contour-tree-like graph, which we call a barycenter tree, we use a minimum spanning tree (MST)-style procedure on $\bar{W}$. In the first pass, we apply Kruskal’s algorithm [18] by sorting edges in ascending order and incrementally connecting components using a union-find structure. During this process, we exclude edges that clearly violate contour tree constraints: extrema are restricted to a single incident edge, and connections between saddles and extrema must respect scalar ordering (e.g., a saddle should not connect to a maximum with a lower function value). If the resulting graph is not connected, we perform a second pass that relaxes these constraints and continues the Kruskal procedure while still preventing cycles, ensuring a spanning tree. Finally, we simplify the structure by removing degree-2 nodes and directly connecting their neighbors, yielding a cleaner tree representation.

This reconstruction relies on the assumption that the relation matrix $\bar{W}$ preserves meaningful structural information. When the FPGW barycenter captures similar structures across input trees, pairwise relations within these structures remain coherent, allowing the underlying contour tree topology to be recovered through the MST-based procedure.

Barycenter tree visualization. We visualize the reconstructed barycenter tree using two complementary views: the bundled view and the metro view, as shown in Fig. 1. The layout follows an orthogonal contour tree drawing scheme [15], which positions nodes according to scalar values while preserving branching hierarchy. The bundled view aggregates structural information across all input trees: the thickness of each barycenter edge encodes the number of input trees that support it, highlighting dominant structural patterns. An input tree is considered to support a barycenter edge if both of the edge’s endpoints are matched to nodes in that input tree. The path between the matched nodes is then associated with the corresponding barycenter edge. To expose per-instance variations, we further apply a MetroSets-based visualization design [16] to the barycenter tree. In the metro view, each input tree is represented as a colored metro line embedded in the barycenter, and the stacking and divergence of these metro lines reveal structural differences across inputs.

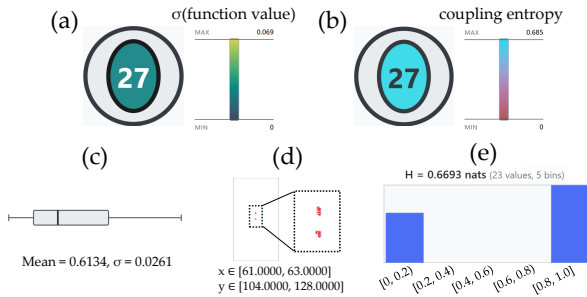


Fig. 2: Uncertainty encodings for a barycenter node. (a-b) Node glyphs and colors encoding three types of uncertainty. Node colors in (a) and (b) encode function value uncertainty and coupling uncertainty, respectively. In both, the ellipse represents coordinate uncertainty. (c-e) Detailed views for node 27. (c) Distribution of function values (boxplot). (d) Spatial distribution of matched points. (e) Histogram of binned coupling values used for entropy computation.

Node uncertainty encodings. Nodes in the barycenter summarize correspondences across multiple input trees and thus carry different types of uncertainty. Fig. 2 illustrates three complementary forms of node-level uncertainty and their visual encodings in (a-b), together with detailed statistics for an example node in (c-e). Fig. 2(a-b) are visualized in the barycenter tree views in Fig. 1, whereas Fig. 2(c-e) are only shown when a barycenter node is inquired for details; see Fig. 6(c) in the supplement for an example.

Function value uncertainty describes the variability of scalar values, measured by the standard deviation of matched critical points across inputs. This value is encoded as node color in Fig. 2(a), where the dark green color of node 27 indicates moderate variability among barycenter nodes (cf. Fig. 1(a)). The detailed distribution is shown in Fig. 2(c) using a boxplot and summary statistics.

Coordinate uncertainty reflects the spatial variation of matched nodes in the underlying domain. In Fig. 2(d), we show both the full distribution and a zoomed-in view of these positions, together with their coordinate ranges. In this example, the matched points are concentrated along a vertical region, which is captured in Fig. 2(a-b) by an ellipse elongated along the vertical axis.

Coupling uncertainty captures the consistency of correspondences across inputs. In Fig. 2(e), we show the distribution of coupling values (after binning) used to compute entropy. This uncertainty is visualized in (b) via node color. Low entropy indicates consistent presence or absence of this feature across inputs, while higher entropy reflects more ambiguous correspondence.

These uncertainty measures provide complementary insights into scalar variation, spatial dispersion, and correspondence consistency, enabling a more nuanced interpretation of the barycenter beyond a single deterministic structure.

4 EXPERIMENT AND RESULT

We evaluate our uncertainty visualization framework on two simulation datasets: Heated Cylinder and Vortex Street. Contour trees are extracted from scalar fields [2, 30, 12, 13] and used to compute FPGW barycenters, from which visualizations are generated [5]. The implementation is available at <https://github.com/mingzhefluorite/contour-tree-fpgw-barycenter>.

4.1 Heated Cylinder

The Heated Cylinder dataset [24] is an ensemble of fluid flow simulations around a heated pole with stochastic velocity perturbations. Heating produces a rising plume with complex vortex structures, where the scalar field encodes vorticity. We analyze the 23 simulations of time-varying data at time step 84. We select four instances 4, 10, 13, and 16 to include in the metro view.

Barycenter tree visualization. Figure 1 shows the approximated barycenter tree of the Heated Cylinder ensemble. Several regions exhibit structural variability across inputs, reflected by the thin edge widths in the bundled view. The dotted regions in Fig. 1(b) highlight subtrees where input trees diverge in connectivity, indicating uncertainty in the presence or arrangement of critical points and motivating further analysis. Outside these regions, most edges remain consistent across inputs, suggesting stable global structure.

Scalar field observations. Figure 3 links structural variations in the barycenter to features in the scalar fields. The zoomed-in views in Fig. 3(a-b), corresponding to the dotted regions in Fig. 1(b), show how inputs follow distinct paths through these subtrees.

We first examine variations in local maxima. The two stable maxima in Fig. 3(a) are represented by barycenter nodes 25 and 28, corresponding to the two unmarked maxima along the central thin flow in all instances in Fig. 3(c). Less stable maxima introduce variability in connectivity, giving rise to nodes 26 and 27. The maxima marked by magenta arrows in instances 4 and 13 are grouped into node 26, while those marked by light green arrows in instances 13 and 16 are grouped into node 27. The metro view makes these differences in the presence and arrangement of local maxima explicit.

We next examine variations in local minima. The orange and sky blue arrows in Fig. 3(b,c) highlight two local minima whose presence varies across inputs. In this region, input 4 contains both minima, inputs 13 and 16 each contain one, and input 10 contains none. This pattern is reflected in the metro view in Fig. 3(b): input 4 (light purple) branches to both minima 9 and 5, inputs 13 (dark purple) and 16 (yellow) connect to minimum 5 only, and input 10 (green) connects to neither.

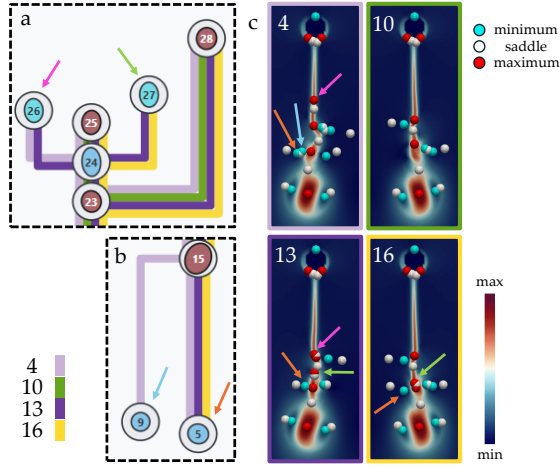


Fig. 3: (a-b) Zoomed-in subtrees (dotted regions) from the metro view of the barycenter tree in Fig. 1(b), showing connectivity for four selected inputs (4, 10, 13, and 16). (c) Corresponding scalar field regions with critical points: saddles (white), minima (cyan), and maxima (red). Magenta and light green arrows in (a) and (c) highlight variations in local maxima, while orange and sky blue arrows in (b) and (c) mark variations in local minima.

Overall, these examples show that the structural uncertainty observed in the barycenter directly reflects variations in the number and arrangement of critical points in the underlying scalar fields.

Topology violation. Our reconstruction of the barycenter network guarantees a connected, tree-like structure through MST extraction, but it does not enforce the scalar ordering constraints required of contour trees. For example, the red arrow in Fig. 1(a) highlights node 14, a saddle connected to two local minima (nodes 4 and 11), whose function value is higher than that of the neighboring saddle node 13. Within the local structure formed by nodes 4, 11, 13, and 14, node 14 has the highest function value and should therefore be a local maximum rather than a saddle.

This inconsistency stems from conflicting scalar orderings across the input trees. Nodes 13 and 14 each represent a group of matched saddle nodes, and their scalar values in the barycenter are obtained by aggregating the corresponding function values from the inputs. When the relative ordering of these saddle groups varies among the input trees, the aggregated scalar values can violate the ordering constraints of a contour tree, resulting in an invalid local configuration in the barycenter.

4.2 Vortex Street

We evaluate our approach on the time-varying Vortex Street dataset [14], from which we select 10 time steps for analysis. The dataset exhibits stable dynamics, where vortices are generated near a low-velocity region and move steadily along the flow direction. We use velocity magnitude as the scalar field.

Barycenter structure and stability. Figure 4 shows the approximated barycenter tree. The structure is highly regular, reflecting consistent vortex evolution over time. The highlighted subtrees capture persistent features in the field: the cyan box marks a sequence of local minima aligned along the central flow, while the red box contains two branching sequences of local maxima corresponding to vortex tracks on either side. These subtrees exhibit low function value uncertainty, indicating stable scalar behavior over time.

Spatial patterns and uncertainty. In this dataset, vortices form on one side of the domain and move steadily to the other side along the flow direction. In Fig. 4, the scalar field is rotated by 90° , so this motion appears vertical, although the true flow direction is horizontal. For the highlighted subtrees in the red box, ellipses for maxima and their neighboring saddles are consistently elongated along the horizontal direction, indicating coherent spatial variation

and steady translation of vortices. The ellipses for the local minima in the cyan box are also aligned with the flow direction but are less elongated, suggesting slower movement compared to the vortices.

Localized variability. Localized uncertainty is concentrated near the vortex birth region. The node indicated by the green arrow in Fig. 4 shows high function value uncertainty but low coordinate uncertainty. This corresponds to the vortex birth region, where scalar values vary despite limited spatial displacement.

Overall, the barycenter tree provides a compact summary of both the stable global structure and localized variability in the dataset.

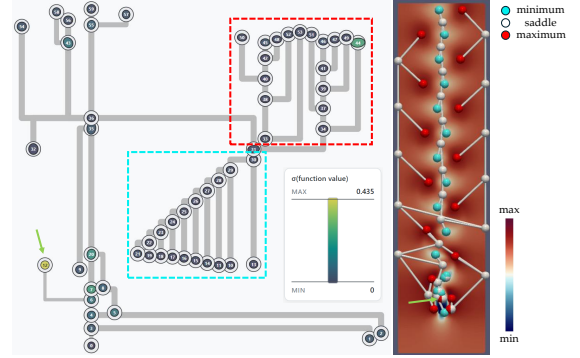


Fig. 4: Vortex Street. Left: Bundled view of the barycenter tree of 10 inputs with node colors encoding function value uncertainty. The red box shows two subtrees, each containing local maxima and neighboring saddles, corresponding to vortex tracks on either side of the central flow. The cyan box shows a cascading branch structure of local minima along the flow direction. Right: A scalar field instance (index 9, rotated 90° counterclockwise) with its contour tree. The green arrow marks a local maximum near the vortex birth region, showing high function value uncertainty and low spatial variation.

5 CONCLUSION AND DISCUSSION

We present a framework for summarizing and visualizing contour tree ensembles using approximated barycenters based on the FPGW distance. Our approach produces a representative tree that captures dominant structural patterns while accommodating variability across inputs through partial correspondences. By reconstructing a barycenter tree and leveraging coupling information, we enable integrated visualization of function value, coordinate, and coupling uncertainties. Through case studies on flow simulation datasets in both ensemble and time-varying settings, we demonstrate that the barycenter tree provides an interpretable summary of ensemble topology, revealing both stable structures and localized variations. The combination of bundled and metro views, together with node-level uncertainty encodings, supports multi-scale analysis from global patterns to detailed correspondences.

Limitations. This work has several limitations. First, visualization scalability is limited: as the number of inputs increases, the metro view becomes difficult to interpret due to overlapping metro lines, and layout readability degrades with increasing number of nodes and branches. Second, the approximated barycenter is not guaranteed to be a valid contour tree, and the FPGW formulation does not explicitly preserve branch structures. As a result, correspondences are established at the node level rather than over coherent branches.

Future work includes a comprehensive comparative study with existing methods for contour tree ensemble analysis. A long term direction is to extend the framework to broader classes of structured data such as graphs and hypergraphs.

ACKNOWLEDGMENTS

This work was partially supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research (ASCR) program under Award DE-SC0021015 to the University of Utah and under Contract No. DE-AC02-05CH11231 to Lawrence

Berkeley National Laboratory, the National Science Foundation under Awards IIS-2145499 and DMS-2301361, and a Provost's Postdoctoral Fellowship at the University of Notre Dame awarded to Mingzhe Li. ChatGPT 5.5 (OpenAI) was used solely for language editing and grammatical polishing.

REFERENCES

- [1] M. Agueh and G. Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011. doi: 10.1137/100805741 2
- [2] J. Ahrens, B. Geveci, and C. Law. ParaView: An end-user tool for large-data visualization. In *Visualization Handbook*, pp. 717–731. Butterworth-Heinemann, Burlington, 2005. doi: 10.1016/B978-012387582-2/50038-1 3, 6
- [3] Y. Bai, R. Diaz Martin, A. Kothapalli, H. Du, X. Liu, and S. Kolouri. Partial Gromov-Wasserstein metric. In *International Conference on Learning Representations*, vol. 2025, pp. 20003–20063, 2025. 2, 6, 7
- [4] Y. Bai, S. Wang, H. Tran, H. Du, J. Wang, and S. Kolouri. Fused Partial Gromov-Wasserstein for structured objects, 2025. doi: 10.48550/arXiv.2502.09934 1, 2, 7
- [5] M. Bostock, V. Ogievetsky, and J. Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011. doi: 10.1109/TVCG.2011.185 3, 6
- [6] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry Theory and Applications*, 24(2):75–94, 2003. doi: 10.1016/S0925-7721(02)00093-7 1, 2, 6
- [7] L. Chapel, M. Z. Alaya, and G. Gasso. Partial optimal transport with applications on positive-unlabeled learning. In *Advances in Neural Information Processing Systems*, vol. 33, pp. 2903–2913. Curran Associates, Inc., 2020. 2, 6
- [8] S. Chowdhury and T. Needham. Gromov-Wasserstein averaging in a Riemannian framework. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 3676–3684, 2020. doi: 10.1109/CVPRW50498.2020.00429 2
- [9] J. Curry, H. Hang, W. Mio, T. Needham, and O. B. Okutan. Decorated merge trees for persistent topology. *Journal of Applied and Computational Topology*, 6(3):371–428, 2022. doi: 10.1007/s41468-022-00089-3 2
- [10] J. Curry, W. Mio, T. Needham, O. B. Okutan, and F. Russold. Topologically attributed graphs for shape discrimination. In *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, vol. 221 of *Proceedings of Machine Learning Research*, pp. 87–101. PMLR, 28 Jul 2023. 2
- [11] V. de Silva, E. Munch, and A. Patel. Categorified Reeb graphs. *Discrete & Computational Geometry*, 55(4):854–906, 2016. doi: 10.1007/s00454-016-9763-9 2
- [12] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological Persistence and Simplification. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pp. 454–463. IEEE, 2000. doi: 10.1109/SFCS.2000.892133 3, 6
- [13] C. Gueunet, P. Fortin, J. Jomier, and J. Tierny. Task-based augmented contour trees with Fibonacci heaps. *IEEE Transactions on Parallel and Distributed Systems*, 30(8):1889–1905, 2019. doi: 10.1109/TPDS.2019.2898436 3, 6
- [14] T. Günther, M. Gross, and H. Theisel. Generic objective vortices for flow visualization. *ACM Transactions on Graphics*, 36(4):141:1–141:11, 2017. doi: 10.1145/3072959.3073684 4
- [15] C. Heine, D. Schneider, H. Carr, and G. Scheuermann. Drawing contour trees in the plane. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1599–1611, 2011. doi: 10.1109/TVCG.2010.270 3, 7
- [16] B. Jacobsen, M. Wallinger, S. Kobourov, and M. Nöllenburg. Metrosets: Visualizing sets as metro maps. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1257–1267, 2021. doi: 10.1109/TVCG.2020.3030475 1, 2, 3
- [17] M. Kraus. Visualization of uncertain contour trees. In *Proceedings of the International Conference on Imaging Theory and Applications and International Conference on Information Visualization Theory and Applications (VISIGRAPP 2010) - IVAPP*, pp. 132–139. INSTICC, SciTePress, 2010. doi: 10.5220/0002817201320139 1
- [18] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):48–50, 1956. doi: 10.2307/203241 3
- [19] F. Lan, B. Gamelin, L. Yan, J. Wang, B. Wang, and H. Guo. Topological characterization and uncertainty visualization of atmospheric rivers. *Computer Graphics Forum*, 43(3):e15084, 2024. doi: 10.1111/cgf.15084 2
- [20] M. Li, S. Palande, L. Yan, and B. Wang. Sketching merge trees for scientific visualization. In *2023 Topological Data Analysis and Visualization (TopoInVis)*, pp. 61–71, 2023. doi: 10.1109/TopoInVis60193.2023.00013 2
- [21] M. Li, C. Storm, A. Y. Li, T. Needham, and B. Wang. Comparing Morse complexes using optimal transport: An experimental study. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 41–45, 2023. doi: 10.1109/VIS54172.2023.00017 2, 7
- [22] M. Li, X. Yan, L. Yan, T. Needham, and B. Wang. Flexible and probabilistic topology tracking with partial optimal transport. *IEEE Transactions on Visualization and Computer Graphics*, 31(10):7951–7969, 2025. doi: 10.1109/TVCG.2025.3561300 2
- [23] A.-P. Lohfink, F. Gartzky, F. Wetzels, L. Vollmer, and C. Garth. Time-varying fuzzy contour trees. In *2021 IEEE Visualization Conference (VIS)*, pp. 86–90, 2021. doi: 10.1109/VIS49827.2021.9623286 2
- [24] A.-P. Lohfink, F. Wetzels, J. Lukasczyk, G. H. Weber, and C. Garth. Fuzzy contour trees: Alignment and joint layout of multiple contour trees. *Computer Graphics Forum*, 39(3):343–355, 2020. doi: 10.1111/cgf.13985 2, 3
- [25] F. Mémoi. On the use of Gromov-Hausdorff distances for shape comparison. *Eurographics Symposium on Point-Based Graphics*, pp. 81–90, 2007. doi: 10.2312/SPBG/SPBG07/081-090 2
- [26] G. Peyré, M. Cuturi, and J. Solomon. Gromov-Wasserstein averaging of kernel and distance matrices. In *Proceedings of The 33rd International Conference on Machine Learning*, vol. 48, pp. 2664–2672. PMLR, New York, Jun 2016. 2
- [27] M. Pont, J. Vidal, J. Delon, and J. Tierny. Wasserstein distances, geodesics and barycenters of merge trees. *IEEE Transactions on Visualization and Computer Graphics*, 28(1):291–301, 2022. doi: 10.1109/TVCG.2021.3114839 2
- [28] M. Pont, J. Vidal, and J. Tierny. Principal geodesic analysis of merge trees (and persistence diagrams). *IEEE Transactions on Visualization and Computer Graphics*, 29(2):1573–1589, 2023. doi: 10.1109/TVCG.2022.3215001 2
- [29] G. Reeb. Sur les points singuliers d’une forme de pfaff complètement intégrable ou d’une fonction numérique (on the singular points of a complete integral pfaff form or of a numerical function). *Comptes Rendus Academy Science Paris*, 222:847–849, 1946. 2
- [30] J. Tierny, G. Favelier, J. A. Levine, C. Gueunet, and M. Michaux. The Topology Toolkit. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):832–842, 2018. doi: 10.1109/TVCG.2017.2743938 3, 6
- [31] T. Vayer, L. Chapel, R. Flamary, R. Tavenard, and N. Courty. Optimal transport for structured data with application on graphs. In *Proceedings of the 36th International Conference on Machine Learning*, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6275–6284. PMLR, 09–15 Jun 2019. 2
- [32] K. Wu and S. Zhang. A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification*, 3:203–223, 2013. doi: 10.1615/IntJ.UncertaintyQuantification.2012003956 1, 2
- [33] L. Yan, T. B. Masood, R. Sridharamurthy, F. Rasheed, V. Natarajan, I. Hotz, and B. Wang. Scalar field comparison with topological descriptors: Properties and applications for scientific visualization. *Computer Graphics Forum*, 40(3):599–633, 2021. doi: 10.1111/cgf.14331 2
- [34] L. Yan, Y. Wang, E. Munch, E. Gasparovic, and B. Wang. A structural average of labeled merge trees for uncertainty visualization. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):832–842, 2020. doi: 10.1109/TVCG.2019.2934242 2
- [35] W. Zhang, P. K. Agarwal, and S. Mukherjee. Contour trees of uncertain terrains. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, New York, 2015. Article 43. doi: 10.1145/2820783.2820823 1

A ADDITIONAL RESULTS

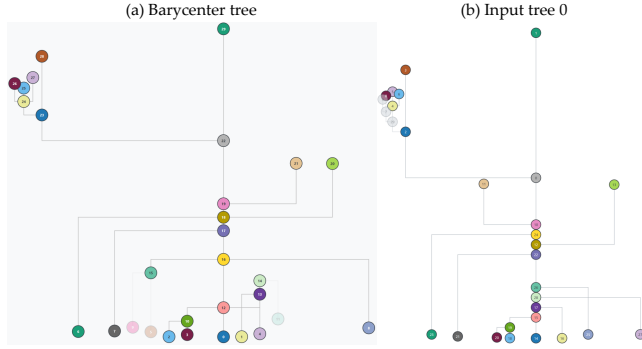


Fig. 5: Heated Cylinder: One-to-one correspondence between the barycenter tree and Input tree 0. Left: Barycenter tree with node colors used as correspondence labels; barycenter nodes unmatched to Input 0 are shown half transparent. Right: Input tree 0 colored by transporting barycenter node colors through the FPGW coupling.

A.1 Contour Tree Correspondences

The FPGW formulation establishes correspondences between the barycenter tree and each input tree. Fig. 5 visualizes this relationship by transferring colors from the barycenter to a representative input tree using the Heated Cylinder dataset. Matched nodes share the same color, while unmatched barycenter nodes appear in half transparent. This mapping enables direct comparison of structures and supports interpretation of differences as edit operations, such as insertion or deletion of features or reassignment of connectivity. The use of partial transport allows such differences to be expressed naturally without forcing artificial matches.

A.2 Interactive Visualization Interface

We implement an interactive visualization interface to explore barycenter trees and their associated uncertainty; see Fig. 6 for an example. The system consists of a Python backend for loading pre-computed data and a web-based frontend for rendering and interaction. Users can select datasets and parameters through a control panel, which dynamically updates the visualization.

The main view (Fig. 6(b)) presents the barycenter tree using the bundled or metro representations. Selecting a node reveals a detailed information panel (Fig. 6(c)), including statistics of all uncertainty measures and the corresponding matched inputs. Additional controls (Fig. 6(d)) allow users to filter input trees, switch between uncertainty encodings (e.g., function value and coupling uncertainty), and adjust layout parameters such as axis scaling. These interactions enable multi-scale exploration, from global structural patterns to detailed node-level analysis.

A.3 Discussion for Contour Tree Topology Violation

As discussed in Sec. 4.1, the topology violation arises because the relative ordering of the two saddle groups summarized by nodes 13 and 14 is inconsistent across the input trees. Fig. 7 further illustrates this phenomenon. The critical points matched to node 13 have nearly identical function values, whereas those matched to node 14 span a much wider range (cf. Fig. 7 middle and right). The aggregated scalar values assigned to the two barycenter nodes may disagree with the ordering implied by the reconstructed tree structure.

A natural question is why the optimization does not simply swap the correspondences of nodes 13 and 14 to preserve the scalar ordering. Our framework jointly considers topological and spatial information when establishing correspondences. Although swapping

the two barycenter nodes could improve the consistency of scalar ordering, it would significantly increase the spatial matching cost because the two groups of matched critical points occupy distinct spatial regions. In this case, the optimization favors spatially coherent correspondences, even when they lead to inconsistent scalar ordering. This behavior differs from purely topological matching approaches, such as Wasserstein barycenters of persistence diagrams, where spatial information is absent and scalar ordering plays a more dominant role.

Inconsistent scalar ordering may prevent the reconstructed barycenter from admitting a valid branch decomposition. The barycenter should be interpreted as a structural summary for visualization and comparison, rather than a valid contour tree. Analyses that rely on branch decomposition, such as persistence-based analysis and simplification, are no longer directly applicable.

B IMPLEMENTATION DETAILS

Our pipeline consists of three stages: pre-processing, barycenter computation, and visualization. We use ParaView [2] and TTK [30] for persistence simplification [12] and contour tree generation [13, 6]. All contour trees are precomputed from scalar fields and loaded into a Python-based analysis environment.

The FPGW framework is implemented based on the open-source PGW Python implementation¹ with edits. The optimization is initialized with multiple random barycenter networks to reduce sensitivity to local minima. After convergence, the barycenter network is converted into a contour-tree-like structure using the reconstruction procedure described earlier. Since this reconstruction is an approximation, transport couplings are recomputed between the reconstructed barycenter tree and each input tree to ensure consistency with the final structure.

The resulting data, including barycenter tree structure, node attributes, layout coordinates, and transport couplings, are exported and used for visualization. The visualization system consists of a Python backend for data processing and a D3.js-based frontend for interactive exploration [5]. The interface supports multiple views, uncertainty encodings, and node-level inspection.

B.1 Parameter Settings

We summarize the key parameters and implementation choices in the FPGW barycenter computation and visualization pipeline.

Persistence simplification. When computing contour trees, we apply persistence simplification with thresholds $\varepsilon = 3\%$ (of the global scalar range) for the Heated Cylinder dataset and $\varepsilon = 1\%$ for the Vortex Street dataset, retaining important features while removing noise.

Feature encoding. Node attributes include spatial coordinates, scalar values, and node types. Spatial coordinates are normalized by the global domain size, and scalar values are normalized by the global range across the dataset. Node types are encoded as one-hot vectors scaled by $1/\sqrt{2}$ to ensure consistent contribution to the feature distance. The attribute discrepancy is computed using Euclidean distance.

Structural encoding. The relation matrix W is defined using shortest-path distances on contour trees, where edge weights correspond to absolute differences in scalar values. All entries are normalized to $[0, 1]$ using the global maximum across the ensemble.

Transport masses. We assign uniform unit mass to all nodes, i.e., $p = 1$ for all inputs and the barycenter. This avoids introducing prior bias and allows the partial transport formulation to handle unmatched nodes naturally.

Choice of partial optimal transport. Both Chapel et al. [7] and Bai et al. [3] proposed formulations for partial Gromov-Wasserstein distance, but with different approaches. Chapel et al. introduced a

¹https://github.com/mint-vu/PGW_Metric

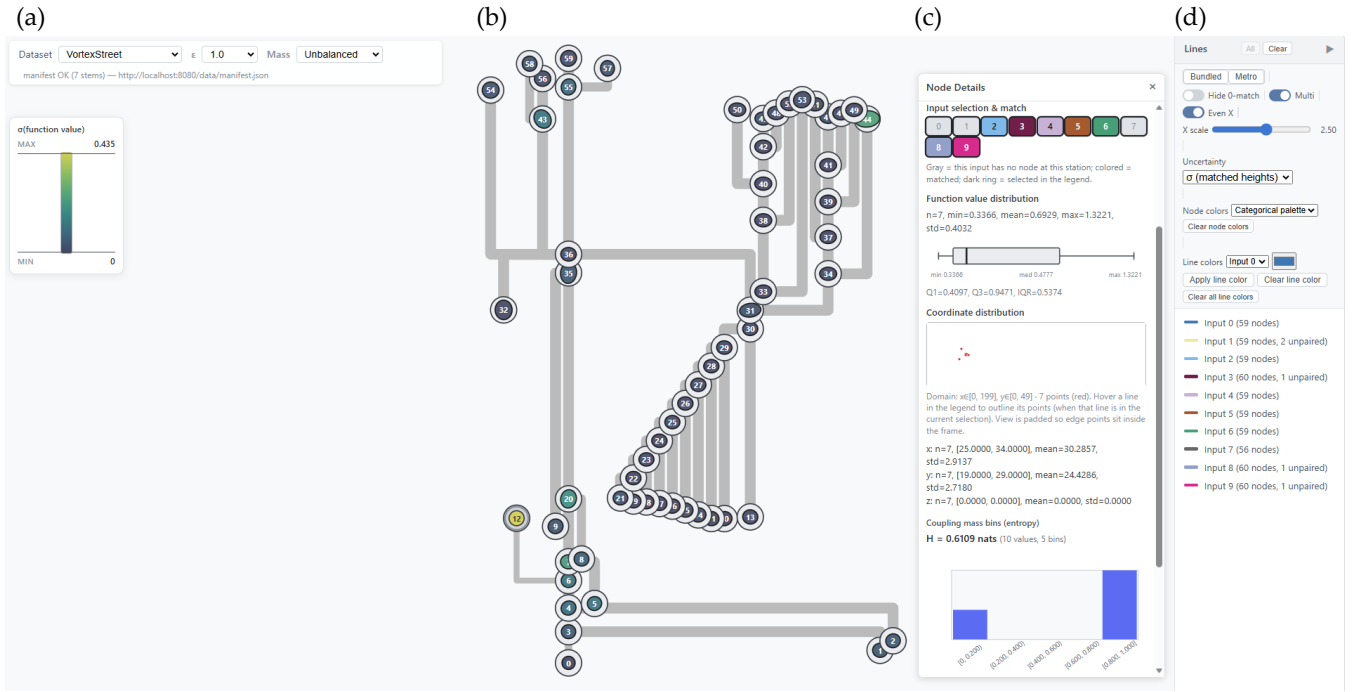


Fig. 6: Visualization interface for barycenter-based contour tree analysis. (a) Dataset and parameter selection panel with precomputed data loaded from a Python backend; the color bar shows the currently selected node-level uncertainty (function value uncertainty in this example). (b) Bundled view of the reconstructed barycenter tree for the Vortex Street dataset; the selected node (node 12) is highlighted. (c) Node information panel displaying detailed statistics of the selected node, including uncertainty measures and corresponding inputs. (d) Interaction panel providing controls for view switching, input selection, uncertainty encoding, and layout scaling.

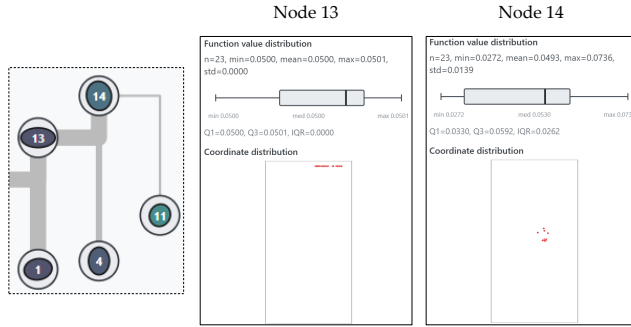


Fig. 7: Heated Cylinder: Illustration of the topology violation caused by inconsistent saddle ordering. Left: Zoomed-in view from Fig. 1(a), highlighting the neighborhood of nodes 13 and 14 in the reconstructed barycenter tree. Middle and right: Function value distributions (top) and spatial distributions (bottom) of the critical points matched to nodes 13 and 14, respectively. The saddles matched to node 13 have nearly identical function values, whereas those matched to node 14 exhibit substantially larger variation.

hyperparameter m to control the amount of mass ignored in the coupling, whereas Bai et al. formulated this via a penalty term weighted by λ .

A key distinction is that the formulation of Bai et al. defines a (semi-)metric, while that of Chapel et al. does not; see [3] for details. We adopt the formulation of Bai et al. [3, 4] due to this property, which provides a stronger theoretical foundation for barycenter interpretation.

Barycenter optimization. In prior work on Morse complexes [21],

the trade-off parameter α in the FPGW formulation is typically set to $\alpha = 0.5$ to balance structural and attribute alignment. However, we observe that structural differences encoded by shortest-path distances in contour trees can be significantly amplified compared to Morse graphs. For instance, swapping the order of two saddles along the trunk (the global minimum-maximum branch) alters shortest-path relations for entire subtrees, resulting in an $O(N)$ -scale change in the relation matrix W .

To mitigate this effect, we set $\alpha = 1/N_{max}$, where N_{max} is the maximum size of the input contour trees, thereby reducing the relative influence of structural discrepancies. The partial transport penalty λ is chosen empirically in the range $[0.4, 0.5]$. This choice is motivated by the property that mass is effectively ignored when its transport cost exceeds 2λ [4, 3]. Since both attribute vectors and relation matrices are normalized to $[0, 1]$, the total transport cost lies within $[0, \sqrt{3}]$. The selected range therefore filters out large mismatches (e.g., aligning distant extrema) while preserving meaningful correspondences.

We initialize the barycenter size as $N_{center} = N_{max}$, assuming that the barycenter does not have more nodes than the largest input tree. If reconstruction yields no degree-2 nodes to simplify or produces ambiguous node types, we increment N_{center} by two and repeat the optimization. To improve robustness, we perform 10 random initializations of the barycenter network and select the solution with the lowest total FPGW objective. Each coupling is similarly chosen from the best of 10 randomized runs.

Reconstruction and layout. The reconstructed barycenter tree is obtained using an MST-style algorithm based on the relation matrix. We adopt an orthogonal contour tree layout [15], where node positions are determined by scalar values along the vertical axis and branch structure along the horizontal axis.

To improve readability in the visualization, we apply two strate-

gies to mitigate node overlap. First, when neighboring nodes have similar scalar values, their vertical positions may become nearly identical. In such cases, we slightly shift the higher node (and all nodes above it) upward to introduce separation along the y -axis. Second, to address overlap along the x -axis, we provide an optional layout adjustment that redistributes nodes evenly according to their sorted horizontal positions. This ensures sufficient spacing between vertical branches. This functionality is exposed as the **Even X** toggle in Fig. 6(d).

Uncertainty encodings. We compute node-level uncertainties based on optimal couplings between the barycenter and input trees. For function value and coordinate uncertainty, we collect all matched input critical points whose coupling values exceed 1% of the row sum for each barycenter node and compute statistics over their scalar values and spatial coordinates.

For coupling uncertainty, we collect the corresponding coupling values themselves. If no values exceed the threshold, a single 0 is added. We quantify this uncertainty using the entropy of the collected values, computed via binning (five bins by default) over the range from 0 to the maximum coupling value in the set.

C RUNTIME PERFORMANCE

Experiments were conducted on a machine with AMD Ryzen 7 9800X3D CPU and 32GB RAM. All runtimes are measured under a multi-start strategy to improve robustness. Specifically, we perform 10 random initializations for the barycenter network and, for each FPGW computation, select the best solution from 10 randomized runs. The reported runtimes in Tab. 1 include the full cost of this procedure.

Table 1: Runtime analysis for barycenter computation. Runtimes are reported in seconds. N_{max} : the maximum size of input contour trees.

Dataset	# Inputs	N_{max}	Runtime (s)
Heated Cylinder ($\epsilon = 3\%$)	23	34	124.40 ± 8.49
Vortex Street ($\epsilon = 1\%$)	10	61	55.76 ± 4.46