# Sketching Merge Trees for Scientific Visualization 

Mingzhe Li, Sourabh Palande, Lin Yan, and Bei Wang


#### Abstract

Merge trees are a type of topological descriptors that record the connectivity among the sublevel sets of scalar fields. They are among the most widely used topological tools in visualization. In this paper, we are interested in sketching a set of merge trees using techniques from matrix sketching. That is, given a large set $\mathcal{T}$ of merge trees, we would like to find a much smaller set of basis trees $\mathcal{S}$ such that each tree in $\mathcal{T}$ can be approximately reconstructed from a linear combination of merge trees in $\mathcal{S}$. A set of high-dimensional vectors can be approximated via matrix sketching techniques such as principal component analysis and column subset selection. However, until now, there has not been any work on sketching a set of merge trees. We develop a framework for sketching a set of merge trees that combines matrix sketching with tools from optimal transport. In particular, we vectorize a set of merge trees into high-dimensional vectors while preserving their structures and structural relations. We demonstrate the applications of our framework in sketching merge trees that arise from time-varying scientific simulations. Specifically, our framework obtains a set of basis trees as representatives that capture the "modes" of physical phenomena for downstream analysis and visualization.


Index Terms-Merge trees, matrix sketching, topology in visualization, ensemble analysis

## 1 INTRODUCTION

Topological descriptors such as merge trees, contour trees, Reeb graphs, and Morse-Smale complexes serve to describe and identify characteristics associated with scalar fields, with many applications in the analysis and visualization of scientific data (e.g., see the surveys [37, 42]). Matrix sketching [69], on the other hand, is a class of mathematical tools that approximate a large data matrix with smaller and sparser matrices [24]. Principal component analysis (PCA) [54], for example, is a type of matrix sketching. We are interested in applying matrix sketching techniques to a set of topological descriptors, specifically merge trees, for scientific visualization.

We formulate our problem as follows: given a large set $\mathcal{T}$ of merge trees, we would like to find a much smaller set of basis trees $\mathcal{S}$ such that each tree in $\mathcal{T}$ can be approximately reconstructed from a linear combination of trees in $\mathcal{S}$. The set $\mathcal{T}$ may arise from a time-varying field or an ensemble of scientific simulations generated with varying parameters and/or different instruments. We aim to develop a merge tree sketching framework that:

- Identifies good representatives that capture topological variations in a set of merge trees as well as outliers; and
- Obtains a compressed representation of a large set of merge trees as a much smaller set of basis trees together with a coefficient matrix for downstream analysis and visualization.

A sketch of $\mathcal{T}$ with $\mathcal{S}$ gives rise to a significantly smaller representation of $\mathcal{T}$. Elements in $\mathcal{S}$ will serve as good representatives of $\mathcal{T}$, whereas elements with large sketching errors will be considered as outliers.

The ability to extract a basis set of merge trees is important for numerous applications, for which scientists are interested in detecting the "modes" of physical phenomena. This extraction could be achieved by computing a basis set, matching merge trees to the basis set, and computing the errors of each input tree w.r.t. that basis set. We could potentially uncover repeated phenomena that provide deep phenomenological insight. Our framework could recover cyclical phenomena for time-varying data or derive consensus sets for ensembles. Our contributions are:

- Mingzhe Li is with the University of Utah. E-mail: mingzhel@sci.utah.edu.
- Lin Yan is with the Argonne National Laboratory. E-mail: lyan@anl.gov.
- Sourabh Palande is with the Michigan State University.

E-mail: sourabh.palande@gmail.com.

- Bei Wang is with the University of Utah. E-mail: beiwang@sci.utah.edu.

Manuscript received $x x$ xxx. 201x; accepted $x x$ xxx. 201x. Date of Publication xx xxx. 201x; date of current version $x x$ xxx. 201x. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org. Digital Object Identifier: $x x . x x x x$ /TVCG.201 $x . x x x x x x x x$

- We combine tools from optimal transport with matrix sketching techniques to give a class of algorithms for sketching a set of merge trees. This is the first time matrix sketching is applied to a set of topological descriptors.
- We introduce a new distance between merge trees by adapting the Gromov-Wasserstein distance $[18,47,55]$ in optimal transport.
- We provide experimental results that demonstrate the utility of our framework in sketching merge trees that arise from scientific simulations. Specifically, we show that understanding how a set of merge trees is approximated by a smaller set can be particularly useful for the study of time-varying scalar fields and ensembles, where our framework can be used to obtain compact representations for downstream analysis and visualization. The basis set extracted from matrix sketching can serve as good representatives in detecting the modes of physical phenomena.
Our framework offers an exciting direction of utilizing randomized linear algebra for topological descriptors in visualization.


## 2 An Overview and a Primer on Matrix Sketching

Data sketching is powerful in the analysis of massive datasets [44] and has enjoyed diverse and exciting advances in recent years. A sketch is a compressed mapping of the full dataset onto a smaller data structure that serves as a summary that retains certain properties of interest. A sketch is typically "easy to update with new or changed data and allows certain queries whose results approximate queries on the full dataset." [56]


Fig. 1: The overall pipeline for sketching a set of merge trees.


Fig. 2:
: Visualizing a time-varying mixture of Gaussian functions (left) together with their corresponding merge trees (right). Merge trees at time step 4 and 7 are selected as the representatives to describe the topology of the ensemble.

We are inspired by the idea of matrix sketching. A sketch of a matrix is another matrix that is significantly smaller than , but still approximates it well [41]. Many matrix sketching techniques build upon numerical linear algebra and vector sketching. A set of highdimensional vectors is sketchable via matrix sketching techniques such as principle component analysis (PCA) and column subset selection (CSS), as illustrated in Fig. 1 (gray box).

Given a dataset of points with features, represented as a $\times$ matrix (with row-wise zero empirical mean), together with a parameter , PCA aims to find a -dimensional subspace of that minimizes the average squared distance between the points and their corresponding projections onto . For every column vector of , PCA finds a -dimensional embedding (a column vector of ) along the subspace to minimize $\|-\| \quad\|\quad-\quad\|$. is a $\times$ matrix whose columns form an orthonormal basis for . is a $\times$ coefficient matrix, whose column encodes the coefficients for approximating using the basis from . That is, $\approx$
Another technique we discuss is CSS, whose goal is to find a small subset of the columns in to form such that the projection error of to the span of the chosen columns is minimized, that is, to minimize $\|-\|\|-\|$, where we restrict to come from columns of . Such a restriction is important for data summarization, feature selection, and interpretable dimensionality reduction [8]. Thus, with PCA or CSS, given a set of high-dimensional vectors, we could find a set of basis vectors such that each input vector can be approximately reconstructed from a linear combination of the basis vectors.

Now, what if we replace a set of high-dimensional vectors by a set of objects that encode topological information of data, specifically topological descriptors? Until now, there has not been any work on sketching a set of merge trees. In this paper, we focus on merge trees, which are a type of topological descriptors that record the connectivity among the sublevel sets of scalar fields. We address the following question: Given a large set $\mathcal{T}$ of merge trees, can we find a much smaller basis set $\mathcal{S}$ as its "sketch"?

Our overall pipeline is illustrated in Fig. 1 and detailed in Sec. 6. In steps 1 and 2, given a set of merge trees $\mathcal{T} \quad\{\quad \ldots \quad\}$ as input, we represent each merge tree as a measure network and employ the Gromov-Wasserstein framework of Chowdhury and Needham [18] to map it to a column vector in the data matrix . In step 3 , we apply matrix sketching techniques, in particular, column subset selection (CSS), to obtain an approximated matrix , where
$\approx \quad \times$. In step 4, we convert each column in into a merge tree (referred to as a sketched merge tree) using minimum spanning trees (MST). Finally, in step 5, we return a set of basis merge trees $\mathcal{S}$ by applying MST to each column in . Each entry in the
coefficient matrix defines the coefficient for basis tree in approximating . With the above pipeline, given a set of merge trees, we could find a set of basis trees such that each input tree can be approximately reconstructed from a linear combination of the basis trees.

## 3 A Simple Motivational Example

Before we dive into the technical details of our approach, we give a motivational example. A time-varying scalar field is generated as a mixture of 2D Gaussian functions that translate and rotate on the plane. We sample 12 scalar fields \{
$\}$ across consecutive time steps, referred to as the Rotating Gaussian dataset, which gives rise to a set of merge trees $\mathcal{T} \quad\{\quad\}$, as shown in Fig. 2. Each merge tree is computed from - ; thus, its leaves correspond to the local maxima (red), internal nodes are saddles (white), and the root is the global minimum (blue).


Fig. 3:
: Visualizing data matrices associated with the sketching, and the coefficient matrix.

We now apply matrix sketching to $\mathcal{T}$ using our pipeline described in Fig. 1. Since the dataset is quite simple, a couple of basis trees are sufficient to obtain very good sketching results. Using , we employ Iterative Feature Selection (IFS) - a type of column subset selection algorithm - from the matrix sketching toolbox. The algorithm produces a set of two basis trees, $\mathcal{S} \quad\{\quad\}$, which are highlighted with green boxes together with their corresponding scalar fields in Fig. 2. The topological structures of these two basis trees are noticeably distinct among the input trees. They clearly serve as good representatives of the entire set $\mathcal{T}$ and capture the structural variations.

We further visualize the data matrix , , , and highlight the coefficient matrix in Fig. 3 (cf. the gray box in Fig. 1). shows that each input tree (column) is well represented (with high coefficient) by one of the two basis trees. In particular, columns in the coefficient matrix with high (yellow or light green) coefficients (w.r.t. the given basis) may be
grouped together, forming two clusters \{ and $\{\quad\}$ whose elements look structurally similar.

## 4 Related Work

Merge trees record the connectivity among the sublevel sets of scalar fields (e.g., [6,14]). They are rooted in Morse theory [52], which characterizes scalar field data by the topological changes in its sublevel sets at isolated critical points. A number of recent works focus on comparing merge trees and their variants (e.g., [6, $31,57,60,67,68]$ ); see [72] for a survey. Recently, Pont et al. proposed a Wasserstein distance between merge trees [57] that equals to the
-Wasserstein distance between persistence diagrams. Wetzels et al. proposed variants of edit distances $[67,68]$ that are independent from branch decomposition trees.

In this paper, we treat merge trees as measure networks and introduce a Gromov-Wasserstein (GW) distance between merge trees based on optimal transport (Sec. 5). See Appendix E for a comparison with the Wasserstein distance [57]. Different from previous distances between merge trees, the GW distance is easy and efficient to compute, and provides explicit structural correspondences between the trees (Sec. 6). Our main focus is to use the GW distance to obtain alignments and vector representations of merge trees that interface with matrix sketching. Gromov introduced GromovHausdorff (GH) distances [34] while presenting a systematic treatment of metric invariants for Riemannian manifolds. GH distances can be employed as a tool for shape matching and comparison (e.g., $[11,45,46,49,50]$ ), where shapes are treated as metric spaces, and two shapes are considered equal if they are isometric. Memoli [47] modified the formulation of GH distances by introducing a relaxed notion of proximity between objects, thus generalizing GH distances to the notion of Gromov-Wasserstein (GW) distances for practical considerations. Since then, GW distances have had a number of variants based on optimal transport [62,63] and measure-preserving mappings [48]. Apart from theoretical explorations [47,61], GW distances have been utilized in the study of graphs and networks [38,70,71], machine learning [12,28], and word embeddings [4]. Recently, Memoli et al. [51] considered the problem of approximating metric spaces using GW distance. Their goal was to approximate a (single) metric measure space modeling the underlying data by a smaller metric measure space. The work presented in this paper instead focuses on approximating a large set of merge trees - modeled as a set of metric measure networks - with a much smaller set of merge trees.

Graph alignment or graph matching is a key ingredient in performing comparisons and statistical analysis on the space of graphs (e.g., [27,35]). It is often needed to establish node correspondences between graphs of different sizes. Edit distances have been used to align contour trees [43]. The approaches that are most relevant here are the ones based on the GW distances [18,55], which employ probabilistic matching ("soft matching") of nodes. Information in a graph can be captured by a symmetric positive semidefinite matrix that encodes distances or similarities between pairs of nodes. Dryden et al. [25] described a way to perform statistical analysis and to compute the mean of such matrices. Agueh et al. [3] considered barycenters of several probability measures, whereas Cuturi et al. [20] and Benamou et al. [7] developed efficient algorithms to compute such barycenters. Peyre et al. [55] combined these ideas with the notion of GW distances [47] to develop GW averaging of distance/similarity matrices. Chowdhury and Needham [18] built upon the work in [55] and provided a GW framework to compute a Frechét mean among these matrices using measure couplings. In this paper, we utilize the GW framework [18] for probabilistic matching among merge trees.

A number of recent works transform topological descriptors from data into feature vectors to be used as input to machine learning models; see [39] for a survey. A primary focus is on vectorizing persistence diagrams. Adams et al. introduced persistence images [2] that transform persistence diagrams into 2D images for classification tasks. Carrière et al. [16] used mappings between points in persistence diagrams to construct vector representations. A neural network layer was also used to embed persistence diagrams in
vector spaces [15].
Our framework generates vectorized representations of merge trees using optimal transport to be interfaced with matrix sketching. Different from previous work, the vectorized merge trees preserve structural correspondences and there exist explicit mappings between pairs of merge trees. In addition, we can reconstruct input merge trees from basis trees, that is, we can reverse engineer merge trees from their vector representations. A number of previous works also utilize the latent representations of inputs from neuron networks as high-dimensional vector representations. However, these approaches often require extensive training and often do not generalize well across diverse datasets. In comparison, our approach is generalizable and does not require training.

Many matrix sketching techniques $[56,69]$ build upon linear algebra and vector sketching. For simplicity, we formulate the problem as follows: Given a $\times$ matrix , we would like to approximate using fewer columns, as a $\times$ matrix such that and are considered to be close with respect to some problem of interest. Basic approaches for matrix sketching include truncated singular value decomposition (SVD), column or row sampling [22,23], random projection [59], and frequent directions [33,41]; see [56,69] for surveys.

The column sampling approach carefully chooses a subset of the columns of proportional to their importance, where the importance is determined by the squared norm (e.g., [22]) or the (approximated) leverage scores (e.g., [23]). The random projection approach takes advantage of the Johnson-Lindenstrauss (JL) Lemma [40] to create an
$\times$ linear projection matrix (e.g., [59]), where . The frequent directions approach [33,41] focuses on replicating properties of the SVD. The algorithm processes each column of at a time while maintaining the best rank- approximation as the sketch.

## 5 Technical Background

We begin by reviewing the notion of a merge tree that arises from a scalar field. We then introduce the technical background needed to vectorize a merge tree as a column vector in the data matrix.


Fig. 4: An example of a merge tree from a height function. From left to right: 2D scalar field visualization, a merge tree embedded in the graph of the scalar field, and an abstract visualization of a merge tree as a rooted tree equipped with a height function.

Let $\rightarrow$ be a scalar field defined on the domain of interest , where is a subset of in our context. Merge trees capture the connectivity among the sublevel sets of , i.e.,
$-\infty$. Two points $\underset{\text { and }}{\in}$ are equivalent, denoted by $\sim$ belong to the same connected if , and and belong to the same connected component of a sublevel set . The merge tree,
is the quotient space obtained by gluing together points in that are equivalent under the relation $\sim$.

To construct a merge tree, we sweep the function value from $-\infty$ to $\infty$, and create a new branch originating at a leaf node for each local minimum of . As increases, such a branch is extended as its corresponding component in grows until it merges with another branch at a saddle point. If is connected, all branches eventually merge into a single component at the global maximum of , which corresponds to the root of the tree. For a given merge tree, leaves, internal nodes, and root node represent the minima, merging saddles, and global maximum of , respectively. Fig. 4 displays an example. Abstractly, a merge tree is a rooted tree equipped with a scalar function defined on its node set,

Our frame-
work utilizes tools from optimal transport, specifically, the GW distance
between measure networks. The GW distance was proposed by Memoli [46,47] for metric measure spaces. Peyre et al. [55] introduced the notion of a measure network and defined the GW distance between such networks. The key idea is to find a probabilistic matching between a pair of networks by searching over the convex set of couplings of the probability measures defined on the networks.

In our context, a finite merge tree can be represented as a measure network using a triple , where is the set of nodes, is a probability measure on , and is an $\times$ matrix capturing the relations between pairs of nodes. For our experiments, is taken to be uniform, that is, - where $\in \quad$ may encode adjacency or shortest path relations (see Sec. 6).
Let
with $\begin{gathered}\text { and } \\ \text { and }\end{gathered} \quad$ be a pair of merge trees with and nodes, respectively. Let denote the set \{ \}. $\}$ and $\}$. A coupling between probability measures and is a joint probability measure on $\times$ whose marginals agree with and . That is, a coupling is represented as an $\times$ non-negative matrix such that each row sums up to and each column sums up to . The distortion of a coupling with an arbitrary loss function is defined as [55]

$$
\begin{equation*}
\mathcal{E} \tag{1}
\end{equation*}
$$

Let $\mathcal{C} \quad \mathcal{C} \quad$ denote the collection of all couplings between and . The Gromov-Wasserstein discrepancy $[55]$ is defined as

$$
\begin{equation*}
\mathcal{D} \quad \mathcal{E} \tag{2}
\end{equation*}
$$

In this paper, we consider the quadratic loss function

and is defined as

$$
\begin{aligned}
& - \\
& |\quad-\quad|
\end{aligned}
$$

It follows from the work of Sturm [61] that such minimizers always exist and are referred to as optimal couplings.

$$
\begin{array}{lll}
T_{1} & T_{2} & \text { A }
\end{array}
$$

$\bar{T}$


Fig. 5: An optimal coupling between two merge trees and . The coupling matrix is visualized in (A): yellow means high and dark blue means low probability. Couplings between the Fréchet mean ${ }^{-}$with and are shown in (B) and (C), respectively.

We give a simple example involving a pair of merge trees in Fig. 5 (top). and contain 8 and 6 nodes, respectively, where nodes are
labeled starting with a index. The optimal coupling is shown below and visualized in Fig. 5 (A):
is an $\times$ matrix, and it shows, for instance, that node 0 in is matched to node 0 in with the highest probability. Node 2 in is matched probabilistically with both node 0 and node 1 in
. Given a pair of merge trees
and with and nodes, respectively, a coupling $\in \mathcal{C} \quad$ can be used to align their nodes. In order to do this, we will need to increase the size of and appropriately into their respective blowup trees and , such that and contain the same number of nodes (where $\leq$ ).
Roughly speaking, let be a node in , and let be the number of nodes in that have a nonzero coupling probability with . The blowup tree is created by making copies of node for each node in , generating a new node set . The probability distribution and the weight matrix are updated from and accordingly. Similarly, we can construct the blowup
of . An optimal coupling between and expands naturally to a coupling between and . After taking appropriate blowups, is now an $\times$ matrix which can be used to align the nodes of the two blowup trees. With a bijective node alignment, we can permute to be a diagonal matrix whose marginals agree with and respectively. Since is a diagonal matrix, we have . Finally,
can be binarized to be an $\times$ permutation matrix (e.g., it has
where , and elsewhere). The GW distance is given by a formulation equivalent to Eqn. (3) based on an optimal coupling, following [18, Definition 2]:

$$
\begin{equation*}
-\quad|\quad-\quad| \tag{4}
\end{equation*}
$$

Given a collection of merge trees $\mathcal{T}$ \{
a Fréchet mean $^{-}$of $\mathcal{T}$ is a minimizer of the functional $\mathcal{T}$ over the space $\mathcal{N}$ of measure networks [18],

Chowdhury and Needham [18] defined the directional derivative and the gradient of the functional $\mathcal{T}$ at and provided a gradient descent algorithm to compute the Fréchet mean. Their iterative optimization begins with an initial guess of the Fréchet mean. At the iteration, there is a two-step process: each is first blown up and aligned to the current Fréchet mean, ; then is updated using the gradient of the functional $\quad \mathcal{T}$ at . Such a two-step process is repeated until convergence where the gradient vanishes. For the complete algorithmic and implementational details, see [18]. If ${ }^{-}-$is $^{-}$ the Fréchet mean, then we have - $\quad-\quad$ where
is the weight matrix obtained by blowing up and aligning $\quad \in \mathcal{T}$ to ${ }^{-}$. That is, when all trees in $\mathcal{T}$ are blown up and aligned to ${ }^{-}$, the weight matrix of ${ }^{-}$is given by a simple elementwise average of the weight matrices of the merge trees.

In the example shown in Fig. 5 (bottom), we compute the Fréchet mean $^{-}$of and, which has 12 nodes. We align both and to ${ }^{-}$via their blowup trees. This alignment gives rise to a coupling matrix between_ and (of size $\times$ ) in Fig. 5 (B), and a coupling matrix between ${ }^{-}$and (of size $\times$) in Fig. 5 (C), respectively. As shown in Fig. 5, root node 0 of is matched probabilistically with root
node 0 of and root node 0 of . Nodes 2 and 7 of ${ }^{-}$are matched probabilistically with node 1 in . Now both and are blown up to be and , each with 12 nodes, and can be vectorized into column vectors of the same size.

## 6 Methods

Given a set $\mathcal{T}$ of merge trees as input, our goal is to find a basis set $\mathcal{S}$ with $\ll$ merge trees such that each tree in $\mathcal{T}$ can be approximately reconstructed from a linear combination of merge trees in $\mathcal{S}$. We propose to combine the GW framework [18] with techniques from matrix sketching to achieve this goal. We detail our pipeline to compute $\mathcal{S}$, as illustrated in Fig. 1.

The first
step is to represent merge trees as measure networks, as described in Sec. 5. Each merge tree $\in \mathcal{T}$ can be represented using a triple In this paper, we define as a uniform distribution on , and as a shortest path distance matrix.

Recall that each node in a merge tree is associated with a scalar value . For a pair of nodes $\in$, if they are adjacent, we define $\quad|\quad-\quad|$, i.e., their absolute difference in function value; otherwise, is the shortest path distance between them in . By construction, the shortest path between two nodes goes through their lowest common ancestor in . The node set of a merge tree is equipped with a function ; therefore, we define in such a way to encode information from

The second step is to convert each merge tree into a column vector of the same size via blowup and alignment to the Fréchet mean.

Having represented each merge tree as a measure network, we can use the GW framework to compute a Fréchet mean of $\mathcal{T}$, denoted as $-\quad-$. Let $\left.\right|^{-} \mid$. In theory, may become as large as $\quad|\quad|$. In practice, is chosen to be much smaller. In our experiment, we choose to be a small constant factor (e.g., 2 or 3 ) times the size of the largest input tree. The optimal coupling between and is an $\times$ matrix with at least nonzero entries. If the number of nonzero entries in each row is greater than , we keep only the largest value. That is, if a node of ${ }^{-}$has a nonzero probability of coupling with more than one node of , we consider the mapping with only the highest probability, so that each coupling matrix has exactly nonzero entries. We then blow up each to obtain
and align ${ }^{-}$with . The above procedure ensures that each blowup tree has exactly nodes, and the binarized coupling matrix between ${ }^{-}$and induces a node matching between them.

We can now vectorize (i.e., flatten) each (an $\times$ matrix) to form a column vector $\in$ of matrix (where ), as illustrated in Fig. 1 (step 2). In practice,
as we store only the upper triangular matrix. Each is a vector representation of the input tree w.r.t. the Fréchet mean ${ }^{-}$. Different from previous vectorization techniques, this process preserves (interpretable) structural correspondences between the vector and the tree.

The third step is to sketch merge trees by applying matrix sketching to the data matrix , as illustrated in Fig. 1 (step 3). By construction, is a $\times$ matrix whose column vectors are vector representations of . We apply matrix sketching techniques to approximate by $\times$. In our experiments, we use two linear sketching techniques from column subset selection (CSS). See Appendix B for implementation details.

Using CSS, the basis set is formed by sampling columns of Let denote the matrix formed by columns of and let denote the projection onto the -dimensional space spanned by the columns of . The goal of CSS is to find such that \| - \| is minimized. We experiment with two variants of CSS.

In the first variant of CSS, referred to as Length Squared Sampling (LSS), we sample (without replacement) columns of with probabilities proportional to the square of their Euclidean norms, i.e.,
|| || || || . We modify the algorithm slightly such that before selecting a new column, we factor out the effects from columns that are already chosen, making the chosen basis as orthogonal as possible.

In the second variant of CSS, referred to as the Iterative Feature Selection (IFS), we use the algorithm proposed by Ordozgoiti et al. [53]. Instead of selecting columns sequentially as in LSS, IFS starts with a random subset of columns. Then each selected column is either kept or replaced with another column, based on the residual after the other selected columns are factored out simultaneously.

For the fourth step,
we convert each column in as a sketched merge tree. Let , where matrices and are obtained using CSS. Let denote the column of . We reshape as an $\times$ weight matrix . We then obtain a tree structure from by computing its minimal spanning tree (MST). In particular, we treat as a pair-wise distance matrix, and the MST constructed from connects all the nodes and minimizes the sum of edge weights.

Finally, we return a set of basis merge trees $\mathcal{S}$ using information encoded in the matrix . Using CSS, each column of corresponds directly to a column in ; therefore, the set $\mathcal{S}$ is trivially formed by the corresponding merge trees from $\mathcal{T}$.

For each experiment, we compute the global sketch error \| - \| , as well as column-wise sketch error \| $\|$, where $\quad$. By construction, $\leq$. For merge trees, we measure the GW distance between each tree and its sketched version , that is
, referred to as the column-wise
GW loss. The global GW loss is defined to be
. For
theoretical considerations, see discussions in Appendix A.

## 7 Experimental Results

We demonstrate the applications of our sketching framework with merge trees that arise from three 2D and one 3D time-varying datasets from scientific simulations. The key takeaway is that, by applying matrix sketching, a large set $\mathcal{T}$ of merge trees is replaced by a much smaller basis set $\mathcal{S}$ such that trees in $\mathcal{T}$ are well approximated by trees in $\mathcal{S}$. Elements in the basis set $\mathcal{S}$ serve as good representatives that capture structural variations among the time instances, thus reflecting the "modes" of the underlying physical phenomena (Sec. 7.1 and Sec. 7.2). In addition, our framework also uncovers cyclical behavior of timevarying datasets that exhibit periodicity (Sec. 7.3 and Appendix C.1). See Appendix C for additional results and runtime analysis.

In practice, we simplify the scalar fields based on persistence before computing the merge trees. See Appendix B for details.

### 7.1 Heated Cylinder Dataset

Two of our datasets come from numerical simulations available online [1]. The first dataset, referred to as the Heated Cylinder with Boussinesq Approximation (Heated Cylinder in short), comes from the simulation of a 2D flow generated by a heated cylinder using the Boussinesq approximation $[36,58]$. The dataset shows a time-varying turbulent plume containing numerous small vortices. We convert each time instance of the flow (a vector field) into a scalar field using the magnitude of its vertical (y) velocity component. We generate a set of split trees (i.e., the merge tree surrounding local maxima) from these scalar fields based on 31 time steps, which correspond to steps 600-630 from the original 2000 time steps. This set captures the evolution of small vortices over time.

To choose the appropriate number of basis trees for this dataset, we use the "elbow method" to determine , similar to cluster analysis. We plot the global GW loss and global sketch error as a function of , and pick the elbow of the curve as the to use. As shown in Fig. 6, is chosen to be three for the Heated Cylinder dataset. In subsequent sections, element-wise GW losses and sketch errors also reaffirm this choice (cf., Fig. 8).

Given 31 merge trees $\mathcal{T} \quad\{$
\} from the Heated Cylinder dataset, we apply two types of column subset selection (CSS) methods, namely IFS and LSS to obtain a set of basis trees $\mathcal{S}$ and reconstruct the sketched trees. Since we are using CSS, the basis trees are elements from the original input. We first demonstrate that the basis trees capture structural variations among the time-varying input. We then investigate

Fig. 6:
: Global GW losses and global sketch errors for varying , the number of basis trees, using IFS and LSS.


Fig. 7: Sketching the
dataset with three basis trees using IFS: (A) basis trees where orange circles highlight topological changes w.r.t. nearby basis trees, (B) scalar fields that give rise to these basis trees. Areas with critical points appearances/disappearances are shown with zoomed views in (C).
the coefficient matrix and show that with only three basis trees, we can obtain sketched trees with small errors.

We first illustrate our sketching results using IFS. Based on our error analysis (Fig. 6), three basis trees appear to be the appropriate choice that strikes a balance between data summarization and structural preservation.

As shown in Fig. 7(A), IFS produces three basis trees, $\mathcal{S}$ \{ \}, which capture noticeable structural variations among the input merge trees. Specifically, moving from to and to , a saddle-maxima pair appears in the merge trees, respectively (highlighted by orange circles). These changes in the basis trees reflect the appearances of critical points in the domain of the time-varying fields; see Fig. 7(B). In Fig. 7(C), we highlight (with orange balls) the appearances of these critical points in the domain. That is, from to
, critical points and appear in the scalar fields, whereas from
to , critical points and appear. Therefore, the three basis trees capture structural changes in the time-varying data, thus reflecting the "modes" of the underlying phenomena. Such "modes" are also confirmed with the coefficient matrix (see Fig. 8), which is a byproduct of the sketching process.

The coefficient matrix, column-wise
sketch error, and GW loss are used to guide our investigation into the quality of individual sketched trees, see Fig. 8. Trees with small GW losses or sketch errors are considered well sketched w.r.t. the chosen basis. The coefficient matrix in Fig. 8(B) contains a number of yellow or light green blocks, indicating that consecutive input trees share similar coefficients w.r.t. the chosen basis; therefore, they are grouped together into three clusters, reflecting the three modes of the underlying phenomena. Such a blocked structure indicates that the chosen basis trees are good representatives of the clusters.

In comparison, using just two basis trees ( and ) does not capture the structural variations as well as three basis trees. In Fig. 8(C), we see a slight degradation in the blocked structure and thus the sketching quality using two basis trees. In particular, trees in the red area of Fig. 8(D) (to ) are not well approximated due to a missing basis tree.


Fig. 8: Sketching the (C-D) basis trees using IFS. (A, C) column-wis loss, (B, D) coefficient matrix. Orange boxes highlight basis trees.
LSS
Coe! cient Matrix


Fig. 9: sketch the dataset with three basis trees using LSS.

Fig. 10: $\quad:$ Global GW losses and global sketch errors for varying , the number of basis trees, using IFS and LSS.

Additionally, we include the sketching results using LSS as an alternative strategy, again with three basis trees according to the "elbow method" (Fig. 6). LSS gives basis trees and
in Fig. 9, which are similar to the ones obtained by IFS (Fig. 7). In other words, for the Heated Cylinder dataset, variations in column selection methods do not affect the quality of sketching results.

