Hyperdimensional Bayesian Time-Mapping (HyperBaT): A Probabilistic Approach to Time Series Mapping of Non-Identical Sequences

Macey Ruble, Ethan Hayes, Matt Welborn, Peter D. Anderson, Alenka Zajić, Milos Prvulovic, Ann M. Pitruzzello

***Abstract*—A common problem in time series analysis is mapping the related elements between two sequences as they progress in time. Methods such as dynamic time warping (DTW) and hidden Markov models (HMM) have good performance in mapping time series signals with repeated (warped) elements relative to a reference signal. However, there is not an adequate method for mapping time series signals with inserted or deleted elements. This work introduces hyperdimensional Bayesian time-mapping (HyperBaT), a machine learning algorithm that maps two time sequence signals that may contain inserted, deleted, or warped elements. Additionally, HyperBaT estimates the common underlying signal shared between the two sequences. The algorithm is presented in a general context so that it can be used in a variety of applications. There are many relevant areas, including speech processing, genetic sequencing, electronic warfare, communications, and radar processing, that process signals containing inserted or deleted elements. As an example, HyperBaT is applied to side-channels where it maps radio frequency (RF) side-channel signals emitted from a computing device processor, which can be used to track control flow execution and monitor for malicious activity.**

***Index Terms*—HMM, DTW, Time Series, Side Channel**

I. Introduction

Comparison of time series sequences is a common task for many applications. Often, it is desired to map the shared elements in two related sequences as they progress in time. This is particularly challenging when the sequences are non-identical because of inserted or deleted elements. A simple visualization of the non-identical time sequence mapping problem with one inserted element can be seen in Fig. 1, where elements in the sequence are visualized as symbols. Mapping in this scenario requires linking elements in a noisy measured signal to elements in a compared signal. The mapped elements in Fig. 1 are connected with lines and the inserted element is identified.

Straightforward distance metrics such as the Euclidean distance are frequently used for time-mapping, but these are sensitive to distortions in the time axis and have difficulty when sequences have different lengths [12], [6]. The dynamic time warping (DTW) algorithm has been developed to map signals of different lengths by allowing a nonlinear alignment of sequences along the time axis [11], [18]. DTW was originally developed in the 1960s [4], [19] and first gained wide usage in speech processing [20], [15]. With DTW, the sequence mapping is equivalent to finding an optimal path through a two dimensional grid. Hidden Markov models (HMM) extend DTW to include a metric based on the stochastic nature of the signals [11], [16].

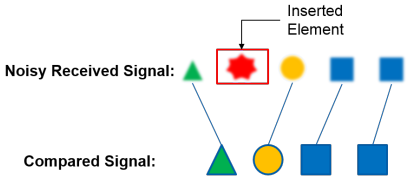

Fig. 1: A symbolic representation of the non-identical time series mapping problem. In this case, a noisy measured signal with an inserted element is mapped to the compared reference signal.

Both DTW and HMMs perform well with signals that have been warped or stretched in time with repeated elements. However, these algorithms are not well suited to map sequences that contain additional inserted segments of unknown length or structure. Previous efforts have attempted to implement insert capabilities with DTW and HMMs [18], but only for very short inserts. A limitation of DTW and HMM methods is that computational complexity grows with the size of allowable inserts [18]. Other methods, such as the segmental DTW and unbounded DTW, seek to address the problem of inserts by matching smaller segments or "keywords" within a longer sequence [3], [14]; however, this requires significant prior knowledge about the expected locations of inserts.

This paper introduces hyperdimensional Bayesian time-mapping (HyperBaT), a novel machine learning algorithm designed to map time series sequences that contain inserted, deleted, and warped segments. HyperBaT uses a three dimensional mapping, rather than the two dimensional mapping of DTW and HMM methods, to capture inserts and deletions in addition to warps. HyperBaT exploits signal statistics and chooses a mapping to satisfy an optimality criterion based on probabilities, which is used to optimally map signals. Unlike DTW and HMM methods, HyperBaT is able to capture insertions and deletions of an arbitrary length without additional complexity. Thus, HyperBaT computation is on the same order as the standard implementations of DTW and HMM methods with $\mathcal{O}(mn)$, where m and n are the lengths of the two sequences. Because HyperBaT outputs probabilities, it can easily be used with machine learning algorithms, such as hidden Markov models (HMM), to model larger scale temporal dynamics.

This work formulates HyperBaT in a general manner so that it can be applied to a variety of applications. While

this work focuses on one-dimensional real signals, HyperBaT can be extended to multi-dimensional signals. As a real world application, HyperBaT is applied to radio frequency (RF) side-channel measurements of activity from computing device processors. While this paper focuses on side-channel measurements of processor activity, similar challenges exist in applications that require separation of inserted or modified segments and identification of the common shared elements. As one example, HyperBaT could be used to align DNA, RNA, or protein sequences with templates for sequence classification, similarity analysis, or identification of point mutations. Speech processing is another possible application of HyperBaT as different dialects or pronunciations can lead to different forms of the same word, which has been a historically challenging problem to overcome [20]. Other applications include electronic warfare, communications, radar processing, finance, and data mining.

This work is organized as follows. Section II mathematically formulates the problem of non-identical time sequence mapping in a general manner. Section III introduces the HyperBaT algorithm. Then, Section IV studies HyperBaT performance and provides real-world examples from side-channel computing processor measurements. Finally, Section V provides concluding remarks.

II. Problem Formulation

The problem of non-identical time sequence mapping is presented in this section. Discrete sequences are considered, which may be sampled from continuous signals. Without loss of generality, we assume the sequence elements to be real signals.

A. Signal Model

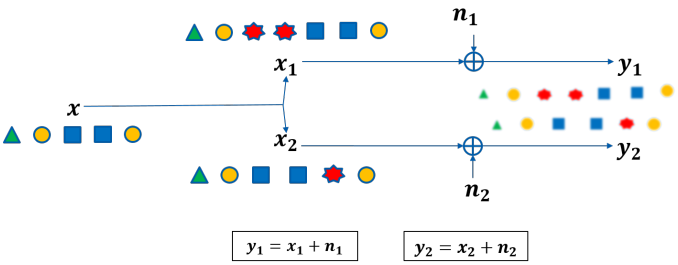

Fig. 2: General model of two sequences that share a common subset of elements.

Let the sequence $\boldsymbol{x} \in \mathbb{R}^{T\times 1}$ be a vector with T elements where $\boldsymbol{x}(i) \sim \mathcal{N}(0, \sigma_x^2)$ for $i = 1, \ldots, T$. The sequences $\boldsymbol{x_1} \in \mathbb{R}^{N_1\times 1}$ and $\boldsymbol{x_2} \in \mathbb{R}^{N_2\times 1}$ are formed by adding additional elements to the vector $\boldsymbol{x}$ to obtain sequences of length N_1 and N_2 respectively. Both $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ contain the full truth sequence $\boldsymbol{x}$ so that the truth sequence is a subset of both sequences $\{\boldsymbol{x}(i)\}_{i=1}^{T} \subseteq \{\boldsymbol{x_1}(i)\}_{i=1}^{N_1}, \{\boldsymbol{x_2}(i)\}_{i=1}^{N_2}$ where the ordering of $\boldsymbol{x}$ is maintained. The sequence $\boldsymbol{x}$ will be referred to as the common subset sequence as it is a subset of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$. Then, noisy measurements of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are observed to give

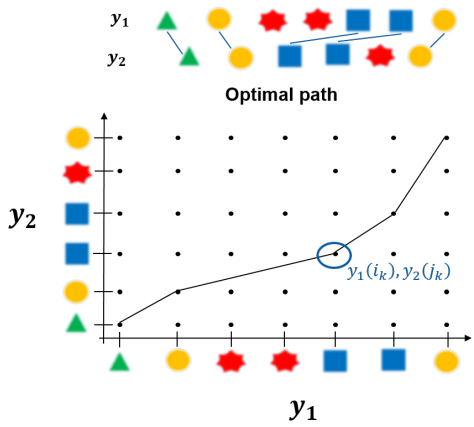

Fig. 3: Correct mapping of the sequences created in Fig. 2.

$$\boldsymbol{y_1} = \boldsymbol{x_1} + \boldsymbol{n_1}, \tag{1a}$$
$$\boldsymbol{y_2} = \boldsymbol{x_2} + \boldsymbol{n_2}, \tag{1b}$$

where $\boldsymbol{y_1}, \boldsymbol{n_1} \in \mathbb{R}^{N_1\times 1}$, $\boldsymbol{y_2}, \boldsymbol{n_2} \in \mathbb{R}^{N_2\times 1}$ and the noise $\boldsymbol{n_1}(i), \boldsymbol{n_2}(j) \sim \mathcal{N}(0, \sigma_n^2)$ for $i = 1, \ldots, N_1$ and $j = 1, \ldots, N_2$. Note that a deletion in one sequence is effectively an insertion into the compared sequence, so deletions are not modeled separately. It should also be noted that the noise terms $\boldsymbol{n_1}$ and $\boldsymbol{n_2}$ are assumed to have the same variance. Additionally, for brevity, the signals $\boldsymbol{x}, \boldsymbol{x_1}, \boldsymbol{x_2}$ are assumed to have the same variance, but the extension to different variances is straightforward. Fig. 2 has a visual representation of the model where the signal is generalized so that elements are represented by shapes.

The premise of this work is that a mapping exists that connects the elements of $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ with the common subset sequence $\boldsymbol{x}$. For a mapping index k,

$$\boldsymbol{x}(k) = \boldsymbol{x_1}(i_k) = \boldsymbol{x_2}(j_k), \qquad k = 1, 2, ..., T, \tag{2}$$

where k is the index of the common signal $\boldsymbol{x}$ that is mapped to the i_k index of $\boldsymbol{x_1}$ and the j_k index of $\boldsymbol{x_2}$. Again, it is noted that the chronological order of $\boldsymbol{x}$ is maintained in $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ so that $i_{k+1} \geq i_k$ and $j_{k+1} \geq j_k$ for all k. The sequence indices i_k and j_k from (2) are defined by a mapping function $\phi(k)$, which is the mapping of the k^{th} element of the common subset sequence to the observations such that [18],

$$i_k = \phi_{y_1}(k), \qquad k = 1, 2, \cdots, T, \tag{3a}$$
$$j_k = \phi_{y_2}(k), \qquad k = 1, 2, \cdots, T. \tag{3b}$$

The mapping is equivalent to a particular path through a two-dimensional grid where the sequences $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ are on orthogonal axes. Then, each pair of elements that are mapped to each other correspond to a point in the path through the grid. Many different mappings exist, each with a different path through the grid. A mapping algorithm aims to choose the optimal mapping that correctly links the sequence elements. Fig. 3 shows the correct mapping for the sequences in Fig. 2.

HyperBaT solves the following problem: Given the measurement sequences $\boldsymbol{y_1} \in \mathbb{R}^{N_1 \times 1}$ and $\boldsymbol{y_2} \in \mathbb{R}^{N_2 \times 1}$, determine the common truth sequence $\boldsymbol{x} \in \mathbb{R}^{T \times 1}$ where T is unknown. Additionally, find the elements $y_1(i_k)$ and $y_2(j_k)$ that correspond to each truth sequence element $x(k)$ for $k = 1, \cdots, T$.

III. HyperBaT

A. Mapping with HyperBaT

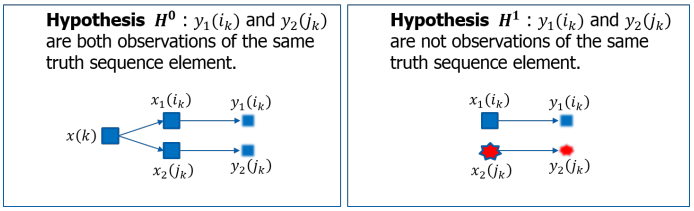

Fig. 4: H^0 and H^1 hypotheses.

HyperBaT implements a probabilistic approach to estimate the optimal mapping function Φ. The key functionality in HyperBaT is the ability to identify an insert in the signal. This is achieved by introducing the concept of an insert hypothesis, which associates inserts with probabilities. Suppose that index k of the mapping function (3a)-(3b) gives an element from each of the two sequences: $y_1(i_k)$ and $y_2(j_k)$. As represented in Fig. 4, the elements $y_1(i_k)$ and $y_2(j_k)$ can be classified into one of two hypotheses:

- **Hypothesis $\boldsymbol{H^0}$**: $y_1(i_k)$ and $y_2(j_k)$ are both observations of the same truth sequence element and $x_1(i_k) = x_2(j_k) = x(k)$.
- **Hypothesis $\boldsymbol{H^1}$**: $y_1(i_k)$ and $y_2(j_k)$ are not observations of the same element and $x_1(i_k) \neq x_2(j_k)$.

The insert hypothesis is associated with a probability by focusing on the difference of the observed elements, $y_1(i_k) - y_2(j_k)$. Under hypothesis H^0,

$$y_1(i_k) - y_2(j_k) = n_1(i_k) - n_2(j_k) \sim \mathcal{N}(0, \sigma^2_{H^0}), \tag{4}$$

where the summation property of Gaussian distributions is used to obtain $\sigma^2_{H^0} = 2\sigma^2_n$. Alternatively, under hypothesis H^1,

$$\begin{aligned} y_1(i_k) - y_2(j_k) &= x_1(i_k) - x_2(j_k) + n_1(i_k) - n_2(j_k), \\ &\sim \mathcal{N}(0, \sigma^2_{H^1}), \end{aligned} \tag{5}$$

where $\sigma^2_{H^1} = 2\sigma^2_x + 2\sigma^2_n$. Signal variance σ^2_x and noise variance σ^2_n are not known, so $\sigma^2_{H^1}$ and $\sigma^2_{H^0}$ are left as unknowns.

Examining $\sigma^2_{H^0}$ and $\sigma^2_{H^1}$, it is clear that $\sigma^2_{H^0} < \sigma^2_{H^1}$. Therefore, on average the magnitude of $y_1(i_k) - y_2(j_k)$ is greater under the H^1 hypothesis than the H^0 hypothesis. Representing the insert hypothesis in terms of the magnitude of the difference between the elements,

$$\text{Hypothesis } H^0: \quad |y_1(i_k) - y_2(j_k)| \sim \mathcal{D}(\sigma_{H^0}), \tag{6}$$

$$\text{Hypothesis } H^1: \quad |y_1(i_k) - y_2(j_k)| \sim \mathcal{D}(\sigma_{H^1}), \tag{7}$$

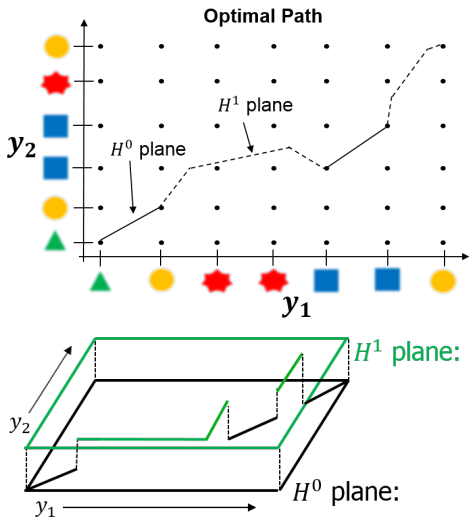

Fig. 5: Correct mapping of the example from Fig. 3 drawn in (left) 2D and (right) 3D.

where $\mathcal{D}(\sigma)$ is the half-normal distribution with standard deviation σ and probability distribution function

$$f_{\mathcal{D}}(x; \sigma) = \sqrt{\frac{2}{\pi}} \frac{1}{\sigma} \mathrm{e}^{-\frac{x^2}{2\sigma^2}}, \quad x > 0. \tag{8}$$

The notation $H_k = H^0$ is used to represent hypothesis H^0 and $H_k = H^1$ to represent hypothesis H^1 for mapping index k with elements $y_1(i_k)$ and $y_2(j_k)$.

The insert hypothesis between two observation elements adds an extra dimension of information in the mapping between the observed sequences $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$, which allows the mapping from Fig. 3 to be extended so that a mapping is a path through a three dimensional grid. Fig. 5 shows that $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ form two dimensions of the grid while the insert hypothesis forms the third. The two mapping levels in the insert hypothesis dimension are referred to as the H^0 and H^1 planes. The path through the grid corresponding to the mapping between two observation sequences is determined by (i_k, j_k, H_k) for $k = 1, \cdots, T$. Observed elements estimated to share a truth sequence element are on the H^0 plane and elements estimated to not share a truth sequence element are on the H^1 plane. Therefore, an insert is detected by observing when the mapped path jumps from the H^0 plane to the H^1 plane for the length of the insert and ends in the H^0 plane. It is the extra dimension of the insert hypothesis that allows inserts to be discovered, regardless of length, and with a minimal, constant computational complexity increase.

Here we define variables that are used in the derivation of HyperBaT. Let N be the number of estimated mapped pairs from the mapping of $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$. Then, the two dimensional mapping information for the mapped elements is

$$\boldsymbol{\Phi} = \begin{bmatrix} i_1 & j_1 \\ i_2 & j_2 \\ \vdots & \vdots \\ i_N & j_N \end{bmatrix} = \begin{bmatrix} \phi_{\boldsymbol{y_1}}(1) & \phi_{\boldsymbol{y_2}}(1) \\ \phi_{\boldsymbol{y_1}}(2) & \phi_{\boldsymbol{y_2}}(2) \\ \vdots & \vdots \\ \phi_{\boldsymbol{y_1}}(N) & \phi_{\boldsymbol{y_2}}(N) \end{bmatrix} \tag{9}$$

and $\mathbf{\Phi}_k = [i_k \ j_k]$ is the two dimensional mapping for index k. The third dimension of the mapping is grouped into

$$\boldsymbol{H}(\mathbf{\Phi}) = \begin{bmatrix} H_1(\mathbf{\Phi}_1) & H_2(\mathbf{\Phi}_2) & \cdots & H_N(\mathbf{\Phi}_N) \end{bmatrix}^T, \quad (10)$$

where $H_k(\mathbf{\Phi}_k) \in \{H_k^0, H_k^1\}$ is the hypothesis for the elements $y_1(i_k)$ and $y_2(j_k)$. The observations are grouped into the vector

$$\boldsymbol{Y}(\mathbf{\Phi}) = \begin{bmatrix} Y_1(\mathbf{\Phi}_1) & \cdots & Y_2(\mathbf{\Phi}_N) \end{bmatrix}^T, \quad (11)$$

where $Y_k(\mathbf{\Phi}_k) = |y_1(i_k) - y_2(j_k)|$. For simplicity in notation, the functional dependencies of the parameters will be neglected in the rest of the derivation.

The mapped path through the three dimensional grid is determined by $\mathbf{\Phi}$ and $\boldsymbol{H}$. Many choices of $\mathbf{\Phi}$ and $\boldsymbol{H}$ exist that correspond with different mapped paths. The best path is chosen by maximizing the joint probability of a mapped path $(\mathbf{\Phi}, \boldsymbol{H})$ and observations $(\boldsymbol{Y})$,

$$\hat{\mathbf{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H}, \mathbf{\Phi}} P(\mathbf{\Phi}, \boldsymbol{H}, \boldsymbol{Y}). \quad (12)$$

The probability of a particular mapping $P(\mathbf{\Phi}, \boldsymbol{H}, \boldsymbol{Y})$ is not directly available. However, the probability can be computed by introducing the standard deviations of the H^0 and H^1 hypotheses as nuisance parameters, which are not of direct interest, but must be accounted for in the analysis. The standard deviation for each mapped pair is placed in the vector

$$\boldsymbol{\sigma}(\boldsymbol{H}) = \begin{bmatrix} \sigma_1(H_1) & \sigma_2(H_2) & \cdots & \sigma_N(H_N) \end{bmatrix}^T, \quad (13)$$

with elements $\sigma_k(H_k) \in \{\sigma_{H^0}, \sigma_{H^1}\}$ where σ_{H^0} and σ_{H^1} are the standard deviations of the H^0 and H^1 hypotheses, respectively. The probability in (12) can be evaluated by integrating over the nuisance parameter vector,

$$\hat{\mathbf{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H}, \mathbf{\Phi}} P(\mathbf{\Phi}, \boldsymbol{H}, \boldsymbol{Y}), \quad (14a)$$

$$= \arg\max_{\boldsymbol{H}, \mathbf{\Phi}} \int_0^\infty \cdots \int_0^\infty P(\mathbf{\Phi}, \boldsymbol{H}, \boldsymbol{Y}, \boldsymbol{\sigma}) \ d\boldsymbol{\sigma}, \quad (14b)$$

where $d\boldsymbol{\sigma} = d\sigma_1 d\sigma_2 \cdots d\sigma_N$. Integrating over the nuisance parameter $\boldsymbol{\sigma}$ provides the best mapping estimate without knowledge of σ_{H^0} and σ_{H^1}.

The integral in (14) is evaluated as shown in Appendix A and the resulting expression can be written as

$$\hat{\mathbf{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H}, \mathbf{\Phi}} \prod_{k=1}^{T} \Lambda_k a_k, \quad (15)$$

which is the multiplication of two terms for each mapping index k. The first term in (15) is the measurement probability

$$\Lambda_k = P(Y_k | \mathbf{\Phi}_k, H_k), \quad (16a)$$

$$= \int_0^\infty P(Y_k | \mathbf{\Phi}_k, \sigma_k) \cdot P(\sigma_k | H_k) \ d\sigma_k. \quad (16b)$$

This term is dependent on whether the mapping is in the H^0 or H^1 plane. A mapping in the H^0 plane is given by

$$\Lambda_k^0 = \mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}}, \quad (17)$$

where σ_t is a threshold standard deviation derived in Section III-C. Alternatively, a mapping in the H^1 plane is given by

$$\Lambda_k^1 = 1 - \Lambda_k^0. \quad (18)$$

The second term in (15) is the transition probability

$$a_k = P(\mathbf{\Phi}_k, H_k | \mathbf{\Phi}_{k-1}, H_{k-1}), \quad (19)$$

which is the probability that the mapping transitions from one point to another.

Each possible mapping has different values of Λ_k and a_k. Similar to DTW and HMM methods, the structure of (15) allows dynamic programming to be used to find the optimal mapping. After the mapped path is determined, warped elements are identified as vertical or horizontal path segments in the H^0 plane. Inserted elements are identified as path segments in the H^1 plane.

B. Estimating the Common Subset Sequence and SNR

The mapping $(\mathbf{\Phi}, \boldsymbol{H})$ given by HyperBaT links the common elements in the two sequences $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$. This information can be used to estimate the common subset of elements $\boldsymbol{x}$. This is achieved by averaging pairs of elements that are mapped together with an H^0 hypothesis. The mapping indices with an H^0 hypothesis are defined as $\boldsymbol{k_0}$, which is of length T such that $y_1(k_0(m))$ and $y_2(k_0(m))$ are estimated to have an H^0 hypothesis for $m = 1, \ldots, T$. Then, the estimated truth sequence element for index k_0 is

$$\hat{x}(k_0(m)) = \frac{y_1(i_{k_0(m)}) + y_2(j_{k_0(m)})}{2}. \quad (20)$$

Estimation of the truth sequence also allows the SNR of $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ to be calculated under the assumption that they have the same SNR. The signal variance is estimated from the T estimated truth sequence elements as

$$\hat{\sigma}_x^2 = \frac{1}{T-1} \sum_{k_0=1}^{T} \left[\hat{\boldsymbol{x}}(k_0) - \sum_{k_0=1}^{T} \frac{\hat{\boldsymbol{x}}(k_0)}{T} \right]^2. \quad (21)$$

Since the elements mapped to the truth sequence are in the H^0 hypothesis plane, $a_{k_0} = \boldsymbol{y_1}(i_{k_0}) - \boldsymbol{y_2}(j_{k_0}) \sim \mathcal{N}(0, 2\sigma_n^2)$ and the noise variance is estimated as

$$\hat{\sigma}_n^2 = \frac{\frac{1}{T-1} \sum_{k_0=1}^{T} \left[a_{k_0} - \sum_{k_0=1}^{T} \frac{a_{k_0}}{T} \right]^2}{2}. \quad (22)$$

The estimated SNR for the observed sequences is then

$$SNR = \frac{\hat{\sigma}_x^2}{\hat{\sigma}_n^2}. \quad (23)$$

C. Computing the Threshold σ_t

The derivation of the HyperBaT distance metric in Appendix A uses the threshold σ_t to separate the prior distribution of σ into H^0 and H^1 regions, which is shown in Fig. 10. This threshold is the only tuning parameter for HyperBaT. Rather than arbitrarily choosing σ_t, it is chosen to achieve a desired probability of incorrectly hypothesizing an insert in the mapping. This occurs when two elements originate from the same common subset element, but are incorrectly estimated to

have an H^1 hypothesis. The desired probability of an incorrect insert hypothesis is defined to be p_t. Then, for the elements $y_1(i_k)$ and $y_2(j_k)$:

$$p_t = P(\hat{H}_k = H^1 | Y_k, \Phi_k, H_k = H^0). \tag{24}$$

Appendix B derives the threshold that achieves the probability of an incorrect insert hypothesis of p_t and the result can be written as

$$\sigma_t = \sqrt{\log(2) x^2 \sigma_n^2}, \tag{25}$$

where $x = \sqrt{\frac{-2\sigma_t^2 \log(\frac{1}{2})}{2\sigma_n^2}}$ is the value of the Q-function that obtains $Q(x) = p_t/2$.

The threshold is a function of the noise variance σ_n^2. However, it should be noted that the noise variance is not assumed to be known in HyperBaT. In practice, the average value for σ_n^2 can be estimated from past data and used in (25) under the assumption that noise variance does not vary significantly through the sampled data.

D. Boundary Conditions

HyperBaT achieves better mapping accuracy if the boundary conditions are specified. DTW is also frequently implemented using a boundary condition constraint to improve performance [7], [13]. The boundary conditions specify the elements in $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$ that map to the first and last elements in the common subset sequence $\boldsymbol{x}$. Thus, the boundary conditions are knowledge of i_k and j_k such that $\boldsymbol{x}(k) \rightarrow \boldsymbol{y_1}(i_k), \boldsymbol{y_2}(j_k)$ for $k = 1$ and $k = T$. It is typically assumed the beginning condition is known (i_k, j_k for $k = 1$). Additional performance gains are achieved if the ending condition is assumed known (i_k, j_k for $k = T$), which will be referred to as a *forced ending*. Many applications do not have knowledge of the ending condition, requiring a non-forced ending implementation. Section IV contains simulations of HyperBaT and DTW with and without forced ending to study the effects of boundary condition knowledge.

IV. Simulation and Experimental Results

A. Simulated Signals

This section compares the performance of HyperBaT and DTW using simulated sequences. Simulations offer the advantage that the underlying signals are known and can be used as ground truth in comparison to various algorithm mappings. The common subset sequence $\boldsymbol{x}$ is simulated as $\boldsymbol{x}(k) \sim \mathcal{N}(0, 1)$ for $k = 1, \ldots, T$. The first sequence $\boldsymbol{y_1}$ is created with no inserted elements and only Gaussian noise is added to $\boldsymbol{x}$ to achieve a specified SNR. The second sequence $\boldsymbol{y_2}$ is generated with the same common subset and SNR, but also contains an insert of a specified length (all inserted elements are together).

Fig. 6 shows an example mapping from HyperBaT where $T = 20$ and $\boldsymbol{y_2}$ has an insert of length 8 following the 8^{th} element. The type of line connecting elements dictates what insert hypothesis plane the elements are mapped to. Each pair of elements that are estimated to have an H^0 hypothesis are connected with a solid line while element pairs estimated to have an H^1 hypothesis are connected with a dashed line. The type of symbol for each element dictates the two dimensional mapping. Stars for both elements result from a diagonal path step and an element is a circle if the mapping index for that sequence progresses while the other sequence mapping index remains stationary. In this example, HyperBaT correctly maps the sequences.

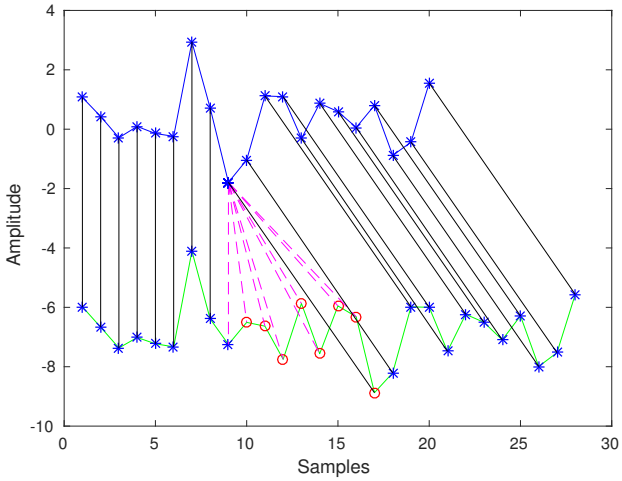

Fig. 6: HyperBaT without forced ending for two simulated sequences. $T = 20$ and $\boldsymbol{y_2}$ is generated by adding 8 inserted elements consecutively. In this example, HyperBaT correctly estimates the mapping.

B. HyperBaT Performance and Comparison with DTW

This subsection compares the performance of HyberBaT with the standard DTW/HMM approach from [18] for different insert lengths and SNRs. The probability of correct mapped points is used as a performance metric to compare the algorithms, which is computed by first calculating the correct mapped pairs (i_k, j_k) for $k = 1, \ldots, T$. Then P_c, the probability of correct mapping for an algorithm, is calculated as:

$$P_c = \frac{N_{\text{correct}}}{T}, \tag{26}$$

where N_{correct} is the number of correct mapped pairs found from the algorithm. Beginning conditions are assumed known and HyperBaT and DTW are run with and without forced ending. The simulated sequences are generated as described in Section IV-A. In each of the simulated scenarios, the insert is added after the 8^{th} element in $\boldsymbol{y_2}$.

Fig. 7 compares the performance of HyperBaT and DTW for several scenarios. Fig. 7 (a) shows the performance of HyperBaT and DTW for a range of SNR values when $T = 20$ and the insert length is 8. It is seen that the forced ending implementation of HyperBaT achieves better correct mapping probability than DTW for all SNRs. However, non-forced end HyperBaT struggles at lower SNR values for this scenario. It should be noted that correct mapping of the sequences up to the insert followed by incorrect mapping once the insert begins gives a probability of 0.4. In Fig. 7 (b) the insert length is varied while keeping SNR fixed at 30 dB and $T = 20$. Here,

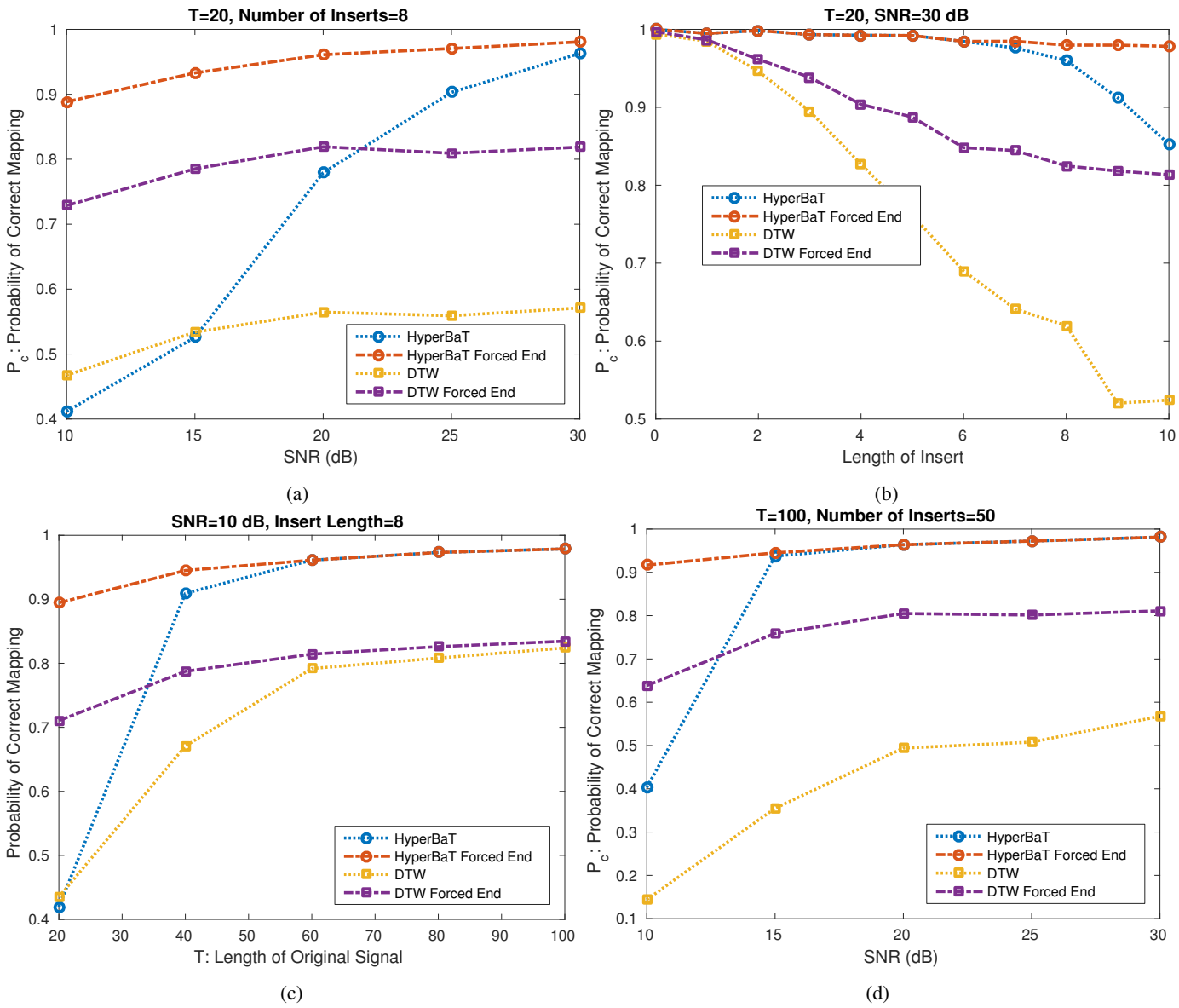

Fig. 7: Simulated HyperBaT and DTW performance with and without forced ending. In (a), results are shown for a range of SNR when $T = 20$ and $\boldsymbol{y_2}$ has an insert of length 8. In (b), insert length is varied while $T = 20$ and SNR$= 30$ dB. In (c), T is varied while SNR is 10 dB and insert length is 8. In (d), SNR is varied while $T = 100$ and insert length is 50.

it is seen that HyperBaT provides further improvement over DTW as the length of the insert increases.

HyperBaT with non-forced ending does not perform well for lower SNR values in Fig. 7 (a). However, T is only 20 in this scenario. Adding more common elements after the insert allows the uncertainty in the mapping to dissipate and the mapping converges to fewer numbers of likely paths. This is seen in Fig. 7 (c), which shows the probability of correct mapping under a scenario where an insert of length 8 appears after the 8^{th} element, but the length of the truth signal is extended so that more elements are added after the insert. It is seen that the non-forced ending implementation of HyperBaT converges with the forced-ending implementation of HyperBaT as the number of elements after the insert increases. This example is useful as it provides insight into practical considerations. Scenarios without forced-ending knowledge and a low SNR require long sequences to reach adequate performance.

The reader may also question what performance is achieved for larger insert lengths. Fig. 7 (d) plots the probability of correct mapping when $T = 100$, where again, $\boldsymbol{y_2}$ has an insert after the 8^{th} element. However, this time the insert is of length 50. This plot shows that even for lengthy inserted sequences, HyperBaT is able to have good mapping performance and that large numbers of elements after the insert allows the non-forced ending implementation of HyperBaT to approach the performance of forced ending HyperBaT implementation.

C. Example Insert in RF Side-Channel Signal From a Computer Processor

Recent research has suggested that radio frequency (RF) side channels may be a useful method for cybersecurity monitoring [5], [9]. The authors in [5] developed a method to measure the RF signal emitted from computing device processors, enabling remote security monitoring via the RF side channel. This physically separate monitoring approach offers many advantages over traditional methods, including a reduced risk of the monitoring capability being compromised during an attack. Callan et al.[5] demonstrated that these emissions can be remotely detected and used to track a program's execution path on the monitored device and generate profiling information.

However, there are unmet challenges with RF side channel monitoring. The monitored processor not only executes the sequential instructions from the program, but also additional instructions, such as those corresponding with a cache miss or operating system interrupt. These inserted instructions are often unpredictable and can change the length of the signal between two measurements of the same program. The length of the inserted segments can vary significantly. Thus, program execution tracking requires an algorithm that can compare signals of different lengths and identify inserted elements without prior knowledge of the length or structure of inserts.

As a real world example, a section of code from a program is executed at two separate instances and side channel measurements are collected. Then, a non-forced ending implementation of HyperBaT is used to map the sequences as shown in Fig. 8 (a). HyperBaT identifies an insert with three elements and estimates the SNR from (23) to be 10.99 dB. Fig. 8 (b) shows the two sequences overlaid after the inserts have been removed.

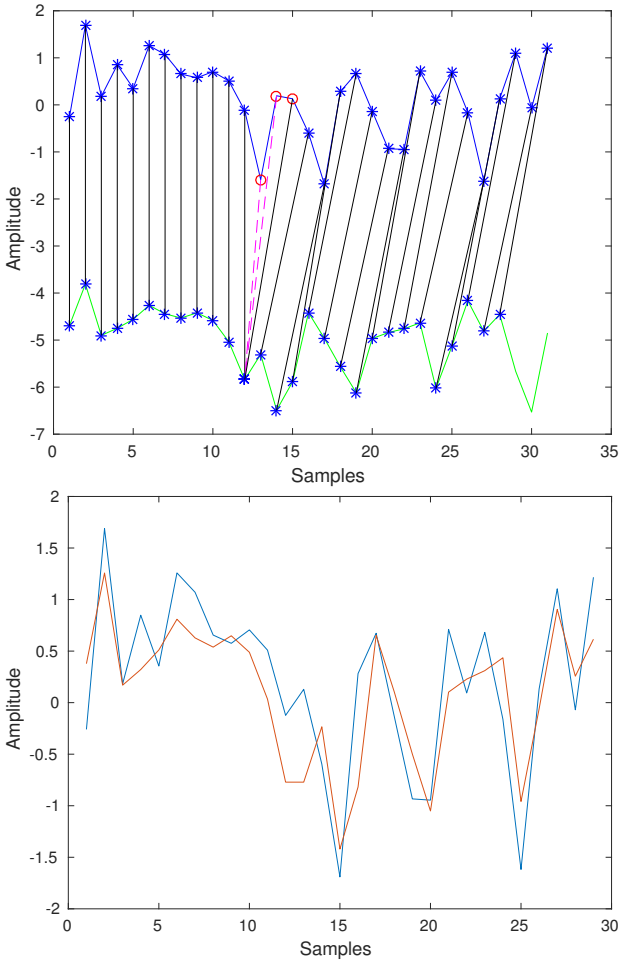

Fig. 8: (a) Two RF side channel measurements from the same executed program are mapped with HyperBaT. (b) The two mapped sequences from (a) are overlaid after removing the inserted elements.

V. Conclusion

HyperBaT has been proposed as a novel algorithm that maps two non-identical time sequences by identifying the common elements between the sequences as well as any inserted, deleted, or warped elements. HyperBaT overcomes the limitations of DTW and HMM methods by utilizing probabilistic methods to explicitly map inserted elements in a separate dimension. This is achieved without requiring prior knowledge of insert size and structure along with computational complexity that does not depend on the allowable number of inserts.

The performance of HyperBaT was compared to DTW for aligning sequences with inserted elements. Simulations show that HyperBaT without forced ending has better probability of correct mapping than DTW without forced ending, particularly for high SNR sequences. Forced ending implementation of HyperBaT performs particularly well, however the knowledge required for forced ending is not always available. Non-forced ending implementation of HyperBaT can achieve results similar to forced ending implementation if there are sufficient numbers of elements after an insert.

The probabilistic output of HyperBaT can readily be used with numerous other data modeling approaches. Future work will include integration of HyperBaT with machine learning methods for large scale time mapping. Additionally, it may be possible to reduce computational complexity to accelerate HyperBaT as has been done with DTW to make it more feasible for large scale data applications such as data mining [8], [1], [2], [21].

VI. Acknowledgements

This work has been supported by DARPA LADS contract FA8650-16-C-7620. The views and findings in this paper are those of the authors and do not necessarily reflect the views of DARPA.

Appendix A
Probability Metrics

A dynamic Bayesian network model (DBN) is used to represent dependencies of variables and evaluate (14), which is visualized as a directed acyclic graphical model in Fig. 9. The graphical model represents dependencies between random variables with arrows. The state at index k is representative of a mapped pair of elements (Φ_k, H_k), which have observation

Y_k. Markovity is assumed for the states so that the state at mapped point k only depends on the state at mapped point $k-1$. The standard deviation σ_k is determined by the hypothesis H_k, and Y_k is the observation of the mapped elements from Φ_k and depends on the standard deviation σ_k. These dependencies are used to simplify the probability in (14).

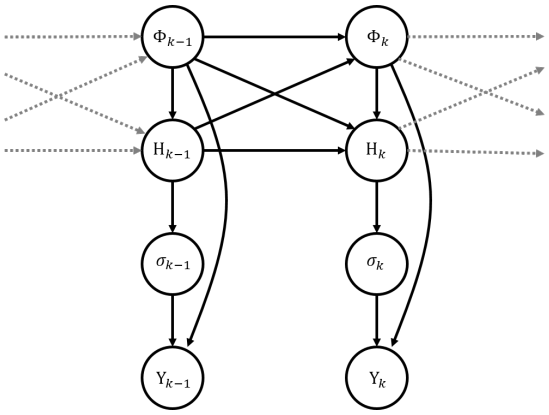

Fig. 9: Graphical model for mapping and observations.

Expanding (14) using the chain rule of probability,

$$\begin{aligned}\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{H}} &= \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \int_0^\infty \cdots \int_0^\infty P(\boldsymbol{\Phi}, \boldsymbol{H}, \boldsymbol{Y}, \boldsymbol{\sigma})\ d\boldsymbol{\sigma}, \\ &= \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \int_0^\infty \cdots \int_0^\infty P(\boldsymbol{Y}, \boldsymbol{\sigma}|\boldsymbol{\Phi}, \boldsymbol{H}) \cdot P(\boldsymbol{\Phi}, \boldsymbol{H})\ d\boldsymbol{\sigma}, \\ &= \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \int_0^\infty \cdots \int_0^\infty P(\boldsymbol{Y}|\boldsymbol{\Phi}, \boldsymbol{H}, \boldsymbol{\sigma}) \\ &\qquad \cdot P(\boldsymbol{\sigma}|\boldsymbol{\Phi}, \boldsymbol{H}) \cdot P(\boldsymbol{\Phi}, \boldsymbol{H})\ d\boldsymbol{\sigma}.\end{aligned} \tag{27}$$

The Markovity assumption of the states $(\boldsymbol{\Phi}_k, \boldsymbol{H}_k)$ (as seen in Fig. 9) is exploited to give,

$$\begin{aligned}\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \int_0^\infty \cdots \int_0^\infty \prod_{k=1}^T P(Y_k|\boldsymbol{\Phi}_k, H_k, \sigma_k) \\ \cdot P(\sigma_k|\Phi_k, H_k) \cdot P(\boldsymbol{\Phi}_k, H_k|\boldsymbol{\Phi}_{k-1}, H_{k-1})\ d\boldsymbol{\sigma}.\end{aligned} \tag{28}$$

Referencing the graphical model from Fig. 9, conditional dependencies on independent terms can be removed (Y_k does not depend on H_k and σ_k does not depend on Φ_k) so that $P(Y_k|\boldsymbol{\Phi}_k, H_k, \sigma_k) = P(Y_k|\boldsymbol{\Phi}_k, \sigma_k)$ and $P(\sigma_k|\Phi_k, H_k) = P(\sigma_k|H_k)$, which allows it to be written as

$$\begin{aligned}\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \int_0^\infty \cdots \int_0^\infty \prod_{k=1}^T P(Y_k|\boldsymbol{\Phi}_k, \sigma_k) \cdot P(\sigma_k|H_k) \\ \cdot P(\boldsymbol{\Phi}_k, H_k|\boldsymbol{\Phi}_{k-1}, H_{k-1})\ d\boldsymbol{\sigma}.\end{aligned} \tag{29}$$

This equation can then be rearranged such that each index k is integrated over σ_k separately, leading to

$$\begin{aligned}\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \prod_{k=1}^T \Bigg\{ \Bigg[\int_0^\infty P(Y_k|\boldsymbol{\Phi}_k, \sigma_k) \cdot P(\sigma_k|H_k)\ d\sigma_k \Bigg] \\ \cdot P(\boldsymbol{\Phi}_k, H_k|\boldsymbol{\Phi}_{k-1}, H_{k-1}) \Bigg\}.\end{aligned} \tag{30}$$

Similar to a standard HMM, the terms can then be grouped into two categories: measurement probabilities (Λ_k) and transition probabilities (a_k),

$$\hat{\boldsymbol{\Phi}}, \hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H},\boldsymbol{\Phi}} \prod_{k=1}^T \Lambda_k a_k, \tag{31}$$

where

$$\Lambda_k = P(Y_k|\boldsymbol{\Phi}_k, H_k) = \int_0^\infty P(Y_k|\boldsymbol{\Phi}_k, \sigma_k) \cdot P(\sigma_k|H_k)\ d\sigma_k \tag{32}$$

and

$$a_k = P(\boldsymbol{\Phi}_k, H_k|\boldsymbol{\Phi}_{k-1}, H_{k-1}) \tag{33}$$

are the measurement and transition probabilities respectively at index k. The following subsections derive analytical forms for (32) and (33).

A. Measurement Probability Λ_k

The first term in the measurement probability (32) is the likelihood of an observation for a particular standard deviation. Simplifying this term:

$$\begin{aligned}P(Y_k|\boldsymbol{\Phi}_k, \sigma_k) &= P\Big(|y_1(i_k) - y_2(j_k)|\Big|\boldsymbol{\Phi}_k, \sigma_k\Big), \\ &= \sqrt{\frac{2}{\pi}}\frac{1}{\sigma_k} \mathrm{e}^{-\frac{|\boldsymbol{y_1}(i_k) - \boldsymbol{y_2}(j_k)|^2}{2\sigma_k^2}},\end{aligned} \tag{34}$$

which follows because Y_k has a half-normal distribution as seen in (8).

The second term of the measurement probability $(P(\sigma_k|H_k))$ is the likelihood of a particular σ_k given a hypothesis and has different values depending on whether the mapping is on the H^0 or H^1 plane. Thus, Λ_k must be evaluated differently for the H^0 and H^1 hypotheses. The likelihood is calculated by expanding the prior on σ_k,

$$\begin{aligned}P(\sigma_k) &= \sum_i P(\sigma_k, H_k = H^i), \\ &= \sum_i P(\sigma_k|H_k = H^i)P(H_k = H^i), \\ &= P(\sigma_k|H_k = H^0)P(H_k = H^0) \\ &\qquad + P(\sigma_k|H_k = H^1)P(H_k = H^1), \\ &= \frac{1}{2}\Big[P(\sigma_k|H_k = H^0) + P(\sigma_k|H_k = H^1)\Big],\end{aligned} \tag{35}$$

where the last step follows by assuming no prior knowledge of the hypothesis so that $P(H_k = H^0) = P(H_k = H^1) = 1/2$.

Jeffrey's prior [10], $P(\sigma_k) \propto 1/\sigma_k^2$ is chosen as an uninformative prior on σ_k. Then,

$$P(\sigma_k) = \frac{1}{\sigma_k^2} \propto P(\sigma_k|H_k = H^0) + P(\sigma_k|H_k = H^1). \quad (36)$$

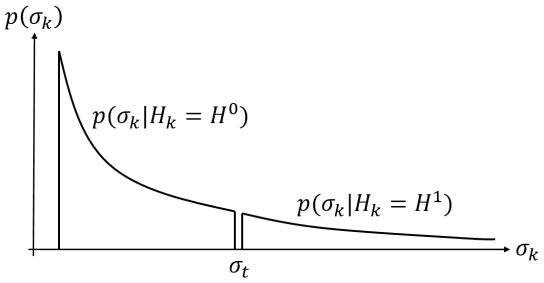

Fig. 10: Jeffreys prior on σ_k is separated into two components, where each component corresponds to a conditional dependence on hypothesis H^0 or H^1.

A simple model is used for $P(\sigma_k|H_k = H^0)$ and $P(\sigma_k|H_k = H^1)$ where they are assumed to be non-overlapping and separated by a threshold σ_t such that,

$$P(\sigma_k|H_k = H^0) = \frac{1}{\sigma_k^2}, \text{ for } 0 \leq \sigma_k < \sigma_t, \quad (37a)$$

$$P(\sigma_k|H_k = H^1) = \frac{1}{\sigma_k^2}, \text{ for } \sigma_t \leq \sigma_k < \infty. \quad (37b)$$

Therefore, the prior on σ_k is viewed as the summation of non-overlapping likelihoods $P(\sigma_k|H_k = H^0)$ and $P(\sigma_k|H_k = H^1)$ as seen in Fig. 10. As a sanity check, it is simple to show that $P(H_k = H^0|\sigma_k) + P(H_k = H^1|\sigma_k) = 1$ is also satisfied by separating the likelihoods on σ_k as in (37a)-(37b).

The probabilities in (37a)-(37b) are used as the second term in the measurement probability (32). Any mapping index with an H^0 hypothesis is on the bottom plane of the mapping grid and has measurement probability $\Lambda_k^0 = \Lambda_k(H_k = H^0)$ where

$$\begin{aligned}\Lambda_k^0 &= \int_0^\infty P(Y_k|\mathbf{\Phi}_k, \sigma_k) \cdot P(\sigma_k|H_k = H^0)\, d\sigma_k, \\ &= \int_0^{\sigma_t} \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_k} \mathrm{e}^{-\frac{|\boldsymbol{y_1}(i_k) - \boldsymbol{y_2}(j_k)|^2}{2\sigma_k^2}} \cdot \frac{1}{\sigma_k^2}\, d\sigma_k \\ &= \int_0^{\sigma_t} \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_k^3} \mathrm{e}^{-\frac{|\boldsymbol{y_1}(i_k) - \boldsymbol{y_2}(j_k)|^2}{2\sigma_k^2}}\, d\sigma_k \\ &= \sqrt{\frac{2}{\pi}} \frac{1}{|y_1(i_k) - y_2(j_k)|^2} \mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}}.\end{aligned} \quad (38)$$

Alternatively, a mapping index with an H^1 hypothesis is on the top plane of the mapping grid and has measurement probability $\Lambda_k^1 = \Lambda_k(H_k = H^1)$ where

$$\begin{aligned}\Lambda_k^1 &= \int_{\sigma_t}^\infty \sqrt{\frac{2}{\pi}} \frac{1}{\sigma_k^3} \mathrm{e}^{-\frac{|\boldsymbol{y_1}(i_k) - \boldsymbol{y_2}(j_k)|^2}{2\sigma_k^2}}\, d\sigma_k \\ &= \sqrt{\frac{2}{\pi}} \frac{1}{|y_1(i_k) - y_2(j_k)|^2} \left(1 - \mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}}\right).\end{aligned} \quad (39)$$

From (38) and (39), the factor $\sqrt{\frac{2}{\pi}} \frac{1}{|y_1(i_k) - y_2(j_k)|^2}$ appears in both the H^0 and H^1 hypothesis. Therefore, it can be neglected giving

$$\Lambda_k^0 = \mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}}, \quad (40a)$$

$$\Lambda_k^1 = 1 - \Lambda_k^0. \quad (40b)$$

B. Transition Probability a_k

The transition probability a_k from (33) determines the probability of moving from one index in the three dimensional mapping to another. Similar to HMM methods, the transition matrix can be learned from the data [17]. Alternatively, the user can define the transition matrix based on prior knowledge of the properties of the inserted elements and the structure of the signal. The transition probability is only nonzero for path steps that satisfy: $i_k - i_{k-1}, j_k - j_{k-1} = \{0, 1\}$.

APPENDIX B
DERIVING THE THRESHOLD σ_t

The estimated hypothesis in HyperBaT is chosen as a maximum likelihood estimate,

$$\begin{gathered}P(Y_k|\Phi_k, H_k = H^0) \underset{\hat{H}_k = H^1}{\overset{\hat{H}_k = H^0}{\gtrless}} P(Y_k|\Phi_k, H_k = H^1), \\ \Lambda_k^0 \underset{\hat{H}_k = H^1}{\overset{\hat{H}_k = H^0}{\gtrless}} \Lambda_k^1\end{gathered} \quad (41)$$

The probability of incorrectly mapping an insert hypothesis is then,

$$\begin{aligned}p_t &= P(\Lambda_k^0 < \Lambda_k^1 | H_k = H^0), \\ &= P\left(\mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}} < 1 - \mathrm{e}^{-\frac{|y_1(i_k) - y_2(j_k)|^2}{2\sigma_t^2}} \middle| H_k = H^0\right), \\ &= P\left(|y_1(i_k) - y_2(j_k)| > \sqrt{-2\sigma_t^2 \log\left(\frac{1}{2}\right)} \middle| H_k = H^0\right).\end{aligned} \quad (42)$$

The probability in (42) is of the half-normal distribution $|y_1(i_k) - y_2(j_k)| \sim \mathcal{D}(\sqrt{2\sigma_n^2})$ and can be evaluated using the Q-function for a normal distribution with variance $2\sigma_n^2$,

$$\frac{p_t}{2} = Q\left(\sqrt{\frac{-2\sigma_t^2 \log(\frac{1}{2})}{2\sigma_n^2}}\right). \quad (43)$$

Let $x = \sqrt{\frac{-2\sigma_t^2 \log(\frac{1}{2})}{2\sigma_n^2}}$ be the value of the Q-function that obtains $Q(x) = p_t/2$. Using this x, the threshold is calculated to be

$$\sigma_t = \sqrt{\log(2) x^2 \sigma_n^2}. \quad (44)$$

This threshold is derived in order to meet a specific probability of incorrectly mapping an insert. Alternatively, the threshold can be computed to meet other criteria such as probability of correctly mapping H^0 elements.

REFERENCES

[1] Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. CoRR, abs/1512.01876, 2015.

[2] Ghazi Al-Naymat, Sanjay Chawla, and Javid Taheri. Sparsedtw: A novel approach to speed up dynamic time warping. In Proceedings of the Eighth Australasian Data Mining Conference - Volume 101, AusDM '09, pages 117–127, Darlinghurst, Australia, Australia, 2009. Australian Computer Society, Inc.

[3] X. Anguera, R. Macrae, and N. Oliver. Partial sequence matching using an unbounded dynamic time warping algorithm. In 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 3582–3585, March 2010.

[4] R. Bellman and R. Kalaba. On adaptive control processes. IRE Transactions on Automatic Control, 4(2):1–9, November 1958.

[5] Robert Callan, Farnaz Behrang, Alenka Zajic, Milos Prvulovic, and Alessandro Orso. Zero-overhead profiling via EM emanations. In ACM Proc. of the Int. Symp. on Software Testing and Analysis, pages 401–412, 2016.

[6] Carmelo Cassisi, Placido Montalto, Marco Aliotta, Andrea Cannata, and Alfredo Pulvirenti. Similarity measures and dimensionality reduction techniques for time series data mining. 09 2012.

[7] Toni Giorgino. Computing and visualizing dynamic time warping alignments in r: The dtw package. Journal of Statistical Software, Articles, 31(7):1–24, 2009.

[8] Omer Gold and Micha Sharir. Dynamic time warping: Breaking the quadratic barrier. CoRR, abs/1607.05994, 2016.

[9] Alenka Zajic Milos Prvulovic Haider Adnan Khan, Monjur Alam. Detailed tracking of program control flow using analog side-channel signals: a promise for iot malware detection and a threat for many cryptographic implementations, 2018.

[10] Harold Jeffreys. An invariant form for the prior probability in estimation problems. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, pages 453–461, 1946.

[11] B-H Juang. On the hidden Markov model and dynamic time warping for speech recognitiona unified view. Bell Labs Technical Journal, 63(7):1213–1243, 1984.

[12] Eamonn Keogh and Chotirat Ann Ratanamahatana. Exact indexing of dynamic time warping. Knowledge and Information Systems, 7(3):358–386, Mar 2005.

[13] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques. CoRR, abs/1003.4083, 2010.

[14] Meinard Müller. Information Retrieval for Music and Motion. Springer-Verlag, Berlin, Heidelberg, 2007.

[15] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28(6):623–635, Dec 1980.

[16] Seiichi Nakagawa and Hirobumi Nakanishi. Speaker-independent english consonant and japanese word recognition by a stochastic dynamic time warping method. IETE Journal of Research, 34(1):87–95, 1988.

[17] Lawrence R Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286, 1989.

[18] Lawrence R Rabiner and Biing-Hwang Juang. Fundamentals of speech recognition. PTR Prentice Hall, 1993.

[19] Hiroaki Sakoe and Seibi Chiba. A dynamic programming approach to continuous speech recognition. In Proceedings of the Seventh International Congress on Acoustics, Budapest, volume 3, pages 65–69, Budapest, 1971. Akadémiai Kiadó.

[20] HIROAKI SAKOE and SEIBI CHIBA. Dynamic programming algorithm optimization for spoken word recognition. 26:43 – 49, 03 1978.

[21] Stan Salvador and Philip Chan. Fastdtw: Toward accurate dynamic time warping in linear time and space. In KDD workshop on mining temporal and sequential data. Citeseer, 2004.