DEEP DECLARATIVE DYNAMIC TIME WARPING FOR END-TO-END LEARNING OF ALIGNMENT PATHS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper addresses end-to-end learnable models for time series data that include a temporal alignment step via dynamic time warping (DTW). Existing approaches to differentiable DTW either differentiate through a fixed warping path or apply a continuous relaxation to the min operator found in the recursive steps used to solve the DTW problem. We instead propose a DTW layer based around deep declarative networks, which we name DecDTW. By formulating DTW as a continuous, inequality constrained optimisation problem, we can compute exact gradients for the solution of the optimal alignment (with respect to the underlying time series) using implicit differentiation. Our formulation yields a major improvement over existing approaches; our DTW layer outputs the optimal warping *path* between two time series as opposed to the DTW discrepancy value. This enables the specification of downstream loss functions on the alignment path itself, useful for instance, to improve the precision of predicted alignments when ground-truth alignments are available. We evaluate our method on two such applications, namely the audio-to-score alignment task in music information retrieval and the visual place recognition task in robotics, demonstrating state-of-the-art results in both.

1 INTRODUCTION

The dynamic time warping (DTW) algorithm computes a similarity measure between two temporal sequences, which is invariant to shifting and non-linear scaling in time. Because of this desirable invariance, DTW is ubiquitous in fields which analyze temporal sequences such as speech recognition, motion capture, time series classification and bioinformatics (Kovar & Gleicher, 2003; Zhu & Shasha, 2003; Sakoe & Chiba, 1978; Bagnall et al., 2017; Berndt & Clifford, 1994; Petitjean et al., 2014; Needleman & Wunsch, 1970). The original formulation of DTW computes the minimum cost matching between elements of the two sequences, called an alignment (or warping) path, subject to temporal constraints imposed on the matches. For two sequences of length m and n, this can be computed by first constructing an $m \times n$ pairwise cost matrix between sequence elements and subsequently solving a dynamic program (DP) using Bellman's recursion in $O(mn)$ time. Figure 2 illustrates the mechanics of the DTW algorithm.

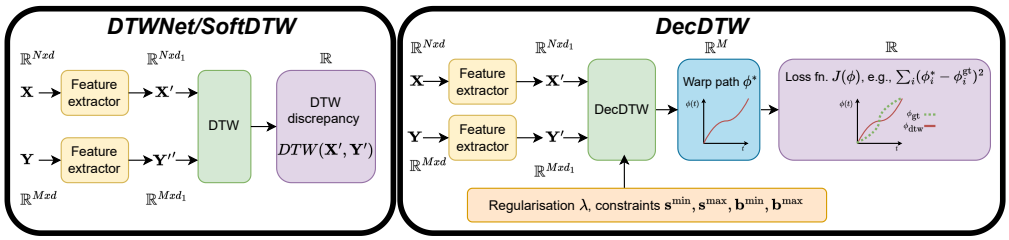


Figure 1: **Left:** DTWNet and SoftDTW layers output the DTW discrepancy between two time series as a differentiable function of its inputs. **Right:** DecDTW outputs the warping *path* ϕ, which can be used for downstream losses (e.g., MSE to a ground truth alignment). DecDTW also allows the regularisation and constraint values to be learnable parameters in an end-to-end system.

There has been interest in recent years around embedding DTW within deep learning models (Cuturi & Blondel, 2017; Cai et al., 2019; Lohit et al., 2019; Chang et al., 2019; 2021), with applications spread across a variety of learning tasks utilising audio and video data (Garreau et al., 2014; Dvornik et al., 2021; Haresh et al., 2021; Chang et al., 2019; 2021), especially where an *explicit* alignment step is desired. However, the classic DTW algorithm solves a discrete optimisation problem; thus the DTW similarity is not differentiable w.r.t. the cost matrix. This is caused by the use of $\min$ functions found in the DP recursion, preventing gradients from being evaluated by unrolling the computations. Several approaches in the literature address this limitation in distinct ways: Cuturi & Blondel (2017) and Chang et al. (2019) use a continuous relaxation of $\min$ in the DP recursion, Cai et al. (2019) and Chang et al. (2021) observe that differentiation is possible after fixing the warping path, and a body of work predicts warping paths directly from sequence elements without the need for an optimisation step (Lohit et al., 2019; Shapira Weber et al., 2019; Grabocka & Schmidt-Thieme, 2018; Kazlauskaite et al., 2019; Abid & Zou, 2018). We note that methods based on DTW differ to methods such as CTC for speech recognition (Graves & Jaitly, 2014), where word-level transcription is far more important than frame-level alignment, and an explicit alignment step is not required.

We propose a novel approach to differentiable DTW, named DecDTW, based on deep implicit layers that moves away from discrete optimisation entirely. By adapting a continuous time formulation of the DTW problem proposed by Deriso & Boyd (2019) as an inequality constrained non-linear program (NLP), we can use the deep declarative networks (DDN) framework (Gould et al., 2021) to define the forward and backward passes in our DTW layer. The forward pass involves solving for the optimal (continuous) warping path; we achieve this using a custom dynamic programming approach similar to the original DTW algorithm. The backward pass uses the identities in Gould et al. (2021) to derive gradients implicitly through the solution computed in the forward pass.

We will show that DecDTW has multiple benefits compared to existing approaches; the most important of which is that our DTW layer enables the optimal alignment *path* output by DTW to be differentiable w.r.t. its inputs, rather than only the DTW similarity value. This is particularly useful for many applications where ground truth alignments between time series pairs are available, allowing us to directly minimise alignment error using gradient descent. We show in our experiments that DecDTW is more effective compared to existing methods which utilise path information (Garreau et al., 2014; Le Guen & Thome, 2019), yielding considerable performance gains in challenging real-world alignment tasks. An overview of our proposed DecDTW layer is illustrated in Figure 1.

Our Contributions First, we propose a novel, inequality constrained NLP formulation of the DTW problem, building on the approach in Deriso & Boyd (2019). Second, we use this NLP formulation to specify our novel DecDTW layer, where gradients in the backward pass are computed implicitly as in the DDN framework (Gould et al., 2021). Third, we show how the alignment path produced by DecDTW can be used to minimise discrepancies to ground truth alignments. Last, we use our method to attain state-of-the-art performance on challenging real-world alignment tasks.

2 Related Works

Differentiable DTW Layers Early approaches to learning with temporal alignment (Zhou & De la Torre, 2009; Su & Wu, 2019; Zhou & De la Torre, 2012) involve learning a linear transformation of observations placed before a DTW alignment step. The objective of learning is to minimise the DTW discrepancy and learning proceeds by alternating between first estimating the temporal alignment using the classic DTW algorithm and then analytically solving for the optimal feature transform. DTW-Net (Cai et al., 2019) and DP-DTW (Chang et al., 2021) adapt the aforementioned approaches to the mini-batch gradient descent setting, where at each descent step the alignment path is estimated using DTW for each training observation and is then fixed. Along this fixed path, gradients of the DTW discrepancy w.r.t. the associated entries in the cost matrix can be computed.

Another class of differentiable DTW methods involves applying a continuous relaxation to the $\min$ operation in the DP recursion. Soft-DTW (Cuturi & Blondel, 2017) proposes replacing the $\min$ step with a *soft-min*, allowing gradients to be backpropagated through the recursion. D^3TW (Chang et al., 2019) combined Soft-DTW with a contrastive loss and OTAM (Cao et al., 2020) extends Soft-DTW to allow for subsequence alignment. Variations on soft-min have been proposed (Hadji et al., 2021) and there has been recent work on outlier rejection (Dvornik et al., 2021), aligning across different domains (Cohen et al., 2021) and spatio-temporal optimal transport (Janati et al., 2020). DecDTW

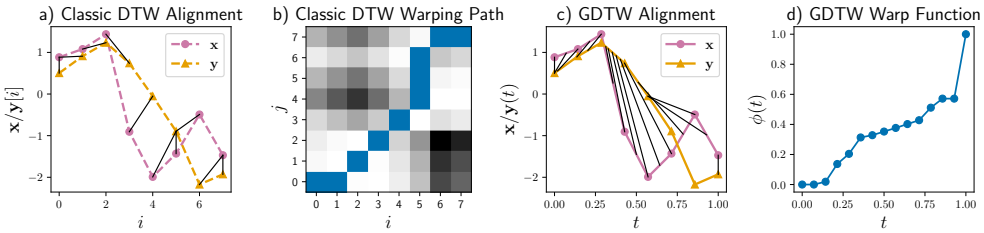


Figure 2: Classic DTW (a) is a discrete optimisation problem which finds the minimum cost warping path through a pairwise cost matrix (b). DecDTW uses a continuous variant of classic DTW (GDTW) (c) and finds an optimal time warp *function* between two continuous time signals (d).

allows the optimal alignment *path* itself to be differentiable w.r.t. layer inputs, which is not easily recoverable from the aforementioned approaches. This property allows us to specify loss functions over the alignment path itself, e.g., error against a ground truth alignment.

There is a limited body of work which allows for differentiable path information (Le Guen & Thome, 2019; Garreau et al., 2014). Notably, DILATE (Le Guen & Thome, 2019) achieves this through Soft-DTW, by deriving a soft approximation to the optimal $m \times n$ alignment path matrix. We show in our experiments that DecDTW is able to utilise path information much more effectively than DILATE, and furthermore has lower compute requirements during training.

Generalised DTW Deriso & Boyd (2019) proposed a generalised optimisation framework for dynamic time warping (GDTW) which transforms the classic discrete DTW optimisation problem over sequences to a variational optimisation problem over continuous time signals. This is achieved by first converting a time series from a discrete sequence to a continuous time signal. Second, the warping path changes from a set of indices in the pairwise cost matrix to a continuous *warp function*. Optional regularisation and constraints on the warp function can be easily included to GDTW to improve alignment quality. We will show that GDTW can be reduced to a continuous optimisation problem, i.e., an inequality constrained NLP, in Section 4. We use this NLP formulation of DTW as the basis for our declarative layer.

Deep Declarative Networks The declarative framework that allows us to incorporate differentiable optimisation algorithms into a deep network is described in Gould et al. (2021). They present theoretical results and analyses on how to differentiate constrained optimization problems via implicit differentiation. Differentiable convex problems have also been studied recently, including quadratic programs (Amos & Kolter, 2017) and cone programs (Agrawal et al., 2019a;b). This technique has been applied to various application domains including optimisation-based control (Amos et al., 2018), video classification (Fernando & Gould, 2016; 2017), action recognition (Cherian et al., 2017), visual attribute ranking (Santa Cruz et al., 2019), few-shot learning for visual recognition (Lee et al., 2019), and camera pose estimation (Campbell et al., 2020; Chen et al., 2020). To the best of our knowledge, this is the first embedding of an inequality constrained NLP within a declarative layer.

3 Continuous Time Formulation for DTW (GDTW)

In this section, we introduce a continuous time version of the DTW problem which is heavily inspired by GDTW (Deriso & Boyd, 2019). The GDTW problem is used to derive the NLP formulation for our DecDTW layer. We will describe how the results in this section can be used to derive a corresponding NLP in Section 4.

3.1 Preliminaries

Signal A time-varying signal $\mathbf{x} : [0, 1] \to \mathbb{R}^d$ is a vector-valued function of time. Signals are assumed to be differentiable (at least piecewise smooth) and can be constructed from a time series comprised of observation times $\mathbf{t} \in [0, 1]^N$ where $0 = t_1 < t_2 < \cdots < t_N \leq 1$ and associated observations or *features* given by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathbb{R}^{N \times d}$, by interpolating between observations (e.g., linear, cubic splines). Without loss of generality, we rescale time for all signals to $[0, 1]$.

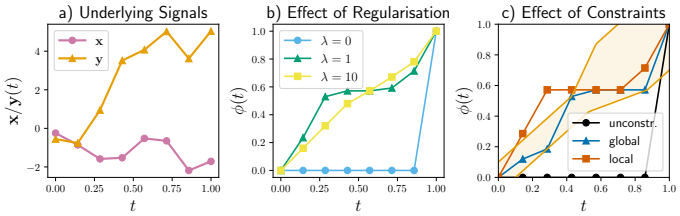


Figure 3: Regularisation (b) and constraints (c) can be applied to prevent pathological warps, e.g., $\lambda = 0$ in (b). Global and local constraints bound warp function values and derivatives, respectively.

Time warp function A time warp function $\phi : [0, 1] \to [0, 1]$ defines correspondences from times in one signal to another, and is similarly assumed to be piecewise smooth. Warp functions typically come with constraints; a common constraint requires that warp functions are non-decreasing, i.e., $\phi'(t) \geq 0$ for all t. We interpret $\mathbf{x} \circ \phi$ to be the *time warped* version of $\mathbf{x}$.

Dynamic time warping The GDTW problem can be formulated as follows. Given two signals $\mathbf{x}$ and $\mathbf{y}$, find a warp function ϕ such that $\mathbf{y} \circ \phi \approx \mathbf{x}$ in some sense over the time series features. In other words, we wish to bring signals $\mathbf{x}$ and $\mathbf{y}$ together by warping time in $\mathbf{y}$. Figure 2 illustrates how GDTW differs from classic DTW for an example sequence pair. We found in our experiments that GDTW outperformed DTW generally for real-world alignment tasks. We attribute this to the former's ability to align *in between* observations, thus enabling a more accurate alignment.

3.2 Optimisation problem formulation for GDTW

GDTW can be specified as a constrained optimisation problem where the decision variable is the time warp function ϕ and input parameters are signals $\mathbf{x}$ and $\mathbf{y}$. Concretely, the problem is given by

$$\begin{aligned} \text{minimise} \quad & f(\mathbf{x}, \mathbf{y}, \lambda, \phi) \triangleq \mathcal{L}(\mathbf{x}, \mathbf{y}, \phi) + \lambda \mathcal{R}(\phi) \\ \text{subject to} \quad & \underbrace{s^{\min}(t) \leq \phi'(t) \leq s^{\max}(t)}_{\textbf{local constraints}}, \quad \underbrace{b(t)^{\min} \leq \phi(t) \leq b(t)^{\max}}_{\textbf{global constraints}} \quad \forall t. \end{aligned} \tag{1}$$

The objective function f can be decomposed into the *signal loss* $\mathcal{L}$ relating to feature error and a warp regularisation term $\mathcal{R}$, where $\lambda \geq 0$ is a hyperparameter controlling the strength of regularisation. Since ϕ, $\mathbf{x}$ and $\mathbf{y}$ are functions, our objective function f is a functional, revealing the variational nature of the optimisation problem. We will see in Section 4 that restricting ϕ to the space of piecewise linear functions will allow us to easily solve Equation 1 while still allowing for expressive warps.

Furthermore, a number of potentially time-varying constraints are usually imposed on ϕ. We partition these constraints into *local* and *global* constraints as in Morel et al. (2018). Local constraints bound the time derivatives of the warp function and can be used to enforce the typical nondecreasing warp assumption. Global constraints bound the warp function values directly, with prominent examples being the R-K (Ratanamahatana & Keogh, 2004) and commonly used Sakoe-Chiba (Sakoe & Chiba, 1978) bands. Furthermore, global constraints are used to enforce boundary constraints, i.e., $\phi(t) \in [0, 1]$ for all t and endpoint constraints, i.e., $\phi(0) = 0, \phi(1) = 1$. Not including endpoint constraints allows for *subsequence* GDTW, where a subsequence $\mathbf{y}$ is aligned to only a portion of $\mathbf{x}$. Figure 3 illustrates how constraints can be used to change the resultant warps returned by GDTW.

Signal Loss The signal loss functional $\mathcal{L}$ in Equation 1 is defined as

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \phi) = \int_0^1 L\big(\mathbf{y}(\phi(t)) - \mathbf{x}(t)\big) \mathrm{d}t, \tag{2}$$

where $L : \mathbb{R}^d \to \mathbb{R}$ is a twice-differentiable penalty function defined over features. In our experiments we use $L(\cdot) \triangleq \| \cdot \|_2^2$, however other penalty functions (e.g., the 1-norm, Huber loss) can be used. The signal loss (Equation 2) measures the separation of features between $\mathbf{x}$ and time-warped $\mathbf{y}$ and is analogous to the classic DTW discrepancy given by the cost along the optimal warping path.

Warp Regularisation The warp regularisation functional $\mathcal{R}$ in Equation 1 is defined as

$$\mathcal{R}(\phi) = \int_0^1 R\big(\phi'(t) - 1\big) \mathrm{d}t, \tag{3}$$

where $R : \mathbb{R} \to \mathbb{R}$ is a penalty function on deviations from the identity warp, i.e., $\phi'(t) = 1$. We use a quadratic penalty $R(u) = u^2$ in this work, consistent with Deriso & Boyd (2019). This regularisation term penalises warp functions with large jumps and sufficiently high λ brings ϕ close to the identity. Regularisation is crucial for preventing noisy and/or pathological warps (Zhang et al., 2017; Wang et al., 2016) from being produced from GDTW (and DTW, more generally), and can greatly affect alignment accuracy. λ can be selected optimally using cross-validation (Deriso & Boyd, 2019).

4 Simplified NLP Formulation for GDTW

While Section 3 provides a high-level overview of the GDTW problem, we did not specify a concrete parameterisation for ϕ, which dictates how the optimisation problem in Equation 3 will be solved. In this section, we provide simplifying assumptions on ϕ, which allows us to solve Equation 3. We follow the approach in Deriso & Boyd (2019) to reduce the infinite dimensional variational problem in Equation 1 to a finite dimensional NLP. This is achieved by first assuming that ϕ is piecewise linear, allowing it to be fully defined by its value at m knot points $0 = t_1 < t_2 < \cdots < t_m = 1$. Knot points can be uniformly spaced or just be the observation times used to parameterise $\mathbf{y}$. Formally, piecewise linearity allows us to represent ϕ as $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_m) \in \mathbb{R}^m$ where $\phi_i = \phi(t_i)$. The other crucial assumption involved in reducing Equation 1 to an NLP is replacing the continuous integrals in the signal loss and warp regularisation in Section 3 with numerical approximations given by the trapezoidal rule. These assumptions yield an alternative optimisation problem given by

$$\begin{aligned} &\text{minimise} \quad \hat{f}(\mathbf{x}, \mathbf{y}, \lambda, \boldsymbol{\phi}) \triangleq \hat{\mathcal{L}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\phi}) + \lambda \hat{\mathcal{R}}(\boldsymbol{\phi}) \\ &\text{subject to} \quad s_i^{\text{min}} \leq \tfrac{\phi_{i+1} - \phi_i}{\Delta t_i} \leq s_i^{\text{max}}, \quad b_i^{\text{min}} \leq \phi_i \leq b_i^{\text{max}} \quad \forall i, \end{aligned} \tag{4}$$

where $\Delta t_i = t_{i+1} - t_i$. The new signal loss $\hat{\mathcal{L}}$ is given by

$$\hat{\mathcal{L}}(\mathbf{x}, \mathbf{y}, \boldsymbol{\phi}) = \sum_{i=1}^{m-1} \frac{L_{i+1} + L_i}{2} \Delta t_i, \quad L_i \coloneqq L(\mathbf{x}(\phi_i) - \mathbf{y}(t_i)), \tag{5}$$

which follows from the definition of $\boldsymbol{\phi}$ and the trapezoidal approximation of the integral in Equation 2. Note, the non-convexity of objective $\hat{f}$ assuming convex L and R is caused by the $\mathbf{x}(\phi_i)$ terms for arbitrary signal $\mathbf{x}$. The new regularisation term $\hat{\mathcal{R}}$ is given by

$$\hat{\mathcal{R}}(\boldsymbol{\phi}) = \sum_{i=1}^{m-1} R\left(\frac{\phi_{i+1} - \phi_i}{\Delta t_i} - 1\right) \Delta t_i, \tag{6}$$

noting that since ϕ' is assumed to be piecewise constant, we can simply use Riemann integration to evaluate the continuous integral in $\mathcal{R}$ exactly. The simplified problem in Equation 4 is actually a continuous non-linear program with decision variable $\boldsymbol{\phi} \in \mathbb{R}^n$ and linear constraints. However, the objective function is non-convex (Deriso & Boyd, 2019) and we will now describe the method used to find good solutions to Equation 4 using approximate dynamic programming.

5 Declarative DTW Forward Pass

Our DecDTW layer encodes an implicit function $\boldsymbol{\phi}^\star = \text{DecDTW}(\mathbf{x}, \mathbf{y}, \lambda, \mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}})$, which yields the optimal warp given input signals $\mathbf{x}, \mathbf{y}$, regularisation λ and global/local constraints $\mathbf{s}^{\text{min}} = \{s_i^{\text{min}}\}_{i=1}^m$, $\mathbf{s}^{\text{max}} = \{s_i^{\text{max}}\}_{i=1}^m$, $\mathbf{b}^{\text{min}} = \{b_i^{\text{min}}\}_{i=1}^m$, $\mathbf{b}^{\text{max}} = \{b_i^{\text{max}}\}_{i=1}^m$. $\boldsymbol{\phi}^\star$ can be used in a downstream loss functions $J(\boldsymbol{\phi}^\star)$, e.g., a discrepancy between the predicted warp $\boldsymbol{\phi}^\star$ and a desired ground truth warp $\boldsymbol{\phi}^{\text{gt}}$. The GDTW discrepancy is trivially recovered by setting $J = \hat{f}$. The DecDTW forward pass yields the solution to Equation 4 given the input parameters. We solve Equation 4 using a dynamic programming (DP) approach as proposed in Deriso & Boyd (2019) instead of calling a general-purpose NLP solver; this is to minimise computation time. The DP approach uses the fact that $\boldsymbol{\phi}$ lies on a compact subset of $\mathbb{R}^m$ defined by the global bounds, allowing for efficient discretisation. Finding the globally optimal solution to the discretised version of Equation 4 can be done using DP. The resultant solution $\boldsymbol{\phi}^\star$ is an approximation to the solution of the continuous NLP, with the approximation accuracy dependent on the resolution of discretisation. We can reduce the approximation error given a fixed resolution M using multiple rounds of refinement. In each round, a tighter discretisation is generated around the previous solution and the new discrete problem solved. A detailed description of the solver can be found in Deriso & Boyd (2019) and the appendix.

6 DECLARATIVE DTW BACKWARD PASS

We now derive the analytical solution of the gradients of $\boldsymbol{\phi}^\star$ recovered in the forward pass w.r.t. inputs $\mathbf{z} = \{\mathbf{x}, \mathbf{y}, \lambda, \mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}}\} \in \mathbb{R}^n$ using Proposition 4.6 in Gould et al. (2021) (note that gradients w.r.t. signals $\mathbf{x}, \mathbf{y}$ are w.r.t. the underlying observations $\mathbf{X}, \mathbf{Y}$). Unlike existing approaches for differentiable DTW, DecDTW allows the regularisation weight λ and constraints to be learnable parameters in a deep network. Let $\tilde{h} = [h_1, \ldots, h_p]$, where each $h_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ corresponds to an *active* constraint from the full set of inequality constraints described in Equation 4, rearranged in standard form, i.e., $h_i(\cdot) \leq 0$. For instance, an active local constraint relating to time t_j can be expressed as $h_i(\mathbf{z}, \boldsymbol{\phi}) = \phi_{j+1} - \phi_j - \Delta t_j s_j^{\text{max}} \leq 0$. Assuming non-singular H (note, this always holds for norm-squared L and R) and $\text{rank}(D_{\boldsymbol{\phi}} \tilde{h}(\mathbf{z}, \boldsymbol{\phi})) = p$, the backward pass gradient is given by

$$D\boldsymbol{\phi}(\mathbf{z}) = H^{-1}A^\top(AH^{-1}A^\top)^{-1}(AH^{-1}B - C) - H^{-1}B, \tag{7}$$

where $D\phi(\mathbf{z}) \in \mathbb{R}^{m \times n}$ is the Jacobian of estimated warp $\boldsymbol{\phi}$ w.r.t. inputs $\mathbf{z}$ and

$$\begin{aligned} A &= D_{\boldsymbol{\phi}} \tilde{h}(\mathbf{z}, \boldsymbol{\phi}) \in \mathbb{R}^{p \times m} \quad & B &= D^2_{\mathbf{z}\boldsymbol{\phi}} \hat{f}(\mathbf{z}, \boldsymbol{\phi}) \in \mathbb{R}^{m \times n} \\ C &= D_{\mathbf{z}} \tilde{h}(\mathbf{z}, \boldsymbol{\phi}) \in \mathbb{R}^{p \times n} \quad & H &= D^2_{\boldsymbol{\phi}\boldsymbol{\phi}} \hat{f}(\mathbf{z}, \boldsymbol{\phi}) \in \mathbb{R}^{m \times m}. \end{aligned} \tag{8}$$

Observe that the objective $\hat{f}$ as defined in Equation 4 only depends on a subset of $\mathbf{z}$ corresponding to $\mathbf{x}, \mathbf{y}, \lambda$ and similarly, constraints $\tilde{h}$ only depend on $\mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}}$. While Proposition 4.6 in Gould et al. (2021) has additional terms in B and H involving Lagrange multipliers, we note that since $\tilde{h}$ is at most first order w.r.t. $\{\mathbf{z}, \boldsymbol{\phi}\}$, these terms evaluate to be zero and can be ignored. We discuss how to evaluate Equation 7 using vector-Jacobian products efficiently in the appendix.

7 EXPERIMENTS

Our experiments involve two distinct application areas and utilise real-world datasets. For both applications, the goal is to use learning to improve the accuracy of predicted temporal alignments using a training set of labelled ground truth alignments. We will show that DecDTW yields state-of-the-art performance on these tasks. We have implemented DecDTW in PyTorch (Paszke et al., 2019) with open source code available to reproduce all experiments at https://anonymised.github.repo.

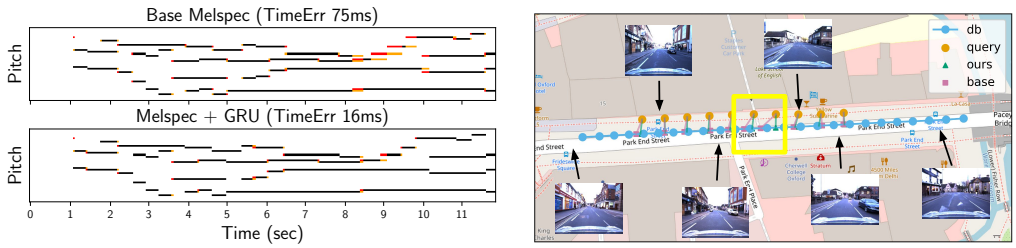
7.1 LEARNING FEATURES FOR AUDIO-TO-SCORE ALIGNMENT

Problem Formulation Our first experiment relates to audio-to-score alignment, which is a fundamental problem in music information retrieval (Thickstun et al., 2020; Ewert et al., 2009; Orio et al., 2001; Shalev-Shwartz et al., 2004), with applications ranging from score following to music transcription (Thickstun et al., 2020). The goal of this task is to align an audio recording of a music performance to its corresponding musical score/sheet music. We use the mathematical formulation proposed in Thickstun et al. (2020) for evaluating predicted audio-to-score alignments against a ground truth alignment, which we will now summarise. An alignment as a monotonic function $\phi : [0, S) \to [0, T)$ which maps positions in a score (measured in beats) to a corresponding position in the performance recording (measured in seconds). Two evaluation metrics between a predicted alignment ϕ and a ground truth alignment ϕ^{gt} are proposed, namely the temporal average error (TimeErr) and temporal standard deviation (TimeDev), given formally by

$$\text{TimeErr}(\phi, \phi^{\text{gt}}) = \frac{1}{S}\int_0^S |\phi(s) - \phi^{\text{gt}}(s)| \text{d}s, \quad \text{TimeDev}(\phi, \phi^{\text{gt}}) = \sqrt{\frac{1}{S}\int_0^S (\phi(s) - \phi^{\text{gt}}(s))^2 \text{d}s}. \tag{9}$$

Thickstun et al. (2020) assumes alignments are piecewise linear in between changepoints (where the set of notes being played changes) within the score, allowing Equation 9 to be analytically tractable.

Methodology Thickstun et al. (2020) provide a method for generating candidate alignments between a performance recording and a musical score as follows: First, synthesize the musical score (using a MIDI file) to an audio recording. Second, extract audio features from the raw audio for both the synthesised score and performance. Third, use DTW to align the two sets of features. Last, use Equation 9 to evaluate the predicted alignment. In these experiments, we learn a feature extractor on top of the base audio features with the goal of improving the alignment accuracy from DTW.



(a) (b)

Figure 4: **(a)** Example audio-to-score alignment before and after training. Red (resp. yellow) identifies notes indicated by audio feature (resp. ground truth) alignments, but not by the ground truth (resp. audio features). DecDTW training yields dramatic improvements for melspec features, even surpassing the highly optimised CQT features (30ms TimeErr). **(b)** Example VPR alignment. Query images are aligned to a database sequence using learned image embeddings. Red (green) illustrates where the pre-trained (resp. fine-tuned) system estimates the query poses using database geotags. The maximum error is reduced from 5.2m to 1.2m after fine-tuning (yellow region).

Experimental Setup We use the dataset in Thickstun et al. (2020), which is comprised of 193 piano performances of 64 scores with ground truth alignments taken from a subset of the MAESTRO (Hawthorne et al., 2019) and Kernscores (Sapp, 2005) datasets. We extract three types of base frequency-domain features from the raw audio; constant-Q transform (CQT), (log-)chromagram (chroma) and (log-)mel spectrogram (melspec). The CQT and chroma features are already highly optimised for pitch compared to melspec features. However, melspec features retain more information from the raw audio compared to CQT and chroma. We evaluated six different methods: *1-2)* Base features (no learning), using both DTW (D) and GDTW (G) for alignments. For remaining methods (3-6), each learns a feature extractor given by a Gated Recurrent Unit (GRU) (Cho et al., 2014), on top of the base features. The learned features are then used to compute the corresponding loss, described as follows: *3)* Soft-DTW (Cuturi & Blondel, 2017) loss; *4)* matching cost along ground truth alignment path (Along GT); *5)* DILATE Le Guen & Thome (2019), which utilises path information for Soft-DTW; and *6)* TimeErr loss training (DecDTW only). For DILATE, we modified the original implementation so that the path penalty matrix$\mathbf{\Omega}$ encodes the deviation from the ground truth path instead of the diagonal. The full set of hyperparameters, including dataset generation and further information on comparison methods are provided in the appendix.

Table 1: Summary of results for the audio-to-score alignment experiments

Reported metrics are TimeErr / TimeDev (ms)						
Feature Type	Base (D)	Base (G)	Soft-DTW	Along GT	DILATE	DecDTW
CQT	35 / 90	19 / 30	49 / 130	49 / 130	37 / 69	**17 / 27**
Chroma	50 / 115	24 / 39	59 / 142	59 / 142	47 / 79	**19 / 31**
Melspec	122 / 235	56 / 81	55 / 153	52 / 145	60 / 100	**16 / 27**

Results A summary of results is presented in Table 1. DTW is used for alignments in Base (D), Soft-DTW, Along GT and DILATE, while GDTW is used for alignments for Base (G) and DecDTW. Training using Soft-DTW fails to improve alignment performance over the validation set, measured by TimeErr, because the ground truth alignment is not used during learning[1]. Similarly, minimising the feature matching cost along the ground truth alignment path also fails to improve alignment accuracy, since the predicted alignment is not used during learning. DILATE, however, improves slightly on Base (D) since it can incorporate estimated path information during learning.

Training using the TimeErr loss with DecDTW however, significantly improves the alignment accuracy compared to Base (G) by effectively utilising both predicted and ground truth path information. DecDTW is able to slightly improve the alignment accuracy of CQT and chroma features and dramatically improves the accuracy of melspec feature alignments. In fact, DecDTW is able to utilise

[1]The randomly initialised GRU feature extractor also degrades alignment accuracy compared to base features.

the additional information in the melspec features to surpass the performance of CQT and chroma. Figure 4a illustrates an example test alignment, with more examples provided in the appendix.

7.2 Transfer Learning for Sequence-based Visual Place Recognition

Our second experiment relates to the Visual Place Recognition (VPR) problem, which is an active area of research in computer vision (Berton et al., 2022; Arandjelovic et al., 2016) and robotics (Garg et al., 2021; Lowry et al., 2015; Cummins & Newman, 2011) and is often an important component of the navigation systems found in mobile robots and autonomous cars. The VPR task involves recognising the approximate geographic location where an image is captured, typically to a tolerance of a few meters (Zaffar et al., 2021; Berton et al., 2022) and is commonly formulated as image retrieval, where a query image is compared to a database of geo-tagged reference images. VPR using image sequences has been shown to improve retrieval performance over single image methods (Garg & Milford, 2021; Xu et al., 2020; Stenborg et al., 2020). In this experiment we first formulate sequence-based VPR as a temporal alignment problem and then fine-tune a deep feature extractor for this alignment task.

Problem formulation Let $\{I_i^r\}_{i=1}^N$, $\mathbf{t}^r \in \mathbb{R}^N$ and $\{\mathbf{x}_i^r\}_{i=1}^N$ be a reference image sequence, associated (sorted) timestamps and geotags, respectively. Similarly, let $\{I_i^q\}_{i=1}^M$, $\mathbf{t}^q \in \mathbb{R}^M$ and $\mathbf{X}^q = \{\mathbf{x}_i^q\}_{i=1}^N$ be the equivalent for a query sequence, noting that query geotags are used only for training and evaluation. Furthermore, when using deep networks for VPR, we have a feature extractor f_θ with weights θ that is used to extract embeddings $\mathbf{Z}^r = \{f_\theta(I_i^r)\}_{i=1}^N \in \mathbb{R}^{N\times d}$ and $\mathbf{Z}^q = \{f_\theta(I_i^q)\}_{i=1}^M \in \mathbb{R}^{M\times d}$ from reference and query images, respectively. Finally, let $\mathbf{x}^r$, $\mathbf{x}^q$, $\mathbf{z}^r$, $\mathbf{z}^q$ be continuous time signals built from geotags and image embeddings using associated timestamps.

We can formulate sequence-based VPR as a temporal alignment problem, where the objective is to minimise the discrepancy between the predicted alignment derived from the query and reference image embeddings and the optimal alignment derived from the geotags. Concretely, the goal of our learning problem is to minimise w.r.t. network weights θ, a measure of localisation error $G(\mathbf{x}^r(\boldsymbol{\phi}), \mathbf{X}^q)$, where $\boldsymbol{\phi} = \text{DTW}(\mathbf{z}^r, \mathbf{z}^q; \theta)$ defined as $\phi_i = \phi(t_i^q)$ is the estimated alignment which maps query images to positions in the reference sequence and $\mathbf{x}^r(\boldsymbol{\phi}) = \{\mathbf{x}^r(\phi_i)\}_{i=1}^M$. We select G as the (squared) maximum error over the queries, given by $G(\mathbf{X}^1, \mathbf{X}^2) = \max\{\|\mathbf{x}_i^1 - \mathbf{x}_i^2\|_2^2 : i = 1, \ldots, M\}$.

Experimental setup We source images and geotags from the Oxford RobotCar dataset (Maddern et al., 2017), commonly used as a benchmark to evaluate VPR methods. This dataset is comprised of autonomous vehicle traverses of the same route captured over two years across varying times of day and environmental conditions. We use a subset of ∼1M images captured across 50+ traverses provided by Thoma et al. (2020) with geotags taken from RTK GPS (Maddern et al., 2020) where available, accurate to 15cm. These images are split into training, validation and test sets, which are used to train a base network for single image retrieval as well as our sequence-based methods. The train and validation sets are captured in the same geographic areas on distinct dates whereas the test set is captured in an area geographically disjoint from validation and training.

Paired sequences are generated using the GPS ground truth; reference sequences have 25 images spaced ∼5m apart and query sequences have 10 images sampled at ∼1Hz. This setup is close to a deployment scenario where geotags are available for the reference image and query images exhibit large changes in velocity compared to the reference. Subsequence GDTW is used to align queries to the reference sequence. In total, we use ∼22k, ∼4k, ∼1.7k sequence pairs for training, validation and testing, respectively. Our test set contains a diverse set of sequences over a large geographic area, and exhibits lighting, seasonal and structural change between reference and query. We provide our paired sequence dataset in the supplementary material. Finally, for feature extraction, we use a VGG backbone with NetVLAD aggregation layer (Arandjelovic et al., 2016) (yields 32k-dim embeddings) and train the single image network using triplet margin loss. We fine-tune the conv5 and NetVLAD layers for sequence training. See the appendix for a detailed description of the training setup.

Results We evaluate all methods by measuring the proportion of test sequences where the maximum error is below predefined distance thresholds; these metrics are commonly used for single image methods (Garg et al., 2021; Berton et al., 2022). Results are split into three different environmental conditions including overcast, sunny and snow. We present results for two methods using the base single image network (1-2) and four methods which perform fine-tuning of the base network for sequence VPR (3-6): *1-2)* Base features (no sequence fine-tuning) with DTW (D) and GDTW (G) alignment, *3)* Fine-tuning using OTAM Cao et al. (2020) (Soft-DTW w/subsequence alignment) loss,

4) Feature cost along the ATE minimising alignment (Along GT) *5)* DILATE loss and *6)* max error loss (DecDTW). Similar to the audio experiments, DTW was used to produce alignments for methods 3-5) and GDTW was used for 6). For DILATE, we again modified the path penalty matrix $\mathbf{\Omega}$ to encode the deviation from the ground truth path. Table 2 summarises the results.

Table 2: VPR experiment results (maximum error): Test Accuracy @2/3/5/10m

Method	Overcast	Sunny	Snow
Base (D)	0.7/12.3/44.3/73.8	0.5/8.7/40.2/62.2	1.6/11.1/44.3/62.3
Base (G)	10.2/31.7/65.6/**90.3**	9.0/33.2/75.1/**99.3**	6.2/25.7/73.2/**100.0**
OTAM/Along GT	0.7/12.3/44.3/73.8	0.5/8.7/40.2/62.2	1.6/11.1/44.3/62.3
DILATE	1.0/11.4/44.3/73.6	1.5/9.7/39.7/59.8	1.8/11.8/46.1/62.7
DecDTW	**21.5/41.6/67.6**/89.1	**24.0/50.6/81.6**/98.3	**14.4/42.6/86.0**/99.1

We found that fine-tuning with the OTAM and Along GT losses is not able to reduce localisation error over the validation set compared to using base features. This is because the OTAM loss does not use GPS ground-truth and the Along GT loss does not use the predicted alignment. DILATE, which utilises path information, yields some degree of improvement over Base (D). However, training with DecDTW to directly reduce the maximum GPS error between predicted and ground truth yields *significant* improvements compared to Base (G) across tighter (< 5m) error tolerances. We provide more results and analysis for the VPR task in the appendix. Figure 4b illustrates an example of a test sequence before and after fine-tuning, with more examples provided in the appendix.

7.3 Scalability

Figure 5 illustrates how inference at test time scales with time series length n for GDTW (required for DecDTW) compared to DTW. Timings are evaluated on an Nvidia GeForce GTX 1080Ti 11Gb. GDTW is between 15-50 times slower than DTW, with the gap increasing with length n. The additional computation is due in large part to the additional discretisation required to solve for the finer-grained continuous time problem defined in Section 4. DTW solves a DP with $O(mn)$ nodes and $O(mn)$ edges, whereas GDTW solves a DP with $O(mM)$ edges and $O(mM^2)$ edges (we set $M = n$). However, by removing rounds of warp refinement, we can reduce this gap to a 4-12x slowdown with an mean accuracy loss of only 0.5%. Further discussion and a comparison of train-time scalability is provided in the Appendix.

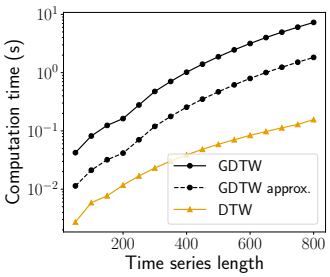


Figure 5: Test time scalability

8 Conclusion and Future Works

We present a novel formulation for a differentiable DTW layer that can be embedded into a deep network, based on continuous time warping and implicit deep layers. Unlike existing approaches, DecDTW yields the *optimal* alignment path as a differentiable function of layer inputs. We demonstrate the effectiveness of DecDTW in utilising path information in challenging, real-world alignment tasks compared to existing methods. One promising direction for future work is in exploring more compact, alternative parameterisations of the warp function to improve scalability, inspired by Zhou & De la Torre (2012). Another direction would be to integrate more sophisticated DTW variants such as jumpDTW (Fremerey et al., 2010), which allows for repeats and discontinuities in the warp, into the GDTW (and DecDTW) formulation. Finally, we presented methodology for allowing regularisation and constraints to be learnable parameters in a deep network but did not explore this in our experiments. Future work will also explore this capability in more detail.

References

Abubakar Abid and James Y Zou. Learning a Warping Distance from Unlabeled Time Series Using Sequence Autoencoders. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019a.

A. Agrawal, S. Barratt, S. Boyd, E. Busseti, and W. Moursi. Differentiating through a Cone Program. *Journal of Applied and Numerical Optimization*, 1(2):107–115, 2019b.

Brandon Amos and J. Zico Kolter. OptNet: Differentiable Optimization as a Layer in Neural Networks. In *Proc. of the International Conference on Machine Learning (ICML)*, 2017.

Brandon Amos, Ivan Jimenez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable MPC for End-to-end Planning and Control. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. NetVLAD: CNN architecture for weakly supervised place recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Anthony Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31, 2017.

Donald J. Berndt and James Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Proc. of the 3rd International Conference on Knowledge Discovery and Data Mining*, 1994.

Gabriele Berton, Riccardo Mereu, Gabriele Trivigno, Carlo Masone, Gabriela Csurka, Torsten Sattler, and Barbara Caputo. Deep Visual Geo-localization Benchmark. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Xingyu Cai, Tingyang Xu, Jinfeng Yi, Junzhou Huang, and Sanguthevar Rajasekaran. DTWNet: a Dynamic Time Warping Network. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Dylan Campbell, Liu Liu, and Stephen Gould. Solving the Blind Perspective-n-Point Problem End-To-End With Robust Differentiable Geometric Optimization. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2020.

Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-Shot Video Classification via Temporal Alignment. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Chien-Yi Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3TW: Discriminative Differentiable Dynamic Time Warping for Weakly Supervised Action Alignment and Segmentation. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Xiaobin Chang, Frederick Tung, and Greg Mori. Learning Discriminative Prototypes with Dynamic Time Warping. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Bo Chen, Alvaro Parra, Jiewei Cao, Nan Li, and Tat-Jun Chin. End-to-End Learnable Geometric Vision by Backpropagating PnP Optimization. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Anoop Cherian, Basura Fernando, Mehrtash Harandi, and Stephen Gould. Generalized Rank Pooling for Activity Recognition. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proc. of the Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST-8)*, 2014.

Samuel Cohen, Giulia Luise, Alexander Terenin, Brandon Amos, and Marc Deisenroth. Aligning time series on incomparable spaces. In *Proc. of The International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2021.

Mark Cummins and Paul Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 30, 2011.

Marco Cuturi and Mathieu Blondel. Soft-DTW: a Differentiable Loss Function for Time-Series. In *Proc. of the International Conference on Machine Learning (ICML)*, 2017.

Dave Deriso and Stephen Boyd. A general optimization framework for dynamic time warping. *arXiv preprint arXiv:1905.12893*, 2019.

Mikita Dvornik, Isma Hadji, Konstantinos G Derpanis, Animesh Garg, and Allan Jepson. Drop-DTW: Aligning Common Signal Between Sequences While Dropping Outliers. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Sebastian Ewert, Meinard Muller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2009.

Basura Fernando and Stephen Gould. Learning End-to-end Video Classification with Rank-Pooling. In *Proc. of the International Conference on Machine Learning (ICML)*, 2016.

Basura Fernando and Stephen Gould. Discriminatively Learned Hierarchical Rank Pooling Networks. *International Journal of Computer Vision*, 2017.

Christian Fremerey, Meinard Müller, and Michael Clausenr. Handling Repeats and Jumps in Score-Performance Synchronisation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

Sourav Garg and Michael Milford. SeqNet: Learning descriptors for sequence-based hierarchical place recognition. *IEEE Robotics and Automation Letters*, 6, 2021.

Sourav Garg, Tobias Fischer, and Michael Milford. Where Is Your Place, Visual Place Recognition? In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Damien Garreau, Rémi Lajugie, Sylvain Arlot, and Francis Bach. Metric Learning for Temporal Sequence Alignment. In *Advances in Neural Information Processing Systems*, 2014.

Stephen Gould, Richard Hartley, and Dylan Campbell. Deep Declarative Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

Josif Grabocka and Lars Schmidt-Thieme. Neuralwarp: Time-series similarity with warping networks. *arXiv preprint arXiv:1812.08306*, 2018.

Alex Graves and Navdeep Jaitly. Towards end-to-end speech recognition with recurrent neural networks. In *Proc. of the International Conference on Machine Learning (ICML)*, 2014.

Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation Learning via Global Temporal Alignment and Cycle-Consistency. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram N. Syed, Andrey Konin, Zeeshan Zia, and Quoc-Huy Tran. Learning by Aligning Videos in Time. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

Hicham Janati, Marco Cuturi, and Alexandre Gramfort. Spatio-temporal alignments: Optimal transport through space and time. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

Ieva Kazlauskaite, Carl Henrik Ek, and Neill Campbell. Gaussian Process Latent Variable Alignment Learning. In *Proc. of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

Lucas Kovar and Michael Gleicher. Flexible Automatic Motion Blending with Registration Curves. In *Proc. of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 2003.

Vincent Le Guen and Nicolas Thome. Shape and time distortion loss for training deep time series forecasting models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-Learning With Differentiable Convex Optimization. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Suhas Lohit, Qiao Wang, and Pavan Turaga. Temporal Transformer Networks: Joint Learning of Invariant and Discriminative Time Warping. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32, 2015.

Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The Oxford RobotCar dataset. *The International Journal of Robotics Research*, 36, 2017.

Will Maddern, Geoffrey Pascoe, Matthew Gadd, Dan Barnes, Brian Yeomans, and Paul Newman. Real-time kinematic ground truth for the oxford robotcar dataset. *arXiv preprint arXiv:2002.10152*, 2020.

Marion Morel, Catherine Achard, Richard Kulpa, and Séverine Dubuisson. Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognition*, 74, 2018.

Saul B. Needleman and Christian D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48, 1970.

Nicola Orio, Diemo Schwarz, and Ircam Pompidou. Alignment of monophonic and polyphonic music to a score. In *International Computer Music Conference (ICMC)*, 2001.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In *Proc. of the IEEE International Conference on Data Mining*, 2014.

Chotirat Ratanamahatana and Eamonn Keogh. Making time-series classification more accurate using learned constraints. In *Proc. of the Fourth SIAM International Conference on Data Mining*, 2004.

H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26, 1978.

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. Visual Permutation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.

Craig Sapp. Online database of scores in the humdrum file format. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, 2005.

Shai Shalev-Shwartz, Joseph Keshet, and Yoram Singer. Learning to align polyphonic music. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, 10 2004.

Ron A Shapira Weber, Matan Eyal, Nicki Skafte, Oren Shriki, and Oren Freifeld. Diffeomorphic Temporal Alignment Nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Erik Stenborg, Torsten Sattler, and Lars Hammarstrand. Using Image Sequences for Long-Term Visual Localization. In *Proc. of the International Conference on 3D Vision (3DV)*, 2020.

Bing Su and Ying Wu. Learning distance for sequences by learning a ground metric. In *Proc. of the International Conference on Machine Learning (ICML)*, 2019.

John Thickstun, Jennifer Brennan, and Harsh Verma. Rethinking evaluation methodology for audio-to-score alignment. *arXiv preprint arXiv:2009.14374*, 2020.

Janine Thoma, Danda Pani Paudel, and Luc V Gool. Soft Contrastive Learning for Visual Localization. In *Advances in Neural Information Processing Systems*, 2020.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17, 2020.

Yizhi Wang, David J Miller, Kira Poskanzer, Yue Wang, Lin Tian, and Guoqiang Yu. Graphical time warping for joint alignment of multiple curves. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

Ming Xu, Niko Snderhauf, and Michael Milford. Probabilistic visual place recognition for hierarchical localization. *IEEE Robotics and Automation Letters*, 6, 2020.

Mubariz Zaffar, Sourav Garg, Michael Milford, Julian Kooij, David Flynn, Klaus McDonald-Maier, and Shoaib Ehsan. VPR-Bench: An Open-Source Visual Place Recognition Evaluation Framework with Quantifiable Viewpoint and Appearance Change. *International Journal of Computer Vision*, 129, 2021.

Zheng Zhang, Romain Tavenard, Adeline Bailly, Xiaotong Tang, Ping Tang, and Thomas Corpetti. Dynamic Time Warping under limited warping path length. *Information Sciences*, 393, 2017.

Feng Zhou and Fernando De la Torre. Canonical Time Warping for Alignment of Human Behavior. In *Advances in Neural Information Processing Systems (Neurips)*, 2009.

Feng Zhou and Fernando De la Torre. Generalized Time Warping for Multi-modal Alignment of Human Motion. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Yunyue Zhu and Dennis Shasha. Warping Indexes with Envelope Transforms for Query by Humming. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2003.

A APPENDIX

We provide additional implementation details for DecDTW, as well as additional qualitative examples for both experiments in this section.

A.1 DYNAMIC PROGRAMMING SOLVER FOR GDTW

The mechanics of the solver are described as follows. For each i we can discretise ϕ_i into M values $\{\phi_{i,j}\}_{j=1}^{M}$ uniformly between global bounds $b_i^{\min}, b_i^{\max}$. This forms a graph where each of the mM values correspond to nodes and furthermore, temporally adjacent nodes, i.e., $\phi_{i-1,j}, \phi_{i,k}$ for all j, k are connected with edges, for a total of $(m-1) \times M^2$ edges. Node costs correspond to signal loss values in Equation 5 and edge costs correspond to warp regularisation values in Equation 6. Edges which violate the local constraints $s_i^{\min}, s_i^{\max}$ are given a cost of ∞. The global minima of this new discrete optimisation problem corresponds to the minimum cost path through the graph and is solved using dynamic programming in $O(mM^2)$ time complexity. Iterative refinement of the

discretisation and solution is performed as described in Deriso & Boyd (2019). We use three rounds of discretisation with discretisation factor $\eta = 0.125$ in all of our experiments. Figure 6 illustrates the mechanics of the DP solver for a subsequence alignment problem and shows that the solution after refinement is comparable to calling an SLSQP solver (Virtanen et al., 2020) directly.

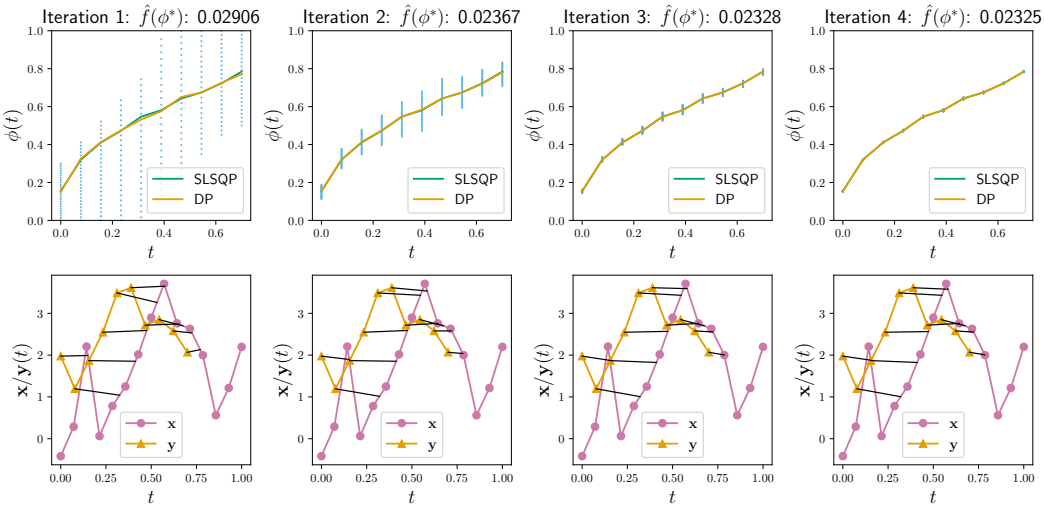


Figure 6: Illustration of the DP solver mechanics on a subsequence alignment problem, where ϕ maps times in the shorter sequence $\mathbf{y}$ to longer sequence $\mathbf{x}$. A uniform discretisation is generated between global constraints in the first iteration. Subsequent discretisations are made between tighter bounds around the previous solution. The objective function value decreases at each iteration until convergence after 3 iterations. The objective value $\hat{f}$ from the SLSQP solver (Virtanen et al., 2020) is 0.02325, close to the DP solver at the final iteration and furthermore, we can see qualitatively that the solved warps are indistinguishable between the DP and SLSQP approach.

A.2 Efficient Computation of Backward Pass

We do not construct the full Jacobian $D\phi(\mathbf{z})$ from Equation 7 explicitly, instead directly computing the vector-Jacobian products required for automatic differentiation libraries. Let $J(\boldsymbol{\phi})$ be a downstream loss function defined over the estimated alignment (e.g., MSE to a ground truth alignment $\boldsymbol{\phi}^{\text{gt}}$). Our goal for the end-to-end model is to compute $DJ(\mathbf{z}) = v^\top D\boldsymbol{\phi}(\mathbf{z})$, where $v^\top = J(\boldsymbol{\phi}) \in \mathbb{R}^{1\times m}$. To do this, we evaluate $v^\top D\boldsymbol{\phi}(\mathbf{z})$ from left to right, pre-computing and caching $H^{-1}A^\top$ beforehand. In addition, note that H is tridiagonal from the definition of $\hat{f}$, where off-diagonal elements correspond only to the warp regularisation component $\hat{\mathcal{R}}$. This allows us to solve $v^\top H^{-1}$ in $O(m)$ time. Finally, observe that B and C are in a sense complementary to each other. Specifically, columns of B relating to $\mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}}$ are zero since the objective function $\hat{f}$ only depends on $\mathbf{x}, \mathbf{y}, \lambda$. Conversely, columns of C relating to $\mathbf{x}, \mathbf{y}, \lambda$ are zero since the constraint functions only depend on $\boldsymbol{\phi}$ and $\mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}}$. This allows efficient evaluation of Equation 7 w.r.t the two blocks $\{\mathbf{x}, \mathbf{y}, \lambda\}$ and $\{\mathbf{s}^{\text{min}}, \mathbf{s}^{\text{max}}, \mathbf{b}^{\text{min}}, \mathbf{b}^{\text{max}}\}$ separately by setting C (resp. B) to zero.

A.3 Train Time Scalability Analysis

We present results for train time scalability in this section using the same experimental setup as the test time scalability analyisis presented in Section 5. Unlike at test time, we aren't able to trade-off warp estimation accuracy for computation time during training. This is because the backward pass computation given by Equation 7, assumes the estimated warp $\phi^\star$ satisfies the first-order optimality conditions for the NLP

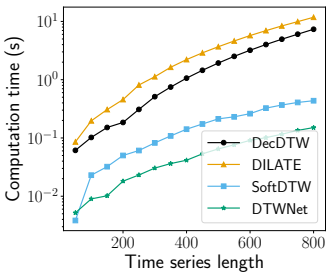


Figure 7: Train time scalability

defined in Equation 4. To ensure this assumption holds, we need to ensure our solver discretises ϕ finely enough such that $\phi^\star$ is sufficiently close to an optimal point (up to a suitable numerical tolerance). In practice, this can be achieved with multiple rounds of iterative refinement (we use three in our experiments, as described in Section 6) and a sufficiently large discretistion level M (again, we set $M = n$ in this analysis).

Figure 7 illustrates the results of our train time scalability analysis. Note, the computation times presented include the *total* time taken for a training iteration, including the forward and backward pass. We observe that a training iteration of DecDTW is 4-20x (resp. 15-50x) slower than SoftDTW (resp. DTWNet), due to the iterative warp refinement required for fine-grained warp estimation in the DecDTW forward pass. However, we observe that DILATE, a relevant method which allows for differentiable path information under SoftDTW, is in fact slower for training compared to DecDTW; we believe this is attributed to our backward pass being explicitly computed rather than computed through unrolling the forward pass.

A.4 Experimental Setup for Audio-to-Score Alignment

Dataset Description Ground truth alignments were generated between the 193 piano performances to their associated scores using the methodology proposed in Thickstun et al. (2020). We used the authors' default hyperparameters for ground truth generation. These 193 performances are divided into train (94), validation (49) and test (50) splits. Furthermore, scores are not shared across splits. Audio is sampled at 22.1kHz with a hop size of 1024 for generating a uniformly spaced time series of base features for each performance and synthesised score. Furthermore, each performance is split into uniform, non-overlapping slices of length 256, which corresponds to $\sim$12 seconds of audio per slice. There were a total of 995, 565 and 541 slices, across the train, validation and test splits, respectively. The hyperparameters used to generate constant-Q transform (48-dim) and chromagram (12-dim) features are identical to the ones used in Thickstun et al. (2020). For the mel spectrogram features, we used 128 mel bands, yielding a 128-dim feature. We provide the full dataset generation procedure in our code.

Learning Task For learned methods (3-6), we apply a feature extractor comprised of a single layer bi-directional GRU Cho et al. (2014) (with random initialisation) on top of base audio features. This feature extractor operates as a sequence-to-sequence model. Specifically, the GRU layer takes as input a length N sequence of base audio features of shape $\mathbb{R}^{N \times d}$, and outputs learned features of shape $\mathbb{R}^{N \times d_1}$, where the output dimension of the GRU layer is set as $d_1 = 128$ in our experiments. The learned features are then L2-normalised before being used to compute both a downstream alignment and training loss for each learned comparison method. Alignments in Table 1 are generated using DTW for Base (D), SoftDTW, Along GT, DILATE and using GDTW for Base (G) and DecDTW.

Comparison Methods The training losses for (3-6) are described as follows: The SoftDTW (Cuturi & Blondel, 2017) method (3) computes the SoftDTW discrepancy between two feature sets and utilises no path information. Along GT (4) computes the discrepancy between features along the path given by ground truth alignments. DILATE (Le Guen & Thome, 2019) (5) is a hybrid loss which consists of a *shape loss* (equivalent to the SoftDTW discrepancy) and a *temporal loss*, which penalises deviations from the predicted and ground truth paths. The relative weighting between both losses is given by a hyperparameter $0 \leq \alpha \leq 1$. Ground truth path information in DILATE is given by a penalty matrix $\mathbf{\Omega}$. Elements within each row (corresponding to a single point in time in the score) of $\mathbf{\Omega}$ (where ground truth alignments are defined) is given a penalty term based on the absolute alignment error. DecDTW (6) is able to train on the $\mathrm{TimeErr}$ loss directly. Full implementation for DecDTW and all comparison methods for this experiment are provided in our code.

Training Hyperparameters We use the Adam (Kingma & Ba, 2015) optimiser with a learning rate of 0.0001 and a batch size of 5 for all methods. We trained for 20 epochs and selected the model which yielded minimum $\mathrm{TimeErr}$ over the validation set for all methods. For DecDTW and Base (G), we set GDTW the regularisation hyperparameter $\lambda = 0.1, 0.7, 0.9$ for CQT, chroma and melspec features, respectively. These regularisation weight values were selected using a grid search over the average $\mathrm{TimeErr}$ across the validation set for the base features. For the SoftDTW (Cuturi &

Blondel, 2017) comparison method we used $\gamma = 1$, noting that results do not meaningfully change for different γ. For DILATE, we found that using $\alpha = 0$ (temporal loss only) and $\gamma = 1$ yielded the best results. Setting too low γ caused learning to fail and increasing γ past 1 did not improve results.

A.5 Experimental Setup for Visual Place Recognition

Construction of Paired Sequences Dataset The single-image dataset used to train the baseline VPR retrieval network is used to bootstrap the paired sequences dataset used for sequence-based fine-tuning. Within each train/val/test split, GPS data is used to produce sequence pairs. We designated a fixed set of reference traverses (6 for training, 1 each for validation and test), with each traverse having an associated full data collection run with a given date/time identifier (e.g., 2015-08-17-13-30-19). Each reference traverse is split into geographically overlapping but distinct reference sequences of length 25, with images spaced 5 meters apart within each reference sequence. Adjacent reference sequences within the same traverse have 20 meters overlap between them for training and 12 meters overlap for validation and test splits. For each reference sequence, five query sequences are sampled. Query sequences are randomly selected from the pool of available remaining (i.e. non-reference) traverses within each split. The starting location of each query sequence is chosen randomly such that the query sequence is geographically contained by the reference sequences.

Our resultant dataset, comprised of $\sim$22k, $\sim$4k, $\sim$1.3k sequence pairs for training, validation and testing, respectively, full covers the geographic region and a large set of conditions encompassed by the RobotCar dataset. A large variety of sequence pairs spanning different locations and condition pairs between query and reference enables features learned by sequence fine-tuning to generalise between training and the unseen (both in geographic location and date) test set. We also ensure that the large number of and uniformly spaced test sequences allow for a comprehensive and challenging evaluation. While we can arbitrarily increase the size of this dataset, we find the variety present in our bootstrapped dataset is enough to yield significant gains for sequence fine-tuning. Full dataset lists for the paired sequence datasets are provided in the supplementary material.

Comparison Methods Similar to the audio experiments, all learned sequence fine-tuning methods apply a loss on top of sequences of extracted image embeddings. The OTAM (Cao et al., 2020) loss is the SoftDTW loss with a minor modification to allow for subsequence alignment. Along GT computes the discrepancy between features along the path given by ground truth alignments. For DILATE (Le Guen & Thome, 2019), similar to the audio experiments, we needed to parameterise the penalty matrix $\mathbf{\Omega}$. The value for element i, j is given by the (squared) GPS error (in meters) between the geotages of query image i and reference image j. Finally, DecDTW is able to use the *task loss*, i.e. the *maximum* (or mean) GPS error along the *optimal* alignment to improve localisation performance.

Hyperparameters for Single Image VPR We train single image NetVLAD representations with VGG-16 backbone pretrained on ImageNet. NetVLAD layer uses 64 clusters, leading to a 32,768 dimensional time series embedding. Images are resized to 240 $\times$ 180 (width $\times$ height), consistent with Thoma et al. (2020). Triplet loss margin is set to 0.1 and 10 negatives per anchor-positive pair are used. The model is trained using the Adam optimizer with a learning rate of 0.00001 and a batch size of 4. Our training setup is based on the recently released visual geolocalization benchmark (Berton et al., 2022) and its associated codebase[2]. We provide training and dataset setup code in the supplementary materials.

Hyperparameters for Sequence-based VPR For the sequence fine-tuning experiments, we use a learning rate of 0.0001 and a batch size of 8 for a maximum of 30 epochs across all methods. In contrast to the audio experiments, query and reference images are both fed through a shared feature extractor comprised of a VGG-16 backbone, followed by a NetVLAD aggregation layer. We checkpoint models during training based on the average maximum GPS error over the full validation set five times per epoch. We selected a fixed regularisation weight of $\lambda = 0.1$ for DecDTW using the validation set and embeddings from the single-image model. For OTAM Cao et al. (2020), we selected $\gamma = 1$, similar to the audio experiments, noting results were insensitive to this parameter.

[2]https://github.com/gmberton/deep-visual-geo-localization-benchmark

This again holds true for DILATE Le Guen & Thome (2019), where similar to the audio experiments, we set $\alpha = 0$ and $\gamma = 1$.

A.6 Additional Results for Audio-to-Score Alignment Experiments

In this section, we investigate if features learned using each comparison method presented in Section 7.1 generalises across both DTW and GDTW alignments at test time. In Table 3, we present results for each comparison method using only DTW alignments at test time. Notably, the main difference to Table 1 is the DecDTW column; the features output by the GRU(s) trained using $\mathrm{TimeErr}$ loss are placed into a DTW alignment layer (as opposed to GDTW) during testing. We can see in Table 3 that features learned with DecDTW produce poorer alignments using DTW for both CQT and chroma features when compared to DILATE. In addition, test alignment performance using DTW is significantly poorer compared to using GDTW (see last column in Table 4). Surprisingly, DecDTW learned features for melspec seem to generalise and perform better than all other comparison methods for DTW alignment at test time (including base features).

Table 3: Results for the audio-to-score alignment experiments (DTW alignment)

Reported metrics are TimeErr / TimeDev (ms)					
Feature Type	Base (D)	SoftDTW	Along GT	DILATE	DecDTW
CQT	**35** / 90	49 / 130	49 / 130	37 / **69**	100 / 210
Chroma	50 / 115	59 / 142	59 / 142	**47 / 79**	45 / 117
Melspec	122 / 235	55 / 153	52 / 145	60 / 100	**31 / 76**

We also present results in Table 4, which evaluate features learned under each comparison method using only GDTW alignments at test time. Features learned with methods based on DTW (i.e. SoftDTW, Along GT, DILATE) perform worse than base features for GDTW alignment. This holds for especially true for DILATE, which yielded accuracy gains when evaluating with DTW alignments compared to base features but significantly degraded accuracy for GDTW alignment. DecDTW as expected, yielded gains compared to base features when evaluated using GDTW alignments.

Table 4: Results for the audio-to-score alignment experiments (GDTW alignment)

Reported metrics are TimeErr / TimeDev (ms)					
Feature Type	Base (G)	SoftDTW	Along GT	DILATE	DecDTW
CQT	19 / 30	22 / 33	22 / 33	24 / 35	**17 / 27**
Chroma	24 / 39	34 / 51	34 / 51	243 / 266	**19 / 31**
Melspec	56 / 81	155 / 219	147 / 211	178 / 229	**16 / 27**

Overall, we conclude that features trained with an underlying alignment method (either DTW or GDTW) tends to specialise to the particular alignment method used in training. Note that the continuous time formulation of GDTW, which allows for interpolation of alignments, and the availability of regularisation λ, cause GDTW alignments to outperform DTW overall at test time across most combinations of methods and features.

A.7 Additional Results for Visual Place Recognition Experiments

In this section, we provide additional results to the ones presented in Table 2. First, we add results for the *independent retrieval* task. The independent retrieval task involves finding the most visually similar reference image for each query, independently, using image embeddings (no DTW optimisation). The base network used to initialise sequence-based fine-tuning is trained using a contrastive loss (specifically, triplet margin loss) over single images. As a result, this network is designed to maximise independent retrieval performance. The results are presented in Table 5. Interestingly, fine-tuning using a sequence-based loss tends to overall decrease independent retrieval performance. This holds uniformly and significantly for DecDTW and also holds in most cases for DILATE. This suggests a trade-off between sequence matching and independent retrieval performance. It is possible that the network has retrained to focus on different areas of the underlying images for the alignment task as opposed to the retrieval task.

Table 5: VPR experiment results (maximum error): Test Accuracy @2/3/5/10m

Method	Overcast	Sunny	Snow
Base (DTW)	0.7/12.3/44.3/73.8	0.5/8.7/40.2/62.2	1.6/11.1/44.3/62.3
Base (GDTW)	10.2/31.7/65.6/**90.3**	9.0/33.2/75.1/**99.3**	6.2/25.7/73.2/**100.0**
OTAM/Along GT	0.7/12.3/44.3/73.8	0.5/8.7/40.2/62.2	1.6/11.1/44.3/62.3
DILATE	1.0/11.4/44.3/73.6	1.5/9.7/39.7/59.8	1.8/11.8/46.1/62.7
DecDTW	**21.5/41.6/67.6**/89.1	**24.0/50.6/81.6**/98.3	**14.4/42.6/86.0**/99.1
Base (Indep.)	0.0/11.6/47.9/77.0	0.0/7.5/40.4/81.1	1.6/10.0/50.1/87.1
DILATE (Indep.)	0.2/9.4/45.5/72.9	0.2/10.2/43.8/74.3	1.6/9.8/50.1/80.9
DecDTW (Indep.)	0.0/9.9/36.6/63.7	0.2/9.9/35.8/56.7	0.4/6.0/39.7/77.2

We additionally present results for the *mean* pose error across the sequence (as opposed to the maximum in Table 5) in Table 6. This measure of positioning error is commonly referred to as the absolute trajectory error (ATE). The conclusions are equivalent to the maximum error case; fine-tuning on the GPS error directly using our declarative DTW layer yields considerable performance improvements at sub 5m tolerances over the comparison methods.

Table 6: Additional VPR experiment results (mean error): Test Accuracy @2/3/5/10m

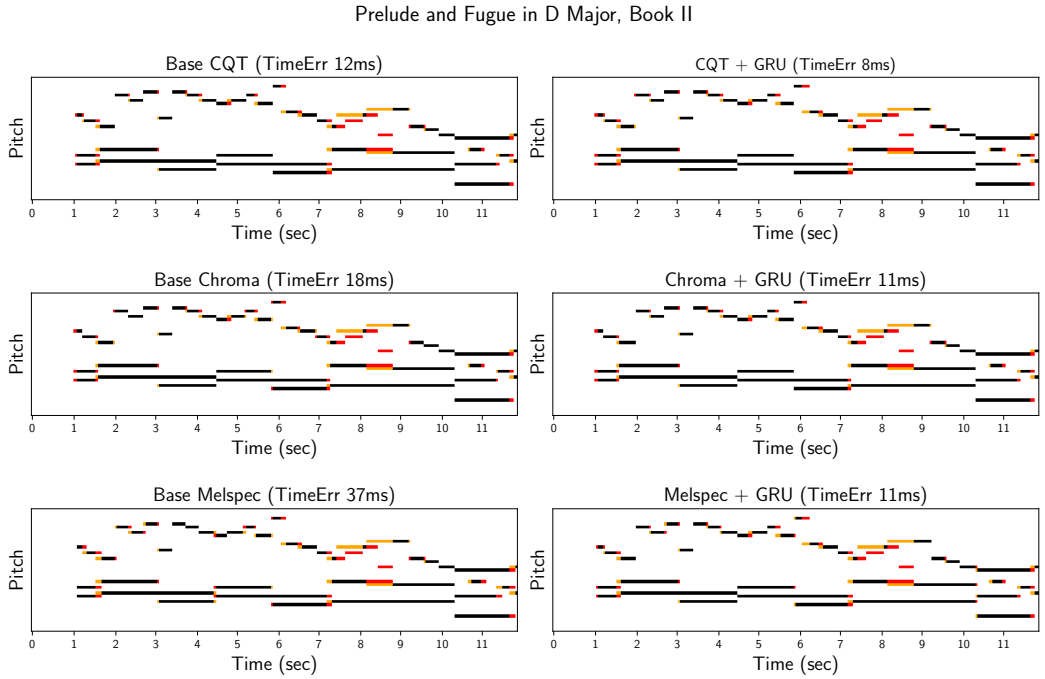
Method	Overcast	Sunny	Snow
Base (DTW)	16.9/48.9/70.0/94.7	16.2/42.9/62.0/91.8	16.2/42.9/62.0/91.8
Base (GDTW)	44.8/70.2/90.1/**98.1**	47.0/85.2/**99.5/99.8**	43.5/79.2/**100.0/100.0**
OTAM/Along GT	16.9/48.9/70.0/94.7	46.2/42.9/62.0/91.8	16.2/42.9/62.0/91.8
DILATE	16.7/49.6/67.6/88.4	16.9/41.2/59.3/84.5	14.9/42.8/61.4/90.9
DecDTW	**54.0/73.4/91.0**/96.1	**63.4/88.4**/98.5/**99.8**	**53.7/88.9**/98.5/**100.0**
Base (Indep.)	18.6/55.7/80.1/91.5	16.9/50.8/85.4/96.8	17.0/58.5/90.9/98.4
DILATE (Indep.)	19.1/51.6/77.7/84.5	26.2/53.5/77.5/89.6	18.0/52.8/84.7/97.1
DecDTW (Indep.)	15.5/43.6/66.8/80.9	17.4/43.8/59.3/72.6	10.2/49.7/78.9/91.6

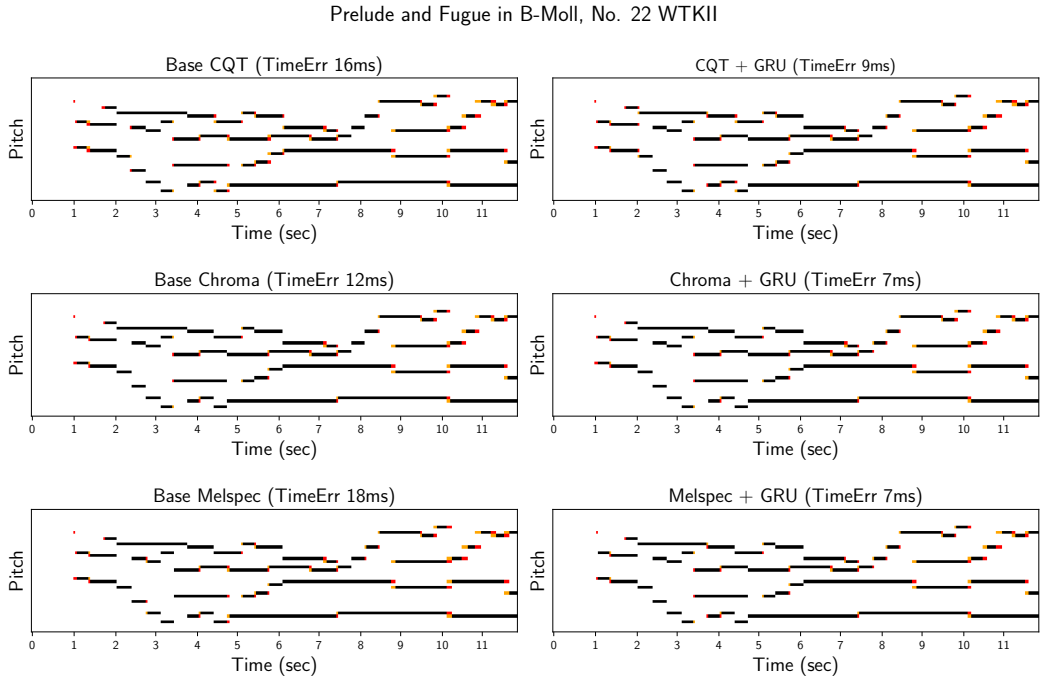
A.8 Additional Qualitative Examples for Audio-to-Score Alignment Experiments

In Figure 8 we present additional visualisations of example alignments for the audio to score alignment task. All features types (CQT, chroma, melspec) and before/after learning are presented in the figures. Example alignments are randomly sampled from the test set. The visualisations are a way proposed by Thickstun et al. (2020) to visually compare two alignments (e.g., an estimated and ground truth alignment) by plotting *performance aligned scores*, where a performance aligned score is generated by mapping the score through a warping function. We generate a performance aligned score using both the ground truth alignment and the estimated alignments using audio features. Red identifies notes indicated by audio feature alignments, but not by the ground-truth, and yellow identifies notes indicated by the ground-truth, but not by the audio features. Note, even if a predicted alignment is close to the ground truth (in a $\mathrm{TimeErr}$ sense), mistakes made by the performer may yield significant looking errors (orange and red) in the visualisations.

A.9 Additional Qualitative Examples for Visual Place Recognition Experiments

We provide additional qualitative examples for the VPR experiments in Figure 9. These figures illustrate the effect of DecDTW fine-tuning on resultant positioning accuracy for a randomly selected set of test sequence pairs. In addition, Figure 9 provides example test images to illustrate the amount of appearance change observed between reference and query images.


Prelude and Fugue in D Major, Book II
Base CQT (TimeErr 12ms)
CQT + GRU (TimeErr 8ms)
Base Chroma (TimeErr 18ms)
Chroma + GRU (TimeErr 11ms)
Base Melspec (TimeErr 37ms)
Melspec + GRU (TimeErr 11ms)
Pitch
Time (sec)
0 1 2 3 4 5 6 7 8 9 10 11


Prelude and Fugue in B-Moll, No. 22 WTKII
Base CQT (TimeErr 16ms)
CQT + GRU (TimeErr 9ms)
Base Chroma (TimeErr 12ms)
Chroma + GRU (TimeErr 7ms)
Base Melspec (TimeErr 18ms)
Melspec + GRU (TimeErr 7ms)
Pitch
Time (sec)
0 1 2 3 4 5 6 7 8 9 10 11

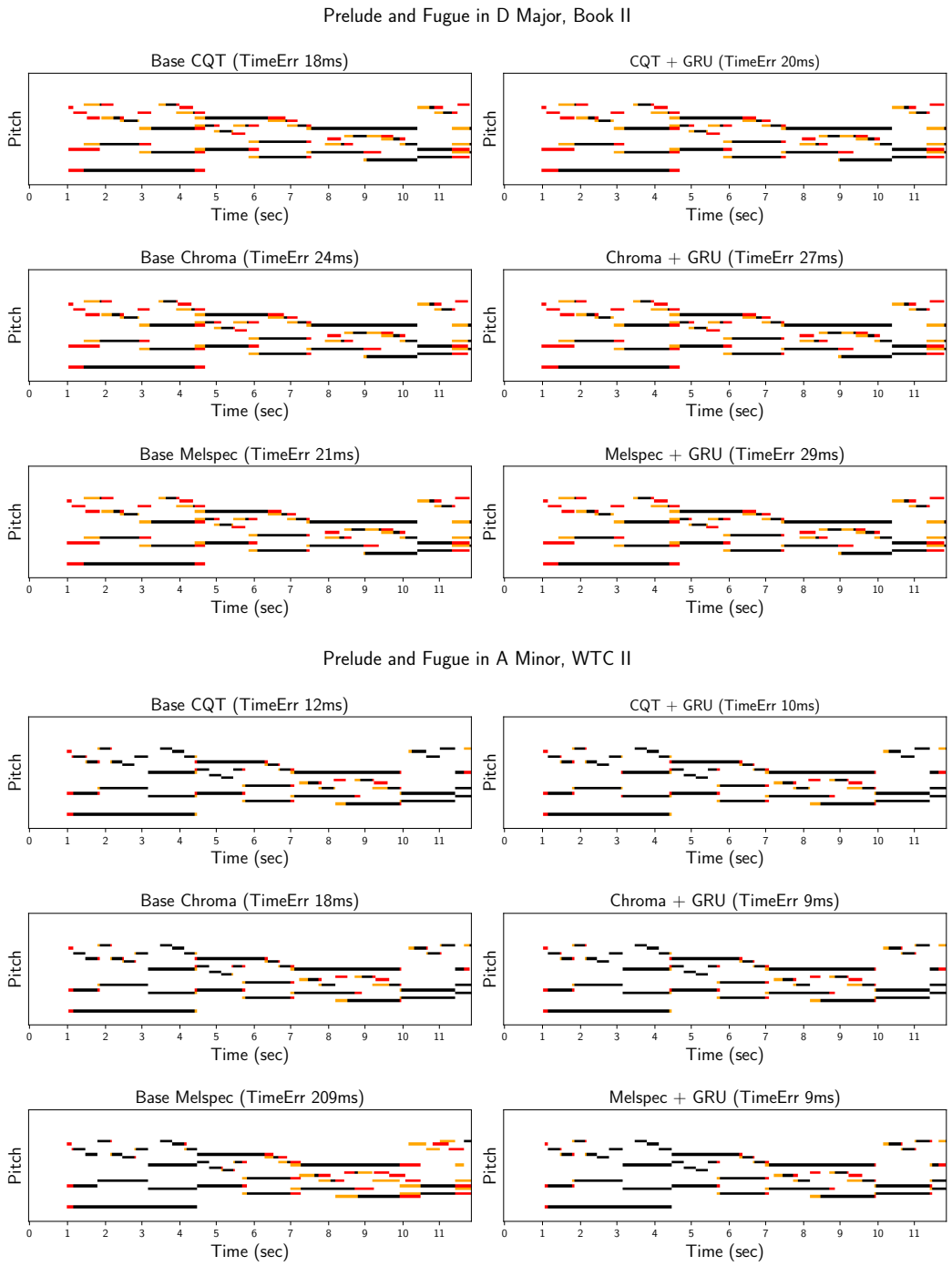


Figure 8: Example test set alignments for the audio-to-score alignment experiments. **Left:** Alignments produced using base features. **Right:** Alignments produced using DecDTW learned features.

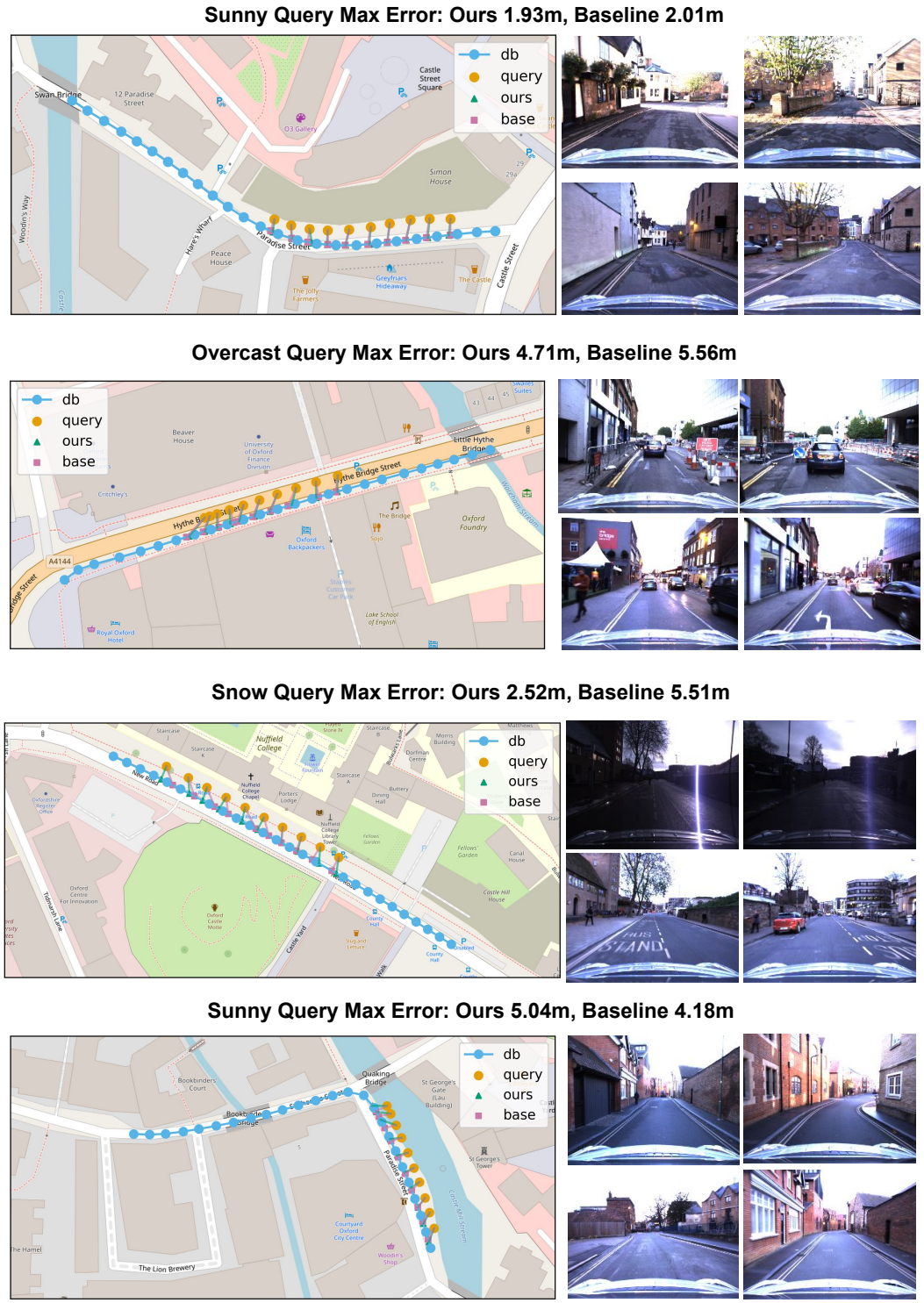


Figure 9: Additional example localisation output for the VPR experiments randomly selected from the test set. Example query images from the sequence are provided in the top row and example database images are provided in the bottom row. Query poses are offset from the database for visualisation purposes only.