Task-Driven Wavelets using Constrained Empirical Risk Minimization

Supplementary Material

6. Supplementary information CERM

In this appendix, we provide the mathematical details of our CERM framework. In Sec. 6.1, we provide a proof of Theorem 2.2. Next, we briefly compare our method to that of Lagrange multipliers in Sec. 6.2. In the subsequent sections, Sec. 6.4 to Sec. 6.6, we provide the computational details for performing SGD on a Riemannian manifold defined by a finite system of equations.

6.1. Proof of Theorem 2.2

Theorem 6.1. *If zero is a regular value of F, then the CERM problem in (2) is equivalent to solving an ordinary ERM problem on a Riemannian manifold $(\mathcal{N}, g_{\mathcal{N}})$ of dimension $p-q$. Here $\mathcal{N} = \mathbb{R}^{p-\tilde{p}} \times \mathcal{M}$ is an embedded C^2-submanifold of $\mathbb{R}^p$ and $\mathcal{M} := F^{-1}(0)$. The equivalent minimization problem is given by*

$$\min_{(\alpha,\theta)\in\mathcal{N}} \mathbb{E}\left(L\left(G\left(X, \alpha \oplus \iota(\theta)\right), Y\right)\right), \tag{7}$$

where $\iota : \mathcal{M} \to \mathbb{R}^{\tilde{p}}$ is the inclusion map.

Proof. The solution set $\mathcal{M} := F^{-1}(0)$ is an embedded C^2-submanifold of $\mathbb{R}^{\tilde{p}}$ of dimension $\tilde{p} - q$ by the Implicit Function Theorem, since zero is a regular value of F. A detailed review of this statement is provided in Theorem 6.2. Since $\mathcal{M}$ is naturally embedded in $\mathbb{R}^{\tilde{p}}$, we may endow it with the pull-back metric $g_{\mathcal{M}}$, turning it into a Riemannian manifold $(\mathcal{M}, g_{\mathcal{M}})$. Here $g_{\mathcal{M}} := \iota^* g_{\text{flat}}$, where g_{flat} is the standard Euclidean metric on $\mathbb{R}^{\tilde{p}}$. The constrained ERM problem can now be reformulated as an ordinary ERM problem on the product manifold $(\mathcal{N}, g_{\mathcal{N}}) := \left(\mathbb{R}^{p-\tilde{p}} \times \mathcal{M},\ g_{\text{flat}} \oplus g_{\mathcal{M}}\right)$. Note that $\dim(\mathcal{N}) = p - q$. Here $g_{\mathcal{N}} = g_{\text{flat}} \oplus g_{\mathcal{M}}$ is the product metric and g_{flat} corresponds* to the standard Euclidean metric on $\mathbb{R}^{p-\tilde{p}}$. Altogether, having these geometric structures in place, the CERM problem in (2) is equivalent to (7), which proves the statement. □

6.2. Relation to Lagrange Multipliers and known RSGD Methods

Lagrange multipliers We compare our strategy with a related alternative, namely the method of Lagrange Multipliers. Lagrange Multipliers can be understood from a geometric perspective by essentially writing down the necessary conditions for stationarity in a special local chart, namely one in which $\mathcal{M}$ is embedded into $\mathbb{R}^{\tilde{p}}$ as the graph of the inverse chart. The resulting necessary conditions for a point $\xi^* \in \mathbb{R}^p$ to solve (2) is the existence of a so-called Lagrange multiplier $\mu^* \in \mathbb{R}^q$ so that

$$\begin{cases} \nabla_{g_{\text{flat}}} H(\xi^*) + \sum_{j=1}^{q} \mu_j^* \pi_{\tilde{p}}^T \nabla_{g_{\text{flat}}} F_j(\pi_{\tilde{p}}(\xi^*)) = 0, \\ F(\pi_{\tilde{p}}(\xi^*)) = 0. \end{cases} \tag{8}$$

Here we have defined $H : \mathbb{R}^p \to \mathbb{R}$ by $H(\xi) := \mathbb{E}\left(L(G(X,\xi), Y)\right)$.

The system of equations in (8) is referred to as the Karush–Kuhn–Tucker (KKT) conditions. For general nonlinear problems, the KKT-conditions constitute a highly nonlinear system of equations and are difficult to solve directly. Many techniques for solving the constrained problem in (2) are based on adaptations of Newton's method for (8), e.g., Sequential Quadratic Programming (SQP) or Interior Point methods to name a few, see [5] for more. The dynamics of such algorithms, i.e., the behavior of the generated sequence of points, takes place in a higher dimensional space $\mathbb{R}^p \times \mathbb{R}^q$ than what we started with and is largely determined by Newton's method for solving (8).

Our approach is fundamentally different from such methods in the following sense. Firstly, the dynamics of our optimization scheme takes place on a *lower* dimensional submanifold $\mathcal{N}$ defined by the constraints. Once we have initialized *any* initial point on $\mathcal{N}$, we use the intrinsic geometry of the manifold to find a next point by following descent trajectories *confined to the manifold*, e.g., geodesics. We therefore satisfy the desired constraints *throughout the entire* optimization procedure thereby exploring the space of feasible parameters directly. Finally, the dynamics of our algorithm is completely determined by the (negative) gradient flow of the objective, and not by Newton's method for (8).

Riemannian Stochastic Gradient Descent As mentioned in the introduction, performing SGD on Riemannian manifolds is well-known and has been studied extensively before, see, e.g., [6, 14, 34, 35, 42]. To the best of our knowledge, however, current methods require an explicit description of charts on the underlying manifold and mostly adopt a completely extrinsic point of view. This process involves manual computations on paper on a per-case basis. For instance, the authors of [34, 35] present methods for three specific cases: the space of positive definite matrices, Grassman, and Stiefel manifolds. Our main contribution and novelty is of a computational nature: we do not require an explicit description of the underlying manifold, thereby avoiding any per-case computations. We only requires an implicit description of

*Formally, we should incorporate the dimension $\tilde{p}$ into the notation for the flat metric on $\mathbb{R}^{\tilde{p}}$. However, to avoid clutter in the notation, we will denote the standard Euclidean metric on any finite-dimensional vector space in the same way.

the manifold given in the form of a finite-dimensional set of equations. We use this knowledge alone to construct the tools needed for performing SGD numerically. This makes our framework highly flexible and enables us to deal with a vast class of constraints for which we can construct appropriate (graph) charts and perform local computations, e.g., computing Riemannian metrics and gradient operators, in an automated fashion. Due to this automation, the user is only required to provide an implementation of a constraint.

Finally, we mention the methods developed in [7, Chapter 7.7], which are most closely related to ours. The main difference between our and their technique is that we adopt an entirely intrinsic point of view and provide explicit algorithms amenable to numerical computations. To be more specific about the differences, let us start by clarifying the point about the intrinsic and extrinsic perspectives. The first key difference is that [7] does not directly follow the gradient flow of the objective $\mathcal{L}$ to be optimized. Instead, they reason extrinsically about the gradient $\nabla_{g_{\mathcal{M}}}\mathcal{L}$ and the associated paths to follow. Specifically, the authors exploit the observation that the tangent space $T_\theta\mathbb{R}^{\tilde{p}}$ of the ambient manifold $\mathbb{R}^{\tilde{p}}$ admits an orthogonal decomposition of the form $T_\theta\mathcal{M} \oplus T_\theta\mathcal{M}^\perp$, where $\theta \in \mathcal{M}$ is the current point in the descent algorithm. We are a bit sloppy here with the notation: to make sense of this decomposition, one has to explicitly embed the tangent space $T_\theta\mathcal{M}$ into $T_\theta\mathbb{R}^{\tilde{p}}$ using the derivative of the embedding map $\iota : \mathcal{M} \to \mathbb{R}^n$, which is intimately tied to the extrinsic point of view. The authors then proceed and explain how to project onto the orthogonal complement $T_\theta\mathcal{M}^\perp$, which is *extrinsic* to $\mathcal{M}$, using the derivative of the constraint. This requires no knowledge of a particular coordinate system on $\mathcal{M}$; the standard coordinates on the ambient vector space can be used. Subsequently, they use this orthogonal decomposition to compute the *embedding* of the gradient $\nabla_{g_{\mathcal{M}}}\mathcal{L}(\theta)$ into $T_\theta\mathbb{R}^{\tilde{p}}$. They then continue to operate extrinsically and suggest defining a so-called retraction map $R : T\mathcal{M} \to \mathcal{M}$ using the extrinsic (flat) geometry. The particular suggested retraction involves projecting a path (straight line) in $\mathbb{R}^{\tilde{p}}$ back to $\mathcal{M}$, which involves solving *another* (nonlinear) constraint optimization problem, see [7, Chapter 7.7, formula 7.76]. This suggestion requires that for each point in the ambient space, there exists a local neighborhood around it, for which an associated unique closest point in $\mathcal{M}$ exists. The authors mention that this is a highly non-trivial issue and conclude that the proposed method is therefore difficult to implement numerically for general manifolds; no alternative is provided.

Our method, on the other hand, *does* provide such a general numerical implementation, and does not require solving any additional optimization problems. Let us discuss the key differences with our method. First, we do not embed $\nabla_{g_{\mathcal{M}}}\mathcal{L}(\theta)$ into the ambient space. Instead, we construct a local coordinate system on $\mathcal{M}$ and perform computations entirely intrinsic to $\mathcal{M}$; we never consider the (extrinsic) orthogonal complement $T_\theta\mathcal{M}^\perp$. Furthermore, our choice to choose specific coordinates on $\mathcal{M}$ allows us to construct a well-defined retraction map, different than the one in [7, Chapter 7.7], amenable to numerical computations. Namely, we directly follow the geodesic, or an approximation of it, in the direction of $-\nabla_{g_{\mathcal{M}}}\mathcal{L}(\theta)$. We can do so because we explicitly compute the Riemannian metric, allowing us to compute geodesics and components of the Levi-Civita connection (explained in Sec. 6.6). In particular, even if we do not use the exact geodesic, we are ensured to remain on $\mathcal{M}$ for sufficiently short integration times, since all computations are performed intrinsically in our coordinate chart. This intrinsic approach means we never follow paths outside of $\mathcal{M}$, which must be projected back. This is why we do not need to solve another constrained optimization problem to get back to $\mathcal{M}$, but the suggestion in [7, Chapter 7.7] does. Finally, our specific choice for a retraction map ensures that the dynamics of our optimization algorithm are completely determined by the gradient flow of $\mathcal{L}$.

6.3. Graph coordinates on $\mathcal{M}$

In this section we explain how to construct a special (local) coordinate system, a so-called graph chart, on $\mathcal{M}$ around a point $\theta^* \in \mathcal{M}$. This chart will be used extensively to perform numerical computations, e.g., to evaluate the Riemannian metric $g_{\mathcal{M}}$. The existence of this special chart is guaranteed by the Implicit Function Theorem and naturally comes up in the proof of the so-called Pre-Image Theorem [19], which provides sufficient conditions for $\mathcal{M} = F^{-1}(0)$ to be an embedded submanifold of $\mathbb{R}^{\tilde{p}}$. Below, we will essentially repeat the proof of this theorem, in a somewhat simplified setting, see [19] for the slightly more general case dealing with smooth maps between general manifolds. The reason for including an explicit proof is that the computational steps form the backbone of our method.

Theorem 6.2 (Pre-image theorem)**.** *Let $F : \mathbb{R}^{\tilde{p}} \to \mathbb{R}^q$ be a map of class C^k, where $k \geq 2$. If zero is a regular value of F, then $F^{-1}(0)$ is an embedded C^k-submanifold of $\mathbb{R}^{\tilde{p}}$ of dimension $\tilde{p} - q$.*

Proof. Assume zero is a regular value of F and let $\theta^* \in F^{-1}(0)$ be arbitrary. Then $DF(\theta^*)$ must have q linearly independent columns. For the sake of concreteness, assume

$$\left[\frac{\partial F}{\partial \theta_{j_1}}(\theta^*) \quad \dots \quad \frac{\partial F}{\partial \theta_{j_q}}(\theta^*)\right] \tag{9}$$

is an isomorphism on $\mathbb{R}^q$, where $j_1 < \dots < j_q$ and $1 \leq j_k \leq \tilde{p}$. This gives rise to the decomposition $\mathbb{R}^{\tilde{p}} = \mathbb{R}^q \oplus \mathbb{R}^{\tilde{p}-q}$, where the first subspace corresponds to the coordinates with multi-index $(j_1, \dots, j_q)$, and the second subspace contains the remaining coordinates. Let

$\pi_q : \mathbb{R}^{\tilde{p}} \to \mathbb{R}^q$ and $\pi_{\tilde{p}-q} : \mathbb{R}^{\tilde{p}} \to \mathbb{R}^{\tilde{p}-q}$ denote the projections onto the first, and second subspace, respectively, and write $v := \pi_q(\theta)$ and $\beta := \pi_{\tilde{p}-q}(\theta)$ for the corresponding coordinates. We may then view F as a function of (v, β). More formally, we define a new map $\tilde{F} : \mathbb{R}^q \oplus \mathbb{R}^{\tilde{p}-q} \to \mathbb{R}^q$ by $\tilde{F}(v, \beta) := F(\nu(v, \beta))$, where $\nu : \mathbb{R}^q \oplus \mathbb{R}^{\tilde{p}-q} \to \mathbb{R}^{\tilde{p}}$ is a permutation which puts the coordinates (v, β) back in the original ordering.

Next, write $v^* = \pi_q(\theta^*)$, $\beta^* = \pi_{\tilde{p}-q}(\theta^*)$ and observe that $D_v\tilde{F}(v^*, \beta^*)$ is an isomorphism on $\mathbb{R}^q$ by construction. Therefore, by the Implicit Function Theorem, there exists a unique C^k-map $\tilde{\zeta} : B \subset \mathbb{R}^{\tilde{p}-q} \to \mathbb{R}^q$, where B is an open neighborhood of β^*, such that $\tilde{\zeta}(\beta^*) = v^*$ and $\tilde{F}\left(\tilde{\zeta}(\beta), \beta\right) = 0$ for all $\beta \in B$. Altogether, this shows that the map $\zeta : B \to F^{-1}(0)$ defined by $\zeta(\beta) := \nu(\tilde{\zeta}(\beta), \beta)$ is a local parameterization of $F^{-1}(0)$, i.e., its inverse $\Lambda := \zeta^{-1}$ is a local chart on $U := \zeta(B) \subset F^{-1}(0)$. Therefore, since $\theta^* \in F^{-1}(0)$ is arbitrary, it follows from this observation that $F^{-1}(0)$ is an embedded C^k-submanifold of dimension $\tilde{p} - q$. □

Remark 6.3 (Relaxation). Strictly speaking, one still needs to show that U is open in $F^{-1}(0)$, and that there is a chart in the ambient manifold $\mathbb{R}^{\tilde{p}}$ in which $F^{-1}(0)$ is locally described by setting the first q coordinates to zero. We omitted the details because they follow in a straightforward manner from our arguments. In particular, the proof of Theorem 6.2 also shows that we may relax the condition that zero is a regular value of F. Specifically, let $\mathcal{R} \subset F^{-1}(0)$ be the set of regular points of F. If $\mathcal{R} \neq \emptyset$, then $\mathcal{R}$ is an embedded C^k-submanifold of $\mathbb{R}^{\tilde{p}}$ of dimension $\tilde{p} - q$.

Remark 6.4 (Graph coordinates and Lagrange Multipliers). The coordinates associated with the chart Λ are commonly referred to as graph coordinates since $\mathcal{M}$ is locally parameterized by the graph of $\tilde{\zeta}$. The existence of Lagrange Multipliers can be proven by writing the necessary conditions for stationarity of the objective in (2) in this chart.

Remark 6.5 (Regularity). If F is C^∞ or analytic, then the manifold inherits the same regularity.

Throughout this paper, we assume that zero is a regular value of F, which guarantees that $(\mathcal{M}, g_\mathcal{M})$ is an embedded C^2 Riemannian manifold. In the discussion below, we will consider a point $\theta^* \in \mathcal{M}$, and explain how to explicitly evaluate the Riemannian metric at this point relative to the chart Λ. In turn, this will enable us to compute gradients. To avoid clutter in the notation, we henceforth assume without loss of generality, that the first q components of $DF(\theta^*)$ are linearly independent, i.e., $(j_1, \ldots, j_q) = (1, \ldots, q)$, and hence $F = \tilde{F}$. Note that this assumption will hold on an entire open neighborhood of θ^*. For points outside this neighborhood, one needs to choose another set of components that constitute a linearly independent system, thereby obtaining a different chart Λ.

In practice, we do not have an explicit formula for the chart Λ constructed in Theorem 6.2. Nonetheless, we can compute with it implicitly as explained below. For the sake of illustration, however, we will first consider a toy example before we proceed, in which explicit computations and formulae are available. We will continue this example throughout this section to complement the otherwise abstract numerical recipes.

Example 6.6 (The unit sphere $\mathbb{S}^2$)**.** Consider the map $F : \mathbb{R}^3 \to \mathbb{R}$ defined by $F(\theta) := \theta_1^2 + \theta_2^2 + \theta_3^2 - 1$. Clearly, $\mathcal{M} = F^{-1}(0)$ corresponds to the unit sphere $\mathbb{S}^2$. We will use Theorem 6.2 to prove that $\mathbb{S}^2$ is a C^∞ two-dimensional embedded submanifold of $\mathbb{R}^3$. While one can easily prove this by constructing explicit charts, e.g., using stereographic projection or polar coordinates, our goal is to demonstrate how to use Theorem 6.2 and explicitly construct the chart Λ.

First observe that $DF(\theta) = 2\begin{bmatrix}\theta_1 & \theta_2 & \theta_3\end{bmatrix}$. Further note that for any $\theta \in F^{-1}(0)$ at least one of the components θ_j must be nonzero. Therefore, $DF(\theta)$ is surjective for all $\theta \in F^{-1}(0)$, i.e., zero is a regular value of F. Consequently, $\mathbb{S}^2 = F^{-1}(0)$ is a 2-dimensional embedded submanifold of $\mathbb{R}^3$ by Theorem 6.2. Moreover, without explicitly constructing charts, we immediately see that $\mathbb{S}^2$ is a C^∞-manifold (analytic even), since F is a C^∞-map. The chart Λ from the proof is easily constructed in this case. To see this, suppose $\theta_1 > 0$, then $\beta = (\theta_2, \theta_3)$, $\zeta(\beta_1, \beta_2) = \left(\sqrt{1 - \beta_1^2 - \beta_2^2}, \beta_1, \beta_2\right)$ and $\Lambda(\theta) = (\theta_2, \theta_3)$. The (maximal) domain of this chart is $U = \{\theta \in \mathbb{S}^2 : \theta_1 > 0\}$.

6.4. Riemannian metric on $\mathcal{N}$

In this section we express the product metric on $\mathcal{N}$ in local coordinates with respect to the chart $\Phi := (\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$. Here $\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}$ denotes the identity map on $\mathbb{R}^{p-\tilde{p}}$. We start by deriving a representation of $g_\mathcal{M}$ relative to Λ. For this purpose, denote the coordinates associated to Λ by $(\lambda^1, \ldots, \lambda^{\tilde{p}-q})$, and the standard coordinates on $\mathbb{R}^{\tilde{p}-q}$ by $\left(\beta^1, \ldots, \beta^{\tilde{p}-q}\right)$. Recall that the pullback metric on $\mathcal{M}$ is given by $g_\mathcal{M} = \iota^*\langle\cdot,\cdot\rangle$. Therefore, in local coordinates, we have $g_\mathcal{M} = (g_\mathcal{M})_{ij}\, d\lambda^i \otimes d\lambda^j$, where $(g_\mathcal{M})_{ij} : U \to \mathbb{R}$ is given by

$$
\begin{aligned}
(g_\mathcal{M})_{ij}(\theta) &= \left\langle \iota_{*,\theta}\left(\frac{\partial}{\partial\lambda^i}\bigg|_\theta\right),\ \iota_{*,\theta}\left(\frac{\partial}{\partial\lambda^j}\bigg|_\theta\right)\right\rangle \\
&= \left\langle \frac{\partial\zeta}{\partial\beta^i}(\Lambda(\theta)),\ \frac{\partial\zeta}{\partial\beta^j}(\Lambda(\theta))\right\rangle, \\
&\qquad 1 \le i, j \le \tilde{p} - q\,,
\end{aligned}
$$

where we recall that $\zeta = (\tilde{\zeta}(\beta), \beta))$ is a local parameterization of the manifold. In practice, we are only interested in

a specific choice for θ, namely $\theta = \theta^*$. For this choice, the chart $\Lambda := \zeta^{-1}$ is explicitly known: $\Lambda(\theta^*) = \beta^*$. Hence, to evaluate the metric at θ^*, we need to explicitly compute $D\zeta(\beta^*)$.

To evaluate $D\zeta(\beta^*)$, first observe that $D\zeta(\beta) = \begin{bmatrix} D\tilde{\zeta}(\beta)^T & \boldsymbol{I}_{(\tilde{p}-q)\times(\tilde{p}-q)} \end{bmatrix}^T$ for any $\beta \in B$. Here $\boldsymbol{I}_{(\tilde{p}-q)\times(\tilde{p}-q)}$ denotes the $(\tilde{p}-q)\times(\tilde{p}-q)$ identity matrix. Furthermore, we can compute the derivative of $\tilde{\zeta}$ by using its defining property (see the proof of Theorem 6.2)

$$F\left(\tilde{\zeta}(\beta), \beta\right) = 0, \quad \beta \in B.$$

More precisely, differentiating both sides of this equation and evaluating at β^* yields

$$D_v F(\theta^*) D\tilde{\zeta}(\beta^*) = -D_\beta F(\theta^*). \tag{10}$$

Both $D_v F(\theta^*)$ and $D_\beta F(\theta^*)$ can be explicitly evaluated. Moreover, $D_v F(\theta^*)$ is a non-singular $q \times q$ matrix. Hence we can compute $D\tilde{\zeta}(\beta^*)$ by solving the linear system of equations in (10). Subsequently, we can explicitly evaluate the components of the Riemannian metric at θ^*:

$$(g_{\mathcal{M}})_{ij}(\theta^*) = \left\langle \frac{\partial \zeta}{\partial \beta^i}(\beta^*), \frac{\partial \zeta}{\partial \beta^j}(\beta^*) \right\rangle, \quad 1 \le i, j \le \tilde{p} - q. \tag{11}$$

Finally, we evaluate the product metric $g_{\mathcal{N}} = g_{\text{flat}} \oplus g_{\mathcal{M}}$ on $\mathcal{N}$ relative to $(\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$ at θ^*:

$$g_{\mathcal{N}}(\alpha, \theta^*) \simeq [g_{\mathcal{N}}(\alpha, \theta^*)]_\Lambda \tag{12}$$

$$:= \begin{bmatrix} \boldsymbol{I}_{(p-\tilde{p})\times(p-\tilde{p})} & \boldsymbol{0}_{(p-\tilde{p})\times(\tilde{p}-q)} \\ \boldsymbol{0}_{(\tilde{p}-q)\times(p-\tilde{p})} & [g_{\mathcal{M}}(\theta^*)]_\Lambda \end{bmatrix}, \tag{13}$$

where $\alpha \in \mathbb{R}^{p-\tilde{p}}$ and $[g_{\mathcal{M}}(\theta^*)]_\Lambda \in \mathrm{GL}(\tilde{p}-q, \mathbb{R})$ is the symmetric matrix whose $(i,j)^{\text{th}}$ component is given by $(g_{\mathcal{M}})_{ij}(\theta^*)$.

Example 6.7 (The unit sphere $\mathbb{S}^2$ - continued)**.** We end this section by continuing Example 6.6 and computing the components of the Riemannian metric $g_{\mathbb{S}^2}$ relative to Λ. This computation is only included to provide a concrete application of the abstract theory above. In practice, the computations, e.g., solving the equation in (10), are implemented numerically. Now, a straightforward computation shows that

$$D\zeta(\beta) = \begin{bmatrix} -\dfrac{\beta_1}{\sqrt{1-\beta_1^2-\beta_2^2}} & -\dfrac{\beta_2}{\sqrt{1-\beta_1^2-\beta_2^2}} \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Therefore, the components of the Riemannian-metric relative to Λ are given by

$$[g_{\mathbb{S}^2}(\theta)]_\Lambda = \frac{1}{1-\theta_2^2-\theta_3^2} \begin{bmatrix} 1-\theta_3^2 & \theta_2\theta_3 \\ \theta_2\theta_3 & 1-\theta_2^2 \end{bmatrix}.$$

6.5. Computing gradients on $\mathcal{N}$

In this section we explain how to compute the gradient of a smooth map $\mathcal{L} : \mathcal{N} \to \mathbb{R}$ relative to $\Phi = (\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$. For notational convenience, we denote the coordinates associated to $(\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$ by $(u^1, \dots, u^{p-q})$, where $\left(u^1, \dots, u^{p-\tilde{p}}\right) = \left(\alpha^1, \dots, \alpha^{p-\tilde{p}}\right)$ and $(u^{p-\tilde{p}+1}, \dots, u^{p-q}) = \left(\lambda^1, \dots, \lambda^{\tilde{p}-q}\right)$ are the coordinates associated to $\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}$ and Λ, respectively. In the next section, we will use these computations to find a minimizer of $\mathcal{L}$ using SGD. We remind the reader that our specific use case is the constrained ERM problem in (2), which corresponds to finding a minimum of

$$\mathcal{L}(\alpha, \theta) = \mathbb{E}\left(L\left(G\left(X, \alpha \oplus \iota(\theta)\right), Y\right)\right).$$

The gradient of $\mathcal{L}$ on $\mathcal{N}$ with respect to $g_{\mathcal{N}}$ is the unique vector field $\nabla_{g_{\mathcal{N}}}\mathcal{L} \in \mathfrak{X}(\mathcal{N})$ satisfying $d\mathcal{L} = g_{\mathcal{N}}(\cdot, \nabla_{g_{\mathcal{N}}}\mathcal{L})$. Such a vector field must exist since $g_{\mathcal{N}}$ is non-degenerate. In local coordinates,

$$d\mathcal{L} = \frac{\partial \mathcal{L}}{\partial u^j} du^j, \quad \nabla_{g_{\mathcal{N}}}\mathcal{L} = c^j \frac{\partial}{\partial u^j},$$

where $c^1, \dots c^{p-q} : \mathcal{N} \to \mathbb{R}$ are smooth (uniquely determined) functions. We can easily determine these functions by plugging them into the defining equation for the gradient and evaluating both sides at $\dfrac{\partial}{\partial u^i}$. This yields the following linear system of equations:

$$c^j (g_{\mathcal{N}})_{ij} = \frac{\partial \mathcal{L}}{\partial u^i}, \quad 1 \le i \le p - q.$$

Here $(g_{\mathcal{N}})_{ij} : \mathbb{R}^{p-\tilde{p}} \times U \to \mathbb{R}$ are the components of $g_{\mathcal{N}}$ relative to Φ. Similar as before, we define $[g_{\mathcal{N}}(\alpha, \theta)]_\Phi \in \mathrm{GL}(p-q, \mathbb{R})$ to be the symmetric matrix whose $(i,j)^{\text{th}}$ component is given by $(g_{\mathcal{N}})_{ij}(\alpha, \theta)$. Then

$$\nabla_{g_{\mathcal{N}}}\mathcal{L} = g_{\mathcal{N}}^{ij} \frac{\partial \mathcal{L}}{\partial u^j} \frac{\partial}{\partial u^i},$$

where $g_{\mathcal{N}}^{ij}(\alpha, \theta)$ are the components of the inverse of $[g_{\mathcal{N}}(\alpha, \theta)]_\Phi$.

In practice, of course, we will not invert the matrix $[g_{\mathcal{N}}(\alpha, \theta^*)]_\Phi$. Instead, we numerically solve the system of equations at our point of interest (α, θ^*) for the unknown coefficients $\left(c^j(\alpha, \theta^*)\right)_{j=1}^{p-q}$ by exploiting the block structure of the metric, see (12). In particular, we immediately see that the first $p - \tilde{p}$ components of $\nabla_{g_{\mathcal{N}}}\mathcal{L}(\alpha, \theta^*)$ are given by $c^j(\alpha, \theta^*) = \dfrac{\partial \mathcal{L}}{\partial \alpha^j}(\alpha, \theta^*)$, where $1 \le j \le p - \tilde{p}$. In other words, since the metric on $\mathbb{R}^{p-\tilde{p}}$ is flat, the associated components of the gradient reduce to the usual ones. On the

other hand, for the coordinates on $\mathcal{M}$, we have

$$\sum_{j=1}^{p-q} c^j(\alpha,\theta^*)(g_{\mathcal{N}})_{ij}(\alpha,\theta^*) =$$
$$= \sum_{j=p-\tilde{p}+1}^{p-q} c^j(\alpha,\theta^*)\left([g_{\mathcal{M}}(\theta^*)]_\Lambda\right)_{(i+\tilde{p}-p,j+\tilde{p}-p)},$$

for $p-\tilde{p}+1 \leq i \leq p-q$ by (12). Therefore, the last $\tilde{p}-q$ components $\left(c^j(\alpha,\theta^*)\right)_{j=p-\tilde{p}+1}^{p-q}$ of $\nabla_{g_{\mathcal{N}}}\mathcal{L}(\alpha,\theta^*)$ can be obtained by solving the linear (square) system

$$[g_{\mathcal{M}}(\theta^*)]_\Lambda \begin{pmatrix} c^{p-\tilde{p}+1}(\alpha,\theta^*) \\ \vdots \\ c^{p-q}(\alpha,\theta^*) \end{pmatrix} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \lambda^1}(\alpha,\theta^*) \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \lambda^{\tilde{p}-q}}(\alpha,\theta^*) \end{pmatrix}. \tag{14}$$

Computing partial derivatives We need one final ingredient to compute the gradient of $\mathcal{L}$. Namely, we need to evaluate its partial derivatives with respect to the coordinate system defined by $\Phi = (\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$. Clearly there is no difficulty in computing $\frac{\partial \mathcal{L}}{\partial \alpha^i}(\alpha,\theta^*)$, since $(\alpha^1,\dots,\alpha^{p-\tilde{p}})$ are the standard coordinates on $\mathbb{R}^{p-\tilde{p}}$, and thus correspond to the "usual" partial derivatives one encounters in calculus on vector spaces. For the partial derivatives with respect to $(\lambda^1,\dots,\lambda^{\tilde{p}-q})$, however, we have to be more careful, and compute from the perspective of the (non-trivial) chart:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda^i}(\alpha,\theta^*) &= \frac{\partial(\mathcal{L}\circ\Phi^{-1})}{\partial \beta^i}(\Phi(\alpha,\theta^*)) \\ &= \frac{\partial}{\partial \beta^i}\Big|_{\beta^*}(\beta \mapsto \mathcal{L}(\alpha,\zeta(\beta)) \\ &= D_\theta\mathcal{L}(\alpha,\theta^*)\frac{\partial \zeta}{\partial \beta^i}(\beta^*), \quad 1 \leq i \leq \tilde{p}-q, \end{aligned} \tag{15}$$

since $\Phi^{-1} = (\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \zeta)$ and $\zeta(\beta^*) = \theta^*$. In the last line we assumed that $\mathcal{L}(\alpha,\cdot)$ has a smooth extension to some open neighborhood $V \subset \mathbb{R}^{\tilde{p}}$ of $\mathcal{M}$ for all $\alpha \in \mathbb{R}^{p-\tilde{p}}$. This is the case for all our applications, where $\mathcal{L}$ comes from the constrained minimization problem in (2).

Altogether, we now have all the ingredients to numerically evaluate the gradient of a smooth map $\mathcal{L}: \mathcal{N} \to \mathbb{R}$ relative to the chart $(\mathrm{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$. The steps are summarized in Algorithm 2.

Example 6.8 (The unit sphere $\mathbb{S}^2$ - continued)**.** We continue our example of the unit sphere and explain how to compute the gradient of a smooth map $\mathcal{L}: \mathbb{S}^2 \to \mathbb{R}$. We assume that

Algorithm 2 Compute $\nabla_{g_{\mathcal{N}}}\mathcal{L}(\alpha,\theta^*)$ relative to Φ given $(\alpha,\theta^*) \in \mathcal{N}$.

1: Compute $DF(\theta^*)$.
2: Compute $D\zeta(\beta^*) = \begin{bmatrix} D\tilde{\zeta}(\beta^*)^T & \boldsymbol{I}_{(\tilde{p}-q)\times(\tilde{p}-q)} \end{bmatrix}^T$ by solving (10).
3: Compute $[g_{\mathcal{N}}(\alpha,\theta^*)]_\Phi$ by evaluating (12).
4: Compute the components of $\nabla_{g_{\text{flat}}}\mathcal{L}(\alpha,\theta^*)$ by evaluating $D_\alpha\mathcal{L}(\alpha,\theta^*)$.
5: Compute the partial derivatives $\frac{\partial \mathcal{L}}{\partial \lambda^i}(\alpha,\theta^*)$ for $1 \leq i \leq \tilde{p}-q$ using (15).
6: Compute the components of $\nabla_{g_{\mathcal{M}}}\mathcal{L}(\alpha,\theta^*)$ by solving (14).

$\mathcal{L}$ can be smoothly extended to an open neighborhood of $\mathbb{S}^2$ in $\mathbb{R}^3$. To compute the gradient relative to Λ, we need to solve the system in (14). For this purpose, we first explicitly compute the inverse of $[g_{\mathbb{S}^2}(\theta)]_\Lambda$:

$$\left([g_{\mathbb{S}^2}(\theta)]_\Lambda\right)^{-1} = \begin{bmatrix} 1-\theta_2^2 & -\theta_2\theta_3 \\ -\theta_2\theta_3 & 1-\theta_3^2 \end{bmatrix}.$$

Again, we stress that in practice, we do not invert this matrix, but solve the system of equations numerically instead. Next, we compute the partial derivatives of $\mathcal{L}$ relative to $\Lambda = (\lambda^1,\lambda^2)$ using (15):

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \lambda^1}(\theta) &= \frac{\partial \mathcal{L}}{\partial \theta_2}(\theta) - \frac{\theta_2}{\theta_1}\frac{\partial \mathcal{L}}{\partial \theta_1}(\theta), \\ \frac{\partial \mathcal{L}}{\partial \lambda^2}(\theta) &= \frac{\partial \mathcal{L}}{\partial \theta_3}(\theta) - \frac{\theta_3}{\theta_1}\frac{\partial \mathcal{L}}{\partial \theta_1}(\theta). \end{aligned}$$

Here $\left(\frac{\partial L}{\partial \theta_j}\right)_{j=1}^3$ denote the partial derivatives with respect to the standard coordinates on $\mathbb{R}^3$, i.e., these are the "usual" partial derivatives from calculus on vector spaces. Hence

$$\nabla_{g_{\mathbb{S}^2}}\mathcal{L}(\theta) = c_1(\theta)\frac{\partial}{\partial \lambda^1}\Big|_\theta + c_2(\theta)\frac{\partial}{\partial \lambda^2}\Big|_\theta \simeq \begin{bmatrix} c_1(\theta) \\ c_2(\theta) \end{bmatrix},$$

where

$$\begin{aligned} c_1(\theta) &= \frac{\partial \mathcal{L}}{\partial \theta_2}(\theta) - \theta_2\left(\theta_1\frac{\partial \mathcal{L}}{\partial \theta_1}(\theta) + \theta_2\frac{\partial \mathcal{L}}{\partial \theta_2}(\theta) + \theta_3\frac{\partial \mathcal{L}}{\partial \theta_3}(\theta)\right), \\ c_2(\theta) &= \frac{\partial \mathcal{L}}{\partial \theta_3}(\theta) - \theta_3\left(\theta_1\frac{\partial \mathcal{L}}{\partial \theta_1}(\theta) + \theta_2\frac{\partial \mathcal{L}}{\partial \theta_2}(\theta) + \theta_3\frac{\partial \mathcal{L}}{\partial \theta_3}(\theta)\right). \end{aligned}$$

6.6. Stochastic Gradient Descent

In this section we explain how to perform SGD on Riemannian manifolds using graph coordinates. For previous work on SGD on Riemannian manifolds, we refer the reader to [6, 14, 34, 35, 42]. The presented technique is completely

intrinsic to the manifold $\mathcal{N}$ and involves following (approximate) geodesics in the direction of the (negative) gradient of $\mathcal{L}$. To explain this idea in more detail, we first briefly recall the notion of geodesics and refer the reader to [18, 19] for a more comprehensive introduction to differential geometry.

6.6.1 Geodesics and parallel transport

The analog of a gradient descent step on a Riemannian manifold $(\mathcal{N}, g_{\mathcal{N}})$ is to follow "a straight line", confined to the manifold, in the direction of the negative gradient. In order to make sense of this, one first needs to generalize the notion of a straight line to arbitrary Riemannian manifolds. On Euclidean vector spaces, one can define a straight line as a curve whose velocity is constant. This notion makes sense on a vector space, since different tangent spaces can be related to one another, but does not make sense on a general manifold. An equivalent notion, which *can* be generalized to a Riemannian manifold, is to define a straight line as a curve whose acceleration is zero. The key idea here is that the notion of acceleration can be made sense of on any Riemannian manifold. More precisely, one can define a so-called affine connection or covariant derivative ∇, not to be confused with the notation for a gradient, which allows one to measure the change of one vector field in the direction of another. Formally, a connection is a differential operator $\nabla : \mathfrak{X}(\mathcal{N}) \times \mathfrak{X}(\mathcal{N}) \to \mathfrak{X}(\mathcal{N})$, which is $C^\infty(\mathcal{N})$-linear in the first variable, $\mathbb{R}$-linear in the second, and satisfies the Leibniz rule. Given two vector fields $V, W \in \mathfrak{X}(\mathcal{N})$, one typically writes $\nabla_V W$ and interprets this new vector field as measuring the change of W in the direction of V.

A connection is a so-called *local* operator in the sense that $\nabla_V W(u)$ is completely determined by $V(u) \in T_u\mathcal{N}$ and the behavior of W in a neighborhood around $u \in \mathcal{N}$. We may therefore write $\nabla_V W(u) = \nabla_{V(u)} W(u)$. This local property can in turn be used to measure the change of a vector field in the direction of a curve. More precisely, given a curve γ, there exists a unique (differential) operator D_t associated to γ and ∇, which enables one to differentiate vector fields $V \in \Gamma(\gamma)$ in the direction of γ. This operator is uniquely determined by three properties: it is $\mathbb{R}$-linear, satisfies the Leibniz rule, and if $V \in \Gamma(\gamma)$ can be extended to a vector field $\tilde{V}$ defined on an open neighborhood of $\gamma(t)$, then $D_t V(t) = \nabla_{\dot\gamma(t)} \tilde{V}(\gamma(t))$. One can now make sense of acceleration by defining it as the derivative of the velocity field $\dot\gamma$ in the direction of γ itself, i.e., acceleration is defined by $D_t\dot\gamma$. A "straight line" or geodesic is then simply defined as a curve whose acceleration field is zero. The existence of geodesics is guaranteed, at least locally, by the existence and uniqueness theorem for ODEs, see the discussion below.

A covariant derivative ∇ allows one to generalize many more familiar concepts from Euclidean vector spaces to Riemannian manifolds. For instance, given a curve $\gamma : [0, T] \to \mathcal{N}$ and tangent vector $V_0 \in T_{\gamma(t_0)}\mathcal{N}$, one may extend V_0 to a vector field $V \in \Gamma(\gamma)$ which "is parallel" to V_0 everywhere, see Figure 6. This extension V is referred to as the parallel transport of V_0 along γ. The notions of geodesics and parallel transport, however, heavily depend on the choice of connection. In general, there exist infinitely many connections on a Riemannian manifold. There exists exactly one connection, however, the so-called Levi-Civita connection, which in a sense is "naturally aligned" with the Riemannian metric. This specific connection may be summarized in a geometric way by the following two conditions, which are usually taken for granted on Euclidean spaces. First, if $\gamma : [0, T] \to \mathcal{N}$ is a curve and $V_0, W_0 \in T_{\gamma(t_0)}\mathcal{N}$ are tangent vectors with angle ϕ between them, then the parallel extensions $V, W \in \Gamma(\gamma)$ must have angle ϕ between them as well at any point on γ (metric compatibility), see Figure 6. Secondly, for any coordinate chart on $\mathcal{N}$, the rate of change of one coordinate direction in the direction of another must not change if we swap directions (torsion free). In this paper we always use the Levi-Civita connection.

Finally, we provide a local description of a geodesic γ. Let $t_0 \in (0, T)$ and assume $(U, u^1, \dots, u^{p-q})$ is any chart containing $\gamma(t_0)$, then there exists a $\delta > 0$ such that $\gamma((t_0 - \delta, t_0 + \delta)) \subset \mathcal{N}$. Write $\partial_l = \frac{\partial}{\partial u^l}$ and observe that for each $1 \le i, j \le p - q$, there exist smooth functions $\Gamma_{ij}^k : U \to \mathbb{R}$ such that $\nabla_{\partial_i}\partial_j = \Gamma_{ij}^k \partial_k$, since $(\partial_l)_{l=1}^{p-q}$ is a frame on U. The coefficients $\left\{\Gamma_{ij}^k : 1 \le i, j, k \le p - q\right\}$ are called the *Christoffel symbols* of ∇ on U. They completely characterize the connection on U. The equation for a geodesic starting at an initial point u_0 with initial velocity V_0 is given by

$$\begin{cases} \ddot\gamma^k(t) + \dot\gamma^i(t)\dot\gamma^j(t)\Gamma_{ij}^k(\gamma(t)) = 0, & 1 \le k \le p - q, \\ \dot\gamma^k(t_0) = V_0^k, & 1 \le k \le p - q, \\ \gamma(t_0) = u_0, \end{cases} \tag{16}$$

see [18]. Here we have expressed γ and the components of its velocity in local coordinates:

$$\dot\gamma(t) = \dot\gamma^i(t)\partial_i\big|_{\gamma(t)}, \quad \gamma^i := u^i \circ \gamma.$$

This is a second-order ordinary differential equation for the unknown curve (geodesic) γ. In general, this equation is *nonlinear*. The existence and uniqueness theorem for ODEs only guarantees the existence of a *local* solution. The solution may be extended outside of U by considering other charts. However, due to the nonlinearity, there may be obstructions to extending the solution beyond a certain point. In general, there is no guarantee that a geodesic can be extended and defined for all $t \in \mathbb{R}$. A manifold with the property that geodesics exist for all time is called *complete*. In particular, any compact manifold is complete [18]. We remark that

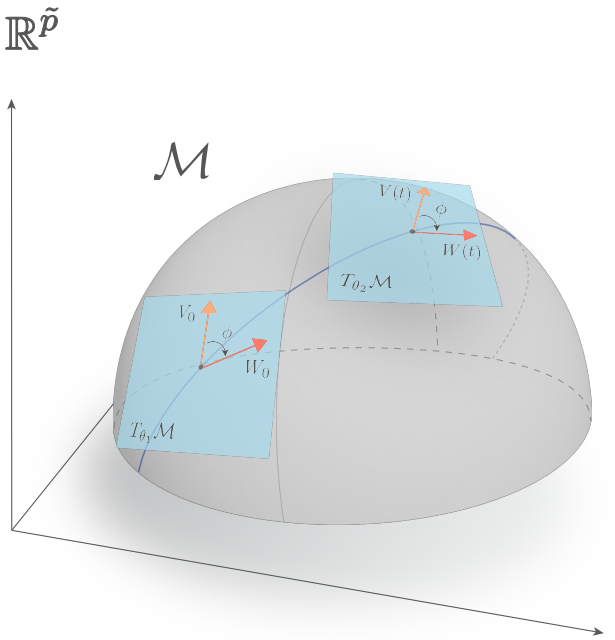


Figure 6. In this figure we depict a curve $\gamma : [0, T] \to \mathcal{M}$ (in blue) on which we have drawn two points, $\gamma(t_0)$ and $\gamma(t)$, for some $t, t_0 \in (0, T)$. In addition, we have drawn the tangent spaces associated to these points. The tangent vectors $V_0, W_0 \in T_{\gamma(t_0)}\mathcal{M}$ are "parallel transported" along γ resulting in vector fields $V, W \in \Gamma(\gamma)$. The Levi-Civita connection is the unique torsion free connection for which the angle between any two vectors $V_0, W_0 \in T_{\gamma(t_0)}\mathcal{M}$ and their parallel extensions remains constant.

for the purpose of SGD local existence is sufficient, since we need to take sufficiently small steps on the manifold to guarantee descent of the objective.

6.6.2 Gradient descent steps

We will now explain how to define a gradient descent step on our manifold of interest $(\mathcal{N}, g_{\mathcal{N}}) = \left(\mathbb{R}^{p-\tilde{p}} \times \mathcal{M}, g_{\text{flat}} \oplus g_{\mathcal{M}}\right)$ by computing approximate solutions of the geodesic equation (16). The main idea is to follow the geodesic starting at our current point (α, θ^*) in the direction of the negative gradient $-\nabla_{g_{\mathcal{N}}}\mathcal{L}(\alpha, \theta^*)$ for a small amount of time. While there exist many efficient techniques to compute high order approximate solutions of ODEs, e.g., Runge-Kutta solvers, they typically rely on evaluating the associated vector field on a neighborhood of the initial condition. In our set up, this would correspond to evaluating the Christoffel symbols at different points on the manifold. While it would be possible to explore nearby points in our chart $\Phi = (\text{id}_{\mathbb{R}^{p-\tilde{p}}}, \Lambda)$, e.g, by computing a second or higher order Taylor-expansion of ζ, our objective is not to just simply explore $\mathcal{N}$. Instead, we are only interested in following paths on $\mathcal{N}$ which lead to a decrease in $\mathcal{L}$. In particular, we are limited to choosing sufficiently small step-sizes, since we wish to stay on descent directions for $\mathcal{L}$. For this reason, since we only need to integrate the geodesic equation for small amounts of time, we use a first or second order Taylor-expansion to approximate the solution of (16).

More precisely, let $\boldsymbol{\gamma} := \begin{bmatrix}\gamma^1 & \dots & \gamma^{p-q}\end{bmatrix}^T$ denote the curve in local coordinates, then

$$\boldsymbol{\gamma}(t_0 + h) = \Phi(u_0) + [V_0]_\Phi h - \frac{1}{2}h^2 V_0^i V_0^j \boldsymbol{\Gamma}_{ij}(u_0) + o(h^2),$$

$$\boldsymbol{\Gamma}_{ij}(u_0) := \begin{bmatrix}\Gamma_{ij}^1(u_0) \\ \vdots \\ \Gamma_{ij}^{p-q}(u_0)\end{bmatrix}$$

as $h \to 0$. For our particular case, we set

$$u_0 = (\alpha, \theta^*), \quad V_0 = -\nabla_{g_{\mathcal{N}}}\mathcal{L}\left(\alpha, \theta^*\right) \simeq -\boldsymbol{c}(\alpha, \theta^*),$$

$$\boldsymbol{c}(\alpha, \theta^*) := \begin{bmatrix}c^1(\alpha, \theta^*) \\ \dots \\ c^{p-q}(\alpha, \theta^*)\end{bmatrix},$$

where $\boldsymbol{c}(\alpha, \theta^*)$ are the components of the gradient relative to Φ. We define *the second order gradient descent step* with

step-size h based at (α, θ^*) for $\mathcal{L}$ by

$$\begin{bmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{bmatrix} = \begin{bmatrix}\alpha\\ \beta^*\end{bmatrix} - \boldsymbol{c}(\alpha,\theta^*)h - \frac{1}{2}h^2 c^i(\alpha,\theta^*)c^j(\alpha,\theta^*)\boldsymbol{\Gamma}_{ij}(\alpha,\theta^*).$$

Here $\Phi(\alpha,\theta^*) = (\alpha,\beta^*)$ is the coordinate representation of (α,θ^*). Similarly, we define *the first order gradient descent step* with step-size h based at (α,θ^*) by

$$\begin{bmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{bmatrix} = \begin{bmatrix}\alpha\\ \beta^*\end{bmatrix} - \boldsymbol{c}(\alpha,\theta^*)h.$$

Note very carefully that the gradient descent steps are taken *in the local coordinate system*. For sufficiently small h, we are guaranteed that the new point $(\tilde{\alpha},\tilde{\beta})$ is contained in the current chart for both the first and second order steps. However, to get back to the manifold, we have to evaluate $\Phi^{-1}(\tilde{\alpha},\tilde{\beta}) = (\tilde{\alpha},\zeta(\tilde{\beta}))$. In addition, we also have to explicitly evaluate the Christoffel symbols. The computational details are given below.

6.6.3 Evaluating the inverse chart

We will use a Taylor expansion to evaluate the inverse chart ζ on $\mathcal{M}$ at $\tilde{\beta}$. Subsequently, we use Newton's method to refine the approximation. The resulting point that we find must necessarily correspond to $\zeta(\tilde{\beta})$, and is thus completely determined by $\tilde{\beta}$, since ζ is locally unique as explained in Theorem 6.2. This justifies the claim made in Sec. 6.2 that the search dynamics of our algorithm is completely determined by the negative gradient flow of $\mathcal{L}$, since $\tilde{\beta}$ is.

Below we provide the computational details for the case of a second order Taylor expansion; the first order case is obtained by ignoring the second order terms. To avoid clutter in the notation, we will henceforth (interchangeably) write

$$\begin{bmatrix}\alpha_{(k+1)}\\ \beta_{(k+1)}\end{bmatrix} = \begin{bmatrix}\tilde{\alpha}\\ \tilde{\beta}\end{bmatrix}, \quad \begin{bmatrix}\alpha_{(k)}\\ \beta_{(k)}\end{bmatrix} = \begin{bmatrix}\alpha\\ \beta^*\end{bmatrix}, \quad \theta_{(k)} = \zeta\left(\beta_{(k)}\right).$$

This notation also emphasizes that we move from a given point at step $k \in \mathbb{N}_0$ to a next point.

The second order Taylor expansion of $\tilde{\zeta}$ around $\beta_{(k)}$ is given by

$$\tilde{\zeta}\left(\beta_{(k+1)}\right) = \tilde{\zeta}\left(\beta_{(k)}\right) + D\tilde{\zeta}\left(\beta_{(k)}\right)d_k + \frac{1}{2}D^2\tilde{\zeta}\left(\beta_{(k)}\right)[d_k,d_k],$$

where $d_k := \beta_{(k+1)} - \beta_{(k)}$, as $\beta_{(k+1)} \to \beta_{(k)}$. We have explained in Section 6.4 how to explicitly compute $D\tilde{\zeta}\left(\beta_{(k)}\right)$, which was needed to evaluate the Riemannian metric. Here we employ the same strategy to compute the second derivative $D^2\tilde{\zeta}\left(\beta_{(k)}\right) \in \mathcal{B}^2(\mathbb{R}^{\tilde{p}-q},\mathbb{R}^q)$, where $\mathcal{B}^2(\mathbb{R}^{\tilde{p}-q},\mathbb{R}^q)$ denotes the space of $\mathbb{R}^q$-valued $\binom{2}{0}$-tensors on $\mathbb{R}^{\tilde{p}-q}$. We start by rewriting (10) as

$$DF\left(\tilde{\zeta}(\beta),\beta\right)\begin{bmatrix}D\tilde{\zeta}(\beta)\\ \boldsymbol{I}_{\mathbb{R}^{\tilde{p}-q}}\end{bmatrix} = 0, \quad \beta \in B.$$

Next, we differentiate both sides with respect to β and evaluate at $\beta_{(k)}$. This yields

$$\begin{aligned} &D_vF\left(\theta_{(k)}\right)D^2\tilde{\zeta}\left(\beta_{(k)}\right)[s_1,s_2] = \\ &-D^2F\left(\theta_{(k)}\right)\left[\begin{pmatrix}D\tilde{\zeta}\left(\beta_{(k)}\right)s_1\\ s_1\end{pmatrix},\begin{pmatrix}D\tilde{\zeta}\left(\beta_{(k)}\right)s_2\\ s_2\end{pmatrix}\right] \end{aligned} \tag{17}$$

for all $s_1,s_2 \in \mathbb{R}^{\tilde{p}-q}$. To compute the $(i,j)^{\text{th}}$ component of $D^2\tilde{\zeta}\left(\beta_{(k)}\right)$ with respect to the standard basis, i.e., in order to compute $\frac{\partial^2\tilde{\zeta}}{\partial\beta^i\partial\beta^j}\left(\beta_{(k)}\right)$, we evaluate both sides of (17) at $(s_1,s_2) = (e_i,e_j)$ and solve the equation for each $1 \le i,j \le \tilde{p}-q$. This equation admits a unique solution, since $D_vF\left(\theta_{(k)}\right)$ is an isomorphism on $\mathbb{R}^q$.

Finally, we approximate $\tilde{\zeta}\left(\beta_{(k+1)}\right)$ using its second (or first) order Taylor expansion and then use Newton's method to evaluate

$$\Phi^{-1}\left(\alpha_{(k+1)},\beta_{(k+1)}\right) = \left(\alpha_{(k+1)},\zeta\left(\beta_{(k+1)}\right)\right).$$

More precisely, we first approximate $\zeta\left(\beta_{(k+1)}\right)$ by

$$\begin{aligned} &\zeta\left(\beta_{(k+1)}\right) \approx \begin{bmatrix}v_{(k+1)}\\ \beta_{(k+1)}\end{bmatrix}, \\ &v_{(k+1)} := \tilde{\zeta}\left(\beta_{(k)}\right) + D\tilde{\zeta}\left(\beta_{(k)}\right)d_k + \frac{1}{2}D^2\tilde{\zeta}\left(\beta_{(k)}\right)[d_k,d_k]. \end{aligned} \tag{18}$$

We then refine this approximation by finding a zero of the map $v \mapsto F\left(v,\beta_{(k+1)}\right)$ using Newton's method and $v_{(k+1)}$ as initial guess. In particular, we solve the equation for v, while $\beta_{(k+1)}$ *remains fixed*. The zero that we find must necessarily correspond to $\zeta\left(\beta_{(k+1)}\right)$, since ζ is locally unique as explained in Theorem 6.2. Altogether, this yields the desired point $\left(\alpha_{(k+1)},\theta_{(k+1)}\right) \in \mathcal{N}$. See Figure 7 for a visualization of the steps described in this section.

6.6.4 Evaluating the Christoffel symbols

We end this section by explaining how to explicitly evaluate the Christoffel symbols Γ^k_{ij} at (α,θ^*). Recall that a connection is locally completely characterized by the Christoffel symbols. The constraints that uniquely determine the Levi-Civita connection, i.e., metric compatibility and torsion-freeness, therefore also impose constraints on the Christoffel symbols. In fact, the standard proof for the existence of the Levi-Civita connection is constructive and establishes an explicit relationship between the Christoffel symbols and the Riemannian metric:

$$\Gamma^k_{ij} = \frac{1}{2}(g_{\mathcal{N}})^{kl}\left(\frac{\partial(g_{\mathcal{N}})_{jl}}{\partial u^i} + \frac{\partial(g_{\mathcal{N}})_{il}}{\partial u^j} - \frac{\partial(g_{\mathcal{N}})_{ij}}{\partial u^l}\right),$$

where $1 \le i,j,k \le p-q$, see [18, 19] for instance. We will use this expression to numerically evaluate the Christoffel symbols.

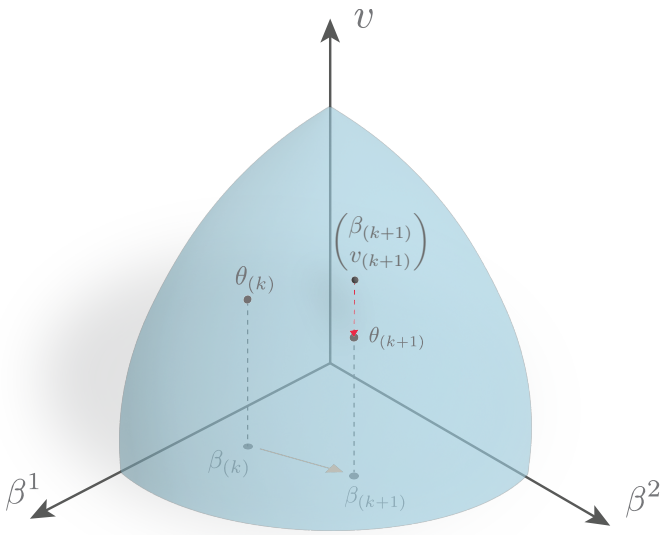


Figure 7. In this figure we visualize the computational steps for performing SGD on $\mathcal{N}$. We assume for the sake of clarity that there are no unconstrained parameters, i.e., $\mathcal{N} = \mathcal{M}$. We start at a previously computed point $\theta_{(k)} \in \mathcal{M}$ with associated coordinates $\beta_{(k)}$ relative to Λ. We remind the reader that the inverse of Λ embeds a patch of $\mathcal{M}$ into $\mathbb{R}^{\tilde{p}}$ as the graph of $\tilde{\zeta}$. Next, we perform a gradient descent step by following the first or second order Taylor expansion of the geodesic (depicted in orange) starting at $\beta_{(k)}$ in the direction of $-\nabla_{g_{\mathcal{M}}}\mathcal{L}\left(\theta_{(k)}\right)$ for a small amount of time. This yields the next point $\beta_{(k+1)}$, which is still contained in the chart. Finally, we evaluate the inverse chart ζ at the new point in two steps. First, we approximate $\tilde{\zeta}\left(\beta_{(k+1)}\right) \approx v_{(k+1)}$ using a first or second order Taylor expansion of $\tilde{\zeta}$, see (18). We then use Newton's method to refine this approximation and compute $\theta_{(k+1)} = \zeta\left(\beta_{(k+1)}\right)$.

It follows immediately from the block structure of the metric $g_{\mathcal{N}}$ in (12) that

$$\Gamma^k_{ij}(\alpha, \theta^*) = 0, \quad 1 \leq i \leq p - \tilde{p},\ 1 \leq j \leq p - q,$$

$$\Gamma^k_{ij}(\alpha, \theta^*) = 0, \quad p - \tilde{p} + 1 \leq i \leq p - q,\ 1 \leq j \leq p - \tilde{p},$$

for all $1 \leq k \leq p - q$. The reason why these coefficients are zero is because there is no interplay between the submanifolds $\mathbb{R}^{p-\tilde{p}}$ and $\mathcal{M}$, which together make up $\mathcal{N}$, and because the metric on $\mathbb{R}^{p-\tilde{p}}$ is flat. In particular, this shows that the component in $\mathbb{R}^{p-\tilde{p}}$ of a geodesic on $\mathcal{N}$ is just a straight line as expected.

It remains to consider the case $p - \tilde{p} + 1 \leq i, j \leq p - q$, which is associated to the non-trivial metric $g_{\mathcal{M}}$ on $\mathcal{M}$. We use the expression in (11) to compute the partial derivatives of the relevant components of $g_{\mathcal{M}}$. More precisely, observe that

$$\frac{\partial\,(g_{\mathcal{M}})_{ij}}{\partial\lambda^l}(\theta^*) = \frac{\partial}{\partial\beta^l}\bigg|_{\beta^*}\left(\beta \mapsto \left\langle \frac{\partial\zeta}{\partial\beta^i}(\beta), \frac{\partial\zeta}{\partial\beta^j}(\beta)\right\rangle\right)$$

$$= \left\langle \frac{\partial\tilde{\zeta}}{\partial\beta^i}(\beta^*), \frac{\partial^2\tilde{\zeta}}{\partial\beta^l\partial\beta^j}(\beta^*)\right\rangle + \left\langle \frac{\partial\tilde{\zeta}}{\partial\beta^j}(\beta^*), \frac{\partial^2\tilde{\zeta}}{\partial\beta^l\partial\beta^i}(\beta^*)\right\rangle$$

for $1 \leq i, j, l \leq \tilde{p} - q$. We can evaluate this expression numerically, since we can explicitly evaluate $D\tilde{\zeta}(\beta^*)$ and $D^2\tilde{\zeta}(\beta^*)$. Finally, to compute the relevant Christoffel symbols, we define vectors $\boldsymbol{w}_{ij}(\beta^*) \in \mathbb{R}^{\tilde{p}-q}$ for each $1 \leq i, j \leq \tilde{p} - q$ by

$$[\boldsymbol{w}_{ij}(\beta^*)]_l := \frac{1}{2}\left(\frac{\partial(g_{\mathcal{M}})_{jl}}{\partial\lambda^i} + \frac{\partial(g_{\mathcal{M}})_{il}}{\partial\lambda^j} - \frac{\partial(g_{\mathcal{M}})_{ij}}{\partial\lambda^l}\right)(\beta^*),$$

where $1 \leq l \leq \tilde{p} - q$. The remaining (non-zero) Christoffel symbols associated to $\mathcal{M}$ can now be computed by solving the following linear system of equations:

$$[g_{\mathcal{M}}(\theta^*)]_\Lambda[\Gamma^k_{\tilde{i}\tilde{j}}(\alpha, \theta^*)]_{k=1}^{\tilde{p}-q} = \boldsymbol{w}_{ij}(\beta^*),$$
$$\tilde{i} = i + p - \tilde{p},\ \tilde{j} = j + p - \tilde{p}.$$

7. Supplementary information Multiresolution Analysis

In this section, we provide the mathematical details needed for our main application. In Sec. 7.3 we review the Discrete Wavelet Transform (DWT), which plays the role of the decoder in our auto-contouring network. In Sec. 7.4 we explain how to set up constraints for learning wavelet filters.

7.1. Formal definition MRA

Definition 7.1 (Formal definition MRA [26])**.** Let $T_k : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ and $\mathcal{D}_j : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ denote the translation and normalized dilation operator, respectively, defined by $T_k\gamma(t) = \gamma(t-k)$ and $\mathcal{D}_j\gamma(t) = 2^{\frac{j}{2}}\gamma(2^j t)$ for $\gamma \in L^2(\mathbb{R}) \cap C_0^\infty(\mathbb{R})$ and $j, k \in \mathbb{Z}$. A multiresolution analysis of $L^2(\mathbb{R})$ is an increasing sequence of closed subspaces $(V_j)_{j\in\mathbb{Z}}$, such that

(i) $\bigcap_{j\in\mathbb{Z}} V_j = \{0\}$,

(ii) $\bigcup_{j\in\mathbb{Z}} V_j$ is dense in $L^2(\mathbb{R})$,

(iii) $\gamma \in V_j$ if and only if $\mathcal{D}_1\gamma \in V_{j+1}$,

(iv) V_0 is invariant under translations,

(v) $\exists\varphi \in L^2(\mathbb{R})$ such that $\{T_k\varphi\}_{k\in\mathbb{Z}}$ is an orthonormal basis for V_0.

Condition (ii) formalizes the idea that any signal in $L^2(\mathbb{R})$ can be arbitrarily well approximated using an appropriate resolution level. Condition (iii) encapsulates the idea that V_{j+1} is the next resolution level with respect to our choice of dilation operators $\mathcal{D}_j$, i.e., there are no other resolution levels between V_j and V_{j+1}. Combined with (iv) it implies that each subspace V_j is invariant under integer shifts. Finally, condition (v) formalizes the idea that the subspaces are spanned by translations and dilations of the map φ; the so-called *scaling function* or *father wavelet*. Indeed, it is straightforward to show that $\{\varphi_{jk} : k \in \mathbb{Z}\}$ is an orthonormal basis for V_j, where $\varphi_{jk} := \mathcal{D}_j T_k \varphi$.

7.2. The scaling equation

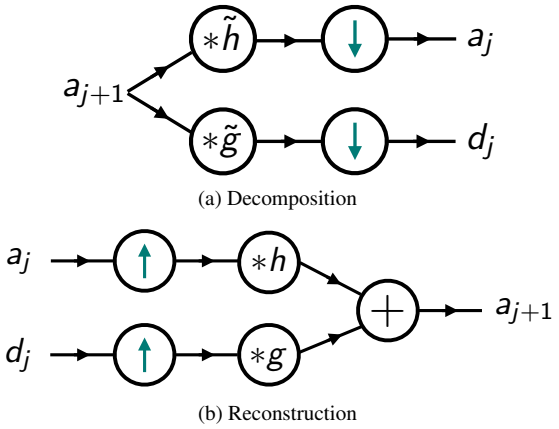
In this section we review the so-called scaling equation, which is key for understanding many fundamental aspects of MRAs, both theoretical and computational. We will heavily rely on it in the subsequent sections to set up the desired constraints and to efficiently compute with wavelets. The key observation is that since $V_0 \subset V_1$, there exists a unique sequence $h \in \ell^2(\mathbb{Z})$ such that

$$\varphi = \sum_{k\in\mathbb{Z}} h_k \varphi_{1k}. \tag{19}$$

This equation is referred to as the *scaling equation*; one of the fundamental properties of a scaling function. Similarly, since $\psi \in W_0 \subset V_1$, there exists a unique sequence $g \in \ell^2(\mathbb{Z})$, the so-called *high-pass filter* associated to h, such that

$$\psi = \sum_{k\in\mathbb{Z}} g_k \varphi_{1k}. \tag{20}$$

For Mallat's mother wavelet, we have $g_k = (-1)^{k-1} h_{1-k}$.



(a) Decomposition

(b) Reconstruction

Figure 8. (a) Decomposing approximation coefficients at level $j+1$ into approximation and detail coefficients at level j. Here $\tilde{h}$ and $\tilde{g}$ are defined in (22) and (23), respectively, and $*$ is the two-sided discrete convolution. The symbol $\downarrow$ corresponds to operator $S^{\downarrow}$, which downsamples a sequence by discarding all terms with odd index. (b) Reconstruction of the approximation coefficients at level $j+1$ from the approximation and detail coefficients at level j. The symbol $\uparrow$ corresponds to operator $S^{\uparrow}$, which upsamples a sequence by putting zeros in between every term.

7.3. The Discrete Wavelet Transform

The scaling equation (19) can be used to derive an efficient scheme for computing a (finite) multiresolution decomposition of a signal γ. More precisely, given initial approximation coefficients a_{j+1} at level $j+1$, the scaling equation can be used to compute the approximation and detail coefficients at level j. Conversely, the orthogonal decomposition $V_{j+1} = V_j \oplus W_j$ can be used to reconstruct a_{j+1} given the approximation and detail coefficients a_j and d_j, respectively, at resolution level j. The mapping associated to these operations is called the (1-level) Discrete Wavelet Transform (DWT). It provides an efficient way to obtain a multiresolution decomposition of a signal. The associated algorithm, which iteratively applies the 1-level DWT, is known as the so-called Pyramid Algorithm [26].

Decomposition Let $a_{j+1} \in \ell^2(\mathbb{Z})$ be approximation coefficients at an initial resolution level $j+1$, where $j \in \mathbb{Z}$. To obtain the approximation and detail coefficients at level j, we first note that

$$\varphi_{jk} = \sum_{l\in\mathbb{Z}} h_{l-2k}\varphi_{j+1,l}, \quad k \in \mathbb{Z}. \tag{21}$$

This relation between φ_{j+1} and φ_j can be easily derived by substituting the right hand side of the scaling equation (19)

into the definition of φ_{jk}. Consequently,

$$a_{jk} = \langle \gamma_{j+1}, \varphi_{jk} \rangle = \left(S^{\downarrow} \left(a_{j+1} * \tilde{h} \right) \right)_k, \quad \tilde{h}_k := h_{-k}, \tag{22}$$

where $* : \ell^2(\mathbb{Z}) \times \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ denotes the two-sided discrete convolution and $S_h^{\downarrow} : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ is defined by $(S^{\downarrow}(c))_k := c_{2k}$. The resulting map $a_{j+1} \mapsto S^{\downarrow} \left(a_{j+1} * \tilde{h} \right)$ is typically referred to as the DWT at level j. An analogous computation for the detail coefficients shows that

$$d_j = S^{\downarrow} \left(a_{j+1} * \tilde{g} \right), \quad \tilde{g}_k := g_{-k}. \tag{23}$$

The decomposition of the approximation coefficients at level $j+1$ into approximation and detail coefficients at level j is illustrated in Figure 8a.

Reconstruction The inverse DWT can be derived in a similar fashion using the decomposition $V_{j+1} = V_j \oplus W_j$. To make the computation explicit, we use (20) and the scaling equation again to write

$$\psi_{jk} = \sum_{l \in \mathbb{Z}} g_{l-2k} \varphi_{j+1,l}, \quad k \in \mathbb{Z}.$$

Consequently, since $V_{j+1} = V_j \oplus W_j$,

$$\begin{aligned} \gamma_{j+1} &= \sum_{k \in \mathbb{Z}} a_{jk} \varphi_{jk} + \sum_{k \in \mathbb{Z}} d_{jk} \psi_{jk} \\ &= \sum_{k,l \in \mathbb{Z}} \left(a_{jk} h_{l-2k} + d_{jk} g_{l-2k} \right) \varphi_{j+1,l} \\ &= \sum_{k \in \mathbb{Z}} \left(S^{\uparrow}(a_j) * h + S^{\uparrow}(d_j) * g \right)_k \varphi_{j+1,k}, \end{aligned}$$

where $S^{\uparrow} : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ is defined by

$$(S^{\uparrow} c)_k := \begin{cases} c_{\frac{k}{2}}, & k \equiv 0 \mod 2, \\ 0, & k \equiv 1 \mod 2. \end{cases}$$

This shows that the approximations coefficients at level $j+1$ are given by

$$a_{j+1} = S^{\uparrow}(a_j) * h + S^{\uparrow}(d_j) * g.$$

The reconstruction procedure is schematically shown in Figure 8b.

Remark 7.2 (Numerical implementation DWT). The convolutions appearing in the decomposition and reconstruction formulae can be efficiently computed using the Fast Fourier Transform (FFT).

7.4. Setting up constraints for wavelet filters

In this section we set up a finite system of equations whose solutions, under some mild non-degeneracy condition, correspond to wavelet filters. Recall that a wavelet filter is a sequence $h \in \ell^2(\mathbb{Z})$ that characterizes a scaling function φ. We reformulate the key requirements on φ, namely that its translates are orthogonal and Fourier transform is nonzero, in terms of the low pass filter h. In turn, this imposes constraints on admissible filters h in the form of a system of equations. Solutions of this system are commonly referred to as Quadratic Mirror Filters (QMFs), see Definition 7.5. We remark that these equations and conditions are well-known and refer the reader to [29, 31] for a more comprehensive treatment.

The refinement mask A first important observation follows from taking the Fourier Transform of the scaling equation, which yields

$$\hat{\varphi}(\xi) = H\left(\frac{\xi}{2} \right) \hat{\varphi}\left(\frac{\xi}{2} \right), \quad H(\xi) := \frac{1}{\sqrt{2}} \sum_{k \in \mathbb{Z}} h_k e^{-2\pi i \xi k}. \tag{24}$$

Here $H : [0,1] \to \mathbb{C}$ is a 1-period map typically referred to as the *refinement mask*. Throughout this paper, we shall abuse terminology and frequently refer to both H and h as the low pass filter associated to φ. Both the low pass filter and refinement mask completely characterize the scaling function. We will formulate necessary conditions on h by analyzing properties of the refinement mask H. The relation in (24) will be used extensively to derive these conditions.

Existence and uniqueness of MRAs The scaling equation plays a seminal role in establishing the existence and uniqueness of an MRA given a candidate h for a low-pass filter. While there is no need to explicitly construct φ, we do briefly discuss its existence here to justify the claim that we are learning wavelets. In addition, the discussion will reveal a necessary condition on H. The idea for proving the existence of a scaling map φ, given a low-pass filter h, is to "reconstruct" its Fourier transform $\hat{\varphi}$ using the scaling equation. To see how, suppose we start with a scaling map φ. Then repeated application of (24) yields

$$\hat{\varphi}(\xi) = \hat{\varphi}\left(\frac{\xi}{2^k} \right) \prod_{j=1}^{k} H\left(\frac{\xi}{2^j} \right), \quad \xi \in \mathbb{R}.$$

Assuming that $\hat{\varphi}$ is continuous at $\xi = 0$, we may consider the limit as $k \to \infty$, which yields

$$\hat{\varphi}(\xi) = \hat{\varphi}(0) \prod_{j=1}^{\infty} H\left(\frac{\xi}{2^j} \right), \tag{25}$$

provided the latter product exists. Since $\hat{\varphi}$ is not identically zero, we must have that $\hat{\varphi}(0) \neq 0$. This imposes a constraint on H, namely $H(0) = 1$. Without loss of generality, we may further assume that $\hat{\varphi}(0) = 1$.

Conversely, if we start with a sequence h instead of a scaling map φ, we may try to use the right-hand side of (25) to define a candidate for $\hat{\varphi}$. More precisely, if the infinite product converges to a map in $L^2(\mathbb{R})$, one may use the inverse Fourier transform to define a corresponding candidate for φ. As it turns out, if h decays sufficiently fast to zero, and we assume that $H(0) = 1$, where we now *define* H via (24), then $\xi \mapsto \prod_{j=1}^{\infty} H\left(\frac{\xi}{2^j}\right)$ is in $L^2(\mathbb{R})$, continuous at $\xi = 0$, and satisfies (25). For a more precise statement, we refer the reader to [12, 29]. In this paper, we exclusively deal with finite sequences h, for which these assumptions are always (trivially) satisfied. Hence we may use (25) to define a *candidate* for a scaling map φ. However, we still need to impose additional constraints on h, to ensure that the translates of φ are orthogonal.

Orthogonality To reformulate the orthogonality conditions into a system of equations for h, we first rewrite the system $\langle \varphi_{0k}, \varphi_{0l} \rangle = \delta_{kl}$ in frequency space. The recurrence relation for the Fourier transform of φ in (24) may then be used to derive a necessary condition on the refinement mask H. Subsequently, we can reformulate this necessary condition as an equivalent condition on h. The details can be found in [31]. Here we only state the relevant results.

Lemma 7.3 (Orthogonality refinement mask)**.** *Suppose $\varphi \in L^2(\mathbb{R})$ satisfies the dilation equation for a refinement mask H with Fourier coefficients $h \in \ell^2(\mathbb{Z})$. If the family $(\varphi_{0k})_{k\in\mathbb{Z}}$ is orthonormal, then*

$$|H(\xi/2)|^2 + |H(\xi/2 + 1/2)|^2 = 1, \tag{26}$$

for a.e. $\xi \in \mathbb{R}^2$.

Proof. See [31]. □

Remark 7.4. The condition in (26) is often referred to as the Quadratic Mirror Filter condition.

Definition 7.5 (Quadratic Mirror Filter)**.** A Quadratic Mirror Filter (QMF) is a sequence $h \in \ell^2(\mathbb{Z})$ which satisfies (26) and $H(0) = 1$.

The reason for introducing this terminology is that QMFs correspond to wavelet filters under an additional non-degeneracy condition. Here we only state the result for finite filters.

Theorem 7.6. *Suppose h is a finite QMF. If $\inf_{0\leq\xi\leq\frac{1}{4}} |H(\xi)| > 0$, then*

$$\varphi := \mathcal{F}^{-1}\left(\xi \mapsto \prod_{j=1}^{\infty} H\left(\frac{\xi}{2^j}\right)\right)$$

is a scaling function and defines an MRA of $L^2(\mathbb{R})$. Here $\mathcal{F} : L^2(\mathbb{R}) \to L^2(\mathbb{R})$ denotes the Fourier transform.

Proof. See [38] Theorem 8.35. □

Remark 7.7. One may expect that any finite filter h satisfying (26) will define a scaling function whose translates are orthogonal. However, this is unfortunately not the case, and the additional requirement that $\inf_{0\leq\xi\leq\frac{1}{4}} |H(\xi)| > 0$ is needed to avoid degenerate cases.

Next, we derive a system of equations for h that is equivalent to (26). To formulate this system of equations, we define operators $M, R : \ell^2(\mathbb{Z}) \to \ell^2(\mathbb{Z})$ by $(Mc)_k := (-1)^k c_k$ and $(Rc)_k := c_{-k}$. For brevity, we will frequently write $\tilde{c} := R(c)$ as before. Even though we are dealing with real-valued filters h in practice, below we state the results for general complex-valued sequences.

Lemma 7.8 (Orthogonality low-pass filter)**.** *Suppose H is a refinement mask with Fourier coefficients $h \in \ell^2(\mathbb{Z})$. Then the orthonormality constraint in (26) is equivalent to the following system of equations:*

$$\begin{cases} \sum_{l\in\mathbb{Z}} |h_l|^2 = 1, & k = 0, \\ \sum_{l\in\mathbb{Z}} h_{l-2k}\overline{h_l} = 0, & k \in \mathbb{N}. \end{cases} \tag{27}$$

Proof. We start by computing the Fourier coefficients $c(h) \in \ell^1(\mathbb{Z})$ of the lefthand-side of (26). To this end, observe that the 2-periodic map $\xi \mapsto H\left(\frac{\xi}{2}\right)$ and its conjugate have Fourier coefficients $\frac{1}{\sqrt{2}}\tilde{h}$ and $\frac{1}{\sqrt{2}}\overline{h}$, respectively. Therefore, since $H \in L^2([0,1])$, the product $\xi \mapsto \left|H\left(\frac{\xi}{2}\right)\right|^2$ is L^1 with Fourier coefficients $\frac{1}{2}\tilde{h} * \overline{h}$. Similarly, the Fourier coefficients of $\xi \mapsto \left|H\left(\frac{\xi+1}{2}\right)\right|^2$ are given by $\frac{1}{2}M(\tilde{h}) * M(\overline{h})$. Hence

$$2c(h) = \tilde{h} * \overline{h} + M(\tilde{h}) * M(\overline{h}).$$

Unfolding the definitions yields

$$(c(h))_k = \frac{1}{2}\sum_{l\in\mathbb{Z}} \left(1 + (-1)^k\right) h_{l-k}\overline{h}_l, \quad k \in \mathbb{Z}.$$

Note that $(c(h))_k = 0$ whenever k is odd, since

$$\left((-1)^k + 1\right) = \begin{cases} 2, & k \equiv 0 \mod 2, \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

The equation in (26) is equivalent to the statement that $(c(h))_k = \delta_{0k}$ for $k \in \mathbb{Z}$, since the Fourier coefficients of a L^1-function are unique. Hence (26) is equivalent to $(c(h))_{2k} = \delta_{0,2k}$ for $k \in \mathbb{Z}$ by the observation in (28). Finally, the latter statement is equivalent to $(c(h))_{2k} = \delta_{0,2k}$ for $k \in \mathbb{N}_0$, since

$$\overline{\sum_{l\in\mathbb{Z}} h_{l-2k}\overline{h_l}} = \sum_{l\in\mathbb{Z}} h_{l+2k}\overline{h_l}$$

for any $k \in \mathbb{Z}$. The two cases in (27) show the demands for $k = 0$ and positive even indices, respectively. This establishes the result. □

Remark 7.9. A more direct way to arrive at (27) is to plug in the dilation relation into $\langle T_k\varphi, \varphi\rangle$ and use the orthogonality of $(\varphi_{1k})_{k\in\mathbb{Z}}$. The equivalence with (26) can then be established in a similar (but slightly different) way.

QMF conditions We are now ready to set up the desired constraints. In general, the QMF conditions are not sufficient to guarantee that h is the low pass filter of a scaling function, see the discussion in Remark 7.7. However, in numerical experiments, we never seem to violate the non-degeneracy condition when only imposing the QMF conditions. For this reason, the only constraints that we impose are the QMF conditions. We do provide an option to include the non-degeneracy condition in Remark 7.12.

To properly write down the QMF conditions as constraints on a sequence h, we introduce some additional notation. Let $\mathcal{A}_M(\mathbb{R})$ denote the space of one-dimensional $\mathbb{R}$-valued two-sided sequences of order M, i.e.,

$$\mathcal{A}_M(\mathbb{R}) := \left\{ a \;\middle|\; a : \{1-M, \dots, M-1\} \to \mathbb{R} \right\}.$$

Note that $\mathcal{A}_M(\mathbb{R})$ is a vector space over $\mathbb{R}$ of dimension $2M-1$. In particular, $\mathcal{A}_M \simeq \mathbb{R}^{2M-1}$. The reason for introducing this notation is to explicitly keep track of the two-sided ordering of sequences. We are now ready to gather all the demands that we have derived, and place them into the general framework of Section 2.

Definition 7.10. Let $M \in \mathbb{N}_{\geq 3}$ be a prescribed order. The QMF-map is the function $F_M : \mathcal{A}_M(\mathbb{R}) \to \mathbb{R}^M$ defined by

$$(F_M(h))_k := \begin{cases} (h^- * h)_0 - 1, & k = 0, \\ (h^- * h)_{2k}, & 1 \leq k \leq M-1, \\ -\sqrt{2} + \sum_{|l|\leq M-1} h_l, & k = M. \end{cases}$$

The first M equations correspond to the orthonormality constraints. Note that we only have to impose $(h^- * h)_{2k} = 0$ for $1 \leq k \leq M-1$, since $(h^- * h)_{2k} = 0$ for $k \geq M$. The last equation corresponds to the condition that $H(0) = 1$. The set of regular points in $F_M^{-1}(0)$ is a real-analytic $(M-2)$-dimensional submanifold of $\mathbb{R}^{2M-1}$ by Remark 6.3. In particular, we can get as many degrees of freedom as desired by choosing a sufficiently large order M.

We summarize the interpretation and importance of the constraints in a theorem.

Theorem 7.11. *If $F_M(h) = 0$ and $\inf_{0\leq\xi\leq\frac{1}{4}} |H(\xi)| > 0$, then h is the low-pass filter of a scaling map φ.*

Remark 7.12 (Imposing the non-degeneracy condition). The additional non-degeneracy condition $\inf_{0\leq\xi\leq\frac{1}{4}} |H(\xi)| > 0$ can be imposed, for instance, by requiring that H has no zeros in $[0, \frac{1}{4}]$. Since we consider finite filters only, the refinement mask H is analytic (entire even). Hence the latter condition may be imposed by requiring that

$$\oint_{\partial\mathcal{E}_r} \frac{H'(z)}{H(z)}\,\mathrm{d}z = 0, \tag{29}$$

where $\mathcal{E}_r \subset \mathbb{C}$ is an ellipse with foci 0 and $\frac{1}{4}$ and $r > 0$ is a free parameter which controls the sum of the major and minor axis. We remind the reader that the above integral counts the zeros of H (up to a scaling factor) in $\mathcal{E}_r$, provided H has no zeros on $\partial\mathcal{E}_r$. For any parameterization of $\partial\mathcal{E}_r$, we can numerically evaluate the integrand of (29) on an associated uniform grid by using the Fourier expansion of H. We may therefore numerically compute a Fourier expansion of the integrand, which in turn allows numerical approximation of the contour integral.

8. Supplementary Information Contour Prediction

In this section we provide the details of our auto-contouring application. In Sec. 8.1 we explain how to represent periodic curves using wavelets. In Sec. 8.2 we provide additional details about the data, e.g., how ground-truth curves are constructed, what preprocessing steps are taken, etc.. Finally, in Sec. 8.3, we present the full details of our network architecture and training schedule. In addition, we provide more examples of learned wavelets.

8.1. Wavelet Representations of periodic curves

We start by explaining how to compute a multiresolution decomposition of a scalar-valued *periodic* signal γ with period $\tau > 0$. First, we address the issue that periodic signals are not contained in $L^2(\mathbb{R})$ by considering the cut-off $\tilde{\gamma}(t) := \gamma(t)\mathbf{1}_{[-\tau,\tau]}(t)$, which is contained in $L^2(\mathbb{R})$. In general, such a cut-off will introduce discontinuities at the boundary points $-\tau$ and τ. These artifacts do not present an

issue for us, however, since (by periodicity) we can restrict our analysis to a strict subset $[I_0, I_1] \subset [-\tau, \tau]$ of length τ.

To compute a multiresolution decomposition of $\tilde{\gamma}$ using the DWT, we need to compute the approximation coefficients $a_{j_1}(\tilde{\gamma}) \in \ell^2(\mathbb{Z})$ of $\tilde{\gamma}$ at some initial resolution level $j_1 \in \mathbb{N}$. To explain how such an initial approximation can be obtained in the first place, we derive an explicit formula for the approximation coefficients $a_{jk}(\tilde{\gamma}) = \langle \tilde{\gamma}, \varphi_{jk} \rangle$. While we will not directly use this formula, we do remark it can be efficiently implemented and provides an alternative method to initialize wavelet coefficients thereby addressing the so-called wavelet crime [1, 23]. For our purposes, this expression will be key for identifying which coefficients to consider, i.e., which spatial locations $k \in \mathbb{Z}$ associated to $a_{jk}(\tilde{\gamma})$ are relevant for representing $\tilde{\gamma}$.

Lemma 8.1 (Initialization approximation coefficients)**.** *Let $\varphi \in L^2(\mathbb{R})$ be the scaling map of an MRA with low-pass filter $h \in \ell^2(\mathbb{Z})$ and associated refinement mask H. Assume h is nonzero for only a finite number of indices $k \in \mathbb{Z}$ so that $supp(\varphi) \subset [-r_1, r_2]$ for some $r_1, r_2 > 0$. If $\gamma \in C^2_{per}([0, \tau])$ is a τ-periodic map with Fourier coefficients $(\gamma_m)_{m \in \mathbb{Z}}$, then*

$$\langle \tilde{\gamma}, \varphi_{jk} \rangle = 2^{-\frac{j}{2}} \sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{k}{2^j}} \prod_{n=1}^{\infty} H\left(-\frac{m}{\tau 2^{j+n}}\right), \tag{30}$$

for any $j \in \mathbb{Z}$ and $k \in \{\lceil r_1 - 2^j\tau \rceil, \ldots, \lfloor 2^j\tau - r_2 \rfloor\}$, where $\omega(\tau) := \frac{2\pi}{\tau}$ is the angular frequency of γ.

Proof. Let $j \in \mathbb{Z}$ and $k \in \{\lceil r_1 - 2^j\tau \rceil, \ldots, \lfloor 2^j\tau - r_2 \rfloor\}$ be arbitrary. A change of variables shows that

$$\langle \tilde{\gamma}, \varphi_{jk} \rangle = 2^{-\frac{j}{2}} \int_{[r_1, r_2]} \tilde{\gamma}\left(2^{-j}(t+k)\right) \varphi(t) \, \mathrm{d}t,$$

since $\mathrm{supp}(\varphi) \subset [-r_1, r_2]$. Note that the latter holds for all $k \in \mathbb{Z}$. For $k \in \{\lceil r_1 - 2^j\tau \rceil, \ldots, \lfloor 2^j\tau - r_2 \rfloor\}$ in particular, we have that $2^{-j}(t+k) \in [-\tau, \tau]$ for all $t \in [-r_1, r_2]$. Therefore, for such k, we may plug in the Fourier expansion for $\tilde{\gamma}$ and compute

$$\begin{aligned} &\int_{[-r_1, r_2]} \tilde{\gamma}\left(2^{-j}(t+k)\right) \varphi(t) \, \mathrm{d}t = \\ &\qquad = \int_{[-r_1, r_2]} \sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{t+k}{2^j}} \varphi(t) \, \mathrm{d}t. \end{aligned}$$

Next, note that that series inside the integral converges pointwise to $\gamma\left(2^{-j}(t+k)\right)\varphi(t)$ on $[-r_1, r_2]$. Furthermore, the partial sums can be bounded from above on $[-r_1, r_2]$ by a constant, since $\gamma \in C^2_{\mathrm{per}}([0, \tau])$ and φ is bounded. Therefore, we may interchange the order of summation and integration

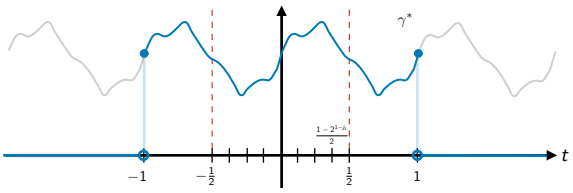


Figure 9. The re-parameterized cut-off signal $\gamma^*(t) = \gamma(\tau t)\mathbf{1}_{[-1,1]}(t)$ depicted in blue. We only need to compute approximation coefficients associated to the smaller region $[-\frac{1}{2}, \frac{1}{2}]$.

by the Dominated Convergence Theorem:

$$\begin{aligned} &\int_{[-r_1, r_2]} \sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{t+k}{2^j}} \varphi(t) \, \mathrm{d}t = \\ &\qquad \sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{k}{2^j}} \int_{[-r_1, r_2]} e^{i\omega(\tau) m \frac{t}{2^j}} \varphi(t) \, \mathrm{d}t. \end{aligned}$$

Finally, changing the domain of integration to $\mathbb{R}$ again, we see that

$$\begin{aligned} &\sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{k}{2^j}} \int_{[-r_1, r_2]} e^{i\omega(\tau) m \frac{t}{2^j}} \varphi(t) \, \mathrm{d}t = \\ &\quad = \sum_{m \in \mathbb{Z}} \gamma_m e^{i\omega(\tau) m \frac{k}{2^j}} \hat{\varphi}\left(-\frac{m}{\tau 2^j}\right). \end{aligned}$$

The stated result now follows from the observation that $\hat{\varphi}(\xi) = \prod_{l=1}^{\infty} H(\frac{\xi}{2^l})$ holds pointwise for any $\xi \in \mathbb{R}$, since h is nonzero for only a finite number of indices, see [29, Theorem 8.34]. □

Remark 8.2. It is straightforward to show that the partial sums converge uniformly on $[-r_1, r_2]$. It is therefore not needed to resort to the Dominated Convergence Theorem.

Remark 8.3. The bounds $\lceil r_1 - 2^j\tau \rceil$ and $\lfloor 2^j\tau - r_2 \rfloor$ are the smallest and largest integer, respectively, for which the Fourier series for γ can be plugged into $\langle \tilde{\gamma}, \varphi_{jk} \rangle$. The bounds are somewhat artificial, however, since the argument may be repeated for any cut-off of γ on $[-s\tau, s\tau]$, where $s \in \mathbb{N}_{\geq 2}$. The choice for s is ultimately irrelevant, however, since we are interested in the minimal number of approximation coefficients needed to cover γ; see the discussion below.

Lemma 8.1 provides a convenient way to initialize approximation coefficients. To explain how, we first re-parameterize γ to have period 1 and consider the cut-off $\gamma^*(t) := \gamma(\tau t)\mathbf{1}_{[-1,1]}(t)$. The motivation for this re-parameterization is that we can now conveniently relate specific approximation coefficients to sample values of γ. To be more precise, recall that $\hat{\varphi}$ is continuous at zero and $H(0) = 1$. Therefore, if the initial resolution level j_1 is sufficiently large, the infinite product in (30) will be close to 1 (for small m). Furthermore, in practice, we have a finite

number of Fourier coefficients, i.e., $\gamma_m = 0$ for $|m| \geq N$. Therefore, if j_1 is sufficiently large relative to N, then

$$a_{j_1 k}(\gamma^*) \approx 2^{-\frac{j_1}{2}} \gamma^*(k 2^{-j_1}), \quad \lceil r_1 - 2^{j_1} \rceil \leq k \leq \lfloor 2^{j_1} - r_2 \rfloor. \tag{31}$$

That is, on sufficiently high-resolution levels the approximation coefficients are close to the (scaled) sample values of the underlying signal; a well-known general fact of MRAs. Consequently, the approximation coefficients needed to cover $[-1, 1]$ (approximately) are $(a_{j_1 k}(\gamma^*))_{k=\lceil r_1 - 2^{j_1} \rceil}^{\lfloor 2^{j_1} - r_2 \rfloor}$. Motivated by this observation, and the fact that we only need γ^* on $[-\frac{1}{2}, \frac{1}{2}]$, we use the scaled sample values in (31) to initialize the coefficients $(a_{j_1 k}(\gamma^*))_{k=-2^{j_1-1}}^{2^{j_1-1}-1}$, which cover $[-\frac{1}{2}, \frac{1-2^{1-j_1}}{2}]$ approximately, see Figure 9.

We stress that in order for the above approximations to be accurate, the initial resolution level j_1 needs to be sufficiently large. Furthermore, to ensure that $-2^{j_1-1} > \lceil r_1 - 2^{j_1} \rceil$ and $2^{j_1-1} - 1 < \lfloor 2^{j_1} - r_2 \rfloor$, we require that

$$j_1 \geq \max \left\{ \left\lceil \frac{\log(r_1 + 1)}{\log(2)} + 1 \right\rceil, \left\lceil \frac{\log(r_2 - 1)}{\log(2)} + 1 \right\rceil \right\}.$$

One can explicitly express the support of φ in terms of the order M of the wavelet. Specifically, the scaling relation can be used to shown that supp $\varphi \subset [1 - M, M - 1]$, thus providing explicit values for r_1 and r_2. A rigorous proof is out of the scope of this paper and we refer the reader to [38, Theorem 8.38].

8.2. Ground Truth

Let $(x, y) \in \mathcal{X} \times \mathbb{R}^{n_s \times n_p}$ be an image (slice) - contour pair, where x is a slice of the CT or MRI scan, y is a sequence of $n_p \in \mathbb{N}$ points approximating the boundary of a simply connected region $R = R(x)$, and $n_s = 2$ is the number of spatial components. Since we only have access to binary masks, and not to the raw annotations themselves, we extract y using OPENCV. We remark that y is not constrained to an integer-valued grid.

Approximation coefficients The ground truth consists of the approximation coefficients of γ^* at an initial resolution level $j_2 \in \mathbb{N}$. Here γ^* is the re-parameterized cut-off of an initial parameterization γ of ∂R as explained in the previous section. We approximate the approximation coefficients using (31), which requires evaluating γ^* on a dyadic grid. To accomplish this, we compute a Fourier expansion for γ. To be more precise, we first parameterize ∂R by arc length resulting in a curve γ. The arc length τ is approximated by summing up the Euclidian distances between subsequent points on y. We re-parameterize γ to have period 1, as explained in Sec. 8.1, and additionally "center" it using the average midpoint of the contours in the training set. The

ROI	$\lvert\mathcal{D}_{\text{train}}\rvert$	$\lvert\mathcal{D}_{\text{val}}\rvert$	$\lvert\mathcal{D}_{\text{test}}\rvert$
Spleen	2509	386	371
Prostate	454	77	63

Table 2. The number of samples (slices) in the train-val-test splits for the prostate and spleen. This count includes empty slices, i.e., slices which do not contain a contour. The split was made on volume (patient) level.

Fourier coefficients of the resulting contour are then computed by evaluating it on an equispaced grid of $[0, 1]$ of size $2N-1$, where $N \in \mathbb{N}$, using linear interpolation and the Discrete Fourier Transform. Since the contours are real-valued, we only store the Fourier coefficients $(\tilde{\gamma}_m)_{m=0}^{N-1} \in (\mathbb{C}^{n_s})^N$.

Fourier coefficients that are too small, i.e., have no relevant contribution, are set to zero. To be more precise, note that the magnitude of the approximated Fourier coefficients will typically stagnate and stay constant (approximately) beyond some critical order, since all computations are performed in finite (single) precision. We locate this critical order $m_0^*(s) \in \mathbb{N}$ for each component $s \in \{1, 2\}$, if present, by iteratively fitting the best line, in the least squares sense, through the points $\left\{ \left(m, \left\| (|[\tilde{\gamma}_{\tilde{m}}]_s|)_{\tilde{m}=m_0}^{m} \right\|_1 \right) : m_0 \leq m \leq N-1 \right\}$ for $1 \leq m_0 \leq N-1$. We iterate this process until the residual is below a prescribed threshold $\delta_N > 0$. In practice, we set $\delta_N = 0.1$. The Fourier coefficients with index strictly larger than $m_0^*(s)$ are set to zero. Finally, we use the approximation in (31) to initialize the approximation coefficients a_{j_2}.

Consistency To have consistent parameterizations for all slices, we ensure that ∂R is always traversed anti-clockwise (using `opencv`). Furthermore, since parameterizations are only determined up to a translation in time, we need to pick out a specific one. We choose the unique parameterization such that γ^* starts at angle zero at time zero relative to the midpoint $c = (c_1, c_2) \in \mathbb{R}^2$ of R. This is accomplished by exploiting the Fourier representation of γ. More precisely, let

$$\gamma(t) = \sum_{|m| \leq N-1} \tilde{\gamma}_m e^{i\omega(\tau) m t}, \quad \omega(\tau) = \frac{2\pi}{\tau},$$

be the initial contour with Fourier coefficients $\eta := (\tilde{\gamma}_m)_{m=1-N}^{N-1}$. The midpoint c of the region enclosed by γ is given by (by Green's theorem)

$$c_s = \frac{1}{\lambda(R)} \int_R u_s \, \mathrm{d}\lambda(u_1, u_2) = (-1)^s \frac{([\eta]_1 * [\eta]_2 * [\eta']_s)_0}{([\eta]_1 * [\eta']_2)_0}, \tag{32}$$

where $s \in \{1, 2\}$. Here λ denotes the Lebesgue measure on $\mathbb{R}^2$ and $[\eta]_s$, $[\eta']_s$ are the Fourier coefficients of $[\gamma]_s$ and its

derivative, respectively.

We can now compute the desired parameterization by determining $t_0 \in [0, \tau]$ such that

$$\arccos\left(\frac{[\gamma(-t_0) - c]_1}{\|\gamma(-t_0) - c\|_2}\right) \approx 0,$$

and then use the shifted parameterization $t \mapsto \gamma(t - t_0)$. While t_0 can be easily found using Newton's method, it suffices in practice to simply re-order y from the start, before computing the Fourier coefficients of γ. More precisely, we first define a shift $\tilde{y}$ of y by

$$\tilde{y}_k := y_{k + k^* \bmod n_p},$$
$$k^* := \operatorname{argmin}\left\{\arccos\left(\frac{[y_k - c]_1}{\|y_k - c\|_2}\right)\right\}_{k=0}^{n_p - 1},$$

where $0 \leq k \leq n_p - 1$, and then compute the Fourier coefficients of the resulting curve.

The resulting dataset $\mathcal{D}$ thus consists of tuples $(x, a_{j_2}(\gamma^*(x)))$. Before feeding the images x into the model, we linearly rescale the image intensities at each instance to $[0, 1]$. Furthermore, we use extensive data augmentation: we use random shifts, random rotations, random scaling, elastic deformations and horizontal shearing. A custom (random) split of the available data was made to construct a train-validation-test split. The sizes of the datasets are reported in Table 2.

8.3. Architecture and Training Details

In this section, we provide the details of the network architecture and optimization procedure.

8.3.1 Architecture

Our network is a hybrid analog of the U-Net. It consists of a two-dimensional convolutional encoder, a bottleneck of fully connected layers, and a one-dimensional decoder. The encoder and decoder are connected through skip-connections. The approximation and detail coefficients at the lowest resolution level j_0 are predicted in the bottleneck. Afterwards, the Pyramid Algorithm takes over to compute approximation coefficients at higher resolution levels (the decoder) using learnable wavelet filters. The needed detail coefficients at the higher resolution levels are predicted using the skip-connections. In practice, the detail coefficients are negligible on sufficiently high-resolution levels. For this reason, we only predict detail coefficients up to a prescribed level j_1. The predictions at higher resolution levels $j_1 < j \leq j_2$ are computed without detail coefficients. The specific values for the architecture were determined using a hyperparameter search.

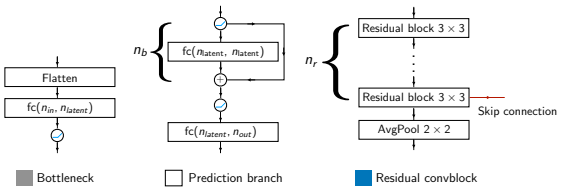


Figure 10. The components of the network: the bottleneck, fully connected prediction layers, and convolutional block, respectively.

Encoder The encoder consists of $n_d \in \mathbb{N}$ down-sampling blocks. Each block consists of $n_r \in \mathbb{N}$ (convolutional) residual blocks, using GELU-activation and kernels of size 3×3, followed by an average-pooling layer of size 2×2. The initial number of filters $n_f \in \mathbb{N}$ used in the first block is doubled after each other block. For example, if $n_d = 5$ and the number of kernels at the first block is $n_f = 32$, then the subsequent blocks have 32, 64, 64, and 128, kernels, respectively.

Bottleneck The encoder is followed by a bottleneck which consists of a stack of fully connected layers. The first layer in the bottleneck compresses the feature map from the encoder path to a feature map with n_c channels using a 1×1 convolution. Next, this compressed feature map is transformed to a vector in $\mathbb{R}^{n_{\text{lat}}}$, where $n_{\text{lat}} \in \mathbb{N}$ refers to the latent dimension of the MLP. Attached to this layer are four branches to predict the approximation and detail coefficients $[v_{j_0}(x)]_s, [w_{j_0}(x)]_s \in \mathbb{R}^{2^{j_0}}$, respectively. Here $s \in \{1, 2\}$ corresponds to the spatial component of the contour. Each branch consists of $n_b \in \mathbb{N}$ fully-connected layers. The first $n_b - 1$ layers map from $\mathbb{R}^{n_{\text{lat}}}$ to itself with GELU-activation and residual connections in between. The final layer transforms the n_{lat}-dimensional output to an element in $\mathbb{R}^{2^{j_0}}$.

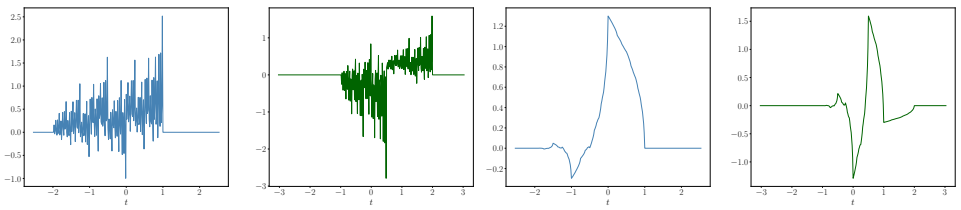
Decoder The detail coefficients at levels $j_0 \leq j < j_1$ are predicted using skip-connections. For each skip-connection, we first compress the feature map from the encoder path to a feature map with n_c channels using a 1×1 convolution. Subsequently, two prediction branches, each having the same architecture as above, are used to predict the detail coefficients in $\mathbb{R}^{2^j}$ (one for each spatial component). The predicted approximation coefficients at level j_0 and detail coefficients at levels $j_0 \leq j \leq j_1 - 1$ are used as input to the Pyramid algorithm to reconstruct approximation coefficients up to level j_1 using learnable wavelet filters. The approximation coefficients at levels $j_1 + 1 \leq j \leq j_2$ are reconstructed without detail coefficients.

Hyperparameters The choices for the hyperparameters were based on a hyperparameter search, optimizing the Dice score. For the spleen and prostate we

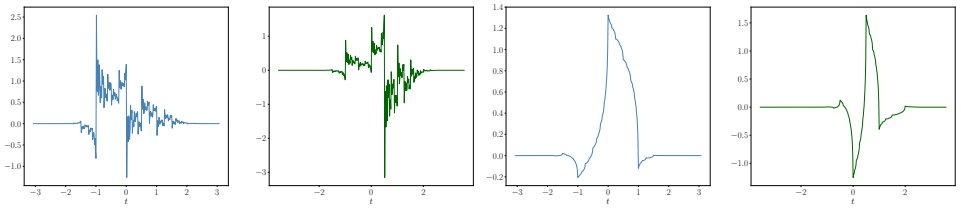
have set $(n_d, n_r, n_{\text{lat}}, n_b, n_c, j_2) = (6, 4, 124, 3, 16, 7)$ and $(n_d, n_r, n_{\text{lat}}, n_b, n_c, j_2) = (5, 4, 116, 2, 16, 7)$ respectively, and considered wavelet orders $3 \leq M \leq 8$. Furthermore, for each order, we used the lowest possible resolution level j_0 and $j_1 = j_2$. In particular, $j_0(M) = 3$ for $M \in \{3, 4\}$ and $j_0(M) = 4$ for $M \in \{5, 6, 7, 8\}$.

8.4. Wavelet Examples

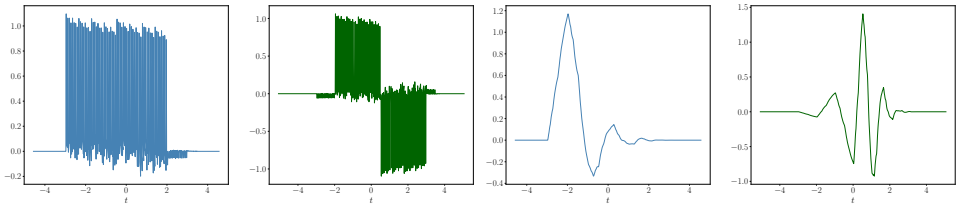
In Figures 11, 12, 13, 14 we show more examples of initialized and task-optimized wavelets.



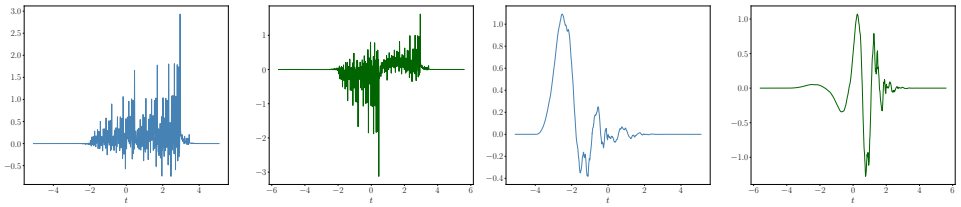
(a) Order 3 - initial wavelet (b) Order 3 - task-optimized wavelet



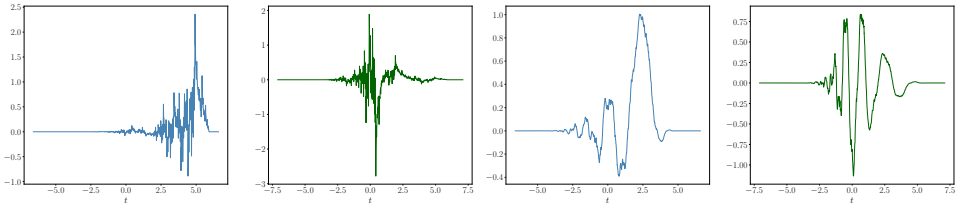
(c) Order 4 - initial wavelet (d) Order 4 - task-optimized wavelet



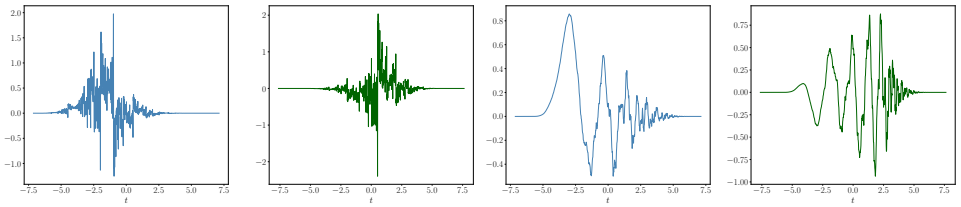
(e) Order 5 - initial wavelet (f) Order 5 - task-optimized wavelet



(g) Order 6 - initial wavelet (h) Order 6 - task-optimized wavelet

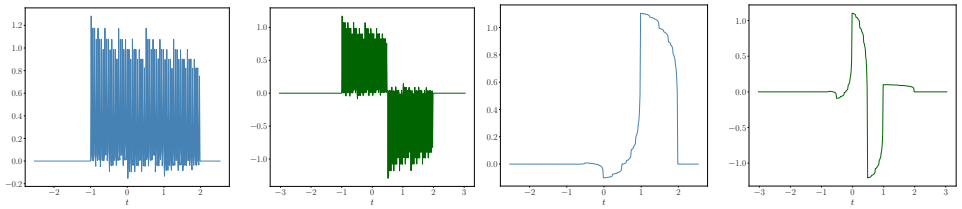


(i) Order 7 - initial wavelet (j) Order 7 - task-optimized wavelet



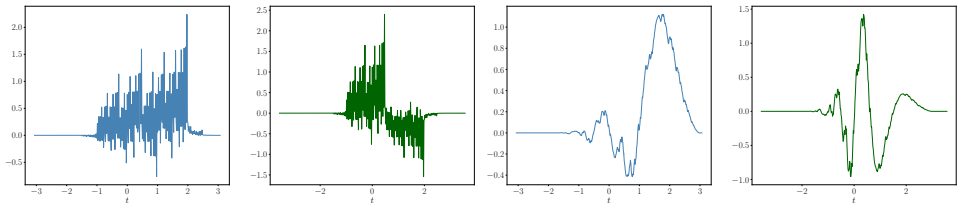
(k) Order 8 - initial wavelet (l) Order 8 - task-optimized wavelet

Figure 11. Initial and final learned wavelets for different orders for first spatial component of spleen.



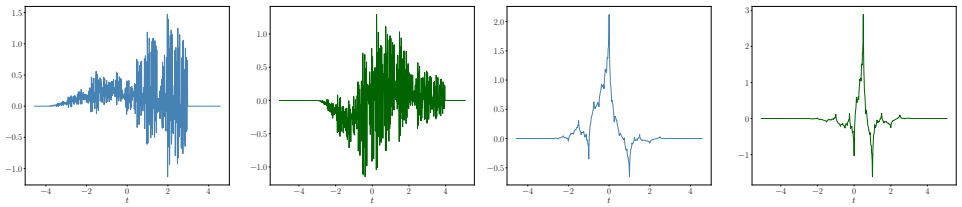
(a) Order 3 - initial wavelet

(b) Order 3 - task-optimized wavelet



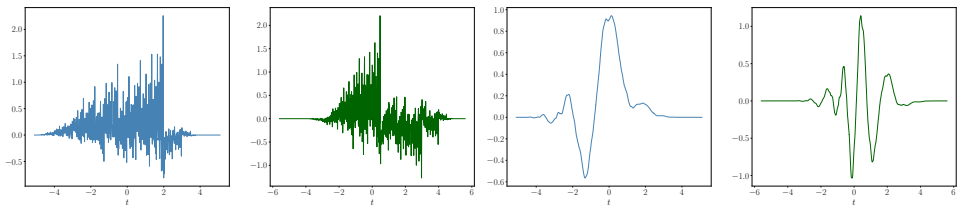
(c) Order 4 - initial wavelet

(d) Order 4 - task-optimized wavelet



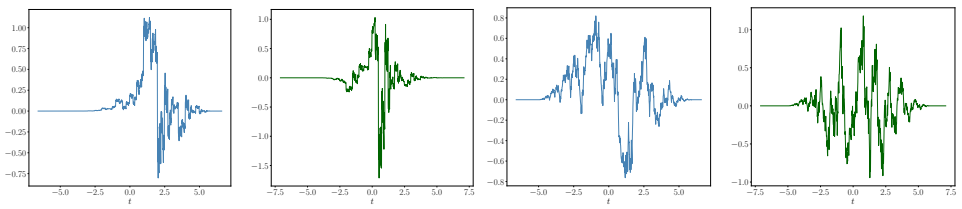
(e) Order 5 - initial wavelet

(f) Order 5 - task-optimized wavelet



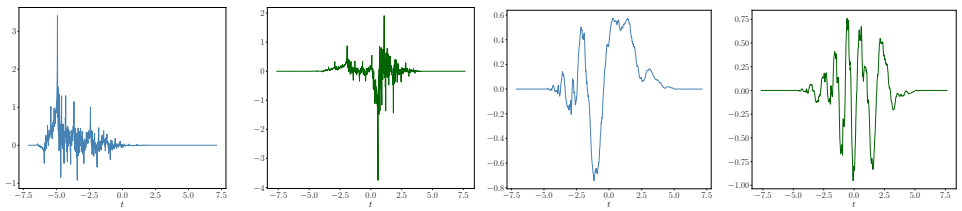
(g) Order 6 - initial wavelet

(h) Order 6 - task-optimized wavelet



(i) Order 7 - initial wavelet

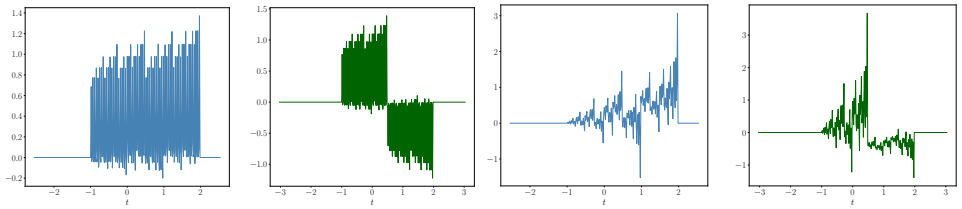
(j) Order 7 - task-optimized wavelet



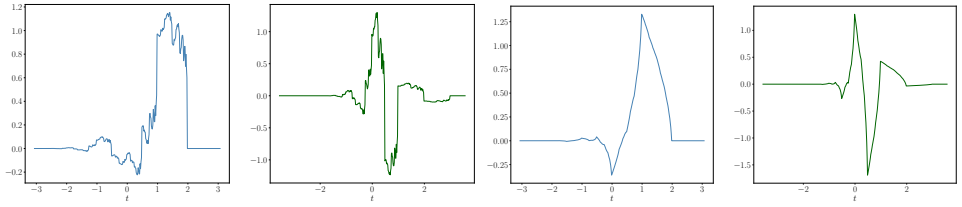
(k) Order 8 - initial wavelet

(l) Order 8 - task-optimized wavelet

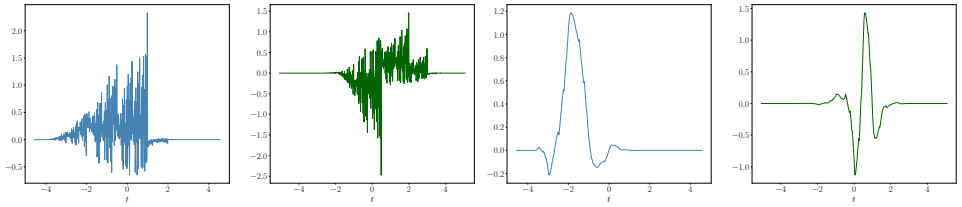
Figure 12. Initial and final learned wavelets for different orders for second spatial component of spleen.



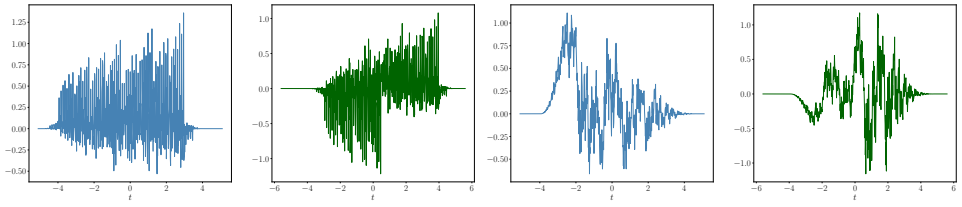
(a) Order 3 - initial wavelet (b) Order 3 - task-optimized wavelet



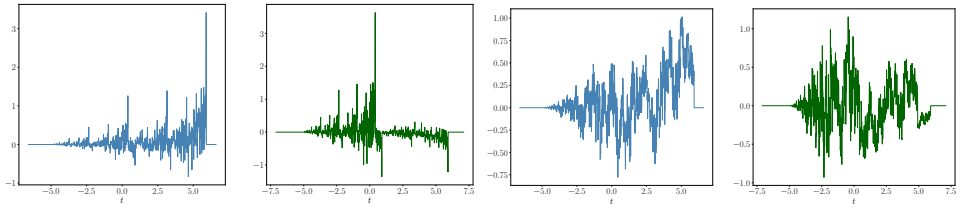
(c) Order 4 - initial wavelet (d) Order 4 - task-optimized wavelet



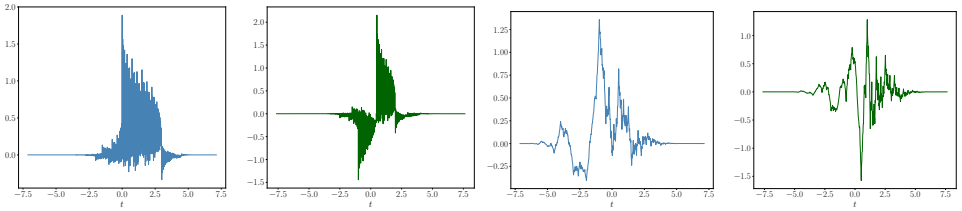
(e) Order 5 - initial wavelet (f) Order 5 - task-optimized wavelet



(g) Order 6 - initial wavelet (h) Order 6 - task-optimized wavelet

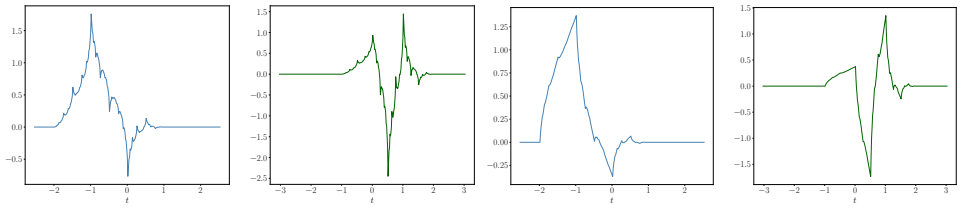


(i) Order 7 - initial wavelet (j) Order 7 - task-optimized wavelet

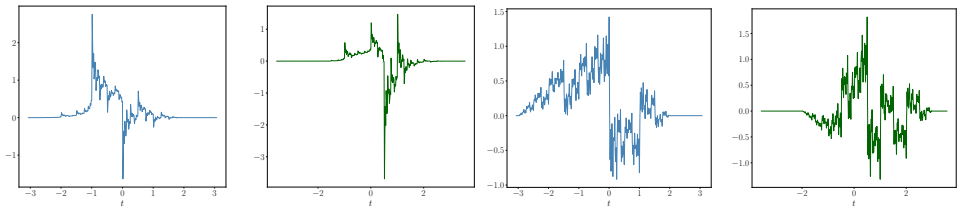


(k) Order 8 - initial wavelet (l) Order 8 - task-optimized wavelet

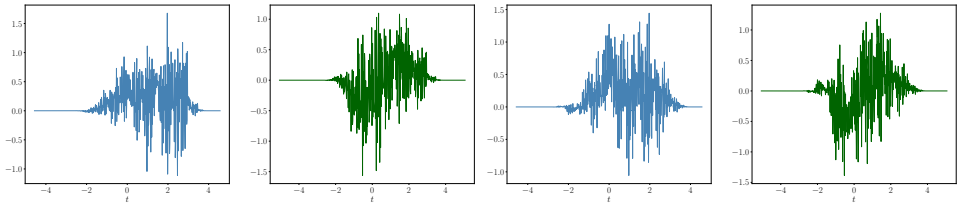
Figure 13. Initial and final learned wavelets for different orders for first spatial component of prostate.



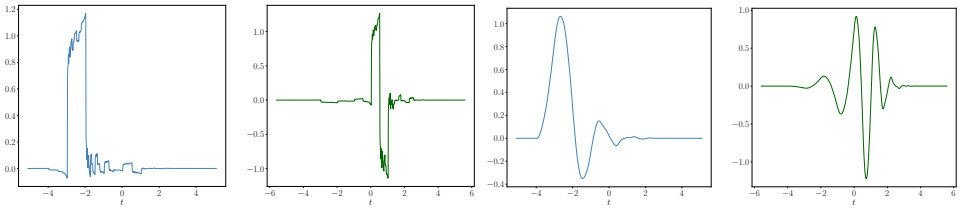
(a) Order 3 - initial wavelet (b) Order 3 - task-optimized wavelet



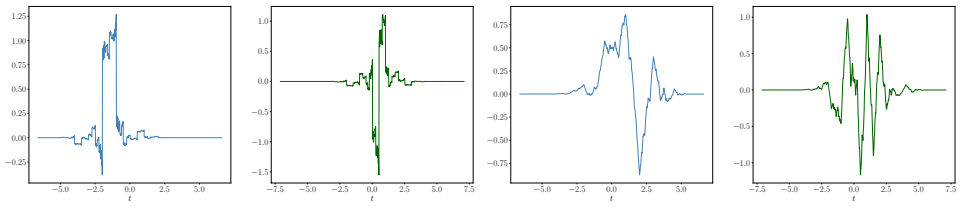
(c) Order 4 - initial wavelet (d) Order 4 - task-optimized wavelet



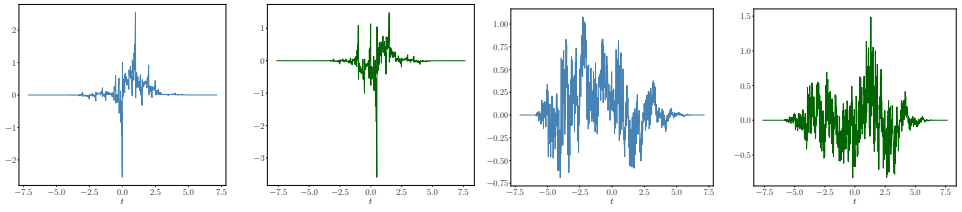
(e) Order 5 - initial wavelet (f) Order 5 - task-optimized wavelet



(g) Order 6 - initial wavelet (h) Order 6 - task-optimized wavelet



(i) Order 7 - initial wavelet (j) Order 7 - task-optimized wavelet



(k) Order 8 - initial wavelet (l) Order 8 - task-optimized wavelet

Figure 14. Initial and final learned wavelets for different orders for second spatial component of prostate.