

Supplementary Materials

Three algorithms (creeping random search, simulated annealing, and pseudo-derivative) used in projection pursuit guided tour optimisation are:

Algorithm 1: Creeping random search (CRS)

input : $f(\cdot)$, α_1 , l_{\max} , cooling

output: \mathbf{A}_l

```

1 Generate random start  $\mathbf{A}_1$  and set  $\mathbf{A}_{\text{cur}} := \mathbf{A}_1$ ,  $I_{\text{cur}} = f(\mathbf{A}_{\text{cur}})$ ,  $j = 1$ ;
2 repeat
3   Set  $l = 1$ ;
4   repeat
5     Generate  $\mathbf{A}_l = (1 - \alpha_j)\mathbf{A}_{\text{cur}} + \alpha_j\mathbf{A}_{\text{rand}}$  and orthogonalise  $\mathbf{A}_l$ ;
6     Compute  $I_l = f(\mathbf{A}_l)$ ;
7     Update  $l = l + 1$ ;
8   until  $l > l_{\max}$  or  $I_l > I_{\text{cur}}$ ;
9   Update  $\alpha_{j+1} = \alpha_j * \text{cooling}$ ;
10  Construct the geodesic interpolation between  $\mathbf{A}_{\text{cur}}$  and  $\mathbf{A}_l$ ;
11  Update  $\mathbf{A}_{\text{cur}} = \mathbf{A}_l$  and  $j = j + 1$ ;
12 until  $\mathbf{A}_l$  is too close to  $\mathbf{A}_{\text{cur}}$  in terms of geodesic distance;
```

Algorithm 2: Simulated annealing (SA)

```

1 repeat
2   Generate  $\mathbf{A}_l = (1 - \alpha_j)\mathbf{A}_{\text{cur}} + \alpha_j\mathbf{A}_{\text{rand}}$  and orthogonalise  $\mathbf{A}_l$ ;
3   Compute  $I_l = f(\mathbf{A}_l)$ ,  $T(l) = \frac{T_0}{\log(l+1)}$  and  $P = \min \left\{ \exp \left[ -\frac{I_{\text{cur}} - I_l}{T(l)} \right], 1 \right\}$ ;
4   Draw  $U$  from a uniform distribution:  $U \sim \text{Unif}(0, 1)$ ;
5   Update  $l = l + 1$ ;
6 until  $l > l_{\max}$  or  $I_l > I_{\text{cur}}$  or  $P > U$ ;
```

Algorithm 3: Pseudo-derivative (PD)

```

1 repeat
2   Generate  $n$  random directions  $\mathbf{A}_{\text{rand}}$ ;
3   Compute  $2n$  candidate bases deviate from  $\mathbf{A}_{\text{cur}}$  by an angle of  $\delta$  while ensuring orthogonality;
4   Compute the corresponding index value for each candidate bases;
5   Determine the search direction as from  $\mathbf{A}_{\text{cur}}$  to the candidate bases with the largest index value;
6   Determine the step size via optimising the index value on the search direction over a 90 degree window;
7   Find the optimum  $\mathbf{A}_{**}$  and compute  $I_{**} = f(\mathbf{A}_{**})$ ,  $p_{\text{diff}} = (I_{**} - I_{\text{cur}})/I_{**}$ ;
8   Update  $l = l + 1$ ;
9 until  $l > l_{\max}$  or  $p_{\text{diff}} > 0.001$ ;
```
