1 Problem Setup

As input, we are given a directed (multi-)graph $G = (V_G, E_G)$, where each vertex $s \in V_G \subseteq S^2$ is a geographic position, and each edge $e = \big(s, s'\big)$ has an associated (Olivier-Ricci) curvature $R_e \in (-2, 1)$ and an associated latency $t_e \in \mathbb{R}_{\geq 0}$.

We want to return a surface in $\mathbb{R}^3$ that is the graph of a function $\rho : S^2 \to \mathbb{R}_{>0}$ whose minimal geodesics g_e between $\big(s, \rho(s)\big)$ and $\Big(s', \rho\big(s'\big)\Big)$ have length ϕ_e that is in a linear relationship with the latency.

The strategy to realize this idea is to create a mesh $M = (V_M, E_M)$ supported on a subset of S^2. We use a standard half-edge setup, so that E_M is a set of ordered pairs (edges are directed). Let P be the support. Then for each $s_i \in P$, we want to assign a $\rho_i \in \mathbb{R}_{>0}$, which in turn gives a point $v_i = (s_i, \rho_i) \in V_M$. This setup is made explicit in `mesh/sphere.py`. Note that there is an assumption that if we project V_M onto S^2, then that set contains V_G. In practice, we can just use a fine enough mesh and map the vertices in V_G to their nearest projections.

A similar setup is found in `mesh/rectangle.py`, where we use $[0, 1]^2$ instead of S^2. In general, this setup just requires that the position of any mesh vertex is controlled by a single scalar parameter.

2 Objective/Loss Functions

To enforce that the mesh approximates our desired surface, we roughly[1] define the objective functions

$$\begin{aligned}
\mathcal{L}_{\text{geodesic}}(M) &\triangleq \sum_{e \in E_G} (\text{least squares residual of edge } e)^2, \\
\mathcal{L}_{\text{curvature}}(M) &\triangleq \sum_{\substack{v \in V_M \\ e \in E_G \\ v \text{ close to } e}} \big(\kappa(v) - R_e\big)^2, \\
\mathcal{L}_{\text{smooth}}(M) &\triangleq -\rho^\intercal L_C^{\text{N}} \rho, \\
\mathcal{L}(M) &\triangleq \lambda_{\text{geodesic}} \mathcal{L}_{\text{geodesic}}(M) + \lambda_{\text{curvature}} \mathcal{L}_{\text{curvature}}(M) + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}(M),
\end{aligned}$$

where the λ's are tunable hyperparameters. The other variables will be defined in the upcoming subsections. Our goal is then to minimize $\mathcal{L}(M)$.

Note that the loss functions (particularly the geodesic and total ones) also have a dependence on the measured latencies. We omit that as a written parameter because they are treated as fixed (we are really optimizing over the manifold, not over the measured latencies).

2.1 Laplacian

Some mesh notation first. If i and j are two indices vertices for which $\big(v_i, v_j\big) \in E_M$, let $c(i, j)$ be the index such that $v_i \to v_j \to v_{c(i,j)}$ traces a triangle counterclockwise. Note that this index exists and is unique assuming we have a mesh without boundary. On a mesh with boundary, if no $c(i, j)$ exists, then the half-edge $\big(v_i, v_j\big)$ lies on the boundary.

We also write ∂M to represent the boundary of our mesh. Abusing notation, we can write things like $v_i \in \partial M$ or $\big(v_i, v_j\big) \in \partial M$.

We define the following variables:

$N_{i,j}$	Outward normal of triangle $v_i \to v_j \to v_{c(i,j)}$
$A_{i,j}$	Area of triangle $v_i \to v_j \to v_{c(i,j)}$
$D_{i,j}$	Vertex triangle areas; diagonal
$\theta_{i,j}$	Measure of $\angle v_i v_{c(i,j)} v_j$
L_C^{N}	Cotangent operator with zero-Neumann boundary condition
L_C^{D}	Cotangent operator with zero-Dirichlet boundary condition
L_C	Cotangent operator in the no-boundary case; sparse

[1]The actual definitions are scaled so that the values are comparable regardless of the choice of mesh.

2.1.1 Forward Computation

We have the following (standard) definition of the Laplace-Beltrami operator on a mesh:

$$
\begin{aligned}
N_{i,j} &= \left(v_i - v_{c(i,j)}\right) \times \left(v_j - v_{c(i,j)}\right), \\
A_{i,j} &= \frac{1}{2}\left\|N_{i,j}\right\|_2, \\
D_{i,j} &= \begin{cases} \frac{1}{3} \displaystyle\sum_{\substack{k \\ (v_i,v_k)\in E_M}} A_{i,k} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \\
\cot\left(\theta_{i,j}\right) &= \frac{\left(v_i - v_{c(i,j)}\right) \cdot \left(v_j - v_{c(i,j)}\right)}{2A_{i,j}}, \\
\left(L_C^{\mathrm{N}}\right)_{i,j} &= \begin{cases} \frac{1}{2}\cot\left(\theta_{i,j}\right) & \text{if } \left(v_i, v_j\right) \in \partial M, \\ \frac{1}{2}\cot\left(\theta_{j,i}\right) & \text{if } \left(v_j, v_i\right) \in \partial M, \\ \frac{1}{2}\left(\cot\left(\theta_{i,j}\right) + \cot\left(\theta_{j,i}\right)\right) & \text{if } \left(v_i, v_j\right), \left(v_j, v_i\right) \in E_M, \\ -\frac{1}{2}\left(\displaystyle\sum_{\substack{k \\ (v_i,v_k)\in E_M}} \cot\left(\theta_{i,k}\right) + \sum_{\substack{k \\ (v_k,v_i)\in E_M}} \cot\left(\theta_{k,i}\right)\right) & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases} \\
\left(L_C^{\mathrm{D}}\right)_{i,j} &= \begin{cases} \frac{1}{2}\left(\cot\left(\theta_{i,j}\right) + \cot\left(\theta_{j,i}\right)\right) & \text{if } \left(v_i, v_j\right) \in E_M,\ v_i \notin \partial M, \text{ and } v_j \notin \partial M, \\ -\frac{1}{2} \displaystyle\sum_{\substack{k \notin \partial M \\ (v_i,v_k)\in E_M \\ (v_k,v_i)\in E_M}} \left(\cot\left(\theta_{i,k}\right) + \cot\left(\theta_{k,i}\right)\right) & \text{if } i = j \text{ and } v_i \notin \partial M, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned}
$$

Flipping our attention to meshes without boundary, the two definitions above coincide, so we can write

$$L_C = L_C^{\text{Neumann}} = L_C^{\text{Dirichlet}}.$$

We take special note of this case as this is what is described in great detail in the original heat method paper.

2.1.2 Reverse Computation

For the ease of notation, assume that we are using the spherical setup, so $v_\ell = \rho_\ell s_\ell$.

We compute

$$
\frac{\partial v_i}{\partial \rho_\ell} = \begin{cases} s_i & \text{if } \ell = i, \\ 0 & \text{otherwise,} \end{cases}
$$

$$\frac{\partial N_{i,j}}{\partial \rho_\ell} = \begin{cases} \left(v_{c(i,j)} - v_j\right) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = i, \\ \left(v_i - v_{c(i,j)}\right) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = j, \\ \left(v_j - v_i\right) \times \frac{\partial v_\ell}{\partial \rho_\ell} & \text{if } \ell = c(i,j), \\ 0 & \text{otherwise,} \end{cases}$$

$$\frac{\partial A_{i,j}}{\partial \rho_\ell} = \frac{1}{4A_{i,j}} N_{i,j} \cdot \frac{\partial N_{i,j}}{\partial \rho_\ell},$$

$$\left(\frac{\partial D}{\partial \rho_\ell}\right)_{i,j} = \begin{cases} \frac{1}{3} \sum\limits_{\substack{k \\ (v_i, v_k) \in E_M}} \frac{\partial A_{i,k}}{\partial \rho_\ell} & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,j}\right) = \begin{cases} \dfrac{\left(v_j - v_{c(i,j)}\right) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\left(\theta_{i,j}\right) \frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = i, \\ \dfrac{\left(v_i - v_{c(i,j)}\right) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\left(\theta_{i,j}\right) \frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = j, \\ \dfrac{\left(2v_{c(i,j)} - v_i - v_j\right) \cdot \frac{\partial v_\ell}{\partial \rho_\ell} - 2\cot\left(\theta_{i,j}\right) \frac{\partial A_{i,j}}{\partial \rho_\ell}}{2A_{i,j}} & \text{if } \ell = c(i,j), \\ 0 & \text{otherwise,} \end{cases}$$

$$\left(\frac{\partial L_C^{\mathrm{N}}}{\partial \rho_\ell}\right)_{i,j} = \begin{cases} \frac{1}{2} \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,j}\right) & \text{if } \left(v_i, v_j\right) \in \partial M, \\ \frac{1}{2} \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{j,i}\right) & \text{if } \left(v_j, v_i\right) \in \partial M, \\ \frac{1}{2} \left(\frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,j}\right) + \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{j,i}\right)\right) & \text{if } \left(v_i, v_j\right), \left(v_j, v_i\right) \in E_M, \\ -\frac{1}{2} \left(\sum\limits_{\substack{k \\ (v_i, v_k) \in E_M}} \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,k}\right) + \sum\limits_{\substack{k \\ (v_k, v_i) \in E_M}} \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{k,i}\right) \right) & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

$$\left(\frac{\partial L_C^{\mathrm{D}}}{\partial \rho_\ell}\right)_{i,j} = \begin{cases} \frac{1}{2} \left(\frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,j}\right) + \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{j,i}\right)\right) & \text{if } \left(v_i, v_j\right) \in E_M,\ v_i \notin \partial M, \text{ and } v_j \notin \partial M, \\ -\frac{1}{2} \sum\limits_{\substack{k \notin \partial M \\ (v_i, v_k) \in E_M \\ (v_k, v_i) \in E_M}} \left(\frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{i,k}\right) + \frac{\partial}{\partial \rho_\ell} \cot\left(\theta_{k,i}\right)\right) & \text{if } i = j \text{ and } v_i \notin \partial M, \\ 0 & \text{otherwise.} \end{cases}$$

2.2 Geodesic Distance via the Heat Method

Here are the variables used for this part of the computation:

γ	Set of points in V_M
h	Mean half-edge length
δ^γ	Heat source (indicator on γ)
$u^{\gamma,\mathrm{N}}$	Heat flow with zero-Neumann boundary condition
$u^{\gamma,\mathrm{D}}$	Heat flow with zero-Dirichlet boundary condition
u^γ	Heat flow
$q^\gamma_{i,j}$	Intermediate value for computation
$m^\gamma_{i,j}$	Intermediate value for computation
$X^\gamma_{i,j}$	Unit vector in same direction as $\nabla u^\gamma_{i,j}$
$p^\gamma_{i,j}$	Intermediate value for computation
$\widetilde{\phi}^\gamma$	Vector of offset geodesic distances
ϕ^γ	Vector of offset geodesic distances

2.2.1 Forward Computation

Say we want to find the geodesic distances to a set of points $\gamma \subseteq V_M$. Following Crane et al's Heat Method, we use the (approximate) heat flow u^γ, where

$$\begin{aligned}
h &= \frac{1}{|E_M|} \sum_{\substack{i,j \\ (v_i,v_j)\in E_M}} \left\|v_i - v_j\right\|_2, \\
\delta^\gamma &= \begin{cases} 1 & \text{if } v_i \in \gamma, \\ 0 & \text{if } v_i \notin \gamma, \end{cases} \\
u^{\gamma,\mathrm{N}} &= \left(D - h^2 L_C^{\mathrm{N}}\right)^{-1} \delta^\gamma, \\
u^{\gamma,\mathrm{D}} &= \left(D - h^2 L_C^{\mathrm{D}}\right)^{-1} \delta^\gamma, \\
u^\gamma &= \frac{1}{2}\left(u^{\gamma,\mathrm{N}} + u^{\gamma,\mathrm{D}}\right), \\
q^\gamma_{i,j} &= u^\gamma_i \left(v_{c(i,j)} - v_j\right), \\
m^\gamma_{i,j} &= q^\gamma_{i,j} + q^\gamma_{j,c(i,j)} + q^\gamma_{c(i,j),i}, \\
\left(\nabla u^\gamma\right)_{i,j} &= N_{i,j} \times m^\gamma_{i,j}, \\
X^\gamma_{i,j} &= -\frac{\left(\nabla u^\gamma\right)_{i,j}}{\left\|\left(\nabla u^\gamma\right)_{i,j}\right\|_2}, \\
p_{i,j} &= \cot\left(\theta_{i,j}\right)\left(v_j - v_i\right), \\
\left(\nabla \cdot X^\gamma\right)_i &= \frac{1}{2} \sum_{\substack{k \\ (v_i,v_k)\in E_M}} \left(p_{i,k} - p_{c(i,k),i}\right) \cdot X^\gamma_{i,k}, \\
\widetilde{\phi}^\gamma &= \left(L_C^{\mathrm{N}}\right)^{+} \cdot \left(\nabla \cdot X^\gamma\right), \\
\phi^\gamma &= \widetilde{\phi}^\gamma - \min\left(\widetilde{\phi}^\gamma\right).
\end{aligned}$$

Here, $\left(L_C^{\mathrm{N}}\right)^{+}$ is the pseudoinverse of L_C^{N} (this is necessary as it is singular).

Note that we're being careful about which pieces have a dependence on γ, as we can reuse certain computations if we want to compute distances from multiple sources. We can get the pairwise distance matrix (that is, get rid of the γ dependence) from

$$\phi_{i,j} = \left(\phi^{\{v_j\}}\right)_i.$$

2.2.2 Reverse Computation

Note that $c\big(i, c(j,i)\big) = j$. This is helpful for reindexing some sums (in particular, the one for $\nabla \cdot X$).

We then have the following partial derivatives:

$$\begin{aligned}
\frac{\partial h}{\partial \rho_\ell} &= \frac{1}{|E_M|}\left(\sum_{\substack{k\\ (v_\ell, v_k)\in E_M}} \frac{(v_\ell - v_k)}{\|v_\ell - v_k\|_2}\cdot\frac{\partial v_\ell}{\partial \rho_\ell} + \sum_{\substack{k\\ (v_k, v_\ell)\in E_M}} \frac{(v_\ell - v_k)}{\|v_\ell - v_k\|_2}\cdot\frac{\partial v_\ell}{\partial \rho_\ell}\right),\\
\frac{\partial u^{\gamma,\mathrm{N}}}{\partial \rho_\ell} &= -\left(D - h^2 L_C^{\mathrm{N}}\right)^{-1}\left(\frac{\partial D}{\partial \rho_\ell} - 2h\frac{\partial h}{\partial \rho_\ell}L_C^{\mathrm{N}} - h^2\frac{\partial L_C^{\mathrm{N}}}{\partial \rho_\ell}\right)u^{\gamma,\mathrm{N}},\\
\frac{\partial u^{\gamma,\mathrm{D}}}{\partial \rho_\ell} &= -\left(D - h^2 L_C^{\mathrm{D}}\right)^{-1}\left(\frac{\partial D}{\partial \rho_\ell} - 2h\frac{\partial h}{\partial \rho_\ell}L_C^{\mathrm{D}} - h^2\frac{\partial L_C^{\mathrm{D}}}{\partial \rho_\ell}\right)u^{\gamma,\mathrm{D}},\\
\frac{\partial u^{\gamma}}{\partial \rho_\ell} &= \frac{1}{2}\left(\frac{\partial u^{\gamma,\mathrm{N}}}{\partial \rho_\ell} + \frac{\partial u^{\gamma,\mathrm{D}}}{\partial \rho_\ell}\right),\\
\frac{\partial q^{\gamma}_{i,j}}{\partial \rho_\ell} &= \begin{cases}
\frac{\partial u_i^\gamma}{\rho_\ell}\left(v_{c(i,j)} - v_j\right) - u_i^\gamma\frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = j,\\
\frac{\partial u_i^\gamma}{\rho_\ell}\left(v_{c(i,j)} - v_j\right) + u_i^\gamma\frac{\partial v_\ell}{\partial\rho_\ell} & \text{if } \ell = c(i,j),\\
\frac{\partial u_i^\gamma}{\rho_\ell}\left(v_{c(i,j)} - v_j\right) & \text{otherwise,}
\end{cases}\\
\frac{\partial m^{\gamma}_{i,j}}{\partial \rho_\ell} &= \frac{\partial q^{\gamma}_{i,j}}{\partial \rho_\ell} + \frac{\partial q^{\gamma}_{j,c(i,j)}}{\partial \rho_\ell} + \frac{\partial q^{\gamma}_{c(i,j),i}}{\partial \rho_\ell},\\
\frac{\partial (\nabla u^\gamma)_{i,j}}{\partial \rho_\ell} &= \frac{\partial N_{i,j}}{\partial \rho_\ell}\times m^\gamma_{i,j} + N_{i,j}\times\frac{\partial m^{\gamma}_{i,j}}{\partial \rho_\ell},\\
\frac{\partial X^{\gamma}_{i,j}}{\partial \rho_\ell} &= -\frac{1}{\left\|(\nabla u^\gamma)_{i,j}\right\|_2}\left(I - X^\gamma_{i,j}\left(X^\gamma_{i,j}\right)^{\mathsf{T}}\right)\frac{\partial (\nabla u^\gamma)_{i,j}}{\partial \rho_\ell},\\
\frac{\partial p_{i,j}}{\partial \rho} &= \begin{cases}
\left(\frac{\partial}{\partial\rho_\ell}\cot\big(\theta_{i,j}\big)\right)(v_j - v_i) - \cot\big(\theta_{i,j}\big)\frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = i,\\
\left(\frac{\partial}{\partial\rho_\ell}\cot\big(\theta_{i,j}\big)\right)(v_j - v_i) + \cot\big(\theta_{i,j}\big)\frac{\partial v_\ell}{\rho_\ell} & \text{if } \ell = j,\\
\left(\frac{\partial}{\partial\rho_\ell}\cot\big(\theta_{i,j}\big)\right)(v_j - v_i) & \text{if } \ell = c(i,j),\\
0 & \text{otherwise,}
\end{cases}\\
\frac{\partial (\nabla\cdot X^\gamma)_i}{\partial \rho_\ell} &= \frac{1}{2}\sum_{\substack{k\\ (v_i, v_k)\in E_M}}\left(\left(\frac{\partial p_{i,k}}{\partial \rho_\ell} - \frac{\partial p_{c(i,k),i}}{\partial \rho_\ell}\right)\cdot X^\gamma_{i,k} + \left(p_{i,k} - p_{c(i,k),i}\right)\cdot\frac{\partial X^\gamma_{i,k}}{\partial \rho_\ell}\right),\\
\frac{\partial \widetilde{\phi}^\gamma}{\partial \rho_\ell} &= \left(L_C^{\mathrm{N}}\right)^{+}\left(\frac{\partial(\nabla\cdot X^\gamma)}{\partial \rho_\ell} - \frac{\partial L_C^{\mathrm{N}}}{\partial \rho_\ell}\phi^\gamma\right),\\
\frac{\partial \phi^\gamma}{\partial \rho_\ell} &= \frac{\partial \widetilde{\phi}^\gamma}{\partial \rho_\ell} - \left(\frac{\partial \widetilde{\phi}^\gamma}{\partial \rho_\ell}\right)_\gamma.
\end{aligned}$$

Note that $\gamma = \arg\min(\phi)$, which is where the final subtraction comes from.

2.3 Curvature

We will define the following:

$B_\epsilon(e)$	A "fat edge" on the sphere around e
$\widetilde{\kappa^G}_i$	The discrete Gaussian curvature at v_i, scaled by vertex area
κ_i^G	The discrete Gaussian curvature at v_i
κ_i^H	The discrete mean curvature at v_i

These quantities can be computed following this tutorial.

2.3.1 Forward Computation

We have

$$\begin{aligned}
\theta_{i,j} &= \arctan\left(\frac{1}{\cot\left(\theta_{i,j}\right)}\right) \bmod \pi, \\
\widetilde{\kappa^G}_i &= 2\pi - \sum_{\substack{k \\ \left(v_i, v_k\right) \in E_M}} \theta_{k,c\left(i,k\right)}, \\
\kappa^G &= D^{-1}\widetilde{\kappa^G}.
\end{aligned}$$

2.3.2 Reverse Computation

Differentiating,

$$\begin{aligned}
\frac{\partial \theta_{i,j}}{\partial \rho_\ell} &= -\frac{\partial \cot\left(\theta_{i,j}\right)}{\partial \rho_\ell} \cdot \frac{1}{1+\cot^2\left(\theta_{i,j}\right)}, \\
\frac{\partial \widetilde{\kappa^G}_i}{\partial \rho_\ell} &= -\sum_{\substack{k \\ \left(v_i, v_k\right) \in E_M}} \frac{\partial \theta_{k,c\left(i,k\right)}}{\partial \rho_\ell}, \\
\frac{\mathrm{d}\kappa^G}{\partial \rho_\ell} &= D^{-1}\left(\frac{\mathrm{d}\widetilde{\kappa^G}}{\partial \rho_\ell} - \frac{\mathrm{d}D}{\partial \rho_\ell}\kappa^G\right).
\end{aligned}$$

2.4 Geodesic Loss

We will define the following in this section:

$\widetilde{\phi}$	Geodesic distances corresponding to edges in E_G
$\widetilde{d}$	Centered version of $\widetilde{\phi}$
d	Normalized and centered version of $\widetilde{\phi}$
β	The least squares linear estimator between d and t
$\mathcal{L}_{\text{geodesic}}(M)$	The sum of squared residuals when using β as an estimator, scaled to be unitless

In this and the following sections, we will abuse notation a bit and write things like ϕ_e to mean $\phi_{i,j}$, where $e = (i,j) \in E_G$.

2.4.1 Forward Computation

We make the following computations:

$$\begin{aligned}
\widetilde{\phi}_e &= \phi_e \text{ when } e \in E_G,\\
\widetilde{d} &= \widetilde{\phi} - \frac{1}{|E_G|}\left(\widetilde{\phi}\cdot\mathbf{1}\right)\mathbf{1},\\
d &= \frac{1}{\sqrt{\frac{1}{|E_G|}\widetilde{d}\cdot\widetilde{d}}}\widetilde{d},\\
\beta_0 &= \frac{1}{|E_G|}t\cdot\mathbf{1},\\
\beta_1 &= \frac{1}{|E_G|}t\cdot d,\\
\mathcal{L}_{\text{geodesic}}(M) &= \frac{1}{|E_G|\operatorname{Var}(t)}\left\|t-(\beta_0\mathbf{1}+\beta_1 d)\right\|_2^2.
\end{aligned}$$

2.4.2 Reverse Computation

The partials of the above quantities are as follows:

$$\begin{aligned}
\frac{\partial\widetilde{\phi}_e}{\partial\rho_\ell} &= \frac{\partial\phi_e}{\partial\rho_\ell},\\
\frac{\partial\widetilde{d}}{\partial\rho_\ell} &= \frac{\partial\widetilde{\phi}}{\partial\rho_\ell} - \frac{1}{|E_G|}\left(\frac{\partial\widetilde{\phi}}{\partial\rho_\ell}\cdot\mathbf{1}\right)\mathbf{1},\\
\frac{\partial d}{\partial\rho_\ell} &= \frac{1}{\sqrt{\frac{1}{|E_G|}\widetilde{d}\cdot\widetilde{d}}}\left(\frac{\partial\widetilde{d}}{\partial\rho_\ell} - \frac{1}{|E_G|}\left(d\cdot\frac{\partial\widetilde{d}}{\partial\rho_\ell}\right)d\right),\\
\frac{\partial\beta_0}{\partial\rho_\ell} &= 0,\\
\frac{\partial\beta_1}{\partial\rho_\ell} &= \frac{1}{|E_G|}t\cdot\frac{\partial d}{\partial\rho_\ell},\\
\frac{\partial\big(\mathcal{L}_{\text{geodesic}}(M)\big)}{\partial\rho_\ell} &= -\frac{2}{|E_G|\operatorname{Var}(t)}\big(t-(\beta_0\mathbf{1}+\beta_1 d)\big)\cdot\left(\frac{\partial\beta_1}{\partial\rho_\ell}d+\beta_1\frac{\partial d}{\partial\rho_\ell}\right).
\end{aligned}$$

2.5 Curvature Loss

We will define the following:

$\mathcal{L}_{\text{curvature}}(M)$	Sum of squares of the differences between vertices actual and desired curvatures

2.5.1 Approximating "Fat Edges"[2] on a Sphere

For this subsection, we will use notation that has been used elsewhere to mean other things. We do this for readability reasons.

Consider u, v, and r all on the unit sphere. Assume that u and v are not antipodal. Our goal is to determine whether r is within a (geodesic) distance of ϵ to the shortest arc between u and v. If this is the case, we write $r \in B_\epsilon\big((u,v)\big)$.

We can find the point $\operatorname{proj}(r)$ nearest to r on the great circle passing through u and v by projecting r onto the plane spanned by u and v and then normalizing the result. The strategy for this is to just use

[2] This name should really be changed...

Graham-Schmidt to get an orthonormal basis $\{v, w\}$ of the plane. Once we have $\operatorname{proj}(r)$, we can find the distance from r to the great circle by taking advantage of the dot product.

$$
\begin{aligned}
\left\|u-(u\cdot v)v\right\|_2^2 &= \|u\|_2^2 + (u\cdot v)^2\|v\|_2^2 - 2(u\cdot v)^2 \\
&= 1 + (u\cdot v)^2 - 2(u\cdot v)^2 \\
&= 1 - (u\cdot v)^2, \\
w &\triangleq \frac{u-(u\cdot v)v}{\left\|u-(u\cdot v)v\right\|_2}, \\
\operatorname{proj}(r) &= \frac{(r\cdot v)v + (r\cdot w)w}{\left\|(r\cdot v)v + (r\cdot w)w\right\|_2}, \\
c &\triangleq \left\|(r\cdot v)v + (r\cdot w)w\right\|_2 \\
&= \sqrt{(r\cdot v)^2 + (r\cdot w)^2} \\
&= \sqrt{(r\cdot v)^2 + \left(r\cdot \frac{u-(u\cdot v)v}{\left\|u-(u\cdot v)v\right\|_2}\right)^2} \\
&= \sqrt{(r\cdot v)^2 + \frac{\big(r\cdot u - (r\cdot v)(u\cdot v)\big)^2}{1-(u\cdot v)^2}} \\
&= \sqrt{\frac{(r\cdot v)^2 - (r\cdot v)^2(u\cdot v)^2}{1-(u\cdot v)^2} + \frac{(r\cdot u)^2 + (r\cdot v)^2(u\cdot v)^2 - 2(r\cdot u)(r\cdot v)(u\cdot v)}{1-(u\cdot v)^2}} \\
&= \sqrt{\frac{(r\cdot u)^2 + (r\cdot v)^2 - 2(r\cdot u)(r\cdot v)(u\cdot v)}{1-(u\cdot v)^2}} \\
\cos(\theta) &= r\cdot \operatorname{proj}(r) \\
&= \frac{(r\cdot v)^2 + (r\cdot w)^2}{\left\|(r\cdot v)v + (r\cdot w)w\right\|_2} \\
&= \left\|(r\cdot v)v + (r\cdot w)w\right\|_2 \qquad \text{(by orthonormality)} \\
&= c \\
\theta &= \arccos(c).
\end{aligned}
$$

Our real question is whether r is "close" (within distance ϵ) to the shortest path from u to v on the unit sphere. There are two cases to consider. The first is that r is very close to u or v. This can be determined by checking $\arccos(r\cdot u) < \epsilon$ or $\arccos(r\cdot v) < \epsilon$ (if either of these is the case, then r is close).

The second case is that r is close to some point that isn't one of the endpoints (this has some overlap with the previous case, but the previous case is easier to check, so we check it first). The trick here is to use the long computation seen above. r is close to the great circle passing through u and v when

$$\arccos(c) < \epsilon.$$

Being a bit more refined, we actually want the angles between $\operatorname{proj}(r)$ and each of u and v to be at most the angle between u and v. In other words,

$$
\begin{aligned}
\max\Big(\arccos\big(\operatorname{proj}(r)\cdot u\big), \arccos\big(\operatorname{proj}(r)\cdot v\big)\Big) &\leq \arccos(u\cdot v) \\
\min\big(\operatorname{proj}(r)\cdot u, \operatorname{proj}(r)\cdot v\big) &\geq u\cdot v.
\end{aligned}
$$

For the left hand side, we can compute

$$\operatorname{proj}(r)\cdot v = \frac{(r\cdot v)v + (r\cdot w)w}{\left\|(r\cdot v)v + (r\cdot w)w\right\|_2}\cdot v$$

$$
\begin{aligned}
&= \frac{r \cdot v}{\left\|(r \cdot v)v + (r \cdot w)w\right\|_2} \\
&= \frac{r \cdot v}{c}. \\
\operatorname{proj}(r) \cdot u &= \frac{r \cdot u}{c}. && \text{(by symmetry)}
\end{aligned}
$$

Putting this together, $r \in B_\epsilon\big((u, v)\big)$ if and only if one of the following is true:

- $r \cdot u > \cos(\epsilon)$;
- $r \cdot v > \cos(\epsilon)$;
- $c > \cos(\epsilon)$ and $\min(r \cdot u, r \cdot v) \geq c\,(u \cdot v)$.

2.5.2 Forward Computation

We have

$$\mathcal{L}_{\text{curvature}}(M) \propto \sum_{e \in E_G} \sum_{\substack{k \\ v_k \in B_\epsilon(e)}} \left(R_e - \kappa_k^G\right)^2.$$

Here, we scale $\mathcal{L}_{\text{curvature}}(M)$ proportional to the number of vertices in the "fat edges," counted with multiplicity.

2.5.3 Reverse Computation

Differentiating,

$$\frac{\partial\big(\mathcal{L}_{\text{curvature}}(M)\big)}{\partial \rho_\ell} \propto \sum_{e \in E_G} \sum_{\substack{k \\ v_k \in B_\epsilon(e)}} -2\left(R_e - \kappa_k^G\right) \frac{\partial \kappa_k^G}{\partial \rho_\ell}.$$

2.6 Smoothness Loss

We will define the following:

$\mathcal{L}_{\text{smooth}}(M)$	A discrete approximation to the Dirichlet energy of κ on M

2.6.1 Forward Computation

Following this tutorial, we have

$$\mathcal{L}_{\text{smooth}}(M) \propto -\kappa^\intercal L_C^{\mathrm{N}} \kappa.$$

In terms of scaling, we divide by the surface area of the mesh when $\rho = 0$ (that is, the area of a flat plane, a sphere, or similar).

2.6.2 Reverse Computation

Differentiating,

$$\frac{\partial\big(\mathcal{L}_{\text{smooth}}(M)\big)}{\partial \rho_\ell} \propto -\left(\frac{\partial \kappa}{\rho_\ell}\right)^\intercal L_C^{\mathrm{N}} \kappa - \kappa^\intercal \frac{\partial L_C^{\mathrm{N}}}{\partial \rho_\ell} \kappa - \kappa^\intercal L_C^{\mathrm{N}} \frac{\partial \kappa}{\rho_\ell}.$$