Over-Squashing in Riemannian Graph Neural Networks

Julia Balla
Massachusetts Institute of Technology
jballa@mit.edu

Abstract

Most graph neural networks (GNNs) are prone to the phenomenon of over-squashing in which node features become insensitive to information from distant nodes in the graph. Recent works have shown that the topology of the graph has the greatest impact on over-squashing, suggesting graph rewiring approaches as a suitable solution. In this work, we explore whether over-squashing can be mitigated through the embedding space of the GNN. In particular, we consider the generalization of Hyperbolic GNNs (HGNNs) to Riemannian manifolds of variable curvature in which the geometry of the embedding space is faithful to the graph's topology. We derive bounds on the sensitivity of the node features in these Riemannian GNNs as the number of layers increases, which yield promising theoretical and empirical results for alleviating over-squashing in graphs with negative curvature.

1 Introduction

Graph Neural Networks (GNNs) have emerged as a powerful tool for modeling relational systems and learning on graph-structured data [1–4]. Most GNN architectures rely on the *message-passing* paradigm in which information is propagated along the edges of the graph, resulting in a class of Message Passing Neural Networks (MPNNs). However, due to an exponentially growing computational tree, the compression of a quickly increasing amount of information into a fixed-size vector leads to informational *over-squashing* [5]. This phenomenon poses a significant challenge on long-range tasks with a large problem radius since it obstructs the diffusion of information from distant nodes.

The over-squashing problem has been analyzed through various lenses such as graph curvature [6], information theory [7], and effective resistance [8], each suggesting a corresponding approach to mitigate the issue by rewiring the graph. Along with several other works, this line of reasoning has resulted in a "zoo" of proposed graph rewiring techniques for over-squashing [9–12]. Recent work has unified the spatial and spectral techniques under a common framework and justified their efficacy by demonstrating that graph topology plays the biggest role in alleviating over-squashing as opposed to MPNN properties such as width or depth [13] .

One potential drawback of many spatial graph rewiring techniques is the distortion of structural information that may be relevant to the learning task. Instead of altering the graph topology, we thus consider augmentations to the MPNN architecture that would make it topology-aware. Specifically, we explore the effects of changing the embedding space of the GNN. The hypothesis behind our approach is that by embedding the negatively curved sections of the graph in hyperbolic space, there would be less information lost at each layer due to the increased representational capacity. However, hyperbolic space is a poor inductive bias for graphs with significant positive curvature, where spherical space would be more suitable. Therefore, we consider a GNN that embeds graphs in Riemannian manifolds of variable curvature.

We study the over-squashing phenomenon in one such model by generalizing the Hyperbolic GNN (HGNN) architecture [14] to Riemannian GNNs (RGNNs). Assuming that there exists a Riemannian manifold where the geometry matches that of the input graph, RGNNs are in principle able to embed the graph in this manifold. While the RGNN architecture is not immediately computationally tractable in its most general form, it provides a means to derive a best-case theoretical result on over-squashing.

J. Balla, Over-Squashing in Riemannian Graph Neural Networks (Extended Abstract). Presented at the Second Learning on Graphs Conference (LoG 2023), Virtual Event, November 27–30, 2023.

We derive a bound on the Jacobian of the node features in a RGNN and show that it relies on the global curvature properties of the embedding space. Based on this bound, we heuristically and empirically demonstrate that our model addresses cases where the graph's curvature is predominantly negative everywhere (e.g. tree-like graphs). We also identify pathological cases where our model may fail on manifolds with both positive and negative curvature. Finally, we propose concrete next steps to complete our theoretical analysis that would justify step (2) in the argument above and motivate the development of tractable methods that approximate general Riemannian GNNs.

2 Riemannian GNNs

For a primer on the Riemannian geometry notions used throughout the following sections, we refer the reader to Appendix A. We define GNNs that embed node representations in a Riemannian space that is faithful to the input graph's topology. Crucially, we assume that we are given an "optimal" Riemannian manifold $(\mathcal{M}, g)$ and that the GNN has access to the distance, exponential map, and logarithmic map functions as differentiable operations. While finding an optimal Riemannian manifold of variable curvature is challenging in practice, there exist methods for its approximation [15–19]. For the purposes of our analysis, we assume this approximation of $(\mathcal{M}, g)$ is exact.

To generalize the Euclidean GNNs to Riemannian manifolds, Liu, Nickel, and Kiela [14] build upon Hyperbolic Neural Networks (HNNs) [20]. Since there is no well-defined notion of vector space structure in Riemannian space, the main idea is to leverage the exponential and logarithmic maps to perform node feature transformation and neighborhood aggregation functions as Euclidean operations in the tangent space $\mathcal{T}_{\mathbf{p}}\mathcal{M}$ of some chosen point $\mathbf{p} \in \mathcal{M}$. In particular, the node update rule is given by

$$\mathbf{x}_i^{(\ell+1)} = \sigma\left(\exp_{\mathbf{p}}\left(\sum_{j\in\mathcal{N}(i)} \tilde{\mathbf{A}}_{ij}\mathbf{W}^{(\ell)}\log_{\mathbf{p}}\left(\mathbf{x}_j^{(\ell)}\right)\right)\right) \tag{1}$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{I})\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix with self-loops, $\mathcal{N}(i)$ is the set of in-neighbors of node i, $\mathbf{W}^{(\ell)}$ is the matrix of trainable parameters at layer ℓ, and σ is a chosen activation function. Note that in the case of the Euclidean manifold, operating in the tangent space of the origin by setting $\mathbf{p} = \mathbf{o}$ recovers a vanilla GNN. Since hyperbolic manifolds fall under the class of manifolds that have a pole $\mathbf{o}$ (i.e., $\exp_{\mathbf{o}} : \mathcal{T}_{\mathbf{o}}\mathcal{M} \to \mathcal{M}$ is a diffeomorphism [21]), Liu, Nickel, and Kiela [14] choose $\mathbf{p} = \mathbf{o}$ across all nodes and layers for HGNNs. However, general Riemannian manifolds do not have a pole, so we let $\mathbf{p} = p(i, \ell) \in \mathcal{M}$ for an arbitrary function p that depends on the current node and/or the layer ℓ. We leave the selection of an optimal function p as future work. We also ensure that the exponential and logarithmic maps are differentiable by restricting $\left\|\sum_{j\in\mathcal{N}(i)} \tilde{\mathbf{A}}_{ij}\mathbf{W}^{(\ell)}\log_{\mathbf{p}}\left(\mathbf{x}_j^{(\ell)}\right)\right\|_2$ to fall within the injectivity radius of $\mathbf{p}$.

3 Sensitivity Analysis

Following the methodology in [13], we assess the over-squashing effect in RGNNs by deriving a bound on the norm of the Jacobian of node features after ℓ layers. Since this involves bounding the differentials of the exponential and logarithmic maps, we first derive the following lemma.

Lemma 1. *Consider a RGNN as in equation (1) with Riemannian manifold $(\mathcal{M}, g)$ with bounded sectional curvature $k \le \kappa_{\mathbf{p}}(\mathbf{u}, \mathbf{v}) \le K$ for all $\mathbf{p} \in \mathcal{M}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{T}_{\mathbf{p}}\mathcal{M}$. Let Df denote the differential of a map f. Then for $\exp_{\mathbf{p}}$ and $\log_{\mathbf{p}}$ in (1) and $i \in V$ we have*

$$\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\|_2 \le \begin{cases} \frac{\sinh\left(\sqrt{-k}r_{i,\exp}\right)}{\sqrt{-k}r_{i,\exp}} & k < K \le 0 \\ \frac{\sinh\left(\sqrt{-k}r_{i,\exp}\right)\sin\left(\sqrt{K}r_{i,\log}\right)}{\sqrt{-kK}r_{i,\exp}r_{j,\log}} & k < 0 < K \\ \frac{\sin\left(\sqrt{K}r_{i,\log}\right)}{\sqrt{K}r_{i,\log}} & 0 \le k < K \\ 1 & k = K = 0 \end{cases} =: \beta_i(k, K)$$

where $r_{i,\exp} = \sup_{\ell}\left\|\sum_{z\in\mathcal{N}(i)} \tilde{\mathbf{A}}_{iz}\mathbf{W}^{(\ell)}\log_{\mathbf{p}}\left(\mathbf{x}_z^{(\ell)}\right)\right\|_2$ denotes the maximum radius around $\mathbf{p}$ for the exponential map and $r_{i,\log} = \sup_{z,\ell}\|\mathbf{x}_z^{(\ell)}\|_g$ is the maximum radius for the logarithmic map.

The proof for the above lemma relies on a well-known sectional curvature comparison result in differential geometry and can be found in Appendix C. We use this lemma to derive a bound on the sensitivity of node features.

Theorem 1. *Under the same assumptions as in Lemma 1, if c_σ is the Lipschitz constant of the nonlinearity σ and $w \geq \left\|\mathbf{W}^{(l)}\right\|_2$ is an upper bound on the spectral norm of all weight matrices, then for $i, j \in V$*

$$\left\|\frac{\partial \mathbf{x}_i^{(\ell)}}{\partial \mathbf{x}_j^{(0)}}\right\|_2 \leq c_\sigma^\ell w^\ell \beta_i(k, K)^\ell \left(\tilde{\mathbf{A}}^\ell\right)_{ij}$$

where $\beta_i(k, K)$ is a bound on the sensitivity of the exponential and logarithmic maps as defined in Lemma 1.

The proof uses induction over the number of layers ℓ and is provided in Appendix D. Note that this bound has the same form as in [13] for classical GNNs, and in fact is equivalent for Euclidean space (i.e., $k = K = 0$). To show that the RGNN is able to compensate for the information bottlenecks arising from taking powers of the adjacency matrix, **it remains to demonstrate that the growth (decay) of $\beta_i(k, K)^\ell$ is able to mitigate the decay (growth) of $(\tilde{\mathbf{A}}^\ell)_{ij}$ as ℓ increases.** In Appendix B, we demonstrate that this property holds for the pathological example of negative curvature mentioned in [6]. While a formal analysis of the variable curvature case is left as future work, we provide a heuristic argument based on the magnitude of k and K.

Heuristic Argument. Assume that $|r_{i,\exp}|$ and $|r_{i,\log}|$ do not grow very small or large as ℓ increases. If $k < 0$ and $|k| << |K|$, $\beta_i(k, K)$ is dominated by the term $\frac{\sinh\left(\sqrt{-k} r_{i,\exp}\right)}{\sqrt{-k} r_{i,\exp}}$ which increases as k grows more negative. Therefore, $\beta_i(k, K)^\ell$ grows large as ℓ increases and thus helps to alleviate over-squashing. On the other hand, if $K > 0$ and $|K| >> |k|$, $\beta_i(k, K)$ is dominated by the term $\frac{\sin\left(\sqrt{k} r_{i,\log}\right)}{\sqrt{k} r_{i,\log}}$ which decreases (albeit non-monotonically) as k grows more positive. Then $\beta_i(k, K)^\ell$ grows small as ℓ increases and instead hinders the flow of information from j to i. This behavior is not problematic since graphs with positive curvature (corresponding to cycles) would have already exchanged overlapping information in the earlier layers. However, an issue may arise in the case when $k < 0 < K$ and $|k| << |K|$ for which $\beta_i(k, K)^\ell$ grows small despite the existence of very negatively curved sections of the graph. □

This argument highlights a limitation of the result in Theorem 1 in that the bound only depends on global sectional curvature bounds k and K. Therefore, $\beta_i(k, K)$ does not target the sensitivity of specific node pairs induced by $(\tilde{\mathbf{A}}^\ell)_{ij}$. Note that if we let $\mathbf{p} = p(i, \ell, \mathbf{x}_i^{(\ell)}) \in \mathcal{M}$ be a function of the current node feature, the neighboring feature aggregation would intuitively depend on the local curvature at $\mathbf{x}_i^{(\ell)} \in \mathcal{M}$. However, this would significantly increase the complexity of the Riemannian GNN model and hence the Jacobian sensitivity derivation.

4 Empirical Results

Given that the special case of Hyperbolic GNNs is well-defined and computationally tractable, we compare the empirical sensitivity of node features in Hyperbolic Graph Convolutional Networks (HGCNs) [22] to Euclidean GCNs. We use the link prediction benchmark datasets (as well as the model hyperparameters) provided in [22]: citation networks (Cora [23] and PubMed [24]), disease propogation trees (Disease), and flight networks (Airport). The Gromov δ-hyperbolicity value of each dataset is reported in Figure 1, where lower δ is more hyperbolic. Since over-squashing is more severe for deeper GNNs, we evaluate GCNs and HGCNs (specifically the Poincaré model) of depth 6. We then consider 100 randomly sampled pairs of nodes that are distance 6 apart and take the average of the norm of their Jacobians, $\frac{1}{100}\sum_{(i,j)}\left\|\frac{\partial \mathbf{x}_i^{(6)}}{\partial \mathbf{x}_j^{(0)}}\right\|_2$. As shown in Figure 1, for three of the four datasets, both the average and maximum sensitivity in the sample are greater in HGCNs than in GCNs at each epoch. For PubMed, while the average sensitivities are roughly equal, the maximum is still always greater for HGCNs, which is consistent with our upper bound in Theorem 1. The results hold even for Cora, which has a higher hyperbolicity value. This suggests that hyperbolic embeddings may be sufficient for alleviating over-squashing even in non-hyperbolic graphs, as the

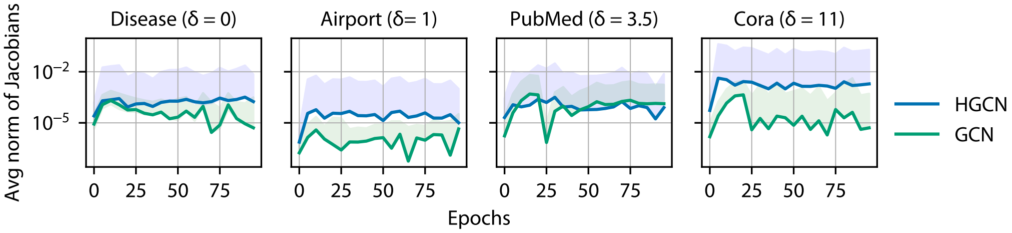

Figure 1: Sensitivity of node representations at layer 6 with respect to the input node features. The solid line denotes the average norm of the Jacobians for a random sample of 100 node pairs that are 6 hops apart. The shaded regions indicate the intervals between the minimum and maximum norm values (where the minimums tend to be very close to the average). We also include the hyperbolicity values δ of each dataset provided in [22].

distortion of positively curved regions could be compensated for by the increased sensitivity between node pairs in those regions.

We limit our empirical analysis to the special case of hyperbolic manifolds since the implementation of Riemannian GNNs as defined in (1) is not immediately feasible. First of all, it is not obvious how the reference point $\mathbf{p}$ should be defined at any given node. Moreover, our analysis assumes that we are given an optimal manifold in which the GNN should embed the graph. As described in Appendix A.4, it is not trivial to obtain the exact manifold for heterogeneous embedding spaces. However, there exist several methods for approximating these manifolds [15–19], many of which have desirable properties such as well-defined origin points for $\mathbf{p}$. We leave an empirical study of over-squashing in RGNNs built on these approximations as future work.

5 Discussion

We derive a bound on the Jacobian of node features in a Riemannian GNN. The bound contains a global curvature-dependent term $\beta_i(k, K)$ that grows exponentially with the number of layers ℓ when the embedding space has a minimum sectional curvature which is very negative and decays exponentially when the space has very positive maximum curvature. Since information bottlenecks have been linked to negative curvature on graphs, the exponential growth when $k < 0$ is a promising result for mitigating over-squashing.

Despite the heuristic argument provided in section 3 and promising empirical results for Hyperbolic GNNs in section 4, we do not formally prove that $\beta_i(k, K)$ compensates for the exponential decay of $(\tilde{\mathbf{A}}^\ell)_{ij}$ as ℓ increases without hindering overall model performance. One potential approach to deriving the relationship between the two terms could involve connecting the $\beta_i(k, K)$ term to edge-based Ricci curvature and utilizing the results in [6]. Using the intuition that the Ricci curvature can be considered as an "average" over sectional curvatures, it may be possible to define a notion of sectional curvature on a graph (e.g. the one proposed by Gu et al. [16]) such that the Balanced Forman curvature in [6] is an average of curvatures assigned to triangles of nodes. This connection may allow one to quantify how $(\tilde{\mathbf{A}}^\ell)_{ij}$ is affected by both local and global sectional curvature. Additionally, due to the Riemannian GNN's dependence on global curvature properties, the model may end up in a pathological scenario when the decay in sensitivity from maximum positive curvature outweighs the growth from the minimum negative curvature. This may call for the introduction of local curvature information into the architecture such that the neighbor aggregation at node i explicitly depends on the curvature near i. It may also be possible to localize the sensitivity bounds by constraining the manifold to have *locally* bounded sectional curvature everywhere.

Finally, while the Riemannian GNN is useful for the theoretical over-squashing analysis, implementing the proposed architecture comes with several challenges. It would be exciting to see the development of models that can more closely approximate Riemannian GNNs while maintaining tractability. For instance, it may be possible to apply the deep Riemannian manifold learning in [25] such that the optimal manifold $(\mathcal{M}, g)$ is parameterized as a neural network itself. We hope that the insights gained from our theoretical results will inspire future work in the development of practical architectures that leverage these findings.

References

[1] Franco Scarselli et al. "The Graph Neural Network Model". In: *IEEE Transactions on Neural Networks* 20.1 (2008), pp. 61–80. 1

[2] Will Hamilton, Zhitao Ying, and Jure Leskovec. "Inductive Representation Learning on Large Graphs". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. 1

[3] Thomas N. Kipf and Max Welling. "Semi-Supervised Classification with Graph Convolutional Networks". In: *International Conference on Learning Representations*. 2017. 1

[4] Justin Gilmer et al. "Neural Message Passing for Quantum Chemistry". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1263–1272. 1

[5] Uri Alon and Eran Yahav. "On the Bottleneck of Graph Neural Networks and its Practical Implications". In: *International Conference on Learning Representations*. 2021. 1

[6] Jake Topping et al. "Understanding over-squashing and bottlenecks on graphs via curvature". In: *International Conference on Learning Representations*. 2022. 1, 3, 4, 7

[7] Pradeep Kr. Banerjee et al. "Oversquashing in GNNs through the lens of information contraction and graph expansion". en. In: *2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. Monticello, IL, USA: IEEE, Sept. 2022, pp. 1–8. 1

[8] Mitchell Black et al. "Understanding Oversquashing in GNNs through the Lens of Effective Resistance". In: *Proceedings of the 40th International Conference on Machine Learning*. ICML'23. Honolulu, Hawaii, USA: JMLR.org, 2023. 1

[9] Ralph Abboud, Radoslav Dimitrov, and Ismail Ilkan Ceylan. "Shortest Path Networks for Graph Property Prediction". In: *The First Learning on Graphs Conference*. 2022. 1

[10] Adrián Arnaiz-Rodríguez et al. "DiffWire: Inductive Graph Rewiring via the Lovász Bound". In: *The First Learning on Graphs Conference*. 2022. 1

[11] Andreea Deac, Marc Lackenby, and Petar Veličković. "Expander Graph Propagation". In: *The First Learning on Graphs Conference*. 2022. 1

[12] Kedar Karhadkar, Pradeep Kr. Banerjee, and Guido Montufar. "FoSR: First-order spectral rewiring for addressing oversquashing in GNNs". In: *The Eleventh International Conference on Learning Representations*. 2023. 1

[13] Francesco Di Giovanni et al. *On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology*. arXiv:2302.02941 [cs, stat]. Feb. 2023. 1–3

[14] Qi Liu, Maximilian Nickel, and Douwe Kiela. "Hyperbolic Graph Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. 1, 2

[15] Bo Xiong et al. "Pseudo-Riemannian Graph Convolutional Networks". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo et al. Vol. 35. Curran Associates, Inc., 2022, pp. 3488–3501. 2, 4

[16] Albert Gu et al. "Learning Mixed-Curvature Representations in Product Spaces". In: *International Conference on Learning Representations*. 2019. 2, 4, 7

[17] Francesco Di Giovanni, Giulia Luise, and Michael M. Bronstein. "Heterogeneous manifolds for curvature-aware graph embedding". In: *ICLR 2022 Workshop on Geometrical and Topological Representation Learning*. 2022. 2, 4, 7

[18] Calin Cruceru, Gary Bécigneul, and Octavian-Eugen Ganea. "Computationally Tractable Riemannian Manifolds for Graph Embeddings". In: *AAAI Conference on Artificial Intelligence*. 2020. 2, 4, 7

[19] Federico Lopez et al. "Symmetric Spaces for Graph Embeddings: A Finsler-Riemannian Approach". en. In: *Proceedings of the 38th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2021, pp. 7090–7101. 2, 4, 7

[20] Octavian Ganea, Gary Becigneul, and Thomas Hofmann. "Hyperbolic Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018. 2

[21] Mitsuhiro Itoh. "Some Geometrical Aspects of Riemannian Manifolds With a Pole". In: *Tsukuba Journal of Mathematics* 4.2 (1980), pp. 291–301. 2

[22] Ines Chami et al. "Hyperbolic Graph Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. 3, 4

[23] Prithviraj Sen et al. “Collective Classification in Network Data”. In: *AI Magazine* 29.3 (2008), p. 93. 3

[24] Galileo Namata et al. “Query-driven Active Surveying for Collective Classification”. In: *Workshop on Mining and Learning with Graphs (MLG)*. 2012. 3

[25] Aaron Lou, Maximilian Nickel, and Brandon Amos. “Deep Riemannian Manifold Learning”. en. In: *Differential Geometry for Machine Learning Workshop at NeurIPS* (2020). 4, 7

[26] Peter Petersen. *Riemannian Geometry*. Vol. 171. Graduate Texts in Mathematics. Cham: Springer International Publishing, 2016. 6, 7

[27] Robin Forman. “Bochner’s Method for Cell Complexes and Combinatorial Ricci Curvature”. In: *Discrete and Computational Geometry* 29.3 (Feb. 2003), pp. 323–374. 7

[28] Yann Ollivier. “Ricci curvature of metric spaces”. In: *Comptes Rendus Mathematique* 345.11 (Dec. 2007), pp. 643–646. 7

[29] Yann Ollivier. “Ricci curvature of Markov chains on metric spaces”. In: *Journal of Functional Analysis* 256 (2007), pp. 810–864. 7

[30] Richard C. Wilson et al. “Spherical and Hyperbolic Embeddings of Data”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.11 (2014), pp. 2255–2269. 7

[31] Maximillian Nickel and Douwe Kiela. “Poincaré Embeddings for Learning Hierarchical Representations”. In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc., 2017. 7

[32] Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. “Neural Embeddings of Graphs in Hyperbolic Space”. In: *13th international workshop on mining and learning from graphs held in conjunction with KDD* (2017). 7

[33] Weiyang Liu et al. “SphereFace: Deep Hypersphere Embedding for Face Recognition”. en. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, July 2017, pp. 6738–6746. ISBN: 978-1-5386-0457-1. 7

[34] Frederic Sala et al. “Representation Tradeoffs for Hyperbolic Embeddings”. en. In: *Proceedings of the 35th International Conference on Machine Learning*. ISSN: 2640-3498. PMLR, July 2018, pp. 4460–4469. 7

A Riemannian Geometry

We first introduce some preliminary notation and concepts in Riemannian geometry. We refer the reader [26] for a more detailed discussion of these concepts.

A Riemannian manifold $(\mathcal{M}, g)$ is a smooth manifold equipped with a *Riemannian metric* $g_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \times \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathbb{R}$ where $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ is the tangent space at the point $\mathbf{x} \in \mathcal{M}$. The Riemannian metric is a local inner product that varies smoothly with $\mathbf{x}$ and allows us to define the geometric properties of a space such as length, angle, and area. For instance, g induces a norm $\|\mathbf{v}\|_g = \sqrt{g_{\mathbf{x}}(\mathbf{v}, \mathbf{v})}$ for any $v \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$.

A.1 Geodesics

The Riemannian metric also gives rise to a notion of distance. For a curve $\gamma : [0, T] \to \mathcal{M}$, the length of γ is given by $L(\gamma) = \int_0^T \|\gamma'(t)\|_g dt$. Thus, for two points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$, the distance is defined as $d_g(\mathbf{x}, \mathbf{y}) = \inf L(\gamma)$ where γ is any curve such that $\gamma(0) = \mathbf{x}$ and $\gamma(T) = \mathbf{y}$. A *geodesic* is a curve that minimizes this length.

A.2 Exponential and Logarithmic Map

For each point $\mathbf{x} \in \mathcal{M}$ and velocity vector $\mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$, there exists a unique geodesic $\gamma : [0, 1] \to \mathcal{M}$ where $\gamma(0) = \mathbf{x}$ and $\gamma'(0) = \mathbf{v}$. The *exponential map* $\exp_{\mathbf{x}} : \mathcal{T}_{\mathbf{x}}\mathcal{M} \to \mathcal{M}$ is defined as $\exp_{\mathbf{x}}(\mathbf{v}) = \gamma(1)$. Its local inverse is called the *logarithm map*, $\log_{\mathbf{x}}(\mathbf{v})$. Note that the distance between two points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ can be represented as $d_g(\mathbf{x}, \mathbf{y}) = \| \log_{\mathbf{x}}(\mathbf{y})\|_g$.

Manifolds where the exponential map is defined on the whole tangent space $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ are called *geodesically complete*. However, geodesic completeness does not guarantee that the exponential map is a global diffeomorphism (i.e. a differentiable bijective map with a differentiable inverse). The radius of the largest ball about the origin in $\mathcal{T}_{\mathbf{x}}\mathcal{M}$ that can be mapped diffeomorphically via the exponential map is called the *injectivity radius* of $\mathcal{M}$ at $\mathbf{x}$.

A.3 Curvature

For each point $\mathbf{x} \in \mathcal{M}$ and pair of linearly independent tangent vectors $\mathbf{u}, \mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$, the *sectional curvature* $\kappa_{\mathbf{x}}(\mathbf{u}, \mathbf{v})$ at $\mathbf{x}$ is defined as the *Gaussian curvature* of the two-dimensional surface obtained by exponentiating a plane spanned by $\mathbf{u}$ and $\mathbf{v}$ at $\mathbf{x}$. The Gaussian curvature of a surface is given by the product of the principal curvatures. Riemannian manifolds of constant sectional curvature κ are called *space forms*, the most common examples being spherical space ($\kappa > 0$), Euclidean space ($\kappa = 0$), and hyperbolic space ($\kappa < 0$). Another form of curvature on a Riemannian manifold is *Ricci curvature*, which is a symmetric bilinear form determining the geodesic dispersion at nearby points. The Ricci curvature of a tangent vector $\mathbf{v}$ at $\mathbf{p}$ is the average of the sectional curvature over all tangent planes containing $\mathbf{v}$.

Several works have also introduced discrete notions of sectional and Ricci curvature on graphs. Gu et al. [16] introduced a discrete notion of sectional curvature for learning product manifolds of mixed curvatures for graph embeddings. Forman [27] and Ollivier [28, 29] proposed edge-based curvature that could recover certain properties of the Ricci curvature on manifolds. Topping et al. [6] used a novel formulation of Ricci curvature to show that over-squashing in GNNs is related to the existence of edges with high negative curvature.

A.4 Riemannian Manifolds for Graph Embeddings

There has been a surge in the development of algorithms that represent graphs as sets of node embeddings in hyperbolic and spherical space due to their favorable geometric inductive biases [30–34]. These space forms are well defined and offer closed-form expressions for geometric operations such as the exponential and logarithmic map, making them suitable for optimization in these spaces.

However, space forms individually may not capture all of the geometric properties of a given graph. On the other hand, heterogeneous manifolds of variable curvature lack computational tractability. Several works have instead embedded graphs in manifolds of mixed curvature by taking Cartesian products of homogenous model spaces [16], adding heterogeneous dimensions to homogenous spaces [17], or limiting the embedding space to certain classes of manifolds [18, 19]. An exciting direction for learnable Riemannian manifolds has been proposed by Lou, Nickel, and Amos [25], where the metric is parametrized by a deep neural network.

B Example: Sensitivity for a Binary Tree in Hyperbolic Space

Suppose that nodes i and j are distance $\ell + 1$ apart and that the receptive field of node i is a binary tree in a RGNN given a manifold with constant negative sectional curvature $k < 0$ (i.e. a Hyperbolic GNN). Then $(\tilde{\mathbf{A}}^{\ell})_{ij} = 2^{-1}3^{-\ell}$ and, by Theorem 1,

$$\beta_i(k,k)^{\ell} = \left(\frac{\sinh\left(\sqrt{-k}r_{i,\exp}\right)}{\sqrt{-k}r_{i,\exp}}\right)^{\ell}$$

Therefore, $\beta_i(k,k)^{\ell} > (\tilde{\mathbf{A}}^{\ell})_{ij}$ when

$$\left(\frac{\sinh\left(\sqrt{-k}r_{i,\exp}\right)}{\sqrt{-k}r_{i,\exp}}\right)^{\ell} > \frac{1}{3^{\ell}} > \frac{1}{2\cdot 3^{\ell}}$$

$$\frac{\sinh\left(\sqrt{-k}r_{i,\exp}\right)}{\sqrt{-k}r_{i,\exp}} > \frac{1}{3}.$$

This example suggests that over-squashing is indeed less severe in HGNNs on graphs exhibiting negative curvature.

C Proof of Lemma 1

We first note a comparison lemma from chapter 6.2 in [26] that yields bounds on the differential of the exponential and logarithmic maps.

Lemma 2. *Assume that $(\mathcal{M}, g)$ satisfies $k \leq K_{\mathbf{x}}(\mathbf{u}, \mathbf{v}) \leq K$ for all $\mathbf{x} \in \mathcal{M}$ and $\mathbf{u}, \mathbf{v} \in \mathcal{T}_{\mathbf{x}}\mathcal{M}$. Let Df denote the differential of a map f. Then for the exponential and logarithmic map at $\mathbf{x}$ and for a radius r around $\mathbf{x}$ we have*

$$\|D\exp_{\mathbf{x}}\|_2 \leq \max\left\{1, \frac{\mathrm{sn}_k(r)}{r}\right\},$$
$$\|D\log_{\mathbf{x}}\|_2 \leq \min\left\{1, \frac{\mathrm{sn}_K(r)}{r}\right\}$$

where $\mathrm{sn}_\kappa(\cdot)$ is the generalized sine function given sectional curvature κ

$$\mathrm{sn}_\kappa(r) := \begin{cases} \frac{\sin(\sqrt{\kappa}r)}{\sqrt{\kappa}} & \text{if } \kappa > 0 \\ r & \text{if } \kappa = 0 \\ \frac{\sinh(\sqrt{-\kappa}r)}{\sqrt{-\kappa}} & \text{if } \kappa < 0 \end{cases}.$$

We use the above lemma to derive a bound on the product of norms of the exponential and logarithmic maps in equation (1) as stated in Lemma 1.

Proof. Let $r_{j,\exp} = \sup_\ell \left\| \sum_{z \in \mathcal{N}(j)} \tilde{\mathbf{A}}_{jz} \mathbf{W}^{(\ell)} \log_{\mathbf{p}}\left(\mathbf{x}_z^{(\ell)}\right) \right\|_2$ denote the maximum radius around $\mathbf{x}$ for the exponential map and $r_{j,\log} = \sup_\ell \|\mathbf{x}_z^{(\ell)}\|_g$ denote the maximum radius for the logarithmic map given equation (1). Applying Lemma 2, there are three possible cases for the bounds k and K:

<u>Case 1:</u> $k < K \leq 0$. We then have

$$\left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 \leq \max\left\{1, \frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}}\right\} \cdot \max_{z \in \mathcal{N}(j)} \min\left\{1, \frac{\sinh\left(\sqrt{-K}r_{j,\log}\right)}{\sqrt{-K}r_{j,\log}}\right\}.$$

Since $\frac{\sinh(x)}{x} > 1$ for all $x \neq 0$, we obtain the bound

$$\left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 \leq \frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}}.$$

<u>Case 2:</u> $k < 0 < K$. We then have

$$\left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 \leq \frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}} \cdot \max_{z \in \mathcal{N}(j)} \min\left\{1, \frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}}\right\}.$$

Since $\frac{\sin(x)}{x} < 1$ for all $x \neq 0$, we obtain the bound

$$\left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 \leq \frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}} \cdot \max_{z \in \mathcal{N}(j)} \frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}}.$$

<u>Case 3:</u> $0 \leq k < K$. We then have

$$\begin{aligned} \left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 &\leq \max\left\{1, \frac{\sin\left(\sqrt{k}r_{j,\exp}\right)}{\sqrt{k}r_{j,\exp}}\right\} \cdot \max_{z \in \mathcal{N}(j)} \min\left\{1, \frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}}\right\} \\ &= \max_{z \in \mathcal{N}(j)} \frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}}. \end{aligned}$$

<u>Case 4:</u> $0 = k = K$. Then we have

$$\left\|D\exp_{\mathbf{p}}\right\|_2 \left\|D\log_{\mathbf{p}}\right\|_2 \leq \max\left\{1, \frac{r_{j,\exp}}{r_{j,\exp}}\right\} \cdot \max_{z \in \mathcal{N}(j)} \min\left\{1, \frac{r_{j,\log}}{r_{j,\log}}\right\} = 1.$$

Combining all of the cases above, we obtain the bound

$$\begin{aligned}\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\| &\leq \begin{cases}\frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}} & k<K\leq 0\\ \frac{\sinh\left(\sqrt{-k}r_{j,\exp}\right)}{\sqrt{-k}r_{j,\exp}}\cdot\max_{z\in\mathcal{N}(j)}\frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}} & k<0<K\\ \max_{z\in\mathcal{N}(j)}\frac{\sin\left(\sqrt{K}r_{j,\log}\right)}{\sqrt{K}r_{j,\log}} & 0\leq k<K\\ 1 & k=K=0\end{cases}\\ &=\beta_j(k,K).\end{aligned}$$

□

D Proof of Theorem 1

Proof. We prove the bound by induction on the number of layers ℓ. For the base case of $\ell = 1$, we have

$$\begin{aligned}\left\|\frac{\partial\mathbf{x}_i^{(1)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2 &= \left\|\frac{\partial}{\partial\mathbf{x}_j^{(0)}}\left[\sigma\left(\exp_{\mathbf{p}}\left(\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\mathbf{W}^{(0)}\log_{\mathbf{p}}\left(\mathbf{x}_z^{(0)}\right)\right)\right)\right]\right\|_2\\ &\leq c_\sigma\left\|D\exp_{\mathbf{p}}\right\|_2\left\|\mathbf{W}^{(0)}\right\|_2\left\|D\log_{\mathbf{p}}\right\|_2\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\left\|\frac{\partial\mathbf{x}_z^{(0)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2\\ &\leq c_\sigma w\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\|_2\tilde{\mathbf{A}}_{ij}\left\|\frac{\partial\mathbf{x}_j^{(0)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2\\ &= c_\sigma w\tilde{\mathbf{A}}_{ij}\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\|_2.\end{aligned}$$

If we let $\beta_i(k,K)$ be the bound on $\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\|$ defined in Lemma 2, the norm of the Jacobian in the base case (i.e. $\ell = 1$) is bounded by

$$\left\|\frac{\partial\mathbf{x}_i^{(1)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2 \leq c_\sigma w\beta_i(k,K)\tilde{\mathbf{A}}_{ij}.$$

We now assume the bound to be satisfied for ℓ layers and use induction to show that it holds for $\ell+1$.

$$\begin{aligned}\left\|\frac{\partial\mathbf{x}_i^{(\ell+1)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2 &= \left\|\frac{\partial}{\partial\mathbf{x}_j^{(0)}}\left[\sigma\left(\exp_{\mathbf{p}}\left(\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\mathbf{W}^{(\ell)}\log_{\mathbf{p}}\left(\mathbf{x}_z^{(\ell)}\right)\right)\right)\right]\right\|_2\\ &\leq c_\sigma w\left\|D\exp_{\mathbf{p}}\right\|_2\left\|D\log_{\mathbf{p}}\right\|_2\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\left\|\frac{\partial\mathbf{x}_z^{(\ell)}}{\partial\mathbf{x}_j^{(0)}}\right\|_2\\ &\leq c_\sigma w\beta_i(k,K)\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\left[c_\sigma^\ell w^\ell\beta_i(k,K)^\ell\left(\tilde{\mathbf{A}}^\ell\right)_{zj}\right]\\ &= c_\sigma^{\ell+1}w^{\ell+1}\beta_i(k,K)^{\ell+1}\sum_{z\in\mathcal{N}(i)}\tilde{\mathbf{A}}_{iz}\left(\tilde{\mathbf{A}}^\ell\right)_{zj}\\ &= c_\sigma^{\ell+1}w^{\ell+1}\beta_i(k,K)^{\ell+1}\left(\tilde{\mathbf{A}}^{\ell+1}\right)_{ij}.\end{aligned}$$

□