Reviewer 1 We thank the reviewer for the insightful and encouraging comments.

Regarding the choice of logm/expm vs. any other pair of invertible functions Logarithmic and exponential maps are the principled way to define a mapping to a locally Euclidean space on Riemannian manifolds (and are, for instance, used in the definition of geodesics on the manifold). For our purpose, they have the additional benefit that they are parameter-free, whereas an autoencoder would have to be learned. On the manifolds that we are considering, they also have closed-form solutions, are differentiable, and are computationally efficient. Please see also Table 2, which lists additional experiments with a 1- and 2-layer autoencoder and which illustrates the advantages of our method.

Regarding the difference to hyperbolic NNs The work of Ganea et al. is indeed related to our work and we tried to highlight these connections. Important differences to our work include: (1) Our formalism is more general as we can handle any Riemannian manifold with differentiable logm/expm. Our formalism allows us also to easily extend GNNs to the Lorentz model which provides important benefits as shown in our experiments. (2) We propose a generalization of message-passing GNNs to Riemannian manifolds whereas the work of Ganea et al. is focused on classic neural-networks such as RNNs. (3) We propose a novel centroid-based method to classify embeddings in Riemannian NNs.

Regarding the benefits of our centroid-based classification There exist no general isometric mappings between Hyperbolic and Euclidean space. Moreover, a parametric function to learn a task-specific mapping between both spaces would (in our experience) likely require many parameters and training examples to learn the complex relations. For this reason, we chose a "non-parametric" centroid-based approach which is computationally efficient and flexible.

	Cora	**Citeseer**	**Pubmed**	**ZINC**
Lorentz	1.014	1.048	1.03	1.285
Poincaré	1.041	1.125	1.083	1.34

Table 1: Relative runtime (defined as clock time of Poincaré or Lorentz / clock time of Euclidean)

	logP	**QED**	**SAS**
1-layer AE	3.2 ± 0.03	6.7 ± 0.01	9.5 ± 0.05
2-layer AE	3.3 ± 0.04	6.9 ± 0.02	9.6 ± 0.08
Log and exp	2.4 ± 0.02	4.7 ± 0.0	7.7 ± 0.06

Table 2: Lorentzian GNN (dimension 256) on ZINC using either an autoencoder or log & exp maps.

Reviewer 2 We thank the reviewer for the insightful and encouraging comments.

Regarding code release Thank you for raising this issue. We do indeed plan to fully open-source our code on Github.

Regaring the time complexity of our method The computational demand of our method is similar to standard GCNs. Logm and Expm have simple closed-form solutions and require only computationally cheap operations (vector norms, elementwise division, cosh, etc.). The mapping between Lorentz and Poincare is also efficient for the same reasons (see Eq. 12/13). Table 1 shows the relative runtime performance for Lorentz and Poincare GNNs compared to Euclidean GNNs. It can be seen that the computational overhead is low and ranges between 1.01-1.34 times the Euclidean model.

Regarding reference points in centroid classification The reference points are actually learned jointly with the GNN using backprop. This enables the model to infer relevant areas in the embedding jointly with the GNN.

Reviewer 3 We thank the reviewer for the comments which will help us improve our paper. However, we respectfully disagree with several aspects of the review:

Regarding experiments on synthetic datasets Please note that a Lorentzian GNN with only 10 dimensions outperforms the Euclidean GNN with 256 dimensions. The improvements are statistical significant as can be seen from the reported error bars. Furthermore, there is a large improvement over Euclidean GNNs in low-dimensions (F1 of 94.1 vs 77.2 in d3) which further illustrates the benefits of Hyperbolic geometry for this task. The goal of these synthetic experiments is to evaluate the ability of models to capture graph topology and our results show clearly that Hyperbolic GNNs outperform Euclidean GNNs on this task in absolute performance and additionally have large benefits in smaller dimensions.

Regarding experiments on real-world datasets The task of molecular prediction is a common benchmark for GNNs. For this task, we do not only compare to Euclidean GNNs, but also to current state-of-the-art methods for molecular prediction, i.e., MPNN, DTNN, GGNN. Our evaluation in Table 3 of the paper shows clearly that we get statistical significant improvements using our approach. In our experiments on Blockchain data, we furthermore compare to ARIMA which is a commonly used and strong baseline for this task. It is therefore unclear to us why the review criticizes a lack of state-of-the-art baselines and/or significant improvements as we clearly show both.

Regarding technical contributions Please see the response to R1 for a discussion of novel contributions compared to hyperbolic NNs. Furthermore, our paper is the first to study the benefits of hyperbolic geometry in the context of GNNs. Our experiments show the advantages of this approach both with regard to representational efficiency (strong performance of low dimensional embeddings) and generalization ability (state-of-the-art results on synthetic and real-world data). We therefore believe that our manuscript makes valuable contributions to advance the state-of-the-art for GNNs.