We thank the reviewers for their positive and valuable feedback. We recall that our paper proposes a general framework to learn **ultrahyperbolic** representations. The proposed representations lie on a pseudo-Riemannian manifold with constant nonzero curvature, they generalize both hyperbolic and spherical representations that are popular in machine learning. The main difficulty of learning such representations is that they lie on a manifold whose metric need not be positive definite, and the manifold is non-Riemannian in most cases (except for the hyperbolic and spherical cases as explained in the paper). We introduce the necessary differential geometry tools (e.g. geodesics, exponential/logarithm maps) to measure dissimilarity between points, and also propose optimizers for differentiable functions defined on such manifolds. In particular, we explain why the pseudo-Riemannian gradient is not a descent direction. We then propose a simple, efficient and non-trivial descent direction defined in the tangent space (see Eq. (12)).

Improving readability: Our contributions are mainly theoretical, and we agree with most reviewers (**R1**,**R3**) that the pseudo-Riemannian optimizer introduced in Section 4.2 is a major contribution. Due to lack of space, we provided the detailed explanations with proofs in the supp. material. However, according to the NeurIPS 2020 website, camera-ready versions are allowed a ninth content page. To improve readability, we will include the extended version of the optimizer subsection in the main paper, if accepted. We will also account for the suggestions of the reviewers as follows.

R1: Thank you for your suggestions. **(1)** We will indicate in Section 2 that for any $\beta < 0$, $\mathcal{Q}_\beta^{p,q}$ is homothetic to $\mathcal{Q}_{-1}^{p,q}$, β can then be considered to be -1. **(2)** We did not exploit the extrinsic distance in Eq. (4), except in the null geodesic case since the formulation is similar in this particular case. The goal of lines 83-88 was to explain that many machine learning approaches consider the extrinsic geometry (i.e. ambient space distance) of the spherical or hyperbolic manifold, or its intrinsic geometry (i.e. geodesic distance). Since both distances are increasing functions of each other in the Riemannian cases, choosing one or the other has no major impact. This is not the case in the ultrahyperbolic case, which is why we only consider the intrinsic geometry. **(3)** We explained in the paper how the hyperbolic and spherical cases are special cases of $\mathcal{Q}_\beta^{p,q}$ (lines 87-88 and lines 68-70). We will make it more explicit as suggested. **(4)** Our code is in the supp. material and will be publicly available. We reported some training times in the supp. material (line 540). On Zachary's dataset, the (Euclidean) optimizer in Section 4.1 is 10% faster than the optimizer in Section 4.2 (165 vs 182 seconds) in the setup of line 540 since it requires less computations. We will report the comparisons. **(5)** Lastly, we will explicitly mention that $\forall \mathbf{x} \in \mathcal{Q}_\beta^{p,q},\ g_{\mathbf{x}}(\cdot,\cdot) = \langle \cdot,\cdot \rangle_q$ where $g_{\mathbf{x}} : T_{\mathbf{x}}\mathcal{Q}_\beta^{p,q} \times T_{\mathbf{x}}\mathcal{Q}_\beta^{p,q} \to \mathbb{R}$.

R3: **(1)** Thank you for mentioning Feragen's work that was among the first to study tree-data in the CV and ML community. We will cite it in the introduction when we mention other machine learning works that were also heavily inspired by Gromov's work. Nonetheless, Feragen et al. consider CAT(0) spaces (e.g. hyperbolic spaces). Our work generalizes both hyperbolic and spherical spaces, the latter is not CAT(0). (**R4**,**R3**) **(2) Motivation of Eq. (9):** As explained in the paper, there exist pairs of points $\mathbf{x}, \mathbf{y} \in \mathcal{Q}_\beta^{p,q}$ for which $\log_{\mathbf{x}}(\mathbf{y})$ is not defined. Eq. (9) approximates the dissimilarity when $\log_{\mathbf{x}}(\mathbf{y})$ is not defined but other choices are possible. When a geodesic does not exist, a standard way in differential geometry to calculate curves (and distances) is to consider broken geodesics. One might then consider instead the dissimilarity $\mathsf{d}_\gamma(\mathbf{x}, -\mathbf{x}) + \mathsf{d}_\gamma(-\mathbf{x}, \mathbf{y}) = \pi\sqrt{|\beta|} + \mathsf{d}_\gamma(-\mathbf{x}, \mathbf{y})$ if $\log_{\mathbf{x}}(\mathbf{y})$ is not defined (see line 422 of the supp. material) since $-\mathbf{x} \in \mathcal{Q}_\beta^{p,q}$ and $\log_{-\mathbf{x}}(\mathbf{y})$ is defined. **(3)** We disagree about the fact that we used hacks to create symmetric weights. Our second dataset has an undirected (hence symmetric) weight matrix by default. Moreover, in Zachary's paper, $\mathbf{C}$ was constructed in an *ad hoc* manner and is almost identical to its transpose (i.e. almost symmetric). The weight matrix $\mathbf{C}$ is illustrated in Fig. 3 of Zachary's paper. Our symmetrized matrix $\mathbf{S} = \mathbf{C} + \mathbf{C}^\top$ is very similar to $2\mathbf{C}$, which is why we considered it. In conclusion, our approach can be applied to any undirected weighted graph.

(**R2**,**R3**,**R4**) **Motivation of ultrahyperbolic representations for graphs:** The choice of geometry to represent graphs is still an open problem in general. It depends on the topology of the graph and the kind of relationships between nodes. For instance, hyperbolic geometry was mathematically shown to be appropriate for tree-like graphs, but not for other types of graphs. Ultrahyperbolic geometry has the advantage of generalizing both hyperbolic and spherical geometries and can describe relationships specific to those geometries. In particular, the geodesic *distance* can be written in the same way as the Poincaré and spherical distances as shown in Eq. (8); some parts of the manifold are hyperbolic or spherical as explained in the paper. The converse is not true. The framework might then automatically learn representations to be part of a same hyperbolic or spherical part of the manifold depending on the context. Those reasons led us to consider hierarchical graphs that were similar to trees, but where the presence of cycles in the graph limited the relevance of hyperbolic geometry. We experimentally validated our intuition. The choice of geometry with constant nonzero curvature (i.e. the optimal number of time and space dimensions q and p) then seems to depend on how much the graph is similar to a tree or to a graph where spherical geometry is appropriate, such as cycle graphs (or a mix of both). We would also like to emphasize that ultrahyperbolic geometry can describe some graph concepts differently, if not better, than hyperbolic and spherical geometries. For instance, it is known that *triadic closure* is a concept in social network theory that is too extreme to hold true across very large, complex networks. In other words, if $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}, \mathbf{z})$ are strongly tied, triadic closure would induce that $(\mathbf{y}, \mathbf{z})$ are strongly tied. As explained in line 144, the fact that we can find triplets that satisfy $\mathsf{d}_\gamma(\mathbf{x}, \mathbf{y}) = \mathsf{d}_\gamma(\mathbf{x}, \mathbf{z}) = 0$ but $\mathsf{d}_\gamma(\mathbf{y}, \mathbf{z}) > 0$ avoids triadic closure.