Supplementary materials

A Visualization of the SSDM

This section gives a visualization of the SSDM. See Figure 1. Recall that the d-th ***sparse hyperplane*** (SHP), denoted by $\Pi_d^{\mathcal{M}}$, is defined by $\Pi_d^{\mathcal{M}} := \{p \in \mathcal{M} \mid \langle e_d, \log_{\mathrm{o}}(p)\rangle_{\mathrm{o}} = 0\}$. This means that $\Pi_d^{\mathcal{M}} = \exp_{\mathrm{o}}(\mathrm{span}\{e_1, e_2, \ldots, e_{d-1}, e_{d+1}, e_{d+2}, \ldots, e_D\})$. For example, $\Pi_1^{\mathcal{M}} = \Pi_1$ in Figure 1 is the image under the exponential map $\exp_{\mathrm{o}}$ of a linear subspace spanned by e_2 and e_3. The d-th element $\delta_d^{\mathcal{M}}(p) = \delta_d(p)$ of SSDM measures the signed distance from the d-th SHP to the point p. For example, $\delta_1(p)$ is the signed distance from Π_1 to p in Figure 1.

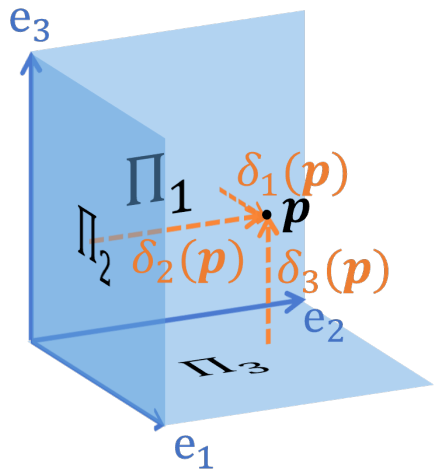

Figure 1: The SSDM's visualization for a 3-dimensional CHMOO case. The hyperplane $\Pi_d^{\mathcal{M}} = \Pi_d$ is the d-th sparse hyperplane (SHP). The d-th element $\delta_d^{\mathcal{M}}(p) = \delta_d(p)$ of SSDM measures the signed distance from the d-th SHP to the point p.

B Detailed explanation of Example 7

The function $f(p) = |p|$ is differentiable at $p \neq 0$ and the derivative is given by $\frac{\mathrm{d}}{\mathrm{d}p} f(p) = \mathrm{sgn}(p)$. Suppose that the learning rate is $\alpha > 0$ and the initial point is $p^{(0)} \neq 0$. By the symmetry about the origin, we can assume that $p^{(0)} > 0$ without loss of generality. Then the gradient descent generates the series $p^{(0)}, p^{(1)}, \ldots$ of points according to the following recursion:

$$p^{(t+1)} \leftarrow p^{(t)} - \alpha \frac{\mathrm{d}}{\mathrm{d}p} f(p^{(t)}) = \begin{cases} p^{(t)} - \alpha & \text{if } p^{(t)} > 0, \\ p^{(t)} + \alpha & \text{if } p^{(t)} < 0. \end{cases} \tag{6}$$

Here, we usually set $p^{(t+1)} \leftarrow p^{(t)}$ if $p^{(t)} = 0$, which we can justify as a subgradient method. We can see from (6) that the algorithm ends up oscillating between $p^{(0)} - \alpha n$ and $p^{(0)} - \alpha(n+1)$ unless $p^{(0)}$ is an integral multiple of α, where $n = \left\lceil \frac{p^{(0)}}{\alpha} \right\rceil$ is the maximum integer that is no greater than $\frac{p^{(0)}}{\alpha}$.

C Proof of Theorem 2

Proof. We prove for the SSDM $\boldsymbol{\delta}^{(\mathbb{D}^D, \boldsymbol{G}_{\cdot}^{\mathrm{P}})}$. It suffices to prove that the absolute values are correct since the logarithmic map at the origin of the Poincarè model does not change the sign of each element. Let $\boldsymbol{h} \in \mathbb{D}^2$ be the foot of the geodesic pass through $\boldsymbol{p}$ on Π_d. Note that $\boldsymbol{h}$ is unique according to Gauss-Bonnet theorem. We have that $\boldsymbol{h} = \mathrm{argmin}_{\boldsymbol{q}} \Delta_{(\mathbb{D}^D, \boldsymbol{G}_{\cdot}^{\mathrm{P}})}(\boldsymbol{p}, \boldsymbol{q})$ from hyperbolic Pythagorean theorem. In the following, we regard the ball of the Poincarè model as a unit ball in

Euclidean space and discuss using elementary geometry. A geodesic in hyperbolic space is now an arc orthogonal to the unit ball and Π_d and passing through $\boldsymbol{p}$. Define $\boldsymbol{p}' = \frac{\boldsymbol{p}}{\boldsymbol{p}^\top \boldsymbol{p}}$ and $\boldsymbol{h}' = \frac{\boldsymbol{h}}{\boldsymbol{h}^\top \boldsymbol{h}}$. Also denote by $\boldsymbol{m}$ the midpoint of $\boldsymbol{p}$ and $\boldsymbol{p}'$ and by $\boldsymbol{j}$ the midpoint of $\boldsymbol{h}$ and $\boldsymbol{h}'$. Note that $\boldsymbol{j}$ is the center of the arc drawn by the geodesic. Since the arc is orthogonal to the unit ball, it also passes through $\boldsymbol{p}'$ and $\boldsymbol{h}'$ according to the power of a point theorem. The subplane including the arc also contains $\boldsymbol{p}$, $\boldsymbol{h}$, and $\boldsymbol{p}'$. Hence, the following discussion is on the subplane. We regard the axis in the subplane on the intersection of the subplane and Π_d as x-axis, and the other axis orthogonal to Π_d to y-axis. We indicate the coordinate of the $\boldsymbol{p}$ in the subplane by $[x \quad y]^\top$ and that of $\boldsymbol{h}$ by $[h \quad 0]^\top$. The coordinates of $\boldsymbol{p}'$ and $\boldsymbol{h}'$ are $\frac{1}{x^2+y^2}[x \quad y]^\top$ and $[1/h \quad 0]^\top$, respectively. See also Figure 2. We have that $|\boldsymbol{m}| = \frac{1}{2}\left(\sqrt{x^2+y^2} + \frac{1}{\sqrt{x^2+y^2}}\right)$ and $|\boldsymbol{j}| = \frac{1}{2}\left(h + \frac{1}{h}\right)$. By similarity of two

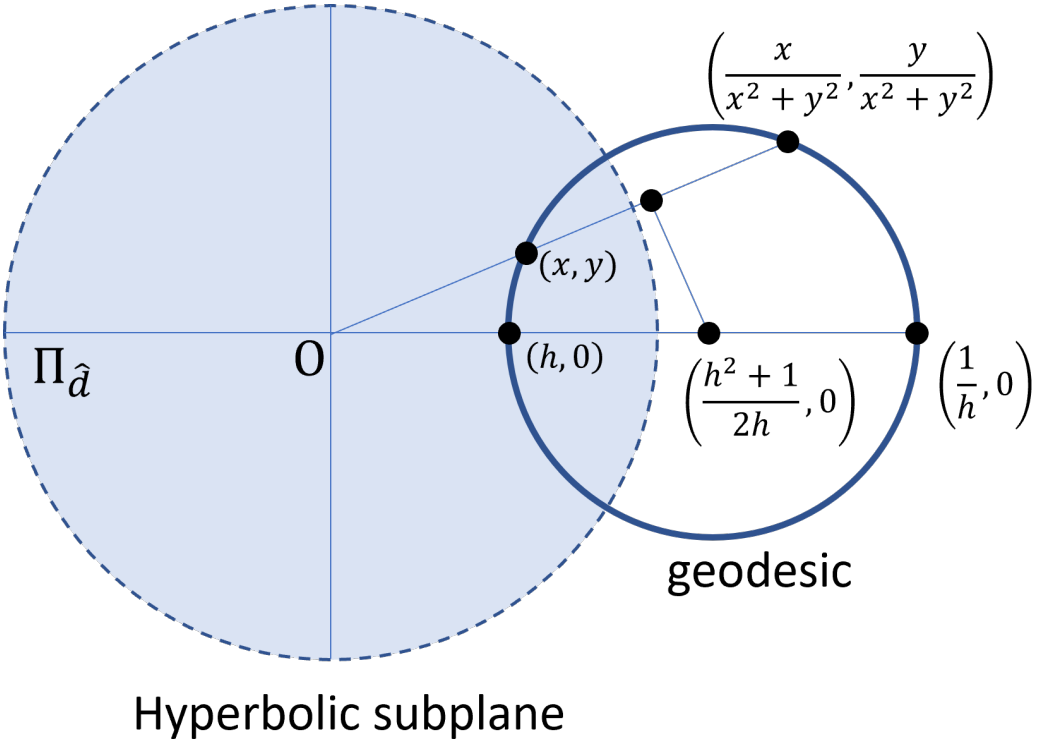

Figure 2: Hyperbolic subdisk.

right triangles, we have that $\frac{\sqrt{x^2+y^2}}{x} = \frac{|\boldsymbol{j}|}{|\boldsymbol{m}|}$. Hence, $|\boldsymbol{j}| = \frac{x^2+y^2+1}{2x}$. Noting that $h < 1 < \frac{1}{h}$, we have that $h = \frac{x^2+y^2-\sqrt{(x^2+y^2)^2-4x^2}}{2x}$. We get the expected result by substituting this to the distance formula of the Poincarè model: $\Delta_{(\mathbb{D}^2, \boldsymbol{G}^{\mathrm{p}}_\cdot)}(\boldsymbol{p}, \boldsymbol{q}) = \operatorname{acosh}\left(1 + \frac{2|\boldsymbol{p}-\boldsymbol{q}|^2}{(1-|\boldsymbol{p}|^2)(1-|\boldsymbol{q}|^2)}\right)$. Specifically,

$$
\begin{aligned}
&\Delta_{(\mathbb{D}^2, \boldsymbol{G}^{\mathrm{p}}_\cdot)}([x \quad y], [h \quad 0]) \\
&= \operatorname{acosh}\left(1 + \frac{2\left((x-h)^2 + y^2\right)}{(1-(x^2+y^2))(1-h^2)}\right) \\
&= \operatorname{acosh}\left(\sqrt{1 + \frac{4y^2}{\left(1-(x^2+y^2)\right)^2}}\right) \\
&= \operatorname{asinh}\left(\frac{2y}{\left(1-(x^2+y^2)\right)^2}\right).
\end{aligned}
\tag{7}
$$

We complete the proof by recalling that $y = p_d$ and $(x^2 + y^2) = \boldsymbol{p}^\top \boldsymbol{p}$. □

D EXPLICIT PSEUDOCODE OF THE HISTA

Algorithm 2 shows the explicit form of HISTA on the Poincaré model. Here, sinhc is defined by

$$\operatorname{sinhc}(x) \coloneqq \begin{cases} 1 & \text{if } x = 0, \\ \frac{\sinh x}{x} & \text{if } x \neq 0. \end{cases} \tag{8}$$

Algorithm 2 HISTA (Explicit form)

Require: $\boldsymbol{p}_{\text{init}} \in \mathbb{D}^D$: initial point,
$\alpha \in \mathbb{R}_{>0}$: learning rate,
$T \in \mathbb{Z}_{\geq 0}$: # iterations.
Ensure: $\boldsymbol{p}_{\text{output}} \in \mathbb{D}^D$
$\boldsymbol{p}^{(0)} \leftarrow \boldsymbol{p}_{\text{init}}$
for $t \leftarrow 1, 2, \ldots, T$ **do**
$\boldsymbol{\gamma}^{(t)} \leftarrow \boldsymbol{\partial}|_{\boldsymbol{p}^{(t-1)}} J$
$\rho^{(t)} \leftarrow \frac{4}{\left(1 - \left|\boldsymbol{p}^{(t-1)}\right|^2\right)^2}$
$\boldsymbol{g}^{(t)} \leftarrow \left(\rho^{(t)}\right)^2 \boldsymbol{\gamma}^{(t)}$
$\boldsymbol{q}^{(t-1)} \leftarrow \rho^{(t)} \cdot \frac{\left[\cosh\left(\left|-\alpha \boldsymbol{g}^{(t)}\right|\right) - \rho^{(t)} \alpha \left(\boldsymbol{g}^{(t)}\right)^\top \left(\boldsymbol{p}^{(t-1)}\right)\right] \boldsymbol{p}^{(t-1)} + \operatorname{sinhc}\left(\left|\alpha \boldsymbol{g}^{(t)}\right|\right) \boldsymbol{g}^{(t)}}{1 + \left(\rho^{(t)} - 1\right) \cosh\left(\left|-\alpha \boldsymbol{g}^{(t)}\right|\right) - \left(\rho^{(t)}\right)^2 \alpha \left(\boldsymbol{g}^{(t)}\right)^\top \left(\boldsymbol{p}^{(t-1)}\right) \operatorname{sinhc}\left(\left|\alpha \boldsymbol{g}^{(t)}\right|\right)}$
$\boldsymbol{\sigma}^{(t)} \leftarrow \operatorname{asinh}\left(\frac{2\boldsymbol{p}}{1 - \boldsymbol{p}^\top \boldsymbol{p}}\right) - \alpha \lambda \mathbf{1}_D$
$\boldsymbol{p}^{(t)} \leftarrow \frac{\sinh\left(\boldsymbol{\sigma}^{(t)}\right)}{\sqrt{1 + \left(\sinh \boldsymbol{\sigma}^{(t)}\right)^\top \left(\sinh \boldsymbol{\sigma}^{(t)}\right)} + 1}$
end for
$p_{\text{output}} \leftarrow p^{(T)}$

E FORMULAE FOR A PRODUCT MANIFOLD.

We discuss the formulae of our sparse representation learning scheme for product manifolds. We do not give detailed proofs, but they can easily be proved using a specific coordinate space. Let $\mathcal{M}^{[1]}, \mathcal{M}^{[2]}, \ldots, \mathcal{M}^{[M]}$ are Riemannian manifolds. The Riemannian product manifold $\mathcal{M} = \mathcal{M}^{[1]} \times \mathcal{M}^{[2]} \times \cdots \times \mathcal{M}^{[M]}$ is given as the topological product manifold of $\mathcal{M}^{[1]}, \mathcal{M}^{[2]}, \ldots, \mathcal{M}^{[M]}$ equipped with the metric tensor defined as follows. For $p = \left(p^{[1]}, p^{[2]}, \ldots p^{[M]}\right) \in \mathcal{M}$, the metric tensor and a direct some decomposition of the tangent space $T_p\mathcal{M}$ is given as follows. For tangent vectors $v, v' \in T_p\mathcal{M}$, let $c : I \to \mathcal{M}$ and $c' : I' \to \mathcal{M}$ be C^∞ curves on $\mathcal{M}$ tangent to v and v' respectively, where $I, I' \subset \mathbb{R}$ is open intervals such that $0 \in I \cap I'$. Here, c being tangent to u means that $c(0) = p$ and $v = \dot{c}\big|_0$, where $\dot{c}\big|_t : C^\infty(\mathcal{M}) \to \mathbb{R}$ for $t \in I$ is defined by $\dot{c}\big|_{t'} f \coloneqq \frac{\mathrm{d}}{\mathrm{d}t}(c(t))\big|_{t'}$. There exist C^∞ curves $c^{[m]} : I \to \mathcal{M}$ and $c'^{[m]} : I' \to \mathcal{M}$ for $m = 1, 2, \ldots, M$ such that $c^{[m]}(t) = c'^{[m]}(t) = p^{[m]}$ for $m = 1, 2, \ldots, M$ and $c(t) = \left(c^{[1]}(t), c^{[2]}(t), \ldots, c^{[M]}(t)\right)$ and $c'(t) = \left(c'^{[1]}(t), c'^{[2]}(t), \ldots, c'^{[M]}(t)\right)$. Then we can define $\tilde{v}^{[m]} \coloneqq c\big|_0 \in T_p\mathcal{M}$ and $\tilde{v}'^{[m]} \coloneqq c'\big|_0 \in T_p\mathcal{M}$ for $m = 1, 2, \ldots, M$. We can prove that it does not depend on the choice of c. In the following, we denote the linear operation to obtain $\tilde{v}^{[m]} \in T_{p^{[m]}}\mathcal{M}^{[m]}$ from $v \in T_p\mathcal{M}$ by $\pi_p^{[m]} : T_p\mathcal{M} \to T_{p^{[m]}}\mathcal{M}^{[m]}$. We define the metric tensor on $T_p\mathcal{M}$ by

$$\langle v, v' \rangle_p^{\mathcal{M}} \coloneqq \sum_{m=1}^{M} \langle \tilde{v}^{[m]}, \tilde{v}'^{[m]} \rangle_p^{\mathcal{M}}. \tag{9}$$

We can prove that the above metric is symmetric and positive-definite.

A tangent vector $\tilde{v}^{[m]} \in T_{p^{[m]}}\mathcal{M}^{[m]}$ can be identified with $v^{[m]} \in T_p\mathcal{M}$ that satisfies

$$\begin{cases} \pi_p^{[m']}\left(v^{[m]}\right) = \tilde{v}^{[m]} & \text{if} \quad m' = m, \\ \pi_p^{[m']}\left(v^{[m]}\right) = 0 & \text{if} \quad m' \neq m. \end{cases} \tag{10}$$

We can prove that the above $v^{[m]}$ is uniquely defined. By the above identification, we can identify each $T_{p^{[m]}}\mathcal{M}^{[m]}$ with $T_p^{[m]}\mathcal{M} \coloneqq \left(\pi_p^{[m]}\right)^{-1}\left(T_{p^{[m]}}\mathcal{M}^{[m]}\right) \subset T_p\mathcal{M}$, which is a linear subspace of $T_p\mathcal{M}$ with the same dimension as $T_{p^{[m]}}\mathcal{M}^{[m]}$. Also, we can prove that $T_{p^{[m]}}\mathcal{M}^{[m]} = \bigoplus_{m=1}^{M} T_p^{[m]}\mathcal{M}$ is the orthogonal direct sum decomposition. Since the restriction $\pi_p^{[m]}\Big|_{T_p^{[m]}\mathcal{M}}$ is one-to-one, we can define its inverse map $\left(\pi_p^{[m]}\Big|_{T_p^{[m]}\mathcal{M}}\right)^{-1} : T_{p^{[m]}}\mathcal{M}^{[m]} \to T_p^{[m]}\mathcal{M}$.

Based on the identification, if we have multiple CHMOOs $\left(\mathcal{M}^{[m]}, \mathrm{o}^{[m]}, \left(\tilde{e}_1^{[m]}, \tilde{e}_2^{[m]}, \ldots, \tilde{e}_{D^{[m]}}^{[m]}\right)\right)_{m=1}^{M}$, we can see that $\left(e_1^{[1]}, e_2^{[1]}, \ldots, e_{D^{[1]}}^{[1]}, e_1^{[2]}, e_2^{[2]}, \ldots, e_{D^{[2]}}^{[2]}, \ldots, e_1^{[M]}, e_2^{[M]}, \ldots, e_{D^{[M]}}^{[M]}\right)$ is an ONB in $T_p\mathcal{M}$, where $e_d^{[m]} = \left(\pi_{\mathrm{o}}^{[m]}\Big|_{T_{\mathrm{o}}^{[m]}\mathcal{M}}\right)^{-1}\left(\tilde{e}_d^{[m]}\right)$ for $m = 1, 2, \ldots, M$ and $d = 1, 2, \ldots, D^{[m]}$, $\mathcal{M} = \mathcal{M}^{[1]} \times \mathcal{M}^{[2]} \times \ldots, \times \mathcal{M}^{[M]}$, and $\mathrm{o} = \left(\mathrm{o}^{[1]}, \mathrm{o}^{[2]}, \ldots, \mathrm{o}^{[M]}\right) \in \mathcal{M}$. Hence, we can define the product CHMOO $\left(\mathcal{M}, \mathrm{o}, \left(e_1^{[1]}, e_2^{[1]}, \ldots, e_{D^{[1]}}^{[1]}, e_1^{[2]}, e_2^{[2]}, \ldots, e_{D^{[2]}}^{[2]}, \ldots, e_1^{[M]}, e_2^{[M]}, \ldots, e_{D^{[M]}}^{[M]}\right)\right)$.

We are interested in the SHP, Riemannian 0-norm, SSDM, its inverse, CH 1-norm, and CHSTO of the product CHMOO. In the following, we derive the formula for the SSDM, which enable us to calculate the others. To achieve that, we review the basic property of the product Riemannian manifold.

Suppose that $p = \left(p^{[1]}, p^{[2]}, \ldots, p^{[M]}\right) \in \mathcal{M}$ and $v \in T_p\mathcal{M}$ and we consider the unique decomposition $v = \sum_m^M v^{[m]}$, where $v^{[m]} \in T_p^{[m]}\mathcal{M}$ for $m = 1, 2, \ldots, M$. Define $\tilde{v}^{[m]} = \pi_p^{[m]}\Big|_{T_p^{[m]}\mathcal{M}}\left(v^{[m]}\right)$ for $m = 1, 2, \ldots, M$. We can see that if we can define $\exp_{p^{[m]}}\left(\tilde{v}^{[m]}\right) \in \mathcal{M}^{[m]}$ for $m = 1, 2, \ldots, M$, it follows that $\exp_p(v) = \left(\exp_{p^{[1]}}\left(\tilde{v}^{[1]}\right), \exp_{p^{[2]}}\left(\tilde{v}^{[2]}\right), \ldots, \exp_{p^{[M]}}\left(\tilde{v}^{[M]}\right)\right)$. We can calculate the logarithmic map similarly.

The distance between two points $p = \left(p^{[1]}, p^{[2]}, \ldots, p^{[M]}\right)$ and $q = \left(q^{[1]}, q^{[2]}, \ldots, q^{[M]}\right)$ is given by

$$\Delta_{\mathcal{M}}(p, q) = \sqrt{\sum_{m=1}^{M}\left[\Delta_{\mathcal{M}^m}\left(p^{[m]}, q^{[m]}\right)\right]^2}. \tag{11}$$

From the above property, we can confirm that the SHP $\mathit{\Pi}_{[m],d}^{\mathcal{M}}$ of the product CHMOO corresponding to $e_d^{[m]} \in T_{\mathrm{o}}^{[m]}\mathcal{M} \subset T_{\mathrm{o}}\mathcal{M}$ is given by

$$\mathit{\Pi}_{[m],d}^{\mathcal{M}} = \left\{ \left(p^{[1]}, p^{[2]}, \ldots, p^{[M]}\right) \middle| p^{[m']} \in \mathit{\Pi}_d^{\mathcal{M}^{[m]}} \text{ if } m' = m, \quad p^{[m']} \in \mathcal{M}^{[m]} \text{ if } m' \neq m, \right\}. \tag{12}$$

Hence, we immediately get the SSDM formula.

$$\boldsymbol{\delta}^{\mathcal{M}}(p) = \begin{bmatrix} \boldsymbol{\delta}^{\mathcal{M}^{[1]}}\left(p^{[1]}\right) \\ \boldsymbol{\delta}^{\mathcal{M}^{[2]}}\left(p^{[2]}\right) \\ \cdots \\ \boldsymbol{\delta}^{\mathcal{M}^{[M]}}\left(p^{[M]}\right) \end{bmatrix}. \tag{13}$$

The above formula enables us to calculate the SHP, Riemannian 0-norm, SSDM, its inverse, CH 1-norm, and CHSTO for the product CHMOO if we can calculate them for each component CHMOO.

For example, we can calculate them for the product of the EVCHMOO and hyperbolic CHMOOs. Note that the product of EVCHMOOs is again a higher dimensional EVCHMOO, while the product of hyperbolic CHMOOs is not a hyperbolic CHMOO since the product of hyperbolic space is not a hyperbolic space.

F NUMERICAL EXPERIMENTS: HISTA AVOIDS THE OSCILLATION

Remark 9 states that our motivation in this paper has almost been achieved. The last thing we need to do is to confirm by numerical experiments that our HISTA avoids the oscillation issue. Hence, we compare the HISTA and RGD for sparse solution and non-sparse solution cases.

We consider minimizing the square distance with the hyperbolic 1-norm regularization: $L(\boldsymbol{z}) = \left[\Delta_{(\mathbb{D}^2, \boldsymbol{G}^{\mathrm{P}})}(\boldsymbol{z}, \boldsymbol{z}')\right]^2 + \lambda \|\boldsymbol{z}\|_{1,(\mathbb{D}^2, \boldsymbol{G}^{\mathrm{P}})}$. Here, we set $\boldsymbol{z}' = [0.0 \quad 0.0]^\top, [0.0 \quad 0.8]^\top, [0.4 \quad 0.8]^\top$. We expect that the true solution is sparse for the first two cases and non-sparse for the last case, though we do not know the analytic solution for the latter two. We set $\lambda = 1.0$ and $\alpha = 0.1$ for all cases.

Figure 3 shows that the HISTA outperforms the RGD for $\boldsymbol{z}' = [0.0 \quad 0.0]^\top, [0.0 \quad 0.4]^\top$ in terms of the objective function's value as well as obtaining a sparse solution. For $\boldsymbol{z}' = [0.4 \quad 0.8]^\top$, the RGD can outperform the HISTA. We also observe a "bounce back" effect by the HISTA, which could be a drawback. Still, the HISTA is stable for all the cases, while the oscillation of the RGD is significant for $\boldsymbol{z}' = [0.0 \quad 0.0]^\top$. See also Figure 4. Our results confirm that the superiority of the HISTA over the RGD in the function value for sparse solution cases. Still, detecting the cause of the bounce back effect by the HISTA would be interesting future work.

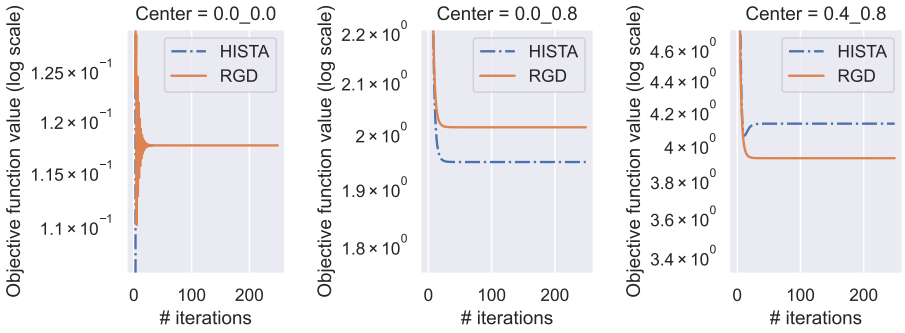

Figure 3: The optimization performances of HISTA and RGD in minimizing the square distance from a fixed point $\boldsymbol{z}'$ with the hyperbolic 1-norm, where $\boldsymbol{z}' = [0.0, 0.0]^\top$ (**Left**), $\boldsymbol{z}' = [0.0, 0.8]^\top$ (**Center**), and $\boldsymbol{z}' = [0.4, 0.8]^\top$ (**Right**). Note that in (**Left**), the blue dashed line indicating HISTA goes infinitely downward because the objective function value reaches 0. See Figure 4 for details.

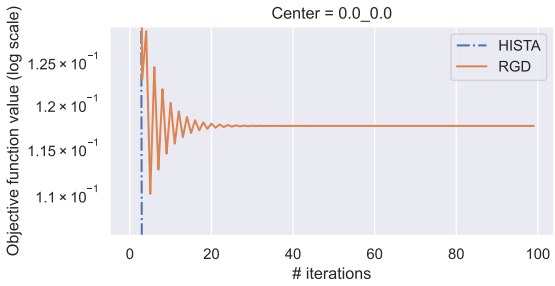

Figure 4: The optimization performances of HISTA and RGD in minimizing the square distance from a fixed point $\boldsymbol{z}'$ with the hyperbolic 1-norm, where $\boldsymbol{z}' = [0.0, 0.0]^\top$. Here we focus on the first 100 iterations to see the oscillation issue of the RGD.

G NUMERICAL EXPERIMENTS ON GRAPH EMBEDDING SETTING

This section gives numerical-experimental results. Note that the objective is not to achieve state-of-the-art representations, but to show how our sparse learning scheme influences HSBRL.

We denote a graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the vertex set and $\mathcal{E}$ is the edge set. Since edges are the most fundamental form of information about entity relations, graph embedding has wide applications. Hence, we choose graph embedding as the problem on which we evaluate the performance of our sparse learning scheme. Our objective here is NOT to maximize the quality of representations, but to compare our sparse learning scheme with possible alternatives. Hence, we use the following simplest graph embedding setting. Let $\mathrm{C}(\mathcal{V}, 2)$ be the set of subsets of $\mathcal{V}$, whose size is two. That is, $\mathrm{C}(\mathcal{V}, 2)$ is the set of unordered vertex pairs. Define the label of a vertex pair $y_{u,v} \in \{-1, +1\}$ and the sample weight $w_{u,v} \in \mathbb{R}_{>0}$ for $u, v \in \mathcal{V}$ such that $u \notin v$ by

$$\begin{aligned} y_{u,v} &\coloneqq \begin{cases} +1 & \text{if } \{u, v\} \in \mathcal{E}, \\ -1 & \text{if } \{u, v\} \notin \mathcal{E}, \end{cases} \\ w_{u,v} &\coloneqq 2 \cdot \frac{|\{\{u', v'\} \in \mathrm{C}(\mathcal{V}, 2) \mid y_{u',v'} = y_{u,v}\}|}{|\mathrm{C}(\mathcal{V}, 2)|}. \end{aligned} \tag{14}$$

Also, define the 0-1 loss function $l_{0\text{-}1} : \mathbb{R} \times \{-1, +1\} \to \{0, +1\}$ by

$$l_{0\text{-}1}(\hat{y}, y) \coloneqq \begin{cases} 0 & \text{if } \operatorname{sgn}(\hat{y}) = y, \\ +1 & \text{otherwise.} \end{cases} \tag{15}$$

The objective of our experimental setting is to minimize the following balanced 0-1 loss:

$$L_{0\text{-}1}\big((\boldsymbol{z}_v)_{v\in\mathcal{V}}; \mathcal{G}\big) \coloneqq \sum_{\{u,v\}\in \mathrm{C}(\mathcal{V},2)} w_{u,v} l_{0,1}\Big(\big[\Delta_{(\mathbb{D}^D, \boldsymbol{G}^{\mathrm{P}}_\cdot)}(\boldsymbol{z}_u, \boldsymbol{z}_v)\big]^2 - \theta, y_{u,v}\Big), \tag{16}$$

where the hyperparameter $\theta \in \mathbb{R}_{>0}$ determines the threshold in labeling the pair to be positive or negative.

The above function $L_{0\text{-}1}$ is not easy to optimize since it is not continuous. Hence, in the optimization step, we replace $l_{0\text{-}1}$ by the hinge loss function $l_{\text{hinge}} : \mathbb{R} \times \{-1, +1\} \to \mathbb{R}_{\leq 0}$ defined by $l_{\text{hinge}}(\hat{y}, y) = \max -\hat{y}y + 1, 0$, widely used in machine learning area, e.g., support vector machines (Cortes & Vapnik, 1995). That is, the loss function in the optimization step is

$$L\big((\boldsymbol{z}_v)_{v\in\mathcal{V}}; \mathcal{G}\big) \coloneqq \sum_{\{u,v\}\in \mathrm{C}(\mathcal{V},2)} w_{u,v} l\Big(\big[\Delta_{(\mathbb{D}^D, \boldsymbol{G}^{\mathrm{P}}_\cdot)}(\boldsymbol{z}_u, \boldsymbol{z}_v)\big]^2 - \theta, y_{u,v}\Big). \tag{17}$$

Also, we add the regularization term $\lambda \sum_{v\in\mathcal{V}} r(\boldsymbol{z}_v)$ to the objective function, where the regularization function $r : \mathbb{D}^D \to \mathbb{R}_{\geq 0}$ is the object of the comparison in the experiments and varies for each method. Note that $\lambda\mathbb{R}_{\geq 0}$ determines the regularization strength. To wrap up, we optimize the function $J\big((\boldsymbol{z}_v)_{v\in\mathcal{V}}; \mathcal{G}\big) \coloneqq L\big((\boldsymbol{z}_v)_{v\in\mathcal{V}}; \mathcal{G}\big) + \lambda \sum_{v\in\mathcal{V}} r(\boldsymbol{z}_v)$.

As a regularization function r, we compare the following three:

$$r(\boldsymbol{z}) = \begin{cases} \|\boldsymbol{z}\|_{1,(\mathbb{D}^D, \boldsymbol{G}^{\mathrm{P}}_\cdot)} & \textbf{hyperbolic 1-norm (H 1-norm)}, \\ \|\boldsymbol{z}\|_1 & \textbf{linear 1-norm}, \\ 0 & \textbf{no regularization}. \end{cases} \tag{18}$$

We use the HISTA for the hyperbolic 1-norm. For the linear-norm, we apply Riemannian gradient descent and traditional shrinkage-thresholding operator. Note that this is also what we propose for a baseline. For the no regularization case, these two are the same. Strictly speaking, we need to regard the problem as the optimization of a function of the product of $|\mathcal{V}|$ hyperbolic spaces since we consider $|\mathcal{V}|$ points in hyperbolic space. The rigorous discussion for the product manifold is given in Appendix E. Still, it shows that what we need to do is to calculate a partial derivative for each point, convert it into a Riemannian gradient, and apply the HISTA or RGD for each point, like existing papers do.

We evaluate the balanced accuracy $1 - L_{0\text{-}1}\big((\boldsymbol{z}_v)_{v\in\mathcal{V}}; \mathcal{G}\big)$ and the sum $\sum_{v\in\mathcal{V}} \|\boldsymbol{z}_v\|_{0,(\mathbb{D}^D, \boldsymbol{G}^{\mathrm{P}}_\cdot)} = \sum_{v\in\mathcal{V}} \|\boldsymbol{z}_v\|_0$ of the 0-norms of the representations. The higher accuracy and lower 0-norm, the

Figure 5: The datasets' structure. Left: **TRLC**, right: **TRC**.

better, but there is a trade-off between the accuracy and the 0-norm. Specifically, the stronger the regularization is, the lower accuracy and lower 0-norm it gets, and vice versa. Hence, we vary the regularization weight λ and observe how the accuracy and the 0-norm changes. For the no regularization method, we vary D instead of λ. In this case, the lower D is, the lower accuracy and lower 0-norm it gets, and vice versa.

As a graph, we consider tree-like structures that are not completely tree, which are our main focus. To see the difference between the CH 1-norm regularization and Linear 1-norm regularization, we experiment on synthetic datasets defined below.

- TREE-ROOTLEAFCUBES (TRLC) consisting of two complete n-ary trees with height h and five m-dimensional cubes. One cube is in between the roots of the two trees, where each vertex of a hyperbody diagonal pair (a most distant pair) in the cube has an edge to the root of a tree. The other four cubes are connected to a leaf of a tree. Two cubes are connected to one tree and the other two cubes are connected to the other tree. Here, each of the former two cubes have one edge to a leaf of the tree, where the two leaves connected to a cube are most distant to each other. The same holds true for the other tree and the latter two cubes.
- TREE-ROOTCUBES (TRC) is similar to TRLC but without the cube connected to the leaves.

Uniform regularization is needed for TRLC since it has cubes both at the root and around the leaves. Conversely, strong regularization around the boundary and weak regularization could work well for TRC since it has a cube only at the root. Hence, our natural expectation is that CH 1-norm regularization works better for TRLC than Linear 1-norm regularization, but the tendency is not clear for TRC. Figure 5 visualizes these graph structures.

To show our sparse learning scheme's behavior in real applications, we conduct the same experiment on the following real datasets.

- ENRON-EMAIL is an email network reflecting the hierarchical tree structure of the company. At the same time, it also contains edges corresponding to cross-departmental communications, which might be an omen of Enron's bankruptcy in 2001. Since it is a mixture of a tree-like structure and a non-tree-like structure, we expect that our sparse learning scheme works better than the **no regularization** method in the ENRON-EMAIL.
- CORA is a citation network, which shows a highly tree-like structure. Since it is highly tree-like, we expect it to be more effective to apply a (non-sparse) hyperbolic embedding method in low-dimensional space than to apply our sparse learning scheme in high-dimensional space. Hence, we do NOT expect that our sparse learning scheme works so effectively in CORA as in ENRON-EMAIL.

Other experimental settings are as follows. We set $n = 2$, $h = 3$, and $m = 3$. The dimension $D = 6$ is fixed for the two regularization methods, while $D = 2, 3, 4, 5, 6$ for the no regularization method. The regularization strength varies among $\lambda = \{1.0, 2.0, 5.0\} \times 10^{\{-3,-2,-1\}}$ for the two regularization methods. The learning rate that achieved the best accuracy is selected from $\lambda = \{1.0, 2.0, 5.0\} \times 10^{\{1,2,3\}}$. The threshold hyperparameter θ is set to 1.0. The number of iterations is set to $T = 10000$.

Figure 6 shows how the accuracy and the 0-norm changes by varying λ or D. The **closer to the left upper corner**, the better. Here, we show the range where the sum of the 0-norms is no smaller than $2|\mathcal{V}|$; otherwise the mean 0-norm would be lower than two, which would be meaningless as representations. We have also plotted the results of the **H 1-norm** regularization optimized by RGD, which shows that RGD fails to get sparse representations, while shrinkage-thresholding operators succeeded. As we have expected, the **H 1-norm** regularization outperforms the others in TRLC. It shows that our **H 1-norm** regularization can select the dimension of each representations efficiently. In TRC, the linear 1-norm regularization outperforms others around where the sum of the 0-norm is

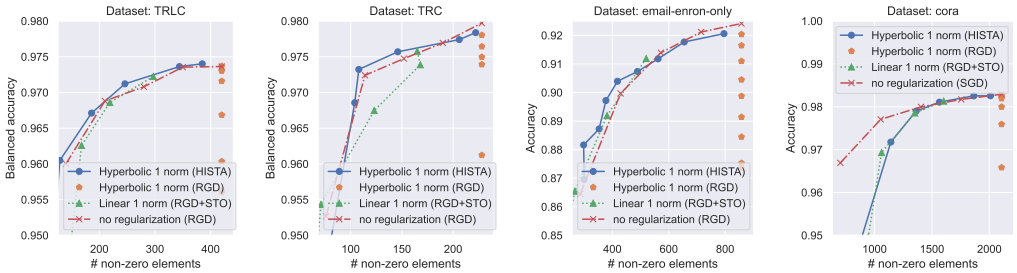

Figure 6: The trade-off between the representation quality (balanced accuracy) and the space complexity (the 0-norm). From left to right: TRLC, TRC, EMAIL-ENRON, **Cora**. The closer to the left upper corner are the graphs, the better.

75, as we have expected. Interestingly, our **H 1-norm** regularization outperforms the **linear 1-norm** regularizations where the sum of the 0-norm is larger. One possible reason is that the **linear 1-norm** regularizations tend to be unstable since the STO changes the representations dramatically around the ball boundary. Although comparing the optimization process is not trivial since they optimize different functions, clarifying the reason for the low performance of the linear regularization would be interesting future work. In ENRON-EMAIL, our **H 1-norm** outperformed other methods both in the balanced accuracy and the sum of 0-norm, when the sum of 0-norm is around 400. While our method's advantage is clear where the sum of 0-norm is small, no method outperforms the remaining in both balanced accuracy and the sum of 0-norm where the sum of 0-norm is large. The investigation of this phenomenon is interesting future work. In CORA, **no regularization** method outperformed both the regularization methods. The reason is that CORA is highly tree-like, so it is most suitable for low-dimensional hyperbolic space. Hence, the result is consistent with our expectations.

H ACRONYM TABLE

To increase readability, we provide the table of acronyms used in this paper in Table 1

RL	representation learning
HSBRL	hyperbolic-space-based representation learning
RGD	Riemannian gradient descent
HISTA	hyperbolic iterative shrinkage-thresholding algorithm
RCS	real coordinate space
CHM	Cartan-Hadamard manifold
CHMOO	CHM with an origin and orthonormal bases
ONB	orthonormal basis
EVCHMOO	Euclidean vector CHMOO
SHP	sparse hyperplane
SSDM	signed SHP distance map
CH	Cartan-Hadamard
H	hyperbolic
ISTA	iterative shrinkage-thresholding algorithm
STO	soft-thresholding operator
CHSTO	Cartan-Hadamard STO
CHISTA	Cartan-Hadamard ISTA

Table 1: Acronyms