Appendices

Appendix .1. Proof for SPTM and M_e boundary

In SPTM, we assume that the original data exhibits a hierarchical structure in Euclidean space. In the subsequent derivation, we posit that lower-level nodes X_t arise from a Gaussian distribution with mean corresponding to the higher-level node X_{t-1}. This concept resembles a diffusion model without a decay coefficient. This assumption is made due to the fact that, in a normal distribution, the probability density is inversely proportional to the square of the Euclidean distance.

$$\boldsymbol{X_1} \sim \mathcal{N}_p(\mathbf{0}, \Sigma_0) \tag{.1}$$

$$\boldsymbol{X_i} \mid \boldsymbol{X_{i-1}} = \boldsymbol{x_{i-1}} \sim \mathcal{N}_p(\boldsymbol{x_{i-1}}, \Sigma_{i-1}) \tag{.2}$$

p is the dimension of the data in Euclidean space. It can be wrote as

$$\boldsymbol{x_t} = \boldsymbol{x_{t-1}} + \boldsymbol{\varepsilon_t} \qquad \boldsymbol{\varepsilon_t} \sim \mathcal{N}_p(\mathbf{0}, \Sigma_{t-1}) \tag{.3}$$

We denote the hierarchy has a depth of t levels, the prior distribution for the nodes in final layer t follows a normal distribution $\mathcal{N}_p(\boldsymbol{\mu}, \Sigma)$. Based on the moment-generating function of the normal distribution and the moment-generating function of the conditional probability, it can be proof that

$$\boldsymbol{X_t} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma) \sim \mathcal{N}_p(\mathbf{0}, \sum_{i=1}^{t} \Sigma_i) \tag{.4}$$

which can be wrote as

$$\boldsymbol{x_t} = \mathbf{0} + \sum_{i=1}^{t} \boldsymbol{\varepsilon_i} \qquad \boldsymbol{\varepsilon_i} \sim \mathcal{N}_p(\mathbf{0}, \Sigma_{i-1}) \tag{.5}$$

$\|\boldsymbol{\varepsilon_i}\|_2$ is also the Euclidean distance between $\boldsymbol{x_i}$ at the ith level and $\boldsymbol{x_{i-1}}$ at the $(i-1)$th level.

For $\boldsymbol{X_t} \sim \mathcal{N}_p(\boldsymbol{\mu}, \Sigma), \boldsymbol{\mu} \neq \mathbf{0}$, $\boldsymbol{\bar{X}_t}$ is the unbiased estimator for $\boldsymbol{\mu}$, so we can just consider $\boldsymbol{X_t} \sim \mathcal{N}_p(\mathbf{0}, \Sigma)$.

Rewrite SPTM with $\boldsymbol{X_t}$,

For $x_t \in \mathbb{R}^p$,

$$\boldsymbol{x'_t} = \left(x'_{t,1}, x'_{t,2}, \ldots, x'_{t,p}, x'_{t,0}\right) = \left(\underbrace{\frac{2r\boldsymbol{x_t}}{1 + \|\boldsymbol{x_t}\|_2^2}}_{\text{dimension-p}}, \frac{-r + r\|\boldsymbol{x_t}\|_2^2}{1 + \|\boldsymbol{x_t}\|_2^2} \right), \qquad \boldsymbol{x'_t} \in \mathbb{S}^n \tag{.6}$$

$$\boldsymbol{x''_t} = \left(x'_t \frac{\sqrt{h^2 - 1}}{r}, h \right) \tag{.7}$$

$$\begin{aligned}
\boldsymbol{x'''_t} &= (x'''_0, x'''_1, \ldots, x'''_n) \\
&= \left(x''_0/(1+h), x''_1/(1+h), \ldots, x''_n/(1+h)\right) \\
&= \boldsymbol{x'_t} \frac{\sqrt{h^2 - 1}}{r(1+h)} \\
&= \left(\boldsymbol{x'_t} \frac{r\sqrt{h^2 - 1}}{r(1+h)(1 + \|\boldsymbol{x_t}\|_2^2)}, \frac{r(-1 + \|\boldsymbol{x_t}\|_2^2)\sqrt{h^2 - 1}}{r(1+h)(1 + \|\boldsymbol{x_t}\|_2^2)} \right) \\
&= \left(\boldsymbol{x'_t} \frac{\sqrt{h^2 - 1}}{(1+h)(1 + \|\boldsymbol{x_t}\|_2^2)}, \frac{(-1 + \|\boldsymbol{x_t}\|_2^2)\sqrt{h^2 - 1}}{(1+h)(1 + \|\boldsymbol{x_t}\|_2^2)} \right)
\end{aligned} \tag{.8}$$

Remember our goal is to have nodes at the root level closer to the center of the Poincaré ball, while nodes at higher levels correspondingly reside closer to the hyperplane where the Poincaré ball is located in the Lorentz model, meaning smaller values of h. In other words, **as i increases in ε_i, h becomes larger,** $h \propto i$.

In our proposed model, our optimization objective aims to minimize the distance between the nodes on the Poincaré ball and their corresponding cluster centers, approximating the distances between nodes and the computed cluster centers on the original Euclidean plane. Next, we will demonstrate that as ε_i increases, where ε_i represents the distance between nodes in our adjacent two layers, and when this distance is not greater than 1.14, the value of h becomes larger. Specifically, h **is proportional to the increase in index** i **when** ε_i<1.14. Notably, **the constraint** ε_i<1.14 **is a very loose constraint and serves as a sufficient but unnecessary condition. In the subsequent content, numerous relaxed inequalities can be observed. This implies that ε_i could potentially be quite large, which also signifies that our approach is feasible yet not unique**.

The equation provided above represents the initial value of h. Therefore, within the same ith level, nodes are assigned the same value of h_i. For each node j, a distinct h_{ij} is obtained during the optimization process.

In the ideal scenario, there would be $d_{poincaré}(\boldsymbol{x_t'''}, \boldsymbol{x_{t-1}'''}) = \|\boldsymbol{x_t} - \boldsymbol{x_{t-1}}\|_2$. Since the curvature c here is actually a parameter that affects similarity transformations, we simplify the calculations by setting it to -1. Then we have

$$\begin{aligned} d_{poincaré}(\boldsymbol{x_t'''}, \boldsymbol{x_{t-1}'''}) &= \operatorname{arcosh}\left(1 + 2\frac{\|\boldsymbol{x_t'''} - \boldsymbol{x_{t-1}'''}\|^2}{\left(1 - \|\boldsymbol{x_t'''}\|^2\right)\left(1 - \|\boldsymbol{x_{t-1}'''}\|^2\right)}\right) \\ &\geq \ln\left(2 + 4\frac{\|\boldsymbol{x_t'''} - \boldsymbol{x_{t-1}'''}\|^2}{\left(1 - \|\boldsymbol{x_t'''}\|^2\right)\left(1 - \|\boldsymbol{x_{t-1}'''}\|^2\right)}\right) \end{aligned} \tag{.9}$$

Then we proceed with a proof by contradiction. We aim to show that $h_i > h_j$. Let's assume $h_i \leq h_j$. Then as $d_{poincaré}(\boldsymbol{x_t'''}, \boldsymbol{x_{t-1}'''}) = \|\boldsymbol{x_t} - \boldsymbol{x_{t-1}}\|_2 = \|\boldsymbol{\varepsilon_t}\|_2$, if

$$\frac{\ln\left(2 + 4\frac{\|\boldsymbol{x_t'''} - \boldsymbol{x_{t-1}'''}\|^2}{(1-\|\boldsymbol{x_t'''}\|^2)\left(1-\|\boldsymbol{x_{t-1}'''}\|^2\right)}\right)}{\|\boldsymbol{\varepsilon_t}\|_2} > 1 \tag{.10}$$

it is confilct, then $h_i > h_j$ is proved.

To show the inequlity (10),

$$
\begin{aligned}
&\|\boldsymbol{x}_{\boldsymbol{t}}''' - \boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2 = \\
&\| \left(\boldsymbol{x}_{\boldsymbol{t}}' \frac{\sqrt{h_i^2-1}}{(1+h_i)(1+\|\boldsymbol{x}_{\boldsymbol{t}}\|_2^2)}, \frac{(-1+\|\boldsymbol{x}_{\boldsymbol{t}}\|_2^2)\sqrt{h_i^2-1}}{(1+h_i)(1+\|\boldsymbol{x}_{\boldsymbol{t}}\|_2^2)} \right) \\
&- \left(\boldsymbol{x}_{\boldsymbol{t-1}}' \frac{\sqrt{h_j^2-1}}{(1+h_j)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)}, \frac{(-1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)\sqrt{h_j^2-1}}{(1+h_j)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)} \right) \| \\
&> \|\boldsymbol{x}_{\boldsymbol{t}}' \frac{\sqrt{h_i^2-1}}{(1+h_i)(1+\|\boldsymbol{x}_{\boldsymbol{t}}\|_2^2)} - \boldsymbol{x}_{\boldsymbol{t-1}}' \frac{\sqrt{h_j^2-1}}{(1+h_j)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)}\| \\
&= \|(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}}) \frac{\sqrt{h_i^2-1}}{(1+h_i)(1+\left\|(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}})\right\|_2^2)} - \boldsymbol{x}_{\boldsymbol{t-1}}' \frac{\sqrt{h_j^2-1}}{(1+h_j)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)}\| \\
&= \frac{2(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}})(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)\sqrt{h_i^2-1} - 2\boldsymbol{x}_{\boldsymbol{t-1}}\sqrt{h_i^2-1}(1+\left\|(\boldsymbol{x}_{\boldsymbol{t-1}}' + \varepsilon_t)\right\|_2^2)}{(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}} + \boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2^2)(1+h_j)} \\
&\geq \frac{\sqrt{h_i^2-1}\left(2(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}})(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2) - 2\boldsymbol{x}_{\boldsymbol{t-1}}(1+\left\|(\boldsymbol{x}_{\boldsymbol{t-1}}' + \varepsilon_t)\right\|_2^2)\right)}{(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}} + \boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2^2)(1+h_j)}
\end{aligned}
\tag{.11}
$$

And we have

$$
\boldsymbol{x}_{\boldsymbol{t}}''' = \frac{h_i - 1}{hi + 1} \tag{.12}
$$

$$
\left(1 - \|\boldsymbol{x}_{\boldsymbol{t}}'''\|^2\right)\left(1 - \|\boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2\right) = \frac{4}{(h_i+1)(h_j+1)} \tag{.13}
$$

Then

$$
\begin{aligned}
&4\frac{\|\boldsymbol{x}_{\boldsymbol{t}}''' - \boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2}{\left(1 - \|\boldsymbol{x}_{\boldsymbol{t}}'''\|^2\right)\left(1 - \|\boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2\right)} \\
&> \frac{\sqrt{h_i^2-1}\left(2(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}})(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2) - 2\boldsymbol{x}_{\boldsymbol{t-1}}(1+\left\|(\boldsymbol{x}_{\boldsymbol{t-1}}' + \boldsymbol{\varepsilon}_{\boldsymbol{t}})\right\|_2^2)(h_i+1)(h_j+1)\right)}{(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2^2)(1+\|\boldsymbol{x}_{\boldsymbol{t-1}} + \boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2^2)(1+h_j)} \\
&\geq \frac{\left\|2\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2\boldsymbol{x}_{\boldsymbol{t-1}} - 2\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2 - 2\boldsymbol{x}_{\boldsymbol{t-1}}\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2)\right\|_2}{\left\|2(1+\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2\boldsymbol{x}_{\boldsymbol{t-1}} - 2\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2\boldsymbol{\varepsilon}_{\boldsymbol{t}} - 2\boldsymbol{x}_{\boldsymbol{t-1}}\|\boldsymbol{x}_{\boldsymbol{t-1}}\|_2)\right\|_2} \\
&= \|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2
\end{aligned}
\tag{.14}
$$

Substitute result (13) to left of (10), we have

$$
\frac{\ln\left(2 + 4\frac{\|\boldsymbol{x}_{\boldsymbol{t}}''' - \boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2}{(1-\|\boldsymbol{x}_{\boldsymbol{t}}'''\|^2)(1-\|\boldsymbol{x}_{\boldsymbol{t-1}}'''\|^2)}\right)}{\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2} > \frac{\ln\left(2 + \|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2\right)}{\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2} \tag{.15}
$$

when $\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2 < 1.146193220620586$, $\frac{\ln\left(2+\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2\right)}{\|\boldsymbol{\varepsilon}_{\boldsymbol{t}}\|_2} > 1$.

Algorithm 2 Stereographic Projection Transition Mapping For NTM

1: **Input**: Train NSTM to obtain topic embeddings X_j and cost matrix M_e.
2: **Output**: Optimized radii r_{topic_j}, r_{word_l}
3: Apply SPTM to map X_j to the Poincaré ball: $X_j''' = (x_0''', x_1''', \ldots, x_n''')$. Apply SPTM to map word embeddings $Y_l = (y_1, \ldots, y_n)$ to the Poincaré ball: $Y_l''' = (y_0''', y_1''', \ldots, y_n''')$.
4: Update X_j''' by normalizing it and multiplying it by initialized radius r_{topic_j}. Update Y_l''' by normalizing it and multiplying it by initialized radius r_{word_l}.
5: Repeat until convergence:

- For each topic j from 1 to k:
  - For each word l from 1 to q:
    - Calculate the Poincaré Disk distance between X_j''' and Y_l''', obtaining $d_{j,l}$, Update entry (j, l) of M_p with $d_{j,l}$.
- Calculate loss: the Mean Squared Error (MSE) between M_e and the calculated matrix M_p.
- Calculate gradient of the loss with respect to r_{topic_j} and r_{word_l}.

Output: Optimized radii r_{topic_j}, r_{word_l}, X_j''', Y_j'''.

Here, only an upper bound for the value of $\|\boldsymbol{\varepsilon_t}\|_2$ is provided without specifying a lower bound. This is because during experimentation, it was observed that excessively large values of $\|\boldsymbol{\varepsilon_t}\|_2$ led to model non-convergence, whereas excessively small values of $\|\boldsymbol{\varepsilon_t}\|_2$ posed no issues. As a result, the investigation here focuses solely on the upper limit of $\|\boldsymbol{\varepsilon_t}\|_2$, while it is hypothesized that the lower limit is 0.

Appendix .2. Experiments on Learning Exponential Mapping with a Three-Layer Fully-Connected Network

As shown in the table below, we generated 100,000 samples in 2 dimensions and applied different curvatures to both exponential mapping and logarithm mapping. Subsequently, we used a three-layer fully connected neural network with ReLU activation to fit these mappings. The hidden dimensions were set to 128, 16, and 2, respectively. The Mean Squared Error (MSE) values were calculated for each experiment.

Dimension	Curvature	Mapping Type	Loss
128	0.0001	Exp	5.75
128	0.0001	Log	5.86
128	0.01	Exp	122.047
128	0.01	Log	0.9410
128	1	Exp	12.20
128	0	Log	0.17
16	1	Exp	12.20
16	1	Log	0.1842
2	1	Exp	7.93
2	1	Log	0.009

Table 1: Loss for different mappings on Poincare disk with varying dimensions and curvatures.

From the results, it can be observed that the MSE values are significantly large for all cases, indicating that a simple neural network is not effective in learning exponential mappings.

Appendix .3. Algorithm of SPTM-TM

	Purity↑			NMI↑		
	WS	20NG	TMN	WS	20NG	TMN
ProdLDA	0.293±0.023	0.417±0.004	0.405±0.157	0.066±0.016	0.321±0.004	0.091±0.101
NVDM	0.367±0.022	0.191±0.000	0.376±0.008	0.110±0.013	0.132±0.006	0.078±0.005
ETM	**0.782**±0.005	0.461±0.002	0.715±0.002	**0.497**±0.003	**0.398**±0.003	**0.383**±0.004
ETM-exp	0.295±0.012	0.170±0.002	0.325±0.006	0.100±0.008	0.120±0.004	0.085±0.003
ETM-constrained	0.380±0.002	0.176±0.004	0.500±0.015	0.197±0.002	0.161±0.003	0.195±0.007
ETM-SPTM	0.754±0.007	**0.490**±0.042	**0.755**±0.028	0.468±0.038	0.376±0.020	0.370±0.004

	topic diversity↑			doc classification acc↑		
	20NG	WS	TMN	20NG	WS	TMN
ProdLDA	0.720±0.003	0.679±0.003	0.557±0.008	0.344±0.005	0.602±0.008	0.617±0.008
NVDM	0.660±0.002	0.481±0.005	0.678±0.008	0.382± 0.002	0.574±0.011	0.591±0.008
ETM	0.782±0.005	0.461±0.002	0.715±0.002	**0.497**±0.003	**0.794**±0.003	**0.715**±0.004
ETM-exp	**0.822**±0.007	0.860±0.003	**0.835**±0.004	0.137±0.003	0.204±0.005	0.207±0.006
ETM-constrained	0.797±0.002	0.845±0.003	0.633±0.002	0.365±0.003	0.716±0.003	0.593±0.002
ETM-SPTM	0.763±0.062	**0.910**±0.011	0.648±0.002	0.495±0.003	0.789±0.010	0.668±0.005

Appendix .4. two-dimensional word embeddings learned by SPTM

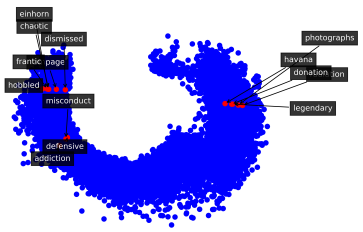

Figure 1: 2D Embedding of TMN

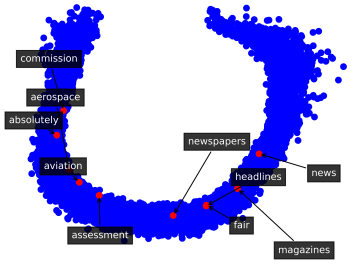

Figure 2: 2D Embedding of Webs

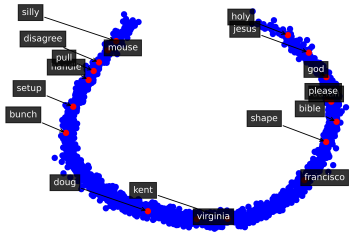

Figure 3: 2D Embedding of 20News

In the TMN image, the top left corner represents the theme of Turmoil/Disorder, while the bottom left corner exhibits various Conflict/Defense themes, and the right side portrays themes leaning towards Charity/Art. In the Webs image, the word distribution related to Aerospace is predominantly on the left side, along with the word distribution related to Fairness. On the other hand, the word distributions of News topics are primarily on the right side. In the 20News image, the Computer-related topic appears in the top left corner, the Religious topic in the top right corner, and the Geography topic at the bottom.

Appendix .5. Purity, NMI, topic diversity, and document classification accuracy for other topic models
ProdLDA and NVDM are not embedding-based topic models, and are only listed here for comparison purposes. When applied to ETM with three different mappings, we observe results similar to those of NSTM.

Table 2: Purity, NMI, topic diversity, and document classification accuracy for document clustering. The symbols "↑" and "↓" indicate "the lower the better" and "the higher the better," respectively. The best result for each dataset is in **bold**. The second result for each dataset is in underline.

Appendix .6. SPTM-VQ-VAE reconstruction samples

In addition, we conducted baseline experiments to evaluate the effectiveness of SPTM when applied to image reconstruction using a Vector Quantised-Variational AutoEncoder (VQVAE). In VQVAE, the codebook vectors play a similar role to topic vectors in topic modeling. We compared the performance of SPTM with vanilla VQVAE method in this setting, demonstrating the potential of SPTM in image reconstruction tasks.

In this study, we conduct an evaluation on three benchmark datasets, namely MNIST, CIFAR-10, and SVHN. Further details about each dataset can be found in the supplementary material, along with implementation specifics and reconstruction samples. We compare our proposed SPTM-VQ-VAE architecture against three baseline models: VAE [30], Poincaré variational auto-encoder (P-VAE) and VQVAE. [31]

P-VAE builds upon the VAE and utilizes the hyperbolic geometry, defining a Riemannian normal distribution in the hyperbolic space as the prior for the VAE. On the other hand, VQ-VAE learns a discrete distribution for the prior and utilizes the copy-gradient technique to back-propagate gradients from decoder to encoder during training. Notably, all three baseline models are self-implemented to ensure a fair comparison.

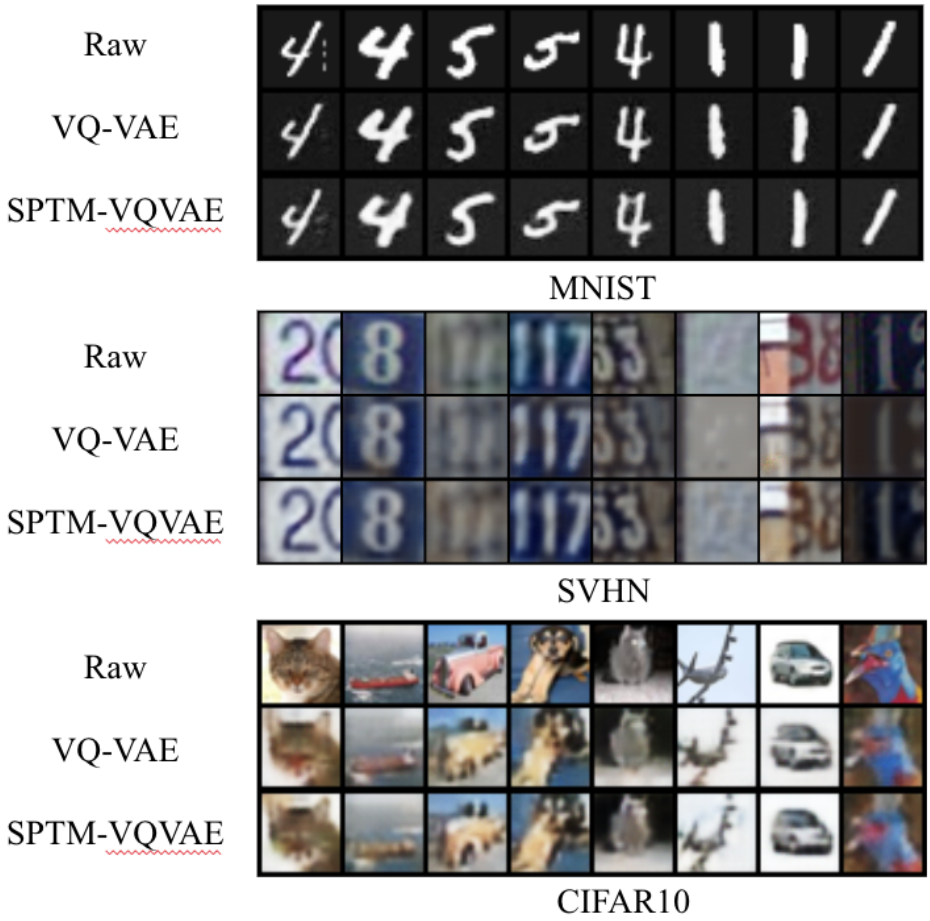

Figure 4: For VQ-VAE and SPTM-VQ-VAE, codebook size is 128, and hidden dimension is 64. For SPTM-VQ-VAE, the curvature is set to -1.