A SAMPLING PROCEDURE

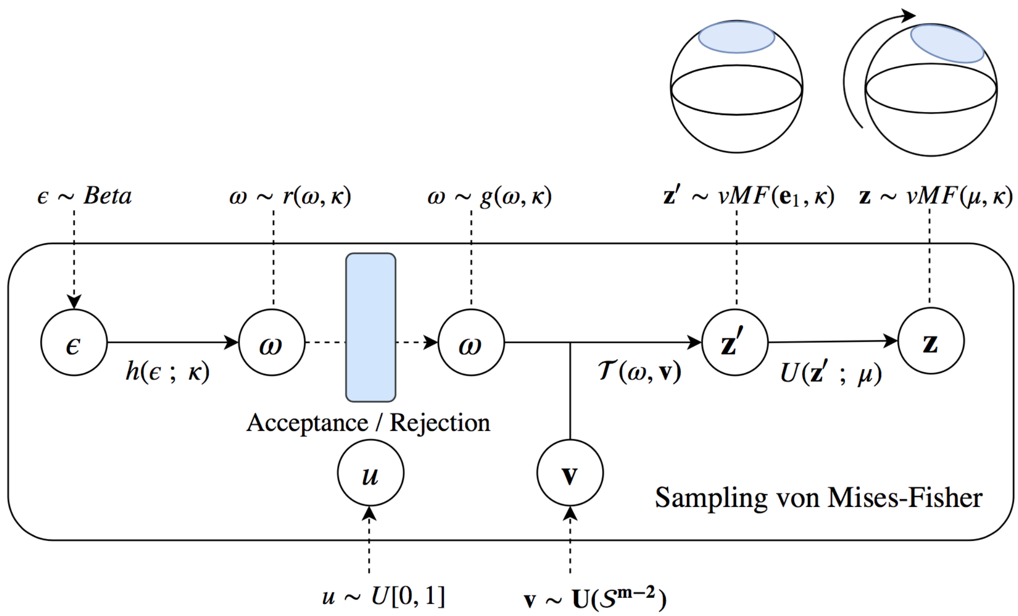


Figure 4: Overview of von Mises-Fisher sampling procedure. Note that as ω is a scalar, the procedure does not suffer from the curse of dimensionality.

The general algorithm for sampling from a vMF has been outlined in Algorithm 1. The exact form of the distribution of the univariate distribution $g(\omega|k)$ is:

$$g(\omega|k) = \frac{2(\pi^{m/2})}{\Gamma(m/2)}\mathcal{C}_m(k)\frac{\exp(\omega k)(1-\omega^2)^{\frac{1}{2}(m-3)}}{B(\frac{1}{2}, \frac{1}{2}(m-1))}, \tag{11}$$

Samples from this distribution are drawn using an acceptance/rejection algorithm when $m \neq 3$. The complete procedure is described in Algorithm 2. The $Householder$ reflection (see Algorithm 3 for details) simply finds an orthonormal transformation that maps the modal vector $\mathbf{e}_1 = (1, 0, \cdots, 0)$ to μ. Since an orthonormal transformation preserves the distances all the points in the hypersphere will stay in the surface after mapping. Notice that even the transform $U\mathbf{z}' = (\mathbb{I} - 2\mathbf{u}\mathbf{u}^\top)\mathbf{z}'$, can be executed in $\mathcal{O}(m)$ by rearranging the terms.

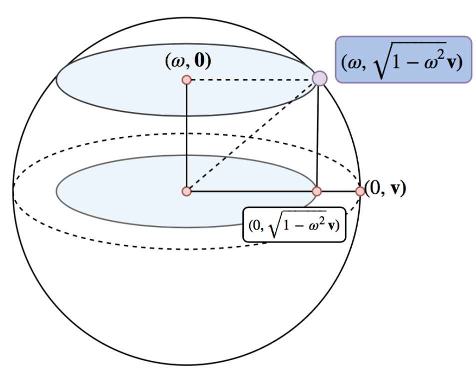


Figure 5: Geometric representation of a single sample in $\mathcal{S}^2$, where $\omega \sim g(\omega|k)$ and $\mathbf{v} \sim U(\mathcal{S}^1)$.

Algorithm 2 $g(\omega|k)$ acceptance-rejection sampling

Input: dimension m, concentration κ

Initialize values:

$b \leftarrow \dfrac{-2k + \sqrt{4k^2 + (m-1)^2}}{m-1}$

$a \leftarrow \dfrac{(m-1) + 2k + \sqrt{4k^2 + (m-1)^2}}{4}$

$d \leftarrow \dfrac{4ab}{(1+b)} - (m-1)\ln(m-1)$

repeat

Sample $\varepsilon \sim \text{Beta}(\frac{1}{2}(m-1), \frac{1}{2}(m-1))$

$\omega \leftarrow h(\varepsilon, k) = \dfrac{1-(1+b)\varepsilon}{1-(1-b)\varepsilon}$

$t \leftarrow \dfrac{2ab}{1-(1-b)\varepsilon}$

Sample $u \sim \mathcal{U}(0,1)$

until $(m-1)\ln(t) - t + d \geq \ln(u)$

Return: ω

Algorithm 3 Householder transform

Input: mean μ, modal vector $\mathbf{e}_1$

$\mathbf{u}' \leftarrow \mathbf{e}_1 - \mu$

$\mathbf{u} \leftarrow \frac{\mathbf{u}'}{||\mathbf{u}'||}$

$U \leftarrow \mathbb{I} - 2\mathbf{u}\mathbf{u}^\top$

Return: U

Table 5: Expected number of samples needed before acceptance, computed using Monte Carlo estimate with 1000 samples varying dimensionality and concentration parameters. Notice that the sampling complexity increases in κ, but decreases as the dimensionality, d, increases.

	$\kappa=1$	$\kappa=5$	$\kappa=10$	$\kappa=50$	$\kappa=100$	$\kappa=500$	$\kappa=1000$	$\kappa=5000$	$\kappa=10000$
$d=5$	1.020	1.171	1.268	1.398	1.397	1.426	1.458	1.416	1.440
$d=10$	1.008	1.094	1.154	1.352	1.411	1.407	1.369	1.402	1.419
$d=20$	1.001	1.031	1.085	1.305	1.342	1.367	1.409	1.410	1.407
$d=40$	1.000	1.011	1.027	1.187	1.288	1.397	1.433	1.402	1.423
$d=100$	1.000	1.000	1.006	1.092	1.163	1.317	1.360	1.398	1.416

B KL DIVERGENCE DERIVATION

The KL divergence between a von-Mises-Fisher distribution $q(\mathbf{z}|\mu, k)$ and an uniform distribution in the hypersphere (one divided by the surface area of $\mathcal{S}^{m-1}$) $p(\mathbf{x}) = \left(\dfrac{2(\pi^{m/2})}{\Gamma(m/2)}\right)^{-1}$ is:

$$\mathcal{KL}[q(\mathbf{z}|\mu,k) \mid\mid p(\mathbf{z})] = \int_{\mathcal{S}^{m-1}} q(\mathbf{z}|\mu,k) \log \frac{q(\mathbf{z}|\mu,k)}{p(\mathbf{z})} d\mathbf{z} \tag{12}$$

$$= \int_{\mathcal{S}^{m-1}} q(\mathbf{z}|\mu,k) \left(\log \mathcal{C}_m(k) + k\mu^T\mathbf{z} - \log p(\mathbf{z})\right) d\mathbf{z} \tag{13}$$

$$= k\mu\, \mathbb{E}_q[\mathbf{z}] + \log \mathcal{C}_m(k) - \log \left(\frac{2(\pi^{m/2})}{\Gamma(m/2)}\right)^{-1} \tag{14}$$

$$= k\frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)} + \left((m/2-1)\log k - (m/2)\log(2\pi) - \log \mathcal{I}_{m/2-1}(k)\right) \tag{15}$$

$$+ \frac{m}{2}\log\pi + \log 2 - \log\Gamma(\frac{m}{2}),$$

B.1 GRADIENT OF KL DIVERGENCE

Using

$$\nabla_k \mathcal{I}_v(k) = \frac{1}{2}\left(\mathcal{I}_{v-1}(k) + \mathcal{I}_{v+1}(k)\right), \tag{16}$$

and

$$\nabla_k \log \mathcal{C}_m(k) = \nabla_k \big((m/2-1)\log k - (m/2)\log(2\pi) - \log \mathcal{I}_{m/2-1}(k)\big) \tag{17}$$

$$= -\frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)}, \tag{18}$$

then

$$\nabla_\kappa \mathcal{KL}[q(\mathbf{z}|\mu,k) \mid\mid p(\mathbf{z})] = \nabla_\kappa k \frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)} + \nabla_k \log \mathcal{C}_m(k) \tag{19}$$

$$= \frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)} + k\nabla_k \frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)} - \frac{\mathcal{I}_{m/2}(k)}{\mathcal{I}_{m/2-1}(k)} \tag{20}$$

$$= \frac{1}{2}k\left(\frac{\mathcal{I}_{m/2+1}(k)}{\mathcal{I}_{m/2-1}(k)} - \frac{\mathcal{I}_{m/2}(k)\left(\mathcal{I}_{m/2-2}(k) + \mathcal{I}_{m/2}(k)\right)}{\mathcal{I}_{m/2-1}(k)^2} + 1\right), \tag{21}$$

Notice that we can use $\mathcal{I}_{m/2}^{exp} = \exp(-k)\mathcal{I}_{m/2}$ for numerical stability.

C PROOF OF LEMMA 2

Lemma 3 (2). *Let f be any measurable function and $\varepsilon \sim \pi_1(\varepsilon|\theta) = s(\varepsilon)\dfrac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}$ the distribution of the accepted sample. Also let $\mathbf{v} \sim \pi_2(\mathbf{v})$, and $\mathcal{T}$ a transformation that depends on the parameters such that if $\mathbf{z} = \mathcal{T}(\omega, \mathbf{v}; \theta)$ with $\omega \sim g(\omega|\theta)$, then $\mathbf{z} \sim q(\mathbf{z}|\theta)$:*

$$\mathbb{E}_{(\varepsilon,\mathbf{v})\sim\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})}\left[f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\right] = \int f(\mathbf{z})q(\mathbf{z}|\theta)d\mathbf{z} = \mathbb{E}_{q(\mathbf{z}|\theta)}[f(\mathbf{z})], \tag{22}$$

Proof.

$$\mathbb{E}_{(\varepsilon,\mathbf{v})\sim\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})}\left[f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\right] = \iint f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})d\varepsilon d\mathbf{v}, \tag{23}$$

Using the same argument employed by Naesseth et al. (2017) we can apply the change of variables $\omega = h(\varepsilon, \theta)$ rewrite the expression as:

$$= \iint f\left(\mathcal{T}(\omega,\mathbf{v};\theta)\right) g(\omega|\theta)\pi_2(\mathbf{v})d\omega d\mathbf{v} =^* \int f(\mathbf{z})q(\mathbf{z}|\theta)d\mathbf{z} \tag{24}$$

Where in * we applied the change of variables $\mathbf{z} = \mathcal{T}(\omega, \mathbf{v}; \theta)$. □

D REPARAMETERIZATION GRADIENT DERIVATION

D.1 GENERAL EXPRESSION DERIVATION

We can then proceed as in 8 using Lemma 2 and the the log derivative trick to compute the gradient of the expectation term $\nabla_\theta \mathbb{E}_{q(\mathbf{z}|\theta)}[f(\mathbf{z})]$:

$$\nabla_\theta \mathbb{E}_{q(\mathbf{z}|\theta)}[f(\mathbf{z})] = \nabla_\theta \iint f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})d\varepsilon d\mathbf{v} \tag{25}$$

$$= \nabla_\theta \iint f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right) s(\varepsilon)\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\pi_2(\mathbf{v})d\varepsilon d\mathbf{v} \tag{26}$$

$$= \iint s(\varepsilon)\pi_2(\mathbf{v})\nabla_\theta \left(f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\right) d\varepsilon d\mathbf{v} \tag{27}$$

$$\begin{aligned} &= \iint s(\varepsilon)\pi_2(\mathbf{v})\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\nabla_\theta\left(f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\right) d\varepsilon d\mathbf{v} \\ &\quad + \iint s(\varepsilon)\pi_2(\mathbf{v}) f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\nabla_\theta \left(\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\right) d\varepsilon d\mathbf{v} \end{aligned} \tag{28}$$

$$\begin{aligned} &= \iint \pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})\nabla_\theta\left(f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\right) d\varepsilon dv \\ &\quad + \iint s(\varepsilon)\pi_2(\mathbf{v}) f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\nabla_\theta \left(\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\right) d\varepsilon d\mathbf{v} \end{aligned} \tag{29}$$

$$\begin{aligned} &= \underbrace{\mathbb{E}_{(\varepsilon,\mathbf{v})\sim\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})}\left[\nabla_\theta f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\right]}_{g_{rep}} \\ &\quad + \underbrace{\mathbb{E}_{(\varepsilon,\mathbf{v})\sim\pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})}\left[f\left(\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\nabla_\theta \log\left(\frac{g(h(\varepsilon,\theta)|\theta)}{r(h(\varepsilon,\theta)|\theta)}\right)\right]}_{g_{cor}}, \end{aligned} \tag{30}$$

where g_{rep} is the reparameterization term and g_{cor} the correction term. Since h is invertible in ε, Naesseth et al. (2017) show that $\nabla_\theta \log \frac{q(h(\varepsilon,\theta),\theta)}{r((h(\varepsilon,\theta),\theta)}$ in g_{cor} simplifies to:

$$\nabla_\theta \log \frac{g(h(\varepsilon,\theta),\theta)}{r((h(\varepsilon,\theta),\theta)} = \nabla_\theta \log g(h(\varepsilon,\theta),\theta) + \nabla_\theta \log |\frac{\partial h(\varepsilon,\theta)}{\partial \varepsilon}|, \tag{31}$$

D.2 GRADIENT CALCULATION

In our specific case we want to take the gradient w.r.t. θ of the expression:

$$\mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{x};\theta)}[\log p_\phi(\mathbf{x}|\mathbf{z})] \quad \text{where } \theta = (\mu,\kappa), \tag{32}$$

The gradient can be computed using the Lemma 2 and the subsequent gradient derivation with $f(\mathbf{z}) = p_\phi(\mathbf{x}|\mathbf{z})$. As specified in Section 3.4 we optimize unbiased Monte Carlo estimates of the gradient. Therefore fixed one datapoint $\mathbf{x}$ and sampled $(\varepsilon,\mathbf{v}) \sim \pi_1(\varepsilon|\theta)\pi_2(\mathbf{v})$ the gradient is:

$$\nabla_\theta \mathbb{E}_{q_\psi(\mathbf{z}|\mathbf{x};\theta)}[\log p_\phi(\mathbf{x}|\mathbf{z})] = g_{rep} + g_{cor}, \tag{33}$$

With

$$g_{rep} \approx \nabla_\theta \log p_\phi\left(\mathbf{x}|\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right), \tag{34}$$

$$g_{cor} \approx p_\phi\left(x|\mathcal{T}(h(\varepsilon,\theta),\mathbf{v};\theta)\right)\left(\nabla_\theta \log g(h(\varepsilon,\theta)|\theta) + \nabla_\theta \log |\frac{\partial h(\varepsilon,\theta)}{\partial\varepsilon}|\right), \tag{35}$$

where g_{rep} is simply the gradient of the reconstruction loss w.r.t θ and can be easily handled by automatic differentiation packages.

For what concerns g_{cor} we notice that the terms $g()$ and $h()$ do not depend on μ. Thus the g_{cor} term w.r.t. μ is 0 an all the following calculations can will be only w.r.t. κ. We therefore have:

$$\frac{\partial h(\varepsilon,k)}{\partial\varepsilon} = \frac{-2b}{((b-1)\varepsilon+1)^2} \quad \text{where } b = \frac{-2k+\sqrt{4k^2+(m-1)^2}}{m-1}, \tag{36}$$

and

$$\nabla_\kappa \log g(\omega|k) = \nabla_\kappa \left(\log \mathcal{C}_m(k) + \omega k + \frac{1}{2}(m-3)\log(1-\omega^2) \right) \tag{37}$$

$$= \nabla_k \log \mathcal{C}_m(k) + \nabla_\kappa \left(\omega k + \frac{1}{2}(m-3)\log(1-\omega^2) \right). \tag{38}$$

So, putting everything together we have:

$$g_{cor} = \log p_\phi(x|z) \cdot \left[-\frac{\mathcal{I}_{m/2}}{\mathcal{I}_{m/2-1}} + \nabla_\kappa \left(\omega\kappa + \frac{1}{2}(m-3)\log(1-\omega^2) + \log\left|\frac{-2b}{((b-1)\varepsilon+1)^2}\right| \right) \right], \tag{39}$$

where

$$b = \frac{-2k + \sqrt{4k^2 + (m-1)^2}}{m-1} \tag{40}$$

$$\omega = h(\varepsilon, \theta) = \frac{1-(1+b)\varepsilon}{1-(1-b)\varepsilon} \tag{41}$$

$$z = \mathcal{T}(h(\varepsilon,\theta), \mathbf{v}; \theta), \tag{42}$$

And the term $\nabla_\kappa \left(\omega\kappa + \frac{1}{2}(m-3)\log(1-\omega^2) + \log\left|\frac{-2b}{((b-1)\varepsilon+1)^2}\right| \right)$ can be computed by automatic differentiation packages.

E COLLAPSE OF THE SURFACE AREA

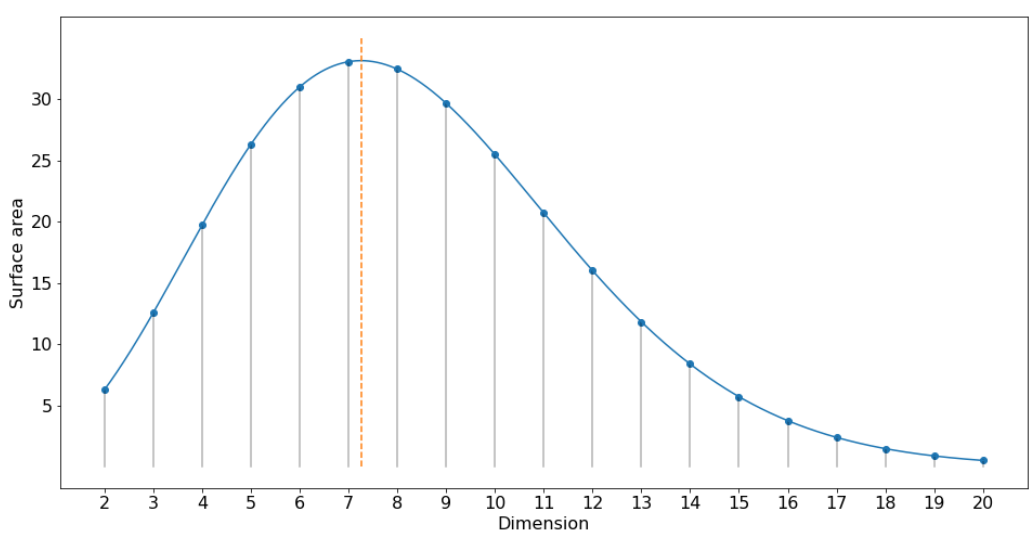


Figure 6: Plot of the unit hyperspherical surface area against dimensionality. The surface area has a maximum for $m = 7$.

F EXPERIMENTAL DETAILS: ARCHITECTURE AND HYPERPARAMETERS

F.1 EXPERIMENT 5.2

Architecture and hyperparameters For both the encoder and the decoder we use MLPs with 2 hidden layers of respectively, [256, 128] and [128, 256] hidden units. We trained until convergence using early-stopping with a look ahead of 50 epochs. We used the Adam optimizer (Kingma and Ba, 2015) with a learning rate of 1e-3, and mini-batches of size 64. Additionally, we used a linear *warm-up* for 100 epochs (Bowman et al., 2016). The weights of the neural network were initialized according to (Glorot and Bengio, 2010).

F.2 EXPERIMENT 5.3

Architecture and Hyperparameters For M1 we reused the trained models of the previous experiment, and used K-nearest neighbors (K-NN) as a classifier with $k = 5$. In the $\mathcal{N}$-VAE case we used the Euclidean distance as a distance metric. For the $\mathcal{S}$-VAE the geodesic distance $\arccos(\mathbf{x}^\top \mathbf{y})$ was employed. The performance was evaluated for $N = [100, 600, 1000]$ observed labels.

The stacked M1+M2 model uses the same architecture as outlined by Kingma et al. (2014), where the MLPs utilized in the generative and inference models are constructed using a single hidden layer, each with 500 hidden units. The latent space dimensionality of $\mathbf{z}_1$, $\mathbf{z}_2$ were both varied in $[5, 10, 50]$. We used the rectified linear unit (ReLU) as an activation function. Training was continued until convergence using early-stopping with a look ahead of 50 epochs on the validation set. We used the Adam optimizer with a learning rate of 1e-3, and mini-batches of size 100. All neural network weight were initialized according to (Glorot and Bengio, 2010). N was set to 100, and the α parameter used to scale the classification loss was chosen between $[0.1, 1.0]$. Crucially, we train this model end-to-end instead of by parts.

F.3 EXPERIMENT 5.4

Architecture and Hyperparameters We are training a Variational Graph Auto-encoder (VGAE) model, a state-of-the-art link prediction model for graphs, as proposed in Kipf and Welling (2016). For a fair comparison, we use the same architecture as in the original paper and we just change the way the latent space is generated using the vMF distribution instead of a normal distribution. All models are trained for 200 epochs on Cora and Citeseer, and 400 epochs on Pubmed with the Adam optimizer. Optimal learning rate $lr \in \{0.01, 0.005, 0.001\}$, dropout rate $p_{do} \in \{0, 0.1, 0.2, 0.3, 0.4\}$ and number of latent dimensions $d_z \in \{8, 16, 32, 64\}$ are determined via grid search based on validation AUC performance. For $\mathcal{S}$-VGAE, we omit the $d_z = 64$ setting as some of our experiments ran out of memory. The model is trained with a single hidden layer with 32 units and with document features as input, as in Kipf and Welling (2016). The weights of the neural network were initialized according to (Glorot and Bengio, 2010). For testing, we report performance of the model selected from the training epoch with highest AUC score on the validation set. Different from (Kipf and Welling, 2016), we train both the $\mathcal{N}$-VGAE and the $\mathcal{S}$-VGAE models using negative sampling in order to speed up training, i.e. for each positive link we sample, uniformly at random, one negative link during every training epoch. All experiments are repeated 5 times, and we report mean and standard error values.

F.3.1 FURTHER EXPERIMENTAL DETAILS

Dataset statistics are summarized in Table 6. Final hyperparameter choices found via grid search on the validation splits are summarized in Table 7.

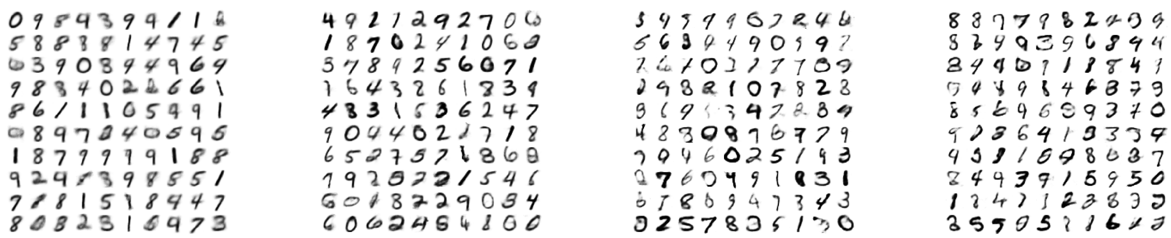
Table 6: Dataset statistics for citation network datasets.

Dataset	Nodes	Edges	Features
Cora	2,708	5,429	1,433
Citeseer	3,327	4,732	3,703
Pubmed	19,717	44,338	500

Table 7: Best hyperparameter settings found for citation network datasets.

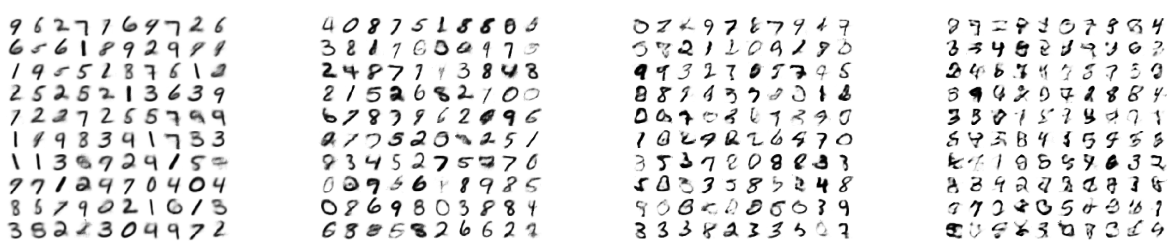
Dataset	Model	lr	p_{do}	d_z
Cora	$\mathcal{N}$-VAE	0.005	0.4	64
	$\mathcal{S}$-VAE	0.001	0.1	32
Citeseer	$\mathcal{N}$-VAE	0.01	0.4	64
	$\mathcal{S}$-VAE	0.005	0.2	32
Pubmed	$\mathcal{N}$-VAE	0.001	0.2	32
	$\mathcal{S}$-VAE	0.01	0.0	32

G VISUALIZATION OF SAMPLES AND LATENT SPACES



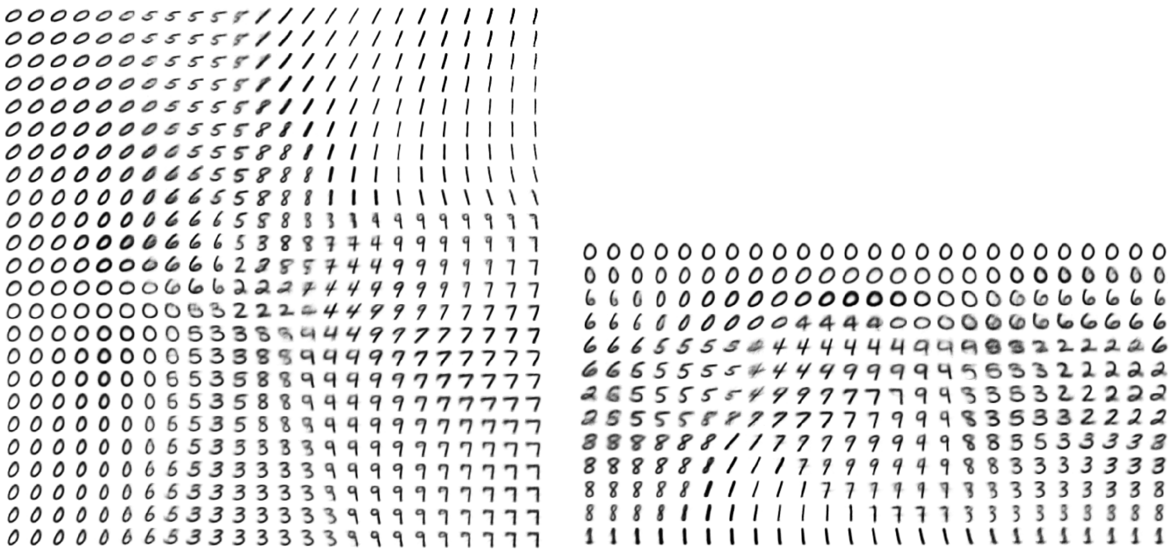
(a) $d = 2$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$

Figure 7: Random samples from $\mathcal{N}$-VAE of MNIST for different dimensionalities of latent space.



(a) $d = 2$ (b) $d = 5$ (c) $d = 10$ (d) $d = 20$

Figure 8: Random samples from $\mathcal{S}$-VAE of MNIST for different dimensionalities of latent space.



(a) $\mathcal{N}$-VAE (b) $\mathcal{S}$-VAE

Figure 9: Visualization of the 2 dimensional manifold of MNIST for both the $\mathcal{N}$-VAE and $\mathcal{S}$-VAE. Notice that the $\mathcal{N}$-VAE has a clear center and all digits are spread around it. Conversely, in the $\mathcal{S}$-VAE instead all digits occupy the entire space and there is a sense of continuity from left to right.

H VISUALIZATION OF CONDITIONAL GENERATION

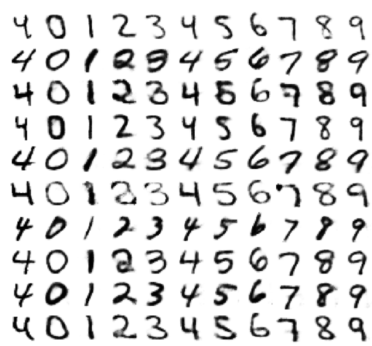


Figure 10: Visualization of handwriting styles learned by the model, using conditional generation on MNIST of M1+M2 with $dim(\mathbf{z}_1) = 50$, $dim(\mathbf{z}_2) = 50$, $\mathcal{S}$+$\mathcal{N}$. Following Kingma et al. (2014), the left most column shows images from the test set. The other columns show analogical fantasies of $\mathbf{x}$ by the generative model, where in each row the latent variable $\mathbf{z}_2$ is set to the value inferred from the test image by the inference network and the class label $\mathbf{y}$ is varied per column.