On the Importance of Looking at the Manifold

Anonymous Author(s)
Affiliation
Address
email

Abstract

Data typically represented in regular domains, such as images, can have a higher level of relational information, either between data samples or even relations within samples. With this perspective data points can be enriched by explicitly accounting for this connectivity. We analyze various approaches for unsupervised representation learning and investigate the importance of considering topological information. We show that each of the representations learned by these models may have critical importance for further downstream tasks, and that accounting for the topological features can improve the modeling capabilities for certain problems.

1 Introduction

It is widely agreed that graphs are the ideal structure to enable relational deep learning [Hamilton et al., 2017]. Prior work has shown that metagraphs incorporating relational information about the dataset can improve unsupervised representation learning in finding less complex models that preserve relational information without loosing representational expressivity [Dumancic and Blockeel, 2017]. In predictive modelling, relational representations can be superior to ordinary ones [Dumancic and Blockeel, 2017, Manica et al., 2019]. In generative tasks, relational distribution comparison was demonstrated to facilitate the learning of generative models across incomparable spaces [Bunne et al., 2019].

Here, we study the impact of the topological information in learning data representations. Specifically, we focus on the trade-off between leveraging data point features and relational information, considering a specturm of models for learning representations. This ranges from Variational Autoencoders [Kingma and Welling, 2013] to node embedding techniques based on random walks on graphs [Grover and Leskovec, 2016], passing through graph neural networks [Veličković et al., 2018] and the proposed Graph-Regularized Variational Autoencoders (GR-VAE), our adaptation of VAEs where the latent space is regularized through a metagraph representing relations between samples of the dataset. The methods considered are evaluated on different datasets and downstream tasks where the impact of the topology can be appropriately assessed. Initially, we examine the impact of implicitly accounting for the topology to validate the GR-VAE in synthetic studies. Thereafter, we move to evaluating all the methods, by comparing performance in downstream tasks based on learned representations in two tasks: text classification and chemical reactions.

2 Methods

In this section we present the different models compared in this study. Our approach is to explore a spectrum of models with varying availability of features and topology (see Figure 1).

2.1 Implicit topological learning

We first explore VAEs [Kingma and Welling, 2013] which only intake features from the nodes, thus serving as a baseline model agnostic to topological information.

Submitted to the Topological Data Analysis and Beyond Workshop at the 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

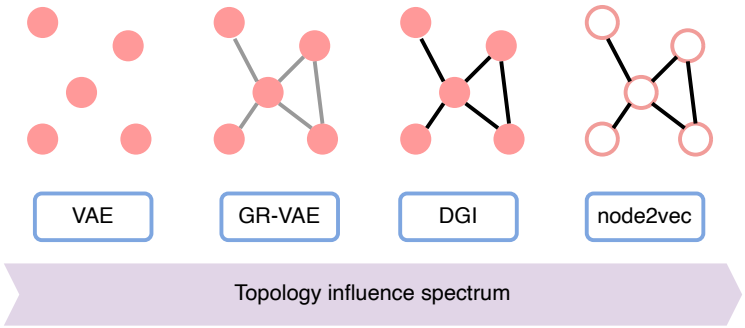

Figure 1: **Topology influence spectrum in the light of the model considered.** From left to right we selected the models in order to smoothly transition from a case where only the point/node features are relevant (left, standard VAE) to the opposite end of the spectrum where only the topological properties are considered (right, node2vec). In the middle we find the cases where the point node features and the topology are blended, either implicitly via a regularizer in the GR-VAE case or explicitly in the DGI case.

Graph-Regularized VAE. We then introduce a variation of VAEs [Kingma and Welling, 2013], defined as Graph-Regularized VAEs (GR-VAE), that augments the trade-off between reconstruction error and KL divergence by a topological constraint. In GR-VAE, the latent space is regularized through a metagraph present in the data. We suggest that accounting for information on how different samples relate (henceforth referred to as graph) may help at obtaining more powerful representations, especially where these relationships are directly involved with a downstream task of interest.

Our approach adds a term to the loss defined by the set of constraints to the samples' representation in the latent space given the distances of the samples' metagraph. For a given set of samples, $\mathbb{S}$, we can compute their distances in the latent space D, as well as over the graph $\mathcal{G}$. For each node, ν, we expect the distances to the other nodes, once embedded in D, to resemble the distances over $\mathcal{G}$. Thus, we enforce a constraint aimed to preserve the relative distances in the two spaces. Formally, fixing a node ν and considering any pair of nodes (i, j), we can define the following penalty term:

$$\phi(d_D, d_{\mathcal{G}}, \nu, i, j) = \begin{cases} (d_D(\nu, j) - d_D(\nu, i))^+ & \text{if } d_{\mathcal{G}}(\nu, i) > d_{\mathcal{G}}(\nu, j) \\ (d_D(\nu, i) - d_D(\nu, j))^2 & \text{if } d_{\mathcal{G}}(\nu, i) = d_{\mathcal{G}}(\nu, j) \\ (d_D(\nu, i) - d_D(\nu, j))^+ & \text{if } d_{\mathcal{G}}(\nu, i) < d_{\mathcal{G}}(\nu, j) \end{cases} \tag{1}$$

where d_D and $d_{\mathcal{G}}$ are metrics defined in the latent space and over the graph respectively. We select L^2 norm as d_D and the geodesic distance [Floyd, 1962] as $d_{\mathcal{G}}$ and then modify the standard VAE loss adding the penalty term computed over the set of samples of interest: $\sum_{\nu \in} \mathbb{S} \sum_{(i,j) \in} \mathbb{S} \times \mathbb{S} \, \phi(d_D, d_{\mathcal{G}}, \nu, i, j)$, and introducing a parmeter $\gamma \geq 0$ regulates the strength of the penalty (see Appendix subsection A.2).

2.2 Explicit topological learning

Notably, GR-VAE is devised to infer topological information solely from a soft constraint, without inductive biases such as graph convolutions. On the other side of the spectrum, graph neural networks (GNN) instead model topology *explicitly*. Here we consider two models from the literature Deep Graph Infomax (DGI) [Veličković et al., 2018] and node2vec [Grover and Leskovec, 2016]. For details see Appendix A.4.

2.3 Datasets

First, for the validation of implicit topological learning, we use a synthetic dataset of point-couds with underlying metagraph connectivity and an extension of MNIST with implicit topology between the labels (connecting them in an chain from 0 to 9). Secondly, we utilized three text datasets involving explicit topological modelling,: Cora, CiteSeer, and PubMed [Sen et al., 2008], and a published chemical representation dataset [Jin et al., 2017] with a compound pair prediction task. For details about the datasets see A.5.

3 Results

We break our experimental results in two parts based on the division previously made beewteen implicit and explicit topological learning (in Section 2).

Implicit topological learning. First, we analyze the validity of implicitly learning the topology through the proposed extended VAE formulation. In this section we present the results on implicit topological learning using VAE and GR-VAE, focusing on MNIST and imposing an artificial, chain-like topology between digits. As we can see in Figure 2, the topology has a stark influence on how the different digits' images organize in the latent space. Interestingly, this behaviour translates into a consistent improvement on the accuracy of downstream tasks directly related to the metagraph as well as in the other metrics considered, namely reconstruction loss and silhouette score (see Table 1) These findings are corroborated when analyzing synthetic datasets and using graph theory algorithms, we can demonstrate that the topology is indeed preserved (see Appendix A.6).

Table 1: **Quantitative results for the MNIST experiment.** We report results for three different models with varying number of dimensions in the latent space: 3, 16, and 64. For each one we explore four training setups, a regular VAE ($\gamma = 0$) and three intensities of GR-VAE ($\gamma \in \{1, 10, 100\}$). We then report the reconstruction loss and the silhouette score of the test samples in the latent space. Furthermore we train two downstream models: k-NN and a classification tree, and we report their average F1 scores over a 5-fold cross-validation. Table A2 adds some extra analysis.

Latent dimensions	**3**				**16**				**64**			
GR γ	**0**	**1**	**10**	**100**	**0**	**1**	**10**	**100**	**0**	**1**	**10**	**100**
Reconstruction loss	**141**	143	147	172	83	**82**	84	105	81	**79**	81	95
Silhouette score	.052	.092	.216	**.195**	.074	.096	.141	**.178**	.055	.060	.112	**.168**
K-NN	.634	.711	.766	**.816**	.928	.933	**.946**	.940	.938	.941	**.947**	.937
Tree	.574	.643	.700	**.753**	.737	.728	.808	**.818**	.692	.725	.777	**.819**

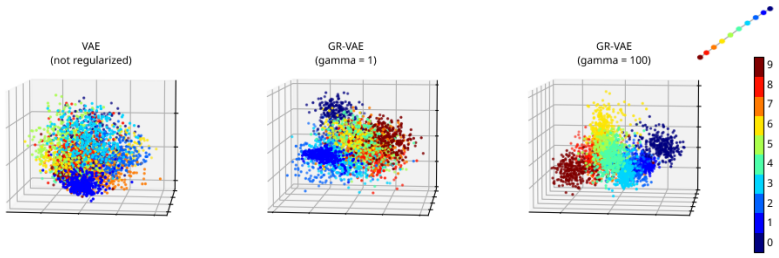

Figure 2: **Qualitative analysis of the latent representations learned in the MNIST case. A.** PCA projection of the samples in the latent space under different training regimes. The original latent space has 16 dimensions. The metagraph is a chain connecting each class from 0 to 9 in order (a representation can be seen on the top right). The samples can be seen coloured by class pertinence. See Figure A3 for more details.

Explicit topological learning. The second set of experiments explores the full topological spectrum, meaning that we account for both implicit and explicit topology on a set of different tasks.

For the text datasets we run all the models with minor adaptations (for details see Appendix A.4). As downstream task we consider the classification of the documents, i.e. nodes, using a logistic regression evaluated on the test set. The splits were reused from Yang et al. [2016]. Our results (see Table 2) show clearly the strong performance of DGI in the three datasets. Interestingly DGI's performance drops when only obtaining batched information. As the authors point out when comparing to GCN, DGI seems to benefit from the fact that it has access to the entire graph [Veličković et al., 2018]. Node2vec outperforms both VAE and GR-VAE in Cora and PubMed, however it falls behind in CiteSeer. We assume that that difference arises due to the relative importance of the graph topology in the different datasets. The relationship between VAE and GR-VAE also reflects this balance. That duality shows how this information may aid in cases where it's more relevant for the downstream task, but it may hinder in cases where the direct link between topology and class (or downstream task) is weaker or straight non-existent.

The results of the experiments run on the chemical reactions dataset can be seen in Table 3. Similarly to the text dataset, using DGI gave the best performance, although the results are more nuanced. The type of encoder seems critical since using an encoder pretrained in a different dataset yielded situations where the DGI performs worse than the VAE encoder alone. The opposite end is when combining all the methods used in the study, where DGI using node embeddings finetuned with a

Table 2: **Results on the text representations.** Accuracy results for the text classification task in the Cora, CiteSeer, and PubMed dataset. In this particular experiment GR-VAE model was trained with equally weighted factors ($\gamma = 1$) of the loss components (reconstruction, KL-divergence and graph regularization).

Model	Cora	CiteSeer	PubMed	Input data
Random	0.152	0.152	0.322	–
VAE	0.530	**0.531**	0.525	V
GR-VAE	0.607	0.492	0.32	V, E
DGI	**0.819**	**0.684**	**0.736**	V, E
DGI (batched training)	**0.738**	**0.611**	**0.722**	V, E
node2vec	**0.719**	0.464	**0.676**	E

Table 3: **Results for chemical reactions experiment.** We report the accuracy on the downstream reaction task. The annotations specify details about the encoder: *Finetuned* denotes that the VAE or GR-VAE has been finetuned on chemical reaction data (on a different split from the downstream reactions), in the case of the DGI the annotation references to which VAE model was used for encoding the SMILES. For each instance of the GR-VAE we display which γ we used in training.

Model	GR γ	Accuracy	Model	GR γ	Accuracy
Random	–	0.5	node2vec	–	0.5
VAE	–	**0.5740**	DGI (VAE)	–	0.5003
VAE (finetuned)	–	0.5613	**DGI (VAE finetuned)**	–	**0.6248**
GR-VAE (finetuned)	0.5	0.5631	**DGI (GR-VAE finetuned)**	**0.5**	**0.6602**
GR-VAE (finetuned)	1	0.5470	DGI (GR-VAE finetuned)	1	0.5617
GR-VAE (finetuned)	2	0.5624	DGI (GR-VAE finetuned)	2	0.5507
GR-VAE (finetuned)	5	0.5543	DGI (GR-VAE finetuned)	5	0.5321

GR-VAE achieves the highest accuracy. It is interesting to observe such a behavior, where we see that among the three top performing models a plain VAE with no topological information is present. This seems to suggest that the quality of the SMILES embedding is key in the task considered.

4 Discussion

Here, we explored the importance of topological information in learning data representations. We demonstrated the addition of inter-sample relational information as a means to improve learned representations, and stressed the trade-off between leveraging sample features and relational information.

We have described a novel loss that expands the VAE by leveraging a relational metagraph and described under which circumstances this added factor becomes a support for further downstream tasks. Most evident are our MNIST results, where adding data that is directly linked to the downstream task of interest creates a more useful arrangement of the latent space, resulting in improvements of the downstream prediction using these embeddings in all the explored setups. It is worth emphasizing that the regularized introduced in the GR-VAE, can not only inject topological awareness into non-topological models, but also be combined with them to achieve superior performance in downstream prediction tasks—as we see in the chemical reaction case. Furthermore, we explore scenarios where the metagraph is less obviously linked to the end prediction. In those, the benefit of adding a graph regularizer (i.e. GR-VAE vs. VAE) is more subtle. Our work opens the door to further exploring ways to evaluate which representation are more useful for given downstream tasks as well as, creating metrics to quantitatively evaluate so. In short, this work aims to be a motivation for looking at the manifold and a small step towards understanding how inter-sample relational information can be beneficial, even in those cases where this data is not explicitly ingested by the model or where the link to a particular end goal may not be obvious.

Statement of Broader Impact

Topology-based representation learning is an exciting field that is still far from its maturity. Nevertheless, understanding the impact of biasing learned representations accounting for relational information, may already help us to extend machine learning applications to unexplored fields, such as polymer biochemistry and green chemistry, that play a pivotal role towards meeting the sustainable development goals (https://www.globalgoals.org/). To increase the impact and the availability of this work we released the source code for the GR-VAE and all the experiments: https://anonymous.4open.science/r/TopoWorkshopNeurIPS_GRVAE_submission/.

Acknowledgments

The authors acknowledge funding from the European Union's Horizon 2020 research and innovation programme.

References

J. Born, M. Manica, A. Oskooei, J. Cadow, and M. R. Martínez. Paccmann rl: Designing anticancer drugs from transcriptomic data via reinforcement learning. In *International Conference on Research in Computational Molecular Biology*, pages 231–233. Springer, 2020.

S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

C. Bunne, D. Alvarez-Melis, A. Krause, and S. Jegelka. Learning generative models across incomparable spaces. In *ICML*, 2019.

S. Dumancic and H. Blockeel. Clustering-based relational unsupervised representation learning with an explicit distributed representation. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

R. W. Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks, 2016.

W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

M. Held and R. M. Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

W. Jin, C. Coley, R. Barzilay, and T. Jaakkola. Predicting organic reaction outcomes with weisfeiler-lehman network. In *Advances in Neural Information Processing Systems*, pages 2607–2616, 2017.

A. Joulin and T. Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Advances in neural information processing systems*, pages 190–198, 2015.

S. Kim, P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, et al. Pubchem substance and compound databases. *Nucleic acids research*, 44(D1): D1202–D1213, 2015.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

M. Manica, J. Cadow, R. Mathis, and M. R. Martínez. Pimkl: Pathway-induced multiple kernel learning. *NPJ Systems Biology and Applications*, 5(1):1–8, 2019.

M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

P. Schwaller, T. Laino, T. Gaudin, P. Bolgar, C. A. Hunter, C. Bekas, and A. A. Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 5(9):1572–1583, 2019.

P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm. Deep graph infomax, 2018.

D. Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

R. J. Williams and D. Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

Z. Yang, W. Cohen, and R. Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International conference on machine learning*, pages 40–48. PMLR, 2016.

A Appendix

A.1 Graph Regularized VAE details

In Figure A1 a detailed description of the GR-VAE architecture is depicted.

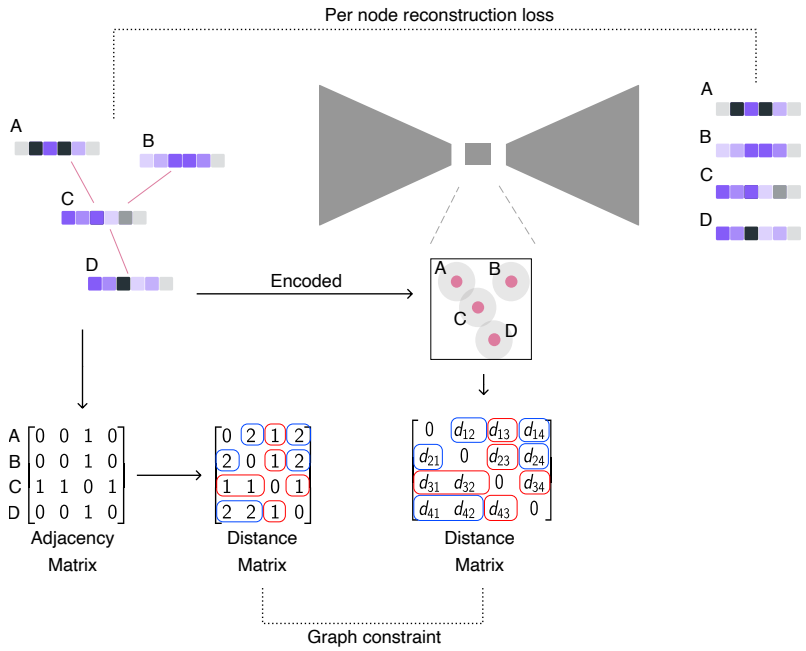

Figure A1: **Complete overview of the GR-VAE approach.** Notice that, the notation $d_{\nu i}$ in the distance matrix abbreviates $d_D(\nu, i)$ from Equation 1.

A.2 GR-VAE loss

The overall loss function of the GR-VAE thus becomes:

$$\mathcal{L}_{\text{GR-VAE}}(\boldsymbol{X};\boldsymbol{\theta}) = \mathcal{L}_{\text{VAE}}(\boldsymbol{X};\boldsymbol{\theta}) + \gamma \sum_{\nu \in \mathbb{S}} \sum_{(i,j) \in \mathbb{S} \times \mathbb{S}} \phi(d_D, d_{\mathcal{G}}, \nu, i, j) \tag{2}$$

where $\boldsymbol{X}$ are the features of the samples in $\mathbb{S}$, $\boldsymbol{\theta}$ the network parameters and $\gamma \geq 0$ regulates the strength of the penalty.

A.3 SMILES embedding

The SMILES VAE used for the chemical reaction dataset was implemented following the description in [Born et al., 2020]. It consists of two layers of stack-augmented GRUs [Joulin and Mikolov, 2015] in both encooder and decoder and is trained with teacher forcing [Williams and Zipser, 1989], token dropout [Bowman et al., 2015] and one-hot encodings.

The dataset consisted of 500,000 molecules represented as canonical SMILES strings from PubChem [Kim et al., 2015].

A.4 Explicit topological learning models

Deep Graph Infomax. Here, we consider to a Deep Graph Infomax (DGI), a state-of-the-art GNN for unsupervised representation learning [Veličković et al., 2018]. DGI relies on maximizing mutual information between subgraphs (themselves derived with GCNs) yielding representations that facilitate downstream node-wise classification tasks.

node2vec. Finally, we utilize node2vec [Grover and Leskovec, 2016], which only consumes topological information but no node-specific features. The node2vec algorithm learns a compressed feature space that maximizes the probability to preserve local neighborhoods. With the exception of node2vec, the specific details for the configuration of each model will depend on the dataset we are evaluating on, thus will be detailed in each of the datasets' results.

A.5 Detailed dataset description

A.5.1 Synthetic data: a qualitative assesment

First, we consider a synthetic dataset with arbitrarily generated graphs on a plane. Each node's features will be composed by the combination of the first two edges directions' (in the case of nodes with a single edge the feature vector is padded with zeros), resulting in a feature vector of 4 dimensions. Thus, each node holds partial, yet insufficient topological information about the graph. As described above, the entire graph is then used to regularize the latent space.

A.5.2 MNIST

On a similar line we expanded this experiments by taking MNIST [LeCun et al., 2010] and generating a topology across the different labels by chaining the samples from 0 all the way to 9. We use this dataset to further test the model's capability of affecting the topology of the latent where the individual node features are of higher complexity, at least when compared to the synthetic data, while maintaining comparable reconstruction performance to the non-constrained scenario.

A.5.3 Text representation

We evaluate three classification datasets: Cora, CiteSeer, and PubMed [Sen et al., 2008]. These datasets contain networks of documents linked by the citation links between documents. The text of the document is represented as a bag-of-words, which we take as a feature vector for each of the documents. Furthermore, each document corresponds to a particular task. We divide each dataset, and use a part of it to train the embedding and the other part on a downstream class prediction task, using the embedding model mentioned above.

A.5.4 Chemical reaction representation

Finally, we analyze the influence of the topology in learning effective representations for molecules in the context of chemical reactions, a topic that has testified a surge in popularity in the recent past as a field for deep learning applications [Schwaller et al., 2019]. To this end, we adopt the dataset compiled by Jin et al. [2017] where we represent reagents, reactants and products using SMILES representations [Weininger, 1988], using the splits provided. For each molecule we extract features using the encoder of a VAE based on stack-augmented GRU layers [Joulin and Mikolov, 2015], as proposed in Born et al. [2020], pretrained on PubChem [Kim et al., 2015] (more details can be found in the Appendix A.3). As for the topological reaction representation we consider a bipartite graph connecting the products to all the reactants and reagents. Each reaction bipartite graph is then used to generate the resulting final graph connecting all the nodes that are shared between different reactions. Using the training split provided by Jin et al. [2017], the models are finetuned as follows: VAE at molecule level, GR-VAE and DGI at reaction level (GR-VAE in an implicit form through the loss regularizer), node2vec on the aggregated graph. Furthermore, DGI uses the different VAEs and GR-VAEs as part of its encoder.

To evaluate the quality of the representations learned and the impact of the topology, we consider the task of predicting whether two molecules are respectively reactant/reagent and products of a valid chemical reaction. The resulting binary classification task has an inherent relation with the underlying reaction network. For VAE, GR-VAE and node2vec we represent a pair of molecules as the concatenation of the encoded molecules/nodes in the respective latent spaces. In the DGI case, we represent the pair as the embedding of a graph connecting the molecules. These representations are then trained on the validation split and later evaluated on the test split as defined by Jin et al. [2017].

A.6 Synthetic data implicit learning results

Here we show the extended results for the implicit learning tasks (i.e. synthetic and MNIST datasets). Figure A2 shows the results of two different graph configurations of point clouds. Figure A3, extends the results shown in Figure 2 by displaying a sample of the original samples and their reconstruction, and the distance matrix between the different centroids for the points of each class.

Figure A4 shows expanded results for a setup where the VAE was mapping to a latent space of 3 dimensions. In that case we can see that with a strong regularizer we still accomplish our desired objective of organizing the point clouds as a chain. This setup, with only 3 dimensions where to map the points, challenges the model and makes it more difficult to obtain reconstructions as faithful to

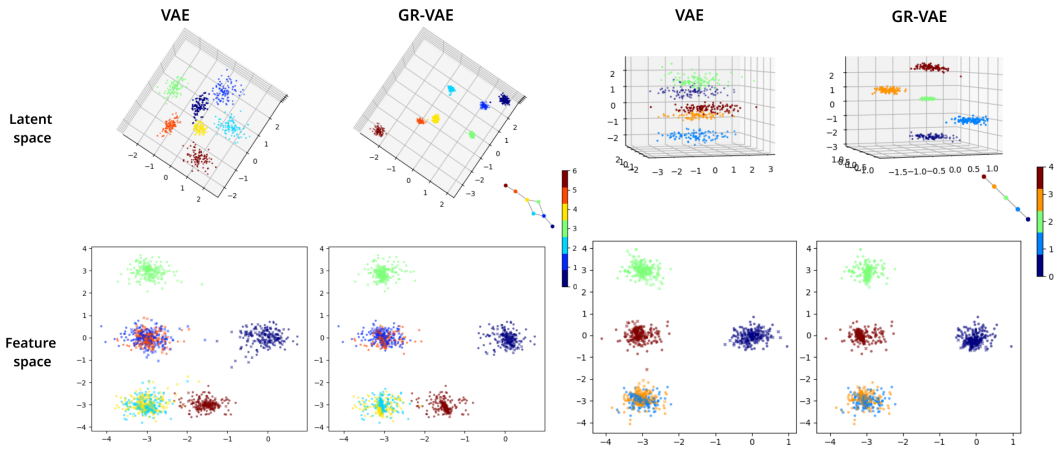

Figure A2: **Learned representations on two synthetic datasets with different topology.** This figure displays qualitatively how GR-VAE affects the latent space topology under different conditions, specifically when compared to its non-constrained counterpart. The plots on top show the latent space, the ones in the bottom show the feature space, for the first two features. The two datasets (left and right) had different topologies, shown as a graph, next the color bar (all colors across plots correspond to the nodes' ids). On the features plots both the original data (round marker) and the reconstructed data (cross marker) are shown.

Table A1: **Shortest Hamiltonian Path (SHP) distance to an ordered chain.** This table displays the distance from each SHP to an ordered chain (0 to 9) using FastDTW. For reference, the average distance of a random connected path is 30.03 ± 6.5 (computed with 1000 random sequences).

	Latent space dimensions		
	3	**16**	**64**
–	23	23	27
1	31	31	27
10	0	31	3
100	0	0	0

the original images as those we saw with models with more dimensions (Figure A3). However it comes useful to display how the embeddings done using the graph regularizer can help at creating clear distinctions between sample groups. For instance, the non-regularized VAE mixes a number of digits (see Figure A4B), while the models that were regularized manage to reconstruct the same digit (i.e. class), usually at the expense of generating reconstructions that are less faithful to the original image in term of details or style

To validate the qualitative assessment on the model's ability to restore the original chain as shown in Figure A3, we computed the Shortest Hamiltonian Paths (SHP) [Held and Karp, 1962] on a fully connected graph of 10 nodes (representing the centroids of the labels in the latent space) where the network topology (i.e. edge weights) was given by the pairwise distances of the centroids. If the SHP of such a graph is a chain from 0 to 9 it proves that the topology is preserved perfectly in the latent space. To compare the different chains we used Dynamic Time Warping Müller [2007], a distance measure based on time series alignment computed with `FastDTW`[1]. An optimal topology corresponds to a DTW distance of 0. The results for the later can be seen in Table A1, the full chains can be seen in Table A2.

[1] https://github.com/slaypni/fastdtw

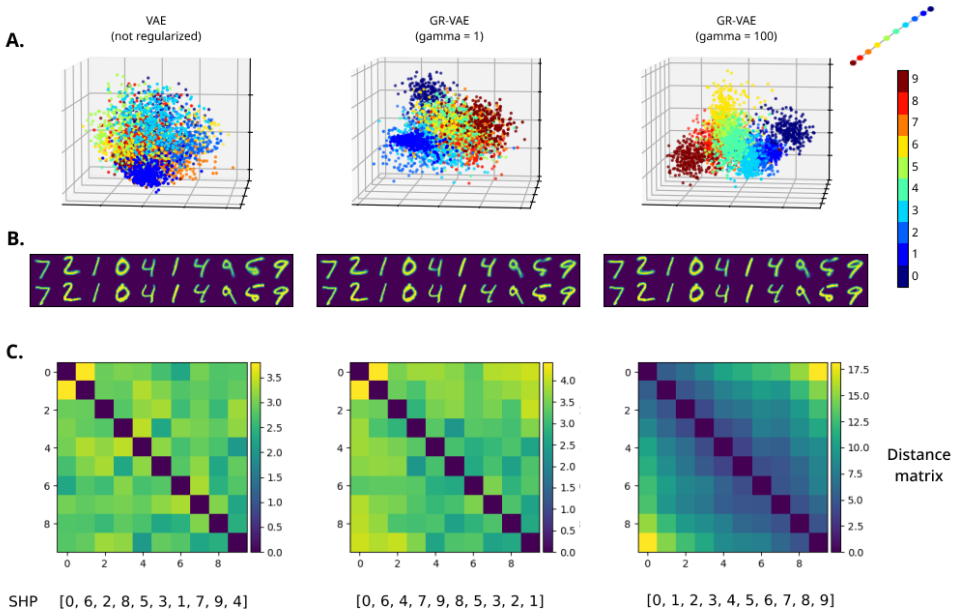

Figure A3: **Qualitative analysis of the latent representations learned in the MNIST case.** Figure with extended information about the MNIST results. **A.** PCA projection of the samples in the latent space under different training regimes. The original latent space has 16 dimensions. The metagraph is a chain connecting each class from 0 to 9 in order (a representation can be seen on the top right). The samples can be seen coloured by class pertinence. **B.** Display of a reduced set of the original samples (bottom row) and their reconstructions (top row). These were taken from the test set. **C.** Distance matrix between the centroids of each label's point cloud. The shortest path Hamiltonian, computed using the centroids, is displayed at the bottom (0 was always used as the starting node).

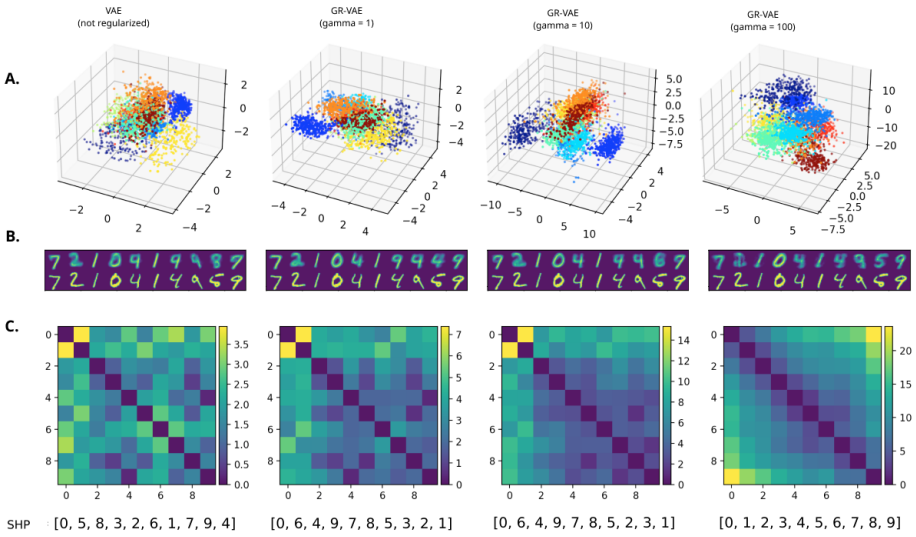

Figure A4: **Qualitative analysis of the latent representations learned in the MNIST case (with a latent space size 3).** Figure with extended information about the MNIST results, it displays same set of experiments run in Figure A3, but using a VAE with a latent space of 3 dimensions. For that reason **A.** directly displays all the latent space dimensions (not a PCA projection). It also includes an extra setting ($\gamma = 10$) for the regularizer.

Table A2: **Shortest Hamiltonian Paths**. Full chains obtained when running SHP over the class centroids of the samples in the latent space. We can see that with the biggest value of the regularizer ($\gamma = 100$) SHPs recover the original chain used for the constraint.

	Latent space dimensions		
γ	**3**	**16**	**64**
no regularizer	0 5 8 3 2 6 1 7 9 4	0 6 2 8 5 3 1 7 9 4	0 6 4 9 7 1 3 5 8 2
1	0 6 4 9 7 8 5 3 2 1	0 6 4 7 9 8 5 3 2 1	0 6 4 9 7 1 3 5 8 2
10	0 6 4 9 7 8 5 2 3 1	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 8 9 7
100	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9