Deep Goal-Oriented Clustering

Anonymous Author
Anonymous Institution

Abstract

Clustering and prediction are two primary tasks in the fields of unsupervised and supervised machine learning. Although much of the recent advances in machine learning have been centered around those two tasks, the interdependent, mutually beneficial relationship between them is rarely explored. E.g., one could reasonably expect a better prediction performance for the downstream task could inform a more appropriate clustering strategy. To this end, we introduce Deep Goal-Oriented Clustering (`DGC`). `DGC` is built upon a variational autoencoder with the latent prior being a Gaussian mixture distribution, thus a probabilistic framework that clusters the data by jointly using supervision via *side-information* and unsupervised modeling of the inherent data structure in an end-to-end fashion. We show the effectiveness of our model on a range of datasets by achieving good prediction accuracies on the side-information, while, more importantly in our setting, simultaneously learning congruent clustering strategies that are on par with the state-of-the-art. We also apply `DGC` to a real-world breast cancer dataset, and show that the discovered clusters carry clinical significance.

1 Introduction

Many of the advances in supervised learning in the past decade are due to the development of deep neural networks (DNN), a class of hierarchical function approximators that are capable of learning complex input-output relationships. Prime examples of such advances include image recognition (Krizhevsky et al., 2012), speech recognition (Nassif et al., 2019), and neural translation (Bahdanau et al., 2015). However, with the explosion of the size of modern datasets, it becomes increasingly unrealistic to manually annotate all available data for training. Hence, understanding inherent data structure through unsupervised clustering is of increasing importance.

Applying DNNs to unsupervised clustering has been studied in the past few years (Caron et al., 2018; Law et al., 2017; Shaham et al., 2018; Tsai et al., 2021), centering around the concept that the input space in which traditional clustering algorithms operate is of importance. Hence, learning this space from data is desirable. Despite the improvements these approaches have made on benchmark clustering datasets, the ill-defined, ambiguous nature of clustering remains a challenge. Such ambiguity is particularly problematic in scientific discovery, sometimes requiring researchers to choose from different, but potentially equally meaningful clustering results when little information is available a priori (Ronan et al., 2016).

When facing such ambiguity, using side-information to reduce clustering ambivalence proves to be a fruitful direction (Xing et al., 2002; Khashabi et al., 2015; Jin et al., 2013). In general, side-information can be categorized as direct or indirect with respect to the final clustering task. Direct side-information straightforwardly details how the data samples should be clustered, and is usually available in terms of constraints, such as the *must-link* and the *cannot-link* constraints (Wang & Davidson, 2010; Wagstaff & Cardie, 2000), or via a pre-conceived notion of similarity (Xing et al., 2002). However, such direct information is rarely available in reality a priori and might not exist in abundance. By comparison, indirect side-information usually carries informative signals on the clustering task and might exist in abundance, but its relation to the clustering task needs to be learned and thus cannot be directly utilized. In this work, we design a framework that learns from such indirect side-information and seamlessly incorporate the learned knowledge into the final clustering task.

Preliminary work. Under review by AISTATS 2022. Do not distribute.

Main Contributions We propose *Deep Goal-Oriented Clustering* (DGC), a probabilistic model that incorporates indirect, informative, side-information when forming a pertinent clustering strategy. Specifically: 1) We combine supervision via side-information and unsupervised data structure modeling in a probabilistic manner; 2) We make minimal assumptions on what form the supervised side-information might take, and assume no explicit correspondence between the side-information and the clusters; 3) We train DGC end-to-end so that the model simultaneously learns from the available side-information while forming a desirable clustering strategy.

2 Related Work

Most related work in the literature can be classified into two categories: 1) Methods that utilize extra side-information to form better, less ambiguous clusters but require specific forms (e.g., pairwise constraints); 2) Methods that can learn from provided labels to lessen the ambiguity in the formed clusters, but rely on the *cluster assumption* (detailed below), and usually assume that the provided labels are discrete and the *ground truth labels*. This excludes the possibility of learning from indirectly related, but still informative side-information.

Side-information for clustering Using side-information to form better clusters has been studied. Wagstaff & Cardie (2000) considered both must-link and cannot-link constraints in the context of K-means clustering. Motivated by image segmentation, Orbanz & Buhmann (2007) proposed a probabilistic model that can incorporate must-link constraints. Khashabi et al. (2015) proposed a nonparametric Bayesian hierarchical model to incorporate noisy side-information as soft-constraints. Vu et al. (2019) utilized constraints and cluster labels as side information. Mazumdar & Saha (2017) gave complexity bounds when provided with an oracle that can be queried for side information. Wasid & Ali (2019) incorporated side information through the use of fuzzy sets. In supervised clustering, the side-information is the a priori known complete clustering for the training set, which is being used as a constraint to learn a mapping between the data and the given clustering (Finley & Joachims, 2005). In contrast, we do not assume that the constraints are given a priori. Instead, we let the side-information guide the clustering procedure during the training process.

The *cluster assumption* If there exists a set of semantic labels associated with the data (e.g. the digit information for MNIST images), the *cluster assumption* states that there exists a direct correspondence between the labels and clusters (Färber et al., 2010; Chapelle et al., 2006). This is high restrictive, especially in the case of utilizing indirect side-information where the information is informative but does not dictate the clustering process. As an example, Kingma et al. (2014) introduced a hierarchical generative model with two variational layers. Originally meant for semi-supervised classification tasks, it can also be used for clustering, in which case all labels are treated as missing since they are the cluster indices. This implies it has to strictly rely on the *cluster assumption*. We show that this approach is a special case of our framework without the probabilistic ensemble component (see Sec. 4.2). Sansone et al. (2016) proposed to address the stringent cluster assumption by modeling the cluster indices and the class labels separately, underscoring the possibility that each cluster may consist of multiple class labels. Deploying a mixture of factor analysers as the underlying probabilistic framework, they also used a variational approximation to maximize the joint log-likelihood. However, their framework cannot be easily scaled to large, modern datasets.

Joint modeling Blei & McAuliffe (2007) incorporated supervision into the latent Dirichlet allocation (LDA) model for document classification. Le et al. (2018) showed that an autoencoder that jointly predicts the targets and the inputs improves performance. Xie & Ma (2019) jointly modeled the reconstruction of a sentence pair and the prediction of the pair's similarity in a VAE (Kingma & Welling, 2014) framework. In this work, we extend the joint modeling literature to clustering and to challenge the commonly assumed cluster assumption.

3 Background & Problem Setup

3.1 Background—Variational Deep Embedding

The starting point for DGC is the *variational auto-encoder* (VAE) (Kingma & Welling, 2014) with the prior distribution of the latent code chosen as a Gaussian mixture distribution, which is introduced in Jiang et al. (2017) as VaDE. We briefly review the generative VaDE approach here to provide the background for DGC. We adopt the notation that lower case letters denote samples from their corresponding distributions; bold, lower case letters denote random variables/vectors; and bold upper case letters denote random matrices.

Assume the prior distribution of the latent code, $\mathbf{z}$, belongs to the family of Gaussian mixture distributions, i.e., $p(\mathbf{z}) = \sum_c p(\mathbf{z}|c)p(c) = \sum_c \pi_c \mathcal{N}(\mu_c, \sigma_c^2 \mathbf{I})$ where c is a random variable, with prior probability π_c, indexing the normal component $p(\mathbf{z}|c)$ that is assumed to be a normal distribution with mean μ_c and variance σ_c^2. VaDE allows for the clustering of the input data in the latent space, with each component of the Gaussian

mixture prior representing an underlying cluster. A VAE-based model can be efficiently described in terms of its generative process and inference procedure. Given an input $\mathbf{x} \in R^d$, the following decomposition of the joint probability $p(\mathbf{x}, \mathbf{z}, c)$ details VaDE's generative process: $p(\mathbf{x}, \mathbf{z}, c) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|c)p(c)$. In words, we sample the component index c from a prior categorical distribution $p(c)$, then sample the latent code $\mathbf{z}$ from the component $p(\mathbf{z}|c)$, and lastly reconstruct the input $\mathbf{x}$ through the reconstruction network $p(\mathbf{x}|\mathbf{z})$. To perform inference and learn from the data, VaDE is constructed to maximize the log-likelihood of the input data $\mathbf{x}$ by maximizing its *evidence lower bound* (ELBO):

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - \mathbb{E}_{q(c|\mathbf{x})} \log \frac{q(c|\mathbf{x})}{p(c)} - \mathbb{E}_{q(\mathbf{z},c|\mathbf{x})} \log \frac{q(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|c)} \tag{1}$$

where, given the input $\mathbf{x}$, $q(\mathbf{z}, c|\mathbf{x})$ denotes the variational posterior distribution over the latent variables, and $\mathbb{E}_d$ denotes the expectation wrt. *distribution* d. With proper assumptions on the prior and variational posterior distributions, the ELBO in Eq. 1 admits a closed-form expression in terms of the parameters of those distributions. We refer readers to Jiang et al. (2017) for additional details.

3.2 Problem Setup

We assume we have a response variable $\mathbf{y}$, and our goal is to leverage $\mathbf{y}$ to inform a better clustering strategy. Abstractly, given the input-output random variable pair $(\mathbf{x}, \mathbf{y})$, we seek to divide the probability space of $\mathbf{x}$ into non-overlapping subspaces that are meaningful in explaining the output $\mathbf{y}$.

In other words, we want to use the prediction task of mapping data points, x, sampled from the probability space of $\mathbf{x}$ to their corresponding sampled outcomes y as a *teaching agent*, to guide the process of dividing the probability space of $\mathbf{x}$ into subspaces that optimally explain y. Since our goal is to discover the subspace-structure without knowing a priori whether such a structure indeed exists, a probabilistic framework is more appropriate due to its ability to incorporate and reason with uncertainty. To this end, we use and extend the VaDE framework, with the following assumption imposed on the latent code that specifically caters to our setting. Assume the input $\mathbf{x}$ carries predictive information with respect to the output $\mathbf{y}$. Since the latent code $\mathbf{z}$ should inherit sufficient information from which the input $\mathbf{x}$ can be reconstructed, it is reasonable to assume that $\mathbf{z}$ also inherits that predictive information. This assumption implies that $\mathbf{x}$ and $\mathbf{y}$ are conditionally independent given $\mathbf{z}$, i.e., $p(\mathbf{x}, \mathbf{y}|\mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{y}|\mathbf{z})$.

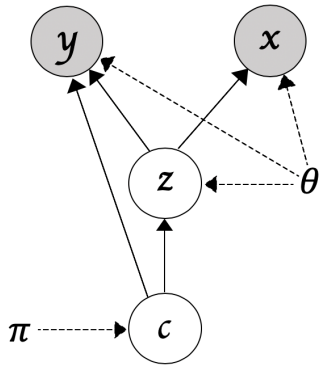


Figure 1: The Bayesian network that underlies the generative process of DGC. θ and π together constitute the generative parameters.

4 Deep Goal-Oriented Clustering

4.1 Generative Process

In order to incorporate $\mathbf{y}$ into a probabilistic model, recall from our previous discussion that $\mathbf{y}$ might manifest with respect to the input differently across different subspaces of the input space. Viewing $p(\mathbf{y}|\mathbf{z})$ as a sample from the space of probability distributions over $\mathbf{y}$ resulting from a functional transformation from $\mathbf{z}$ to this space, we assume that the ground truth transformation function, g_c, is different for each subspace indexed by c. If $\mathbf{z} \sim p(\mathbf{z}|c)$ for some index c, we assume $p(\mathbf{y}|\mathbf{z}, c) \propto g_c(\mathbf{z})$. As a result, we learn a different mapping function for each subspace.

The overall generative process of our model is as follows: **1**. Generate $c \sim \text{Cat}(\pi)$; **2**. Generate $z \sim p(\mathbf{z}|c)$; **3**. Generate $x \sim p(\mathbf{x}|\mathbf{z})$; **4**. Generate $y \sim p(\mathbf{y}|\mathbf{z}, c)$. The Bayesian network that underlies DGC is shown in Fig. 1, and the joint distribution of $\mathbf{x}, \mathbf{y}, \mathbf{z}$, and c can be decomposed as: $p(\mathbf{x}, \mathbf{y}, \mathbf{z}, c) = p(\mathbf{y}|\mathbf{z}, c)p(\mathbf{x}|\mathbf{z})p(\mathbf{z}|c)p(c)$.

4.2 Inference & Variational Lower Bound

The joint variational posterior distribution $q(\mathbf{z}, c|\mathbf{x}, \mathbf{y})$ can be factorized as $q(\mathbf{z}, c|\mathbf{x}, \mathbf{y}) = q(c|\mathbf{x}, \mathbf{z}, \mathbf{y}) \cdot q(\mathbf{z}|\mathbf{x}, \mathbf{y})$. Since the autoencoding component of DGC (i.e. VaDE) is meant for capturing the inherent data structure and does not make use of the side-information, we omit the variable $\mathbf{y}$ in $q(\mathbf{z}|\mathbf{x}, \mathbf{y})$ for the rest of the paper as it does not depend on $\mathbf{y}$. With this setup, we have the following variational lower bound (see the Appendix for a detailed derivation)

$$\log p(\mathbf{x}, \mathbf{y}) \geq \mathbb{E}_{q(\mathbf{z},c|\mathbf{x},\mathbf{y})} \log p(\mathbf{y}|\mathbf{z}, c) + \mathbb{E}_{q(\mathbf{z},c|\mathbf{x},\mathbf{y})} \log \frac{p(\mathbf{x}, \mathbf{z}, c)}{q(\mathbf{z}, c|\mathbf{x}, \mathbf{y})} = \mathcal{L}_{\text{ELBO}} \,. \tag{2}$$

The first term in $\mathcal{L}_{\text{ELBO}}$ allows for a probabilistic ensemble of classifiers based on the subspace index. This can be seen as follows

$$\mathbb{E}_{q(\mathbf{z},c|\mathbf{x},\mathbf{y})} \log p(\mathbf{y}|\mathbf{z},c) = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\sum_{c'} \lambda_{c'} \log p(\mathbf{y}|\mathbf{z},c') \right] \approx \frac{1}{M} \sum_{l=1}^{M} \left[\sum_{c'} \lambda_{c'} \log p(\mathbf{y}|\mathbf{z}^{(l)},c') \right] \tag{3}$$

where $\lambda_{c'} = q(c = c'|\mathbf{x},\mathbf{z},\mathbf{y})$ and l indexes the Monte Carlo samples used to approximate the expectation with respect to $q(\mathbf{z}|\mathbf{x})$. The probabilistic ensemble allows the model to maintain necessary uncertainty with respect to the discovered subspace structure until an unambiguous structure is captured.

The variational lower bound described in Eq. 2 holds regardless of the prior distribution we choose for the latent code $\mathbf{z}$. Although we choose the mixture distribution as the prior in this work, choosing $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ and disregarding the probabilistic ensemble component would recover the exact model introduced in Kingma et al. (2014) (when all labels are missing); which is therefore a special case of `DGC`.

4.3 Mean-field Variational Posterior Distributions

Following `VAE` (Kingma et al., 2014), we choose $q(\mathbf{z}|\mathbf{x})$ to be $\mathcal{N}\left(\mathbf{z}|\tilde{\mu}_{\mathbf{z}}, \tilde{\sigma}^2_{\mathbf{z}} I\right)$ where $\left[\tilde{\mu}_{\mathbf{z}}, \tilde{\sigma}^2_{\mathbf{z}}\right] = h_\theta(\mathbf{x};\theta)$. h_θ is parameterized by a feed-forward neural network with weights θ. Although it may seem unnatural to use a unimodal distribution to approximate a multimodal distribution, when the learned $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$ becomes discriminative, dissecting the $\mathcal{L}_{\text{ELBO}}$ in the following way indicates such an approximation will not incur a sizeable information loss (see Appendix for derivation):

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(\mathbf{z},c|\mathbf{x},\mathbf{y})} \log p(\mathbf{y}|\mathbf{z},c) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}|\mathbf{z}) - \mathbb{KL}\left(q(c|\mathbf{x},\mathbf{z},\mathbf{y})||p(c)\right) - \sum_{c'} \lambda_{c'} \mathbb{KL}\left(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|c')\right) \tag{4}$$

where $\lambda_{c'}$ denotes $q(c = c'|\mathbf{x},\mathbf{z},\mathbf{y})$. Analyzing the last term in Eq. (4), we notice that if the learned variational posterior $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$ is very discriminative and puts most of its weight on one specific index c, all but one $\mathbb{KL}$ terms in the weighted sum will be close to zero. Therefore, choosing $q(\mathbf{z}|\mathbf{x})$ to be unimodal to minimize that specific $\mathbb{KL}$ term is appropriate, as $p(\mathbf{z}|c)$ is a unimodal normal distribution for all c.

Choosing $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$ appropriately requires us to analyze the proposed $\mathcal{L}_{\text{ELBO}}$ in greater detail (see the Appendix for a detailed derivation):

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q(\mathbf{z},c|\mathbf{x},\mathbf{y})} \log p(\mathbf{y}|\mathbf{z},c)}_{①} + \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \log \frac{p(\mathbf{x},\mathbf{z})}{q(\mathbf{z}|\mathbf{x})}}_{②} - \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \mathbb{KL}\left(q(c|\mathbf{x},\mathbf{z},\mathbf{y})||p(c|\mathbf{z})\right)}_{③}. \tag{5}$$

We make two observations: 1) ② does not depend on c; and 2) the expectation over $q(\mathbf{z}|\mathbf{x})$ does not depend on c, and thus has no influence over our choice of $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$. Therefore, we choose $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$ to maximize $\left(① - ③\right)$ and ignore the expectation for $q(\mathbf{z}|\mathbf{x})$. Casting finding $q(c|\mathbf{x},\mathbf{z},\mathbf{y})$ as an optimization problem, we have

$$\begin{aligned} \min_{q(c|\mathbf{x},\mathbf{z},\mathbf{y})} \quad & f_0(q) = \mathbb{KL}\left(q(c|\mathbf{x},\mathbf{z},\mathbf{y})||p(c|\mathbf{z})\right) \\ & \quad - \mathbb{E}_{q(c|\mathbf{x},\mathbf{z},\mathbf{y})} \log p(\mathbf{y}|\mathbf{z},c) \\ \text{s.t.} \quad & \sum_k q(c|\mathbf{x},\mathbf{z},\mathbf{y}) = 1, \quad q(c|\mathbf{x},\mathbf{z},\mathbf{y}) \geq 0, \quad \forall k\,. \end{aligned} \tag{6}$$

The objective functional f_0 is convex over the probability space of q, as the *Kullback-Leibler divergence* is convex in q and the expectation is linear in q. Analytically solving the convex program (8) (see the Appendix for a detailed derivation), we obtain

$$q(c = k|\mathbf{x},\mathbf{z},\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{z},c = k) \cdot p(c = k|\mathbf{z})}{\sum_k p(\mathbf{y}|\mathbf{z},c = k) \cdot p(c = k|\mathbf{z})} \tag{7}$$

To better facilitate understanding, we interpret Eq. 7 in two extremes. If $\mathbf{y}$ is evenly distributed across the different subspaces, i.e., the ground truth transformations g_c are the same for all c, then $q(c|\mathbf{x},\mathbf{z},\mathbf{y}) = p(c = k|\mathbf{z})$, which is what one would choose for unsupervised clustering (Jiang et al., 2017). However, if the supervised task is informative while the unsupervised task is not ($p(c|\mathbf{z})$ is uniform), the likelihoods $\{p(\mathbf{y}|\mathbf{z},c = k)\}_k$ will dominate q. Therefore, one could interpret any in-between scenario as a balance that automatically weights the supervised and the unsupervised tasks based on how strong their signals are with respect to grouping the latent probability space into different subspaces.

Since we do not assume access to the side-information at test time, it would prohibit us from evaluating $q(c|x,z,y)$ based on Eq. 7 for a test sample x. To remedy this, we pre-train a simple neural network, f, to predict y based on an input sample x. At test time, we then use $\tilde{y} = f(x)$ to evaluate $q(c|x,z,\tilde{y})$ for the test sample x.

4.4 A Confidence Booster

For a given input pair (x, y), we want the variational posterior indicated by Eq. 7 to be as confident as possible. In other words, the entropy of the probability distribution, $q(c|x, z, y)$, should be small. To encourage this behavior, we regularize the entropy of the side-information network for a given y across clusters, i.e., $\mathbb{H}\left(\mathbf{norm}\{p(y|z, c = k)\}_{k=1}^{K}\right)$, where K denotes the number of clusters, $\mathbb{H}$ denotes the entropy operator, and **norm** denotes the softmax function that transitions $\{p(y|z, c = k\}_{k=1}^{K}$ into a proper probability distribution. We can also directly regularize the entropy of $q(c|x, z, y)$ or $p(c|z)$, but we find regularizing the side-information network works best. The total loss we optimize is

$$\mathcal{L}_{\text{Loss}} = \mathcal{L}_{\text{ELBO}} - \mathbb{H}\left(\mathbf{norm}\{p(\mathbf{y}|\mathbf{z}, c = k)\}_{k=1}^{K}\right) . \quad (8)$$

Since $\mathbb{H}\left(\mathbf{norm}\{p(\mathbf{y}|\mathbf{z}, c = k)\}_{k=1}^{K}\right)$ is non-negative and does not depend on $q(c|\mathbf{x}, \mathbf{z}, \mathbf{y})$, $\mathcal{L}_{\text{Loss}}$ is still a proper lower bound and the convexity of $\mathcal{L}_{\text{ELBO}}$ with respect to $q(c = c'|\mathbf{x}, \mathbf{z}, \mathbf{y})$ is preserved.

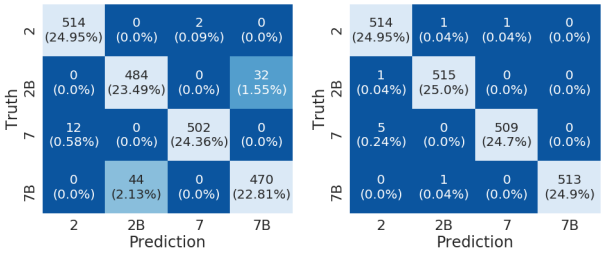
5 Experiments

We investigate the efficacy of DGC on a range of datasets. We refer readers to the Appendix for the experimental details, e.g., the train/validation/test split, the chosen network architecture, the choices of learning rate and optimizer. We also provide an additional experiment on the Street View House Number (SVHN) dataset (Netzer et al., 2011) that investigates the impact of the hyperparameter determining the number of clusters desired on DGC as a case study in the Appendix.

A general note on side-information We give a brief summary on the role of side-information in the experiments we provide:

- In Sec. 5.1, we test if we can use side-information to improve an already well-performing VaDE.
- We show that we can make use of continuous side-information in Sec. 5.2.
- In Sec. 5.3, we demonstrate that DGC is capable of utilizing fine-grained details as side-information to better form meta clusters in a natural manner.
- Finally, using the recurrence information that is collected in a real-world breast cancer dataset, we show DGC discover clusters that unveil patient characteristics that go beyond the recurrence information itself, and demonstrate the framework's usefulness in a real scenario.

We point out that most of the methods we compare DGC against do not, and *more importantly cannot*, utilize side-information in a sensible, end-to-end manner.



(a) VaDE Confusion Matrix (b) DGC Confusion Matrix

Figure 2: Confusion matrices for Noisy MNIST. Abbreviations, 2B/7B, in the row/column labels denotes digits 2/7 with background. Rows represent the predicted clusters, and columns represent the ground truth.

Additionally, we only directly use ground-truth side-information during training. At test time, we only use the prediction for the side-information obtained from the pre-trained network as we detailed in Sec. 4.3.

5.1 Noisy MNIST

We introduce a synthetic data experiment using the MNIST dataset, which we name the *noisy MNIST*, to illustrate that the supervised part of DGC can enhance the performance of an otherwise well-performing unsupervised counterpart. Further, we explore the behavior of DGC without its unsupervised part to demonstrate the importance of capturing the inherent data structure. We extract images that correspond to the digits 2 and 7 from MNIST. For each digit, we randomly select half of the images for that digit and superpose CIFAR-10 images onto those images as noisy backgrounds (see the Appendix for image samples). The binary random variable $\mathbf{y}$ indicates what digit each image belongs to. Our goal is to cluster the images into 4 clusters: digits 2 and 7, with and without background. However, we are only using the binary responses for supervision and have no direct knowledge of the background. We therefore parameterize the task networks, $\{p(\mathbf{y}|\mathbf{z}, c = k\}_{k=1}^{4}$, as Bernoulli distributions where we learn the parameters (the probabilities).

As a baseline, the unsupervised approach, VaDE, already performs well on this dataset, achieving a clustering accuracy of 95.6% when the desired number of clusters is set to 4. Fig. 2a shows that VaDE distinguishes well based on the presence or absence of the noisy background, and the incorrectly clustered samples are mainly due to VaDE's inability to differentiate the underlying digits. This is reasonable behavior: if the background signal dominates, the network may focus on the background for clustering as it has no explicit knowledge about the digits.

DGC performs nearly perfectly with the added super-

vision obtaining a clustering accuracy of 99.55%. DGC handles the difficulty of distinguishing between digits under the presence of strong, noisy backgrounds well as it makes almost no mistakes in distinguishing between digits, Fig. 2b . This added supervision does not overshadow the original advantage of VaDE (i.e., distinguishing whether the images contain background or not). Instead, it enhances the overall model in cases where the unsupervised part, i.e., VaDE, struggles. Furthermore, as detailed in Sansone et al. (2016) and earlier sections, most existing approaches that take advantage of available labels rely on *the cluster assumption*, which assumes a one-to-one correspondence between the clusters and the labels used for supervision. This experiment is a concrete example that demonstrates DGC does not need to rely on such an assumption to form a sound clustering strategy. Instead, DGC is able to work with class labels that are only partially indicative of what the final clustering strategy should be, making DGC more applicable to general settings.

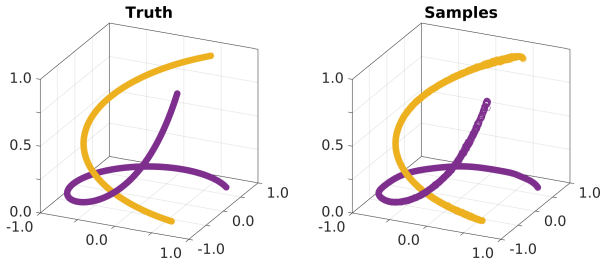
Ablation study To further test the importance of each part of our model, we ablate the probabilistic components (i.e., we get rid of the decoder and the loss terms associated with it, so that only the supervision will inform how the clusters are formed in the latent space) and perform clustering using only the supervised part of our model. We find that clustering accuracy degrades from the nearly-perfect accuracy obtained by the full model to 50%. Coupled with the improvements over VaDE, this indicates that each component of our model contributes to the final accuracy and that our original intuition that supervision and clustering may reinforce each other is correct.

5.2 Pacman

In this experiment we test DGC's ability to learn a clustering strategy when facing a *continuous* response as *side-information*. The Pacman-shaped data consists of two annuli and each point in the two annuli is associated with a continuous response value (see the Appendix for a detailed explanation). These response values decrease linearly (from 1 to 0) in one direction for the inner (yellow) annulus, and increase exponentially (from 0 to 1) in the opposite direction for the outer (purple) annulus. Figure 3a contains a 3D illustration of the dataset. We use linear/exponential rates for the responses to not only test our model's ability to detect different trends, but also to test its ability to fit different rates.

Table 1: Pacman clustering accuracies

Models	ACC
VaDE	50.4% ± 0%
NN-DGC	81.6% ± 5.3%
AUG-SS	82.3% ± 4.6%
DGC	**93.1% ± 2.6%**



(a) Truth (b) DGC Samples

Figure 3: 3D Pacman truth (a) and DGC samples (b).

Our goal is to separate the two annuli depicted in Fig. 3a. This is challenging as the annuli were deliberately chosen to be very close to each other. We applied various traditional unsupervised learning methods including K-means and hierarchical clustering to only the 2D Pacman-shaped data (i.e., not using the responses, but only the 2D Cartesian coordinates). Besides hierarchical clustering with single linkage (and not other distance metric), none of the unsupervised methods managed to separate the two annuli. Moreover, these approaches also result in different clustering strategies as they are based on different distance metrics (see the Appendix for these results). This phenomenon again reflects the fundamental problem for clustering methods in general: the concept of clustering is inherently subjective, and different distance metrics can potentially produce different, but sometimes equally meaningful, clustering results. Applying DGC with the input x as the 2D Cartesian point coordinates and the responses y as the response values described previously, we are able to distinguish the two annulli wholly based on the discriminative information carried by the responses. We parameterize the task networks, $\{p(\mathbf{y}|\mathbf{z}, c = k\}_{k=1}^{2}$, as Gaussian distributions where we learn the means and the covariance matrices. As the generated samples from Fig. 3b shows, both the Pacman shape and its corresponding response values are captured.

The generated samples from DGC substantiate the model's ability to appropriately learn and use the side-information provided by the response values to obtain a sensible clustering strategy. Unlike most previously discussed methods, DGC can work with continuous response values. This is highly attractive, as it lends itself to any general regression setting in which one would believe the desired clustering strategy should be informed by the regression task.

Finally, we compare DGC to VaDE, its ablated version, and a baseline method to substantiate the efficacy of our proposed framework. First, although the solu-

tion to the convex programming in Eq. 8 provides an optimal choice of $q(c|\mathbf{z}, \mathbf{y})$ from a theoretical standpoint, our proposed framework, specifically the proposed $\mathcal{L}_{\text{ELBO}}$ (Eq. 2), holds for any choice of $q(c|\mathbf{z}, \mathbf{y})$. We thus ablate the convex programming component of our model and parameterize $q(c|\mathbf{z}, \mathbf{y})$ using a neural network (NN-DGC). Second, recall (from Sec. 4.2) by choosing $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, the unsupervised part of DGC recovers exactly the semi-supervised (SS) approach introduced by Kingma et al. (2014) in the case when all labels (that correspond to the clusters in our case) are missing. Since SS is not expected to perform well in a purely unsupervised setting, we include the probabilistic ensemble component as an augmentation (AUG-SS).

The results in Tab. 1 are obtained from training each model 100 times, and demonstrate that: 1) without the additional responses $\mathbf{y}$, VaDE cannot distinguish between the two annuli at all, emphasizing the importance of exploiting the additional information; 2) the convex programming (Eq.8) is crucial to the success of DGC and it is difficult for a neural network to find the same optimal distribution; 3) the choice of the prior on the latent code $\mathbf{z}$ is also of paramount importance, and the Gaussian mixture distribution is more suitable for modeling clusters than an isotropic Gaussian.

5.3 CIFAR 100-20

We apply DGC to the CIFAR 100-20 dataset where the dataset setup is ideal for demonstrating the advantage of being able to utilize useful side-information for clustering. The dataset provides two types of labels for each image: one that indicates which 100 fine-grained classes the image belongs to, and one which indicates which 20 super-classes that image belongs to.

Aligning with the clustering literature, our goal is to cluster the CIFAR images into the 20 super-classes. Different from other approaches, our framework also utilizes the fine-grained classes as the side-information to aid clustering. For comparison, we compare to SCAN (Gansbeke et al., 2020), RUC (Park et al., 2020), and the current SOTA approach on this task, SPICE (Niu & Wang, 2021). We also compare to our unsupervised counterpart, i.e., VaDE, and a baseline K-means model. The results are shown in Tab. 2. We next expound upon the results and our findings.

First, by utilizing the fine-grained information, DGC expectedly outperforms its unsupervised counterpart, VaDE, by a significant margin, substantiating the advantage of using informative side-information. Secondly, to test if it is the side-information dominating the clustering accuracy, and to demonstrate the importance of seamlessly incorporating the side-information within a probabilistic framework, we perform an ablation study where we apply k-means on the last hidden layer embeddings of the input data obtained from the pre-trained network that provides DGC with the estimated side-information $\mathbf{y}$ at test time. As shown in Tab. 2, such a simple baseline does well, outperforming VaDE and SCAN. However, DGC still outperforms this baseline by a large margin. Finally, with the aid of the fine-grained information, DGC achieves better accuracy than the state-of-the-art unsupervised method, SPICE, and other two leading performance methods. As discussed, the methods DGC is comparing to do not, and cannot, utilize side-information in a sensible manner, proving the theme of this work: utilizing informative side-information helps clustering.

Table 2: CIFAR 100-20 clustering accuracy

Models	ACC
K-means	51.4%
VaDE	45.2%
SCAN	50.7%
RUC	54.3%
SPICE	58.4%
DGC	**59.8%**

Additional Study As DGC is not restricted by the cluster assumption, we can also use the 20 super-classes as the side-information to help cluster the images into 100 clusters. In this setting, compared to the clustering accuracy of 35.1% for VaDE, DGC obtains 47.6%, substantiating the utility of DGC in a practical scenario where using less expensive side-information can help categorizing data in a more fine-grained manner.

5.4 Carolina Breast Cancer Study (CBCS)

We apply DGC to a real-world breast cancer dataset collected as part of the Carolina Breast Cancer Study (CBCS). The dataset consists of 1,713 patients, each of which has 2-4 associated histopathological images and a list of biological markers like the Pam50 gene expressions (Troester et al., 2018) and ER status.

As an exploratory investigation, we use the binary indicator for breast cancer recurrence as the response variable $\mathbf{y}$. Applying deep learning techniques, supervised or unsupervised, to analyze histopathological images of breast cancer has gained traction in recent years (Xie et al., 2019). Distinguished from those methods, our goal is to inspect whether the discovered clusters, whose formation is influenced both by the supervised recurrence information and the unsupervised reconstruction signal, carry meaningful information in terms of survival rate or gene expression. Since DGC is not restricted by the cluster assumption, we train three clusters for analysis despite the binary side-information. We parameterize the task networks, $\{p(\mathbf{y}|\mathbf{z}, c = k)\}_{k=1}^{3}$, as Bernoulli distributions and learn the associated parameters. See the Appendix for experimental details.

Table 3: Tumor characteristics per cluster. Features are color-coded as low, intermediate, or high risk.

		Cluster 0 N(%)	Cluster 1 N(%)	Cluster 2 N(%)
ER Status	Positive	20 (74.1)	58 (58.0)	43 (57.3)
	Negative	7 (25.9)	42 (42.0)	32 (42.7)
Grade	Low	8 (29.6)	16 (16.0)	4 (5.3)
	Medium	7 (25.9)	25 (25.0)	26 (34.7)
	High	12 (44.4)	59 (59.0)	45 (60.0)
Tumor Subtype	Luminal A	14 (51.9)	28 (28.0)	17 (22.6)
	Luminal B	7 (25.9)	20 (20.0)	18 (24.0)
	ER-/HER2+	1 (3.7)	9 (9.0)	3 (4.0)
	Basal-like	4 (14.8)	42 (42.0)	37 (49.3)

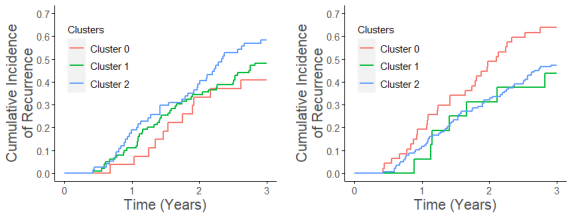
To investigate whether the three clusters that we discovered were identifying meaningful differences in tumor biology, we examine the differences in rates of cancer recurrence and features of tumor aggressiveness between the clusters. We also compare to the baseline clusters obtained from the purely unsupervised `VaDE` to corroborate the importance of the added side-information. Using a Kaplan-Meier estimator to estimate risk differences for time to cancer recurrence within three years, we obtained a p-value of 0.0024 for the differences in recurrence risk among the clusters, and observed that Cluster 0 had the lowest risk of recurrence (RRD) and Cluster 2 had the highest risk (see Fig. 4a).

Furthermore, we observed substantial differences in recurrence risk at three years of follow-up between the clusters, particularly Clusters 0 and 2 (see Fig. 4a). By comparison, with a p-value of 0.073, the differences in recurrence risk among the clusters from `VaDE` is less significant than that from `DGC` .

Comparing tumor characteristics, we observed that Cluster 0 contained more indolent tumors, characterized by good-prognosis features such as estrogen-receptor (ER) positivity, low grade, and Luminal A tumor subtype (see Tab. 3). In contrast, more aggressive tumor characteristics were featured in Clusters 1 and 2, such as negative ER status, high grade, and Basal-like tumor subtype, although Cluster 1 appeared to be intermediate in some characteristics. Coupled with the differences in cancer outcomes, these differences in tumor characteristics indicate that the method successfully distinguished between tumors with low-risk features (Cluster 0) and tumors with intermediate- and high-risk features (Clusters 1 and 2).

Table 4: `DGC` RRD between clusters

Comparsion	RRD (95% CI)
Cluster 1 VS Cluster 2	11.3% (-4.4, 26.9)
Cluster 0 VS Cluster 2	18.8% (-3.1, 40.7)



(a) `DGC` (b) `VaDE`

Figure 4: Kaplan-Meier curves for `DGC` & `VaDE`

We include the same table that characterizes tumor characteristic for clusters obtained from `VaDE` in the Appendix. On the contrary to the clusters obtained from `DGC`, clustering from `VaDE` that has the highest recurrence rate does not have the most high grade patients. The ER subtypes and the tumor subtypes of the patients in that cluster also do not corroborate the high recurrence rate. This indicates that using recurrence side-information, via our `DGC` approach, resulted in more meaningful clusters.

6 Conclusion

We have introduced `DGC`, a probabilistic framework that allows for the integration of both supervised and unsupervised information when searching for a congruous clustering in the latent space. This is an extremely relevant, but daunting task, where previous attempts are either largely restricted to discrete, supervised, ground-truth labels or rely heavily on the side-information being provided as manually tuned constraints. To the best of our knowledge, this is the first attempt to simultaneously learn from generally indirect, but informative side-information to form a sensible clustering strategy, all the while making minimal assumptions on either the form of the supervision or the relationship between the supervision and the clusters. This method is applicable to a variety of fields where an instance's input and task are defined but its membership is important and unknown. Training the model in an end-to-end fashion, we demonstrate on various datasets that `DGC` is capable of learning congruent clustering strategies that align with both the side-information and the inherent data structure.

References

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2015.

Blei, D. M. and McAuliffe, J. D. Supervised topic models. In *NIPS*, 2007.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M.

Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018.

Chapelle, O., Schölkopf, B., and Zien, A. Semi-supervised learning. 2006.

Färber, I., Günnemann, S., Kriegel, H.-P., Kröger, P., Müller, E., Schubert, E., Seidl, T., Zimek, A., and Muenchen, L.-M.-U. On using class-labels in evaluation of clusterings. 2010.

Finley, T. and Joachims, T. Supervised clustering with support vector machines. In *ICML '05*, 2005.

Gansbeke, W. V., Vandenhende, S., Georgoulis, S., Proesmans, M., and Gool, L. V. Scan: Learning to classify images without labels. In *ECCV*, 2020.

Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. In *IJCAI*, 2017.

Jin, X., Luo, J., Yu, J., Wang, G., Joshi, D., and Han, J. Reinforced similarity integration in image-rich information networks. *IEEE Transactions on Knowledge and Data Engineering*, 25:448–460, 2013.

Khashabi, D., Liu, J. Y., Wieting, J., and Liang, F. Clustering with side information: From a probabilistic model to a deterministic algorithm. *ArXiv*, abs/1508.06235, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *NIPS*, 2014.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

Law, M. T., Urtasun, R., and Zemel, R. S. Deep spectral clustering learning. In *ICML*, 2017.

Le, L., Patterson, A., and White, M. Supervised autoencoders : Improving generalization performance with unsupervised regularizers. 2018.

Mazumdar, A. and Saha, B. Query complexity of clustering with side information, 2017.

Nassif, A. B., Shahin, I., Attili, I. B., Azzeh, M., and Shaalan, K. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7: 19143–19165, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. 2011.

Niu, C. and Wang, G. Spice: Semantic pseudo-labeling for image clustering. *ArXiv*, abs/2103.09382, 2021.

Orbanz, P. and Buhmann, J. M. Nonparametric bayesian image segmentation. *International Journal of Computer Vision*, 77:25–45, 2007.

Park, S., Han, S., Kim, S., Kim, D., Park, S., Hong, S., and Cha, M. Improving unsupervised image clustering with robust learning. *ArXiv*, abs/2012.11150, 2020.

Ronan, T., Qi, Z., and Naegle, K. M. Avoiding common pitfalls when clustering biological data. *Science Signaling*, 9:re6–re6, 2016.

Sansone, E., Passerini, A., and Natale, F. Classtering: Joint classification and clustering with mixture of factor analysers. In *ECAI*, 2016.

Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., and Kluger, Y. Spectralnet: Spectral clustering using deep neural networks. *ArXiv*, abs/1801.01587, 2018.

Troester, M., Sun, X., Allott, E. H., Geradts, J., Cohen, S., Tse, C.-K. J., Kirk, E. L., Thorne, L., Mathews, M., Li, Y., Hu, Z., Robinson, W., Hoadley, K., Olopade, O., Reeder-Hayes, K., Earp, H. S., Olshan, A., Carey, L., and Perou, C. Racial differences in pam50 subtypes in the carolina breast cancer study. *JNCI: Journal of the National Cancer Institute*, 110, 2018.

Tsai, T. W., Li, C., and Zhu, J. Mice: Mixture of contrastive experts for unsupervised image clustering. *ArXiv*, abs/2105.01899, 2021.

Vu, V.-V., Do, Q., Dang, V.-T., and Toan, D. An efficient density-based clustering with side information and active learning: A case study for facial expression recognition task. *Intelligent Data Analysis*, 23: 227–240, 02 2019.

Wagstaff, K. and Cardie, C. Clustering with instance-level constraints. In *AAAI/IAAI*, 2000.

Wang, X. and Davidson, I. Flexible constrained spectral clustering. In *KDD '10*, 2010.

Wasid, M. and Ali, R. Fuzzy side information clustering-based framework for effective recommendations. *Computing and Informatics*, 38:597–620, 01 2019.

Xie, J., Liu, R., Luttrell, J., and Zhang, C. Deep learning based analysis of histopathological images of breast cancer. *Frontiers in Genetics*, 10, 2019.

Xie, Z. and Ma, S. Dual-view variational autoencoders for semi-supervised text matching. In *IJCAI*, 2019.

Xing, E. P., Ng, A. Y., Jordan, M. I., and Russell, S. J. Distance metric learning with application to clustering with side-information. In *NIPS*, 2002.