Learning Representations without Compositional Assumptions

Tennison Liu [1] **Jeroen Berrevoets** [1] **Zhaozhi Qian** [1] **Mihaela van der Schaar** [1 2]

Abstract

This paper addresses unsupervised representation learning on tabular data containing multiple views generated by distinct sources of measurement. Traditional methods, which tackle this problem using the multi-view framework, are constrained by predefined assumptions that assume feature sets share the same information and representations should learn globally shared factors. However, this assumption is not always valid for real-world tabular datasets with complex dependencies between feature sets, resulting in localized information that is harder to learn. To overcome this limitation, we propose a data-driven approach that learns feature set dependencies by representing feature sets as graph nodes and their relationships as learnable edges. Furthermore, we introduce `LEGATO`, a novel hierarchical graph autoencoder that learns a smaller, latent graph to aggregate information from multiple views dynamically. This approach results in latent graph components that specialize in capturing localized information from different regions of the input, leading to superior downstream performance.

1. Introduction

Tabular datasets encountered in the real world often contain distinct feature sets, or views, that originate from different sources of measurement. For instance, the UK Biobank (Bycroft et al., 2018) contains measurements of sociodemographic factors, heart and lung function, genomic data, and electronic health records, each providing information on a different aspect of a patient's medical state, but also dependent on one another to form a holistic medical context.

While different feature sets can be consolidated into a single table, doing so can result in suboptimal learning performance due to heterogeneity among feature sets and the loss of valuable relational information. A common approach is then *multi-view learning* (Xu et al., 2013), which examines each feature set separately and integrates information from multiple views to learn representations. This task can be difficult, particularly when labels for supervision are not available, which can help disambiguate the dependencies between views and task-relevant information. In the unsupervised learning setting, models rely on data assumptions and inductive biases to learn good representations automatically (Locatello et al., 2019).

Existing multi-view learning methods often rely on *compositional assumptions*, which assume that information is distributed and should be aggregated in predetermined patterns. The classic multi-view inductive bias assumes that views provide similar task-relevant information (Yan et al., 2021), guiding how information is aggregated, with the goal of learning robust and generalized representations that are invariant across views (Federici et al., 2019). These assumptions have been widely used in image, text, and speech domains, such as audio-visual speech recognition (Huang & Kingsbury, 2013) and image-caption models (Radford et al., 2021), where the settings are more controlled and the number of views is limited. In these domains, systematic and aligned data collection ensures maximal information overlap between feature sets, making inter-view relationships and information aggregation strategies known in advance.

However, these assumptions may not hold for tabular multi-view data, especially those collected *in-the-wild* (ITW), where relationships between feature sets are significantly more opaque. Examples of this include electronic health records (Johnson et al., 2023), biobanks (Nagai et al., 2017), and stock market data (Xu & Cohen, 2018). In these datasets, information is more likely to exist in localized clusters of views in unknown patterns, rather than being globally present in all views (Xu et al., 2013). This is particularly true when dealing with tabular problems that typically have more than two feature sets. Our findings indicate that compositional assumptions are inadequate when learning on tabular data collected in-the-wild, failing to capture localized information in representations.

To overcome this challenge, we propose a method to model relationships between feature sets and dynamically aggre-

[1]DAMTP, University of Cambridge, Cambridge, UK [2]Alan Turing Institute, London, UK. Correspondence to: Tennison Liu <tl522@cam.ac.uk>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

gate potentially localized information. We represent feature sets as graph nodes and their relationships as learnable edges. Furthermore, we introduce the Latent Graph AuTOencoder (LEGATO), a novel graph neural network that learns a smaller, latent graph. This architecture innovates on existing autoencoder architectures that learn compact node embeddings, but do so on the same topology as the input graph. Our method learns a smaller *graph*, which is crucial, as it allows for end-to-end learning of information aggregation strategies without relying on predefined assumptions. We term the latent graph a *decomposable representation* to emphasize that, by design, it can be decomposed into node representations that specialize in aggregating information from different regions of the input. We evaluate the effectiveness of our method by testing its ability to transfer to downstream tasks, as a good representation should facilitate subsequent problem-solving.

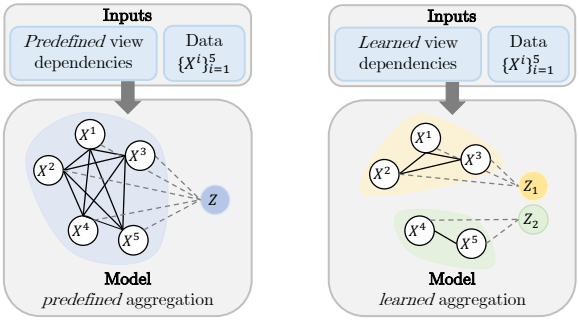
Contributions. 1. We identify the challenges associated with learning representations from heterogeneous tabular feature sets collected in real-world settings and showcase the limitations of existing unsupervised learning methods that heavily rely on predefined compositional assumptions. **2.** Instead of relying on predefined assumptions, we propose a novel approach that treats feature sets as graphs to capture dependencies, which to the best of our knowledge, is a novel way to represent multi-view data. **3.** We introduce LEGATO, a novel graph autoencoder architecture that learns a smaller latent graph. This smaller graph induces a decomposable representation by dynamically aggregating localized information in a hierarchical manner. We conduct simulation studies to demonstrate the effectiveness of our model in learning data-driven aggregation strategies. Moreover, we showcase the superior downstream performance of our method on multiple real-world datasets.

2. Problem Definition

2.1. Notation

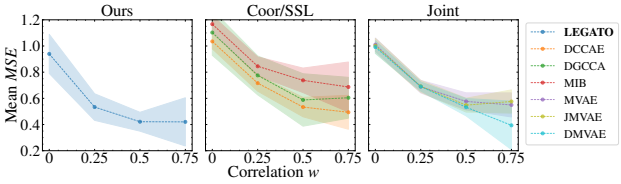
In this paper, we use the terms "feature sets" and "views" interchangeably. We consider K different feature sets, depicting one instance $X = \{X^k : k \in [K]\}$. Each X^k is sampled from a space $\mathcal{X}^k \subseteq \mathbb{R}^{d^k}$, and $\mathcal{X} = \mathcal{X}^1 \times \cdots \times \mathcal{X}^k$. With X the random variable, we have $x = \{x^k : k \in [K]\}$ as its realization. For each x^k, we have a d-dimensional view embedding $h^k \in \mathcal{H}^k \subseteq \mathbb{R}^d$ produced using an encoder function $g^k : \mathcal{X}^k \to \mathcal{H}^k$.[1] Correspondingly, $f^k : \mathcal{H}^k \to \mathcal{X}^k$ denotes the view decoder function. We are agnostic to the exact architecture of $g^k(\cdot)$ and $f^k(\cdot)$ for generality. We have access to a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, with N iid samples. We use superscript to indicate view and subscript for the sample, such that x_i^k is the k^{th} view of the i^{th} sample. When the context is clear, we may drop the subscript to declutter exposition.

[1] We assume the embedding dimension is d for all views for notation convenience, but this restriction is not necessary.

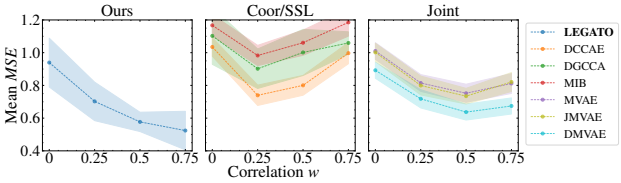


(a) *Predefined* aggregation. (b) *Learned* aggregation.

Figure 1: **Dynamically aggregating information**. Solid lines represent information sharing and dashed lines represent aggregation. Existing methods (1a) assume views share the same information and aggregate information globally. In comparison, our method (1b) learns dependencies and aggregation strategy in a data-driven manner.



(a) Views are *globally* correlated.



(b) Views are *locally* correlated.

Figure 2: **Effect of view correlation on learning (K=6).** When views are globally correlated, higher correlation improves performance for all models. When local correlation increases, the performance of existing methods deteriorates as they fail to learn localized information.

2.2. Challenges of Learning In-The-Wild

Compositional assumptions. Compositional assumptions are two-fold: they reflect beliefs on how information is shared between feature sets, and how information should be aggregated in a representation. The multi-view assumption is the predominant compositional assumption made in existing works—it posits that important information co-occurs in all available views, leading to a focus on maximizing mutual information among multiple view representations (Federici

et al., 2019). This approach has been successfully applied in many domains, especially image, text, and speech, where it is known apriori (e.g. through careful data collection) that the semantically meaningful variations exist in all signals (Vrigkas et al., 2015; Radford et al., 2021). By learning shared information, these methods improve the robustness and generalization of multi-view representations. More recently, methods have also considered the possibility that each view may contain unique information not present in other views (Xu et al., 2013), with the aim of retaining both view-specific and globally shared information.

Multi-view data collected in-the-wild. Tabular feature sets collected in-the-wild (ITW) present a different challenge, as information is rarely presented in known patterns across different views. We argue that tabular multi-view data found in the real world are characterized by two main features: ► **Localized information** - where different sources of information are concentrated in localized subsets of views, as opposed to the globally shared information assumed by existing methods, and ► **A larger number of views** - resulting in more complex dependencies and localized clusters of information. We provide further discussions and detailed case studies on these feature sets and their characteristics in Appendix A.

These characteristics make the representation learning task more challenging. Existing methods use the multi-view inductive bias to infer a common representation z (and optionally a set of view-specific representations $\{z_i\}_{i=1}^K$), leading to a global aggregation of information, as visualized in Figure 1a. However, these assumptions are inadequate to address problems ITW, which contain localized information that manifests in unknown ways. Additionally, the large number of possible view combinations (combinatorial in K) makes it infeasible to explicitly consider different local aggregation patterns. Our method, as depicted in Figure 1b, addresses this challenge by proposing a novel approach to dynamically learn dependencies and aggregate information, without relying on predefined assumptions.

Learning challenges. Given the learning capacity and expressiveness of modern neural networks, it is natural to wonder whether incorrectly specified compositional assumptions are truly detrimental in practice. While representations may be biased, they can still implicitly learn all localized sources of information. We empirically show that this is not the case in a simulation study (described in Section 5.1), where the downstream task is to predict the latent variables that generated each view. Existing methods perform better as the global correlation between latent variables increases (Figure 2a). This is intuitive because views contain more information about latent variables in other views, which can be effectively learned using the multi-view inductive bias. However, when latent variables are only locally correlated (Figure 2b), increased correlation does not lead to improved performance. This is because higher correlation only provides locally useful information, which is overlooked when incorrect compositional assumptions are used.

3. Proposed Method

We propose a framework for *learning* information aggregation patterns from data without predefined compositional assumptions. This requires accounting for localized information sharing between views, which can be naturally represented using graphs. Our method makes two contributions: first, we learn the view dependencies as edges in a graph, which, to the best of our knowledge, is a novel way to represent multi-view data. Second, we introduce `LEGATO`, a novel graph autoencoder that learns a smaller latent graph. The latent graph produces a decomposable representation that aggregates localized information. To complete the autoencoder, the latent graph is unpooled to reconstruct each view individually, with the hierarchical process trained end-to-end. Our proposed method is illustrated in Figure 3.

3.1. Learning the Multi-view Graph

We define an initial graph on view embeddings, where nodes represent views and edges represent the inter-view relationships, i.e. $G^{(0)} := \left(H^{(0)}, A^{(0)}\right)$. $A^{(0)} \in [0,1]^{K\times K}$ is the adjacency matrix between K views and $H^{(0)} \in \mathbb{R}^{K\times d}$ is the node feature matrix, where the k^{th} row is the view embedding h^k. We are agnostic to the view encoder-decoder architecture and first obtain view embeddings $h^k = g^k(x^k)$ independently for each view $k \in [K]$.

The adjacency matrix $A^{(0)}$ is rarely known. In the most general setting, every node can be connected to every other node, ignoring localized structure. This reflects the multi-view inductive bias, which assumes that each view shares the same information with all other views (as shown in Figure 1a). Clearly, this is not the case ITW, as certain views will only share information locally with other views.

We propose to learn the localized graph structure. Specifically, $\texttt{GRAPHLEARNER} : \mathbb{R}^{K\times d} \rightarrow [0,1]^{K\times K}$, which takes as input the view embeddings and returns the adjacency matrix. We first apply a non-linear transformation to each view embedding:

$$e_i = \texttt{LeakyReLU}\left(W[h_i \| 1_i]\right) \tag{1}$$

where $W \in \mathbb{R}^{d'\times f}$ applies a linear transformation, followed by a $\texttt{LeakyReLU}(\cdot)$ activation. We encode view information for view i through the concatenation operation $\|$ of h_i and the one-hot encoding 1_i to obtain a d'-dimensional input. Then, we compute the inner product between views, normalized by the `sigmoid` function $\sigma(\cdot)$:

$$A^{(0)}_{ij} = \sigma(e_i^T \cdot e_j) \tag{2}$$

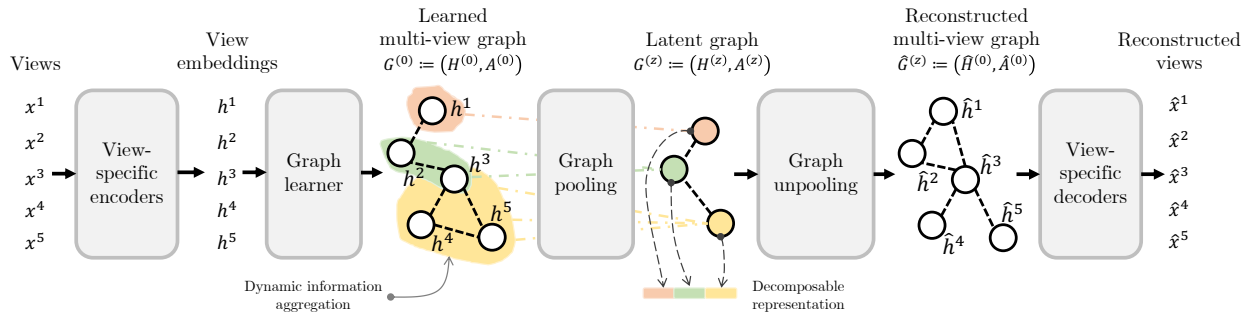


Figure 3: **High-level illustration of LEGATO.** The latent graph dynamically pools information by considering both view embeddings and dependencies. The latent graph returns a decomposable representation for downstream tasks.

The normalized coefficients take values $\in [0, 1]$ to represent the dependence between views. Note that the mechanism is invariant to the ordering of inputs and that $A^{(0)}$ is a symmetric matrix. We additionally apply a threshold function to $A^{(0)}$, where entries < 0.1 are considered uninformative and zeroed out. As we want informative local neighbors to be found, we add a regularization term $\mathcal{L}_{spar} = \frac{1}{NK^2}\sum_{i=1}^{N}\|A_i^{(0)}\|_1$, where $\|\cdot\|_1$ denotes the $p = 1$ matrix norm. This term encourages sparsity in the adjacency matrix and reduces the learning of spurious dependencies between views (e.g. by learning a fully-connected graph).

A distinction. We emphasize that our goal is not *relational inference*, which seeks to infer relationships between views from observation data (Kipf et al., 2018; Hajiramezanali et al., 2020; Hasanzadeh et al., 2021). In this problem, a correctly recovered relational structure *is* the object of inference. This stands in stark contrast to our work, where a partially correct structure is satisfactory, as our main purpose is to aggregate information while considering local dependencies. As we shall show later, even learning a partially correct structure can greatly improve the learned representations.

3.2. LEGATO: Latent Graph Autoencoder

After learning an initial adjacency matrix, the next step is to aggregate information shared between views in a latent graph. To do this, we leverage the intuition that views with similar information should be aggregated together. We introduce `LEGATO`, a hierarchical procedure that learns a latent graph while pooling essential information (Cai et al., 2021). In more detail, we transform $G^{(0)}$ through a pooling step to obtain a latent graph $G^{(z)}$. This transformation pools information shared between views, so that each latent node aggregates localized information. Next, we take an unpooling step to reconstruct the graph $\hat{G}^{(0)}$, and the entire hierarchical model is trained end-to-end as an autoencoder.

We use *graph neural networks* (GNN) to learn the latent graph representation (Gilmer et al., 2017; Zhou et al., 2020). However, existing graph autoencoders are unsuitable for our purposes. The latent graphs in existing works learn compact node embeddings on the same graphical structure as the input graph, where similarity objectives are used to encourage embeddings of topologically connected nodes to be more similar (Kipf & Welling, 2016b; Simonovsky & Komodakis, 2018). In contrast, the latent graph learned in `LEGATO` is a smaller, pooled graph that aggregates information from input views with stronger dependencies. We provide an overview of GNN methods and elaborate on related graph autoencoder methods in Appendix B.

Graph pooling. The latent graph $G^{(z)} := (H^{(z)}, A^{(z)})$ is a pooled graph with $K' < K$ nodes. Here, $A^{(z)} \in [0, 1]^{K' \times K'}$ and $H^{(z)} \in \mathbb{R}^{K' \times r}$, where each row is a r-dimensional latent node embedding. We propose a graph pooling operation $(H^{(z)}, A^{(z)}) = \text{POOL}(H^{(0)}, A^{(0)})$ by adapting the `DiffPool` algorithm (Ying et al., 2018). In our experiments, we set $K' = K/2$, which was found to be a robust setting. Additionally, we note that by setting $K' = 1$, we can perform global aggregation, similar to existing methods.

Pooling strategy. The pooling strategy is learned through a separate network that considers localized dependencies and view embeddings. This is different from traditional compositional assumptions that predefine the pattern of aggregation. Specifically, we learn a pooling matrix $P \in [0, 1]^{K \times K'}$ in an input-dependent way by considering both view embeddings in $H^{(0)}$ and view dependencies in $A^{(0)}$. Intuitively, views that are dependent on each other likely contain similar information and should be aggregated together. To operationalize this insight, we learn the pooling matrix through a GNN:

$$P = \texttt{softmax}\left(\text{GNN}_{pool}(A^{(0)}, H^{(0)})\right) \tag{3}$$

The $\texttt{softmax}(\cdot)$ is applied in a *row*-wise fashion. Consequently, P indicates how information should be aggregated, where P_{ij} describes the contribution of the i^{th} view in the multi-view graph to the j^{th} node in the latent graph.

Latent embeddings. We employ a separate GNN to update view embeddings using neighboring views' embeddings through message passing. This network produces $Z \in \mathbb{R}^{K \times r}$, where each row now contains the updated r-dimensional embedding for each view:

$$Z = \text{GNN}_{embed}(A^{(0)}, H^{(0)}) \tag{4}$$

By combining the pooling matrix and the transformed embeddings in Equations (3) and (4), we can now define the complete POOL operation. Mathematically, we can obtain the latent graph using the following equations:

$$A^{(z)} = P^T A^{(0)} P \in \mathbb{R}^{K' \times K'} \tag{5}$$

$$H^{(z)} = P^T Z \in \mathbb{R}^{K' \times r} \tag{6}$$

As in Equation (6), the latent embeddings are constructed through a weighted combination of transformed view embeddings using the pooling strategy in P. This reflects the intuition that if a latent node pools information from a set of views, then its embedding should be constructed from those views. Correspondingly, the latent adjacency matrix Equation (5) considers existing connectivity strength in $A^{(0)}$ and P to compute a weighted sum of edges between neighboring nodes.

Orthogonality loss. In practice, it can be difficult to train the pooling function GNN_{pool} using only gradient signal from an unsupervised loss. Instinctively, the function can learn a degenerate assignment where information is evenly pooled in the latent nodes, akin to the degeneracy of clustering (Alguwaizani, 2012). This would achieve the opposite of our desired objective, as we want latent nodes to aggregate different localized information. To alleviate this issue, we introduce an orthogonality regularization:

$$\mathcal{L}_{orth} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{C} \sum_{k=2}^{K'} \sum_{j=1}^{k-1} \left\| \rho\left(h_i^k, h_i^j\right) \right\|_1 \tag{7}$$

where $C = \frac{K' \cdot (K'-1)}{2}$ is the number of pairwise correlations and $\rho(\cdot, \cdot)$ is calculated using cosine similarity. This term encourages orthogonality in the embeddings by decorrelating them. This encodes the intuition of decomposable representations, that each component should specialize in aggregating information from different local regions of the input, resulting in better representations for downstream tasks (Mathieu et al., 2019).

3.3. Completing the Graph Autoencoder

Graph unpooling. The unpooling step decodes the original multi-view input from the pooled latent graph. We define the unpooling step $(\hat{H}^{(0)}, \hat{A}^{(0)}) = \texttt{UNPOOL}(A^{(z)}, H^{(z)})$, where, $\hat{H}^{(0)}$ and $\hat{A}^{(0)}$ have the same dimensions as the input multi-view graph. Unpooling is mathematically identical to the pooling steps described in Equations (3) to (6). The intuition is also similar, in that the input graph is reconstructed based on a weighted combination of adjacency patterns and embeddings of the latent nodes. After the unpooling step, the node embeddings are passed through the corresponding view-specific decoders to reconstruct the views $\hat{x} = \{\hat{x}^k \ : \ k \in [K]\}$.

Training. It is worth mentioning that multiple pooling and unpooling steps can be stacked, leading to the network gradually operating on more compressed latent graphs. For training the hierarchical model, we specify a reconstruction loss defined on the multi-view graphs:

$$\begin{aligned} \mathcal{L}_{recon} = & \frac{1}{NK} \sum_{i=1}^{N} \sum_{k=1}^{K} \|x_i^k - \hat{x}_i^k\|_2^2 \\ & + \frac{1}{N} \sum_{i=1}^{N} \|A^{(0)} - \hat{A}^{(0)}\|_2^2 \end{aligned} \tag{8}$$

where the first term is a loss on reconstructed node embeddings and the second term is a loss on the recovered graph structure, together forming the graph reconstruction loss. This loss is combined with the regularization terms to form the training objective: $\mathcal{L}_{recon} + \alpha \mathcal{L}_{orth} + \beta \mathcal{L}_{spar}$, where $\mathcal{L}_{spar}$ regularizes the sparsity of the learned multi-view graph to reduce learning of spurious dependencies between views and $\mathcal{L}_{orth}$ is an orthogonality regularization that decorrelates latent node embeddings. α and β are the corresponding weighting terms for the two regularization terms. This expression and the hierarchical procedure are fully differentiable and can be trained end-to-end using autograd techniques.

3.4. Latent Graph and Decomposable Representations

Existing unsupervised algorithms learn a latent representation that integrates different sources of information shared between views. However, this often results in representations that entangle localized information and are difficult to differentiate for downstream models. Our learned latent graph is decomposable and is expected to better preserve information and make it more amenable for downstream tasks (Lipton, 2018).

Decomposable representations. We claim that the latent graph is decomposable, as nodes act as specialized components that extract localized information from different regions in the input, and are encouraged to be orthogonal through $\mathcal{L}_{orth}$. To make the representation more suitable for downstream models, we include an additional readout step that converts the latent graph into a vector $\texttt{READOUT} : \mathbb{R}^{K' \times r} \times [0,1]^{K' \times K'} \to \mathbb{R}^r$. We use mean pooling to aggregate the node embeddings $z = \frac{1}{K'} \sum_{k=1}^{K'} h^k$ and produce a vector representation that is composed of orthogonal components. Future works can consider more advanced readout strategies, including those

that take into account graph topology (Buterez et al., 2022).

Our approach can be informally compared to convolutional networks that extract localized information from natural images, which contain features in localized patches (LeCun et al., 2010). Importantly, pixels are related in a grid-like pattern and convolutional networks exploit this structure to learn and pool localized information. In our case, the relationships between views are not known a priori. Instead of making predefined assumptions, we model multi-view data as graphs and learn localized dependencies as edge weights. Subsequently, our graph autoencoder facilitates locality in information aggregation to compose representations.

4. Related Works

This work proposes a novel graph autoencoder for unsupervised representation learning on tabular multi-view data collected ITW. As such, there are two lines of related works: multi-view learning methods and GNN architectures.

Multi-view learning. Many existing methods assume that good bits of information co-occur in multiple views, and aim to extract globally present information. Figure 1a depicts the generative view of this assumption. One predominant approach is to obtain a *joint representation* by integrating view representations onto the same latent space $z = f\left(g^1(x^1), \ldots, g^k(x^k)\right)$. Ngiam et al. (2011) leveraged stacked autoencoders to obtain joint representation, whereas Srivastava & Salakhutdinov (2012a;b) used probabilistic graphical models to infer z. More recent works have used variational autoencoders (VAE) (Kingma & Welling, 2013). Suzuki et al. (2016) introduced a joint encoder structure to learn joint representations, whereas Wu & Goodman (2018) and Shi et al. (2019) proposed to combine view representations into a joint representation using product-of-experts (PoE) and mixture-of-experts (MoE) respectively.

Another approach learns *coordinated representations* by placing regularization $\phi(\cdot)$ on the correlation structure between representations to create a coordinated latent space, i.e. $\arg\max_{h_{1:K}} \phi(h_{1:K})$. Prominent methods are based on canonical correlation analysis (CCA), which learns a common space where the linear canonical correlation between two views is maximized (Hardoon et al., 2004). Subsequent works have introduced non-linear extensions (Akaho, 2006; Andrew et al., 2013; Wang et al., 2015). These methods rely heavily on pair-wise coordination and cannot efficiently scale to more views. To address this, Benton et al. (2017) generalized CCA-style analysis to more than two views. Recent works have also adopted *self-supervised learning* (SSL) objectives, which roughly maximize the mutual information between paired views. Federici et al. (2019) employs a mutual information bottleneck (MIB) to only retain mutual information between views. CLIP (Radford et al., 2021) contrastively maximizes (minimizes) cosine similarity of paired (unpaired) image-text samples.

Table 1: **Related works.** Comparison of representative *unsupervised multi-view learning methods* based on **training objective**, **assumed generative view (Asm)**, and desiderata: **(1)** scales to > 2 views, **(2)** learns localized information, and **(3)** dynamically learns aggregation strategy.

	Method	Objective	Asm	(1)	(2)	(3)
Joint	Suzuki et al. (2016)	Recon	fig 1a	✓	✗	✗
	Wu & Goodman (2018)	Recon	fig 1a	✓	✗	✗
	Zhang et al. (2019)	Recon	fig 1b	✓	✓	✗
	Lee & Pavlovic (2021)	Recon	fig 1b	✓	✓	✗
Coor.	Andrew et al. (2013)	CCA	fig 1a	✗	✗	✗
	Wang et al. (2015)	CCA	fig 1a	✗	✓	✗
	Benton et al. (2017)	CCA	fig 1a	✓	✗	✗
SSL	Federici et al. (2019)	MIB	fig 1a	✗	✗	✗
	Radford et al. (2021)	Contrastive	fig 1a	✗	✗	✗
	Tian et al. (2020)	Contrastive	fig 1a	✓	✗	✗
	LEGATO	**Recon**	**NA**	✓	✓	✓

(a) (b)

Figure 4: ***Assumed* compositional structure.**

The training objectives in CCA and SSL-based methods explicitly encourage learning of a view-invariant representation. A similar effect is implicit in joint representation methods, which can discard localized variations in shared representation spaces (Daunhawer et al., 2021; Wolff et al., 2022). When employed ITW, this bias towards global information can lead to fine-grained localized information being overlooked. Recent methods have additionally sought to preserve view-specific information (the generative model view of this assumption is presented in Figure 1b). MFM (Tsai et al., 2018) factorizes z into view-specific factors and shared factors but requires label information. Perhaps most similar to our work, Ye et al. (2016) and DMVAE (Lee & Pavlovic, 2021) aim for decomposable representations by explicitly separating shared and view-specific factors. However, both methods still rely on assumptions of global information and additionally, target information that manifests privately in each view. Our work does not require compositional assumptions and is capable of learning appropriate aggregation by accounting for inter-view relationships. We compare representative works in Table 1.

GNN. Graph autoencoders map graphs into a representation space to subsequently decode graph information from latent representations. Wang et al. (2016); Simonovsky & Komodakis (2018) embeds a graph into a continuous representation $z \in \mathbb{R}^r$ to ensure topologically close nodes have similar representations. You et al. (2018) focuses on graph

Table 2: **GNN methods.** Overview of representative *graph autoencoder* architectures based on **encoder/decoder (Enc/Dec)** architecture, **latent representation (Lat Rep)** and **aim**. Sim = similarity measure, DP = decision process.

Method	Enc/Dec	Lat Rep	Aim
Wang et al. (2016)	MLP/MLP	$z \in \mathbb{R}^r$	Node embeddings
Kipf & Welling (2016b)	GNN/Sim	$A, H^{(z)}$	
You et al. (2018)	RNN/DP	$A, H^{(z)}$	Graph generation
Simonovsky & Komodakis (2018)	GNN/MLP	$z \in \mathbb{R}^r$	
De Cao & Kipf (2018)	GNN/MLP	$z \in \mathbb{R}^r$	
LEGATO	**GNN/GNN**	$\mathbf{A}^{(z)}, \mathbf{H}^{(z)}$	**Information aggregation**

generation, recursively learning node embeddings to generate a graph sequentially. Instead of graph embeddings, Kipf & Welling (2016b) infers a latent embedding for each node in the input graph. These works focus on learning embeddings on a fixed input graph $G^{(0)}$, making them unsuitable for our purpose of dynamic and localized information aggregation. Our method is novel in that it hierarchically learns a smaller latent graph $G^{(z)}$ whose node embeddings represent locally aggregated information. We provide an overview of related architectures in Table 2.

Previous methods have used GNNs for multi-view data by either: **1.** processing each view as a separate graph and using GNN to integrate node representations between graphs (Kim et al., 2020; Ma et al., 2020); or **2.** constructing an instance graph, where nodes represent instances of the data and edges represent relationships between them across views (Wei et al., 2019; Gao et al., 2020). In this work, we are the first to represent views as nodes and learn edge weights to indicate view dependencies.

5. Empirical Investigations

Having introduced the challenges of learning from multi-view data ITW and our proposed method to address it, we now turn to quantitatively evaluating our method:

1. **Learning ITW:** *What is the problem?* Section 5.1 employs a simulation of ITW multi-view data to probe the performances of different compositional assumptions.
2. **Insights:** *How does it work?* We use interpretability methods to interpret the graphs and latent aggregations.
3. **Performance:** *Does it work?* Section 5.2 evaluates downstream performance of our method against state-of-the-art benchmarks on real world dataset.
4. **Gains:** *Why does it work?* We deconstruct our method to investigate its sources of performance gain.

Benchmarks. We evaluate our method against 7 state-of-the-art methods, in line with benchmarks found in recent works (Federici et al., 2019; Zhang et al., 2019; Lee & Pavlovic, 2021). We consider two coordinated representation methods: **DCCAE** (Wang et al., 2015) and **DGCCA** (Benton et al., 2017); three joint representation methods: **JMVAE** (Suzuki et al., 2016), **MVAE** (Wu & Goodman, 2018), and **DMVAE** (Lee & Pavlovic, 2021); and one SSL method: **MIB** (Federici et al., 2019). We also include a vanilla **Transformer** model (Vaswani et al., 2017), which takes in a sequence of view embeddings and is pretrained using a reconstruction loss. For all results, we report the mean $\pm$ std averaged over 10 runs. Our implementation can be found at `https://github.com/tennisonliu/LEGATO` and at the wider lab repository `https://github.com/vanderschaarlab/LEGATO`. We provide additional information about implementation details, dataset preprocessing, and hyperparameters tuning in Appendix C.

5.1. Synthetic Simulation

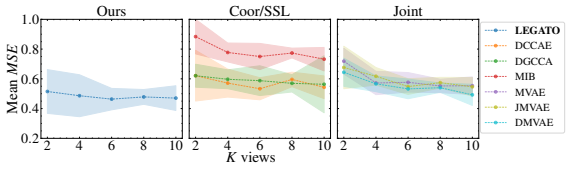
In Section 2.2, we characterized real-world ITW data as having more complex view dependencies, giving rise to clusters of localized information, and a larger number of views. In this subsection, we investigate the effect of these two characteristics on the quality of representations. We consider two view correlation settings, ▶ `global` : all views are globally correlated with each other, and ▶ `local` : views are locally correlated. We construct the following simulation as it is difficult in practice to have natural datasets that possess the required degree of view interaction.

Simulation setting. We simulate multi-view data with $K < 10$ views. Each view is generated from a scalar latent variable such that $z_k \sim \mathcal{N}(k, 1)$ and $z_k \to x_k \;\; \forall\, k \in [K]$. We simulate `global` correlation between views by computing $z_k \leftarrow (1-w) \cdot z_k + w \cdot z_1 \; \forall\, k \in [K]$, such that information from z_1 is shared across all views. Additionally, w controls how much information is shared, with a larger w indicating higher degrees of overlap, and $w = 0$ meaning each view is mutually independent. To simulate `local` correlation, we sample each pair of latent variables from a multivariate normal distribution, i.e.:

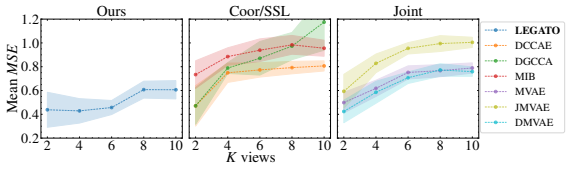
$$z_1, z_2 \sim \mathcal{N}(\mu, \mathbf{\Sigma}), \;\; \text{where } \mu = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, \; \mathbf{\Sigma} = \begin{bmatrix} 1 & w \\ w & 1 \end{bmatrix}$$

which with K views would give us $K/2$ localized clusters, where each cluster of two views is correlated while being mutually independent of other clusters. We generate 100-dimensional feature vectors for each view using a non-linear transformation, $x_k = MLP_k(z_k)$, where $MLP_k(\cdot)$ is a randomly initialized single-layer MLP with $\texttt{Tanh}(\cdot)$ activation. The downstream task is the recovery of view-specific latent variables $\{z\}_{i=1}^K$, which is a good proxy for whether representations learn localized information.

Results. We consider w in range $\{0.0, 0.25, 0.50, 0.75\}$ and

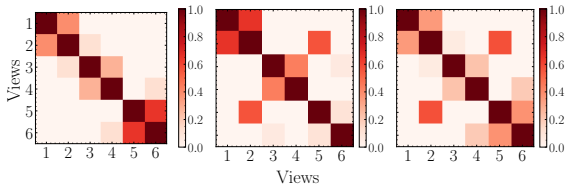


(a) Views are *globally* correlated.

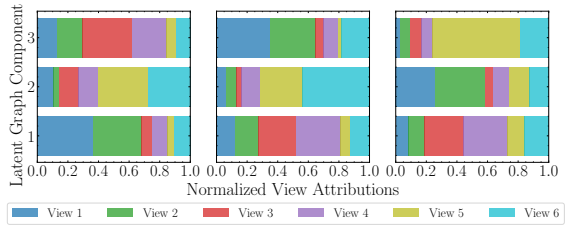


(b) Views are *locally* correlated.

Figure 5: **Effect of K on learning (w=0.5).** When views are globally correlated, more views lead to better performance. When local correlation increases, performance worsens as more localized clusters of information emerge.



(a) Learned edge weights $\in [0, 1]$ in multi-view graph.



(b) Contribution of views to latent node embeddings.

Figure 6: **Model inspection (K=6).** Our method dynamically learns view dependencies and latent nodes (components) specialize in aggregating localized information.

K in range $\{2, 4, 6, 8, 10\}$. We plotted the effect of view correlation w and the number of presented views K on representation quality by evaluating the mean MSE in Figures 2 and 5 respectively. As we previously noted, higher global correlation improves the performance of all models, as each view contains more information about all other views. However, increased local correlation is found to decrease the performance of existing methods, which are biased by their compositional assumptions to overlook localized information in favor of globally present factors. Additionally, while views are globally correlated, a larger number of views lead to better performance. In contrast, when views are locally correlated, performances of conventional methods deteriorate quickly as more localized clusters emerge. In comparison, our work is the only one that can effectively learn localized information with higher degrees of local correlation and a larger number of views.

Table 3: **Downstream classification results on three multi-view datasets.** Bold indicates the best performance.

	Method	TCGA (AUROC ↑)	UK Biobank (AUROC ↑)	UCI-MFS (ACC ↑)
Baselines	DCCAE	0.673 ± 0.047	0.624 ± 0.041	0.742 ± 0.034
	DGCCA	0.620 ± 0.073	0.669 ± 0.058	0.688 ± 0.031
	JMVAE	0.695 ± 0.034	0.718 ± 0.043	0.825 ± 0.057
	MVAE	0.656 ± 0.039	0.715 ± 0.059	0.818 ± 0.042
	DMVAE	0.676 ± 0.029	0.688 ± 0.049	$\mathbf{0.825 \pm 0.043}$
	MIB	0.620 ± 0.083	0.696 ± 0.067	0.813 ± 0.036
	Transformer	0.679 ± 0.080	0.711 ± 0.064	$\mathbf{0.825 \pm 0.029}$
Ablation	NoHier	0.652 ± 0.036	0.710 ± 0.041	0.782 ± 0.034
	NoGraph	0.696 ± 0.032	0.698 ± 0.030	0.794 ± 0.046
	NoReg	0.688 ± 0.039	0.679 ± 0.032	0.801 ± 0.037
	LEGATO	$\mathbf{0.703 \pm 0.051}$	$\mathbf{0.720 \pm 0.038}$	0.824 ± 0.030

Model inspection. We investigate the inner workings of our proposed method and the learned multi-view graph and latent graph embeddings. We visualize learned dependencies in the multi-view graphs in Figure 6a and use Integrated Gradients (Sundararajan et al., 2017) to visualize the contribution of each view to latent node embeddings in Figure 6b. We note that, while our method is not designed for *relational inference*, it can dynamically learn dependencies between views. Additionally, we see the specialization of latent nodes to aggregate information from different regions of the input, where each node focuses on extracting information from more correlated views.

5.2. Overall Performance

Datasets. We now move on to evaluate our method on three real-world datasets. ► **TCGA** (Tomczak et al., 2015) is a multi-omics dataset containing 7295 cancer cell lines with 4 views: mRNA expressions, DNA methylation, microRNA expressions, and reverse-phase protein array. The downstream task is to predict one-year mortality from cancer. ► **UK Biobank** (Sudlow et al., 2015) is a large population-based medical database. We extract a lung mortality dataset containing 9 views based on the given feature categorizations.[2] The views include patient demographics, view and lifestyle factors, physical measures, recorded medical conditions, biomarkers, physical measures, geographical information, treatment history, and family/heredity conditions. The downstream task is the binary classification of lung cancer mortality. ► **UCI-MFS** (van Breukelen et al., 1998) is more representative of a traditional multi-view task, where views share similar information. Here, all views contain handcrafted features extracted from images of handwriting. The downstream task is to predict the handwritten numerals (0-9). We describe dataset characteristics and pre-processing in Appendix C.

[2] https://biobank.ctsu.ox.ac.uk/crystal/cats.cgi

Ablation study. Our method is designed with a number of characteristics in mind. Having empirically demonstrated strong overall results, an immediate question is how important these characteristics are for performance. Specifically, we consider the sources of gain from **(a)** *hierarchical graph pooling* (**NoHier**), we consider removing the pooling layer, relying simply on GCN layers, **(b)** *multi-view graph learning* (**NoGraph**), we replace the learned input graph with a fully-connected graph, and **(c)** *orthogonality regularization* (**NoReg**), we remove the orthogonality regularization.

Results. We report downstream classification performance in Table 3. We first analyze the performance on **TCGA** and **UK Biobank**, which are more representative of tasks found *in-the-wild*, with more complex view dependencies and a higher number of views. We note that in these settings, LEGATO achieves superior performance, being particularly suited for learning the complex dependencies between views and aggregating localized information. We additionally find that joint representation methods perform better than their coordinated counterparts, likely as the emphasis on shared information aggregation is implicit rather than explicitly enforced in CCA and SSL methods. Next, we investigate performance on **UCI-MFS**, which is more representative of traditional multi-view tasks. Here, we find that our model performs on par with state-of-the-art methods. This is likely because the multi-view assumption holds true, empowering baseline methods (e.g. **DMVAE**, **Transformer**) that exploit the multi-view inductive bias. On our ablation settings, we observe all three aspects are crucial for performance, with a notable 8% performance gain over a GCN network with no latent graph learning. Similarly, orthogonality regularization improves model performance by encouraging orthogonal components. We note that this is more crucial on ITW datasets with more views, as this regularization better encourages the learning of localized information.

6. Discussion

Existing multi-view methods make compositional assumptions on the existence of global information, often neglecting localized information when deployed on tabular data ITW. In this work, we represent multi-view data as graphs and their dependencies as learnable edge weights. Moreover, we propose LEGATO, a novel autoencoder that learns a latent graph as a decomposable representation, where each of the latent components specializes in learning different aspects of localized information. Our method empirically demonstrated its effectiveness in learning representations on traditional multi-view tasks but excelled on ITW multi-view datasets with more complex localized dependencies. **Future works.** We see several directions for future research. One avenue is the development of better GNN or attention mechanisms tailored to capture localized dependencies more effectively. Additionally, investigating advanced optimization strategies, regularization techniques, and loss functions that account for the specific challenges of multi-view learning in tabular data could lead to improved model performance and generalization. Lastly, while we used an unsupervised reconstruction loss, we believe that the incorporation of more advanced semi- and self-supervised objectives can better leverage unlabeled data to enhance representation learning.

Acknowledgements

We thank the anonymous ICML reviewers as well as members of the van der Schaar lab for many insightful comments and suggestions. Tennison Liu would like to thank AstraZeneca for their sponsorship and support. Jeroen Berrevoets thanks W.D. Armstrong Trust for their support. This work is also supported by the National Science Foundation (NSF, grant number 1722516) and the Office of Naval Research (ONR).

References

Akaho, S. A kernel method for canonical correlation analysis. *arXiv preprint cs/0609071*, 2006.

Alguwaizani, A. Degeneracy on k-means clustering. *Electronic Notes in Discrete Mathematics*, 39:13–20, 2012.

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. Deep canonical correlation analysis. In *International conference on machine learning*, pp. 1247–1255. PMLR, 2013.

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Benton, A., Khayrallah, H., Gujral, B., Reisinger, D. A., Zhang, S., and Arora, R. Deep generalized canonical correlation analysis. *arXiv preprint arXiv:1702.02519*, 2017.

Buterez, D., Janet, J. P., Kiddle, S. J., Oglic, D., and Liò, P. Graph neural networks with adaptive readouts. In *Advances in Neural Information Processing Systems*, 2022.

Bycroft, C., Freeman, C., Petkova, D., Band, G., Elliott, L. T., Sharp, K., Motyer, A., Vukcevic, D., Delaneau, O., O'Connell, J., et al. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562(7726): 203–209, 2018.

Cai, C., Wang, D., and Wang, Y. Graph coarsening with neural networks. In *9th International conference on Learning Representations*, 2021.

Chapman, J. and Wang, H.-T. Cca-zoo: A collection of regularized, deep learning based, kernel, and probabilistic cca methods in a scikit-learn style framework. *Journal of*

Open Source Software, 6(68):3823, 2021. doi: 10.21105/joss.03823. URL https://doi.org/10.21105/joss.03823.

Daunhawer, I., Sutter, T. M., Chin-Cheong, K., Palumbo, E., and Vogt, J. E. On the limitations of multimodal vaes. In *International Conference on Learning Representations*, 2021.

De Cao, N. and Kipf, T. Molgan: An implicit generative model for small molecular graphs. *arXiv preprint arXiv:1805.11973*, 2018.

Defferrard, M., Bresson, X., and Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016.

Federici, M., Dutta, A., Forré, P., Kushman, N., and Akata, Z. Learning robust representations via multi-view information bottleneck. In *International Conference on Learning Representations*, 2019.

Gao, J., Lyu, T., Xiong, F., Wang, J., Ke, W., and Li, Z. Mgnn: A multimodal graph neural network for predicting the survival of cancer patients. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1697–1700, 2020.

Ghandi, M., Huang, F. W., Jané-Valbuena, J., Kryukov, G. V., Lo, C. C., McDonald III, E. R., Barretina, J., Gelfand, E. T., Bielski, C. M., Li, H., et al. Next-generation characterization of the cancer cell line encyclopedia. *Nature*, 569(7757):503–508, 2019.

Ghosh, P., Neufeld, A., and Sahoo, J. K. Forecasting directional movements of stock prices for intraday trading using lstm and random forests. *Finance Research Letters*, 46:102280, 2022.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.

Hajiramezanali, E., Hasanzadeh, A., Duffield, N., Narayanan, K., and Qian, X. Bayrel: Bayesian relational learning for multi-omics data integration. *Advances in Neural Information Processing Systems*, 33:19251–19263, 2020.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

Hasanzadeh, A., Hajiramezanali, E., Duffield, N., and Qian, X. Morel: Multi-omics relational learning. In *International Conference on Learning Representations*, 2021.

Huang, J. and Kingsbury, B. Audio-visual deep learning for noise robust speech recognition. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 7596–7599. IEEE, 2013.

Johnson, A. E., Bulgarelli, L., Shen, L., Gayles, A., Shammout, A., Horng, S., Pollard, T. J., Moody, B., Gow, B., Lehman, L.-w. H., et al. Mimic-iv, a freely accessible electronic health record dataset. *Scientific Data*, 10(1):1–9, 2023.

Kim, E.-S., Kang, W. Y., On, K.-W., Heo, Y.-J., and Zhang, B.-T. Hypergraph attention networks for multimodal learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14581–14590, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Kipf, T., Fetaya, E., Wang, K.-C., Welling, M., and Zemel, R. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pp. 2688–2697. PMLR, 2018.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016a.

Kipf, T. N. and Welling, M. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*, 2016b.

LeCun, Y., Kavukcuoglu, K., and Farabet, C. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE international symposium on circuits and systems*, pp. 253–256. IEEE, 2010.

Lee, M. and Pavlovic, V. Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2021.

Li, Y., Tarlow, D., Brockschmidt, M., and Zemel, R. Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*, 2015.

Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Liu, T., Qian, Z., Berrevoets, J., and van der Schaar, M. GOGGLE: Generative modelling for tabular data by learning relational structure. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fPVRcJqspu.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In *international conference on machine learning*, pp. 4114–4124. PMLR, 2019.

Ma, H., Bian, Y., Rong, Y., Huang, W., Xu, T., Xie, W., Ye, G., and Huang, J. Multi-view graph neural networks for molecular property prediction. *arXiv preprint arXiv:2005.13607*, 2020.

Mathieu, E., Rainforth, T., Siddharth, N., and Teh, Y. W. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pp. 4402–4412. PMLR, 2019.

Nagai, A., Hirata, M., Kamatani, Y., Muto, K., Matsuda, K., Kiyohara, Y., Ninomiya, T., Tamakoshi, A., Yamagata, Z., Mushiroda, T., et al. Overview of the biobank japan project: study design and profile. *Journal of epidemiology*, 27(Supplement_III):S2–S8, 2017.

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. Multimodal deep learning. In *ICML*, 2011.

Quiroga, R. Q., Kraskov, A., Koch, C., and Fried, I. Explicit encoding of multimodal percepts by single neurons in the human brain. *Current Biology*, 19(15):1308–1313, 2009.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., and Monfardini, G. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. Modeling relational data with graph convolutional networks. In *The Semantic Web: 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, Proceedings 15*, pp. 593–607. Springer, 2018.

Shi, Y., Paige, B., Torr, P., et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Shiokawa, Y., Date, Y., and Kikuchi, J. Application of kernel principal component analysis and computational machine learning to exploration of metabolites strongly associated with diet. *Scientific reports*, 8(1):3426, 2018.

Simonovsky, M. and Komodakis, N. Graphvae: Towards generation of small graphs using variational autoencoders. In *International conference on artificial neural networks*, pp. 412–422. Springer, 2018.

Snoek, J., Larochelle, H., and Adams, R. P. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

Srivastava, N. and Salakhutdinov, R. Learning representations for multimodal data with deep belief nets. In *International conference on machine learning workshop*, volume 79, 2012a.

Srivastava, N. and Salakhutdinov, R. R. Multimodal learning with deep boltzmann machines. *Advances in neural information processing systems*, 25, 2012b.

Sudlow, C., Gallacher, J., Allen, N., Beral, V., Burton, P., Danesh, J., Downey, P., Elliott, P., Green, J., Landray, M., et al. Uk biobank: an open access resource for identifying the causes of a wide range of complex diseases of middle and old age. *PLoS medicine*, 12(3):e1001779, 2015.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Suzuki, M., Nakayama, K., and Matsuo, Y. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

Tian, Y., Krishnan, D., and Isola, P. Contrastive multiview coding. In *European conference on computer vision*, pp. 776–794. Springer, 2020.

Tomczak, K., Czerwińska, P., and Wiznerowicz, M. Review the cancer genome atlas (tcga): an immeasurable source of knowledge. *Contemporary Oncology/Współczesna Onkologia*, 2015(1):68–77, 2015.

Tsai, Y.-H. H., Liang, P. P., Zadeh, A., Morency, L.-P., and Salakhutdinov, R. Learning factorized multimodal representations. *arXiv preprint arXiv:1806.06176*, 2018.

van Breukelen, M., Duin, R. P., Tax, D. M., and Den Hartog, J. Handwritten digit recognition by combined classifiers. *Kybernetika*, 34(4):381–386, 1998.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Vrigkas, M., Nikou, C., and Kakadiaris, I. A. A review of human activity recognition methods. *Frontiers in Robotics and AI*, 2:28, 2015.

Wang, D., Cui, P., and Zhu, W. Structural deep network embedding. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1225–1234, 2016.

Wang, W., Arora, R., Livescu, K., and Bilmes, J. On deep multi-view representation learning. In *International conference on machine learning*, pp. 1083–1092. PMLR, 2015.

Wei, Y., Wang, X., Nie, L., He, X., Hong, R., and Chua, T.-S. Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 1437–1445, 2019.

Wolff, J., Klein, T., Nabi, M., Krishnan, R. G., and Nakajima, S. Mixture-of-experts vaes can disregard variation in surjective multimodal data. *arXiv preprint arXiv:2204.05229*, 2022.

Wu, M. and Goodman, N. Multimodal generative models for scalable weakly-supervised learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Xu, C., Tao, D., and Xu, C. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634*, 2013.

Xu, Y. and Cohen, S. B. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1970–1979, 2018.

Yan, X., Hu, S., Mao, Y., Ye, Y., and Yu, H. Deep multi-view learning methods: a review. *Neurocomputing*, 448: 106–129, 2021.

Ye, T., Wang, T., McGuinness, K., Guo, Y., and Gurrin, C. Learning multiple views with orthogonal denoising autoencoders. In *International Conference on Multimedia Modeling*, pp. 313–324. Springer, 2016.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W., and Leskovec, J. Hierarchical graph representation learning with differentiable pooling. *Advances in neural information processing systems*, 31, 2018.

You, J., Ying, R., Ren, X., Hamilton, W., and Leskovec, J. Graphrnn: Generating realistic graphs with deep auto-regressive models. In *International conference on machine learning*, pp. 5708–5717. PMLR, 2018.

Zhang, C., Liu, Y., and Fu, H. Ae2-nets: Autoencoder in autoencoder networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2577–2585, 2019.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., and Sun, M. Graph neural networks: A review of methods and applications. *AI Open*, 1:57–81, 2020.

Zhou, K., Dong, Y., Wang, K., Lee, W. S., Hooi, B., Xu, H., and Feng, J. Understanding and resolving performance degradation in deep graph convolutional networks. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2728–2737, 2021.

A. Feature Sets In-The-Wild

Multi-view observations contain multiple observations of the same phenomenon and can originate from different modalities (e.g. image and text) but also be multiple observations of the same modality (e.g. multiple tabular datasets). Multi-view learning is the method to integrate information from multiple senses to interact with the world—we see objects, hear sounds, smell odors, and feel texture. Neuroscience research has shown that the brain jointly integrates information from multiple origins and that such synthesis is crucial to reasoning even without explicit labels for multi-view observations (Quiroga et al., 2009). While humans can easily learn through multiple senses in an unsupervised way, training a machine with analogous capabilities is a more challenging task.

The classic multi-view inductive bias hypothesis suggests that views provide the same task-relevant information (Yan et al., 2021). This assumption is helpful for many problems encountered in image, speech, and text domains. However, this is because data collection procedures in these problems, for example, audio-visual speech recognition (Huang & Kingsbury, 2013) and image-caption models (Radford et al., 2021), are carefully controlled to ensure views align and provide the same task-relevant information. However, multi-view data collected in less-controlled, in-the-wild settings are rarely aligned to the same degree. This is especially the case in tabular feature sets, where the higher heterogeneity across feature sets obscures their relationships. Here we consider a few examples:

- **Biobanks.** Examples of this include UK Biobank (Bycroft et al., 2018) and Biobank Japan (Nagai et al., 2017). These are large-scale biomedical databases that gather a variety of information including physical measures, lifestyle data, cognition and hearing functions, biomarkers, genetic data, and health outcomes. In these precision health datasets, each view provides information on a different aspect of the patient's medical state. It is far more likely for different sources of information to manifest in different clusters of views than for the information to be globally shared across all views.

- **Multi-omics.** Examples of this include The Cancer Genome Atlas Program (Tomczak et al., 2015) and Cancer Cell Line Encyclopedia (Ghandi et al., 2019). These problems combine datasets of different omic groups for biological analysis, including genetic, RNA splicing, DNA methylation, histone H3 modification, microRNA expressions, and also subject lineage and ethnicity data, and therapeutics data. Specific disease biomarkers likely manifest in only certain omic groups.

- **Stock market** (Ghosh et al., 2022). The stock market is described from multiple measurements, including trading data of individual stocks, different sources of stock market news (e.g. tweets, financial reports), technical indicators, market indices, and wider economic indicators. Evidently, different stocks can be highly dependent on other stocks in the same industry and also industry indicators.

In these settings, learning representations that aggregate information globally across all views will not achieve the desired learning effect. Indeed, we argue that the larger number of view and more complex localized dependencies give rise to localized sources of information that exists in clusters of views.

B. Graph Neural Networks

Graph Neural Network (GNN) is a type of deep learning model that can operate on graph-structured data, such as a social network or a molecule. They use neural networks to learn and make predictions on the nodes and edges of the graph. A variety of GNNs have been proposed in recent years, including those employing convolutional networks (Defferrard et al., 2016; Hamilton et al., 2017), recurrent architectures (Li et al., 2015) and recursive networks (Scarselli et al., 2008). Most of these approaches can be generalized using the neural *message passing* proposed by Gilmer et al. (2017), where node representations are iteratively updated by aggregating features from neighboring nodes.

Neural message-passing algorithms can be mathematically described in the following architecture:

$$H^{(k)}, A^{(k)} = MP\left(A^{(k-1)}, H^{(k-1)}\right) \tag{9}$$

where $H^{(K)}$ are the node embeddings computed after k steps of message passing, $A^{(k)}$ is the adjacency matrix, and $MP(\cdot)$ is some message propagation function. There are many ways to implement the message passing function, but generally using a combination of linear transformations and non-linear activations. One popular model is the graph convolutional

network (GCN) (Kipf & Welling, 2016a), where the node-wise update can be described using:

$$h_i^{(k)} = f\left(W^{(k)} \sum_{j \in \mathcal{N}(i) \cup i} \frac{h_j^{(k-1)}}{\sqrt{\tilde{d}_j^{(k-1)} \tilde{d}_i^{(k-1)}}}\right) \tag{10}$$

where f is some non-linear function, and $\mathcal{N}(i)$ is the set of all neighboring nodes of j as indicated in $A^{(k-1)}$. $h_j^{(k-1)}$ is the embedding of the jth node in $H^{(k-1)}$ and $\tilde{d}_j^{(k-1)}$ is the jth row in $\tilde{D}$ where $\tilde{D} = \sum_j \tilde{A}_{ij}^{(k-1)}$. Finally, $W^{(k)}$ are the learnable weights of the layer k. While GCN applies the same transformation to each node embedding to compose messages, RGCN (Schlichtkrull et al., 2018) considers different edge types to result in different message transformations:

$$h_i^{(k)} = f\left(\sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|} W_r^{(k)} h_j^{(k-1)}\right) \tag{11}$$

where $\mathcal{R}$ denotes the set of edges types and $W_r^{(k)}$ is the transformation matrix for edge type r.

Graph autoencoders. Graph autoencoders are a variant of GNNs that map nodes into a compact latent space and decode graph information from the latent representations. They are mainly used to extract low-dimensional embeddings while preserving a graph's topological information. SDNE (Wang et al., 2016) uses a stacked autoencoder to learn a graph embedding that preserves first and second-order proximity in the graph. VGAE (Kipf & Welling, 2016b) encodes both structural information and node feature information at the same time to learn *node embeddings*. An inner product measure on node embeddings to recover graph structural information, encoding the intuition that nodes that are closely connected in the graph should have similar representations. GraphRNN (You et al., 2018) follows a similar intuition to encode each latent node recursively for the purposes of dynamic graph generation. (Simonovsky & Komodakis, 2018) embeds a graph into a single vector representation, which is then used to reconstruct both the adjacency matrix and the node feature matrix. Existing works focus on learning graph embeddings given an input graph, but they are inherently "flat" and do not learn hierarchical representations. Perhaps similar to our work, Liu et al. (2023) used graph autoencoders to learn a generative model for tabular data but learned a flat graph to model dependencies present in a single feature set.

C. Implementation Details

C.1. Training and Hyperparameters

Training. All models are implemented in PyTorch. The data is split 60-20-20 into an unlabeled training set, labeled training set, and test set respectively, and all reported results are averaged over 10 runs, where different data splits are sampled for each run. All experiments are run on an NVIDIA Tesla K40C GPU.

Hyperparameters. Models are trained using the Adam (Kingma & Ba, 2014) with default parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. For all experiments, we use batch size of 64, but tune the learning rate $\eta \in \{0.001, 0.01, 0.1\}$ and weight decay $\in \{0.001, 0.01, 0.1\}$. These and other architecture-specific hyperparameter settings (specific hyperparameters discussed below) are searched using Bayesian Optimization (Snoek et al., 2012) with a search budget of 10 runs, and where the search objective is the validation set loss. Additionally, we employ early stopping to terminate model training after 20 epochs of no improvement on the validation set, after which the best model is returned for evaluation.

C.2. Model Implementation

While we are agnostic to the specific GNN architecture employed in our `POOL` and `UNPOOL` layers, we implemented GCN (Kipf & Welling, 2016a) and RGCN (Schlichtkrull et al., 2018). Specifically, for RGCN, we employ the basis decomposition proposed in Schlichtkrull et al. (2018):

$$W_r^{(k)} = \sum_{b=1}^{B} a_r^k b V_b^{(k)} \tag{12}$$

Therefore, where the weights $W_r^{(k)}$ form a weighted combination of a basis transformation $V_b^{(k)}$ with coefficients $a_{rb}^{(k)}$. We choose $B = 5$, reducing the number of learnable parameters in our model.

The dimensionality of our latent graph is chosen to be $K' = K/2$, so the latent graph has half the number of nodes as the multi-view graph. We found this to be a robust setting that worked well in our experiments. Additionally, we add a node normalization layer after each layer (Ba et al., 2016; Zhou et al., 2021). This reduces the problem of node (or view) dominance when computing new node embeddings, allowing for comparable contributions to the weighted combination in Equation (6). This operation is formally expressed as: $\text{Norm}(h^k) = (h^k - \mu^k)/\sqrt{var^k}$, where μ^k and var^k are the mean and variance of h^k calculated per dimension over the mini-batch. Lastly, we consider $\alpha \in \{0.001, 0.01, 0.1\}$ and $\beta \in \{0.001, 0.01, 0.1\}$ for our unsupervised learning objective.

Encoder/decoder networks. For both our model and the baselines we compared against, we use view encoders/decoders with a single `ReLU`-activated hidden layer. We tune the dimensionality of the hidden representation by considering $d \in \{50, 60, 70, 80, 90, 100\}$. We use the same encoder/decoder architectures to ensure a fair comparison.

C.3. Baseline Implementation

In this subsection, we provide further details on the implementation of benchmarks we compare against, including **DCCAE** (Wang et al., 2015), **DGCCA** (Benton et al., 2017), **JMVAE** (Suzuki et al., 2016), **MVAE** (Wu & Goodman, 2018), **DMVAE** (Lee & Pavlovic, 2021) and **MIB** (Federici et al., 2019).

DCCAE (Wang et al., 2015) is trained using two objectives: CCA objective to encourage view embeddings to be similar, and a reconstruction objective, where λ is a weighting parameter used to trade off the two objectives. We tune $\lambda \in \{0, 0.5, 1\}$ and use $\varepsilon = 0.001$, which is the default setting used to regularize CCA calculations. As **DCCAE** is designed with two views in mind, we modify the CCA objective when we have more views $\sum_{i=1}^{K} \phi(h_i, h_1)$. We use the implementation by Chapman & Wang (2021), which is publicly available at `https://github.com/jameschapman19/cca_zoo`.

DGCCA (Benton et al., 2017) generalizes CCA objectives to more than two views. We similarly use $\varepsilon = 0.001$, tune $\lambda \in \{0, 0.5, 1\}$, and use the implementation by Chapman & Wang (2021).

JMVAE (Suzuki et al., 2016) integrates view embeddings using a neural network to infer a stochastic latent variable z. The latent variable is stochastic, and the model is trained using the ELBO loss. We use the implementation available at `https://github.com/masa-su/jmvae`.

MVAE (Wu & Goodman, 2018) integrates view embeddings using a product-of-expert (POE) model, $q(z|x) = \Pi_{k=1}^{K} q(h^k|x)$. We use the implementation available at `https://github.com/mhw32/multimodal-vae-public`.

DMVAE (Lee & Pavlovic, 2021) uses a VAE architecture and introduces a separate latent variable $\{z^i\}_{i=1}^{K}$ for each view. We use the publicly available implementation `https://github.com/seqam-lab/DMVAE`.

MIB (Federici et al., 2019) introduces a variational information bottleneck to discard information that is not shared between views, where a hyperparameter λ is introduced to bottleneck superfluous information. We consider $\lambda \in \{0, 0.5, 1\}$ and use the implementation at `https://github.com/mfederici/Multi-View-Information-Bottleneck`.

C.4. TCGA Preprocessing

We analyze 1-year mortality based on the comprehensive observations from multiple omics on 7295 cancer cell lines (i.e. samples) data consists of observations from 4 distinct views on each cell line across 3 different omics layers: 1. mRNA expressions, 2. DNA methylation, 3. microRNA expressions, and 4. reverse phase protein array.

For constructing multiple views and labels, the following datasets were downloaded from `http://gdac.broadinstitute.org`:

- DNA methylation (epigenomics): *Methylation_Preprocess.Level_3.2016012800.0.0.tar.gz*
- microRNA expression (transcriptomics): *miRseq_Preprocess.Level_3.2016012800.0.0.tar.gz*
- mRNA expression (transcriptomics): *mRNAseq_Preprocess.Level_3.2016012800.0.0.tar.gz*
- RPPA (proteomics): *RPPA_AnnotateWithGene.Level_3.2016012800.0.0.tar.gz*
- clinical labels: *Clinical_Pick_Tier1.Level_4.2016012800.0.0.tar.gz*

Time to death or censoring in clinical labels was converted to a binary label for 1-year mortality. We imputed missing values within the observed views with mean values. To focus our experiments on the integrative analysis and to avoid *curse-of-dimensionality* in the high-dimensional multi-omics data, we extracted low-dimensional representations (i.e., 100 features) using the kernel-PCA (with polynomial kernels) on each view (Shiokawa et al., 2018).

C.5. UK Biobank Preprocessing

We used data from the UK Biobank (Sudlow et al., 2015), a large prospective cohort of half a million men and women recruited between 2006-10 from across the UK with ongoing follow-up. As lung cancer screening is only considered in ever-smokers, we include individuals without a previous diagnosis of lung cancer at baseline who self-reported as current or former smokers. Lung cancer diagnosis were determined through linked national cancer registry, right censored at 31/07/2019 (Sudlow et al., 2015).

The lung cancer dataset is extracted from UK Biobank using the scripts provided in `https://github.com/callta/synthetic-data-analyses/tree/main/code` by executing preprocessing scripts sequentially. Additionally, we dropped all variables with more than 25% missingness and all rows with more than 1% missingness. We normalized continuous variables such their values lay between 0 and categorical variables were one-hot encoded. To manage missing data, we used mean imputation. Lastly, we extracted relevant 9 views by using the feature categorizations provided at `https://biobank.ctsu.ox.ac.uk/crystal/cats.cgi`. The specific variable names included in each view can be found in the `.json` files and the preprocessing instructions included in our code at `https://github.com/tennisonliu/LEGATO/tree/master/exps/biobank_exp`.