# Orthogonal Multi-Manifold Enriching of Directed Networks

**Ramit Sawhney***
IIIT Delhi
ramits@iiitd.ac.in

**Shivam Agarwal***
Manipal Institute of Technology
shivamag99@gmail.com

**Atula Tejaswi Neerkaje***
Manipal Institute of Technology

**Kapil Pathak***
IISc Bangalore

## Abstract

Directed Acyclic Graphs and trees are widely prevalent in several real-world applications. These hierarchical structures show intriguing properties such as scale-free and bipartite nature, with fine-grained temporal irregularities among nodes. Building on advances in geometrical deep learning, we explore a time-aware neural network to model trees and Directed Acyclic Graphs in multiple Riemannian manifolds of varying curvatures. To jointly utilize the strength of these manifolds, we propose **M**ulti-Manifold **R**ecursive **I**nteraction **L**earning (**MRIL**) on Directed Acyclic Graphs where we introduce an inter-manifold learning mechanism that recursively enriches each manifold with representations from sibling manifolds. We propose the integration of the Stiefel orthogonality constraint which stabilizes the training process in Riemannian manifolds. Through a series of quantitative and exploratory experiments, we show that our method achieves competitive performance and converges much faster on data spanning several domains.

## 1 Introduction

Abundant information in the form of graphs has been made available due to the explosive growth of social media (Fink et al., 2015; Tambuscio et al., 2015). Along with the individual event information, various patterns in events such as dialogues (Qiu et al., 2021; Zahiri and Choi, 2018; Poria et al., 2018), tweets (Zubiaga et al., 2016; Ma et al., 2017) etc., can be analyzed where the dependencies within such events can give us more information than the individual event. Many real-world structures such as social media networks (Tambuscio et al., 2015), retweet propagation trees (Ma et al., 2017), disease propagation trees (Chami et al., 2019) etc., can be studied and analyzed along with the intra-event dependencies and represented in the form of directed acyclic graphs (DAGs) or trees (Jiang et al., 2021). These graph structures provide additional structure over the individual node representations. In many of these applications, the graphs are characterized by irregular structures as well as distinct time-stamps of the events. To illustrate our intuition, Figure 1 exemplifies fake news detection in tweets with temporal context along with cardinality information. The users' preferences, their influence, and time of occurrence of the posts may provide useful context about the original post. Hence, the cardinality and temporal variation among the graph nodes are potentially useful in learning generalized node representations for such applications (Dou et al., 2021).

Graph structures such as retweet propagation trees (Ma et al., 2017), dependency parse trees (Socher et al., 2013) etc., have hierarchical and scale-free nature (see Figure 1). Due to the scale-free nature, these structures with their node representations embedded in the Euclidean space may suffer major distortions (Aparicio et al., 2015; Chen et al., 2012a). The quality of such learned representations is determined by whether the geometry of embedding space of the representations matches with the structure of the data (Gu et al., 2019). Spherical embed-

* Indicates equal contribution. Proceedings of the 25$^{th}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. 

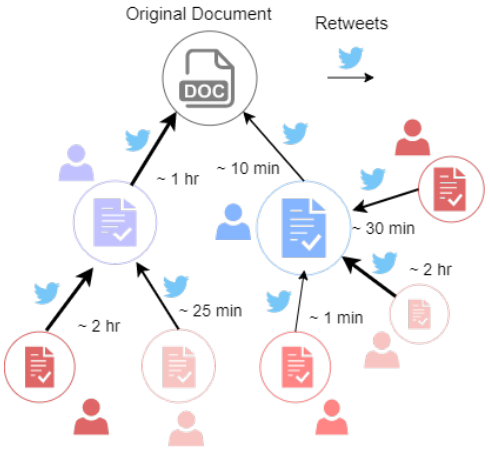



Figure 1: Illustration of the cardinal influence and temporal context, specifically in fake news detection with scale-free network settings. The original document is a news article and other nodes depict the retweets to the original post.

dings perform well under noisy conditions with low distortion on real-world data (Wilson et al., 2014). Poincaré embeddings capture hyperbolic properties in real-world graphs by learning shallow embeddings with hyperbolic distance metrics and Riemannian optimization (Nickel and Kiela, 2017). In real-world applications, graph data may have varying structure, and single-manifold embeddings may not capture the graph information with minimum distortion (Krioukov et al., 2010). Hence, an interaction mechanism among different manifolds such as Euclidean, Poincaré, and Spherical will capture a wider range of curvatures than single-manifold embeddings. Moreover, for many applications such as retweet propagation (Ma et al., 2017), conversation trees, and dialogue structures (Shen et al., 2021), a challenge may arise due to exploding and vanishing gradients while learning in these manifolds due to long-term dependencies in the graphs.

Building on previous work (Gu et al., 2019), we propose **M**ulti-Manifold **R**ecursive **I**nteraction **L**earning (**MRIL**) on Directed Acyclic Graphs (§2). Here, we introduce an inter-manifold learning mechanism (§2.2) that produces enriched node representations from Euclidean, Poincaré, and Spherical manifolds for learning non-trivial geometric properties from dynamically changing manifolds along with the Stiefel manifold constraint (§2.3). To account for the influence of every predecessor of the given node, we introduce a cardinality preserving attention mechanism in each of the manifolds (§2.1). We also add a time-aware component to capture temporal irregularities between nodes (§2.1). We evaluate our model on various tasks such as emotion recognition in conversations, fake news detection, rumour detection, and fine-grained sentiment classification to demonstrate the effectiveness of our model (§4). Through ablative qualitative and quantitative analyses, we validate the importance of each model component such as manifold combinations (§4.3), gated recursive interaction (§4.4), and cardinality and time-aware components (§4.2). We summarize our contributions as follows:

- We propose **MRIL**: **M**ulti-Manifold **R**ecursive **I**nteraction **L**earning over directed acyclic graphs and trees, where we introduce an inter-manifold learning mechanism to produce enriched node representations through gated recursive interaction learning.
- To address cardinality and temporal irregularities, we integrate a cardinality preserving attention along with time-aware components during recursive information flow through a generalized Recursive Riemannian Transition (RRT) Network.
- To stabilize the training process, we integrate the Stiefel manifold constraint during multi-manifold learning, which helps to mitigate the exploding and vanishing gradient problem, and achieve faster convergence.
- Through extensive experiments on various tasks such as emotion recognition in conversations, fake news detection, rumour detection, fine-grained sentiment classification, we show the applicability of **MRIL** that achieves competitive performance on these tasks.

## 2 Methodology

Let $t = (V, E)$ denote a directed acyclic graph with nodes $V$ and edges $E$, where the directed edge $e_{jk} = (k, j)$ connects the node $k$ with its successor node $j$. The set of direct predecessors of node $j$ is given by $\mathbb{C}(j)$ and feature vector as $x_j^{\mathcal{M}}$. Let a subgraph of $t$ be denoted by $t_j$ having a node $j$ without outgoing edges.

### 2.1 Recursive Riemannian Transitions

The role of the Riemannian Recursive Transition Network (RRT) is described by its ability to recursively encode information about every single node in $t_j$ through bottom-up message passing. As shown in

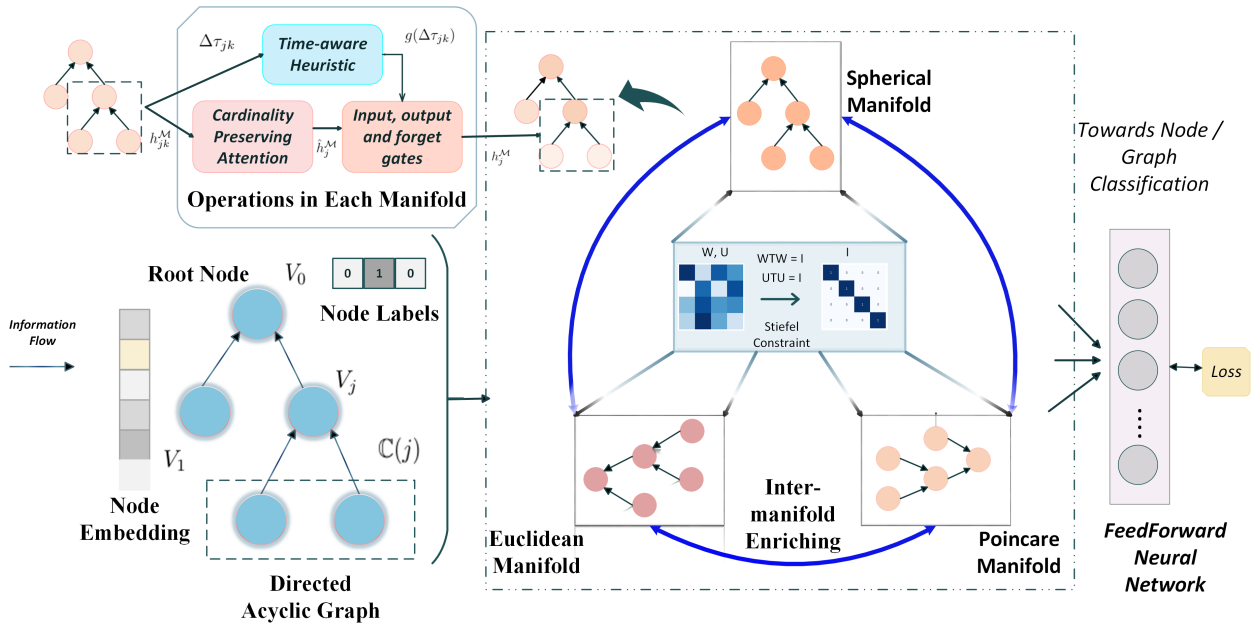



Figure 2: An overview of the model components of **MRIL**: Riemannian Recursive Transition Network (RRT) and the Gated Recursive Interaction Learning mechanism. For a given manifold, RRT takes a node and its predecessor nodes' representation as input and returns a cardinality and time-aware manifold-specific node representation. In the Gated Recursive Interaction mechanism, we introduce an inter-manifold enriching mechanism for each manifold over a shared set of weight matrices which gives enriched node representations for downstream tasks.

Figure 2, starting with the nodes with no incoming edges, every node $j$ is processed using its predecessor nodes $\mathbb{C}(j)$ in a recursive manner. To capture diverse structural properties of the graph, we model them in three component spaces: Poincaré ($\mathbb{P}$), Spherical ($\mathbb{S}$), and Euclidean ($\mathbb{R}$), which can be considered as Riemannian manifolds in the gyrovector space. We will first briefly describe general mathematical preliminaries on Riemannian manifolds before describing the network architecture and transition equations.

**Riemannian Manifolds and Gyrovector Spaces:** A Riemannian manifold in the gyrovector space is a smooth manifold denoted by $(\mathcal{M}, g_x^{\mathcal{M}})$, defined by $\mathcal{M} = \{x \in \mathbb{R}^n | -\omega\|x\|_2^2 < 1\}$ where $\omega$ is the sectional curvature, and $g_x^{\mathcal{M}}$ is termed as the Riemannian metric. In order to perform mathematical operations in the manifold $\mathcal{M}$, any operand must be first projected onto the manifold. The mapping of a given tangent vector $v \in \mathcal{T}_x\mathcal{M}$ to a point $\exp_x^{\mathcal{M}}(v)$ on the manifold $\mathcal{M}$ is called the exponential map $\exp_x^{\mathcal{M}}(v)$ . The inverse of this operation is the logarithmic map $\log_x^{\mathcal{M}}(y)$, which maps a point $y \in \mathcal{M}$ to a point $\log_x^{\mathcal{M}}(y)$ on the tangent space at $x$. The gyrovector space provides an algebra characterized by Möbius Addition ($\oplus_\omega$), Möbius Matrix Multipliction ($\otimes_\omega$), and Möbius Pointwise Multiplication ($\odot_\omega$)[1]. We obtain Poincaré ball ($\mathbb{P}$), Euclidean ($\mathbb{R}$) and Spherical ($\mathbb{S}$) geometries when $\omega < 0$, $\omega = 0$ and $\omega > 0$ respectively. Using the above preliminaries, we will first define our network on a generalized gyrovector space $\mathcal{M}$.

Using different values of the curvature $\omega$, the generalized RRT can be instantiated in the Poincaré, Spherical, and Euclidean manifolds. Hence, the RRT can be abstracted as a single step in the bottom-up traversal (Figure 2), which takes in a node $j$, its predecessors $\mathbb{C}(j)$, the parameter $\omega$, and returns the hidden state $h_j^{\mathcal{M}}$ and cell memory $c_j^{\mathcal{M}}$ of $j$ in the manifold $\mathcal{M} \in \{\mathbb{P}, \mathbb{S}, \mathbb{R}\}$, given by,

$$h_j^{\mathcal{M}}, c_j^{\mathcal{M}} = \text{RRT}(j, \mathbb{C}(j); \omega) \tag{1}$$

We will now describe each individual component of the RRT in detail.

**Cardinality Preserving Attention:** Across several domains such as dialogue modelling, sentiment

[1]We define mathematical operations in detail in the Appendix.

analysis and conversation threads, each node $j$ may show varied number of interactions and relations (Shen et al., 2021; Zubiaga et al., 2016), where each predecessor node $k \in \mathbb{C}(j)$ may exert diverse influences on node $j$. Let $h_k^{\mathcal{M}}, c_k^{\mathcal{M}} = \text{RRT}(k, \mathbb{C}(k); \omega)$ denote the previous hidden and memory states of node $k, \forall k \in \mathbb{C}(j)$. We seek to aggregate the final hidden states $h_k^{\mathcal{M}}$ of each predecessor, to obtain a combined representation of all the predecessors of node $j$. To do so, we use a Cardinality Preserving Attention mechanism to simultaneously capture the cardinality information (Zhang and Xie, 2020) of the direct predecessor set $\mathbb{C}(j)$. We compute the attention coefficients $\alpha_{jk}, \forall k \in \mathbb{C}(j)$ based on the geodesic distance $d_{\mathcal{M}}$ between node $j$ and node $k$, given as,

$$\alpha_{jk} = \underset{k \in \mathbb{C}(j)}{\text{softmax}}\left(-\lambda d_{\mathcal{M}}(h_k^{\mathcal{M}}, x_j^{\mathcal{M}}) \oplus_{)}\right) \tag{2}$$

where $\lambda$ is a learnable parameter. To calculate the weighted aggregation of predecessor hidden states, we integrate the cardinality preserving mechanism with the generalization of weighted averages over gyrovector spaces. The cardinality-aware aggregated hidden state $\tilde{h}_j$ is obtained based on the Möbius gyromidpoint (Ungar, 2008), given by,

$$\tilde{h_j}^{\mathcal{M}} = |\mathbb{C}(j)| \sum_{k \in \mathbb{C}(j)} \frac{1}{2} \boxtimes_\omega \frac{\alpha_{jk}\gamma(h_k^{\mathcal{M}})}{\sum_{l \in \mathbb{C}(j)} \alpha_{jl}(\gamma(h_l^{\mathcal{M}}) - 1)} h_k^{\mathcal{M}} \tag{3}$$

where $|\mathbb{C}(j)|$ is the cardinality of the predecessor set, $\boxtimes_\omega$ is Möbius scalar multiplication, and $\gamma(\cdot) = (\frac{2}{1-\omega\|\cdot\|^2})$ is called the conformal factor. We will now describe how the the aggregated hidden state $\tilde{h}_j^{\mathcal{M}}$ is integrated with time-aware components to obtain a final time and cardinality aware hidden representation for node $j$ using a set of transition functions.

**Time-Aware Component:** Several domains, such as conversation trees, and information cascades on social media, are characterized by intricate temporal patterns and irregularities between nodes (Backstrom et al., 2013; Zubiaga et al., 2016). Let $\Delta\tau_{jk}$ denote the difference in timestamps $\tau_j$ and $\tau_k$ of nodes $j$ and $k, \forall k \in \mathbb{C}(j)$ respectively (such as timestamps of posts in conversation threads on social media). We follow existing work (Baytas et al., 2017) and use a time-aware heuristic function $g(\Delta\tau_{jk}) = 1/\Delta\tau_{jk}$. Using this heuristic, we get the time-aware adjusted memory state ${}^*c_k^{\mathcal{M}}, \forall k \in \mathbb{C}(j)$, given by,

$${}^S c_k^{\mathcal{M}} = \exp_o{}^{\mathcal{M}}(\tanh(\log_o{}^{\mathcal{M}}(\boldsymbol{W}^{(c)} \otimes_\omega c_k^{\mathcal{M}}))) \tag{4}$$

$${}^* c_k^{\mathcal{M}} = -{}^S c_k^{\mathcal{M}} \oplus_\omega c_k^{\mathcal{M}} \oplus_\omega {}^S c_k^{\mathcal{M}} \odot_\omega g(\Delta\tau_{jk}) \tag{5}$$

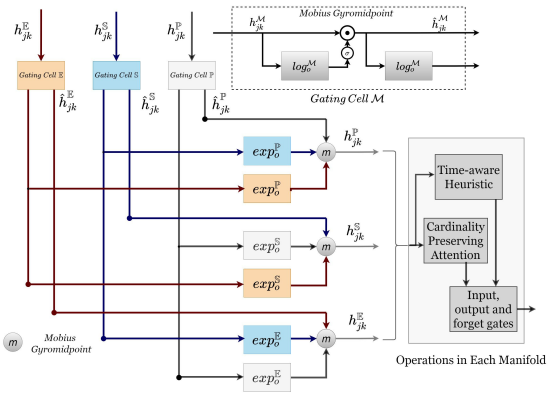



Figure 3: Gated Recursive Interaction Learning

where $\boldsymbol{W}^{(\cdot)}$ is a learnable parameter.

Using the aggregated predecessor hidden state $\tilde{h}_j$, we define the input gate $i_j$, output gate $o_j$ and intermediate cell state $u_j$. We then implement multiple forget gates $f_{jk}$ to selectively incorporate information for each predecessor node $k$. We represent these equations as,

$$i_j^{\mathcal{M}} = \sigma(\log_o{}^{\mathcal{M}}(\boldsymbol{W}^{(i)} \otimes_\omega x_j^{\mathcal{M}} \oplus_\omega \boldsymbol{U}^{(i)} \otimes_\omega \tilde{h_j}^{\mathcal{M}})) \tag{6}$$

$$o_j^{\mathcal{M}} = \sigma(\log_o{}^{\mathcal{M}}(\boldsymbol{W}^{(o)} \otimes_\omega x_j^{\mathcal{M}} \oplus_\omega \boldsymbol{U}^{(o)} \otimes_\omega \tilde{h_j}^{\mathcal{M}})) \tag{7}$$

$$u_j^{\mathcal{M}} = \tanh(\log_o{}^{\mathcal{M}}(\boldsymbol{W}^{(u)} \otimes_\omega x_j^{\mathcal{M}} \oplus_\omega \boldsymbol{U}^{(u)} \otimes_\omega \tilde{h_j}^{\mathcal{M}})) \tag{8}$$

$$f_{jk}^{\mathcal{M}} = \sigma(\log_o{}^{\mathcal{M}}(\boldsymbol{W}^{(f)} \otimes_\omega x_j^{\mathcal{M}} \oplus_\omega \boldsymbol{U}^{(f)} \otimes_\omega h_k^{\mathcal{M}}) \tag{9}$$

Finally, we selectively incorporate information from forget gates $f_{jk}^{\mathcal{M}}$ of the predecessor nodes with the temporal information from the adjusted cell memories ${}^*c_k^{\mathcal{M}}$, by combining the forget gates and the adjusted cell memories of predecessor nodes to obtain the cell memory state $c_j^{\mathcal{M}}$. The hidden state of node $j$ in the manifold $\mathcal{M}$ is obtained using the output gate $o_j^{\mathcal{M}}$ and cell memory state $c_j^{\mathcal{M}}$, given by,

$$c_j^{\mathcal{M}} = i_j^{\mathcal{M}} \odot_\omega u_j^{\mathcal{M}} \oplus_\omega \sum_k f_{jk}^{\mathcal{M}} \odot_\omega {}^* c_k^{\mathcal{M}} \tag{10}$$

$$h_j^{\mathcal{M}} = o_j^{\mathcal{M}} \odot \tanh(\log_o{}^{\mathcal{M}}(c_j^{\mathcal{M}})) \tag{11}$$

## 2.2 Gated Recursive Interaction Learning

To characterize varied spatial and structural features shown by real-world graphs (Chen et al., 2012b; Saito et al., 2012; Gu et al., 2019), we seek to learn rich structural representations at each level of the graph on multiple manifolds. With this motive, we instantiate the Recursive Riemannian Transitions over 3 manifolds, namely Euclidean ($\mathbb{R}$),

Poincaré ($\mathbb{P}$) and Spherical ($\mathbb{S}$), over a shared set of weight matrices $\boldsymbol{W}^{(\cdot)}$. Let $\mathbb{M} = \{\mathbb{P}, \mathbb{S}, \mathbb{R}\}$ represent the set of available manifolds . Using the hidden states $h_k^{\mathcal{M}}, \forall \mathcal{M} \in \mathbb{M}$ obtained as $h_k^{\mathcal{M}}, c_k^{\mathcal{M}} = \text{RRT}(k, \mathbb{C}(k); \omega)$ of the predecessor nodes $k, \forall k \in \mathbb{C}(j)$, we recursively enrich subgraph $t_j$ in each manifold with varied representations from every other manifold before applying RRT operations, as shown in Figure 3. To control the influence of each manifold, we first apply a sigmoid gating function on the hidden states of the Poincaré ($h_k^{\mathbb{P}}$), Euclidean ($h_k^{\mathbb{R}}$) and Spherical ($h_k^{\mathbb{S}}$) manifolds to obtain gated tangent hidden states $\hat{h}_k^{\mathcal{M}}, \forall k \in \mathbb{C}(j), \forall \mathcal{M} \in \mathbb{M}$ as,

$$\hat{h}_k^{\mathcal{M}} = \log_{\text{o}}{}^{\mathcal{M}}(\sigma(\boldsymbol{W}^{(p)} \otimes h_k^{\mathcal{M}}) \odot h_k^{\mathcal{M}}) \tag{12}$$

Next, we adaptively enrich predecessor hidden states from every manifold with diverse representations from sibling manifolds. By using sigmoid gates, each hidden state is then adaptively weighted with hidden states from every other sibling manifold through the Möbius gyromidpoint to obtain an enriched representation of $h_k^{\mathcal{M}}, \forall \mathcal{M} \in \mathbb{M}$, given as,

$$h_k^{\mathcal{M}} = |\mathbb{M}| \sum_{\mathcal{M}' \in \mathbb{M}} \frac{1}{2} \boxtimes_{\omega} \frac{\gamma(\exp_{\text{o}}{}^{\mathcal{M}}(\hat{h}_k^{\mathcal{M}'})) \exp_{\text{o}}{}^{\mathcal{M}}(\hat{h}_k^{\mathcal{M}'})}{\sum\limits_{\mathcal{M}'' \in \mathbb{M}} (\gamma(\exp_{\text{o}}{}^{\mathcal{M}}(\hat{h}_k^{\mathcal{M}''})) - 1)} \tag{13}$$

These hidden states are then used in the Riemannian Recursive Transition network to obtain hidden states of parent node $j$, given as,

$$\begin{aligned} h_j^{\mathbb{P}}, c_j^{\mathbb{P}} &= \text{RRT}(j, \mathbb{C}(j); \omega < 0) \\ h_j^{\mathbb{R}}, c_j^{\mathbb{R}} &= \text{RRT}(j, \mathbb{C}(j); \omega = 0) \\ h_j^{\mathbb{S}}, c_j^{\mathbb{S}} &= \text{RRT}(j, \mathbb{C}(j); \omega > 0) \end{aligned} \tag{14}$$

The final hidden state $h_j^{\mathbb{F}}, \forall j \in t$ is obtained by pooling the logarithmic projections of the 3 manifold hidden states. We refer to the full model as **MRIL**: **M**ulti-Manifold **R**ecursive **I**nteraction **L**earning on Directed Acyclic Graphs, given as,

$$h_j^{\mathbb{F}} = \text{MRIL}(t_j) \tag{15}$$

where $t_j$ is the subgraph of $t$ with $j$ as a sink node.

### 2.3 Stiefel Constrained Optimization

Modelling dialogue structures and online conversation threads require recursively capturing information over long-term dependencies (Majumder et al., 2019; Aragón et al., 2017). As a consequence, a well-documented challenge arises in the form of exploding and vanishing gradients, which has been observed in recursive models such as RNNs (Hochreiter and Schmidhuber, 1997), GRUs (Wolter and Yao, 2018), and even LSTMs (Kanuparthi et al., 2019). A similar problem arises for the RRT, wherein the cell state update in eq. (10) evolves through a linear recursive equation, while all other states are bounded by sigmoid and tanh activations, causing an imbalance in gradient magnitudes leading to vanishing and exploding gradients over long-term dependencies (Kanuparthi et al., 2019). This vanishing problem can be controlled by keeping network weight matrices $\boldsymbol{W}$ close to orthogonal (Arjovsky et al., 2016).

The Stiefel manifold is a Riemannian manifold consisting of orthogonal matrices, given by $\mathbb{V} = \{\boldsymbol{X} \in \mathbb{R}^{m \times n} : \boldsymbol{X}^T \boldsymbol{X} = \boldsymbol{I}_n\}$. Contrary to existing work on Riemannian manifolds and hyperbolic learning, every network weight matrix $\boldsymbol{W}$ in **MRIL** is constrained to the Stiefel manifold. We use QR decomposition (Absil et al., 2009) to obtain an initial random weight matrix on the Stiefel manifold. QR decomposition is the decomposition of a given matrix $\boldsymbol{A}$ into the product of an orthogonal matrix $\boldsymbol{Q}$ and upper triangular matrix $\boldsymbol{R}$. The decomposition equation is formally stated as $\boldsymbol{A} = \boldsymbol{QR}$. For network matrix initialization, a random weight matrix $\boldsymbol{A}$ is first initialized. Next, it is decomposed into $\boldsymbol{A} = \boldsymbol{WR}$ using QR decomposition to obtain $\boldsymbol{W}$, the initialized weight matrix on the Stiefel manifold.

The optimization for **MRIL** proceeds through gradient descent, wherein the computed gradient vectors are projected onto the Stiefel manifold before the gradient update step. Let $\boldsymbol{G} = \frac{\partial \mathcal{L}}{\partial \boldsymbol{W}}$ denote the computed gradient of the objective function $\mathcal{L}$ with respect to weight matrix $\boldsymbol{W}$. During gradient descent, the euclidean gradient is first projected onto the manifold before the weight matrix is updated (Meghwanshi et al., 2018) . The projection of the gradient $\boldsymbol{G}$ on the tangent space at $\boldsymbol{W}$ is denoted as follows,

$$\boldsymbol{G}_{\text{proj}} = \boldsymbol{G} - \frac{\boldsymbol{W}}{2} \left( \boldsymbol{W}^T \boldsymbol{G} + \boldsymbol{G}^T \boldsymbol{W} \right) \tag{16}$$

## 3 Experiments

### 3.1 Datasets

In this section, we explore various classification tasks involving DAGs and trees. We present a summary of dataset properties in Table 2. We explore two kinds of tasks, namely node classification and sink (root) node classification.

Table 1: Performance comparison over graph and non-graph based methods, averaged over 5 independent runs. * indicates that the result is significantly ($p < 0.005$) better than the existing state-of-the-art methods under Wilcoxon's signed rank test. **Bold** and *italics* denotes best and second best performance respectively.

| Type | Models | MELD | IEMOCAP | EmoryNLP | UPFD-Pol | UPFD-Gos | SST-5 | Disease | PHEME | Twitter16 |
|---|---|---|---|---|---|---|---|---|---|---|
| No Structure | MLP | 0.37 | 0.44 | 0.21 | 0.76 | 0.75 | 0.42 | 0.31 | 0.60 | 0.63 |
| | LSTM | 0.41 | 0.46 | 0.22 | 0.71 | 0.86 | 0.47 | 0.33 | 0.62 | 0.70 |
| Graph Structure | GCN | 0.61 | 0.61 | 0.35 | 0.80 | 0.92 | 0.45 | 0.70 | 0.70 | 0.68 |
| | GAT | 0.61 | 0.62 | 0.34 | 0.81 | 0.91 | 0.44 | 0.70 | 0.71 | 0.71 |
| | HGCN | *0.63* | *0.63* | **0.39** | 0.81 | *0.94* | 0.45 | **0.74** | 0.71 | 0.71 |
| | HGAT | *0.63* | **0.67** | 0.33 | 0.82 | 0.93 | 0.45 | *0.72* | *0.73* | 0.72 |
| | TreeLSTM | 0.57 | 0.58 | *0.36* | 0.79 | 0.84 | 0.50 | 0.52 | 0.71 | 0.69 |
| | AttnTreeLSTM | 0.59 | 0.60 | *0.36* | *0.83* | 0.87 | *0.54* | 0.59 | 0.72 | *0.71* |
| | **MRIL(Ours)** | **0.64*** | **0.67** | **0.39** | **0.85*** | **0.97*** | **0.57*** | **0.74** | **0.77*** | **0.79*** |

Table 2: Dataset Statistics

| Dataset | Nodes | Edges | Avg. Graph Size | # Classes |
|---|---|---|---|---|
| MELD | 13.7k | 29.6k | 9.57±5.79 | 7 |
| IEMOCAP | 10k | 19.7k | 66.8±22.32 | 6 |
| EmoryNLP | 12.6k | 33.9k | 14.05±5.61 | 7 |
| UPFD-Pol | 41k | 40.7k | 130.74±130.55 | 2 |
| UPFD-Gos | 314.2k | 308.7k | 57.51±45.23 | 2 |
| SST | 442.6k | 430.7k | 37.33±18.41 | 5 |
| Disease | 1k | 1k | 1k | 2 |
| PHEME | 90.5k | 83.8k | 13.63±16.56 | 2 |
| Twitter16 | 6.8k | 6.6k | 16.62±24.73 | 2 |

**Node Classification:** **MELD** (Poria et al., 2018) , **IEMOCAP** (Busso et al., 2008) and **EmoryNLP** (Zahiri and Choi, 2018) are datasets which contain dialogues from TV shows structured as DAGs, where the task is to predict the emotion of each dialogue. We evaluate these tasks using Weighted F1 score. **Disease** (Chami et al., 2019) dataset shows a disease propagation tree, where node represents a state of being infected or not by SIR disease. For this task we use Macro F1 score. **SST-5** (Socher et al., 2013) is a corpus with fully labelled parsed trees. It allows fine-grained sentiment classification of the sentences based on their compositional structures. The metric for **SST** is accuracy. By following recursive topological bottom-up traversal, a label $\hat{y}_j$ is predicted for every single node $j$ in the graph by passing the output hidden state to a multi-layer perceptron (MLP), where the prediction step is given as,

$$\hat{y}_j = \text{MLP}(\text{MRIL}(t_j)) \tag{17}$$

**Root Classification:** **UPFD-Pol** and **UPFD-Gos** (Dou et al., 2021) consist of retweet propagation trees where the root node is a news piece and other nodes are users who retweeted it. The task is to predict whether the news piece is fake or not. Additional context in terms of reply tree structure with the time of posting is used. **PHEME** (Zubiaga et al., 2016) and **Twitter16** (Ma et al., 2017) are rumour prediction tasks on claims made on social media, where each arising conversation tree with the time of posting forms the claim, labelled as either a true or a false rumour. Following existing work, macro F1 is used as the metric for these tasks. The label $\hat{y}_0$ is only predicted for the root at the end of the bottom-up recursive traversals, with the final root node prediction step given as,

$$\hat{y}_0 = \text{MLP}(\text{MRIL}(t)) \tag{18}$$

**Baselines:** We compare **MRIL**[2] with various existing methods. Without regard to graph topology, **MLP** serves as the traditional feature-based neural network method. **LSTM** uses weak graph information in the from of a flattened graph structure. In tree/DAG-like methods, we compare performance of our model with **Tree-LSTM** (Tai et al., 2015) and **Attentive Tree-LSTM** (Ahmed et al., 2019). Further we also take graph-based structures in Euclidean space such as **GCN** (Kipf and Welling, 2017), **GAT** (Veličković et al., 2018) and their hyperbolic versions **HGCN** (Chami et al., 2019), **HGAT** (Gulcehre et al., 2019).

## 4 Results

### 4.1 Performance Comparison

We evaluate **MRIL** over various domains spanning dialogue modelling, social networks, and sentiment analysis in Table 1. We first observe that the graph-based approaches outperform MLP and LSTM methods, emphasizing the importance of structural information for improved feature representations for such tasks (Wu et al., 2020; Shen et al., 2021). Among approaches which leverage graph structure, models which utilize hyperbolic learning (**HGCN**, **HGAT**, **MRIL**) generally perform better

[2]We release the code at `https://github.com/atutej/MRIL`

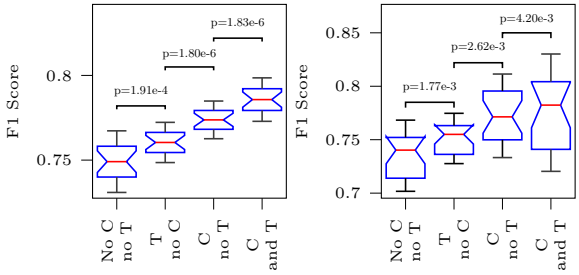
Table 3: Ablation study over manifold components of MRIL over 5 independent runs. **Bold** and *italics* denotes best and second best performance respectively. * and † indicate that the result is significant ($p < 0.005$) with respect to euclidean and non-euclidean single-manifold models, respectively under Wilcoxon's signed rank test.

| Model | MELD | IEMOCAP | EmoryNLP | UPFD-Pol | UPFD-Gos | SST-5 | Disease | PHEME | Twitter16 |
|---|---|---|---|---|---|---|---|---|---|
| Euclidean RRT | 0.56 | 0.63 | 0.35 | 0.78 | 0.91 | 0.52 | 0.61 | 0.69 | 0.71 |
| Poincare RRT | 0.61* | 0.64* | 0.36* | *0.82** | 0.92* | 0.53* | *0.73** | 0.72* | 0.75* |
| Stiefel RRT | 0.61* | 0.62 | 0.35 | 0.81* | 0.92* | *0.56** | 0.64* | 0.72* | 0.73* |
| Euclidean + Spherical + Stiefel | 0.62† | 0.63 | 0.36* | 0.77 | 0.93† | 0.55* | 0.63* | 0.73† | 0.72* |
| Euclidean + Poincare + Stiefel | *0.63*† | *0.65*† | *0.37*† | 0.81* | *0.96*† | 0.55* | 0.70* | *0.75*† | *0.77*† |
| **MRIL** | **0.64**† | **0.67**† | **0.39**† | **0.85**† | **0.97**† | **0.57**† | **0.74**† | **0.77**† | **0.79**† |

than euclidean methods due to improved representations of scale-free structures (Nekovee et al., 2007; Sala et al., 2018; Leskovec et al., 2007). We further observe that recursive approaches (**TreeLSTM**, **AttnTreeLSTM**, **MRIL**) perform better for root node classification tasks which have long-term dependencies. **MRIL** significantly ($p < 0.005$) outperforms existing methods on most tasks. We attribute the performance of **MRIL** to the following aspects: 1) Cardinality and Time Aware Components for capturing individual (Shen et al., 2021; Wu et al., 2020) and cardinal (Zhang and Xie, 2020; Zubiaga et al., 2016) influences of predecessor nodes, along with delicate temporal granularities (Wu et al., 2020; Fourney et al., 2017) and 2) Ability to learn representations in 3 diverse manifolds - Poincaré, Spherical, and Euclidean, for capturing diverse structural characteristics of real-world directed graphs (Wilson et al., 2014; Fushimi et al., 2011; Huang and Li, 2007) and 3) Gated Recursive Interaction Learning Mechanism which helps in recursively obtaining rich representations through controlled enriching of each manifold with sibling manifolds. We describe the effectiveness of these individual factors in the next sections.

## 4.2 Impact of Cardinalty and Temporal Components

We will now examine the individual impact of the cardinality aware attention mechanism and time aware component of **MRIL** in Figure 4. We first analyze the effect of the time-aware (TA) component, followed by cardinality preserving attention (CA), and finally both CA and TA. We observe significant ($p < 0.005$) improvements on incorporating the time-aware component, validating the ability of **MRIL** in capturing irregularities in temporal information existing in various domains such as rumour and fake-news detection (Fourney et al., 2017; Zubiaga et al., 2016). We next observe how the cardinality attention mechanism impacts the overall performance of the model. We observe significant performance improvements on incorporating the cardinality aware component, due to its ability to capture varied influences and rich representations of each predecessor node (Kaligotla et al., 2016), while simultaneously preserving varied cardinality of interactions between each node and its predecessor nodes (Zhang and Xie, 2020). Finally, we observe that the fully augmented **MRIL** achieves the best performance. This observation suggests that temporal and feature influences may have independent influences, and **MRIL** adaptively learns the best representations by dynamically utilizing information from both components to an extent.




(a) Twitter16 (b) PHEME

Figure 4: Confidence intervals over cardinality preserving attention (C) and time-aware (T) components. Results are averaged over 5 independent runs with the $p$-value under Wilcoxon's signed rank test.

## 4.3 Ablation Study

We probe the performance variations over individual manifolds in Table 3. We first observe that the Poincaré manifold offers significant ($p < 0.005$) improvements in performance across all tasks, emphasizing its effectiveness in capturing intricate scale-free dynamics observed in real-world graphs

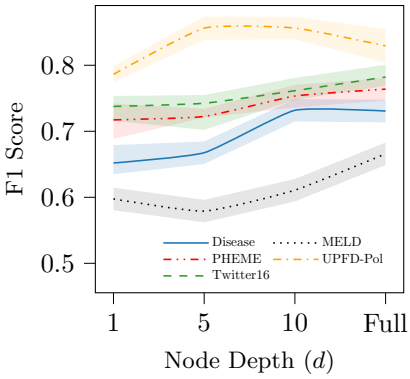
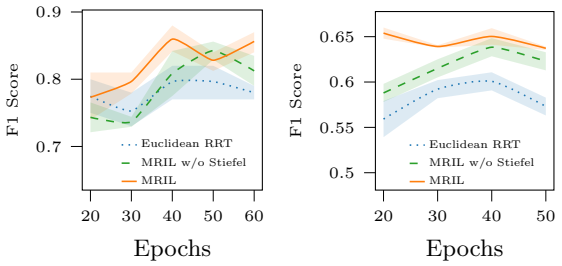



Figure 5: Performance sensitivity of gated interaction learning over increasing values of node depth cutoff ($d$) for which interaction is enabled. Shaded region indicates first standard deviation.




(a) UPFD-Pol (b) MELD

Figure 6: Performance sensitivity curves with increasing epochs for MRIL, MRIL without Stiefel, and the Euclidean RRT. Results are averaged over 5 independent runs. Shaded region indicates first standard deviation.

(Leskovec et al., 2007). We observe that inducing the Stiefel constraint on top of the Euclidean RRT offers further improvements, by providing a regularizing effect and further accelerating training (Arjovsky et al., 2016). The improvements can be attributed to each manifold component performing independent functions, wherein the interaction between Euclidean, Poincaré, or Spherical manifolds provide improved feature representations, while the Stiefel constraint stabilizes activations (Huang et al., 2018) in these manifolds for accelerated training. Finally, the significant performance improvement in full **MRIL** emphasizes the strength of **MRIL** through Stiefel-optimized dynamic learning of optimal representations of nodes at each level of the graph by controlled information flow from each manifold.

### 4.4 Impact of Gated Recursive Interaction

In Figure 5, we analyze the sensitivity of **MRIL** over the gated interaction mechanism. We study the performance changes by gradually enabling the recursive interaction mechanism only for nodes with depth less than some cutoff $d$. In particular, predecessor nodes with depth $>= d$ are not enriched with representations from sibling manifolds. When $d = 1$, **MRIL** degenerates into a simple ensemble over all the manifolds. We note that the increase in performance correlates with the depth cutoff $d$, showing that the enriching has a positive effect on the representational capacity of the network. Our observations on the positive impact of manifold enriching tie up with existing work (Gu et al., 2019). The best performance for almost all datasets is obtained when the interaction is enabled for all nodes, indicating the ability of **MRIL** to dynamically combine and learn rich representations over highly-influential long-term dependencies at each level of the graph through controlled information flow augmented from sibling manifolds.

### 4.5 Computational Complexity

We analyze the impact of the Stiefel constraint from an optimization perspective in Figure 6. We observe that **MRIL** indeed converges much quicker than its counterpart without the Stiefel constraint. Performance of the Euclidean RRT worsens for larger epochs, demonstrating the regularizing properties of the Stiefel manifold (Huang et al., 2018) and our proposed enriching mechanism. Our observations empirically confirm that the Stiefel constraint accelerates training, and we observe that **MRIL** converges 56% faster on average when compared to standard Riemannian optimization.

## 5 Conclusion

In this paper, we introduced **MRIL**, an approach to recursively learn enriched representations of deep structures such as trees and DAGs. We first explored the incorporation of time-aware components for capturing irregularities in time elapsed between nodes. To capture the size and diverse influences of each predecessor set on various manifolds, we proposed a generalized cardinality preserving attention on the gyrovector space. We then introduced a recursive gated interaction mechanism for controlled enrichment of contrasting representations between manifolds at each level of the graph. We demonstrated how **MRIL** captures long-term dependencies in deep structures by mitigating the van-

ishing gradient problem through constrained optimization on the Stiefel manifold. Through a series of exploratory experiments, we then showed that **MRIL** achieves competitive performance and converges faster than existing recursive and graph-based methods over various real-world datasets.

**Acknowledgements**

We thank Petar Veličković for reviewing the paper and providing useful feedback and support.

**References**

Absil, P.-A., Mahony, R., and Sepulchre, R. (2009). *Optimization algorithms on matrix manifolds*. Princeton University Press.

Ahmed, M., Samee, M. R., and Mercer, R. E. (2019). Improving tree-lstm with tree attention. In *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*, pages 247–254. IEEE.

Aparicio, S., Villazón-Terrazas, J., and Álvarez, G. (2015). A model for scale-free networks: Application to twitter. *Entropy*, 17(8):5848–5867.

Aragón, P., Gómez, V., and Kaltenbrunner, A. (2017). To thread or not to thread: The impact of conversation threading on online discussion. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11.

Arjovsky, M., Shah, A., and Bengio, Y. (2016). Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, pages 1120–1128. PMLR.

Backstrom, L., Kleinberg, J., Lee, L., and Danescu-Niculescu-Mizil, C. (2013). Characterizing and curating conversation threads: expansion, focus, volume, re-entry. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 13–22.

Baytas, I. M., Xiao, C., Zhang, X., Wang, F., Jain, A. K., and Zhou, J. (2017). Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74.

Busso, C., Bulut, M., Lee, C.-C., Kazemzadeh, A., Mower, E., Kim, S., Chang, J. N., Lee, S., and Narayanan, S. S. (2008). Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Chami, I., Ying, Z., Ré, C., and Leskovec, J. (2019). Hyperbolic graph convolutional neural networks. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Chen, W., Fang, W., Hu, G., and Mahoney, M. W. (2012a). On the hyperbolicity of small-world and tree-like random graphs. In Chao, K.-M., Hsu, T.-s., and Lee, D.-T., editors, *Algorithms and Computation*, pages 278–288, Berlin, Heidelberg. Springer Berlin Heidelberg.

Chen, W., Fang, W., Hu, G., and Mahoney, M. W. (2012b). On the hyperbolicity of small-world and tree-like random graphs. In Chao, K.-M., Hsu, T.-s., and Lee, D.-T., editors, *Algorithms and Computation*, pages 278–288, Berlin, Heidelberg. Springer Berlin Heidelberg.

Cui, Y., Jia, M., Lin, T.-Y., Song, Y., and Belongie, S. (2019). Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277.

Dou, Y., Shu, K., Xia, C., Yu, P. S., and Sun, L. (2021). User preference-aware fake news detection. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2051–2055, New York, NY, USA. Association for Computing Machinery.

Fink, C., Schmidt, A. C., Barash, V., Cameron, C. J., and Macy, M. W. (2015). Complex contagions and the diffusion of popular twitter hashtags in nigeria. *Social Network Analysis and Mining*, 6:1–19.

Fourney, A., Racz, M. Z., Ranade, G., Mobius, M., and Horvitz, E. (2017). Geographic and temporal trends in fake news consumption during the 2016 us presidential election. In *CIKM*, volume 17, pages 6–10.

Fushimi, T., Kubota, Y., Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2011). Speeding up bipartite graph visualization method. volume 7106, pages 697–706.

Gu, A., Sala, F., Gunel, B., and Ré, C. (2019). Learning mixed-curvature representations in product spaces. In *International Conference on Learning Representations*.

Gulcehre, C., Denil, M., Malinowski, M., Razavi, A., Pascanu, R., Hermann, K. M., Battaglia, P., Bapst, V., Raposo, D., Santoro, A., and de Freitas, N. (2019). Hyperbolic attention networks. In *Proceedings of the 7th International Conference on Learning Representations*.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Huang, L., Liu, X., Lang, B., Yu, A. W., Wang, Y., and Li, B. (2018). Orthogonal weight normalization: Solution to optimization over multiple dependent stiefel manifolds in deep neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Huang, W. and Li, C. (2007). Epidemic spreading in scale-free networks with community structure. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(01):P01014.

Jiang, J., Wang, A., and Aizawa, A. (2021). Attention-based relational graph convolutional network for target-oriented opinion words extraction. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1986–1997, Online. Association for Computational Linguistics.

Kaligotla, C., Yücesan, E., and Chick, S. E. (2016). The impact of broadcasting on the spread of opinions in social media conversations. In *Proceedings of the 2016 Winter Simulation Conference*, WSC '16, page 3476–3487. IEEE Press.

Kanuparthi, B., Arpit, D., Kerg, G., Ke, N. R., Mitliagkas, I., and Bengio, Y. (2019). h-detach: Modifying the LSTM gradient towards better optimization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of the 5th International Conference on Learning Representations*.

Krioukov, D., Papadopoulos, F., Kitsak, M., Vahdat, A., and Boguñá, M. (2010). Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106.

Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., and Hurst, M. (2007). Patterns of cascading behavior in large blog graphs. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 551–556. SIAM.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988.

Ma, J., Gao, W., and Wong, K.-F. (2017). Detect rumors in microblog posts using propagation structure via kernel learning. Association for Computational Linguistics.

Majumder, N., Poria, S., Hazarika, D., Mihalcea, R., Gelbukh, A., and Cambria, E. (2019). Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825.

Meghwanshi, M., Jawanpuria, P., Kunchukuttan, A., Kasai, H., and Mishra, B. (2018). Mctorch, a manifold optimization library for deep learning. *arXiv preprint arXiv:1810.01811*.

Nekovee, M., Moreno, Y., Bianconi, G., and Marsili, M. (2007). Theory of rumour spreading in complex social networks. *Physica A: Statistical Mechanics and its Applications*, 374(1):457–470.

Nickel, M. and Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Poria, S., Hazarika, D., Majumder, N., Naik, G., Cambria, E., and Mihalcea, R. (2018). Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Qiu, L., Liang, Y., Zhao, Y., Lu, P., Peng, B., Yu, Z., Wu, Y. N., and Zhu, S.-C. (2021). SocAoG: Incremental graph parsing for social relation inference in dialogues. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 658–670, Online. Association for Computational Linguistics.

Saito, K., Kimura, M., Ohara, K., and Motoda, H. (2012). Graph embedding on spheres and its application to visualization of information diffusion data. In *Proceedings of the 21st International Conference on World Wide Web*, pages 1137–1144.

Sala, F., De Sa, C., Gu, A., and Ré, C. (2018). Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR.

Shen, W., Wu, S., Yang, Y., and Quan, X. (2021). Directed acyclic graph network for conversational emotion recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint*

*Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1551–1560, Online. Association for Computational Linguistics.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Tambuscio, M., Ruffo, G., Flammini, A., and Menczer, F. (2015). Fact-checking effect on viral hoaxes: A model of misinformation spread in social networks. In *Proceedings of the 24th international conference on World Wide Web*, pages 977–982.

Ungar, A. A. (2008). A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks. In *Proceedings of the 6th International Conference on Learning Representations*.

Wilson, R. C., Hancock, E. R., Pekalska, E., and Duin, R. P. (2014). Spherical and hyperbolic embeddings of data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2255–2269.

Wolter, M. and Yao, A. (2018). Complex gated recurrent neural networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Wu, Z., Pi, D., Chen, J., Xie, M., and Cao, J. (2020). Rumor detection based on propagation graph neural network with attention mechanism. *Expert systems with applications*, 158:113595.

Zahiri, S. M. and Choi, J. D. (2018). Emotion detection on tv show transcripts with sequence-based convolutional neural networks. *ArXiv*, abs/1708.04299.

Zhang, S. and Xie, L. (2020). Improving attention mechanism in graph neural networks via cardinality preservation. In *IJCAI: proceedings of the conference*, volume 2020, page 1395. NIH Public Access.

Zubiaga, A., Liakata, M., Procter, R., Wong Sak Hoi, G., and Tolmie, P. (2016). Analysing how people orient to and spread rumours in social media by looking at conversational threads. *PloS one*, 11(3):e0150989.

# Supplementary Material: Orthogonal Multi-Manifold Enriching of Directed Networks

## A Riemannian Manifolds and Gyrovector Spaces

A Riemannian manifold in the gyrovector space is a smooth manifold denoted by $(\mathcal{M}, g_x^{\mathcal{M}})$, defined by $\mathcal{M} = \{x \in \mathbb{R}^n | -\omega\|x\|_2^2 < 1\}$ where $\omega$ is the sectional curvature, and $g_x^{\mathcal{M}}$ is termed as the Riemannian metric.

**Möbius Addition ($\oplus_\omega$)** The Möbius addition for a pair of points $x, y$ in $\mathcal{M}$ is defined as:

$$x \oplus_\omega y = \frac{(1 + 2\omega\langle x, y\rangle + \omega\|y\|^2)x + (1 - \omega\|x\|^2)y}{1 + 2\omega\langle x, y\rangle + \omega^2\|x\|^2\|y\|^2} \tag{19}$$

In particular, for $\omega$=0, the formula recovers the addition formula in Euclidean space.

**Exponential Map** maps a tangent vector $v$ in the tangent space $T_x\mathcal{M}$ to a point $\exp_x(v)$ on the manifold, given by:

$$\exp_x(v) := x \oplus_\omega \left(\tanh\left(\frac{\sqrt{\omega}\lambda_x\|v\|}{2}\right)\frac{v}{\sqrt{\omega}\|v\|}\right) \tag{20}$$

**Logarithmic Map** maps a point $y \in \mathcal{M}$ to a point $\log_x(y)$ on the tangent space at $x$,

$$\log_x(y) := \frac{2}{\sqrt{\omega}\lambda_x}\tanh^{-1}(\sqrt{\omega}\| - x \oplus_\omega y\|)\frac{-x \oplus_\omega y}{\| - x \oplus_\omega y\|} \tag{21}$$

**Möbius Vector Multiplication ($\otimes_\omega$)** The Möbius Vector Multiplication multiplies features $x \in \mathcal{M}^n$ by matrix $M \in \mathbb{R}^{n' \times n}$, defined as:

$$M \otimes_\omega x = \frac{1}{\sqrt{w}}\tanh(\frac{\|Mx\|}{\|x\|}\tanh^{-1}(\sqrt{w}\|x\|))\frac{Mx}{\|Mx\|} \tag{22}$$

**Möbius Pointwise Multiplication ($\odot_\omega$)** The Möbius Pointwise Multiplication multiplies $x, y \in \mathcal{M}^n$ element-wise, given by:

$$x \odot_\omega y = diag(\log_x^{\mathcal{M}}(x)) \otimes_\omega y \tag{23}$$

**Möbius Scalar Multiplication ($\boxtimes_\omega$)** The Möbius scalar multiplication with scalar $r$ and vector $x \in \mathcal{M}^n$ is given by the following formula:

$$r \boxtimes_\omega x = \frac{1}{\sqrt{w}}\tanh(r\tanh^{-1}(\sqrt{w}\|x\|))\frac{x}{\|x\|} \tag{24}$$

## B Experimental Settings

### B.1 Baselines

- MLP: Features of each labelled node are individually fed to fully-connected layers without considering any underlying structural information.
- LSTM: DAG structures are flattened into a linearized stream and fed to an LSTM before classification.
- GCN: Graphs along with node features are fed to a Graph Convolutional Network followed by classification layers.
- GAT: Graphs along with node features are fed to a Graph Attention Network followed by classification layers.
- HGCN: Graphs along with node features are fed to a Hyperbolic Graph Convolutional Network followed by classification layers.
- HGAT: Graphs along with node features are fed to a Hyperbolic Graph Attention Network followed by classification layers.
- TreeLSTM: Each graph is fed to a TreeLSTM which recursively encodes each node starting with nodes with no incoming edges and ending with nodes with no outgoing edges.
- AttnTreeLSTM: Variant of TreeLSTM where an attentive mechanism is applied during child state aggregation.

### B.2 Datasets and Preprocessing

- MELD(Poria et al., 2018) is an Emotion Recognition dataset from the TV show Friends. There are 7 emotion labels which include neutral, happiness, surprise, sadness, anger, disgust, and fear. The conversation graph is given as $t = (V, E)$ where each labelled node $v_j \in V$ is considered an utterance in the conversation. As described in Shen et al. (2021), the directed edge $e_{kj}$ connects node $v_k$ to node $v_j$ if $v_k$ is a previous utterance by the same speaker, representing

remote information. The directed edge $e_{lj}$ connects node $v_l$ to node $v_j$ if $v_l$ is an utterance by a different speaker after $v_k$ but before $v_j$, representing local information. The node $v_k$ and all nodes $v_l$ satisfying the above constraints make up the predecessor set $\mathbb{C}(j)$ of node $v_j$ (node $j$ for simplicity). We only use textual information for each node and encode node features using RoBERTa. We use an 70-20-10 train-val-test split.

- IEMOCAP(Busso et al., 2008) is an emotion recognition dataset where each conversation in IEMOCAP comes from the performance by actors based on a script. There are 6 types of emotion, namely neutral, happiness, sadness, anger, frustrated, and excited. The conversation graph is constructed in the same manner as described for MELD. Node features are encoded using RoBERTa. Since this dataset has no validation split, we use the last 20 dialogues from the training set for validation (Shen et al., 2021).

- EmoryNLP(Zahiri and Choi, 2018) consists of TV show scripts as conversations collected from Friends, but is different from MELD in terms of scenes and emotion labels. Labels include neutral, sad, mad, scared, powerful, peaceful, and joyful. The conversation graph is constructed in the same manner as described for MELD. Node features are encoded using RoBERTa. We use an 80-10-10 train-val-test split.

- UPFD-Pol and UPFD-Gos consist of retweet propagation trees arising from a news article. As described in Dou et al. (2021), each node feature of the retweets is the averaged BERT encodings of the historical posts of the user who retweeted it. The news is also encoded using BERT. We only use this arising tree structure for our tasks, where $t = (V, E)$ is the graph and the directed edge $e_{kj}$ connects node $v_k$ to node $v_j$ based on the scheme defined by Dou et al. (2021). We use the same 70-20-10 train-val-test split.

- SST-5 (Socher et al., 2013) is a corpus with fully labelled parsed trees. It allows fine-grained sentiment classification of sentences based on their compositional trees (Tai et al., 2015), given as $t = (V, E)$. We use a 70-10-20 train-val-test split. Node features are 100-dimensional Glove embeddings.

- Disease consists of a disease propagation tree simulated by Chami et al. (2019) using the SIR disease spreading model, where the label of a node is whether the node was infected or not. Based on the model, they build a tree network $t = (V, E)$, where node features indicate the susceptibility to the disease. Due to the unavailability of the inductive variant, we only use the transductive variant of this dataset. We use the same 30-10-60 split described by Chami et al. (2019).

- PHEME (Zubiaga et al., 2016) is a rumour classification dataset, where the goal is to classify whether a tweet is a true or false rumour. The tree structure is given as $t = (V, E)$ where the directed edge $e_{kj}$ connects a reply node $v_k$ with the node it $v_j$ which it replies too. The labelled root node hence has no outgoing edges. Node features are RoBERTa embeddings. We use a 70-10-20 train-val-test split.

- Twitter16(Ma et al., 2017) is also a rumour classification task. The propagation tree for this dataset also contains retweet and reply propagation information, but we only use the tree created by replies as described for PHEME. Node features are RoBERTa embeddings. We use a 70-10-20 train-val-test split.

### B.3 Loss Function

To mitigate class imbalance in certain tasks, we apply Class-Balanced loss (Cui et al., 2019) along with Focal Loss (Lin et al., 2017) to train MRIL, which introduces a weighting factor that is inversely proportional to the number of samples per class, yielding the loss $\mathcal{L}$ as:

$$\mathcal{L}(\widehat{\mathbf{y}}_\mathbf{i}, y_i) = \text{CB}_{focal}(\widehat{\mathbf{y}}_\mathbf{i}, y_i; \beta, \gamma) \tag{25}$$

where $\beta$ and $\gamma$ are hyperparameters.