HYDRA: HYperbolic Deception detection using gRaph Attention

Baljyot Singh Modi
2022133

Vijval Ekbote
2022569

Yashovardhan Singhal
2022591

1 Introduction

> "The essence of lying is in deception, not in words" — *John Ruskin*

Much is made of the human tendency to trust, and the consequent exploitation of this trust by those willing to deceive. In the online space in particular, these deceptions are extremely common, and can broadly be referred to as Digital Deception, which is a form of communication comprising a "technologically mediated message" which intentionally presents information in a way that deceives the receiver [6]. This form of deception is often seen in the form of disinformation campaigns [7] and spearphishing [4], among others. The widespread presence of such deceptions thus makes their detection an interesting problem. We tackle this particular problem in the context of the game "Diplomacy", in which 7 players try to gain control over Europe by forging and breaking alliances. The dataset we use [8] contains conversations recorded from players playing the game. Each conversation contains multiple messages and their corresponding labels from the points of view of both the sender and the receiver. A message can be either a truth or a lie, and thus, given a message, predicting its authenticity becomes a binary classification problem.

We employ various approaches to tackle this problem, and present our results. Inspired by the hierarchical nature of textual data and the effectives of hyperbolic spaces in modelling such structures, we investigate the effectiveness of Gated Recurrent Units (GRUs) in this problem. We further experiment with context GRUs, Graph Attention Networks (GATs), and propose an architecture HYDRA (HYperbolic Deception detection using gRaph Attention)

2 Related Works

2.1 Deception Detection in the context of Diplomacy

This work [8] introduced this task using the context of the game, "Diplomacy". A detailed exposition of the game is presented, followed by a detailed breakdown of how the authors used said game to formulate the task of deception detection. The authors define what a "lie" is in this context, and then outlined their strategy to create the dataset. Since such a dataset did not exist before, they had to manually collect the relevant data using a game engine and a chat system, and engaged different players who would annotate messages as and when they appeared in their respective conversations. These annotations form the ground truth for this dataset. Apart from the creation of this novel dataset, the authors also presented the results they obtained using a number of baselines, which included classical methods (Bag of Words, Harbingers), neural methods (Long Short Term Memory Networks (LSTMs) cite, context LSTMs), and an evaluation of human performance on the test set. Finally, the authors present some samples which were examples of true and false positives, and true and false negatives.

2.2 Hyperbolic Neural Networks

This work [5] presented a solid formalization of neural networks operating in hyperbolic space, and derived the fundamental operations often used in neural networks, in the context of hyperbolic spaces. They present proofs and derive results for hyperbolic formulations of matrix vector multiplication, addition, scalar multiplication etc., in hyperbolic space. In addition to this, this work presents generalizations of multinomial logistic regression, recurrent neural networks, and gated recurrent units to hyperbolic space (in particular, the Poincare Ball Space). Finally, they presented their results on 2 different tasks, textual entailment, and noisy prefix

recognition, demonstrating performance comparable and in some case superior to euclidean neural networks.

2.3 Graph Attention Networks

Proposed and implemented an efficient masked self attention mechanism which operates over graph nodes, allowing each node to attend to the representations of other nodes. This enabled learnable weights between nodes, and generalized to any graph without any knowledge of its structure beforehand. The authors show that they obtained results that were comparable to and in some cases surpassed state of the art methods on datasets such as Pubmed, Citeseer, Cora, and a protein-protein interaction dataset.

2.4 Using RAG and LLMs for Deception Detection

[1] proposed RADDICL, a domain-agnostic framework using Retrieval-Augmented Generation (RAG) for few-shot deception detection. Their approach integrates a chat-tuned LLM with a vector store to retrieve contextually relevant, labeled examples from the DIFrauD benchmark. By embedding definitions of deception and contextual prompts, the model performs zero-to-few-shot classification with full explainability. RADDICL outperforms other few-shot and ICL methods, showing that RAG-enhanced reasoning can be effective in high-variance, low-resource deception settings.

3 Background

3.1 Hyperbolic GRUs

3.1.1 Motivation Behind Hyperbolic Neural Networks

A large number of fields operate using data having an inherently hierarchial structure, and the Euclidean space has been shown to not be as good as Hyperbolic Space for embedding such representations [9, 5]. Thus, hyperbolic neural networks were proposed to leverage the powerful representational power of Hyperbolic space in this context. They have been shown to be at least as effective (and in some case, even better) as standard Euclidean neural networks on tasks such as graph classification [3], textual entailment, and noisy-prefix recognition. [5].

3.1.2 Geometric Preliminaries: Riemannian Manifolds

A **manifold** M of dimension n is a topological space that locally resembles R^n. At each point $x \in M$, the **tangent space** T_xM provides a first-order linear approximation of the manifold near x. A **Riemannian metric** $g = (g_x)_{x \in M}$ is a smoothly varying family of inner products $g_x : T_xM \times T_xM \to R$, which endows the manifold with geometric notions such as length, angle, and curvature. The pair (M, g) defines a **Riemannian manifold**.

3.1.3 Geometric Preliminaries: The Poincare Ball Model

The Poincaré ball model is one of the five isometric models of hyperbolic space. The Poincaré ball (D^n, g_D) is defined as the open unit ball $D^n = \{x \in R^n : \|x\| < 1\}$, equipped with the Riemannian metric:

$$g_D(x) = \lambda_x^2 g_E, \quad where \lambda_x = \frac{2}{1 - \|x\|^2},$$

and g_E is the standard Euclidean metric tensor. This metric is conformal to the Euclidean one, preserving angles. The induced distance between two points $x, y \in D^n$ is:

$$d_D(x, y) = \cosh^{-1}\left(1 + 2\frac{\|x - y\|^2}{(1 - \|x\|^2)(1 - \|y\|^2)}\right).$$

Due to the conformal nature of the metric, the angle between two tangent vectors $u, v \in T_xD^n$ is given by the standard Euclidean inner product formula:

$$\cos(\angle(u, v)) = \frac{\langle u, v \rangle}{\|u\|\|v\|}.$$

3.1.4 Commonly Used Operations in Hyperbolic Neural Networks

We work in the Poincaré ball model $D_c^n = \{x \in R^n : \|x\| < 1/\sqrt{c}\}$, with constant negative curvature $-c < 0$. Below are core operations used in hyperbolic neural networks:

Möbius Addition. For $x, y \in D_c^n$, the Möbius addition is defined as:

$$x \oplus_c y = \frac{(1 + 2c\langle x, y \rangle + c\|y\|^2)x + (1 - c\|x\|^2)y}{1 + 2c\langle x, y \rangle + c^2\|x\|^2\|y\|^2}$$

This operation generalizes Euclidean addition and satisfies gyrocommutativity and associativity. As $c \to 0$, it recovers standard vector addition.

Möbius Scalar Multiplication. For $r \in R$, $x \in D_c^n$, scalar multiplication is given by:

$$r \otimes_c x = \frac{1}{\sqrt{c}} \tanh\left(r \tanh^{-1}\left(\sqrt{c}\|x\|\right)\right) \frac{x}{\|x\|}, \quad r \otimes_c 0 = 0$$

It reduces to the Euclidean scalar multiplication in the limit $c \to 0$.

Exponential and Logarithmic Maps. Given a point $x \in D_c^n$, the exponential map $\exp_x^c : T_x D_c^n \to D_c^n$ and logarithmic map $\log_x^c : D_c^n \to T_x D_c^n$ are defined as:

$$\exp_x^c(v) = x \oplus_c \left(\tanh\left(\frac{\sqrt{c}\lambda_x^c \|v\|}{2}\right) \frac{v}{\sqrt{c}\|v\|}\right),$$

$$\log_x^c(y) = \frac{2}{\sqrt{c}\lambda_x^c} \tanh^{-1}\left(\sqrt{c}\|-x \oplus_c y\|\right) \frac{-x \oplus_c y}{\|-x \oplus_c y\|}.$$

where $\lambda_x^c = \frac{2}{1-c\|x\|^2}$.

Möbius Function Application. For a function $f : R^n \to R^m$, its Möbius counterpart $f_c : D_c^n \to D_c^m$ is defined via:

$$f_c(x) = \exp_0^c\left(f\left(\log_0^c(x)\right)\right)$$

This allows applying standard Euclidean layers within hyperbolic space using tangent space projections.

Möbius Matrix-Vector Multiplication. Given a linear map $M : R^n \to R^m$, and $x \in D_c^n$ with $Mx \neq 0$, the Möbius matrix-vector product is:

$$M^{\otimes^c}(x) = \frac{1}{\sqrt{c}} \tanh\left(\frac{\|Mx\|}{\|x\|} \tanh^{-1}(\sqrt{c}\|x\|)\right) \frac{Mx}{\|Mx\|}$$

This operation extends linear transformations to hyperbolic space while preserving the manifold structure.

3.1.5 Formulation of Hyperbolic GRU

Hyperbolic GRU Cell. Given input $x \in D_c^n$ and hidden state $h \in D_c^n$, the Hyperbolic GRU cell computes the next state $h' \in D_c^n$ as follows:

$$\begin{aligned}
&\mathrm{h}_x = [h, x] \\
&z = \sigma\left(\log_0^c\left(proj_x\left(W_z^{\otimes^c}(h_x)\right)\right)\right) \\
&r = \sigma\left(\log_0^c\left(proj_x\left(W_r^{\otimes^c}(h_x)\right)\right)\right) \\
&r \otimes h = r \odot h \\
&\tilde{h}_x = proj_x\left(\exp_0^c\left([\log_0^c(r \otimes h), \log_0^c(x)]\right)\right) \\
&\tilde{h} = \exp_0^c\left(\tanh\left(\log_0^c\left(proj_x\left(W^{\otimes^c}(\tilde{h}_x)\right)\right)\right)\right) \\
&h' = proj_x\left((1-z) \otimes h \oplus_c z \otimes \tilde{h}\right)
\end{aligned}$$

4 Dataset

The Diplomacy dataset contains pairwise conversations annotated by the sender and the receiver for deception (and conversely truthfulness). The 17,289 messages are gathered from 12 games.

4.1 Structure

Each file in the Diplomacy dataset is a JSON Lines file, where each line represents an entire game dialog as a JSON object. The fields of each object are described below.

`speaker` The sender of the message.

`recipient` The receiver of the message.

`messages` The raw message string.

`sender_labels` A boolean value (true or false) indicating the sender's label.

`receiver_labels` The receiver's label, which can be true, false, or NOANNOTATION.

`score_delta` The current game score of the sender
minus the game score of the recipient.

`absolute_message_index` The index of the message in the entire game
across all dialogs.

`relative_message_index` The index of the message within the current
dialog.

`game_id` An identifier indicating which of the 12 games
the dialog belongs to.

4.2 Class Imbalance

The Diplomacy dataset exhibits a significant class imbalance, particularly in the labeling of messages. The `sender_labels` and `receiver_labels` fields, which indicate deceptive or notable behaviors, are skewed, with the majority class (e.g., non-deceptive or neutral messages) comprising approximately 95% of the data, while the minority class (e.g., deceptive messages) accounts for only about 5%. This imbalance is inferred from the class weights used in training models on this dataset, such as `[1.0 / 0.1, 1.0 / 0.90]`, which suggest a 20:1 ratio favoring the negative class. Such a distribution is common in deception detection tasks, where

deceptive instances are rare compared to honest or neutral communication.

This imbalance poses challenges for machine learning models, as they may become biased toward predicting the majority class, leading to poor performance on the minority class (deceptive messages). To mitigate this, techniques such as class weighting or oversampling the minority class can be employed

4.3 Power Dynamics

Power dynamics play a crucial role in the Diplomacy dataset, reflecting the strategic nature of the game where players negotiate alliances and betrayals to gain territorial control. The `score_delta` field, which measures the difference between the sender's and recipient's game scores, serves as a proxy for relative power within a dialog. A positive `score_delta` indicates the sender holds a stronger position, potentially influencing their messaging strategy—e.g., issuing demands or leveraging dominance. Conversely, a negative `score_delta` suggests the recipient has the upper hand, which may lead to more conciliatory or deceptive messages from the sender to regain favor.

These dynamics may influence the likeliness of lying; a stronger player may feel empowered to lie to their neighbor on the other hand the weaker player might be gullible to deception.

5 Methodology

5.1 HYDRA Architecture

In this section, we present the architecture for our model, HYDRA (HYperbolic Deception detection using gRaph Attention)

5.1.1 Data Preparation

We load our dataset for passing to model in a way such that each sample in a batch is an entire conversation. Each sample thus containes messages from that conversation. We do this to enable us to use the context of previous messages later on.

5.1.2 Extracting Embeddings from Hyperbolic GRU

We pass each message of each conversation to a Hyperbolic GRU (HyGRU). We use the final hidden state obtained from this HyGRU to get a representation of the message, to be used further.

5.1.3 Construction of Message Graph

GNNs (Graph Neural Networks) are widely used due to their ability to model complex relationships from the training data by taking into account relationships between entities (nodes). We believe that each conversation can be modeled as a graph, with an encoding for each message acting as a node, and our task ultimately being a node classification task, with the task at each node being binary classification between truth and lie. To pass our data to such a model, we need to construct a graph, the creation of which we detail below. Based on the embeddings obtained from the HyGRU, we construct a graph which we use as input for 2 graph attention networks (GATs). This graph is constructed using a similarity measure, namely, dot product between normalized embeddings obtained from the HyGRU. For each message, we select a certain number of messages based on whether their respective similarities with the current message cross a certain threshold (0.7 in our case). This enables us to create relatively sparse graphs which carry important information, as compared to creating a graph in which each message has edges connecting it to every previous message. We thus improve efficiency and also include similarity to give us a richer representation of the conversations. One thing to note is that a message is only connected to similar (based on threshold) messages from the same conversation, since players did not have access to messages from other conversations. This ensures we model the environment of the game as closely as possible. Also, for each message embedding, we perform the logarithmic map before passing it to the GAT.

5.1.4 Learning Node Embeddings using GATs

Once we obtain the graph as detailed above, we pass the node embeddings through 2 Euclidean GAT layers. These layers will help each node (message embedding) attend to the relevant nodes based on the final task (binary classification).

5.1.5 Final Binary Classification

Once we pass the nodes through the GAT layers, the resultant embedding for each node is extracted, and concatenated with the respective embedding obtained for the message (corresponding to that node) from the HyGRU, along with the respective power difference between the players (who conducted this particular conversation). We then simply pass this representation obtained for each message through a fully connected layer, with 2

output nodes.

5.2 Retrieval-Augmented Generation (RAG):

While Graph Neural Networks (GNNs) and Hyperbolic LSTMs effectively model relationships and long-term deception patterns, they lack explicit use of historical deception knowledge from past conversations. Inspired by [2], Retrieval-Augmented Generation (RAG) enhances the model by retrieving relevant deceptive and truthful messages from a knowledge base of past Diplomacy conversations to aid classification. Using a vector database like ChromaDB, RAG indexes previous interactions and retrieves similar past deceptive patterns to provide additional context.

The RAG process involves generating embeddings for historical messages and queries using the 'all-MiniLM-L6-v2' SentenceTransformers model, storing them in ChromaDB for efficient similarity-based retrieval. During inference or training, the model embeds the input message and retrieves the top-k (e.g., k=5) most similar messages. Unlike typical RAG systems that use a Large Language Model (LLM) like GPT-4 or BERT for generation, this approach adapts RAG for classification, feeding retrieved and original message embeddings into a custom 'ContextLSTMWithRAG' model. This model uses bidirectional LSTMs to encode message, conversation, and retrieved context features, with BERT embeddings as input, culminating in a binary deception classification.

This approach improves generalization by exposing the model to diverse deception strategies across games and enhances explainability by linking decisions to retrieved examples. A persistent ChromaDB ensures the knowledge base remains available, and batched processing (max 5,000 documents) optimizes performance. Future work could explore different embedding models or adjust the 'top-k' value to further improve results.

6 Results and Observations

We obtained results as given in Tables 1 and 2, using various configurations. We also obtained a confusion matrix using our best model, as shown in figure 3

- **False Positive (FP):**
  sure a strong england is always a threat to me so id be happy to help annoy them in the north seascandinavia
  True Label: 0, **Predicted Label:** 1

- **True Positive (TP):**
  hi italy just opening up communication and i want to know what some of your initial thoughts on the game are and ifhow we can work together
  True Label: 1, **Predicted Label:** 1

- **False Negative (FN):**
  ok france has told me outright theyre wanting to make sure that italy is divided fairly so i doubt theyll expect me pushing past you toward the west
  True Label: 1, **Predicted Label:** 0

- **True Negative (TN):**
  by the way any suggestions for me in the north ive been hearing different things from germany and england so im at a bit of a crossroads in terms of who to trust and what to do
  True Label: 0, **Predicted Label:** 0

Analysis

From the results, we observe that modeling message embeddings in hyperbolic space leads to significantly better performance compared to the ContextLSTM baselines. This aligns with the intuition that textual data, especially conversational structures, are inherently hierarchical — a structure that hyperbolic space represents more naturally and effectively.

By incorporating a Graph Attention Network (GAT), the model is further able to leverage relational information from previous messages exchanged between players, which enhances its ability to detect deceptive behavior.

Furthermore, introducing both past and future conversational context yields the best performance, suggesting the value of broader discourse context. However, this setup serves only as an experimental upper bound, as future context would not be available during real-time or human evaluation.

We observe similar trends on the receiver side as well — models operating in hyperbolic space continue to outperform the ContextLSTM baselines. Interestingly, while the RAG + BERT + ContextLSTM model achieves the 2nd highest overall accuracy in the receiver setting, its Lie F1 score is lower than that of our proposed models. This indicates that its macro F1 is dominated by performance on the truthful class, which is not desirable in the context of deception detection, where identifying lies is the primary goal.

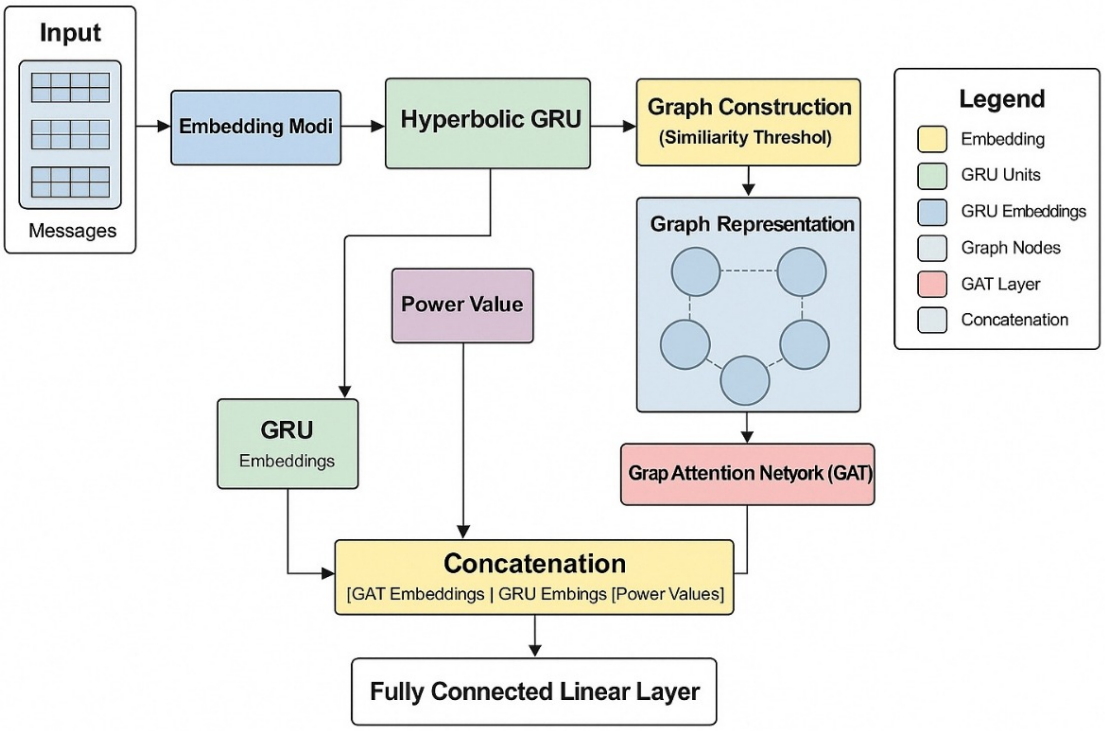

Figure 1: HYDRA Architecture

Model	Sender/Receiver	Macro F1	Lie F1	Accuracy
Baseline1(ContextLSTM_GLoVE)	Sender	0.515	0.151	0.787
Baseline2(ContextLSTM_GLoVE_Power)	Sender	0.527	0.171	0.794
Baseline3(ContextLSTM + Power* [8])	Sender	0.572	0.27	Not reported
HyGRU + BERT	Sender	0.564	0.198	0.873
HYDRA	Sender	**0.578**	**0.222**	**0.879**
HyGRU + BERT + GAT + Future context*	Sender	0.584	0.235	0.876
RAG + BERT + ContextLSTM	Sender	0.5253	0.1583	0.8092

Table 1: Model performance for Actual Lie Prediction across different configurations. The * indicates that using future context was just an experiment. Actual human evaluation used only past context

7 Conclusion and Future Work

Conclusion

In this study, we demonstrated that hyperbolic space-based models, particularly those enhanced with Graph Attention Networks (GAT), outperform traditional ContextLSTM models in detecting deception in conversational data. By incorporating relational context and experimenting with future and previous conversation data, we observed substantial improvements in model performance. Overall, our approach provides a promising direction for advancing deception detection in dialog systems.

Future Work

In future work, we aim to explore the use of Graph Attention Networks (GAT) in hyperbolic space. Since our experiments with GRU showed improved results when message embeddings were computed in hyperbolic space, we believe that extending this approach to GAT could yield further performance gains. Leveraging the geometric properties of hyperbolic space may allow GAT to better capture hierarchical or tree-like structures in message graphs, which are often present in real-world communication data.

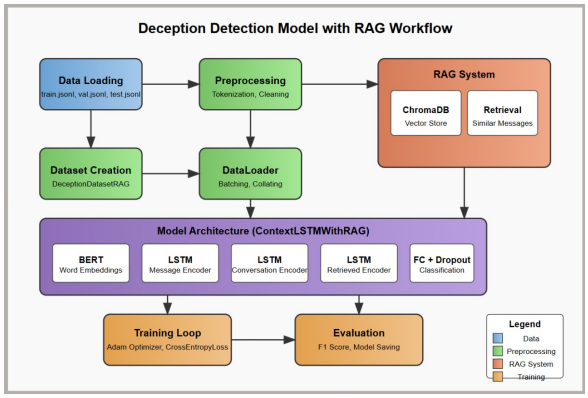

Figure 2: Our RAG-based model

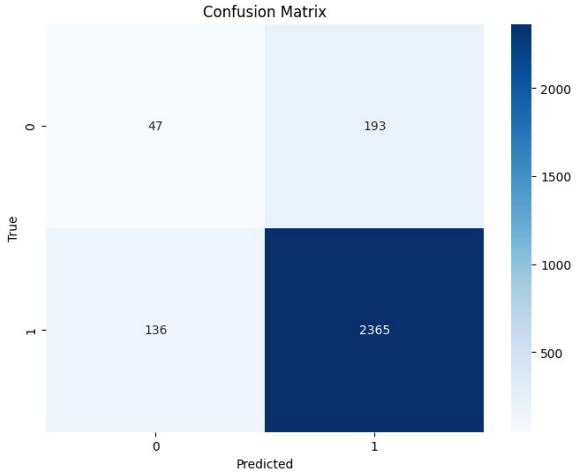

Figure 3: Confusion Matrix for our best model

Model	Sender/Receiver	Macro F1	Lie F1	Accuracy
Baseline1(ContextLSTM_GLoVE)	Receiver	0.501	0.061	0.889
Baseline2(ContextLSTM_GLoVE_Power)	Receiver	0.511	0.126	0.815
Baseline3(ContextLSTM + Power* [8])	Receiver	0.533	0.13	Not reported
HyGRU + BERT	Receiver	0.515	0.0833	**0.902**
HYDRA	Receiver	0.532	**0.16**	0.830
HyGRU + BERT + GAT + Future context	Receiver	0.542	0.169	0.845
RAG + BERT + ContextLSTM	Receiver	**0.540**	0.131	**0.904**

Table 2: Model performance for Suspected Lie Prediction across different configurations. The * indicates that using future context was just an experiment. Actual human evaluation used only past context

References

[1] Dainis Boumber, Bryan E Tuck, Rakesh M Verma, and Fatima Zahra Qachfar. 2024. Llms for explainable few-shot deception detection. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics*, pages 37–47.

[2] Dainis A. Boumber, Fatima Zahra Qachfar, and Rakesh Verma. 2024. Domain-agnostic adapter architecture for deception detection: Extensive evaluations with the DIFrauD benchmark. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5260–5274, Torino, Italia. ELRA and ICCL.

[3] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32.

[4] Rachna Dhamija, J Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 581–590.

[5] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. *Advances in neural information processing systems*, 31.

[6] Jeffrey T Hancock. 2007. Digital deception. *Oxford handbook of internet psychology*, 61:289–301.

[7] Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559*.

[8] Denis Peskov and Benny Cheng. 2020. It takes two to lie: One to lie, and one to listen. In *Proceedings of ACL*.

[9] Frederic Sala, Chris De Sa, Albert Gu, and Christopher Ré. 2018. Representation tradeoffs for hyperbolic embeddings. In *International conference on machine learning*, pages 4460–4469. PMLR.