Efficient Environmental Claim Detection with Hyperbolic Graph Neural Networks

Darpan Aswal[1,2], **Manjira Sinha**[3]
[1]Department of Computer Science, Université Paris-Saclay
[2]MICS, CentraleSupélec, Université Paris-Saclay
[3]TCS Research, India
Correspondence: darpanaswal@gmail.com

Abstract

Transformer based models, especially large language models (LLMs) dominate the field of NLP with their mass adoption in tasks such as text generation, summarization and fake news detection. These models offer ease of deployment and reliability for most applications, however, they require significant amounts of computational power for training as well as inference. This poses challenges in their adoption in resource-constrained applications, especially in the open-source community where compute availability is usually scarce. This work proposes a graph-based approach for Environmental Claim Detection, exploring Graph Neural Networks (GNNs) and Hyperbolic Graph Neural Networks (HGNNs) as lightweight yet effective alternatives to transformer-based models. Re-framing the task as a graph classification problem, we transform claim sentences into dependency parsing graphs, utilizing a combination of word2vec & learnable part-of-speech (POS) tag embeddings for the node features and encoding syntactic dependencies in the edge relations. Our results show that our graph-based models, particularly HGNNs in the poincaré space (P-HGNNs), achieve performance superior to the state-of-the-art on environmental claim detection while using up to **30x fewer parameters**. We also demonstrate that HGNNs benefit vastly from explicitly modeling data in hierarchical (tree-like) structures, enabling them to significantly improve over their euclidean counterparts. We make our implementation publicly available [1].

1 Introduction

Claim verification and claim detection (Soleimani et al., 2020; Levy et al., 2014) are complex NLP tasks that involves the detection of fake claims using facts as well as contextual information within the given claims. Often, these claims exhibit hierarchical and nested information such as conditional statements (Kargupta et al., 2025). Environmental claim detection (Stammbach et al., 2022) involves additional elements from greenwashing (de Freitas Netto et al., 2020) that are often used by corporations to promote products and mislead customers.

Recent work for claim detection, similar to many industrial NLP applications (Chkirbene et al., 2024), has predominantly relied on transformer-based architectures (Ni et al., 2024). However, this reliance on these massive, black-box models presents two issues. First, they require large-scale computational resources which makes them economically and environmentally expensive, leaving behind a large carbon footprint (Faiz et al., 2023). Second, their lack of interpretability (Lin et al., 2023) is a significant issue in high stakes domains like claim verification, where explaining a classification is equally important as the classification itself (Atanasova, 2024; Brundage et al., 2020). The increasing scrutiny on sustainability claims further necessitates interpretability and computational efficiency in models.

To address these challenges of cost and interpretability, we propose a lightweight framework for graph-based claim detection. We re-frame the problem of environmental claim detection as a graph classification task, explicitly modeling the syntactic and hierarchical structure of sentences using dependency parsing graphs (Nivre, 2010) with word embeddings for node features. This representation provides a natural fit for Graph Neural Networks (GNNs) (Wu et al., 2020) which are designed to learn from such structured data. Compared to transformers, our approach offers an interpretable approach to syntactic and semantic learning while significantly reducing computational overhead (Feng et al., 2025; Li et al., 2025; Peng et al., 2021). Furthermore, given the tree-like nature of dependency graphs, we investigate Hyperbolic Graph Neural Networks (HGNNs) (Zhou et al., 2023), a geometric learning architecture particularly suited to such

[1]https://github.com/darpanaswal/ecd-hgnn

Proceedings of the IJCNLP-AACL 2025 Student Research Workshop, pages 24–35

hierarchically structured data. The research questions for the study are as follows.

RQ1. Can graph-based models match SOTA performance for environmental claim detection while using just a fraction of the compute as that of LLMs?

RQ2. Can syntactically enriched explicit hierarchical modeling of NLP tasks advantage hyperbolic models over their euclidean counterparts?

2 Related Work

The proliferation of misinformation on social media has shown the need for automated fact-checking and verification systems (Aïmeur et al., 2023). Fake news detection aims to classify entire articles or posts as credible or fake (Shu et al., 2017), often involving analyzing of multiple signals such as textual content and writing style (Przybyla, 2020). While early approaches relied on feature engineering and machine learning methods (Khanam et al., 2021), recent work relies on transformer models for fake news detection (Yi et al., 2025).

Claim detection and verification offer a more detailed approach to fact-checking. Claim detection (Levy et al., 2014) focuses on identifying factual statements within larger texts and separating them from non-factual ones. Claim verification (Soleimani et al., 2020) on the other hand assesses the accuracy of detected claims using evidence and facts from trusted sources. While fact checking is widely utilized for social-media content (Wasike, 2023), these methods have been applied to specific, high-stakes topics such as verification of climate-related claims (Diggelmann et al., 2020) and analyzing contrarian (Coan et al., 2021) or fake claims about climate change (Al-Rawi et al., 2021). Environmental claim detection (Stammbach et al., 2022) is one such specialized sub-domain of fact verification research. Specifically, it deals with greenwashing (de Freitas Netto et al., 2020) – the corporate form of misinformation – which involves using vague or misleading language to create an exaggeratedly positive public image of a company's environmental credentials.

Large Language Models (LLMs) (Naveed et al., 2025) are transformer-based models (Lin et al., 2022) pre-trained on vast amounts of text data which enables them to achieve state-of-the-art performance in downstream tasks such as sentiment analysis, machine translation and named entity recognition (Miah et al., 2024; Zhang et al., 2023; Yan et al., 2019). The application of these models has evolved from fine-tuning (Wu et al., 2025) task specific models such as BERT and RoBERTa (Soleimani et al., 2020; Stammbach et al., 2022), to in-context learning (Dong et al., 2022) with modern, multi-billion parameter models. While powerful, the high computational costs (Faiz et al., 2023) and lack of interpretability (Lin et al., 2023) of these models pose challenges for wide-scale adoption.

Graph Neural Networks (GNNs) (Wu et al., 2020) offer an alternative learning paradigm by operating on structured-data. Prior work has utilized GNNs to explicitly model hierarchical and relational dependencies (Mi and Chen, 2020), making graph structures such as constituency parsing (Li et al., 2020b) and dependency parsing graphs (Nivre, 2010) a natural fit for representing sentence structures in NLP tasks. These models can integrate rich semantic information from word embeddings, knowledge graphs, and even sentence embeddings from pre-trained language models (Mikolov et al., 2013; Opdahl et al., 2022; Li et al., 2020a). Geometric deep learning (Bronstein et al., 2017) generalizes these models to non-euclidean spaces (Coxeter, 1998). Extending GNNs, Hyperbolic GNNs (Zhou et al., 2023), are particularly well suited to model hierarchically structured data such as dependency parsing graphs.

3 Methodology

We begin our experimentation by transforming a dataset $D = \{c_1, c_2, \ldots, c_N\}$ of N environmental claims into a corresponding set of dependency parsing graphs $G = \{G_1, G_2, \ldots, G_N\}$, converting each claim c_i into a unique graph structure

$$G_i = (V_i, E_i)$$

where V_i is the graph's set of vertices (or nodes) and E_i is its set of edges.

3.1 Dependency Graph Construction

For each claim c_i in the dataset, we generate a directed dependency graph using spaCy's built-in DependencyParser. Claim c_i, which is a sequence of tokens $t_i = \{t_i^{(1)}, t_i^{(2)}, \ldots t_i^{(k)}\}$ is mapped to its corresponding graph $G_i = (V_i, E_i)$ where the vertex set $V_i = \{v_1, \ldots v_k\}$ represents the tokens, and the edge set E_i represents the syntactic dependencies between them. A directed edge $(v_h, v_j) \in E_i$ exists if the token t_h is the syntactic head of token t_j. Each edge is labeled with its dependency type

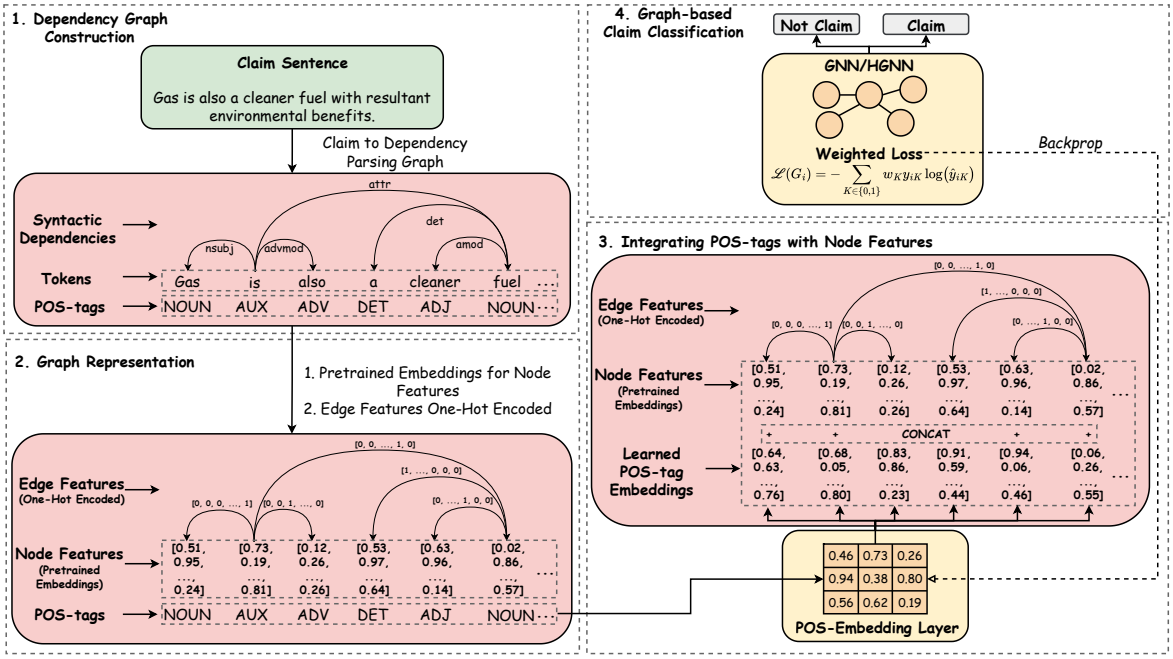

Figure 1: Overview of the Graph-based Claim Detection Pipeline. Step 1: Claim sentence to dependency graph conversion. Steps 2: Dependencies are one-hot encoded as edge features. Node features are initialized with pretrained embeddings. Step3: Node features are concatenated with POS-tag embeddings learned by embedding layer. Step 4: Graph classification using a GNN/HGNN architecture trained with a weighted loss function.

$d \in D$, where D is the set of all 45 unique dependency relations present in the dataset. We utilize the following node and edge attributes from the dependency graphs [2].

- **Token text**: Represented as the graph's nodes; corresponds to tokens in the claim sentences.
- **Dependency relation**: Specifies the type of syntactic dependency between a token and its head. Describes how the token relates to its syntactic governor.
- **Token head**: Also represented as the graph nodes, it identifies the governor token for a given dependent token.
- **Token Part-Of-Speech (POS) tag**.

3.2 Graph Representation

To prepare the graphs for the GNN models, we define the node and edge feature representations.

3.2.1 Node Features

Each node $v \in V_i$ is associated with a feature vector $x_v \in \mathbb{R}^{d_{node}}$. For this vector, we utilize word2vec (Mikolov et al., 2013), a pre-trained word embedding model. $x_v = W_e(\text{token}(v))$, where W_e is the word2vec embedding lookup matrix and the token(v) is the word corresponding to node v.

3.2.2 Edge Features

The syntactic dependency type of each edge, corresponding to one of the 45 unique relations in the dataset, is encoded into a feature vector. For an edge $e = (v_h, v_j)$, its feature vector $e_{hj} \in \mathbb{R}^{|D|}$ is a one-hot encoding of its dependency type $d(e)$.

3.3 Integrating POS-tags with Node Features

Next, we augment the node features with the POS-tags. Let $\mathcal{P}$ be the set of all unique POS-tags in the dataset. We introduce a learnable embedding matrix $W_p \in \mathbb{R}^{|\mathcal{P}| \times d_{pos}}$, where d_{pos} is the dimension of the POS-tag embeddings. This layer is trained with the GNN model. The final feature vector for a node v, denoted x'_v, is the concatenation of its word embedding and its learned POS tag embedding:

$$x'_v = \left[W_e(\text{token}(v)) \,\|\, W_p(\text{pos}(v))\right]$$

The dimension of this augmented feature vector becomes $d'_{node} = d_{node} + d_{pos}$.

3.4 Weighted Loss for Imbalanced Data

Lastly, to address the inherent imbalance present in the dataset, we employ a weighted cross-entropy loss function. This strategy assigns a higher penalty to misclassifications of the minority class, thereby

encouraging the model to pay more attention to it. The loss for a single graph G_i with true one-hot label y_i and predicted probabilities $\hat{y}_i$ is defined as:

$$\mathcal{L}(G_i) = -\sum_{k=0}^{1} w_k.y_{ik}log(\hat{y}_{ik})$$

The weight for each class k, w_k, is calculated as the inverse of its frequency in the training set, effectively balancing the contribution of each class to the overall loss.

3.5 Graph-based Claim Classification

The final stage of our pipeline involves classifying the entire graph representation of a claim sentence. The augmented node feature vectors and the edge feature vectors are fed into either a GNN or an HGNN model which provides the final classification for the claim sentences, classifying them into two possible categories – 'Claim' and 'Not Claim'.

4 Experimental Setup

4.1 Dataset

We utilize the Environmental Claim Detection (ECD) dataset (Stammbach et al., 2022), a dataset comprised of environmental claims extracted from various corporate communications of publicly listed companies, including sustainability reports, earnings calls, and annual reports. While the authors initially collected 3,000 sentences, they removed samples with tied annotations, reporting results on the filtered dataset of 2,647 samples. We use this same 2,647-sample dataset for all our experiments to ensure a direct comparison. The dataset is imbalanced, with 665 sentences (25.1%) labeled as claim statements and 1,982 sentences (74.9%) labeled as not claim statements.

4.2 Models

We conduct our analysis with Euclidean and Hyperbolic GNN architectures. For training our models, we utilized the HGNN toolkit from (Liu et al., 2019). We experiment with the two standard models of hyperbolic space – the Poincaré Ball (Nickel and Kiela, 2017), which represents the hyperbolic space inside a unit disk and the Lorentz Hyperboloid Model (Nickel and Kiela, 2018) which embeds the space on a hyperboloid (Reynolds, 1993) in a higher-dimensional Minkowski space (Naber, 2012). Our models are trained with a total of 4 GNN layers. The first layer's dimensionality $d_{in} = d_{word2vec} + d_{pos}$, where $d_{word2vec} = 300$. The other 3 GNN layers are of dimensionality 256. For training, we utilize the AMSGrad and the Riemannian AMSGrad optimizers for the GNN and HGNN respectively [2].

4.3 Evaluation Metrics

For evaluating our models, we use five primary metrics to assess their performance on the claim detection task – Accuracy, Precision, Recall, F1-score, and AUC-ROC [2]. Given the high imbalance in the dataset, we use the F1-score and AUC-ROC as our primary metrics

5 Results & Observations

To obtain the best performance for each model configuration, we grid-search over the dropout rate, POS-embedding dimension and class weights. Next, we describe our results in detail in relation to the research questions described earlier.

5.1 HGNNs Match SOTA Performance with upto 30x Fewer Parameters (RQ1.)

In Table 1, we first establish the baselines using the results from (Stammbach et al., 2022) with 4 transformer models – DistilBERT, ClimateBERT, RoBERTa$_{\text{base}}$, and RoBERTa$_{\text{large}}$. While we use F1-score and AUC-ROC as our primary metrics, we include the standard accuracy in our tables solely for a direct comparison with the baseline metrics. In Table 2, we see that our graph-based models achieve performance better than or comparable to these state-of-the-art transformers. Firstly, our simplest models – labeled GNN, L-HGNN (for HGNN in the lorentz space) and P-HGNN (for HGNN in the poincaré space) – achieve respectable test F1 and accuracy scores.

Augmenting the models with the POS-tag embeddings uniformly boosts performance across all architectures. Specifically, we observe increments in the test F1 and accuracy scores for all three models. Notably, P-HGNN-POS achieves both our best overall test F1 and accuracy scores of **84%** and **92.1%** respectively, beating the best test accuracy reported in Table 1 (91.7%) and coming very close to the best test F1 score (84.9%), both achieved by their largest model RoBERTa$_{\text{large}}$ consisting of **355 million parameters**. Both GNN-POS and L-HGNN-POS also show competitive test F1 scores of **78.5%** and **79.4%** respectively, while achieving near SOTA accuracy scores of **89.1%** and **89.4%**.

Model	dev				test			
	pr	rc	F1	acc	pr	rc	F1	acc
DistilBERT	**77.5**	**93.9**	**84.9**	91.7	74.4	**95.5**	83.7	90.6
ClimateBERT	76.9	90.9	83.3	90.9	76.5	92.5	83.8	90.9
$\text{RoBERTa}_{\text{base}}$	74.7	**93.9**	83.6	90.6	73.3	94.0	82.4	89.8
$\text{RoBERTa}_{\text{large}}$	**80.5**	**93.9**	**86.7**	**92.8**	**78.5**	92.5	**84.9**	**91.7**

Table 1: Results reported by (Stammbach et al., 2022) on their ECD-dataset.

Model	grid-search parameters			dev					test				
	Dropout Rate	POS Embedding Dimension	Class Weights	pr	rc	F1	acc	auc	pr	rc	F1	acc	auc
GNN	0.1	–	–	**79.3**	69.7	74.2	87.9	**0.93**	78.7	71.6	75.0	87.9	0.93
L-HGNN	0.1	–	–	70.3	78.8	74.3	86.4	0.92	73.7	83.6	78.3	88.3	0.93
P-HGNN	0	–	–	71.0	74.2	72.6	86.0	0.91	74.4	**86.6**	80.0	89.1	0.94
GNN-POS	0.3	32	–	75.4	74.2	74.8	87.5	0.92	77.9	79.1	78.5	89.1	0.94
L-HGNN-POS	0.1	64	–	70.5	65.2	67.7	84.5	0.92	78.3	80.6	79.4	89.4	0.93
P-HGNN-POS	0.3	128	–	75.4	74.2	74.8	87.5	**0.93**	**85.9**	82.1	**84.0**	**92.1**	**0.95**
Balanced-GNN	0.1	–	[1,1.5]	78.6	66.7	72.1	87.2	**0.93**	81.7	73.1	77.2	89.1	0.93
Balanced-L-HGNN	0.25	–	[0.8,1.6]	77.8	74.2	76.0	**88.3**	**0.93**	75.7	79.1	77.4	88.3	0.93
Balanced-P-HGNN	0.2	–	[1,1.5]	73.2	78.8	75.9	87.5	0.92	73.7	83.6	78.3	88.3	0.93
Balanced-GNN-POS	0.25	32	[0.6678,1.9897]	72.9	77.3	75.0	87.2	**0.93**	76.7	83.6	80.0	89.4	0.93
Balanced-L-HGNN-POS	0	16	[0.6678,1.9897]	73.5	75.8	74.6	87.2	**0.93**	74.0	85.1	79.2	88.7	0.94
Balanced-P-HGNN-POS	0.3	32	[0.8,1.6]	73.6	**80.3**	**76.8**	87.9	**0.93**	80.3	85.1	82.6	90.9	0.94

Table 2: We report precision, recall, F1 score, accuracy and the auc-roc score on the dev and test sets of the ECD dataset. The best performance per split is indicated in bold, the second best is underlined.

Next, we address the imbalance in the dataset through the introduction of a weighted loss function. The Balanced-GNN improves the test F1-score by over 2 points compared to the standard GNN (from **75.0%** to **77.2%**), demonstrating the effectiveness of the weighted loss for the Euclidean model. The impact on the hyperbolic models is more nuanced, with slight shifts in the precision-recall trade-off resulting in minor changes to the F1-score. In Table 3, we show the best GNN configurations taken from Table 2 along with all their corresponding weight-balanced versions trained with the same dropout rates. While the early stopping criterion favors the best test F1-score during training, we can still observe the generally expected trend of dropping precision and increasing recall for both the dev and test sets when applying class weights. Interestingly, the Balanced-L-HGNN model does not always follow this pattern as strictly as its euclidean or poincaré counterparts.

Finally, the models incorporating all enhancements – POS embeddings and class balancing (-POS-Balanced) – demonstrate the most overall robust performances, effectively addressing both feature representation and data imbalance. The Balanced-GNN-POS model achieves a strong test F1 and accuracy scores of **80.0%** and **89.4%**, a clear improvement over its unbalanced version with test F1 and accuracy scores of **78.5%** and **89.1%**. Most significantly, while the P-HGNN-POS model achieves our highest test F1-score of **84.0%**, the Balanced-P-HGNN-POS model achieves a competitive F1-score **82.6%** while substantially boosting test recall from **82.1%** to **85.1%** which is our best test recall score after P-HGNN (**86.6%**) and also achieving the best test accuracy of **90.9%** after P-HGNN-POS (**92.1%**).

Furthermore, it is worth noting the consistently high AUC-ROC scores across all our model configurations, as detailed in Table 2. The test set AUC-ROC values range from 0.93 to 0.95, indicating a strong ability of the models to distinguish between the 'Claim' and 'Not Claim' classes. This high level of class separability further reinforces the reliability of our graph-based approach for the task of environmental claim detection.

The key advantage of our approach lies in its computational efficiency. In Table 4, we detail the parameter counts for all (Stammbach et al., 2022) transformer models as well as our own GNN models. While $\text{RoBERTa}_{\text{large}}$, the best performing model for environmental claim detection

from (Stammbach et al., 2022) consists of **355 million parameters**, our GNN and HGNN models are significantly more lightweight. Our models consist of 4 GNN layers, a 256-dimensional hidden state, and 45 unique dependency relations (edge types). We calculate the size of our graph models to be approximately **12M parameters**, nearly **30 times smaller** than $\text{RoBERTa}_{\text{large}}$ [2]. *Therefore, we conclude that our graph-based models achieve better than or comparable to SOTA results.*

5.2 HGNNs Consistently Outperform GNNs (RQ2.)

In Table 2, we observe that hyperbolic GNN models, particularly those in the poincaré space consistently outperform their euclidean counterparts under most configurations for both the F1 and accuracy scores. For example on the test set, both the L-HGNN with an F1-score of **78.3%** and accuracy of **88.3%** as well as the P-HGNN with an F1-score of **80.0%** and an accuracy of **89.1%** surpass the standard GNN with an F1 score of **75.0%** and accuracy of **87.9%**. Similarly, this trend is continued in other configurations and the performance gap widens with the inclusion of richer features, as seen with P-HGNN-POS (**84%** F1 and **92.1%** accuracy) outperforming GNN-POS (**78.5%** F1 and **89.1%** accuracy). This consistent advantage shows that hyperbolic space models significantly benefit from explicit hierarchical modeling of the data using tree-like structures such as dependency parsing graphs. We achieve better test scores with HGNNs than with GNNs under most configurations, indicating a low hyperbolicity (i.e., a strong hierarchical structure) in the ECD dataset. *Therefore, we conclude that explicit hierarchical modeling of environmental claims allows the geometric properties of hyperbolic models to benefit from this hierarchy and improve over their euclidean counterparts.*

6 Discussion

In this study, we investigate the efficacy of Graph Neural Networks (GNNs) and their hyperbolic counterparts (HGNNs) for Environmental Claim Detection. We construct dependency parsing graphs of claim sentences to explicitly model them as hierarchical structures, hence benefiting from the geometric properties of the hyperbolic space. Leveraging simple word embeddings for node features, we also incorporate POS-tags and a weighted loss function to enhance performance and address data imbalance.

Our results indicate that graph-based models, particularly those in the hyperbolic space, can achieve performance superior to SOTA transformer-based architectures. The P-HGNN-POS model, our best-performing configuration, achieves a test F1-score of **84.0%** and an accuracy of **92.1%**, even surpassing the **91.7%** accuracy of the much larger $\text{RoBERTa}_{\text{large}}$ model. This performance is achieved with approximately **12 million parameters**, a nearly 30-fold reduction compared to the **355 million parameters** of $\text{RoBERTa}_{\text{large}}$. These findings highlight the potential of graph-based models as lightweight, efficient, and effective alternatives to LLMs for specialized NLP tasks.

Takeaway for RQ.1: Graph-based models offer a computationally efficient alternative to large transformers for environmental claim detection without compromising performance.

Furthermore, our results demonstrate the potential of hyperbolic geometry for NLP tasks like claim detection. Across various configurations, HGNNs consistently outperform GNNs, and this performance gap becomes more pronounced with the introduction of richer syntactic features, as seen in the superior performance of P-HGNN-POS over GNN-POS. This suggests that tree-like modeling of sentence structure creates a hierarchical representation that is naturally well-suited to the geometric properties of hyperbolic space.

Takeaway for RQ.2: Explicit hierarchical modeling of claims significantly benefits hyperbolic models, indicating their potential for NLP tasks.

Our findings underscore two critical points for the field. First, the dominance of transformer-based models is not absolute; for specific, well-defined tasks like environmental claim detection, specialized and lightweight models like GNNs can provide more efficient and effective solutions. Second, the inherent, often implicit, hierarchical nature of linguistic data can be powerfully exploited by choosing geometric spaces – like hyperbolic space – that align with this underlying structure. This highlights the vast potential in exploring geometries beyond euclidean for learning efficient representations for NLP tasks.

7 Conclusion

In this work, we introduce an efficient graph-based methodology for environmental claim detection,

[2] See Appendix for more details.

Model	grid-search parameters		dev					test				
	Dropout Rate	Class Weights	pr	rc	F1	acc	auc	pr	rc	F1	acc	auc
GNN	0.1	–	**79.3**	69.7	74.2	**87.9**	**0.93**	78.7	71.6	75.0	87.9	**0.93**
Balanced-GNN	0.1	[0.6678,1.9897]	68.3	**84.8**	75.7	86.4	**0.93**	69.2	**80.6**	74.5	86.0	**0.93**
Balanced-GNN	0.1	[0.8,1.6]	72.4	83.3	**77.5**	**87.9**	**0.93**	70.1	**80.6**	75.0	86.4	**0.93**
Balanced-GNN	0.1	[1,1.5]	78.6	66.7	72.1	87.2	**0.93**	**81.7**	73.1	**77.2**	**89.1**	**0.93**
L-HGNN	0.1	–	70.3	78.8	74.3	86.4	0.92	73.7	**83.6**	**78.3**	**88.3**	**0.93**
Balanced-L-HGNN	0.1	[0.6678,1.9897]	71.6	**80.3**	75.7	87.2	**0.93**	68.8	82.1	74.8	86.0	**0.93**
Balanced-L-HGNN	0.1	[0.8,1.6]	75.7	**80.3**	**77.9**	**88.7**	**0.93**	73.0	80.6	76.6	87.5	**0.93**
Balanced-L-HGNN	0.1	[1,1.5]	**79.7**	71.2	75.2	88.3	**0.93**	**75.4**	73.1	74.2	87.2	**0.93**
P-HGNN	0	–	**71.0**	74.2	72.6	86.0	0.91	**74.4**	**86.6**	**80.0**	**89.1**	**0.94**
Balanced-P-HGNN	0	[0.6678,1.9897]	69.1	**84.8**	**76.2**	**86.8**	**0.92**	71.8	83.6	77.2	87.5	0.93
Balanced-P-HGNN	0	[0.8,1.6]	69.9	77.3	73.4	86.0	0.91	68.2	**86.6**	76.3	86.4	0.93
Balanced-P-HGNN	0	[1,1.5]	68.4	78.8	73.2	85.7	**0.92**	71.2	85.1	77.6	87.5	0.93

Table 3: Results for all weight-balanced GNNs. For the best base GNN configurations, we show all the corresponding weight-balanced GNNs at the same dropout rate as the base model. For each model type, i.e., GNN, L-HGNN and P-HGNN, the best performance per split is indicated block-wise in bold, while the second best in underlined.

Model	Parameter-count
DistilBERT	66m
ClimateBERT	82m
$\text{RoBERTa}_{\text{base}}$	125m
$\text{RoBERTa}_{\text{large}}$	**355m**
GNN/HGNN	12m
GNN-POS/HGNN-POS	12m

Table 4: Number of parameters for the transformer models used by (Stammbach et al., 2022) compared to our GNN and HGNN models. Models prefixes are dropped since they do not affect the parameter sizes. m stands for million. Largest model size is in bold while the smallest is underlined.

positioning GNNs and HGNNs as lightweight yet effective alternatives to transformer-based architectures. We reformulate the task as a graph-classification problem, transforming claim sentences into dependency parsing graphs with simple word and POS-tag embeddings as node features and encoding syntactic dependencies as edge relations. Our results demonstrate that GNNs achieve performance comparable or superior to SOTA models with a 30-fold reduction in parameters. Furthermore, we consistently observe that HGNNs outperform their GNNs, affirming that the geometric properties of HGNNs gain significant advantage from the explicit hierarchical modeling of the data. Our findings call for a shift beyond over-reliance on transformers, demonstrating that specialized models can yield more efficient solutions for targeted NLP tasks without a loss of capability.

Future work. First, we plan to compare static word2vec embeddings for node features with sentence embeddings from transformer models like RoBERTa. Second, we plan to move beyond simple one-hot encoded edge features to a knowledge-enhanced schema based on principles from universal dependencies (De Marneffe et al., 2021). Third, we plan to experiment with alternative graph representations beyond dependency parsing such as constituency parsing. Fourth, we plan to conduct a sensitivity analysis to quantify the impact of parsing inaccuracies on model performance, investigating whether domain-adapted parsers could yield better results. Lastly, to assess the generalisability of this study, we intend to extend our work to more NLP tasks, models such as graph attention networks (Veličković et al., 2017), and benchmark datasets such as FEVER (Thorne et al., 2018) and Climate-Fever (Diggelmann et al., 2020).

8 Limitations

We highlight the limitations of our work as follows.

- Our approach deliberately utilizes word2vec embeddings for node features to create a maximally lightweight and efficient model. However, they do not encode the sequential dependencies between words. Similarly, our edge features are simple one-hot encodings of dependency types, which treat all syntactic relations as independent and do not capture potential similarities between them.
- Our methodology relies on the output of the dependency parser to construct the graphs. While modern parsers are highly accurate, any errors in graph construction are propagated as noise to the GNN and HGNN models. We do not analyze the impact of such parsing errors on final model performance in this study.
- The scope of our experiments is focused on a

single, relatively small, English-only dataset. While the results are strong, the generalisability of our graph-based approach to other claim detection domains, larger datasets, and other languages is yet to be established.

- The transformer baselines used for comparison are from the original environmental claim detection paper (Stammbach et al., 2022). We do not benchmark our models against more recent, state-of-the-art LLMs such as Llama3 (Dubey et al., 2024) and GPT-4o (Hurst et al., 2024), limiting the assessment of our approach against the current SOTA.

References

Esma Aïmeur, Sabrine Amri, and Gilles Brassard. 2023. Fake news, disinformation and misinformation in social media: a review. *Social Network Analysis and Mining*, 13(1):30.

Ahmed Al-Rawi, Derrick O'Keefe, Oumar Kane, and Aimé-Jules Bizimana. 2021. Twitter's fake news discourses around climate change and global warming. *Frontiers in Communication*, 6:729818.

Pepa Atanasova. 2024. Generating fact checking explanations. In *Accountable and Explainable Methods for Complex Reasoning over Text*, pages 83–103. Springer.

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42.

Miles Brundage, Shahar Avin, Jasmine Wang, Haydn Belfield, Gretchen Krueger, Gillian Hadfield, Heidy Khlaaf, Jingying Yang, Helen Toner, Ruth Fong, et al. 2020. Toward trustworthy ai development: mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*.

Zina Chkirbene, Ridha Hamila, Ala Gouissem, and Unal Devrim. 2024. Large language models (llm) in industry: A survey of applications, challenges, and trends. In *2024 IEEE 21st International Conference on Smart Communities: Improving Quality of Life using AI, Robotics and IoT (HONET)*, pages 229–234. IEEE.

Travis G Coan, Constantine Boussalis, John Cook, and Mirjam O Nanko. 2021. Computer-assisted classification of contrarian claims about climate change. *Scientific reports*, 11(1):22320.

Harold Scott Macdonald Coxeter. 1998. *Non-euclidean geometry*. Cambridge University Press.

Sebastião Vieira de Freitas Netto, Marcos Felipe Falcão Sobral, Ana Regina Bezerra Ribeiro, and Gleibson Robert da Luz Soares. 2020. Concepts and forms of greenwashing: A systematic review. *Environmental Sciences Europe*, 32:1–12.

Marie-Catherine De Marneffe, Christopher D Manning, Joakim Nivre, and Daniel Zeman. 2021. Universal dependencies. *Computational linguistics*, 47(2):255–308.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2020. Climate-fever: A dataset for verification of real-world climate claims. *arXiv preprint arXiv:2012.00614*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2023. Llmcarbon: Modeling the end-to-end carbon footprint of large language models. *arXiv preprint arXiv:2309.14393*.

Tao Feng, Yihang Sun, and Jiaxuan You. 2025. Grapheval: A lightweight graph-based llm framework for idea evaluation. *arXiv preprint arXiv:2503.12600*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Priyanka Kargupta, Runchu Tian, and Jiawei Han. 2025. Beyond true or false: Retrieval-augmented hierarchical analysis of nuanced claims. *arXiv preprint arXiv:2506.10728*.

Zeba Khanam, BN Alwasel, H Sirafi, and Mamoon Rashid. 2021. Fake news detection using machine learning approaches. In *IOP conference series: materials science and engineering*, volume 1099, page 012040. IOP Publishing.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *Proceedings of COLING 2014, the 25th International conference on computational linguistics: Technical Papers*, pages 1489–1500.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020a. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Jun Li, Yifan Cao, Jiong Cai, Yong Jiang, and Kewei Tu. 2020b. An empirical comparison of unsupervised constituency parsing methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3278–3283.

Youjia Li, Vishu Gupta, Muhammed Nur Talha Kilic, Kamal Choudhary, Daniel Wines, Wei-keng Liao, Alok Choudhary, and Ankit Agrawal. 2025. Hybrid-llm-gnn: integrating large language models and graph neural networks for enhanced materials property prediction. *Digital Discovery*, 4(2):376–383.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. 2022. A survey of transformers. *AI open*, 3:111–132.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Qi Liu, Maximilian Nickel, and Douwe Kiela. 2019. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32.

Li Mi and Zhenzhong Chen. 2020. Hierarchical graph attention network for visual relationship detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13886–13895.

Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdl Safran, Sultan Alfarhood, and MF Mridha. 2024. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and llm. *Scientific Reports*, 14(1):9603.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Gregory L Naber. 2012. *The geometry of Minkowski spacetime*. Springer.

Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2025. A comprehensive overview of large language models. *ACM Transactions on Intelligent Systems and Technology*, 16(5):1–72.

Jingwei Ni, Minjing Shi, Dominik Stammbach, Mrinmaya Sachan, Elliott Ash, and Markus Leippold. 2024. Afacta: Assisting the annotation of factual claim detection with reliable llm annotators. *arXiv preprint arXiv:2402.11073*.

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30.

Maximillian Nickel and Douwe Kiela. 2018. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International conference on machine learning*, pages 3779–3788. PMLR.

Joakim Nivre. 2010. Dependency parsing. *Language and Linguistics Compass*, 4(3):138–152.

Andreas L Opdahl, Tareq Al-Moslmi, Duc-Tien Dang-Nguyen, Marc Gallofré Ocaña, Bjørnar Tessem, and Csaba Veres. 2022. Semantic knowledge graphs for the news: A review. *ACM Computing Surveys*, 55(7):1–38.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. 2021. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044.

Piotr Przybyla. 2020. Capturing the style of fake news. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 490–497.

William F Reynolds. 1993. Hyperbolic geometry on a hyperboloid. *The American mathematical monthly*, 100(5):442–455.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *European Conference on Information Retrieval*, pages 359–366. Springer.

Dominik Stammbach, Nicolas Webersinke, Julia Bingler, Mathias Kraus, and Markus Leippold. 2022. Environmental claim detection. *Available at SSRN 4207369*.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. *arXiv preprint arXiv:1803.05355*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Ben Wasike. 2023. You've been fact-checked! examining the effectiveness of social media fact-checking against the spread of misinformation. *Telematics and Informatics Reports*, 11:100090.

Xiao-Kun Wu, Min Chen, Wanyi Li, Rui Wang, Limeng Lu, Jia Liu, Kai Hwang, Yixue Hao, Yanru Pan, Qingguo Meng, et al. 2025. Llm fine-tuning: Concepts, opportunities, and challenges. *Big Data and Cognitive Computing*, 9(4):87.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. Tener: adapting transformer encoder for named entity recognition. *arXiv preprint arXiv:1911.04474*.

Jingyuan Yi, Zeqiu Xu, Tianyi Huang, and Peiyang Yu. 2025. Challenges and innovations in llm-powered fake news detection: A synthesis of approaches and future directions. In *Proceedings of the 2025 2nd International Conference on Generative Artificial Intelligence and Information Security*, pages 87–93.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning*, pages 41092–41110. PMLR.

Min Zhou, Menglin Yang, Bo Xiong, Hui Xiong, and Irwin King. 2023. Hyperbolic graph neural networks: A tutorial on methods and applications. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5843–5844.

A Appendix

A.1 Generating Dependency Parsing Graphs of Environmental Claims

We now provide a working example of the process of converting claim sentences into their dependency parsing graphs. The feature vectors shown are for illustrative purposes and do not represent actual embedding values. Let the claim sentence $\mathbf{C}$ be *"Gas is also a cleaner fuel with resultant environmental benefits."*

- **Dependency Parsing $\mathbf{C}$:** The claim sentence $\mathbf{C}$ is first transformed into its corresponding dependency parsing graph using the spaCy dependency parser. Figure 2 illustrates this transformation.
- **Node Features (x'_v):** Each node's feature vector is the concatenation of its word embedding and a randomly initialized, trainable POS tag embedding. For the node 'cleaner' (an 'ADJ'), with $d_{word} = 4$ and $d_{pos} = 2$,

$$x'_{\text{cleaner}} = [\underbrace{W_e(\text{'cleaner'})}_{\text{word2vec}} \,\|\, \underbrace{W_p(\text{'ADJ'})}_{\text{POS emb.}}]$$

$$= \left[\begin{pmatrix} 0.21 \\ -0.45 \\ 0.67 \\ 0.09 \end{pmatrix} \middle\| \begin{pmatrix} 0.62 \\ 0.15 \end{pmatrix} \right] = \begin{pmatrix} 0.21 \\ -0.45 \\ 0.67 \\ 0.09 \\ 0.62 \\ 0.15 \end{pmatrix}$$

- **Edge Features (e_{hj}):** Each dependency relation is one-hot encoded. The *amod* relation from 'fuel' to 'cleaner', being the 5th unique relation out of 45, is represented as: $e_{\text{fuel, cleaner}} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & \cdots & 0 \end{pmatrix}^T \in \mathbb{R}^{45}$

A.2 Training Configuration

Table 5 shows the (fixed) hyperparameters used for training our GNN and HGNN models. Dropout rate, POS-embedding dimension and class-weights were optimized through grid-search.

A.3 Evaluation Metrics

Let TP, FP, TN, and FN be the number of True Positives, False Positives, True Negatives, and False Negatives, respectively. The metrics are defined as follows.

- **Accuracy:** The proportion of correctly classified instances among the total instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Hyperparameter	Value
Number of GNN Layers	4
Learning Rate	0.001
Hyperbolic Learning Rate	0.001
Patience (Early Stopping)	8
Activation Function	Leaky ReLU
Leaky ReLU Slope	0.5
Optimizer	AMSGrad
Hyperbolic Optimizer	Riemannian AMSGrad
Embedding Dimension	256
Number of Centroids	30
Maximum Epochs	30
Edge Types	45
Number of Classes	2
Initialization Method	Xavier
Gradient Clipping	1.0

Table 5: GNN and HGNN Training Configuration

- **Precision:** The ratio of correctly predicted positive observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall:** The ratio of correctly predicted positive observations to all observations in the actual class.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:** The harmonic mean of Precision and Recall.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **AUC-ROC:** The Area Under the Receiver Operating Characteristic Curve. It measures the model's ability to distinguish between positive and negative classes across all classification thresholds.

A.4 Parameter Size Calculation for Graph Models

Here, we detail the number of trainable parameters for our graph models. We first calculate the number of trainable parameters for the base model (with only word embeddings) and then for the model augmented with POS-tag embeddings (-POS).

Base GNN/HGNN Model (without POS). The base model's parameters are distributed across an input projection layer, three hidden GNN layers, and a final classifier.

- Layer 1 (Input Projection): Maps the 300-dimensional word embeddings to the 256-dimensional hidden space for each of the 45

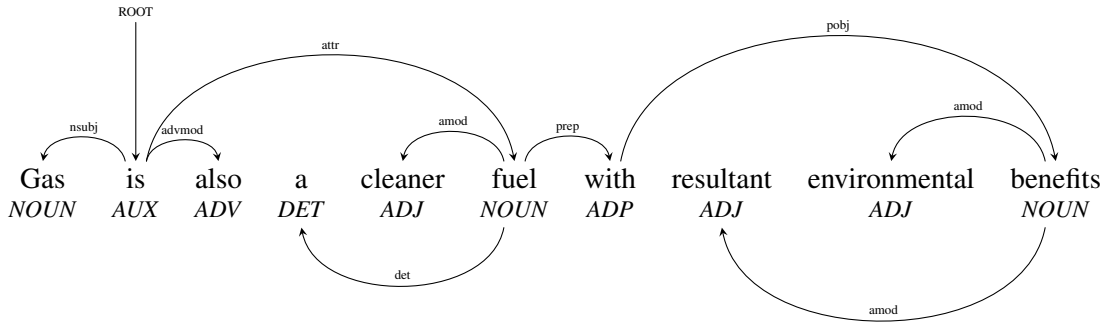

Figure 2: The transformation of the example claim into a dependency graph. The graph shows tokens and their POS tags as nodes, with syntactic dependencies as labeled, directed edges.

relations. 45 relations $\times$ 300 input_dim $\times$ 256 output_dim) + 256 bias = 3,456,256.

- Layers 2, 3, & 4: These three layers map the 256-dimensional hidden state to another 256-dimensional hidden state for each relation. 3 layers $\times$ [(45 relations $\times$ 256 input_dim $\times$ 256 output_dim) + 256 bias] = 8,848,128.
- Final Classifier: (256 $\times$ 2 output classes) + 2 bias = 514.
- Total number of parameters = 3,456,256 + 8,848,128 + 514 = **12,304,898**.

GNN-POS/HGNN-POS Model (with POS). Adds learnable POS embeddings. Using an example POS dimension (d_{pos}) be 16:

- POS Tag Embeddings: 18 vocab_size $\times$ 16 pos_dim = 288
- Layer 1 (Input Projection): The input dimension is now 300 + 16 = 316. (45 $\times$ 316 $\times$ 256) + 256 = 3,640,576.
- Layers 2, 3, & 4: Unchanged from the base model. Parameters: 8,848,128
- Final Classifier: Unchanged from the base model. Parameters: 514
- Total number of parameters (POS) = 288 + 3,640,576 + 8,848,128 + 514 = **12,489,506**.