IS CONTEXT ATTRIBUTION ALL YOU NEED TO ATTAIN GENERALIZABILITY IN NON-FINE-TUNED TRANSFORMER? A FRAMEWORK FOR FAKE NEWS DETECTION IN CROSS DATASET EVALUATION SETTINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training Large Language Models (LLMs) is a costly and lengthy process. Besides, when trained on a particular dataset, the model loses a big part of its *generalizability*. We propose a novel method of context attribution for the transformer model that proves to be more efficient and *generalizable*. We show that in an example of a fake news detection task, utilizing three distinct datasets and outperforming the baseline model in both the same dataset and cross-dataset zero-shot test. Particularly, with our method, we observe improvement by **15%** in terms of accuracy and by **18%** in terms of F1 score on the Covid Fake dataset, **3.5%** of accuracy and **6%** on F1 with the LIAR dataset, **23%** of accuracy and **25%** of F1 on Kaggle Fake News Dataset, and **5%** accuracy and **25%** of F1 on Fake News Net. All results are calculated compared to the BERT model with a fine-tuned classification layer for the fake detection task. In a cross-dataset zero-shot test using fine-tuned Fake News Net dataset to predict Covid Fake data, we observe **5%** accuracy and **25%** F1 boost. We also introduce a novel loss function with corresponding auxiliary metrics. This loss shows better *generalizibility* properties and faster convergence. It also stabilizes the training stage, producing smoother and more reliable training curves. Besides, we show that using our model, one can see if the dataset is descriptive of the domain area or not by measuring how well it generalizes across topics and datasets.

1 INTRODUCTION

Training Large Language Models (LLMs) is a very expensive and lengthy process. Besides, when trained on a particular dataset, the model loses much of its *generalizability*. As a potential solution to the problem, we propose a new framework that is used with transformer (Vaswani et al., 2017) architecture in order to improve results not only for the specific task but with a great effect of **generalizibility**. We call this framework a **Context Attribution Model (CAM)** and show that using it instead of a classification layer in transformer models such as BERT (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019) and others is a way of increasing models' performance.

2 DATASETS

For this research, we utilize four different datasets of fake news 3. Three of them are binary labels: Fake COVID (Patwa et al., 2020), Fake News Kaggle (Lifferth, 2018), FakeNewsNet (Shu et al., 2018), and one contains multiple labels LIAR (multi-label) (Wang, 2017). So, we prove that our model evaluates better in both binary and multi-label approaches.

3 METHODOLOGY

3.1 PROBLEM DEFINITION

Consider a large language model that takes a document $d = [t_1, t_2, ..., t_T] = [t_i]$, i.e., a sequence of tokens of length T, as input and produces a sequence of contextual embedding for each token in N-dimensional vector space, R^N. We can define such language model as: $E = LLM([t_i]) = (\vec{e}_1, \vec{e}_2, \vec{e}_3, ..., \vec{e}_{n_d})$, where $\vec{e}_k$ is in R^N.

Any categorical label is a concept that can have multiple attributes representing it. For example, the label "fake" can be attributed to terms or a set of concepts that represent or contextualize the fake pieces of news. On that idea, we extrapolate the categorical label $y^{(i)}$ as a spectrum for several concept attribution terms (i.e., synonyms representing the concept), $C_{y^{(i)}} = c_1, c_2, c_3, \ldots, c_{m_C}$, such that; $\forall i \exists j : ATTR(C^{(j)}, y^{(i)})$, meaning - each categorical label has at least one concept attribution term. Thus a given classification task can be expressed as, $\tau = \tau_1, \tau_2, \tau_3, \ldots, \tau_{l_\tau}$, comprising subtask, t_i for each label $y^{(i)}$, of attributing the document $[d]$ by learning the attribution transformation function:

$$C_{y^{(i)}} = ATTR(E, T_{y^{(i)}})$$

Based on these assumptions we define the following optimization task: there exists a learnable transformation tensor, $T_{y^{(i)}}$, for each label $y^{(i)}$ such that it can operate on $LLM([t_i])$ token representations $\vec{e}_k$ for a given document $[d]$ to project those tokens on to an abstract concept space. The tensor $T_{y^{(i)}}$ can be optimized using customized loss functions during the training for a classification task. Vectors in the concept space can be used as the attributions in the downstream tasks to improve the discriminator function's performance and provide *generalizability* for the classification of $LLM([t_i])$.

3.2 ALGORITHM PROPOSAL

We find projection of all document tokens, collectively defined as E, by applying attribution transformation operator, $C_{y^{(i)}} = ATTR(E, T_{y^{(i)}})$, where $E_{T\times n}, T_{n\times|C_{y^{(i)}}|} \implies P_{T\times|C_{y^{(i)}}|}$. $P_{y^{(i)}}$ - is the projection of E using learned tensor $T_{y^{(i)}}$ As our hypothesis states, to use the concept attribution vectors in the subspace $R^{|C_{y^{(i)}}|}$ for the downstream tasks, we apply the $tanh$ normalization function. This helps us to scale the subspace vector components within a bounded region of $[-1, +1]$. Thus each token in $LLM([t_i])$ will have a concept space vector $\vec{p}_{y^{(i)}}$ in $R^{|C_{y^{(i)}}|}$, where each component of the vector represents the attribution score for the respective concept [1].

4 RESULTS

We prove the efficiency of our novel framework via multiple benchmarks. Our data comes from different sources, such as Twitter, political websites, news websites, and more. Our main contributions include:

- Demonstration of a significant boost in all metrics for the task of fake news classification
- Our approach outperforms existing solutions in cross-dataset zero-shot evaluations
- We achieve stabilization of the training process, ensuring consistent and reliable model performance over time

[1] Full derivation of the attribution function and details of the auxiliary loss computation are provided in Appendix A

Table 1: Invidual Dataset Experimental Results

Data	Architecture	# Dim.	Accuracy	CS Accuracy	F1	CS F1
Fake COVID News	BERT	N/A	0.7145	N/A	0.6966	N/A
	CAM-BERT	64	**0.8645**	**0.7743**	0.8627	**0.7637**
Liar (multi-label)	BERT	N/A	0.2221	N/A	0.1211	N/A
	CAM-BERT	64	**0.2580**	**0.2362**	0.2034	**0.1590**
Liar (binary-label)	BERT	N/A	0.5666	N/A	0.3617	N/A
	CAM-BERT	64	**0.6267**	**0.5877**	**0.6002**	**0.5855**
Kaggle Fake News	BERT	N/A	0.6550	N/A	0.6337	N/A
	CAM-BERT	128	**0.8868**	0.6436	**0.8859**	0.5900
Fake News Net	BERT	N/A	0.7548	N/A	0.4302	N/A
	CAM-BERT	128	**0.8028**	**0.7131**	**0.6833**	**0.6302**

We use the datasets provided in the table 3 for the evaluation. All of our evaluations are based on the BERT transformer architecture. We train the model on different data and test it through different domains (e.g., training on political news and testing on COVID-related news). We also show that by using the CAM Framework, we don't need to train the whole underlying model. With CAM-Model fine-tuning, we achieve performance close to the entire model training but with far less memory consumption during training and without the loss in *generalizability*.

4.1 Evaluation Metrics

As a standard measure, we report accuracy and F1 Macro score, as well as precision and recall. Due to the nature of our novel loss, we also introduce additional metrics: Concept Space Accuracy and Concept Space F1. This is evaluated as accuracy and F1-score, where the label is chosen based on the $tanh$ normalization between concept spaces (the label corresponding to the closest concept space is assigned[2]). We obtain two latent representations after projecting the transformer embedding onto the concept spaces. On the means of these representation vectors, we apply $tanh$ to obtain normalized values, and based on the concept proximity, space is assigned. The interpretation of the CS Accuracy and CS F1 is exactly the same as for the standard Accuracy and F1, only calculation changes.

4.2 Experimental Results

4.2.1 Invidividual Dataset Evaluation, CAM-BERT vs BERT

After fine-tuning the default BERT and various configurations of CAM-BERT on all of the datasets that we select as our benchmarks, we compare them in the table 1 [3]. This experiment set proves that our model is superior if used for the same dataset. We also claim it is superior and generalizes much better than fine-tuned BERT.

4.2.2 Cross-dataset Evaluation, CAM-BERT vs BERT

We define labels in the following manner: 1 - stands for fake for all of the datasets, and 0 - encodes true. By doing that, we ensure that the evaluation is consistent so that we can test it in a zero-shot manner. For the cross-dataset case 2 [4] we see that the best-performing model is the one with 128 latent dimensions. It

[2]Review Appendix A section 2 for more details on how we calculate these metrics

[3]This is a truncated table with the most important details, please find the complete result in Appendix A 4

[4]This is a truncated table with the most important details, please find the complete result in Appendix A 5

Table 2: Cross-dataset Experimental Results

(Train) → (Test)	**Model**	**# Dims.**	**Accuracy**	**CS Accuracy**	**F1**	**CS F1**
(Gossip) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	3	0.5234	**0.6145**	0.3436	**0.6052**
(Gossip) → (Politifact)	BERT	N/A	0.5909	N/A	0.3714	N/A
	CAM-BERT	128	**0.6941**	**0.6468**	**0.6395**	0.6155
(NewsNet) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	64	**0.5482**	**0.6191**	**0.4064**	**0.6116**

could capture the most context attributions of the fake and non-fake news. The performance gain by using the CAM model compared to BERT is evident since in all of the cases without any model retraining, by only doing fine-tuning, we achieve a performance boost of **25%** on the F1-score.

4.3 Ablation study of the Inter-Space and Intra-space losses

We test two situations for the auxiliary losses. In the first set of experiments, Loss is defined as a weighted sum of the Cross-Entropy Loss with the Auxiliary Losses. In the other set of experiments, we exclusively train the Inter-Space and Intra-Space objective functions [5]. From this study, We conclude that sometimes training only inter-space and intra-space objectives may lead to better results and faster convergence, also proving better *generlizability* without utilizing additional layers.

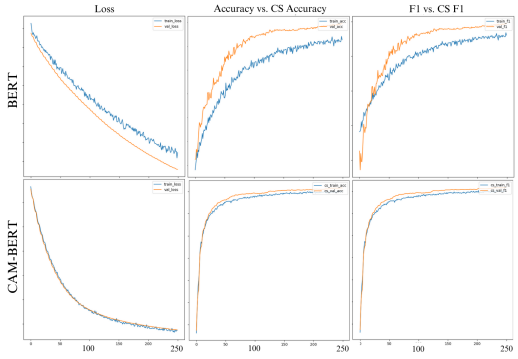

Figure 1: Ablation study comparison of BERT vs. CAM-BERT with Auxiliary Losses and Auxiliary Metrics

5 Conclusion

In summary, this study concentrated on a new approach to concept embedding for classification in the domain of automated fake news detection using language models. The research involved conducting experiments to validate the effectiveness of the suggested technique in multiple scenarios with the same dataset and cross-dataset. We have also introduced a novel loss function that proves to be efficient for the given task and allows better *generalizability*, faster convergence, and more stable and reliable training. This loss function and the model architecture led to the definition of Concept Space metrics, such as CS Accuracy and CS F1. We explain how to evaluate those and how they help us better understand and utilize the CAM Framework.

[5]The results are available in Appendix A 6, 7, 1

REFERENCES

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

William Lifferth. Fake news, 2018. URL https://kaggle.com/competitions/fake-news.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Parth Patwa, Shivam Sharma, Srinivas PYKL, Vineeth Guptha, Gitanjali Kumari, Md. Shad Akhtar, Asif Ekbal, Amitava Das, and Tanmoy Chakraborty. Fighting an infodemic: COVID-19 fake news dataset. *CoRR*, abs/2011.03327, 2020. URL https://arxiv.org/abs/2011.03327.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. URL https://api.semanticscholar.org/CorpusID:203626972.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *CoRR*, abs/1809.01286, 2018. URL http://arxiv.org/abs/1809.01286.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

William Yang Wang. "liar, liar pants on fire": A new benchmark dataset for fake news detection. *CoRR*, abs/1705.00648, 2017. URL http://arxiv.org/abs/1705.00648.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/dc6a7e655d7e5840e66733e9ee67cc69-Paper.pdf.

A APPENDIX

A.1 DATASETS

Table 3: Benchmarks Tested

Dataset	Total true news	Total fake news
Fake COVID (Patwa et al., 2020)	3,360	3,060
LIAR (multi-label) (Wang, 2017)	6,400	6,400
LIAR (binary-label)	6,400	6,400
Fake News Kaggle (Lifferth, 2018)	10,387	10,413
FakeNewsNet (Shu et al., 2018)	18,000	6,000

A.2 ATTRIBUTION FUNCTION AND AUXILIARY LOSSES

The transformation function, $ATTR(E, T_{y^{(i)}})$ is computed as follows:

- **Linear Transformation**: for each attribution for the concept, $C_{y^{(i)}} = c_1, c_2, c_3, \ldots, c_{m_C}$ we perform a dot product with all tokens resulting in, $P_{y^{(i)}} = E \times T_{y^{(i)}}$
- **Normalization Constraints**: As our hypothesis states, to use the concept attribution vectors in the subspace $R^{|C_{y^{(i)}}|}$ for the downstream tasks, we apply the $tanh$ normalization function. This helps us to scale the subspace vector components within a bounded region of $[-1, +1]$. Thus, we define:

$$ATTR(E, T_{y^{(i)}}) = tanh[\forall_{j,k}(P_{y^{(i)}})] = tanh[\forall_{j,k}(E \times T_{y^{(i)}})]$$

  Thus, each token in $LLM([d])$ will have a concept space vector $\vec{p}_{y^{(i)}}$ in $R^{|C_{y^{(i)}}|}$, where each component of the vector represents the attribution score for the respective concept to the token.
- **Regularization And Generalization Terms**: We define these terms to avoid overfitting and to achieve *generalizability* as follows:

$$w(\theta) = \frac{1}{|C_{y^{(i)}}| \cdot T} \sum_{\forall_{j,t}} tanh[P_{j,t}])$$

$$J_1(\theta) = (L \cdot [\sum_{k=1}^{|C_{y^{(i)}}|} (1 - w(\theta)] + (1 - L) \cdot [\sum_{k=1}^{|C_{y^{(i)}}|} (1 + w(\theta))]$$

  Where L is a vector of labels where each label is in 1 to 1 correspondence with the number of concept spaces.

$$J_2(\theta) = \sum_{k=1}^{|C_{y^{(i)}}|} \frac{1}{\sum_{c_j \in C_{y^k}}^{n} Var[c_j]}$$

A.3 TRAINING OBJECTIVE

As defined in the previous section, we use auxiliary losses in conjunction with the main Cross Entropy Loss function. The final form of an objective function is as follows:

$$loss = CrossEntropyLoss(X, y|\theta) + \lambda_1 J_1(P_{y^{(i)}}, y|\theta) + \lambda_2 J_2(P_{y^{(i)}}|\theta)$$

Where λ_1 and λ_2 are the weights of the auxiliary losses. Setting λ_2 to high values would mean more robust regularization of the intra-space loss. This loss ensures that the set of concepts describing the concept space does not converge into a single representation. Setting λ_1 to higher values would make inter-space loss dominant compared to Cross Entropy. This type of behavior is unique to each specific use case. In the experiment section, we show in some instances, optimizing Concept Spaces representation also optimizes the target logits. However, this might not always be the case.

We provide an illustration to describe the architecture and show which particular parts are used for the loss evaluation in fig 2.

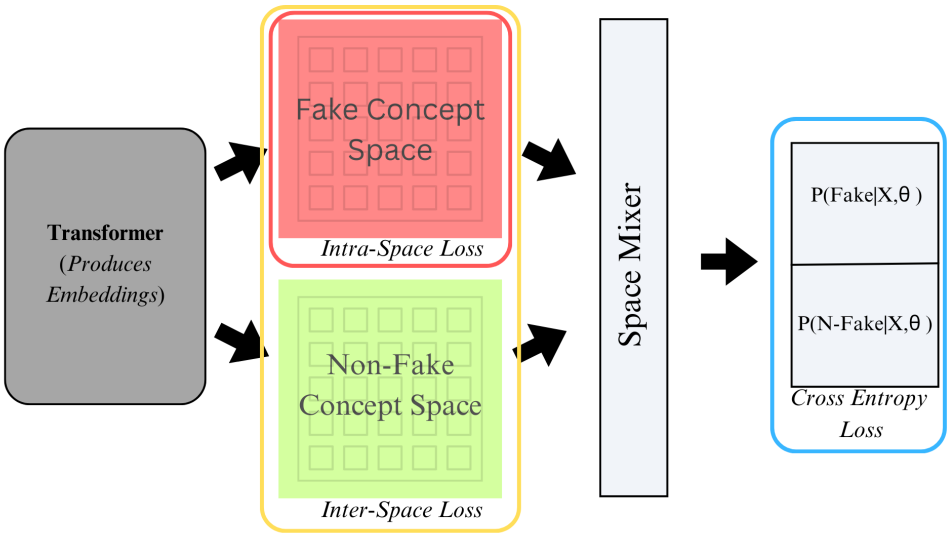

Figure 2: CAM-Model diagram with specifying outputs used for loss components computation

A.4 Context Attribution Model Explained

Previously, we have introduced the Auxiliary Metrics, which help us describe and evaluate how our model performs. The reason these metrics are optimized is based on the definition of the loss function. Let us define the goal of the optimization for the inter-space loss $J_1(\theta)$, introduced in the previous section. In the model, we train transformation tensor $T_{y^{(i)}}$ for each label $y^{(i)}$. Each label is assigned a unique concept space that would attribute the context of this label. For instance, the "fake" concept is assigned to one concept space, and the "non-fake" concept is assigned to another concept space 2 (note that in the general case, we can have arbitrarily many labels, each assigned with specific concept space, attributing its context).

The goal of this model is to extend the existing architecture, so after we have attributed the contexts with some tensor transformation, we want the model to make a decision taking both attributions into account. For that, we introduce the Space Mixer layer. It combines the output of the concepts and learns to attribute given text to the specific label. However, we hypothesize that a decision can be made solely based on the context attribution of the token embeddings on each of the concept spaces.

A.4.1 CALCULATING CS ACCURACY AND CS F1

The main goal of $J_1(\theta)$ is to ensure that the projections on correct and incorrect concept spaces are highly polarized. Specifically, we would want that when the embeddings are projected on the correct concept space (fake concept space for the fake piece of news), we would like that representation to approach certain polarization. When this same fake embedding is projected onto the non-fake concept space, we would like this polarization to be orthogonal.

This is achievable due to the defined $tanh$ normalization that we use in the concept space projection. Based on the loss definition and the fact that the values of the projection dimensions are bounded between -1 and 1, we state that correct representations will have a representation close to the point $[1, 1, ..., 1]$, while incorrect representations will look like $[-1, -1, ..., -1]$ (this is exactly what the definition of the inter-space loss tells us). If that is the case, then the mean of the correct representation would also approach 1 for that case (note, by "correct" here, we mean the one that matches the concept space assigned with a label to the underlying true label of the embeddings that is being processed).

With that, we have a criteria of label assigning. After projecting embeddings on the concept spaces, take the one that has the highest mean of the representation vectors. This would also mean that the produced context attribution closely attributes to this corresponding concept space or label. Formally:

$$pred_{cam} = \underset{i \in |L|}{\text{argmax}}[\frac{1}{T \cdot |C_{y^{(i)}}|} \sum_{t,k}^{T,|C_{y^{(i)}}|} C_{y^i}[t,k]]$$

It is clear that when evaluating this, we do not use the space mixer or anything after that. These metrics are designed to give us an idea of why the final model with auxiliary losses works well and *generalizes* much better. Training this loss not only gives some very good results in both the same dataset and cross-dataset tests but also converges at much higher rates. This metric is a good measurement of our generalizability. To prove that, one can find the experiments we run and see that sometimes, when we make zero-shot tests, predictions provided by the concept spaces only, without a space mixer, are more accurate. That means that by just attributing the context, we capture the intricate structure of the problem rather than the structure of the certain dataset.

A.5 FULL PERFORMANCE COMPARATIVE STUDY

Table 4: Same Dataset Experimental Results

Data	Architecture	# Dim.	Accuracy	CS Accuracy	F1	CS F1
Fake COVID News	BERT	N/A	0.7145	N/A	0.6966	N/A
	CAM-BERT	3	0.7949	0.5234	0.7828	0.3436
	CAM-BERT	64	**0.8645**	**0.7743**	0.8627	**0.7637**
	CAM-BERT	128	0.8808	0.6528	**0.8797**	0.5967
Liar (multi-label)	BERT	N/A	0.2221	N/A	0.1211	N/A
	CAM-BERT	3	0.2362	0.1824	0.1540	0.1079
	CAM-BERT	64	**0.2580**	**0.2362**	0.2034	**0.1590**
Liar (binary-label)	BERT	N/A	0.5666	N/A	0.3617	N/A
	CAM-BERT	3	0.5900	0.5838	0.5280	0.5825
	CAM-BERT	64	**0.6267**	**0.5877**	**0.6002**	**0.5855**
	CAM-BERT	128	0.6251	0.5744	0.5971	0.4194
Kaggle Fake News	BERT	N/A	0.6550	N/A	0.6337	N/A
	CAM-BERT	3	0.8069	0.6408	0.8030	0.6396
	CAM-BERT	64	0.8685	**0.6834**	0.8672	**0.6497**
	CAM-BERT	128	**0.8868**	0.6436	**0.8859**	0.5900
Fake News Net	BERT	N/A	0.7548	N/A	0.4302	N/A
	CAM-BERT	3	0.7557	0.2732	0.4336	0.2400
	CAM-BERT	64	0.7955	0.6468	0.6654	0.6090
	CAM-BERT	128	**0.8028**	**0.7131**	**0.6833**	**0.6302**

Table 5: Cross-dataset Experimental Results

(Train) → (Test)	Model	# Dims.	Accuracy	CS Accuracy	F1	CS F1
(Gossip) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	3	0.5234	**0.6145**	0.3436	**0.6052**
	CAM-BERT	64	**0.5375**	0.6064	**0.3806**	0.5874
	CAM-BERT	128	0.5373	0.5691	0.3797	0.4712
(Gossip) → (Politifact)	BERT	N/A	0.5909	N/A	0.3714	N/A
	CAM-BERT	3	0.5909	0.4403	0.3714	0.3699
	CAM-BERT	64	0.6695	0.6259	0.6085	**0.6258**
	CAM-BERT	128	**0.6941**	**0.6468**	**0.6395**	0.6155
(NewsNet) → (CovidFake)	BERT	N/A	0.5234	N/A	0.3436	N/A
	CAM-BERT	3	0.5234	0.6024	0.3436	0.5881
	CAM-BERT	64	**0.5482**	**0.6191**	**0.4064**	**0.6116**
	CAM-BERT	128	0.5480	0.5956	0.4056	0.5354

A.6 INTER-SPACE AND INTRA-SPACE LOSSES PERFORMANCE STUDY

Note that sometimes, when we only use Inter-Space and Intra-Space loss, we still report Accuracy and F1, even though these are not optimized during training. This is just to show that sometimes optimizing Concept Spaces is the same as optimizing the standard model output layer. On 2, we have shown where each loss is applied. From there, one may see that the Space Mixer weights and classification layer weights are not optimized by Intra-space and Inter-space losses.

Table 6: Evaluation of the model trained with Inter-Space and Intra-Space Losses only (same dataset)

Data	# Dims.	Accuracy	CS Accuracy	F1	CS F1	Precision	Recall
GossipCop	3	**0.7566**	0.7053	0.4307	0.6387	0.3783	0.5000
	64	0.2818	0.6857	0.2538	0.6366	**0.5683**	**0.5175**
	128	0.7448	**0.7442**	**0.4436**	**0.6522**	0.4861	0.4985
Fake News Net	3	0.7548	0.6196	0.4302	0.5962	0.3774	0.5000
	64	0.7815	0.7055	0.6016	0.6325	0.7203	0.5928
	128	**0.7973**	**0.7321**	**0.6703**	**0.6370**	**0.7350**	**0.6500**

Table 7: Evaluation of the model trained with Inter-Space and Intra-Space Losses only (cross-dataset)

(Train) → (Test)	# Dims.	Loss	Accuracy	CS Accuracy	F1	CS F1
(Gossip) → (CovidFake)	3	0.7034	0.5234	0.5292	0.3436	0.3606
	64	0.6892	0.4759	**0.5967**	0.3238	**0.5549**
	128	0.7052	0.4401	0.5543	0.3472	0.4312
(NewsNet) → (CovidFake)	3	0.7046	0.5234	0.5379	0.3436	0.3830
	64	0.6888	0.4761	**0.6260**	0.3238	**0.6039**
	128	0.7055	0.4338	0.5778	0.3467	0.4927