DMIX: Adaptive Distance-aware Interpolative Mixup

Ramit Sawhney[†,*], **Megh Thakkar**[§,*], **Shrey Pandit**[§,*], **Ritesh Soun**[♣]
Di Jin[★], **Diyi Yang**[△], **Lucie Flek**[†]
[†]Conversational AI and Social Analytics (CAISA) Lab, University of Marburg
[§]BITS, Pilani
[♣]Sri Venkateswara College, DU
[★]Amazon Alexa AI
[△]Georgia Institute of Technology
rsawhney@mathematik.uni-marburg.de, lucie.flek@uni-marburg.de

Abstract

Interpolation-based regularisation methods such as Mixup, which generate virtual training samples, have proven to be effective for various tasks and modalities. We extend Mixup and propose DMIX, an adaptive distance-aware interpolative Mixup that selects samples based on their diversity in the embedding space. DMIX leverages the hyperbolic space as a similarity measure among input samples for a richer encoded representation. DMIX achieves state-of-the-art results on sentence classification over existing data augmentation methods on 8 benchmark datasets across English, Arabic, Turkish, and Hindi languages while achieving benchmark F1 scores in 3 times less number of iterations. We probe the effectiveness of DMIX in conjunction with various similarity measures and qualitatively analyze the different components. DMIX being generalizable, can be applied to various tasks, models and modalities.

1 Introduction

Deep learning models, though effective for many applications are prone to overfitting in absence of sufficient training data. Data augmentation techniques can efficiently use this limited training data (Liu et al., 2021; Shi et al., 2020). Interpolation-based augmentation techniques such as Mixup (Zhang et al., 2018) have shown improved performance across different modalities. Mixup over latent representations of inputs leads to further improvements (Chen et al., 2020a). However, Mixup does not account for the spatial distribution of dataset samples, but choosing samples randomly for interpolation-based augmentation.

While randomization in Mixup helps, augmenting Mixup's sample selection strategy with logic based on the similarity of the samples to be mixed can lead to improved generalization (Chen et al., 2020b). The relative spatial position of samples can be leveraged to produce more suitable synthetic inputs for training underlying models (Xu et al., 2021). Further, natural language possesses hierarchical structures and complex geometries, which the standard Euclidean space cannot capture effectively (Ganea et al., 2018). Hyperbolic geometry presents a solution in defining similarity between latent representations (Tifrea et al., 2019).

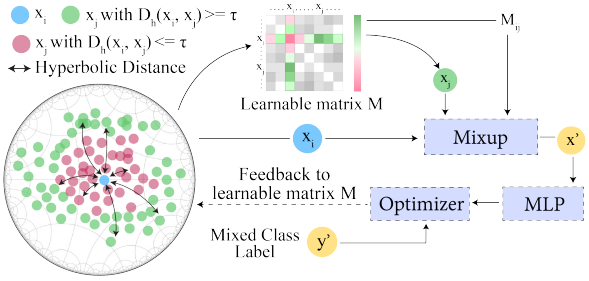

Figure 1: Overview of DMIX showing the sample selection based on the hyperbolic distance and using distance matrix M to perform interpolation.

We propose DMIX, an adaptive distance-aware interpolative data augmentation method. Instead of choosing random inputs from the complete training distribution as in the case of Mixup, DMIX samples instances based on the (dis)similarity between latent representations of samples in the hyperbolic space. Furthermore, DMIX performs interpolations with trainable pair-wise parameters derived from the spatial distribution of the samples rather than sampling mixing ratios randomly from standard distributions, making it adaptive for pair-wise interpolation. Our contributions are:

- We propose DMIX, a novel adaptive distance-aware interpolative regularization method developed over the spatial distribution of dataset sampled in the hyperbolic space.
- DMIX outperforms existing interpolative data augmentation baselines for 8 benchmark sentence classification tasks across four languages.
- DMIX achieves threshold F1 scores with 3 times less number of iterations than random Mixup

[*]Equal contribution.

while being generalizable across tasks, datasets, and modalities.

2 Methodology

We present an overview of DMIX in Figure 1. We first introduce interpolative Mixup (§2.1), and then formulate DMIX by leveraging the relative sample distribution in the hyperbolic space (§2.2).

2.1 Interpolative Mixup

Given two data samples $x_i, x_j \in X$ with labels $y_i, y_j \in Y$, and $i, j \in [1, N]$, Mixup (Zhang et al., 2018) uses linear interpolation with mixing ratio r to generate the synthetic sample x' and corresponding mixed label y',

$$\begin{aligned} x' &= \text{Mixup}(x_i, x_j) = r \cdot x_i + (1-r) \cdot x_j \\ y' &= \text{Mixup}(y_i, y_j) = r \cdot y_i + (1-r) \cdot y_j \end{aligned} \tag{1}$$

Interpolative Mixup (Chen et al., 2020a) performs linear interpolation over the latent representations of models. Let $f_\theta(\cdot)$ be a model with parameters θ having K layers, $f_{\theta,n}(\cdot)$ denotes the n-th layer of the model and h_n is the hidden space vector at layer n for $n \in [1, K]$ and h_0 denotes the input vector. To perform interpolative Mixup at a layer $k \sim [1, K]$, we calculate the latent representations separately for the inputs for layers before the k-th layer. For input sample x_i, we let h_n^i denote the hidden state representations at layer n,

$$\begin{aligned} h_n^i &= f_{\theta,n}(h_{n-1}^i), \quad n \in [1, k] \\ h_n^j &= f_{\theta,n}(h_{n-1}^j), \quad n \in [1, k] \end{aligned} \tag{2}$$

We then perform Mixup over individual hidden state representations h_k^i, h_k^j from layer k as,

$$h_k = \text{Mixup}(h_k^i, h_k^j) = r \cdot h_k^i + (1-r) \cdot h_k^j \tag{3}$$

The mixed hidden representation h_k is used as the input for the continuing forward pass,

$$h_n = f_{\theta,n}(h_{n-1}); \quad n \in [k+1, K] \tag{4}$$

2.2 DMIX: Distance-aware Mixup

Though Mixup helps generalize models better, it selects samples completely randomly for interpolation. Augmenting the sample selection strategy with intelligence derived from the spatial distribution of the samples to be mixed can lead to improved generalization. Hence, we formulate distance-aware Mixup, or DMIX. To perform DMIX, we first create a learnable matrix $\mathbf{M}_{N \text{x} N}$, which is used to perform Mixup between pair of samples. We use the hyperbolic distance as our similarity metric to initialize matrix $\mathbf{M}$ as it effectively captures the hierarchical structures and complex geometries that natural language text possesses. The hyperbolic distance $\mathcal{D}_h$ between sentence embeddings $e_i = f_\theta(x_i)$ and $e_j = f_\theta(x_j)$ is,

$$\mathcal{D}_h(e_i, e_j) = 2 \tan^{-1}(\|(-e_j) \oplus e_i\|) \tag{5}$$

Here, $\oplus$ represents the Möbius addition $\oplus$ for a pair of points $x, y \in \mathcal{B}$, defined as,

$$x \oplus y := \frac{(1 + 2\langle x, y \rangle + ||y||^2)x + (1 - ||x||^2)y}{1 + 2\langle x, y \rangle + ||x||^2||y||^2} \tag{6}$$

, $\langle ., . \rangle$, $|| \cdot ||$ are Euclidean inner product and norm.

We initialize $\mathbf{M}$ using hyperbolic distance $\mathcal{D}_h$ and normalize it row wise to scale the values,

$$\mathbf{M}_{ij} = \mathcal{D}_h(e_i, e_j); \quad \mathbf{M}_i = \frac{\mathbf{M}_i}{max(\mathbf{M}_i)} \tag{7}$$

Using learnable matrix $\mathbf{M}$, we change the Mixup formulation (Equation 1) for samples i and j and define DMixup as,

$$\text{DMixup}(x_i, x_j) = (1 - \mathbf{M}_{ij}) * x_i + \mathbf{M}_{ij} * x_j \tag{8}$$

DMIX is defined for one sample as compared to Mixup which is defined for two samples. To perform DMIX over a sample x_i, we create a set S_i of the most diverse samples in the dataset based on a threshold. To create this set, we select samples having $\mathbf{M}_{ij}$ above a threshold τ,

$$S_i = \{x_k | x_k \in X, \mathbf{M}_{ik} \geq \tau\} \tag{9}$$

We use τ to control the diversity of the selected samples. $\tau = \text{T} * max(\mathbf{M}_i)$ at each step of the training, where T is a hyperparameter $\in (0, 1)$. To perform DMIX, we operate DMixup over samples x_i and a random sample $x_j \in S_i$,

$$\text{DMIX}(x_i) = \text{DMixup}(x_i, x_j), \quad x_j \in S_i \tag{10}$$

We replace the Mixup operation in Equation 3 with the DMIX operation in Equation 10 to evaluate DMIX. The final hidden state output h_K is passed through a multi-layer perceptron (MLP) g_ϕ for classification. We optimize the network using KL Divergence loss between the final output $g_\phi(h_K)$ and mixed label $y' = \text{DMixup}(y_i, y_j)$, which also trains matrix $\mathbf{M}$ end-to-end.

3 Experimental Setup

We evaluate DMIX on standard English, GLUE, and multi-lingual datasets in 4 languages (Table 1).

Dataset	Language	Classes	Samples
TRAC (2020)	English	3	5,329
TREC-Coarse (2002)	English	6	5,952
TREC-Fine (2002)	English	47	5,952
CoLA (2018)	English	2	10,657
SST-2 (2013)	English	2	12,693
AHS (2018)	Arabic	2	3,950
TTC (2017)	Turkish	6	3,600
HASOC (2019)	Hindi	2	5,983

Table 1: Datasets, languages, # classes and # samples.

3.1 Training Setup

DMIX is performed over a layer randomly sampled from all the layers of the model. We use a learning rate of 2e-5, batch size of 8 and a weight decay of 0.01 for all the combinations, DMIX, DMix-NT, and Mixup. For the baselines, we sample r from a beta distribution following previous works. All hyperparameters were selected based on validation F1-score. We use BERT for English and mBERT for other languages as the base model f_θ for our experiments, and their [CLS] token representation as the sentence embeddings to calculate the distances (Equation 5). Due to resource constraints, we only use $10,000$ samples of SST-2 for training, but do not change the validation and test split.

3.2 Evaluation

We compare DMIX with word-mixup (WMix) and sentence-mixup (SMix) (Guo et al., 2019), and interpolative Mixup (TMix) (Chen et al., 2020a)[1].

F1 We use F1 score to evaluate the classification performance of DMIX and its variants.

Diversity Following Gontijo-Lopes et al. (2020), we use diversity defined as the number of training steps required to obtain a benchmark F1 score.

4 Results and Analysis

4.1 Performance Comparison and Ablation

We observe that distance-constrained Mixup significantly ($p < 0.01$) outperforms all baselines across the datasets (Table 2) validating that similarity-based sample selection improves model performance, likely owing to enhanced diversity or minimizing sparsification across tasks. Within distance-constrained Mixup, we observe that DMIX, the hyperbolic distance variant outperforms Euclidean distance (Euc-DMIX) measures (Table 3). This suggests that the hyperbolic space is more capable of capturing the complex hierarchical information present in sentence representations, leading to better comparisons and sample selection. We also compare DMIX and its variants with their non-trainable versions (denoted by -NT in Table 3). These methods have matrix M fixed, and only select samples based on their relative positions in the embedding space. We observe that for all variants, the non-trainable counterparts perform poorer than the trainable counterparts, indicating that M is able to capture sample-specific information relative to other samples, generating more suitable sample selection and mixing ratio for performing interpolative data augmentation.

[1]We provide an extended comparison with other baselines in the Appendix.

Dataset	f_θ	+WMix	+SMix	+TMix	+DMix
TRAC	72.52	73.52	74.20	75.41	**78.67***
TREC-Coarse	97.08	96.10	96.59	97.52	**97.80***
TREC-Fine	86.86	87.13	87.89	90.16	**91.14***
CoLA	84.91	84.95	85.14	85.30	**95.94***
SST-2	90.32	91.34	91.21	91.66	**92.44***
AHS	66.39	67.10	68.30	70.19	**74.98***
TTC	91.10	90.18	91.15	91.30	**92.16***
HASOC	76.13	77.24	76.30	77.44	**80.27***

Table 2: Performance comparison in terms of F1 score of baseline methods with DMIX (average of 10 runs). * shows significant ($p < 0.01$) improvement over TMix.

Dataset	TMix	Euc-DMIX NT	DMIX NT	Euc-DMIX	DMIX
TRAC	75.41	76.52*	78.16*	77.02*	**78.67**$^{*\diamond}$
TREC-Coarse	97.52	97.55	97.66	97.53	**97.80***
TREC-Fine	90.16	89.70	90.20	89.12	**91.14**$^{*\diamond}$
CoLA	85.30	85.73*	86.81*	86.23*	**95.94**$^{*\diamond}$
SST-2	91.05	91.15	92.31*	91.92*	**92.44**$^{*\diamond}$
AHS	70.19	72.23*	74.65*	72.41*	**74.98**$^{*\diamond}$
TTC	91.30	90.66	91.40	91.50	**92.16**$^{*\diamond}$
HASOC	77.44	78.96*	79.96*	79.38*	**80.27**$^{*\diamond}$

Table 3: Ablation study of DMIX with distance constraints using different similarity techniques (average of 10 runs). Improvements are shown with blue. *, $^\diamond$ show significant ($p < 0.01$) improvement over TMix and DMIX-NT, respectively.

4.2 Analyzing Convergence of DMIX

We validate *"Does* DMIX *converge faster than TMix?"*. We observe that across all datasets, DMIX achieves a benchmark F1 score in less number of training iterations compared to TMix (Figure 2). Since DMIX selects samples for Mixup in an adaptive distance-aware manner, it is able to generate more diverse and suitable interpolations leading to faster generalization of the underlying base model. DMIX requires 3 times less number of iterations on an average compared to TMix, or

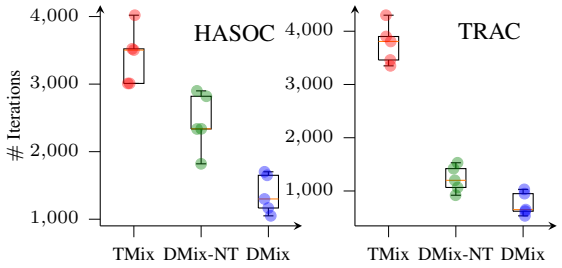

Figure 2: Diversity comparison of TMix with DMIX and DMIX-NT as number of training steps required to achieve benchmark F1 scores (TRAC:75, HASOC:77).

random Mixup, and hence is more generalizable and effective across languages.

4.3 Impact of Sample Selection and Distance-Aware Mixing Ratio

Model	TTC	TREC-Coarse	AHS
TMix	91.30	97.52	70.19
+ M-Ratio	91.66	96.90	72.43
+ M-Threshold	92.02	97.10	73.31
DMix	**92.16**	**97.80**	**74.98**

Table 4: Ablation study over matrix $\mathbf{M}$ (F1 scores). M-Ratio denotes $\mathbf{M}$ is used only for performing mixup and sample selection is random. M-Threshold denotes that $\mathbf{M}$ is used to select samples based on the distance and mixup is performed with a random ratio.

We probe the individual impact of using matrix $\mathbf{M}$ for distance-based sample selection and using it for performing mixup in Table 4. We observe that both the applications of matrix $\mathbf{M}$ lead to improvements over TMix. Using matrix $\mathbf{M}$ for sample selection obtains larger improvements compared to using it as the ratio for performing mixup. This suggests that the selection of inputs for interpolation is more important than the mixing ratio when performing interpolative regularization.

4.4 Layer-wise Ablation

Mixup Layer Set	CoLA		HASOC		AHS	
	TMix	DMIX	TMix	DMIX	TMix	DMIX
{3,4}	79.45	79.70	76.86	77.46	69.37	65.66
{0, 1, 2}	80.18	94.08	76.39	77.99	69.28	71.98
{6, 7, 9}	82.91	94.63	77.12	79.44	70.11	73.45
{7, 9, 12}	85.30	95.63	77.44	80.19	70.19	74.32
{3, 4, 6, 7, 9, 12}	84.03	**95.94**	76.99	**80.27**	70.03	**74.98**

Table 5: Layer-wise ablation (F1 scores) when performing interpolative augmentations.

We compare the performance of DMIX and TMix for different sets of mixup layers in Table 5. TMix attains the best performance when the layer set $\{7, 9, 12\}$ is used since layers 6, 7, 9 and 12 contain the most amount of syntactic and semantic information (Chen et al., 2020a). Interestingly, DMIX achieves the best performance when the layer is sampled from the set $\{3, 4, 6, 7, 9, 12\}$. This suggests that the surface-level information contained in layers 3 and 4 (Jawahar et al., 2019) is effectively leveraged by the distance-aware matrix M, leading to further improvements over purely syntactic and semantic information in layers $\{6, 7, 9, 12\}$.

4.5 Effect of Varying Thresholds

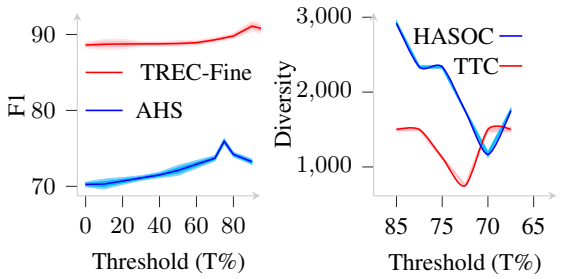

Figure 3: Change in performance in terms of F1 and Diversity with varying threshold T in % for DMIX.

We perform a study by varying the threshold τ for DMIX and present it in Figure 3. A decreasing τ denotes a larger distribution space for sampling instances for Mixup, and a T of 0% decomposes it to TMix or random Mixup. We observe an initial increase in the performance as we constrain the embedding space, suggesting the sampling of more diverse samples for interpolation. We observe a drop in performance when the constrain becomes very high, indicating that further expanding the sampling space does not lead to more diverse synthetic samples. This shows the existence of an optimum set of input samples for performing Mixup, and we conjecture it can be related to the sparsity in the embedding distribution of different languages.

5 Conclusion

We propose DMIX, a novel data augmentation technique that interpolates samples intelligently chosen based on their hyperbolic distance in the embedding space. DMIX achieves state-of-the-art results over existing data augmentation approaches on 8 standard and multilingual datasets in English, Arabic, Turkish, and Hindi languages, requiring 3 times less number of iterations than random mixup. DMIX being independent of the underlying model and modality, holds potential to be applied on text, speech, and vision downstream tasks.

6 Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) as a part of the Junior AI Scientists program under the reference 01-S20060. We thank the anonymous reviewers for their valuable inputs.

References

Nuha Albadi, Maram Kurdi, and Shivakant Mishra. 2018. Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 69–76. ACM.

Shiladitya Bhattacharya, Siddharth Singh, Ritesh Kumar, Akanksha Bansal, Akash Bhagat, Yogesh Dawer, Bornini Lahiri, and Atul Kr. Ojha. 2020. Developing a multilingual annotated corpus of misogyny and aggression. In *Proceedings of the Second Workshop on Trolling, Aggression and Cyberbullying*, pages 158–168, Marseille, France. European Language Resources Association (ELRA).

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020a. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020b. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018. Hyperbolic neural networks. In *Advances in Neural Information Processing Systems*.

Raphael Gontijo-Lopes, Sylvia J Smullin, Ekin D Cubuk, and Ethan Dyer. 2020. Affinity and diversity: Quantifying mechanisms of data augmentation. *arXiv preprint arXiv:2002.08973*.

Hongyu Guo, Yongyi Mao, and Richong Zhang. 2019. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does BERT learn about the structure of language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3651–3657, Florence, Italy. Association for Computational Linguistics.

Amit Jindal, Arijit Ghosh Chowdhury, Aniket Didolkar, Di Jin, Ramit Sawhney, and Rajiv Ratn Shah. 2020. Augmenting NLP models using latent feature interpolations. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6931–6936, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Deniz Kilinç, Akin Ozcift, Fatma Bozyiğit, Pelin Yildirim, Fatih Yucalar, and Emin Borandağ. 2017. Ttc-3600: A new benchmark dataset for turkish text categorization. *Journal of Information Science*, 43:174–185.

James P Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2021. Fnet: Mixing tokens with fourier transforms. *ArXiv*, abs/2105.03824.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Linlin Liu, Bosheng Ding, Lidong Bing, Shafiq Joty, Luo Si, and Chunyan Miao. 2021. MulDA: A multilingual data augmentation framework for low-resource cross-lingual NER. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5834–5846, Online. Association for Computational Linguistics.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th Forum for Information Retrieval Evaluation*, FIRE '19, page 14–17, New York, NY, USA. Association for Computing Machinery.

Colin Raffel, Noam M. Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *ArXiv*, abs/1910.10683.

Linqing Shi, Danyang Liu, Gongshen Liu, and Kui Meng. 2020. Aug-bert: An efficient data augmentation algorithm for text classification. In *Communications, Signal Processing, and Systems*, pages 2191–2198, Singapore. Springer Singapore.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Sinong Wang, Han Fang, Madian Khabsa, Hanzi Mao, and Hao Ma. 2021. Entailment as few-shot learner. *ArXiv*, abs/2104.14690.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2018. Neural network acceptability judgments. *arXiv preprint arXiv:1805.12471*.

Shusheng Xu, Xingxing Zhang, Yi Wu, and Furu Wei. 2021. Sequence level contrastive learning for text summarization. *ArXiv*, abs/2109.03481.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Soyoung Yoon, Gyuwan Kim, and Kyumin Park. 2021. SSMix: Saliency-based span mixup for text classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3225–3234, Online. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

A Extended Analysis

Model	CoLA	TREC-Coarse	TREC-Fine	SST-2
XLNet (2019)	70.20	94.58	87.49	**97.00**
T5-small (2020)	71.60	95.55	86.21	91.80
FNet (2021)	78.00	96.89	89.97	94.00
EFL (2021)	86.40	93.36	80.90	96.90
EMix (2020)	85.21	97.44	90.04	91.13
SSMix (2021)	86.76	97.60	90.24	92.95
DMix (Ours)	**95.94**	**97.80**	**91.14**	92.44

Table 6: Performance comparison with additional baselines and interpolative augmentation methods.

We compare the performance of DMIX on standard English and GLUE datasets with additional baselines and interpolative augmentation methods like EMix (Jindal et al., 2020) and SSMix (Yoon et al., 2021).

B Dataset Details

1. **TRAC**. (Bhattacharya et al., 2020) is a collection of posts, comments, and other content from popular social media, streaming and sharing platforms. For the purpose of our experiments, we perform the aggression classification task, for which, the data is labelled into 3 classes based on the level of aggression.

2. **TREC-Coarse**. (Li and Roth, 2002), The Text REtrieval Conference-Coarse is a question classification dataset consisting of 6 classes. The data is sourced from English questions by USC, TREC 8, TREC 9, TREC 10 and manually constructed questions.

3. **TREC-Fine**. (Li and Roth, 2002) contains the same set of questions as TREC-Coarse grouped into 47 fine-grained classes instead of 6.

4. **CoLA**. (Warstadt et al., 2018), abbreviation for the Corpus of Linguistic Acceptability is a part of GLUE (Wang et al., 2018) benchmark. It is a collection of English sentences from 23 linguistic publications that are annotated for their grammatical acceptability.

5. **SST-2**. (Socher et al., 2013) is a GLUE (Wang et al., 2018) benchmark dataset consisting of English sentences from movie reviews. Samples in the dataset are annotated for sentiment classification task.

6. **AHS**. (Albadi et al., 2018) is an Arabic hate speech classification dataset focusing mainly on Saudi Twittersphere. The data has been collected over a span of 6 months from March 2018 to August 2018 and has 3950 samples classified into 2 classes.

7. **TTC**. (Kilinç et al., 2017), Turkish Text Categorization dataset consists of 3600 Turkish documents (news/texts) classified into 6 classes. The data is obtained between the period from May 2015 to July 2015.

8. **HASOC**. (Mandl et al., 2019) consists of content sampled from social media platforms. We perform the binary Hate/Offensive content classification task on the Hindi dataset for the purpose of our experiments.

C Experimental Setup

We mention the optimal hyperparameter settings in Table 8.

Sentence	TMix	DMix-NT	DMix
Intellectuals and the so-called Secular are more illiterate Uneducated and illiterate	OAG	NAG	NAG
She must be sent to jail for anti national activities under NSA and PSA	NAG	CAG	CAG
Lion king fan hit like	OAG	CAG	NAG
kapil why are u listening to these ch*tsss ....give them shut up call...insane idiots	CAG	CAG	OAG
Great Job Mr Jahangir Sir I support you	NAG	CAG	NAG
Absolute fantastic movie please go and watch the movie first.	CAG	NAG	NAG

Table 7: Qualitative analysis of the performance obtained by TMix, DMIX-NT, and DMIX. The color intensity of each word corresponds to the token-level attention score. Green denotes correct prediction and red denotes incorrect prediction. (NAG: Non Aggressive, OAG: Overtly Aggressive, CAG: Covertly Aggressive).

Parameter	Value
Optimizer	BERTAdam
Learning Rate	2e-5
Batch Size	8
$\beta_1, \beta_2, \epsilon$	0.9, 0.999, 1e-6
# Epochs	5
Evaluation Metric	F1 Score
Base Model	BERT-base-uncased, BERT-base-multilingual-uncased
Classifier (over architecture)	Linear layer
Hardware	Nvidia P100

Table 8: Model and training setup for DMix.

D Comparison with Contrastive Learning

Contrastive learning involves training the underlying model to learn an embedding space in which similar sample pairs stay close to each other while dissimilar ones are far apart. Hence, their training objective directly involves training using this embedding vector of the input samples in the dataset. DMIX however chooses samples based on their spatial distribution in the embedding space, but does not have a training objective optimizing on their position in the embedding space. The training of DMIX is still supervised in nature and involves learning over the mixed label of the individual samples being used for interpolation.

E Qualitative Analysis

To further analyze DMIX, we perform a qualitative study by choosing examples from the dataset and compare the predictions made by TMix and DMIX-NT with DMIX. We analyze token-level attention assigned to the individual terms by BERT, where color intensity corresponds to the attention score. We present these results in Table 7.