HYPMIX: Hyperbolic Interpolative Data Augmentation

Ramit Sawhney†‡; Megh Thakkar†§; Shivam Agarwal‡; Di Jin*, Diyi Yang△, Lucie Flek†
†ShareChat
‡Conversational AI and Social Analytics (CAISA) Lab, University of Marburg
§BITS, Pilani
*Amazon Alexa AI
△Georgia Institute of Technology
ramitsawhney@sharechat.co, lucie.flek@uni-marburg.de

Abstract

Interpolation-based regularisation methods for data augmentation have proven to be effective for various tasks and modalities. These methods involve performing mathematical operations over the raw input samples or their latent states representations - vectors that often possess complex hierarchical geometries. However, these operations are performed in the Euclidean space, simplifying these representations, which may lead to distorted and noisy interpolations. We propose HypMix, a novel model-, data-, and modality-agnostic interpolative data augmentation technique operating in the hyperbolic space, which captures the complex geometry of input and hidden state hierarchies better than its contemporaries. We evaluate HypMix on benchmark and low resource datasets across speech, text, and vision modalities, showing that HypMix consistently outperforms state-of-the-art data augmentation techniques. In addition, we demonstrate the use of HypMix in semi-supervised settings. We further probe into the adversarial robustness and qualitative inferences we draw from HypMix that elucidate the efficacy of the Riemannian hyperbolic manifolds for interpolation-based data augmentation.

1 Introduction

Deep learning methods have improved the state-of-the-art in a wide range of tasks. Yet, when only limited training data is available, they are prone to overfitting (Zou and Gu, 2019). Numerous data augmentation techniques have been proposed, which involve performing operations such as cropping or rotation (Lecun et al., 1998), or paraphrasing (Kumar et al., 2019) individual examples. However, these methods are modality- or dataset-dependent and require domain expertise. Compared to such alteration-based methods, interpolation-based approaches such as Mixup (Zhang et al., 2018) have shown improved performance and generalizability across different modalities. Mixup generates virtual training samples from convex combinations of individual inputs and labels to expand the training distribution. Performing Mixup over the latent representations of inputs has led to further improvements, as the hidden states of deep neural networks carry more information than raw input samples, (Verma et al., 2019a; Chen et al., 2020a). However, most data augmentation methods can only utilize existing labeled data.

Semi-supervised learning methods, on the other hand, can leverage unlabeled data for training. Several semi-supervised methods use interpolation based regularization methods over unlabeled samples to predict soft labels, and combine them with existing labeled samples to increase the overall training data (Verma et al., 2019b; Chen et al., 2020b). Semi-supervised methods use consistency based regularization training (Miyato et al., 2019) which makes the model predictions robust to perturbations on unlabeled samples. However, current semi-supervised learning methods do not generalize across modalities or datasets.

Existing data-augmentation and semi-supervised learning methods operate in the Euclidean space, which is a simplified representative geometry. Representations across modalities inherently possess properties that the Euclidean space is incapable of modeling, and can be better expressed using the more general hyperbolic space (Ganea et al., 2018). The interference of sound waves is hyperbolic in nature, which generates hyperboloid waveforms
Natural language text exhibits hierarchical structure in a variety of respects and embeddings are more expressive when represented in the hyperbolic space (Dhingra et al., 2018). Data augmentation using Möbius operations over images has shown more diversification and generalization compared to Euclidean operations (Zhou et al., 2021). Performing interpolative operations over representations having complex geometry in the hyperbolic space can lead to more suitable representations for model training.

Building on prior research in limited data and data augmentation studies, and the hyperbolic characteristics of speech, text, and vision, we propose HYPMIX\textsuperscript{1}: a model, data, and modality agnostic interpolative regularization method operating in the hyperbolic space. We further extend HYPMIX to semi-supervised settings, which is especially effective in extremely low resource environments. We probe the effectiveness of HYPMIX through extensive experiments over three different tasks for supervised and semi-supervised settings on benchmark and low-resource datasets across speech, text, and vision in different languages with varying class label distribution. HYPMIX outperforms current state-of-the-art modality and task specific data augmentation methods across all the datasets for both supervised and semi-supervised conditions.

Our contributions can be summarized as:

- We propose HYPMIX, a novel model, data, and modality agnostic interpolative regularization based data augmentation method functioning in the hyperbolic space.

- We devise a novel Möbius Gyromidpoint Label Estimation (MGLE) method to predict soft labels for unlabeled data, and extend HYPMIX to a hyperbolic semi-supervised learning method.

- HYPMIX outperforms several strong baselines and Euclidean counterparts across speech, text, and vision across benchmark and low-resource datasets, including semi-supervised settings for Urdu and Arabic tasks.

- We further probe the effectiveness of HYPMIX in comparison to existing methods through layer-wise ablation studies and adversarial robustness.

\textsuperscript{1}Our code is available at: https://github.com/caisa-lab/hypmix-emnlp.

2 Background and Related Work

**Data Augmentation** enables use of limited training data, with approaches involving modifying the individual training instances, such as cropping (Simonyan and Zisserman, 2015) or paraphrasing (Wei and Zou, 2019; Kumar et al., 2019). Mixup techniques (Zhu et al., 2019) perform interpolation among input samples and have proven to perform better than modifying individual instances as it incorporates the prior knowledge that linear interpolations of feature vectors should lead to linear interpolations of the associated targets. Recent works (Jindal et al., 2020a; Verma et al., 2019a) perform Mixup operations over hidden state representation of input samples instead of the inputs, as high-level representations are often low-dimensional and carry more useful information of input samples as compared to raw inputs. Latent interpolation methods have not been generalized across modalities and operate in the simplified Euclidean space which is unable to capture the complex characteristics possessed by latent state representations.

**Semi-supervised Learning** methods leverage unlabeled data which is typically available in larger quantities (Clark et al., 2018). Consistency regularization methods for semi-supervised learning predict soft labels for unlabeled data and train models on different permutations of labeled and unlabeled data (Verma et al., 2019b; Chen et al., 2020a). Chen et al. (2020b) uses a label guessing strategy on different augmentations of unlabeled data and combines it with labeled data for training models. However, these methods perform label prediction for unlabeled data using Euclidean operations.

**Hyperbolic Learning** has proven to be effective in representing information where relations among data points possess hierarchical and tree-like nature (Aldecoa et al., 2015). Learning in the hyperbolic space has been applied to various natural language processing (Dhingra et al., 2018; Gulcehre et al., 2019; Tay et al., 2018), and computer vision tasks (Khrulkov et al., 2020; Peng et al., 2020) as well as graph (Chami et al., 2019), sequence (Tay et al., 2018), and financial (Sawhney et al., 2021) learning. However, the ability of the hyperbolic space to model complex representations while performing interpolative operations across modalities is unexplored.
3 Methodology: HYP MIX

We first formulate the task and introduce the hyperbolic space (Ganea et al., 2018) and Mixup (Zhang et al., 2018) (§3.1). Using the hyperbolic operations, we then introduce Mixup in the hyperbolic space (§3.2), and extend it to operate on the hidden state representations of neural networks. We call the resulting approach HYP MIX. An overview of the steps is presented in Figure 1. We formulate HYP MIX for both supervised (§3.3) and semi-supervised (§3.4) methods. We test HYP MIX on classification tasks across speech, text, and vision.

3.1 Preliminaries

Hyperbolic Space is a non-Euclidean geometry with constant negative curvature. We use the Poincaré ball model of the hyperbolic space (Ganea et al., 2018), defined as $B = \{ x \in \mathbb{R}^n : \| x \| < 1 \}$, where the manifold $B$ is endowed with the Riemannian metric $g^B_x = \lambda_x^2 g^E$, where the conformal factor $\lambda_x = \frac{1}{\sqrt{1 - \| x \|^2}}$ and $g^E = \text{diag}[1, \ldots, 1]$ is the Euclidean metric tensor. We denote the tangent space centered at point $x$ as $T_x B$. We use the Möbius gyrovector space to generalize standard mathematical operations to the hyperbolic space: Möbius Addition, $\oplus$, for a pair of points $x, y \in B$, $x \oplus y := \frac{(1 + 2(x, y) + \| y \|^2)x + (1 - \| x \|^2)y}{1 + 2(x, y) + \| x \|^2 \| y \|^2}$ (1)

where, $\langle , \rangle$ denotes the Euclidean inner product given by $\langle x, y \rangle = x_0y_0 + x_1y_1 + \ldots + x_{n-1}y_{n-1}$, and $\| \cdot \|$ denotes the norm given by $\| x \| = \sqrt{\langle x, x \rangle}$.

We define the exponential and logarithmic maps to project vectors between the Euclidean and hyperbolic space respectively.

Exponential Mapping$^2$ maps the tangent vector $v$ to the point $\exp_x(v)$ on the Poincaré ball,

$$\exp_x(v) := x \oplus \left( \tanh \left( \frac{\lambda_x \| v \|}{2} \right) \frac{v}{\| v \|} \right)$$ (2)

Logarithmic Mapping maps a point $y \in B$ to a point $\log_x(y)$ on the tangent space at $x$,

$$\log_x(y) := \frac{2}{\lambda_x} \text{arctanh} \left( \| x \| - \| x \oplus y \| \right) \frac{-x \oplus y}{\| x \| - \| x \oplus y \|}$$ (3)

For exponential and logarithmic mapping, we choose the tangent space center $x = 0$ and use $\exp_0(\cdot)$ and $\log_0(\cdot)$.

Möbius Scalar Multiplication $\odot$ multiplies matrix $x \in B$ with scalar $r \in \mathbb{R}$,

$$r \odot x = \tan \left( r \tan^{-1}(\| x \|) \right) \frac{x}{\| x \|}$$ (4)

Möbius Gyromidpoint $M_g$ calculates the hyperbolic weighted addition for gyrovectors $\{ x_1, \ldots, x_n \}$ and weights $\{ \alpha_1, \ldots, \alpha_n \}$,

$$M_g(x_1, x_2, \ldots, x_n, \alpha_1, \alpha_2, \ldots, \alpha_n) = (x_1 \odot \alpha_1) \oplus (x_2 \odot \alpha_2) \ldots \oplus (x_n \odot \alpha_n)$$ (5)

Mixup (Zhang et al., 2018) involves training a neural network on convex combinations of a pair of instances and their labels. For two labeled data points $(x_i, y_i)$ and $(x_j, y_j)$, mixup uses linear interpolation with mixing ratio $r$ to generate the synthetic sample $x'$ and corresponding mixed label $y'$,

$$x' = \text{mix}(x_i, x_j) = r \cdot x_i + (1 - r) \cdot x_j$$

$$y' = \text{mix}(y_i, y_j) = r \cdot y_i + (1 - r) \cdot y_j$$ (6)

By leveraging the hyperbolic operations and Mixup, we define Mixup in the hyperbolic space.

$^2$We use the implementation by geoopt: https://geoopt.readthedocs.io/
3.2 Formulating Mixup in Hyperbolic Space

For inputs possessing complex geometrical properties, performing mathematical operations in the Euclidean spaces often lead to vectorial distortions which can be stabilized by using the hyperbolic space (Ganea et al., 2018). To minimize these distortions, we formulate MixH, Mixup in the hyperbolic space by leveraging hyperbolic operations as building blocks. First, we replace Euclidean arithmetic addition (+) and scalar product (·) with their Möbius counterparts: addition (⊕) and scalar multiplication (⊙) respectively. We then transform inputs to the hyperbolic space using the exponential mapping \( \exp_0(·) \), perform Mixup to generate convex combinations of pairs of inputs \( x_i, x_j \), and map them back to the Euclidean space using the logarithmic mapping \( \log_0(·) \). Formally,

\[
\text{MixH}(x_i, x_j) = \log_0(r \circ \exp_0(x_i) \oplus (1 - r) \circ \exp_0(x_j)) \tag{7}
\]

We now extend MixH as a generalizable interpolative regularizer over hidden state representations across neural network layers.

3.3 HYPMIX: Interpolative MixH

Previous works (Chen et al., 2020b; Jindal et al., 2020b) applying interpolation based regularization in the latent space of neural networks operate in the Euclidean space, which cannot capture the complex geometries of hidden state vectors (Tifrea et al., 2019). To better model the fine-grained information present in latent representations using the hyperbolic space, we extend MixH to the hidden representation space. Let \( f_{\theta,n}(·) \) denote any general base model with parameters \( \theta \) having \( N \) layers. \( f_{\theta,n}(·) \) denotes the \( n \)-th layer of the model and \( h_n \) is the hidden space vector at layer \( n \) for \( n \in [1, N] \) and \( h_0 \) denotes the input vector.

We introduce HYPMIX as hyperbolic interpolation at a layer \( k \sim [1, N] \), for which we first calculate the latent representations separately for the inputs for layers before the \( k \)-th layer. For input samples \( x_i, x_j \), we let \( h_n^i, h_n^j \) denote their respective hidden state representations at layer \( n \) of \( f_{\theta}(·) \),

\[
\begin{align*}
  h_n^i &= f_{\theta,n}(h_{n-1}^i), \quad n \in [1, k] \\
  h_n^j &= f_{\theta,n}(h_{n-1}^j), \quad n \in [1, k] \tag{8}
\end{align*}
\]

We then apply MixH over the individual hidden state representations \( h_k^i, h_k^j \) from layer \( k \) as:

\[
\begin{align*}
  h_k &= \text{MixH}(h_k^i, h_k^j) \\
  &= \log_0(r \circ \exp_0(h_k^i) \oplus (1 - r) \circ \exp_0(h_k^j)) \tag{9}
\end{align*}
\]

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

\[
h_n = f_{\theta,n}(h_{n-1}); \quad n \in [k + 1, N] \tag{10}
\]

We define HYPMIX \( (f_{\theta}(·), r, k) \) for a layer \( k \) and mixing ratio \( r \) to obtain the final hidden layer representation \( h_N \) as,

\[
h_N = \text{HYPMIX}(x_i, x_j, f_{\theta}(·), r, k) \tag{11}
\]

**Supervised Network Optimization** For classification, we apply a perceptron \( g_\phi(·) \) with parameters \( \phi \) to calculate the class logits from the final hidden state output \( h_N \). We optimize the model using KL-divergence loss (KL) to bring the model output distribution closer to the mixed label distribution. We minimize the loss \( L \) between the mixed label \( y' \) and logits obtained from HYPMIX,

\[
L = \text{KL} (\text{mix}(y_i, y_j)) || g_\phi(\text{HYPMIX}(x_i, x_j, f_{\theta}(·), r, k))) \tag{12}
\]

3.4 Hyperbolic Semi-supervised Learning

Semi-supervised training methods leverage unlabeled data to improve the training for limited or low resource settings (Verma et al., 2019b). We extend HYPMIX to effectively utilize \( p \) labeled data points, \( X_l = \{x_1^l, x_2^l, \ldots, x_p^l\} \) and \( q \) unlabeled data points, \( \{X_u, y_u\} \), with a separate hyperbolic mixup process for labeled data points during training.

- **Formulation**
  - For labeled data: \( h_n^l = f_{\theta,n}(h_{n-1}^l) \)
  - For unlabeled data: \( h_n^u = f_{\theta,n}(h_{n-1}^u) \)
  - Mixup process: \( h_k = \text{MixH}(h_k^l, h_k^u) \)

- **Objective**
  - Minimize KL divergence between mixed labels and predictions:
  
  \[
  L = \text{KL} (\text{mix}(y_i, y_j)) || g_\phi(\text{HYPMIX}(x_i, x_j, f_{\theta}(·), r, k))) \tag{12}
  \]

- **Implementation**
  - Adapt HYPMIX to incorporate semi-supervised learning objectives, adjusting the loss function to include unlabeled data constraints.
\(X_u = \{x_1^u, x_2^u, \ldots, x_q^u\}\) using a semi-supervised training strategy in the hyperbolic space (Figure 2).

We first use existing data augmentation techniques across different modalities to increase the unlabeled training data \(X_u\). For an unlabeled sample \(x_i^u\), we generate \(Z\) augmented samples using different augmentation methods such as backtranslation (Edunov et al., 2018) and combine them to generate unlabeled augmented sets, \(X_a = \{X_{a,1}, X_{a,2}, \ldots, X_{a,Z}\} | X_{a,z} = \{x_1^{a,z}, x_2^{a,z}, \ldots, x_q^{a,z}\}, z \in [1, Z]\).

Möbius Gyromidpoint Label Estimation (MGLE) predicts soft logits for unlabeled and augmented data in the hyperbolic space, allowing us to combine the unlabeled data with limited training data using HYPMIX for training. For an unlabeled sample \(x_i^u\) and corresponding augmented samples \(\{x_1^{a,1}, x_2^{a,2}, \ldots, x_q^{a,Z}\}\), we compute the Möbius Gyromidpoint \(M_y\) of the hyperbolic mapped outputs \(\{g_\phi(f_\theta(x_1^u)), g_\phi(f_\theta(x_2^u)), \ldots, g_\phi(f_\theta(x_q^u))\}\) with fixed weights \(\{w_0, w_1, \ldots, w_Z\} | w_i \in (0.5, 1.5)\), where weight \(w_0\) is applied to the original unlabeled sample. The weights control the contribution of different augmentation techniques based on their augmentation quality. We map the predicted output logits to the Euclidean space using \(\log_\theta(\cdot)\) to predict the soft logits \(y_i^u\),

\[
y_i^u = \log_\theta(M_y(\exp_0(g_\phi(f_\theta(x_1^u))), \exp_0(g_\phi(f_\theta(x_2^u))), \ldots, \exp_0(g_\phi(f_\theta(x_q^u))))
\]

(13)

We sharpen the output \(y_i^u\) with a hyperparameter temperature \(T\), to prevent it from being too uniform if the model predictions are random,

\[
y_i^u = \frac{(y_i^u)^\frac{T}{\Gamma}}{||y_i^u||_1}
\]

(14)

where \(\cdot\) is the \(l_1\)-norm of the vector.

**Semi-supervised Network Optimization**

For optimizing the model in semi-supervised settings, we use the training set \(X = X_T \cup X_u \cup X_a\) with labels \(Y = Y_T \cup Y_u \cup Y_a\), where \(Y_u\) is used for both unlabeled and augmented inputs, i.e. \(Y_a = Y_u\). We then uniformly sample two elements, \(x_i, x_j \sim X\) and the corresponding labels \(y_i, y_j \sim Y\), and apply HYPMIX(\(x_i, x_j\)). We optimize the model using KL-divergence loss \(L\) over the model outputs and the mixed labels \(\text{mix}(y_i, y_j)\).

\[
L = \text{KL}(|\text{mix}(y_i, y_j)| || g_\phi(\text{HYPMIX}(x_i, x_j)))
\]

(15)

---

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Class Labels</th>
<th># Classes</th>
<th># Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>ESC-10 (2015)</td>
<td>Sound Source</td>
<td>10</td>
<td>400</td>
</tr>
<tr>
<td>USSK (2017)</td>
<td>Sound Source</td>
<td>10</td>
<td>8,732</td>
</tr>
<tr>
<td>Urdu SER (2018)</td>
<td>Emotion</td>
<td>4</td>
<td>400</td>
</tr>
<tr>
<td>DB Pedia (2012)</td>
<td>Wiki Topic</td>
<td>14</td>
<td>630,000</td>
</tr>
<tr>
<td>Arabic HS (2018)</td>
<td>Hate Speech</td>
<td>2</td>
<td>3,950</td>
</tr>
<tr>
<td>Vi-CIFAR-10 (2009)</td>
<td>Object</td>
<td>10</td>
<td>60,000</td>
</tr>
<tr>
<td>CIFAR-100 (2009)</td>
<td>Object</td>
<td>100</td>
<td>60,000</td>
</tr>
</tbody>
</table>

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

---

4 Experimental Setup

4.1 Datasets and Preprocessing

We consider benchmark and low-resource datasets across speech, text, and vision spanning a varying number of classes, languages, and class imbalances for a comprehensive evaluation of HYPMIX. We choose these datasets based on existing works across different task settings and baselines for a fair comparison with HYPMIX. We also choose datasets with comparatively lower language resources, different structures, and language roots, leading to a more diverse evaluation of HYPMIX. We summarize dataset statistics in Table 1. We follow the same preprocessing across all datasets as done by previous works (Jindal et al., 2020b), (Chen et al., 2020b), (Verma et al., 2019a).

4.2 Task Setup

We evaluate HYPMIX on three different settings for an extensive analysis: supervised training with limited training data, semi-supervised training with low resource data, and a fully supervised setup with complete training data.

**Speech**

Following previous works, we use EnvNet-v2 with strong augmentation (Tokozume et al., 2018) as our base architecture \(f_\theta(\cdot)\) followed by a fully connected layer \(g_\phi(\cdot)\). We modify MIXH to account for the auditory perception and amplitude of speech signals (Tokozume et al., 2018). We use Fourier and Inverse Fourier Transform to generate augmented samples. We compare HYPMIX with the current state-of-the-art method Speechmix (Jindal et al., 2020b) across multiple settings.

**Text**

Following Chen et al. (2020b), we use BERT-base (Devlin et al., 2019) as the backbone architecture \(f_\theta(\cdot)\) for English datasets and BERT-base-arabic (Safaya et al., 2020) for the Arabic dataset. We use a two layer MLP with hidden size 128 as the classifier \(g_\phi(\cdot)\) and generate augmented data using back-translation (Edunov et al., 2018).
We compare HypMix with TMix and MixText (Chen et al., 2020b) for supervised and semi-supervised training respectively.

### Vision
Following Verma et al. (2019a), we use PreActResNet18 (He et al., 2016) as the backbone architecture \((f_\theta(\cdot))\) and a linear layer as the classifier \((g_\phi(\cdot))\). We compare HypMix with manifold mixup (Verma et al., 2019a) for different settings.

#### 4.3 Training Setup
**Speech** We use Nesterov’s accelerated gradient (Sutskever et al., 2013) using momentum of 0.9, weight decay of \(5e^{-4}\), learning rate of 0.01 and mini-batch size of 64 for 2000 epochs. For ESC-10, we train the model on 5 folds, and for UrbanSound8k, we train the model on 10 folds to report the average error rate. We randomly sample the mixing ratio from a uniform distribution, \(r \sim U(0, 1)\). For semi-supervised training, we use 50 unlabeled samples from each class.

**Text** We use AdamW (Loshchilov and Hutter, 2019) optimizer with a learning rate \(1e^{-5}\) for the BERT encoder and \(1e^{-3}\) for the MLP. We follow Chen et al. (2020b) to sample the mixing ratio \(r\) from a beta distribution based on the number of labeled samples. For semi-supervised setting, we use 1000 unlabeled samples from each class.

**Vision** We use Nesterov’s accelerated gradient (Sutskever et al., 2013) using momentum of 0.9 and learning rate of 0.1, batch size of 100 to train for 2000 epochs. Following Verma et al. (2019a), we sample the mixing ratio \(r \sim \text{Beta}(2, 2)\), where Beta denotes the Beta distribution.

### 5 Results and Analysis
#### 5.1 Supervised Training with Limited Data
We compare HypMix in a limited training data setup with baseline methods in Table 2. We observe that Euclidean mixup techniques (EUCMix) improve the performance over base models, indicating the importance of using the latent representation space of neural network architectures to perform interpolative regularization (Verma et al., 2019a). HypMix further improves performance \((p < 0.01)\) over Euclidean methods across modalities, validating that the hyperbolic space is able to better capture the complex geometry of latent representations for different inputs when performing interpolative operations.

HypMix shows maximum improvement when applied on extremely low training data, with samples in order of \(n = 10\). This is in line with works (Zhou et al., 2021) which suggest that the variation generated by Möbius operations is very high as compared to Euclidean operations, leading to much more diverse samples from a small training set. This paves a path for better utilization of low resource datasets for downstream tasks across different modalities by leveraging the hyperbolic space. For all modalities, the relative improvement over the baseline architecture reduces with increasing number of labeled samples per class \((n)\). This is in line with works (Verma et al., 2019b; Chen et al., 2020b) observing similar trends, suggesting that with an increase in number of labeled samples, the overall diversity of interpolative representations saturates, leading to lower relative improvements.

Across modalities, we observe maximum improvement when HypMix is applied on speech datasets, since speech waves inherently possess hyperbolic nature (Khan and Panigrahi, 2016), and their interpolative augmentation closely resembles hyperbolic wave interference (Chaturvedi et al., 1998). Improvements due to HypMix on text datasets lies with works stating that text inherently displays tree-like hierarchical characteristics and can be better represented using Riemannian ge-
We observe greater improvements when semi-supervised training is applied in the hyperbolic space, indicating the importance of using hyperbolic semi-supervised methods, respectively, under Wilcoxon’s signed-rank test.

Table 3: Performance comparison of HypMix in semi-supervised and semi-supervised methods, respectively, under terms of % Error rate. $n$ is the number of labeled training samples per class. Improvements are shown with blue (↓) and poorer performance with red (↑). Bold shows the best result. *, ♦ show significant ($p < 0.01$) improvement over Euclidean supervised semi-supervised methods, respectively, under Wilcoxon’s signed-rank test.

Table 4: Layer-wise ablation (% Error rate) on ESC-10.

Table 3: Performance comparison of HypMix in semi-supervised settings with state-of-the-art methods in terms of % Error rate. $n$ is the number of labeled training samples per class. Improvements are shown with blue (↓) and poorer performance with red (↑). Bold shows the best result. *, ♦ show significant ($p < 0.01$) improvement over Euclidean supervised semi-supervised methods, respectively, under Wilcoxon’s signed-rank test.

Across modalities, speech datasets that are augmented with simpler methods such as mathematical transforms show larger improvements as compared to text datasets that are augmented with more complicated methods like backtranslation. We attribute this difference to the proximity of augmented unlabeled samples to the original unlabeled data distribution, suggesting that better augmentation methods and controlling the weights for Möbius Gyroid midpoint Label Estimation (MGLE, §3.4) based on the augmentation quality is an important factor for the performance of semi-supervised methods.

5.2 Semi-Supervised Results: Low-Resource

We probe the effect of using hyperbolic semi-supervised learning (§3.4) for low resource datasets using HypMix in Table 3. Using semi-supervised learning shows significant improvements over their supervised counterparts trained with limited data for both Euclidean and hyperbolic (HypMix) representations, indicating the importance of using unlabeled and augmented data as additional training data. For both Euclidean and hyperbolic methods, we see larger improvements with increasing the number of labeled samples $n$, due to the increased number of permutations of labeled-labeled, unlabeled-labeled, and unlabeled-unlabeled samples encountered during training. We observe greater improvements when semi-supervised training is applied in the hyperbolic space with HypMix (Figure 3) for both speech and text as compared to EuCMIX, indicating that the hyperbolic space is able to generate less noisy, yet more diverse samples by effectively modeling the complex latent space representations.

Table 4: Layer-wise ablation (% Error rate) on ESC-10.
Table 5: Layer-wise ablation (% Error rate) on AG News with \( n = 10 \) labeled samples per class.

<table>
<thead>
<tr>
<th>Layer Set</th>
<th>EUC Mix(( \downarrow ))</th>
<th>HYPMix(( \downarrow ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \phi )</td>
<td>30.5</td>
<td>30.5</td>
</tr>
<tr>
<td>{7, 9, 12}</td>
<td>25.9</td>
<td>22.7</td>
</tr>
<tr>
<td>{6, 7, 9, 12}</td>
<td>27.8</td>
<td>21.8</td>
</tr>
<tr>
<td>{6, 7, 9}</td>
<td>28.1</td>
<td>24.9</td>
</tr>
</tbody>
</table>

Table 6: Performance comparison in terms of % Error rate of HypMix with baselines in supervised settings with full training data. **Bold** shows the best result. \( \circ \) show significant (\( p < 0.01 \)) improvement over previous state-of-the-art method under Wilcoxon’s signed-rank test.

<table>
<thead>
<tr>
<th>Model</th>
<th>ESC-10 (( \downarrow ))</th>
<th>US8K (( \downarrow ))</th>
<th>UrduSER (( \downarrow ))</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnvNet-v2 (2018)</td>
<td>10.9</td>
<td>24.9</td>
<td>10.1</td>
</tr>
<tr>
<td>+ BC Learning (2018)</td>
<td>8.6</td>
<td>21.7</td>
<td>8.7</td>
</tr>
<tr>
<td>+ SpeechMix (2020b)</td>
<td>7.1</td>
<td>20.8</td>
<td>6.2</td>
</tr>
<tr>
<td>+ HYPMix-Input</td>
<td>6.5( ^\circ )</td>
<td>20.9</td>
<td>5.0( ^\circ )</td>
</tr>
<tr>
<td>+ HYPMIX</td>
<td>6.3( ^\circ )</td>
<td>20.4( ^\circ )</td>
<td>2.5( ^\circ )</td>
</tr>
</tbody>
</table>

5.4 Supervised HYPMIX with Complete Data

5.5 Robustness to Adversarial Attacks

Table 7: Classification errors on adversarial examples generated using white box FGSM and I-FGSM attacks.

<table>
<thead>
<tr>
<th>Method</th>
<th>ESC-10</th>
<th>US8K</th>
<th>UrduSER</th>
</tr>
</thead>
<tbody>
<tr>
<td>FGSM</td>
<td>87.92</td>
<td>97.50</td>
<td>68.74</td>
</tr>
<tr>
<td>I-FGSM</td>
<td>82.50</td>
<td>97.47</td>
<td>77.50</td>
</tr>
<tr>
<td>HYPMIX</td>
<td>82.75</td>
<td>97.47</td>
<td>66.24</td>
</tr>
</tbody>
</table>

5.6 Cost of Hyperbolic Operations

HYPMIX requires additional hyperbolic transformations such as exponential and logarithmic mappings and tangential and hyperbolic tangential operations on-top of EUC Mix. However, on a GPU, they can be carried out in parallel. Hence, HYPMIX requires longer time only by a constant factor.
compared to EUCMIX, with the individual operations having similar time complexity, which is of the order of the input dimensions of the latent representations of the samples to be mixed. We compare the per iteration time taken by HYPMIX with EUCMIX in Table 8.

<table>
<thead>
<tr>
<th>Method</th>
<th>AGNews</th>
<th>ESC-10</th>
</tr>
</thead>
<tbody>
<tr>
<td>EUCMIX</td>
<td>0.826</td>
<td>0.862</td>
</tr>
<tr>
<td>HYPMIX</td>
<td>0.869</td>
<td>0.893</td>
</tr>
</tbody>
</table>

Table 8: Time (in seconds) per iteration for EUCMIX and HYPMIX.

6 Conclusion and Future Work

Drawing inspiration from works showing that speech, text, and vision data inherently possess hyperbolic characteristics and can be better represented in the hyperbolic space, we propose HYPMIX, a model, data, and modality agnostic interpolative regularization method operating in the hyperbolic space. We devise a Möbius Gyro-midpoint Label Estimation (MGLE) technique to predict labels for unlabeled training data and combine it with HYPMIX to formulate a hyperbolic semi-supervised learning method. HYPMIX outperforms existing methods for benchmark and low resource datasets across speech, text, and vision in supervised and semi-supervised settings with complete and limited training data. HYPMIX is also more robust to white-box adversarial attacks compared to Euclidean methods. HYPMIX being model, data, and modality agnostic can be extended to downstream tasks across modalities and interpolative augmentation for data such as sequences and graphs. As future work, we plan to evaluate HYPMIX on larger datasets and a variety of tasks such as the GLUE and SuperGLUE benchmarks, and tasks comprising multimodal settings.

7 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. Ramit Sawhney is supported by ShareChat. We thank the anonymous reviewers for their valuable inputs.

References


