AdaPT: A Set of Guidelines for Hyperbolic Multimodal Multilingual NLP

Ramit Sawhney[1,2]*, **Shrey Pandit**[3]*, **Vishwa Shah**[4]*, **Megh Thakkar**[5,6]*, **Shafiq Joty**[7]
[1]MBZUAI, [2] Georgia Institute of Technology, [3] UT Austin
[4]Carnegie Mellon University , [5]Mila – Quebec AI Institute,
[6]Université de Montréal , [7]Salesforce Research,
ramit.sawhney@mbzuai.ac.ae, shreypandit@utexas.edu
vishwavs@andrew.cmu.edu

Abstract

The Euclidean space is the familiar space for training neural models and performing arithmetic operations. However, many data types inherently possess complex geometries, and model training methods involve operating over their latent representations, which cannot be effectively captured in the Euclidean space. The hyperbolic space provides a more generalized representative geometry to model the hierarchical complexities of the tree-like structure of natural language. We propose AdaPT a set of guidelines for initialization, parametrization, and training of neural networks, which adapts to the dataset and can be used with different manifolds. AdaPT can be generalized over *any* existing neural network training methodology and leads to more stable training without a substantial increase in training time. We apply AdaPT guidelines over two state-of-the-art deep learning approaches and empirically demonstrate its effectiveness through experiments on three tasks over 12 languages across speech and text. Through extensive qualitative analysis, we put forward the applicability of AdaPT as a set of guidelines optimally utilizing the manifold geometry, which can be extended to various downstream tasks across languages and modalities.

1 Introduction

Using the Euclidean geometric space for representing latent embeddings, performing mathematical operations, and training neural models is common and has proved to be effective across various tasks and modalities (Bahdanau et al., 2015; He et al., 2015; Bordes et al., 2013). This is mainly because the Euclidean space is more convenient to use, and it is a natural generalization of the visual three-dimensional space. However, studies have shown that complex data types such as graphs and text exhibit a non-Euclidean and complex nature, in which case the standard Euclidean space may not be the most suitable geometric representation space (Bronstein et al., 2017). This has led to works defining neural models in the hyperbolic space using Möbius operations of the Riemannian geometry (Ganea et al., 2018), outperforming standard Euclidean methods across a variety of domains (Nickel and Kiela, 2017; Chami et al., 2019; Shimizu et al., 2021). The hyperbolic space has proven significantly effective for textual entailment tasks (Ganea et al., 2018), as well as for interpolative augmentation for text and speech domains (Sawhney et al., 2021). These approaches consider a *fixed* radius of curvature for the Poincaré ball model used as the hyperbolic representation during the course of training the models. They also use a default radius of curvature across the datasets and do not consider the specific extent of hyperbolic nature possessed by the dataset.

The δ-hyperbolicity of a space is a measure of its tree-likeliness, indicating the extent of hierarchical or hyperbolic nature this space possesses (Gromov, 1987). Tifrea et al. (2019) incorporate Gromov's calculation of δ-hyperbolicity for a space and define the δ-hyperbolicity for a dataset. The δ-hyperbolicity of a dataset can be used to estimate the optimum radius of the Poincaré disk in the hyperbolic space to represent the embedded dataset (Khrulkov et al., 2020). This can lead to a more suitable Riemannian manifold representation that can model complex geometries and latent representations of the dataset for performing mathematical operations and effectively training models.

*equal contribution

Findings of the Association for Computational Linguistics: NAACL 2024, pages 1757–1771
June 16-21, 2024 ©2024 Association for Computational Linguistics

However, as the dataset level δ-hyperbolicity is determined using the latent embeddings given by the underlying encoder, the optimal radius of curvature changes as the weights of the base model are updated during the course of training. Therefore, we hypothesize that a parameterized radius of curvature which is jointly optimized with the neural network training can effectively represent these embeddings at all steps of the training.

We propose **Ada**ptive **P**oincaré **T**ransfer (ADAPT), a set of guidelines that is based on initialization, parameterization and training of neural network, independent of model, dataset and modality, developed using standard Möbius operations. ADAPT can be generalized over *any* existing neural model to equip it with the capabilities of the hyperbolic space in representing complex geometries, both at the input and the latent representation level. ADAPT is optimized for a dataset, as it operates in a Riemannian space with a Poincaré disk having a dataset-specific radius, and hence, it is the maximally suitable representation geometry.

This radius of curvature is jointly optimized with the neural network training, enabling the model to *'adapt'* to the dynamic latent representations of the input samples. To show the generalizability of ADAPT, we apply it over two existing state-of-the-art deep learning approaches, Variational Information Bottleneck (Mahabadi et al., 2021), which uses the information bottleneck principle on the latent representation, and SSMix (Yoon et al., 2021), saliency-aware interpolation.

Through extensive experiments on datasets in 12 languages on sentence classification, natural language inference, named-entity recognition, and speech classification tasks, we present the improved performance of the existing methods when the proposed set of guidelines in ADAPT are followed, without any considerable increase in training time and resource requirements. By performing comprehensive qualitative experiments, we further analyze the effect of using ADAPT, and put forward its applicability for numerous multilingual language processing tasks leveraging the hyperbolic space. Our contributions are:

- We propose ADAPT, a generalized model, data, task, and modality agnostic set of guidelines that enables *any* existing deep learning methods to adapt to the hyperbolic space.
- We derive dataset-specific hyperbolicities for a general dataset and encoder, and use it to parameterize the Poincaré radius of curvature.
- We apply the guidelines of ADAPT on two existing state-of-the-art neural network training methods. Through extensive experiments on benchmark datasets in 12 languages across three different tasks for text and speech using latent and input-level representations, we obtain significant improvements over existing baseline methods.
- We further provide an in-depth analysis of ADAPT through qualitative experiments, putting forward its applicability for downstream tasks, datasets, and modalities.

2 Related Work

Hyperbolic Learning has been an effective way of representing information when the data possess hierarchical tree-like information (Aldecoa et al., 2015). Learning in hyperbolic space has already been applied in natural language processing tasks (Dhingra et al., 2018; Gulcehre et al., 2019; Nickel and Kiela, 2017), computer vision (Khrulkov et al., 2020; Peng et al., 2020), graph learning (Chami et al., 2019), sequence learning (Tay et al., 2018). (Chami et al., 2019) shows hyperbolic structure preserves the hierarchical structure and leads to improved performance when compared to euclidean analog even in a low dimensional embeddings. Tifrea et al. (2019) propose the dataset level δ-hyperbolicity metric to empirically measure the tree-likeliness of the dataset. Khrulkov et al. (2020) estimate the radius of curvature of the Poincaré disk using the corresponding δ-hyperbolicity. These works, however, do not incorporate the dataset-specific hyperbolicity in training the underlying neural networks and use a constant curvature throughout the training process.

Regularization and Data Augmentation techniques are used for improving model generalization in the absence of required training data and avoiding model overfitting. Variational Information Bottleneck (Mahabadi et al., 2021) extends the information bottleneck principle to a neural training objective and is effective in training models in low resource settings suppressing irrelevant features and preventing overfitting. Mixup (Zhang et al., 2018b) techniques perform convex combinations over raw inputs or their latent representations (Chen et al., 2020; Verma et al., 2019) to generate synthetic training data. Saliency-aware interpolative regularization approaches (Yoon et al., 2021; Kim et al., 2020) have been introduced, which show performance improvement over randomized mixup methods. These methods function in the simplified Euclidean space, which is unable to capture the complex characteristics possessed by the input samples or their latent representations.

Multilingual NLP is gaining widespread attention, but only a very small subset of languages are well-represented in progressing technologies and applications (Joshi et al., 2020). Techniques successful in the high resource scenario may not be effective for low resource languages that are of a different language family or very distinctive in linguistic and typological terms (Feng et al., 2021). A language agnostic set of guidelines can prove effective for wider research in multilingual NLP.

3 AdaPT Formulations

We first formulate AdaPT using several model, modality, task, and dataset agnostic operations which we later use to effectively leverage the hyperbolic space over existing state-of-the-art methods (§4). To give an overview of how initialization and parameterization work in AdaPT: (i) We first obtain the hyperbolicity i.e. the hierarchical tree-likeness of the dataset (§3.2)(ii) This helps us obtain the Poincaré ball radius of curvature for projection in the hyperbolic space to capture dataset's structure (§3.3)(iii) Finally, we propose trainable curvature to adapt to the dynamic nature of the encodings during training (§3.4). We discuss the hyperbolic mathematical operations needed for AdaPT in section 3.1.

3.1 Hyperbolic Arithmetic Operations

In this section we describe the preliminaries of Hyperbolic geometry that are helpful in understanding the formulations. Hyperbolic space is a non-Euclidean geometry with a constant negative curvature (Ganea et al., 2018). To effectively leverage the hyperbolic representation space, we first describe the hyperbolic variants of basic arithmetic operations. Following Chami et al. (2019), we use the Poincaré ball model of the hyperbolic space to perform mathematical operations[1], where the manifold is defined as $\mathbb{D}^n_\kappa = \{x \in \mathbb{R}^n : \kappa||x||^2 < 1\}$. This manifold centred at 0, has the conformal factor $\lambda^\kappa_x = \frac{2}{1-\kappa\|x\|^2}$, where κ is the radius of curvature of the Poincaré ball.

Möbius Addition, $\oplus_\kappa$ for a pair of points x, y,

$$x \oplus_\kappa y = \frac{(1 + 2\kappa\langle x, y\rangle + \kappa\|y\|^2)x + (1 - \kappa\|x\|^2)y}{1 + 2\kappa\langle x, y\rangle + \kappa^2\|x\|^2\|y\|^2} \quad (1)$$

where, $\langle ., . \rangle$ denotes the Euclidean inner product and $|| \cdot ||$ denotes the Euclidean norm.

We project vectors between Euclidean and hyperbolic space using exponential & logarithmic maps.

Exponential Mapping maps the tangent vector u to the point $\exp^\kappa_x(u)$ on the Poincaré ball,

$$\exp^c_x(u) = x \oplus_c \left(\tanh\left(\sqrt{c}\frac{\lambda^c_x\|u\|}{2}\right)\frac{u}{\sqrt{c}\|u\|}\right) \quad (2)$$

Logarithmic Mapping maps a point y to a point $\log^\kappa_{\mathrm{x}}(\mathrm{y})$ on the tangent space at x,

$$\log^\kappa_x(y) = \frac{2}{\sqrt{\kappa}\lambda^\kappa_x}\tanh^{-1}(\sqrt{\kappa}\| - x \oplus_\kappa y\|)\frac{-x \oplus_\kappa y}{\| - x \oplus_\kappa y\|} \quad (3)$$

For exponential and logarithmic mapping, we choose the tangent space center $x = 0$ and use $\exp^\kappa_0(\cdot)$ and $\log^\kappa_0(\cdot)$.

Möbius Scalar Multiplication $\odot_\kappa$ multiplies $x \in \mathbb{D}^n$ with scalar $r \in \mathbb{R}$,

$$r \odot_\kappa x = \frac{1}{\sqrt{\kappa}}\tanh\left(r\tanh^{-1}(\sqrt{\kappa}\|x\|)\right)\frac{x}{\|x\|} \quad (4)$$

Weighted Möbius gyromidpoint M_κ of a set of points $x_1, .., x_n$ according to weights $\alpha_1, .., \alpha_n$ calculates the hyperbolic weighted pooling,

$$M_\kappa(x_1, ., x_n, \alpha_1, ., \alpha_n) = \frac{1}{2}\odot_\kappa\left(\sum_{i=1}^{n}\frac{\alpha_i\lambda^\kappa_{x_i}}{\sum_{j=1}^{n}\alpha_j(\lambda^\kappa_{x_j} - 1)}x_i\right) \quad (5)$$

Hyperbolic Linear Layer $(HL(\cdot, \cdot))$ performs Möbius matrix vector multiplication of input x with weight matrix $W : \mathbb{R}^n \to \mathbb{R}^m$,

$$HL(x, W) = \frac{1}{\sqrt{\kappa}}\tanh\left(\frac{\|Wx\|}{\|x\|}\tanh^{-1}(\sqrt{\kappa}\|x\|)\right)\frac{Wx}{\|Wx\|} \quad (6)$$

3.2 Calculating the Dataset Hyperbolicity $\mathcal{H}$

A space is $\mathcal{H}$-hyperbolic if there exists a value $\mathcal{H}$ with the property that every point on the edge of a geodesic triangle lies within $\mathcal{H}$ of another edge. Following Khrulkov et al. (2020), we utilize the distances of the encoded representations of samples to calculate the extent of the hyperbolic nature of the datasets. For *any* encoder f_θ and input x, we obtain the vector representation for x as $f_\theta(x)$.

[1] We use the implementation by geoopt: `https://geoopt.readthedocs.io/`

For the metric space S we use the euclidean distance given by the $L2$ norm between the encoded representations. We define distance function $d(\cdot, \cdot)$,

$$d(p, q) = L2(f_\theta(p), f_\theta(q)) \tag{7}$$

The *Gromov Product* for points $p, q, r \in S$ is,

$$(q, r)_p = \frac{1}{2}(d(p, q) + d(p, r) - d(q, r)) \tag{8}$$

Using the *Gromov Product*, $\mathcal{H}$ is defined as the minimum value for which the following condition holds true for any four point combination $p, q, r, s \in S$,

$$(p, r)_s \geq \min((p, q)_s, (q, r)_s) - \mathcal{H} \tag{9}$$

Intuitively, this suggests that the metric relations between any four points are similar to what would have been in a tree, a theoretically 0-hyperbolic space, up to an additive constant $\mathcal{H}$.

To quantify $\mathcal{H}$-hyperbolicity for the dataset X in our experiments, we use a scale-invariant metric, defined as $\mathcal{H}_{rel}(X) = \frac{2\mathcal{H}(X)}{diam(X)}$, where diam(X) denotes the diameter of the set, defined as the maximal pairwise distance of the dataset samples in the representation space,

$$diam(X) = max\{d(x, y) | \forall x, y \in X\} \tag{10}$$

3.3 Estimating the Radius of Curvature $\mathcal{R}$

Previous works like Chami et al. (2019) use a fixed curvature across datasets when training neural networks in the hyperbolic space. As the extent of hyperbolic nature varies with the dataset, a common curvature is not suitable when operating in the hyperbolic space. Hence, we derive the radius of curvature $\mathcal{R}$ for a given hyperbolicity $\mathcal{H}$ obtained from §3.2. Tifrea et al. (2019) derives the hyperbolicity of a standard Poincaré disk ($\mathcal{H}_p$) as $\mathcal{H}_p = \log(1 + \sqrt{2}) \approx 0.88$. The diameter of a standard Poincaré ball is infinity, which yields a $\mathcal{H}_{rel}$ values of 0. From a computational perspective, we follow Khrulkov et al. (2020) to calculate the *effective* value of $\mathcal{H}_{rel}(\cdot)$. For clipping value ϵ, we consider points whose Euclidean norm does not exceed $1 - \epsilon$ to obtain the relative diameter $diam_p$. For a standard Poincaré ball, the relative hyperbolicity $\mathcal{H}_{rel_p}$ becomes,

$$\mathcal{H}_{rel_p} = \frac{\mathcal{H}_p}{(diam_p/2)} \approx \frac{0.88}{(diam_p/2)} \tag{11}$$

For dataset X with relative hyperbolicity $\mathcal{H}_{rel}(X)$, the adapted radius of curvature $\mathcal{R}(X)$ of the Poincaré disk is estimated as,

$$\mathcal{R}(X) = \left(\frac{\mathcal{H}_{rel_p}}{\mathcal{H}_{rel}(X)}\right)^2 \tag{12}$$

We use this curvature in-place of κ when performing hyperbolic operations.

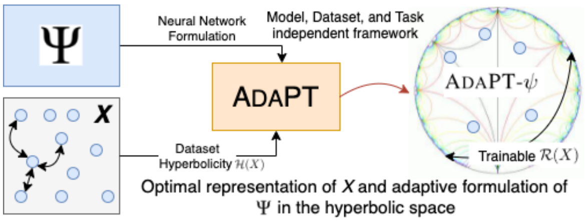
Ψ
Neural Network Formulation
Model, Dataset, and Task independent framework
ADAPT
ADAPT-ψ
X
Dataset Hyperbolicity $\mathcal{H}(X)$
Trainable $\mathcal{R}(X)$
Optimal representation of X and adaptive formulation of Ψ in the hyperbolic space

Figure 1: An overview of applying ADAPT to *any* neural network ψ with dataset X to give ADAPT-ψ.

3.4 Parameterizing the Radius of Curvature

Previous works performing operations in the hyperbolic geometric space keep a constant radius of curvature $\mathcal{R}$ during the course of the training. Since the hyperbolic space is sensitive to latent representations of the samples (Ganea et al., 2018), a constant curvature is not effective in capturing the complex geometries of these representations as the weights of the underlying model are updated. To capture the dynamic nature of the geometric representation of encodings, we propose training the model with a parameterized radius of curvature, initialised with $\mathcal{R}$ obtained using Equation 12. Hence, the radius of curvature is also jointly optimized along with the neural network training with optimizer function $O(\cdot)$,

$$\mathcal{R}_t \leftarrow O(\mathcal{R}_{t-1}, \alpha, \frac{\partial L}{\partial \mathcal{R}}) \tag{13}$$

, $\mathcal{R}$ is the trainable radius of curvature, α is the learning rate, and L being the loss calculated that incorporates the trainable curvature.

We define ADAPT as the cumulative application of necessary hyperbolic arithmetic operations (§3.1) and parameterized adaptive radius of curvature (§3.4), giving the optimal formulation of *any* given neural network method ψ in the hyperbolic space, **ADAPT-ψ** as shown in Figure 1.

4 ADAPT-ing State-of-the-art Methods to the Hyperbolic Space

To validate the effectiveness of ADAPT, we apply it over two existing state-of-the-art neural network training methods, Variational Information Bottleneck (VIB) (Mahabadi et al., 2021) and Saliency-Based Span Mixup (SSMix) (Yoon et al., 2021) and define them in the hyperbolic space as ADAPT-VIB and ADAPT-SSMix.

4.1 ADAPT

Algorithm 1 ADAPT OVERVIEW

```
$M \leftarrow$ Model Architecture
$F(\theta) \leftarrow$ Eucledian set of operations of $M$ performed in the forward-pass with trainable weights $\theta$.
$g(\phi) \leftarrow$ Subset of $F(\theta)$ chosen for transformations in hyperbolic space.
$f(\theta) \leftarrow$ Remaining set of operations after excluding $g(\phi)$.
$F(\theta) = f(\varphi) \bigcup g(\phi)$
$X \leftarrow$ set of inputs.
$Y \leftarrow$ true predictions.
$\mathcal{R}_0 \leftarrow$ initialized to $\mathcal{R}(X)$ as mentioned in (12)
$T \leftarrow$ number of update steps.

$\textbf{ADAPT}(g(\phi), \mathcal{R})(u) = \log_0^{\mathcal{R}}(g^h(\phi^h, \exp_0^{\mathcal{R}}(u), \mathcal{R}))$
where $g^h(\phi)$ is the hyperbolic analogous of $g(\phi)$ obtained from combining equivalent hyperbolic operations.

for $t \in \{1, \ldots, T\}$ do
    $F(\theta) = f(\varphi) \bigcup \textbf{ADAPT}(g(\phi), \mathcal{R}_t)$
    $Y' = F(\theta)(x)$
    $L = \text{LOSS}(Y, Y')$
    $\mathcal{R}_t \leftarrow O(\mathcal{R}_{t-1}, \alpha, \frac{\partial \mathrm{L}}{\partial \mathcal{R}})$        ▷ as mentioned in (13)
end for
```

We provide a generalized idea of how ADAPT guidelines can be applied for neural network training methods (Figure 1). Let $F(\theta)$ represent the set of operations constituting the forward pass of the model. A set of these operations are chosen for transformation in hyperbolic space $g(\phi)$. The choice of $g(\phi)$ is made based on the essential components of the model which have optimal representation and are a factor for the improved model performance, as shown in their corresponding work.

4.2 ADAPT-VIB

Variational information bottleneck (VIB) (Mahabadi et al., 2021) suppresses irrelevant features and reduces overfitting of the underlying base model when fine-tuning on low-resource target tasks. It addresses this problem of overfitting by adding a regularization term to the training loss to suppress irrelevant information. However, VIB performs operations on the latent encodings in the Euclidean space, which is not the most suitable representation space given the complex geometry of these latent embeddings.

We formulate VIB in the hyperbolic space, and propose Adaptive Poincaré Variational Information Bottleneck (ADAPT-VIB) using definitions from §3. As Information Bottleneck aims to learn maximal representation and suppress irrelevant features, we transform the bottleneck layers to hyperbolic space and these form our $g(\phi)$ for applying 1. ADAPT-VIB maps the sentence embedding from a pretrained encoder f_θ to a latent representation z using a shallow multi-layer perceptron (MLP_s) followed by hyperbolic linear (HL)[2] layers. This is the only input to the task-specific classifier, and this shallow network is trained using a combination of reducing compression loss and maximizing mutual information. Formally, to perform ADAPT-VIB for input $x \in X$ using encoder f_θ, we first feed the sentence embedding $f_\theta(x)$ through the shallow MLP_s and project this into the hyperbolic space using the $\exp_0^{\mathcal{C}(X)}(\cdot)$ mapping. We then use $HL(.,.)$ to obtain the mean vector μ and covariance matrix Σ,

$$\begin{aligned} \mu(x) &= \log_0^{C(X)}(HL(\exp_0^{C(X)}(MLP_s(f_\theta(x))), W_\mu)) \\ \Sigma(x) &= \log_0^{C(X)}(HL(\exp_0^{C(X)}(MLP_s(f_\theta(x))), W_\Sigma)) \end{aligned} \tag{14}$$

where W_μ and W_Σ are trainable weights. Following Mahabadi et al. (2021), we obtain $z = \mathcal{N}(\mu(x), \Sigma(x))$. We define $r(z) = \mathcal{N}(\mu_0, \Sigma_0)$ as an estimate of the prior probability $p(z)$, and $p_\theta(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x))$ as the estimate of the posterior probability of z. For output classifier $q_\phi(y|z)$ for labels y, we use the variational estimate of information bottleneck $L_{\text{ADAPT}-\text{VIB}}$ given by Alemi et al. (2017) to optimize the network,

$$\begin{aligned} L_{\text{ADAPT}-VIB} = \beta \mathop{\mathbb{E}}_{x}[KL[p_\theta(z|x), r(z)]] + \\ \mathop{\mathbb{E}}_{z \sim p_\theta(z|x)}[-\log q_\phi(y|z)] \end{aligned} \tag{15}$$

where β is a hyperparameter and $q_\phi(y|z)$ is estimated using an MLP classifier(MLP_{clf}).

4.3 ADAPT-SSMix

Saliency measures how each portion of the input affects the final prediction and is indicative of its degree of importance. Saliency-aware interpolative augmentation has proven to be effective over standard mixup (Zhang et al., 2018a) for various modalities as it preserves the locality of samples being interpolated (Yoon et al., 2021; Kim et al., 2020). For span-based interpolation, the least salient span of one input is replaced with the most salient span of another input. The saliency of a span is defined as the pooled saliency over each portion of the input sample k, given as $\delta L/\delta k$ for classification loss L. Existing saliency-aware interpolative methods

[2] Details provided in section 3.1

operate in the simplified Euclidean space, which is not capable enough to model the inherent complex geometries possessed by the portion gradients due to the hyperbolic nature of their latent representations. As saliency computation constitutes an essential step in the mixup, we choose that as our $g(\phi)$ as described in 1. We utilize the operations defined in §3 to formulate saliency calculation in the hyperbolic space. We use weighted Möbius gyromidpoint (M_κ) [2] to obtain the measure of the saliency from the gradient vector $\delta L/\delta e$ of each token instead of the standard Euclidean norm [2].

For an input token $x \in X$ having an embedding vector representation e of dimension n, gradient $\delta L/\delta e$ is also an n dimensional vector. As we are concerned with the magnitude we take a square of each value and project them into hyperbolic space with curvature $\mathcal{C}(X)$ using $\exp_0^{\mathcal{C}(X)}(\cdot)$. We then compute the weighted midpoint of these n values in the vector, assigning equal weight of 1 to all input units. We map the hyperbolic saliency H back to the Euclidean space using $\log_0^{\mathcal{C}(X)}(\cdot)$ to obtain S_x, the saliency of token x,

$$H_x = M_{\mathcal{C}(X)}(\exp_0^{\mathcal{C}(X)}([(\delta L/\delta e_0)^2, (\delta L/\delta e_1)^2, \ldots, (\delta L/\delta e_n)^2], 1, 1, \ldots, 1)$$

$$S_x = \log_0^{\mathcal{C}(X)}(H_x) \tag{16}$$

Span saliency value is obtained by mean pooling over the saliency value of the tokens in the span. For input samples x_i and x_j, we replace the least salient portion $x_i[p:q]$, S^i_{min} in x_i with the most salient portion in $x_j[u:v]$, S^j_{max} to generate $\tilde{x}$ with transport η from $[p:q] \rightarrow [u:v]$. We denote this procedure as AdaPT-SSMix,

$$\tilde{x} = \text{AdaPT-SSMix}(x_i, x_j), \quad \tilde{x}_k = \begin{cases} x_{i,k} & k \notin [p:q] \\ x_{j,k+\eta} & k \in [p:q] \end{cases} \tag{17}$$

For the mixup ratio $\lambda = |x_j[u:v]|/|\tilde{x}|$, we define mixup loss L_{mix} as,

$$\mathcal{L}_{mix}(x_i, x_j) = \lambda * \text{CE}(y_i || f_\theta(\text{AdaPT-SSMix}(x_i, x_j))) + (1-\lambda) * \text{CE}(y_j || f_\theta(\text{AdaPT-SSMix}(x_i, x_j))) \tag{18}$$

, where CE denotes the cross entropy loss. For samples x_i and x_j, we optimize our network as a mean of four losses, giving loss $\mathcal{L}_{\text{AdaPT}-SSMix}$,

$$\mathcal{L}_{\text{AdaPT}-SSMix} = \frac{1}{4} * \Big(\text{CE}(y^i || f_\theta(x^i)) + \text{CE}(y^j || f_\theta(x^j)) + \mathcal{L}_{mix}(x^i, x^j) + \mathcal{L}_{mix}(x^j, x^i)\Big) \tag{19}$$

	Dataset	Language	# Classes
Text	CoNLL-2003 2003	English	4
	RTE 2009	English	2
	MRPC 2005	English	2
	XNLI 2018	Hi, Tr, Ur, En, Zh, Ru, Es, Ar, De, Sw	3
Speech	Urdu SER 2020	Urdu	4
	EmoVO 2014	Italian	7
	ShEMO 2019	Persian	6

Table 1: Datasets, languages, and # classes.

5 Experimental Setup

5.1 Datasets and Preprocessing

We consider various benchmark and low-resource datasets across text and speech (Table 1). For text, we compare our methods over standard datasets such as RTE (Bentivogli et al., 2009), MRPC (Dolan and Brockett, 2005), Conll-2003 (Tjong Kim Sang and De Meulder, 2003), and XNLI (Conneau et al., 2018) in Hindi (Hi), Turkish (Tr), Urdu (Ur), English (En), Chinese (Zh), Russian (Ru), Arabic (Ar), German (De), and Swahili (Sw). For speech, we use low resource speech classification datasets, Urdu SER (Urdu) (Latif et al., 2020), EmoVO (Italian) (Costantini et al., 2014), and ShEMO (Persian) (Mohamad Nezami et al., 2019).

Text For both AdaPT-VIB and AdaPT-SSMix, we follow the same preprocessing steps as previous works, VIB (Mahabadi et al., 2021) and SSMix (Yoon et al., 2021), for a fair comparison.

Speech We resample the audio files to a frequency of 16kHz. We then define a feature extractor for preparing the inputs which takes as input the sampling frequency of the model and normalizes the data to zero-mean and unit-variance.

5.2 Task Setup

AdaPT-VIB For text, we evaluate AdaPT-VIB on NLI tasks in multiple languages and NER for English. For NLI, we train on 600 samples from the original backtranslated sentences used for training XNLI. For speech modality, we evaluate our methods on speech classification datasets for speech emotion recognition task in different languages.

AdaPT-SSMix We validate our approach on NLI as well as sentence classification tasks over standard datasets in multiple languages.

5.3 Calculating Hyperbolicity $\mathcal{H}$

For practical computations, we find the $\mathcal{H}$ values for fixed points $s = s_0, s_0 \in S$ as it is independent of s (Fournier et al., 2015). For a set of points, we find the matrix G of pairwise Gromov products using Equation (8). The value of $\mathcal{H}$ is equal to the largest coefficient in the matrix $(G \otimes G) - G$, where $\otimes$ denotes the min-max matrix product,

$$X \otimes Y = \max_{\mathrm{k}} \min\{X_{ik}, Y_{kj}\} \tag{20}$$

Owing to the computational complexities of Equations 8 and 20, we compute the $\mathcal{H}_{rel}X$ in batches. For each run, we sample 200 points from the training datasets, and find the corresponding $\mathcal{H}_{rel}$. We average the results across 10 runs.

5.4 Training Setup

ADAPT-VIB-Text We use AdamW optimizer with a learning rate of 2e-5 with a batch size of 8, and train for 10 epochs. Following Mahabadi et al. (2021), we vary β over $\{10^{-4}, 10^{-5}, 10^{-6}\}$ and the output dimension of the hyperbolic linear layer $HL(\cdot, \cdot)$ over $\{12, 18, 24, 36, 48, 72, 96, 144, 192, 288, 384\}$. For datasets in English, we use BERT (Devlin et al., 2019) as our base model f_θ and for other languages, we use mBERT as our base model f_θ.

ADAPT-VIB-Speech We use AdamW optimizer with a learning rate of 1e-4 and batch size of 8 for 8 epochs. We use a linear annealing schedule for β and set $\beta = \text{epoch x} \beta_0$ where β_0 is set to 1e-5. The dimension of information bottleneck is set to 512 and use a train-test ratio of 80:20 for all datasets. For ShEMO, we sample 500 samples via stratified sampling. We use wav2vec2-large-xlsr-53 (Conneau et al., 2021) as f_θ.

ADAPT-SSMix Following Yoon et al. (2021), we set a maximum sequence length of 128, batch size of 32, with AdamW optimizer with eps of 1e-8 and weight decay of 1e-4. We train with a learning rate of 5e-5 for 200,000 iterations. We follow previous works to choose the span length for saliency-based interpolation. For datasets in English, we use BERT (Devlin et al., 2019) and for other languages, we use mBERT as our base model f_θ.

Dataset ($\mathcal{H}$)	f_θ	+VIB	+HVIB	+HVIB-$\mathcal{C}$	ADAPT-VIB
Hi (0.16)	40.22	41.13	43.34*	44.21*	**45.34***
Tr (0.18)	40.65	41.67	43.95*	44.01*	**44.69***
En (0.13)	43.29	46.68	48.57*	50.19*	**50.45***
Zh (0.12)	42.32	46.03	47.10*	46.22*	**51.35***
Ru (0.15)	41.55	45.10	**47.88***	45.12	46.72*
Es (0.26)	52.15	55.18	55.97	55.61	**56.81***
CoNLL (0.19)	92.80	94.51	94.55	94.68*	**94.92***

Table 2: Performance comparison in terms of accuracy(%) of ADAPT-VIB for NLI and F1 score for NER. Improvements are shown with green (↑). Bold shows the best result. $*$ shows significant (p < 0.01) improvement over VIB, under Wilcoxon's signed-rank test. Lower value of $\mathcal{H}$, signifies more tree-like structure of the data.

6 Results and Analysis

6.1 Performance Comparison: ADAPT-VIB

Text We present the results of applying ADAPT over variational information bottleneck (VIB) (Mahabadi et al., 2021) in Table 2. We observe that using variational information bottleneck performs better than the base model (f_θ), by reducing overfitting during training by suppressing irrelevant information, and allows to keep only relevant and concise information which is more suitable for training the neural network. We further find that hyperbolic variational information bottleneck (HVIB, constant radius of curvature) significantly improves ($p < 0.01$) the performance over the Euclidean VIB. This validates that the hyperbolic space is better able to capture the hierarchical nature of text (Tifrea et al., 2019) and is a more suitable geometry to calculate the maximally compressed representation of the latent embeddings. Further improvements are observed when we use dataset (X) specific radius of curvature ($\mathcal{R}(X)$) to define the Poincaré disk (HVIB-$\mathcal{C}$, constant radius of curvature), indicating that it better captures the extent of hyperbolic nature of the dataset, and is the better representative geometry for the same. We obtain the best performance across most of the datasets when we parameterize the radius of curvature $\mathcal{R}$, essentially infusing VIB with ADAPT (ADAPT-VIB). This validates our hypothesis that a trainable curvature is capable of adapting to the stochastic hidden representations of input samples in conjunction with the dynamically changing weights of the underlying model being fine-tuned, and captures the optimal geometric representation.

Speech We observe that using variational information bottleneck (VIB) strategy over latent representation with XLSR (C.1) performs better than

XLSR (f_θ) (Table 3). This suggests that information bottleneck is able to overcome overfitting in low-resource settings and achieve generalization. Hyperbolic variational information bottleneck (HVIB) further improves performance in most cases as it leverages the hyperbolic space for learning bottleneck layers. This validates that hyperbolic geometry is better able to capture the relevant features of speech signals and acoustic wave interference, which follows hyperboloid geometry (Khan and Panigrahi, 2016). We observe better performance when we use a dataset specific radius of curvature (HVIB-$\mathcal{C}$) to represent the Poincaré space as it is better able to apprehend the hyperbolic curvature of the dataset. Trainable curvature (ADAPT-VIB) achieves significantly best performance ($p < 0.01$) as it allows to fine-tune the curvature to the optimal value and adjust to the hyperbolic precision of the dataset. The hyperbolic bottleneck layer weights adjust to the hyperbolicity of the hidden representations while the underlying encoder model is fine-tuned. The substantial improvement in performance for speech compared to text can be attributed to the fact that speech waves innately possess hyperbolic nature(Khan and Panigrahi, 2016).

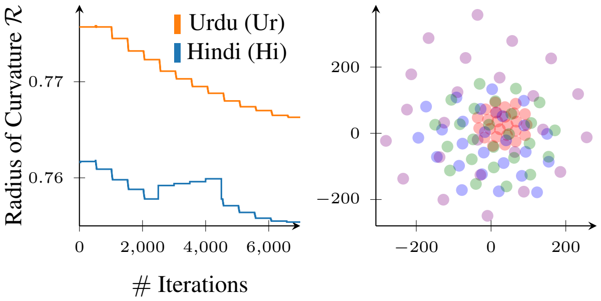

Figure 2: Change in curvatures to account for the shift in embedding distribution before and after training with ADAPT-VIB. Hi-before and Ur-before denote the embeddings before training; Hi-after and Ur-after denote the embeddings after training.

6.2 Performance Comparison: ADAPT-SSMix

We compare the performance of applying ADAPT over SSMix for XNLI tasks in Table 4. SSMix (Euclidean saliency-aware mixup) achieves better performance than base model f_θ. This shows the importance of using semantically salient spans for mixup as the generated samples are more related to the prediction (Yoon et al., 2021). Using the hyperbolic variant (HSMix) further improves performance suggesting that hyperbolic space is better able to relatively quantify the saliency measure of tokens which are measured using the token wise training loss vector and choose relevant spans for mixup. We observe best performance when the saliency computation is performed with dataset specific radius of curvature (ADAPT-SSMix) as it uses hyperbolic operations adapted for the dataset to compute saliency. This validates its capability to better model the network gradient space and adjust better to the dataset hierarchical properties.

Dataset Hyperbolicity $\mathcal{H}$	ShEMO 0.24	Urdu SER 0.21	EmoVO 0.18
f_θ	59.20	81.25	29.66
+ VIB	51.00	90.00	37.28
+ HVIB	60.40*	90.42	41.52*
+ HVIB-$\mathcal{C}$	60.50*	82.50	42.55*
+ ADAPT-VIB	**63.40***	**92.50***	**54.23***

Table 3: Performance comparison in terms of accuracy(%) of ADAPT-VIB on speech datasets. Improvements are shown with green (↑). Bold shows the best result. ∗ shows significant (p < 0.01) improvement over VIB, under Wilcoxon's signed-rank test. Lower value of $\mathcal{H}$, signifies more tree-like structure of the data.

Model Hyperbolicity $\mathcal{H}$	RTE 0.11	MRPC 1.30	Ar 0.26	De 0.21	Zh 0.12	Sw 0.14
f_θ	62.20	86.60	63.91	68.72	65.21	55.87
SSMix	67.73	86.72	65.42	70.11	67.81	57.59
HSMix	67.61	87.06*	65.87	72.71*	68.55*	58.27*
ADAPT-SSMix	**68.23***	**88.01***	**66.10***	**73.13***	**69.12***	**58.71***

Table 4: Performance comparison in terms of accuracy(%) of ADAPT-SSMix for classification and NLI. Improvements are shown with green (↑). Bold shows the best result. ∗ shows significant (p < 0.01) improvement over SSMix, under Wilcoxon's signed-rank test. Lower value of $\mathcal{H}$, signifies more tree-like structure of the data.

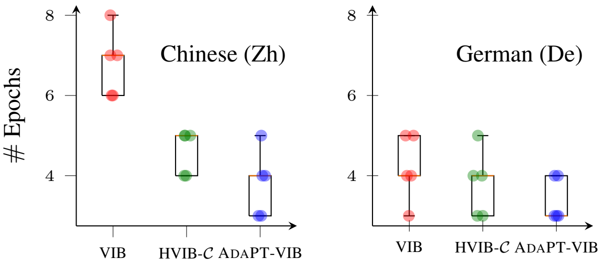

Figure 3: Computational efficiency comparison of VIB with HVIB and ADAPT-VIB in terms of training epochs required to achieve benchmark accuracy (Chinese (Zh): 42%, German (De): 45%).

Source
Target
	Ar ($\mathcal{H} = 0.27$)
	De ($\mathcal{H} = 0.22$)
		Zh($\mathcal{H} = 0.12$)
		Sw($\mathcal{H} = 0.15$)

f_θ	46.36	42.65
SSMix	43.59	43.01
ADAPT-SSMix + source $\mathcal{C}$	45.68	43.27
ADAPT-SSMix + target $\mathcal{C}$	**46.58**	**43.67**

Table 5: Accuracy(%) comparison for Zero-Shot Cross-Lingual transfer on XNLI.

6.3 Probing the Adaptiveness of the Curvature with the Embedding Shift

We validate the ability of the parameterized adaptive curvature to model the dynamic complex geometry of the inputs during the neural network training. We observe that the embedding space expands [3] as the model is trained using ADAPT-VIB as shown in Figure 2, denoting a more hyperbolic space on account of greater maximal distance between latent representations. To adapt to this change, the corresponding radius of curvatures decrease, according to the relation in Equation 12, optimally modeling the hyperbolic nature of the dataset during each iteration and leading to improved performance.

6.4 Effect of Hyperbolic Curvatures on Zero-shot Transferability

We compare the performance for zero-shot cross-lingual NLI using ADAPT-SSMix in Table 5. For ADAPT-SSMix, we experiment with using the source language's curvature and the target language's curvature during its formulation. We observe that in both the settings, ADAPT-SSMix performs better than SSMix for zero-shot transfer, revalidating the effectiveness of the hyperbolic space. Interestingly, we observe better performance for ADAPT-SSMix when the hyperbolicity of the target dataset is used for its formulation. This suggests that the model learns to represent the training distribution better to the complex geometries possessed by the target dataset, improving zero-shot transfer performance on the target dataset.

Dataset	VIB	HVIB	HVIB-$\mathcal{C}$	ADAPT-VIB
Hi	0.447	0.452	0.448	0.461
Tr	0.452	0.448	0.456	0.459
Urdu SER	2.711	2.525	2.800	2.850

Table 6: Time (in s/iter) for VIB, HVIB, HVIB-$\mathcal{C}$, and ADAPT-VIB.

6.5 Computational Efficiency of ADAPT

We verify the computational efficiency of ADAPT by applying it over VIB, as the number of epochs required to achieve a benchmark accuracy (Figure 3). On an average, ADAPT-VIB achieves the benchmark accuracy in lesser number of training epochs as compared to VIB. Further, the per iteration training time is almost the same as shown in Table 6. Thus, ADAPT-VIB improves over the baselines with no extra computation overhead.

7 Conclusion, Future Work, Limitations

Drawing inspiration from works showing that various datasets and their latent representations inherently possess hyperbolic characteristics and can be better represented in the hyperbolic space, we propose ADAPT, a data and task independent set of guidelines that can be applied over *any* existing neural network training method to maximally leverage the hyperbolic space. ADAPT obtains significant improvements over existing training methodologies on three tasks in 12 languages across text and speech without any computational overhead. As future work, we plan to extend ADAPT to multimodal and graph neural network training methods. Though ADAPT is capable of utilizing the optimal representation space as it has a trainable curvature, it is difficult to theoretically claim when to use it purely based on the δ-hyperbolicity of the datasets as it is an underexplored area of research. We leave the deeper analysis of the hyperbolic space for NLP applications as future work.

References

Rodrigo Aldecoa, Chiara Orsini, and Dmitri Krioukov. 2015. Hyperbolic graph generator. *Computer Physics Communications*, 196:492–496.

Alex Alemi, Ian Fischer, Josh Dillon, and Kevin Murphy. 2017. Deep variational information bottleneck. In *ICLR*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR 2015 : International Conference on Learning Representations 2015*.

Luisa Bentivogli, Bernardo Magnini, Ido Dagan, Hoa Trang Dang, and Danilo Giampiccolo. 2009. The fifth PASCAL recognizing textual entailment challenge. In *Proceedings of the Second Text Analysis Conference, TAC 2009, Gaithersburg, Maryland, USA, November 16-17, 2009*. NIST.

[3] We provide more details in the supplementary.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.

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.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. 2019. Hyperbolic graph convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 4869–4880.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. 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.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised cross-lingual representation learning for speech recognition. pages 2426–2430.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Giovanni Costantini, Iacopo Iaderola, Andrea Paoloni, and Massimiliano Todisco. 2014. EMOVO corpus: an Italian emotional speech database. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. Embedding text in hyperbolic spaces. In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, New Orleans, Louisiana, USA. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. 2021. A survey of data augmentation approaches for NLP. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988, Online. Association for Computational Linguistics.

Hervé Fournier, Anas Ismail, and Antoine Vigneron. 2015. Computing the gromov hyperbolicity of a discrete metric space. *Inf. Process. Lett.*, 115(6):576–579.

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

Mikhael Gromov. 1987. Hyperbolic groups. In *Essays in group theory*, pages 75–263. Springer.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. Hyperbolic attention networks. In *International Conference on Learning Representations*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Md Nazoor Khan and Simanchala Panigrahi. 2016. *Interference*, page 98–185. Cambridge University Press.

Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. 2020. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. 2020. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning (ICML)*.

Siddique Latif, Adnan Qayyum, Muhammad Usman, and Junaid Qadir. 2018. Cross lingual speech emotion recognition: Urdu vs. western languages. In *2018 International Conference on Frontiers of Information Technology (FIT)*, pages 88–93.

Siddique Latif, Adnan Qayyum, Muhammad Usman, and Junaid Qadir. 2020. Cross lingual speech emotion recognition: Urdu vs. western languages.

Rabeeh Karimi Mahabadi, Yonatan Belinkov, and James Henderson. 2021. Variational information bottleneck for effective low-resource fine-tuning. In *International Conference on Learning Representations*.

Omid Mohamad Nezami, Paria Jamshid Lou, and Mansoureh Karami. 2019. Shemo: a large-scale validated database for persian speech emotion detection. *Language Resources and Evaluation*, 53(1):1–16.

Maximilian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations.

Wei Peng, Jingang Shi, Zhaoqiang Xia, and Guoying Zhao. 2020. Mix dimension in poincaré geometry for 3d skeleton-based action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 1432–1440, New York, NY, USA. Association for Computing Machinery.

Ramit Sawhney, Megh Thakkar, Shivam Agarwal, Di Jin, Diyi Yang, and Lucie Flek. 2021. HypMix: Hyperbolic interpolative data augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9858–9868, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ryohei Shimizu, YUSUKE Mukuta, and Tatsuya Harada. 2021. Hyperbolic neural networks++. In *International Conference on Learning Representations*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Hyperbolic representation learning for fast and efficient neural question answering. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*.

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

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019. Manifold mixup: Better representations by interpolating hidden states. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6438–6447, Long Beach, California, USA. PMLR.

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. 2018a. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*.

Zhichao Zhang, Shugong Xu, Shan Cao, and Shunqing Zhang. 2018b. Deep convolutional neural network with mixup for environmental sound classification. In *Chinese conference on pattern recognition and computer vision (prcv)*. Springer.

A Change in the Embedding Space during training

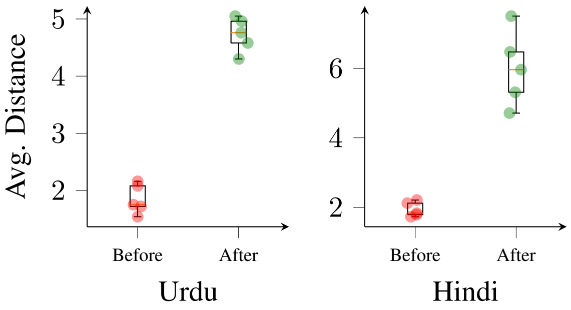

Figure 4: Change in average distance between embedding of sentences for Urdu and Hindi datasets before and after training with ADAPT-VIB.

We measure the change in the average pairwise distance of a random sample of inputs using the base model (mBERT) as an encoder before and after training using ADAPT-VIB and show the results in Figure 4. We observe an expansion in the embedding space during the course of training, which is effectively captured by the dynamically training radius of curvature being jointly optimized along with the neural network architecture. This validates our hypothesis that a parameterized radius of curvature has the ability to represent the stochastic nature of latent representations having a complex geometry optimally during the training and leads to significant performance gains.

B Task Setup

We evaluate ADAPT across three tasks for an extensive comparison with baseline methods.

Classification Tasks We assess ADAPT-SSMIX on standard datasets for text classification. We evaluate the ability of ADAPT-VIB on low-resource datasets for speech classification task.

NLI We evaluate the ability of ADAPT-VIB and ADAPT-SSMIX on Natural Language Inference(NLI) task for text in multiple languages in low-resource and full-resource settings respectively.

NER For text, we perform Named Entity Recognition task in English to measure the improvement by leveraging ADAPT-VIB.

For all tasks, we compare the performance with base-models and Euclidean counterparts.

C Experiment Setup

C.1 Variational Information Bottleneck

Text We use BERT (Devlin et al., 2019) as the backbone architecture ($f_\theta(\cdot)$), where BERT-base is utilized [4] for English datasets and mBERT [5] for all other datasets. For latent representations, $\mu(x)$ and $\sum(x)$ we vary the dimensions in the range {12,18,24,36,48,72,96,144,192,288,384}. We use a linear layer on top with hidden size same as dimension of $\mu(x)$, which acts as the classifier ($q_\phi(y|z)$). The MLP through which ($f_\theta(\cdot)$) is passed to compute compressed representations is a shallow multi-layer perceptron with 768, $\frac{2304+D}{4}$, $\frac{768+D}{2}$ hidden units with a ReLU non-linearity, where $D =$ is equal to the dimension of $\mu(x)$. We compare ADAPT-VIB for text with VIB[6](Mahabadi et al., 2021), HVIB, HVIB-$\mathcal{C}$ and the base model.

Speech We use XLSR-53[7](Conneau et al., 2021) built on wav2vec 2.0 as the backbone architecture ($f_\theta(\cdot)$) for all languages. For latent representations, $\mu(x)$ and $\sum(x)$ we set the dimension to be 512. The MLP through which ($f_\theta(\cdot)$) is passed to compute compressed representations is a shallow multi-layer perceptron with 1024, $\frac{3072+D}{4}$, $\frac{1024+D}{2}$ hidden units with a ReLU non-linearity, where $D =$ is equal to the dimension of $\mu(x)$. We use a two layer MLP with hidden size 512 and TanH activation as the classifier ($q_\phi(y|z)$). We compare ADAPT-VIB for speech with VIB, HVIB, HVIB-$\mathcal{C}$ and the base model.

C.2 Saliency-Aware Interpolation

We perform sequence classification task built upon encoders BERT-base and mBERT for English and other languages respectively. For mixing two examples x^i and x^j, the length of least salient span of x^i, S^i_{min} is denoted as l_a and the length of most salient span of x^j, S^i_{min} is denoted ad l_b. We set $l_a = l_b = max(min([\lambda_0|x^i|], |x^j|))$ where λ_0 is set as 0.1. We compare ADAPT-SSMIX for text with SSMix[8](Yoon et al., 2021), HSMix, HSMix-$\mathcal{C}$ and the base model.

C.3 Training Setup

Variational Informational Bottleneck
For both modalities, we initialize the curvature of the Poincaré space with the respective dataset curvatures calculated $R(.)$. Following (Bowman et al., 2016; Mahabadi et al., 2021), we use a linear annealing schedule for β and set $\beta = min(1, \text{epoch x}\, \beta_0)$. While training we average over 5 posterior samples to compute the loss (Alemi et al., 2017), i.e. we compute $p(y|x) = \frac{1}{5}\sum_{i=1}^{5} q_\phi(y|z_i)$, where $z_i\; p_\theta(z|x)$.

Text: We use AdamW optimizer with a learning rate of 2e-5 with a batch size of 8, trained for 10 epochs. Following Mahabadi et al. (2021), we vary β over $\{10^{-4}, 10^{-5}, 10^{-6}\}$ and the output dimension of the hyperbolic linear layer $HL(\cdot,\cdot)$ over $\{12, 18, 24, 36, 48, 72, 96, 144, 192, 288, 384\}$.

Speech: We use the AdamW optimizer with a learning rate of 1e-4 and batch size of 8 trained for 8 epochs.

Saliency-Aware Interpolation Following Yoon et al. (2021), we set a maximum sequence length of 128, batch size of 32, with AdamW optimizer with eps of 1e-8 and weight decay of 1e-4. We train with a learning rate of 5e-5 for 200,000 iterations. We follow previous works to choose the span length for saliency-based interpolation.

We carry out all the experiments on a Tesla P100 GPU. We list the detailed training setups in Table 10 and Table 11. We use the existing available codes for both VIB and SSMix and develop over the same to run over experiments.

[4] https://huggingface.co/bert-base-uncased

[5] https://huggingface.co/bert-base-multilingual-uncased

[6] Code available at: https://github.com/rabeehk/vibert

[7] https://huggingface.co/facebook/wav2vec2-large-xlsr-53

[8] Code available at: https://github.com/clovaai/ssmix

Dataset	Task	# Classes	# Train Instances	# Val Instances	# Test Instances
XNLI	Inference	3	600	2,500	5,000
CoNLL-2003	NER	4	14,987	3,466	3,684

Table 7: Datasets statistics used for ADAPT-VIB experiments on Text Data.

Dataset	Labels	# Classes	# Train Instances	# Test Instances
Urdu SER	Emotion	4	320	80
ShEMO	Emotion	6	400	100
EMOVO	Emotion	7	470	118

Table 8: Datasets statistics used for ADAPT-VIB experiments for Speech Emotion Recognition.

D Datasets

We consider various benchmark as well as low-resource datasets across text and speech for an extensive evaluation of ADAPT. We present statistics of the datasets for VIB-Text in 7, VIB-Speech in 8, and SSMix in 9.

Text Datasets

XNLI[9](Conneau et al., 2018) is an evaluation corpus for language transfer and cross-lingual sentence classification in 15 languages. It is a crowd-sourced collection of $5,000$ test and $2,500$ dev pairs for the MultiNLI corpus. The pairs are annotated with textual entailment and translated into 14 languages: French, Spanish, German, Greek, Bulgarian, Russian, Turkish, Arabic, Vietnamese, Thai, Chinese, Hindi, Swahili and Urdu. Following (Conneau et al., 2018) we use XNLI-MT (TRANSLATE TRAIN) data for training - 392,703 samples. For Information Bottleneck experiments we sample a balanced subset of 600 samples from the training data to understand the performance in low-resource settings.

RTE (Bentivogli et al., 2009)[10] is used for Recognising Textual Entailment in 2 sentences. It consists of $2,500$ training instances and $3,000$ testing instances.

MRPC (Dolan and Brockett, 2005)[10] consist of English sentence pairs where each pair is labeled if it is a paraphrase or not. $3,700$ sentece pairs are part of the training set and $1,700$ are part of the test set.

CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003)[11] has been used for the Named Entity Recognition task. The dataset covers data in two languages English and German of which we use the English data. The training set consists of about $14,987$ sentences in the training set, $3,466$ sentences in the dev set and $3,684$ sentences in the test set.

Speech Datasets

Urdu Speech Emotion Recognition[12] (Latif et al., 2018) contains 100 clips corresponding to 4 emotion labels, for a total of 400 sound samples. We split the dataset into train and test split with a ratio of $80 : 20$.

ShEMO[13] (Mohamad Nezami et al., 2019) contains 3000 semi-natural utterances, equivalent to 3 hours and 25 minutes of speech data extracted from online radio plays. The ShEMO covers speech samples of 87 native-Persian speakers for five basic emotions as well as neutral state. We sample 500 samples balanced according to labels and use a train and test split in the ratio $80 : 20$.

EmoVO Corpus[14] (Costantini et al., 2014) is an Italian emotional speech database which containing voice clips of up to 6 actors who played 14 sentences simulating 6 emotional states and the neutral state, hence resulting in 588 audio samples. We split the dataset into train and test split with a ratio of $80 : 20$.

E Preprocessing

Text For both ADAPT-VIB and ADAPT-SSMix, we follow the same preprocessing steps as previous works, VIB (Mahabadi et al., 2021) and SSMix (Yoon et al., 2021), for a fair comparison.

Speech We first read the audio files and resample it to a frequency of 16kHz as XLSR- wav2vec

[9] XNLI publicly available at: https://github.com/facebookresearch/XNLI

[10] RTE, MRPC dataset publicly available at:https://huggingface.co/datasets/glue

[11] CoNLL-2003 dataset publicly available at https://www.clips.uantwerpen.be/conll2003/ner/

[12] Urdu SER dataset publicly available at: https://github.com/siddiquelatif/URDU-Dataset

[13] ShEMO dataset publicly available at: https://github.com/pariajm/sharif-emotional-speech-dataset

[14] EMOVO dataset publicly available at:http://voice.fub.it/EMOVO

Dataset	Task	# Classes	# Train Instances	# Test Instances
RTE	Entailment Recognition	2	2,500	3,000
MRPC	Paraphrase Detection	2	3,700	1,700
XNLI	Inference	3	392,703	5,000

Table 9: Datasets statistics used for ADAPT-SSMix experiments on Text Data.

2.0 was majorly pretrained on data sampled at this frequency. To make the inputs compatible to our model, We then define a feature extractor for preparing the inputs which takes as input the sampling frequency of the model and normalizes the data to zero-mean and unit-variance. The padding value for batch implementation is set to 0.0. For ShEMO we randomly crop $2s$ of audio from each recording and use it for training.

Parameter	Modality	Value
Optimizer	Text	AdamW
	Speech	AdamW
Learning Rate	Text	2e-5
	Speech	1e-4
Batch Size	Text	8
	Speech	8
$\beta_1, \beta_2, \epsilon$ for AdamW	Text	0.9, 0.999, 1e-8
	Speech	0.9, 0.999, 1e-6
# Epochs	Text	10
	Speech	8
Evaluation Metric		Accuracy
Base Model $f_\theta(.)$	Text	BERT-base-uncased, BERT-base-multilingual-uncased
	Speech	XLSR-53
Encoder Output Dimension $\lvert f_\theta(x)\rvert$	Text	768
	Speech	1024
MLP Shallow $MLP_s(.)$	Text	768, $\frac{2304+\lvert z\rvert}{4}$, $\frac{768+\lvert z\rvert}{2}$
(input dim, hidden dim, output dim)	Speech	1024, $\frac{3072+\lvert z\rvert}{4}$, $\frac{1024+\lvert z\rvert}{2}$
Information Bottleneck linear layer dim, $\lvert z\rvert$	Text	384 (optimal)
	Speech	512
MLP Classifier $MLP_{clf}(.)$	Text	Linear Layer
(over architecture)	Speech	2 layer MLP with hidden size 512
Hardware		Tesla P100

Table 10: Model and training setup for ADAPT-VIB.

Parameter	Value
Optimizer	AdamW
Learning Rate	1e-5, 5e-5
Batch Size	32
$\beta_1, \beta_2, \epsilon$	0.9, 0.999, 1e-8
# Iterations	200,000
Evaluation Metric	Accuracy
Base Model	BERT-base-uncased, BERT-base-multilingual-uncased
Classifier (over architecture)	We follow Yoon et al. (2021)
Hardware	Tesla P100

Table 11: Model and training setup for ADAPT-SSMix.