IPA-CLIP: Integrating Phonetic Priors into Vision and Language Pretraining

Anonymous ACL submission

Abstract

Large-scale Vision and Language (V&L) pretraining has recently become the standard backbone of multimedia systems. While it has shown remarkable performance even in zero-shot scenarios, it often performs in ways not intuitive to humans. Particularly, they do not consider the pronunciation of the input, which humans would utilize to process language. Thus, this paper inserts phonetic prior into Contrastive Language-Image Pretraining (CLIP), one of the V&L pretrained models, to make it consider the pronunciation similarity among its language inputs. To achieve this, we first propose a phoneme embedding that uses the phoneme relationships on the International Phonetic Alphabet (IPA) chart as a phonetic prior. Next, by distilling the CLIP text encoder, we train a pronunciation encoder employing the IPA-based embedding. The proposed model named IPA-CLIP comprises this pronunciation encoder and the original CLIP encoders (image and text). Quantitative evaluations show that IPA-CLIP accurately processes words in a more phonetic manner, which is promising for downstream tasks. A qualitative evaluation verifies a high correlation to human perception regarding pronunciation similarity.

1 Introduction

Vision and Language (V&L) pretraining from large-scale image-text datasets has gained increasing attention as a fundamental model of multimedia systems. Contrastive Language-Image Pretraining (CLIP) (Radford et al., 2021) is one of such V&L pretrained models consisting of an image encoder and a text encoder that share their bi-modal embedding space. It uses far larger training data than previous models, which guarantees its effectiveness in various applications including image classification and retrieval (Radford et al., 2021), object detection (Shi et al., 2022), image generation (Crowson et al., 2022), and image captioning (Galatolo et al., 2021). This also allows it to perform well even in scenarios not seen in the training set.

However, in many cases, such models do not behave in a way intuitive to humans. One of the reasons is that they do not consider the phonetic similarity among words, which humans would consciously or unconsciously utilize to express the meanings of words intuitively. For example, an English speaker who uses the word "*Lump*" in a conversation might have a connotation of something **heavy and round**, akin to other similar-sounding words "*Bump*", "*Slump*", or "*Plump*". Humans also use phonetic similarity to process spoken language (Hahn and Bailey, 2005), especially when they hear unknown or nonsense words (in short, nonwords). Such nonwords may force humans to recall their similar-sounding words. For instance, the pronunciation of a nonword "*Britch*" might remind English speakers of a similar-sounding word "*Bridge*", thus the meaning of "*Britch*" might be recognized as something related to a bridge. Meanwhile, another nonword "*Brish*" (rhymes with "*Fish*") might be less perceived so because of its less phonetic similarity to "*Bridge*". Without knowing phonetic relationships, conventional models can not consider such correspondences.

The goal of this paper is to insert phonetic priors into V&L pretrained models to make them consider phonetic similarity. This would enable them to associate nonwords with their phonetically similar words, which will make them better correspond to human expectations towards nonwords. A possible approach to insert phonetic knowledge into a pretrained model is to change the tokenizer of their text encoders and retrain the whole model. Yet, existing tokenizing and embedding methods (Sennrich et al., 2016; El Boukkouri et al., 2020; Ma et al., 2020) are not sufficient as they do not consider phonetic similarity. One obstacle is that their language input is usually written with graphemes, which do not necessarily correspond to phonemes.

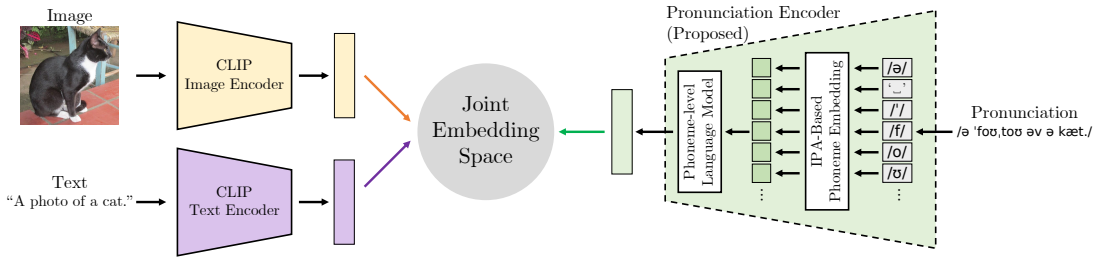


Figure 1: Overview of the proposed IPA-CLIP model.

Furthermore, retraining the original model could be another drawback since it requires both a huge amount of data and huge computational costs.

To tackle these problems, we first integrate a phonetic prior into a general character-level embedding. The proposed IPA-based phoneme embedding exploits the International Phonetic Alphabet (IPA) chart (International Phonetic Association, 1999) as a phonetic prior. The chart defines phonetic similarity among phonemes in spoken languages (e.g., the English consonant /k/ is more similar to /g/ than /m/). Next, we take a distillation approach to extend a V&L pretrained model to accept language inputs written with phonemes. Specifically, we implement IPA-CLIP as illustrated in Fig. 1, a model which extends CLIP. It consists of three encoders: the original CLIP image and text encoders, and a newly trained pronunciation encoder. The input of the pronunciation encoder is an array of phonemes written with IPA phonetic symbols (e.g., /ə ˈfoʊˌtoʊ əv ə kæt./ for "A photo of a cat."). This allows the IPA-based phoneme embedding in the encoder to phonetically process each phoneme. The distillation approach reduces the cost of extending CLIP to a new pronunciation modality. Moreover, since the pronunciation encoder maps pronunciations onto the CLIP bi-modal embedding space, applications using CLIP will be able to accept pronunciation inputs, even if the target languages lack orthography, just by replacing encoders.

The contributions of this paper are four-fold: (1) We propose an IPA-based phoneme embedding which integrates phonetic similarity on the IPA chart into its phoneme embedding space, (2) We implement IPA-CLIP by extending CLIP, to phonetically process pronunciation inputs using the IPA-based phoneme embedding, (3) We confirm the agreement of the IPA-based phoneme embedding with the phonetic relationships on the IPA chart, and (4) We verify the general ability of IPA-CLIP in multimodal retrieval tasks when existing or nonsense words are input as well as its agreement with human perception.

2 Related Work

2.1 Phonetics and Computational Approaches

IPA transcription is one of the most common alphabetic systems used to describe pronunciation. It assigns a unique symbol to each phoneme while also providing symbols for other phonetic components such as stresses and syllable boundaries. The IPA chart denotes the relationships among phonemes that can appear in spoken languages (See Appendix A.1 for the whole IPA chart), assigning each phoneme with multiple phonetic attributes. Consonants have three attributes: *voicing*, *place of articulation*, and *manner of articulation*. Vowels also have three attributes: *height*, *backness*, and *roundedness*. For example, the voiceless velar plosive /k/, as in "***C**oat*", possesses "voiceless" (*voicing*), "velar" (*place*), and "plosive" (*manner*) attributes, and the close-mid back rounded vowel /o/, as in "***Co**at*", possesses "close-mid" (*height*), "back" (*backness*), and "rounded" (*roundedness*) attributes.

Several studies integrate such phonetic knowledge into the calculation of the phonetic similarity between words. Vitz and Winkler (1973) propose a dissimilarity measure between two word pronunciations based on the edit distance. Hahn and Bailey (2005) incorporate phonetic features into the edit distance, regarding two English phonemes sharing certain attributes (e.g., /k/ and /g/) as closer than other pairs of phonemes sharing fewer attributes (e.g., /k/ and /m/). Parrish (2017) proposes a bigram model based on phonetic features for poetic applications. Bay et al. (2017) use the structure of the IPA chart to calculate the phonetic similarity for text transformation. They regard all three consonant attributes, as described above, as categorical, and *height* and *backness* of the vowel attributes

as continuous. When calculating the similarity between consonants, they check and count which out of three attributes two consonants have in common. For vowels, they manually reconstruct the vowel chart on a 2D Cartesian plane (they ignore *roundedness*) and measure the Euclidean distance.

Recent Natural Language Processing (NLP) techniques also obtain neural phoneme embeddings that reflect phonetic relationships without explicit supervision. Kolachina and Magyar (2019) confirm if phoneme-level Word2vec (Mikolov et al., 2013a,b) learns the phoneme relationships, concluding that Word2vec captures them from the phonological restrictions in the training data quite well. Boldsen et al. (2022) perform a similar analysis of character embeddings in multiple languages using some language models, showing strong correlations between the learned character relationships and actual phonetic relationships.

For constructing the IPA-based phoneme embedding, this paper follows the usage of the IPA chart by Bay et al. (2017) and treats two vowel attributes as continuous while other attributes as categorical. We also compare the IPA-based embedding with a neural phoneme embedding obtained via training without such a prior as a baseline.

2.2 CLIP Extensions for Other Types of Data

Many methods extend CLIP to other modalities to spread its effectiveness in other multimodal tasks. For the audio, some train a new audio encoder in addition to the original image and text encoders using multimodal datasets. Guzhov et al. (2022) train three encoders for each modality simultaneously using uni- and multimodal datasets. Wu et al. (2022) distill the CLIP image encoder to train only an additional audio encoder with an audio-image dataset. Elizalde et al. (2022) use a text-audio dataset to train audio and text encoders from scratch. All these methods employ contrastive learning for training like the original CLIP.

Within the image and text modalities, Carlsson et al. (2022) expand the CLIP text encoder, which was trained mainly on the English vocabulary, to process multiple languages. They first prepare a number of English sentences and then machine-translate them into multiple languages to obtain a multilingual dataset. Using this, similar to Wu et al. (2022), they train a multilingual text encoder with multilingual sentence pairs by distilling the CLIP text encoder using Mean Squared Error (MSE) loss.

The proposed IPA-CLIP extends CLIP for pronunciation inputs. To this end, we take a similar distillation approach as proposed by Carlsson et al. (2022) to reduce the costs of training a pronunciation encoder compatible with the CLIP encoders. We automatically convert English sentences into IPA phonetic transcriptions using dictionaries and then use the text-pronunciation pairs in place of their multilingual sentence pairs.

3 IPA-CLIP: Phonetic Embedding Distillation of CLIP

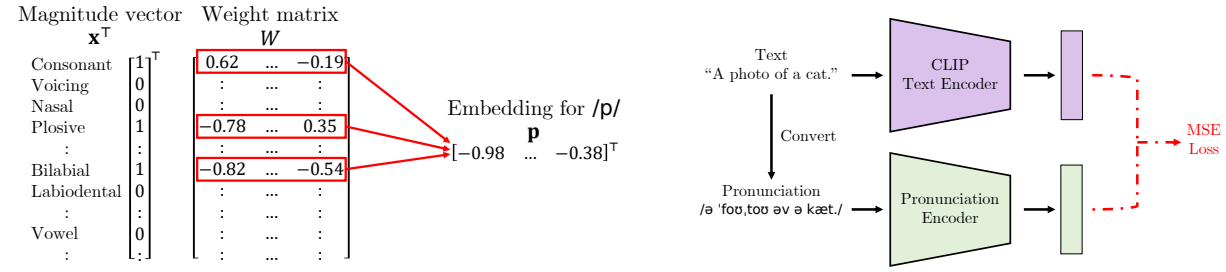
The overview of the proposed IPA-CLIP is illustrated in Fig. 1. It consists of three encoders: The CLIP image encoder, the CLIP text encoder, and a new pronunciation encoder, all of which share the same multimodal embedding space.

3.1 IPA-based Phoneme Embedding

This section proposes a phoneme embedding that considers the phoneme relationships on the IPA chart. This phoneme embedding layer works by replacing the word embedding layer of language models including BERT. As it is based on the IPA chart, the pronunciation input to this layer is theoretically universal and not specific to any language.

As mentioned in Section 2.1, the IPA chart assigns three attributes for each phoneme. Inspired by previous work (Bay et al., 2017), we treat the two vowel attributes, *height* and *backness*, as continuous and thus consider the extent of the difference between these attributes. For instance, the **close** front unrounded vowel /i/ is treated as more similar to the **close-mid** front unrounded vowel /e/ than the **open** front unrounded vowel /a/. In contrast, the other four attributes are regarded as categorical and thus we only consider whether two phonemes have any attribute in common.

As shown in Fig. 2a, the proposed method calculates the phoneme embedding $\mathbf{p}$ as a linear combination $\sum_i x_i \mathbf{w}_i$, where x_i is a magnitude and $\mathbf{w}_i$ is a feature vector for the i-th attribute. In detail, for each phoneme, we calculate the multiplication of the transpose of the N-dimensional sparse magnitude vector $\mathbf{x}$ and the $N \times D$ feature matrix W, which is written as $\mathbf{p} = \mathbf{x}^\top W$. A magnitude vector $\mathbf{x}$ also includes attributes for letters other than phonemes such as stresses, spaces, commas, and exclamation marks, which are also projected onto the same phoneme embedding space despite not being phonemes. The aim of this is to ensure the equiv-



(a) Calculation of the proposed IPA-based phoneme embedding for the voiceless bilabial plosive /p/.

(b) Distillation procedure.

Figure 2: Detailed illustration of the construction of the pronunciation encoder of IPA-CLIP.

alent flexibility of the input of the pronunciation encoder to the CLIP text encoder. Thus, the pronunciation encoder can differentiate between homophonic texts such as "everyday" vs. "every day" and "a cat" vs. "a cat!". More detailed examples of this calculation are available in Appendix A.2.

3.2 Training by Distilling CLIP Text Encoder

The training of the pronunciation encoder of IPA-CLIP is based on the distillation methods proposed by Carlsson et al. (2022) and Wu et al. (2022). Although the implementation of this paper focuses only on English, the distillation itself can be applied to other languages if resources are available.

The distillation procedure is illustrated in Fig. 2b. First, to create sentence-pronunciation pairs from a number of sentences, we convert each sentence to its pronunciation using an existing dictionary. Specifically, by looking up the dictionary, all words in a sentence are replaced with their pronunciations. For example, a sentence "a photo of a cat." is converted to its pronunciation /ə ˈfoʊˌtoʊ əv ə kæt./. We ignore cases in the Latin alphabet and do not exclude letters other than the Latin alphabet.

With this dataset, our pronunciation encoder is distilled from the CLIP text encoder, where the weights of the text encoder are frozen during the training. Given a sentence-pronunciation pair, the pronunciation encoder is trained to output the identical pronunciation embedding to the sentence embedding calculated by the text encoder. MSE loss is employed for the training objective as opposed to the contrastive loss used in training CLIP (Radford et al., 2021), because it is known to work better for the distillation purpose (Carlsson et al., 2022).

3.3 Implementation

DistilBERT (Sanh et al., 2019) is adopted as the architecture of the pronunciation encoder. We replace its word embedding with the proposed IPA-based phoneme embedding and add an additional linear layer to match the dimensionality of its output to that of the CLIP encoders, but we do not modify any other part. The pronunciation encoder is trained from scratch.

As training data, we use a dataset compiled by Carlsson et al. (2022), which is a mixture of sentences taken from some image captioning datasets. In addition, to increase the vocabulary, sentences consisting of only one word are also created using Spell Checker Oriented Word Lists (SCOWL)[1], an English wordlist that comprises 102,305 words. For text-to-pronunciation conversion, we use the Carnegie-Mellon University (CMU) Pronouncing Dictionary[2], which is also used by many previous studies (Parrish, 2017; Bay et al., 2017; Kolachina and Magyar, 2019), resulting in training data of 1,168,451 sentences in total. The pretrained CLIP model called ViT-L/14[3] is used throughout the evaluation. See Appendix A.3 for more details.

4 Quantitative Evaluations

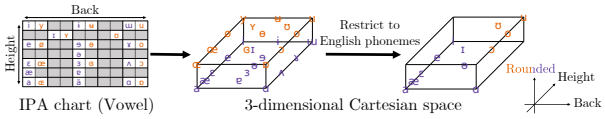
This section evaluates both the proposed IPA-based phoneme embedding and IPA-CLIP in a quantitative manner. With prior knowledge of phonetics, IPA-CLIP learns both phoneme embeddings and pronunciation embeddings through distillation.

A baseline method in these experiments employs a pronunciation encoder that uses an ordinary character-level (phoneme-level) embedding layer instead of the IPA-based one. Thus, its phoneme embedding does not consider phonetic relationships but implicitly learns such relationships only

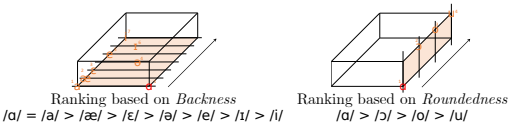
[1] http://wordlist.aspell.net/ (Accessed Jan. 19, 2023)

[2] https://github.com/menelik3/cmudict-ipa/ (Accessed Jan. 19, 2023)

[3] https://github.com/openai/CLIP/blob/main/model-card.md (Accessed Jan. 19, 2023)



(a) Conversion of the IPA chart to 3-dimensional Cartesian space.



(b) Ground-truth rankings of /ɑ/ on the *height* axis.

Figure 3: Core ideas of measuring the correlation of the vowel layout on a phoneme space to phonetic relationships.

from the phonological restrictions in the training data. We also test whether the weights of the feature matrix W of IPA-CLIP should be trainable or frozen as randomly initialized values. If frozen, the mapping of W becomes a random mapping (We call this setting ***Proposed (Frozen)***). If trainable, it reflects the phonetic relationships learned from the English phonological rules, which could be both a boon and a bane (We call this ***Proposed (Trainable)***). In any case, the weights of the DistilBERT and the additional linear layer are always trained.

Note that the experiments in the following sections measure performance only towards English phonemes for a fair comparison with the baseline.

4.1 Experiment on Phoneme Spaces

First, the proposed IPA-based embedding (proposed methods) is compared with the conventional phoneme-level embedding (baseline method). We measure the three characteristics of the learned phoneme embedding spaces with different metrics: (1) How distinct the distributions of the consonant cluster and the vowel cluster are, (2) How the consonant layout represents the phonetic relationships among consonants, and (3) How the vowel layout represents the phonetic relationships among vowels. The clear distinction between consonants and vowels makes IPA-CLIP easier to distinguish the two types. The accordance of the phoneme layouts with the IPA Chart helps IPA-CLIP to process phoneme differences based on phonetic similarity.

4.1.1 Consonant and Vowel Distributions

To measure the distinctness of consonants and vowels on the phoneme embedding space, we calculate the silhouette coefficient (Rousseeuw, 1987), a metric for a clustering technique, between the consonant and vowel clusters on the embedding space. A higher silhouette coefficient for a consonant (vowel) cluster means that the consonant (vowel) cluster is well apart from the vowel (consonant) cluster. After computing these values for all consonants (vowels), the silhouette coefficient for the consonant (vowel) cluster, s_{C_c} $(s_{C_v}) \in [-1, 1]$, is aggregated by averaging the values among all consonants (vowels). See Appendix A.4 for the mathematical formulation of this metric.

4.1.2 Consonant Distribution and Phonetics

To measure the consistency of the consonant layout on the phonetic space and the consonant categorization on the IPA chart, we calculate the mean Average Precision (mAP), a metric for retrieval tasks, for each consonant attribute. First, for each consonant, all other consonants are retrieved based on the Euclidean distance on the phonetic space. Then, to calculate the Average Precision (AP), we regard consonants that share the focused attribute as *relevant*. For instance, when the voiced consonant /b/ is evaluated in terms of *voicing* attribute, the set of its *relevant* consonants, R, then becomes a set of all voiced consonants containing e.g., /d/, /m/, and /g/. If the retrieved ranking for the consonant /b/ is [/p/,/d/,/t/,/m/,/g/, ...] in order, the AP score for /b/, $\mathrm{AP}_{/\mathrm{b}/}$, is

$$\mathrm{AP}_{/\mathrm{b}/} = \frac{1}{|R|}\left(\frac{1}{2} + \frac{2}{4} + \frac{3}{5} + \cdots\right). \quad (1)$$

The mAP metric for each attribute is calculated by averaging the AP scores among all consonants.

4.1.3 Vowel Distribution and Phonetics

Figure 3 illustrates the core idea of measuring the correlation between the vowel layout on the phonetic space and the vowel order on the IPA chart. First, inspired by Bay et al. (2017), we map every vowel onto the 3D Cartesian space that replicates the IPA chart. The three axes of the Cartesian space represent vowel attributes of *height*, *backness*, and *roundedness*, respectively. Next, for each attribute/axis, we calculate Spearman's rank correlation between the vowel distribution on this Cartesian space and that on the phonetic space. We create two ground-truth rankings for each attribute by sorting vowels that share one of the other two attributes. For instance, as illustrated in Fig. 3, when evaluating the vowel /ɑ/ on the *height* attribute, the following two rankings are calculated: (1) the ranking of the back vowels: /ɑ/ > /ɔ/ > /o/ > /u/,

Table 1: Quantitative evaluation of phoneme embedding spaces. Silhouette coefficient (Sil), mean Average Precision (mAP), and Spearman's rank correlation (RCorr) denote the distinctness between consonants and vowels, consistency of consonant distributions with phonetics, and correlation of vowel distributions to phonetics, respectively.

	Sil ↑		mAP ↑ (Consonant)			RCorr ↑ (Vowel)		
Method	s_{C_c}	s_{C_v}	Voicing	Place	Manner	Height	Back	Round
Baseline	−0.014	0.054	0.589	0.421	0.342	0.541	0.592	0.753
Proposed (Trainable)	−0.036	0.217	0.705	0.767	0.642	**0.891**	0.680	**0.925**
Proposed (Frozen)	**0.233**	**0.568**	**0.735**	**0.810**	**0.837**	0.890	**0.685**	**0.925**

and (2) the ranking of the unrounded vowels: /ɑ/ = /a/ > /æ/ > /ɛ/ > /ə/ > /e/ > /ɪ/ > /i/. With these ground-truth rankings, we calculate two rank correlations between each of the two ground-truth rankings and the ranking of the vowels retrieved based on the Euclidean distance on the phonetic space. Lastly, for each vowel attribute, we compute the average value among all vowels to be the rank correlation metric on the phonetic space.

4.1.4 Results and Discussions

The results of this experiment are shown in Table 1. Overall, *Proposed (Frozen)* performs best in almost all metrics. *Proposed (Trainable)* is also comparable except for the silhouette coefficient.

The great advantage of *Proposed (Frozen)* to the other methods is the silhouette coefficient. A high silhouette coefficient means that the distributions of the consonant and the vowel clusters are distinct and thus have little overlap on the phoneme embedding space. As the coefficient drops in the baseline and the *Proposed (Trainable)* methods, this comparison indicates that the embeddings learned from the phonological restrictions in sentences do not clearly distinguish consonants and vowels. This contradicts the fact that such neural embeddings are known to represent phonetic relationships quite well (Kolachina and Magyar, 2019; Boldsen et al., 2022). Yet, since even the baseline performs moderately in the other metrics, such embeddings seem to learn relationships within consonants and within vowels well even without explicit priors.

Moreover, between the baseline and the proposed methods, the increase of mAP and rank correlation is observed. This suggests the effectiveness of the proposed IPA-based embedding in differentiating both within consonants and within vowels.

4.2 Experiments on Pronunciation Spaces

Second, the performance of IPA-CLIP is discussed in the following three multimodal retrieval tasks: (1) Retrieval-based image classification from the pronunciations of existing words, (2) Image retrieval from the pronunciations of nonwords, and (3) Text retrieval from the pronunciations of nonwords. Note that the nonwords here denote such words that do not exist in the English vocabulary but sound similar to certain existing words. Measuring the performance on these tasks evaluates the accordance of the pronunciation encoder (1) with the image encoder for existing words, (2) with the image encoder for nonwords, and (3) with the text encoder for nonwords, respectively.

Here, the ImageNet (Deng et al., 2009) validation dataset is used as a source of image-text pairs. The dataset provides 50 images for each of its 1,000 classes. We convert each class label into its pronunciation in the same way as Section 3.3. By removing the classes where this conversion failed, we obtain 912 classes with 50 images each in total. Note that we use the class labels identical to the ones used by the authors of CLIP (Radford et al., 2021), which differ from the class labels that ImageNet provides. As the previously described related work does not consider multimodal retrieval tasks, this section only evaluates the original CLIP as a comparison method, as it outperforms other methods (Carlsson et al., 2022; Wu et al., 2022).

4.2.1 Image Classification from Pronunciation

The retrieval-based image classification task is similar to the one in a previous study (Radford et al., 2021). Here, IPA-CLIP classifies an image by measuring the cosine similarities between the embedding of the image and those of the class labels in the form of, e.g., /ə ˈfoʊˌtoʊ əv <CLASS>/ ("A photo of <CLASS>"). For example, given an image and two class labels "Dog" and "Cat", IPA-CLIP first calculates the embedding of the image and those of the two pronunciations /ə ˈfoʊˌtoʊ əv dɔg/ and /ə ˈfoʊˌtoʊ əv kæt/. It then measures the cosine similarity between the image embedding and each of the pronunciation embeddings. The class label of the image is determined by finding the image-

label pair that gives the maximum similarity. We also filter out classes by measuring the word frequency (as the Zipf scale, which we call *Zipf frequency*) of their labels using an existing Python package (Speer et al., 2018). This allows us to see how rare class labels, which would never appear or appear few in the distillation process of IPA-CLIP, affect the classification results. We compare the accuracy of our methods with CLIP, which classifies images from text labels using the text encoder. To see the modality gap between the IPA-CLIP pronunciation encoder and the CLIP text encoder, we also merge the two by taking the average of their embeddings on the joint embedding space, which we call "***Proposed (Frozen) + CLIP***".

4.2.2 Nonword-to-Image Retrieval

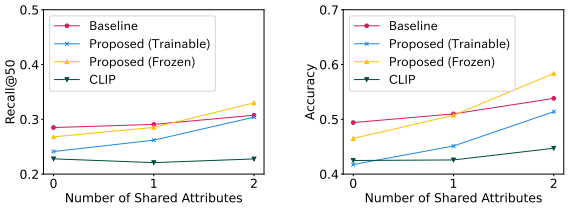
To evaluate the robustness of IPA-CLIP towards nonwords having certain similar-sounding existing words, a set of nonwords is prepared by slightly modifying the class labels of ImageNet. First, we focus only on the labels whose Zipf frequency is three or more (297 classes). Then, for labels starting with a sole consonant (216 classes satisfy this), the initial consonant is substituted with other consonants (e.g., from /dɛsk/: "Desk" to /zɛsk/, /nɛsk/, etc.) to make nonwords that sound similar to the original word. Next, by removing words that happen to exist in the English vocabulary, we obtain 3,530 nonwords stemming from any of the 216 classes. Meanwhile, text equivalents are also prepared by automatically converting each phoneme into its spelling ("Zesk" for /zɛsk/) to evaluate the text-based original CLIP. See Appendix A.6 for a more detailed procedure of this nonword creation.

With these nonwords, an image retrieval task is performed. Given a nonword, the objective is to retrieve the images belonging to the class from which the nonword stems. For instance, given the nonword /zɛsk/, we measure how many of the 50 images in the class "Desk" are retrieved from the pronunciation embedding of /ə ˈfoʊˌtoʊ əv zɛsk/.

Recall@50 is measured as a metric. We split the evaluation based on how phonetically similar the nonword is to its original word by counting the number of shared attributes between the two contrasting consonants. This assesses whether each method captures the phonetic similarity among consonants. Giving always similar scores regardless of the number of shared attributes mean that the method does not consider phonetic similarity, while a high correlation between the scores and the number of shared attributes indicates that the method associates nonwords with their similar-sounding words according to phonetic similarity.

Table 2: Accuracies of the image classification on 1,000-class ImageNet (Deng et al., 2009) dataset. We use Zipf frequency to filter out the classes having less frequent and rare label names.

Zipf Frequency	≥ 0.0	≥ 1.5	≥ 3.0	≥ 4.5
Number of Classes	912	492	297	29
Baseline	0.600	0.696	0.777	0.877
CLIP (Radford et al., 2021)	**0.712**	0.751	0.765	0.891
Proposed (Trainable)	0.581	0.686	0.769	0.886
Proposed (Frozen)	0.590	0.683	0.764	0.885
Proposed (Frozen) + CLIP (Radford et al., 2021)	0.705	**0.765**	**0.799**	**0.897**



(a) Nonword-to-Image (b) Nonword-to-Text

Figure 4: Results of (a) image retrieval and (b) text retrieval from nonwords written with either phonetic symbols (Baseline and Proposed) or texts (CLIP).

4.2.3 Nonword-to-Text Retrieval

The procedure of the nonword-to-text retrieval is similar to the one described in Section 4.2.2, but this experiment targets texts instead of images. We use 3,530 nonwords prepared in Section 4.2.2. In this experiment, models retrieve the text of the class from which the nonword stems. For example, given the nonword /zɛsk/, we assess whether each method can retrieve the text embedding of the text "A photo of desk" among the text embeddings of 216 classes. Accuracy is measured as a metric.

4.2.4 Results and Discussions

The results of the image classification are shown in Table 2. It indicates a strong effect of the rareness of the class labels on the performance. As can be seen on the left side, the proposed methods perform much worse than CLIP when the classes contain rare words. This is mainly because these models, as student models, have not been exposed much to these rare words during the distillation. In contrast, as the rare words drop out from the evaluation, their performance comes to be comparable.

Most interesting is "*Proposed (Frozen) + CLIP*". Despite its simple fusion strategy of the two modali-

ties, it performs best in almost all settings. This suggests the effectiveness of introducing the pronunciation modality into existing V&L pretrained models. Looking into the 297-class confusion matrix (See supplementary materials) revealed the characteristics of each encoder. The pronunciation encoder is more sensitive to pronunciation differences, while the text encoder is stronger against the meaning gaps. For example, *Proposed (Frozen)* misclassified "*Block plane*" as "*Buckle*" since they sound similar. In contrast, CLIP misclassified "*Screw*" as "*Metal Nail*" since their meanings are similar. As "*Proposed (Frozen) + CLIP*" correctly classified both, averaging the embeddings of the two encoders could have compensated for their weaknesses.

Next, Fig. 4 shows the results of the nonword-to-image and -text retrieval tasks. The tendency is similar throughout the two tasks: The baseline method retrieves best when the number of shared attributes is 0 or 1, while *Proposed (Frozen)* performs best when it is 2. This suggests that *Proposed (Frozen)* associates nonwords with the original word only when the words are phonetically similar, which confirms that *Proposed (Frozen)* considers the phonetic similarity between consonants more accurately than the other methods in the pronunciation embedding space.

We also observed that the proposed methods always outperform CLIP in these nonword-centered tasks. This verifies that the proposed pronunciation modality makes CLIP robust against nonwords.

5 Qualitative Evaluation

This section evaluates how much the proposed methods attune the CLIP embedding space to actual human perception regarding pronunciation similarity. We use the pronunciation similarity rankings collected by Vitz and Winkler (1973). In each of their four trials, native English speakers rated the sound similarity between a given target word and each of its 25 comparison words. The four trials differ only in the target word, which is "*Sit*", "*Plant*", "*Wonder*", and "*Relation*", respectively, as well as its comparison words. More details are explained in Appendix A.8. In our evaluation, given a target word, we first calculate the cosine similarity between the word and each of its comparison words on the pronunciation space to create a similarity ranking. Then, its rank correlation to the ground truth is measured as a metric. A higher value means that the embedding space better fits human perception regarding pronunciation similarity.

Table 3: Qualitative evaluation of the pronunciation embedding space. Scores denote rank correlations between the word similarity measured on the embedding space of each method and the ground truth rated by humans.

Target Word	Sit	Plant	Wonder	Relation
Baseline	0.535	0.397	**0.693**	0.442
Proposed (Trainable)	**0.642**	**0.549**	0.526	0.485
Proposed (Frozen)	0.385	0.420	0.640	**0.504**
CLIP (Radford et al., 2021)	0.353	0.402	0.585	0.304

Table 3 shows the results. All pronunciation-based methods outperform the text-based CLIP, which verifies that the phonetic prior forces similar-sounding words to become closer to each other. Within the pronunciation-based methods, the performance of *Proposed (Frozen)* is particularly bad when the target word is "*Sit*". This is due to the short syllable length of its comparison words, yielding much more possible similar-sounding words than the other target words. Thus, its phonetic knowledge could have disturbed the calculation of the embeddings of such short syllable words, which would be a shortcoming of the proposed approach. Nevertheless, since this evaluation covers just these four specific cases, the results do not spotlight which of the pronunciation-based methods works best in general.

6 Conclusion

We proposed an IPA-based phoneme embedding and IPA-CLIP which integrate the phonetic relationships on the IPA (International Phonetic Alphabet) chart into a character/phoneme-level embedding and the Vision and Language pretrained model CLIP. The phonetic prior enables it to process inputs even if they contain nonsense words (nonwords). Evaluations showed the effectiveness of the IPA-based phoneme embedding against conventional embeddings and the potential of IPA-CLIP to outperform the original CLIP in some multimodal retrieval tasks. When nonwords are input, IPA-CLIP performs always better than CLIP, which verifies its robustness against nonwords. Further evaluation verified the correlation between its pronunciation embedding space and human perception regarding pronunciation similarity.

For future work, further analysis is needed to investigate under which conditions the proposed approach has advantages over text-based methods.

Ethics Statement

We have evaluated ethics and social concerns in this research and believe there are only limited concerns.

We first hope that the proposed IPA-CLIP as well as the IPA-based phoneme embedding will be effectively used in pronunciation-related downstream tasks such as image-pronunciation matching, image captioning (image-to-pronunciation), and image generation (pronunciation-to-image). However, as they provide a method to relate nonwords with their phonetically similar words, this research could potentially impair the dignity of proper nouns including peoples' names, even though it is not our intended use. For instance, some might perceive unpleasant if their names are associated with such existing words that have negative and unpleasant meanings. This might also occur when IPA-CLIP is applied to multimodal downstream tasks if no modification is made to the implementation of this paper. One example is image generation, which can generate images having unpleasant content for the pronunciation of a name.

Second, since IPA-CLIP is based on OpenAI's pretrained CLIP models which are trained using data extensively collected from the Web, IPA-CLIP would inherit existing biases that those models already have. Also, in the current implementation, we do not consider dialects and other regional differences in pronunciations, which could have a minor impact on the use of our framework.

Finally, we declare that all data used in this paper are properly cited and used in accordance with their respective licenses.

References

Benjamin Bay, Paul Bodily, and Dan Ventura. 2017. Text transformation via constraints and word embedding. In *Proc. 8th Int. Conf. Comput. Creativity*. Atlanta, GA, USA, 49–56.

Sidsel Boldsen, Manex Agirrezabal, and Nora Hollenstein. 2022. Interpreting character embeddings with perceptual representations: The case of shape, sound, and color. In *Proc. 60th Annual Meeting Assoc. Comput. Linguist. (Volume 1: Long Papers)*. Dublin, Ireland, 6819–6836. https://doi.org/10.18653/v1/2022.acl-long.470

Fredrik Carlsson, Philipp Eisen, Faton Rekathati, and Magnus Sahlgren. 2022. Cross-lingual and multilingual CLIP. In *Proc. 13th Lang. Resour. Evaluation Conf.* Marseille, Bouches-du-Rhône, France, 6848–6854.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. VQGAN-CLIP: Open domain image generation and editing with natural language guidance. Comput. Res. Reposit., arXiv preprint, arXiv:2204.08583. https://doi.org/10.48550/arXiv.2204.08583

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A large-scale hierarchical image database. In *Proc. 2009 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* Miami, FL, USA, 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Hicham El Boukkouri, Olivier Ferret, Thomas Lavergne, Hiroshi Noji, Pierre Zweigenbaum, and Jun'ichi Tsujii. 2020. CharacterBERT: Reconciling ELMo and BERT for word-level open-vocabulary representations from characters. In *Proc. 28th Int. Conf. Comput. Linguist.* Barcelona, Cataluña, Spain, 6903–6915. https://doi.org/10.18653/v1/2020.coling-main.609

Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2022. CLAP: Learning audio concepts from natural language supervision. Comput. Res. Reposit., arXiv preprint, arXiv:2206.04769. https://doi.org/10.48550/arxiv.2206.04769

Federico Galatolo, Mario Cimino, and Gigliola Vaglini. 2021. Generating images from caption and vice versa via CLIP-guided generative latent space search. In *Proc. Int. Conf. Image Process. Vis. Eng.* Prague, Czech, 166–174. https://doi.org/10.5220/0010503701660174

Andrey Guzhov, Federico Raue, Jörn Hees, and Andreas Dengel. 2022. AudioCLIP: Extending CLIP to image, text and audio. In *Proc. 2022 IEEE Int. Conf. Acoust. Speech Signal Process.* Singapore, 976–980. https://doi.org/10.1109/ICASSP43922.2022.9747631

Ulrike Hahn and Todd M. Bailey. 2005. What makes words sound similar? *Cognition* 97, 3 (2005), 227–267. https://doi.org/10.1016/j.cognition.2004.09.006

International Phonetic Association. 1999. *Handbook of the International Phonetic Association: A Guide to the Use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, England, UK.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proc. 3rd Int. Conf. Learn. Represent.* San Diego, CA, USA, 15 pages.

Sudheer Kolachina and Lilla Magyar. 2019. What do phone embeddings learn about Phonology?. In *Proc. 16th Workshop Comput. Res. Phonetics, Phonol., Morphol.* Firenze, Toscana, Italy, 160–169. https://doi.org/10.18653/v1/W19-4219

Wentao Ma, Yiming Cui, Chenglei Si, Ting Liu, Shijin Wang, and Guoping Hu. 2020. CharBERT: Character-aware pre-trained language model. In *Proc. 28th Int. Conf. Comput. Linguist.* Barcelona, Cataluña, Spain, 39–50. https://doi.org/10.18653/v1/2020.coling-main.4

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. Comput. Res. Reposit., arXiv preprint, arXiv:1301.3781. https://doi.org/10.48550/arXiv.1301.3781

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Adv. Neural Inf. Process. Syst.*, Vol. 26. Lake Tahoe, NV, USA, 3111–3119.

Allison Parrish. 2017. Poetic sound similarity vectors using phonetic features. In *Proc. 13th AAAI Conf. Artif. Intell. Interact. Digital Entertain.*, Vol. 13, 2. Snowbird, UT, USA, 99–106. https://doi.org/10.1609/aiide.v13i2.12971

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proc. 38th Int. Conf. Mach. Learn., Proc. Mach. Learn. Res.*, Vol. 139. Online, 8748–8763.

Peter J. Rousseeuw. 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Applied Math.* 20 (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. Comput. Res. Reposit., arXiv preprint, arXiv:1910.01108. https://doi.org/10.48550/arXiv.1910.01108

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proc. 54th Annual Meet. Assoc. Comput. Linguist.*, Vol. 1. Berlin, Germany, 1715–1725. https://doi.org/10.18653/v1/P16-1162

Hengcan Shi, Munawar Hayat, Yicheng Wu, and Jianfei Cai. 2022. ProposalCLIP: Unsupervised open-category object proposal generation via exploiting CLIP cues. In *Proc. 2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.* New Orleans, LA, USA, 9611–9620.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. LuminosoInsight/wordfreq: v2.2. https://doi.org/10.5281/zenodo.1443582

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *J. Mach. Learn. Res.* 9, 86 (2008), 2579–2605.

Paul C. Vitz and Brenda Spiegel Winkler. 1973. Predicting the judged "similarity of sound" of English words. *J. Verbal Learn. Verbal Behav.* 12, 4 (1973), 373–388. https://doi.org/10.1016/S0022-5371(73)80016-7

Ho-Hsiang Wu, Prem Seetharaman, Kundan Kumar, and Juan Pablo Bello. 2022. Wav2CLIP: Learning robust audio representations from CLIP. In *Proc. 2022 IEEE Int. Conf. Acoust. Speech Signal Process.* Singapore, 4563–4567. https://doi.org/10.1109/ICASSP43922.2022.9747669

A Appendix

A.1 International Phonetic Alphabet Chart

Figure 5 shows the International Phonetic Alphabet (IPA) chart (International Phonetic Association, 1999) used in this paper. The chart connects almost all phonemes that can appear in any natural language and the proximity on it indicates phonetic similarities. On the chart, each phoneme is characterized by multiple phonetic attributes. Consonants have three attributes: *voicing*, *place of articulation*, and *manner of articulation*. Vowels also have three attributes: *height*, *backness*, and *roundedness*. According to the chart, for example, the voiceless velar plosive /k/, as in "***C**oat*", possesses "voiceless" (*voicing*), "velar" (*place*), and "plosive" (*manner*) consonant attributes, and the close-mid back rounded vowel /o/, as in "***Co**at*", possesses "close-mid" (*height*), "back" (*backness*), and "rounded" (*roundedness*) vowel attributes. Some consonants such as the voiced labial–velar approximant /w/ have multiple places of articulation. In this case, /w/ is characterized by four consonant attributes ("voiced", "labial", "velar", and "approximant").

A.2 IPA-based Phoneme Embedding

This section describes the details of the calculation of the proposed IPA-based phoneme embedding. As shown in Fig. 2a, the proposed method calculates the phoneme embedding $\mathbf{p}$ as a linear combination $\sum_i x_i \mathbf{w}_i$, where x_i is a magnitude and $\mathbf{w}_i$ is a feature vector for the i-th attribute. Specifically, for each phoneme, the proposed IPA-based phoneme embedding calculates the multiplication of the transpose of the N-dimensional sparse magnitude vector $\mathbf{x}$ and the $N \times D$ feature matrix W,

	Bilabial		Labio-dental		Dental		Alveolar		Post-alveolar		Palatal		Velar		Glottal	
Nasal		m		ɱ				n				ɲ		ŋ		
Plosive	p	b					t	d			c	ɟ	k	g	ʔ	
Sibilant affricate							ts	dz	tʃ	dʒ	tɕ	dʑ				
Sibilant fricative							s	z	ʃ	ʒ	ɕ	ʑ				
Nonsibilant fricative	ɸ	β	f	v	θ	ð					ç	ʝ	x	ɣ	h	ɦ
Approximant				ʋ				ɹ				j		ɰ		
Lateral approximant								l				ʎ		ʟ		

Symbols to the left in each column are voiceless consonants, symbols to the right are voiced consonants

(a) Consonants (Pulmonic)

	Front		Near-front		Central		Near-back		Back	
Close	i	y			ɨ	ʉ			ɯ	u
Near-close			ɪ	ʏ				ʊ		
Close-mid	e	ø			ɘ	ɵ			ɤ	o
Mid					ə					
Open-mid	ɛ	œ			ɜ	ɞ			ʌ	ɔ
Near-open	æ				ɐ					
Open	a	ɶ			ä				ɑ	ɒ

Symbols to the left in each column are unrounded vowels, symbols to the right are rounded vowels

(b) Vowels

Figure 5: IPA Chart (International Phonetic Association, 1999) for pulmonic consonants and vowels used in this paper. It connects almost all phonemes occurring in natural languages regarding their phonetic relationships. English phonemes, as used in this paper, are colored in red.

Table 4: Examples of attributes that each of the dimensions of the magnitude vector $\mathbf{x}$ represents.

Attribute	Category	Range	Examples of $\mathbf{x}$ /p/	/v/	/e/	/ʊ/	','
Consonant	Consonant	$x_i \in \{0,1\}$	1	1	0	0	0
Voicing		$x_i \in \{0,1\}$	0	1	0	0	0
Manner 1: Nasal		$x_i \in \{0,1\}$	0	0	0	0	0
Manner 2: Plosive		$x_i \in \{0,1\}$	1	0	0	0	0
⋮		⋮	⋮	⋮	⋮	⋮	⋮
Place 1: Bilabial		$x_i \in \{0,1\}$	1	0	0	0	0
Place 2: Labiodental		$x_i \in \{0,1\}$	0	1	0	0	0
⋮		⋮	⋮	⋮	⋮	⋮	⋮
Vowel	Vowel	$x_i \in \{0,1\}$	0	0	1	1	0
Height		$0 \leq x_i \leq 1$	0	0	$\frac{2}{6}$	$\frac{1}{6}$	0
Backness		$0 \leq x_i \leq 1$	0	0	0	$\frac{2}{4}$	0
Roundedness		$x_i \in \{0,1\}$	0	0	0	1	0
Primary stress /ˈ/	Others	$x_i \in \{0,1\}$	0	0	0	0	0
Secondary stress /ˌ/		$x_i \in \{0,1\}$	0	0	0	0	0
Char ' ': Space		$x_i \in \{0,1\}$	0	0	0	0	0
Char ',': Comma		$x_i \in \{0,1\}$	0	0	0	0	1
Char '!': Exclamation		$x_i \in \{0,1\}$	0	0	0	0	0
⋮		⋮	⋮	⋮	⋮	⋮	⋮

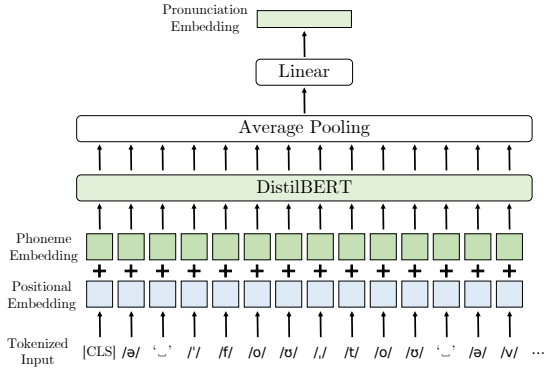


Figure 6: Illustration of the pronunciation encoder used in IPA-CLIP. IPA-CLIP employs the proposed IPA-based phoneme embedding in its phoneme embedding layer.

written as

$$\mathbf{p} = \mathbf{x}^\top W = \sum_{i=1}^{N} x_i \mathbf{w}_i. \quad (2)$$
$$= x_1\mathbf{w}_1 + x_2\mathbf{w}_2 + \cdots + x_N\mathbf{w}_N.$$

Since $\mathbf{x}$ is sparse, only the feature vectors where x_i is non-zero are summed.

Table 4 shows examples of the N attributes and magnitudes in the vector $\mathbf{x}$ for some phonemes. As shown in the table, $\mathbf{x}$ also includes attributes for letters other than phonemes such as stresses, spaces, commas, and exclamation marks. In the proposed method, we also project these letters onto the same phoneme embedding space despite not being phonemes. The aim of this is to ensure the equivalent flexibility of the input of the pronunciation encoder to the CLIP text encoder (Radford et al., 2021). Thus, the pronunciation encoder can differentiate between homophonic texts such as "everyday" vs. "every day" and "a cat" vs. "a cat!".

A.3 Details of Implementation and Data Preparation

As the architecture of the pronunciation encoder, we adopt DistilBERT (Sanh et al., 2019), a light and efficient version of BERT. Figure 6 illustrates the implementation. We replace its word embedding with the proposed IPA-based phoneme embedding and add an additional linear layer to match the dimensionality of its output to that of the CLIP encoders (Radford et al., 2021), but we do not modify any other part of DistilBERT. The pronunciation encoder is trained from scratch.

For the training data to distill the original CLIP models, we use a list of English sentences compiled by Carlsson et al. (Carlsson et al., 2022). It is a mixture of sentences taken from several image caption datasets, which could be strongly linked with the visual domain. In addition, to increase the vocabulary, we prepare sentences consisting of only one word using Spell Checker Oriented Word Lists (SCOWL)[1], an English wordlist that comprises 102,305 words. We convert these sentences

into pronunciation written with IPA symbols using the Python package eng-to-ipa[4]. The package uses the Carnegie-Mellon University (CMU) Pronouncing Dictionary[2], which is also used by many pieces of previous work (Parrish, 2017; Bay et al., 2017; Kolachina and Magyar, 2019). We remove sentences containing words whose pronunciations are not provided in the package. This results in training data of 1,168,451 sentences in total. Following the implementation of the previous work (Carlsson et al., 2022), we fix the size of the validation split as 1,000, resulting in a split of 1,167,451 sentences for training and 1,000 sentences for validation.

Although the main part of this paper discusses IPA-CLIP distilled from the pretrained CLIP model called ViT-L/14, we also test on another base model called ViT-B/32 (The results will be described in Appendix A.5). ViT-B/32 is the simplest and lightest model, while ViT-L/14 is a more recently released and larger model on the OpenAI's model card[3]. These models employ Transformers for both image and text encoders. We train our pronunciation encoder with a learning rate 5×10^{-5}, a batch size 32, and the Adam optimizer (Kingma and Ba, 2015), up to 50 epochs. Training a model took four days using a single NVIDIA RTX A6000 GPU. This paper reports results calculated using models trained only once for each setting.

A.4 Silhouette Coefficient among Consonant and Vowel Clusters

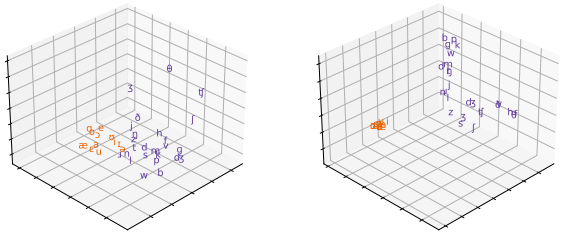
This section first describes the mathematical formulation of the calculation of the silhouette coefficient metric in Section 4.1.1. The silhouette coefficient (Rousseeuw, 1987) measures the distinctness of the consonant and vowel clusters on the embedding space. Given that C_c (respectively C_v) is a set of consonants (vowels), c is an element of C_c, and x_c is the embedding vector of c, the coefficient s_c for the consonant c is calculated as

$$s_c = \frac{b_c - a_c}{\max(a_c, b_c)}, \tag{3}$$

where

$$\begin{aligned} a_c &= \frac{1}{|C_c| - 1} \sum_{\hat{c} \in C_c, c \neq \hat{c}} d(x_c, x_{\hat{c}}), \\ b_c &= \frac{1}{|C_v|} \sum_{v \in C_v} d(x_c, x_v), \end{aligned} \tag{4}$$

[4] https://pypi.org/project/eng-to-ipa/ (Accessed Jan. 19, 2023)



(a) Baseline (b) Proposed (Frozen)

Figure 7: Visualization of the phoneme embedding spaces. Consonants and vowels are shown in orange and purple, respectively, to compare their distributions.

and $d(\cdot, \cdot)$ is the Euclidean distance. The silhouette coefficient for the consonant cluster, s_{C_c}, is then calculated by averaging the coefficients among all consonants. The coefficient for the vowel cluster, s_{C_v}, is also achieved in the same way, by swapping c and v in the equations. s_{C_c} (respectively s_{C_v}) ranges between $[-1, 1]$, where a high value indicates that the consonant (vowel) cluster is well distinct from the vowel (consonant) cluster.

Figure 7 shows the actual distribution of the two clusters on the phoneme spaces calculated by the baseline and the *Proposed (Frozen)* methods. The scatter plot shows the distribution of all consonants and vowels on the three-dimensional spaces suppressed by Principal Component Analysis (PCA). Both s_{C_c} and s_{C_v} of *Proposed (Frozen)* will be higher than those of the baseline method (See Table 1 and Table 5) because the consonant and vowel clusters are more distinct in the phoneme space of Figure 7b than that of Figure 7a.

A.5 Results for Different CLIP Models

Table 5 shows the results of our quantitative evaluation of the phoneme embedding spaces with different pretrained CLIP models (Radford et al., 2021), along with the validation loss at the point of 50 epochs. This table covers all the results shown in Table 1. For all metrics, we confirmed no significant difference in performance among the choice of the base models.

A.6 Nonword Creation

This section explains the more detailed procedure of the nonword creation in Section 4.2.2 and Section 4.2.3. In both sections, we use the same set of nonwords created by slightly modifying the 1,000 class labels of ImageNet (Deng et al., 2009).

To create nonwords that sound similar to certain common existing words, we first focus only on the

Table 5: Quantitative evaluation of the phoneme embedding spaces with different pretrained CLIP models (Radford et al., 2021). The silhouette coefficient (Silhouette), the mean Average Precision (mAP), and Spearman's rank correlation (Rank Corr.) denote the distinctness between consonants and vowels, consistency of consonant distributions with phonetics, and correlation of vowel distributions to phonetics, respectively.

		Silhouette ↑		mAP ↑ (Consonant)			Rank Corr. ↑ (Vowel)			Loss ↓
Base	Method	s_{C_c}	s_{C_v}	Voicing	Place	Manner	Height	Back	Round	MSE
ViT-B/32	Baseline	−0.006	0.049	0.585	0.433	0.394	0.524	0.574	0.796	**0.0084**
	Proposed (Trainable)	0.036	0.193	**0.753**	0.763	0.717	**0.913**	0.666	0.882	0.0096
	Proposed (Frozen)	**0.252**	**0.558**	0.735	**0.812**	**0.845**	0.889	**0.688**	**0.925**	0.0092
ViT-L/14	Baseline	−0.014	0.054	0.589	0.421	0.342	0.541	0.592	0.753	**0.027**
	Proposed (Trainable)	−0.036	0.217	0.705	0.767	0.642	**0.891**	0.680	**0.925**	0.028
	Proposed (Frozen)	**0.233**	**0.568**	**0.735**	**0.810**	**0.837**	0.890	**0.685**	**0.925**	0.028
	Upper Bound	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	—

Table 6: Candidate consonants and corresponding spellings used to generate nonwords from the class labels of ImageNet (Deng et al., 2009).

Consonant	/s/	/n/	/f/	/l/	/z/	/b/	/ɹ/	/p/	/g/	/k/	/d/	/m/	/θ/	/t/	/ʤ/	/j/	/h/	/v/	/ʃ/	/ʧ/	/w/
Spelling	's'	'n'	'f'	'l'	'z'	'b'	'r'	'p'	'g'	'k' or 'c'	'd'	'm'	'th'	't'	'j'	'y'	'h'	'v'	'sh'	'ch'	'w'

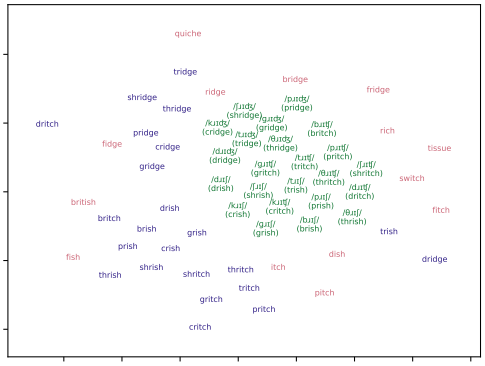


Figure 8: t-SNE (van der Maaten and Hinton, 2008) visualization of the text and pronunciation embeddings of nonwords. Text embeddings of existing words (red) and nonwords (purple) are calculated by CLIP (Radford et al., 2021), while pronunciation embeddings of nonwords (green) are calculated by IPA-CLIP.

labels whose Zipf frequency is three or more (297 classes) calculated using an existing Python package (Speer et al., 2018). Then, for labels starting with a sole consonant (216 classes satisfy this), we substitute the initial consonant with other possible consonants (e.g., from /dɛsk/: "Desk" to /zɛsk/, /nɛsk/, etc.) to make a set of nonwords which sound similar to the original word. Next, we remove the generated words that happen to exist in the English vocabulary. To check this, we use the SCOWL wordlist[1] and the CMU dictionary[2]. This process yields 3,530 nonwords stemming from either of the 216 classes.

During this preparation, we also prepare the text equivalents by automatically converting each phoneme into its spelling ("Zesk" for /zɛsk/) so that we can also evaluate the text-based original CLIP. Table 6 lists all consonants used for the substitution along with their spelling correspondents. As shown in the table, the candidate consonants are selected from all consonants appearing at the beginning of English words, except for /ð/, which becomes identical to /θ/ when spelled.

A.7 Visualization of Text and Pronunciation Embedding Spaces

As an additional analysis of the difference between CLIP (Radford et al., 2021) and IPA-CLIP, we visualize how words and nonwords are distributed on their shared embedding space. The scatter plot, shown in Fig. 8, illustrates the embeddings of existing words and nonwords sounding similar to "*Bridge*", calculated by either CLIP or IPA-CLIP. It reveals that IPA-CLIP places nonwords such as "*Pridge*" (/pɹɪʤ/) and "*Britch*" (/bɹɪʧ/) in positions close to their similar-sounding existing word "*Bridge*" (/bɹɪʤ/). In contrast, CLIP does not place any nonwords, even "*Pridge*", near "*Bridge*". This supports the results that IPA-CLIP considers the phonetic similarity of words.

A.8 Details of Qualitative Evaluation

This section explains the details of the pronunciation similarity rankings collected through four trials of psychological experiments conducted by Vitz and Winkler (Vitz and Winkler, 1973). In each of their four trials, native American English speakers rated the pure sound similarity between

a given target word and each of its 25 comparison words. Their four experiments differ only in the target word, which is "*Sit*", "*Plant*", "*Wonder*", and "*Relation*", respectively, as well as its comparison words. In the first three experiments, comparison words are a mixture of valid and nonsense English words that have a similar syllable structure as the target word. In the last experiment, this constraint for the syllable structure is removed. For example, the comparison words for the target word "*Sit*" include "*Pit*", "*Sat*", and "*Tass*", all of which have the same syllable structure (Consonant+Vowel+Consonant) as "*Sit*". Meanwhile, those for the target word "*Relation*" include "*Belation*", "*Fascinating*", and "*Get*", which do not necessarily have the same syllable structure as "*Relation*".