Geolocation-Aware Robust Spoken Language Identification

Qingzheng Wang, Hye-jin Shim, Jiancheng Sun, and Shinji Watanabe
Carnegie Mellon University
qingzhew@andrew.cmu.edu, shimhz6.6@gmail.com, jianches@andrew.cmu.edu, shinjiw@ieee.org

***Abstract*—While Self-supervised Learning (SSL) has significantly improved Spoken Language Identification (LID), existing models often struggle to consistently classify dialects and accents of the same language as a unified class. To address this challenge, we propose geolocation-aware LID, a novel approach that incorporates language-level geolocation information into the SSL-based LID model. Specifically, we introduce geolocation prediction as an auxiliary task and inject the predicted vectors into intermediate representations as conditioning signals. This explicit conditioning encourages the model to learn more unified representations for dialectal and accented variations. Experiments across six multilingual datasets demonstrate that our approach improves robustness to intra-language variations and unseen domains, achieving new state-of-the-art accuracy on FLEURS (97.7%) and 9.7% relative improvement on ML-SUPERB 2.0 dialect set.**

***Index Terms*—spoken language identification, geolocation conditioning, dialect robustness, cross-domain generalization.**

I. Introduction

Spoken language identification (LID) is becoming increasingly essential as speech technology expands toward multilingual scalability. With the emergence of speech foundation models trained on hundreds or even thousands of languages [1]–[6], accurately identifying the language of an utterance has become a critical first step in both dataset curation pipelines and runtime systems. For instance, LID enables language-aware automatic speech recognition (ASR) by routing input to the appropriate language-specific module [1], and supports large-scale multilingual dataset construction through filtering and annotation [7]–[10].

Recent advances in self-supervised learning (SSL) have improved the robustness and cross-lingual transferability of speech representations, which can be fine-tuned for LID with high accuracy [1], [2], [11]. Prior studies have shown that SSL models predominantly capture phonetic representations [12], [13], making them particularly effective for distinguishing languages with distinct sound patterns.

However, dialects and accents within the same language often differ significantly in phonetic representations, which can lead to misclassifications of these intra-language variations as another language. For instance, English encompasses a wide range of regional dialects and accents, such as American and Indian English, which differ phonetically despite sharing the same language identity. One potential solution is to assign fine-grained dialect or accent labels for classification, but it is incompatible with most downstream tasks such as ASR and speech translation, which operate at the language level and expect to generalize across dialectal and accented variations.

TABLE I
Accuracy (%) with joint prediction of language ID and meta features from lang2vec [14]. Orange /**Bold**: best overall.

Meta Info	ML-SUPERB 2.0	
	Dev	Dialect
LID-only	89.0	73.4
Geolocation	**89.5**	**73.8**
Inventory	88.8	68.2
Phonology	88.8	73.6
Syntax	88.9	67.2

To address this challenge, we explore using language-level meta information as auxiliary supervision to guide the model to learn unified representations for dialectal and accented variations. Among several candidates, including geolocation, phonology, phonetic inventory, and syntax, we compare their effectiveness by predicting each as an auxiliary task jointly with LID. Our preliminary results (Table I) with the ML-SUPERB 2.0 [15] show that geolocation provides the most consistent improvement, suggesting that it can serve as a strong signal to unify intra-language variations.

Motivated by this finding, we propose geolocation-aware LID, a novel framework that incorporates language-level geolocation information into SSL-based LID models. Specifically, we introduce geolocation prediction as an auxiliary task at both intermediate layers of the SSL encoder and the downstream embedding extractor. Predicted geolocation vectors from intermediate layers are injected into subsequent layers as conditioning signals, encouraging the model to develop more compact and consistent representations for dialectal and accented speech within the same language.

Our key contributions are as follows: (i) we propose geolocation-aware LID, a new approach that incorporates geolocation prediction and conditioning into the SSL-based LID model; (ii) we empirically demonstrate the effectiveness of language-level geolocation signals in improving robustness to intra-language variations; (iii) we develop a robust LID system supporting 157 languages, achieving **new state-of-the-art (SOTA)** accuracy with relative improvements of 0.5% on FLEURS (97.7%) [16], and 2.0% and 9.7% on ML-SUPERB 2.0 [15] development (88.6%) and dialect (86.8%) set, respectively. Relevant code, model weights (including our

SOTA checkpoint), and training logs are publicly available.[1]

II. Related Studies

A. Geographic Information for LID and Speech Processing

The integration of geographic information into spoken language identification remains unexplored. Foley et al. [17] explored *utterance-level* speech geolocation prediction as a proxy task to improve LID, showing that geolocation-pretrained encoders yield better performance than directly fine-tuned SSL models. To our knowledge, this is the only work on using geolocation information for spoken language identification. In the field of textual language identification, Dunn et al. [18] showed similar benefits by incorporating geographic priors into region-specific LID models. More broadly, geographic information has been leveraged in ASR via geolocation vectors for dialect modeling [19] and location-aware language models for local vocabulary [20]. In this work, we extend this line of research by predicting *language-level* geolocation and injecting the predicted geolocation as conditioning information into SSL representations to improve spoken LID performance.

B. Intermediate Layer Prediction and Conditioning

Prediction at intermediate layers has proven effective for regularizing training in ASR models. For example, applying Connectionist Temporal Classification (CTC) loss to encoder layers [21], [22] and adding LID-aware CTC loss in SSL encoder layers [23] have been used. While auxiliary prediction tasks provide useful training signals, conditioning intermediate representations on these predictions allows subsequent layers to explicitly use these signals. For instance, self-conditioned CTC [24] conditioned final predictions on intermediate layer predictions to relax the conditional independence assumption. Chen et al. [25] extended this by conditioning intermediate layers on LID predictions to improve multilingual ASR performance. Beyond the ASR scope, Lu et al. [26] leveraged language- and speaker-specific information extracted from intermediate layers to adapt the pretrained SSL encoder. However, conditioning on geolocation information has not yet been explored. In this paper, we propose to condition SSL encoder intermediate layers on geolocation predictions.

III. Geolocation-Aware LID

To enhance robustness against dialectal and accented variations, we extend the SSL-based LID framework by incorporating geolocation information. As shown in Fig. 1, our architecture builds on the conventional SSL-based LID pipeline, consisting of a pretrained upstream SSL encoder, a downstream language embedding extractor, and a classification head. We introduce an auxiliary geolocation prediction task at both intermediate layers of the SSL encoder and the output of the embedding extractor. To enable the SSL encoder to directly utilize geolocation information, we inject intermediate-layer geolocation predictions as conditioning signals into subsequent encoder layers.

[1]https://github.com/espnet/espnet/tree/master/egs2/geolid/lid1

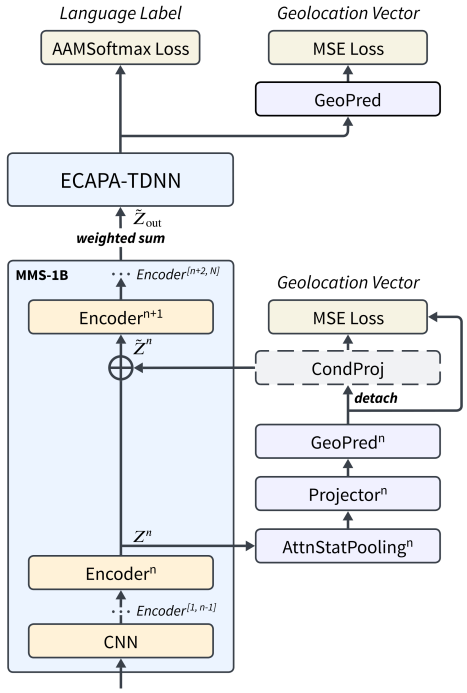

Fig. 1. Overview of the proposed geolocation-aware LID architecture. Geolocation vectors are predicted from a set of selected intermediate layers and the downstream embedding extractor. Intermediate predictions are detached and re-injected into the encoder via a conditioning projection module (dashed block), with design choices (shared vs. independent, frozen vs. trainable) depending on layer positions. A weighted sum of all hidden states of encoder layers is passed to ECAPA-TDNN for embedding extraction.

A. SSL-based LID Framework

In this section, we describe the core architecture of our SSL-based LID model (left side of Fig. 1). We use MMS-1B [1] as the SSL encoder, a 1B-parameter model based on wav2vec 2.0 [27] pretrained on over 1,400 languages (see large blue block in Fig. 1). The sub-branches extending from MMS-1B for geolocation conditioning will be described in Section III-D.

Given a raw audio input, the model first applies a convolutional waveform encoder (the CNN block in Fig. 1) that extracts a T-length sequence of D-dimensional acoustic features $X \in \mathbb{R}^{T\times D}$. These features are then processed by a stack of N-layer Transformer encoders [28] $\{\text{Encoder}^n\}_{n=1}^N$ (yellow blocks in Fig. 1; layers $\text{Encoder}^{[1,n-1]}$ and $\text{Encoder}^{[n+2,N]}$ are omitted for clarity):

$$Z^n = \text{Encoder}^n(Z^{n-1}), \tag{1}$$

where $Z^n = (\mathbf{z}_t^n \in \mathbb{R}^D | t = 1, \ldots, T)$ is the n-th layer output, with $Z^0 = X$. The final SSL encoder output Z_{out} is obtained through a *weighted sum* of all encoder hidden states [29], [30]:

$$Z_{\text{out}} = \sum_{n=0}^{N} \alpha^n Z^n, \tag{2}$$

where α are learnable parameters satisfying $\sum_{n=0}^N \alpha^n = 1$.

The aggregated SSL representation Z_{out} is then processed by ECAPA-TDNN [31], followed by MMS-1B in Fig. 1, to

extract language embeddings. This module processes frame-level features through a series of ECAPA blocks, which incorporate 1-D convolutional layers and squeeze-and-excitation Res2Blocks [32], [33]:

$$H = \text{ECAPA-Blocks}(Z_{\text{out}}), \tag{3}$$

where $H \in \mathbb{R}^{T' \times C}$ with C channels and T' frames after convolutions. Then, the frame-level features are aggregated using attentive statistics pooling [31], [34]:

$$\mathbf{s} = \text{AttnStatPooling}(H), \tag{4}$$

where $\mathbf{s} \in \mathbb{R}^{2C}$ contains the pooled mean and standard deviation statistics. Finally, the pooled statistics are projected to obtain the language embedding $\mathbf{e} \in \mathbb{R}^{E}$:

$$\mathbf{e} = \text{Projector}(\mathbf{s}), \tag{5}$$

where the projector includes batch normalization [35] followed by a linear transformation.

We adopt the AAMSoftmax [36] loss function enhanced with the sub-center technique [37], as implemented in ESPnet-SPK [38], to perform language classification (see the top of Fig. 1):

$$\mathcal{L}_{\text{class}} = \text{AAMSoftmax}(\mathbf{e}, y; K, m, s), \tag{6}$$

where y is the ground-truth language label, K is the number of sub-centers capturing intra-class variations, m is the angular margin, and s is the scaling factor.

B. Geolocation Vectors

To utilize geographic information, we use the geolocation vectors provided by the `lang2vec` project [14] to represent the abstract geolocation of each language. These vectors are derived from estimated geographic coordinates of languages, obtained from typological resources like Glottolog [39]. The coordinates of each language are transformed into vectors by computing normalized great-circle distances to 299 uniformly distributed reference points on Earth (generated via a spherical Fibonacci lattice [40]). The resulting 299-dimensional vectors with values between $[0, 1]$ provide a continuous and structured encoding that is well-suited for both prediction tasks and integration into high-dimensional hidden spaces.

C. Geolocation Prediction as an Auxiliary Task

To guide the model to learn language-discriminative representations, we incorporate an auxiliary geolocation prediction task into the fine-tuning process. Given a speech utterance in language l with ground-truth geolocation vector $\mathbf{v}_l$, we predict the geolocation vector from the language embedding $\mathbf{e}$ in (5):

$$\hat{\mathbf{v}}_l = \text{GeoPred}(\mathbf{e}), \tag{7}$$

where $\text{GeoPred}(\cdot)$ is a linear projection module (upper-right block in Fig. 1). The geolocation prediction loss is defined as:

$$\mathcal{L}_{\text{geo}} = \text{MSE}(\hat{\mathbf{v}}_l, \mathbf{v}_l), \tag{8}$$

where MSE denotes the mean squared error loss. We combine the classification loss in (6) and $\mathcal{L}_{\text{geo}}$ as:

$$\mathcal{L}_1 = (1 - \lambda)\mathcal{L}_{\text{class}} + \lambda\mathcal{L}_{\text{geo}}, \tag{9}$$

where $\lambda \in [0, 1]$ balances the classification and geolocation prediction objectives.

D. Conditioning the SSL Encoder on Geolocation Predictions

While geolocation prediction provides explicit supervision for LID, its output is not directly incorporated into the SSL representation. To enable the SSL encoder to explicitly use the geolocation information, we inject geolocation conditioning signals into the intermediate layers of the SSL encoder.

We select a subset of intermediate layers $\mathcal{M} \subseteq \{1, \ldots, N\}$ from the SSL encoder defined in (1). For each selected layer $n \in \mathcal{M}$, the frame-level hidden states Z^n, introduced in (1), are processed to obtain intermediate language embeddings and geolocation predictions:

$$\mathbf{e}^n = \text{Projector}^n(\text{AttnStatPooling}^n(Z^n)), \tag{10}$$

$$\hat{\mathbf{v}}_l^n = \text{GeoPred}^n(\mathbf{e}^n), \tag{11}$$

where all modules are layer-specific and correspond to the purple blocks in the right sub-branches of Fig. 1. Unlike $\mathbf{e}$ in (5) and $\hat{\mathbf{v}}_l$ in (7) extracted from the downstream module, $\mathbf{e}^n$ and $\hat{\mathbf{v}}_l^n$ capture distinct characteristics at each depth.

As each dimension of the geolocation vector encodes the distance to a fixed reference point, the geolocation vector is numerically sensitive: slight perturbations in its values can shift the implied geolocation. To prevent distortion by gradients from the downstream classification objective in (6), we detach the predicted geolocation vector into $\bar{\mathbf{v}}_l^n$ before projecting it into the conditioning signal $\mathbf{c}^n$:

$$\bar{\mathbf{v}}_l^n = \text{detach}(\hat{\mathbf{v}}_l^n), \tag{12}$$

$$\mathbf{c}^n = \text{CondProj}(\bar{\mathbf{v}}_l^n), \tag{13}$$

where CondProj is a linear layer (the dashed block in Fig. 1) and $\mathbf{c}^n \in \mathbb{R}^D$. This detachment only blocks the gradient from the CondProj layer; the original $\hat{\mathbf{v}}_l^n$ remains connected to the computational graph and is supervised by the intermediate-layer geolocation loss for layer n:

$$\mathcal{L}_{\text{geo}}^n = \text{MSE}(\hat{\mathbf{v}}_l^n, \mathbf{v}_l). \tag{14}$$

Therefore, the geolocation prediction modules in (10) and (11) are optimized only by the intermediate-layer geolocation objective. The effect of detachment will be shown in Section V-B.

As the sole interface between the geolocation predictions and the SSL encoder, the design of CondProj in (13) plays a crucial role in shaping how geolocation signals are represented and utilized. This module can be configured to be either shared or independent across layers, and either frozen or trainable during fine-tuning. Shared vs. independent controls whether the geolocation signal is tailored for each layer, while frozen vs. trainable determines whether it remains fixed or is adaptively modulated. As no configuration is universally optimal, we empirically evaluate these design choices in Section V-A.

The geolocation conditioning signal is then added to each frame of the hidden states (see the $\oplus$ operation in Fig. 1):

$$\tilde{\mathbf{z}}_t^n = \mathbf{z}_t^n + \mathbf{c}^n, \tag{15}$$

forming the conditioned representation $\tilde{Z}^n = (\tilde{\mathbf{z}}_t^n \in \mathbb{R}^D | t = 1, \ldots, T)$ that serves as input to the subsequent layer. With the conditioning signals injected into the selected layers $n \in \mathcal{M}$, the final SSL encoder output in (2) becomes:

$$\tilde{Z}_{\text{out}} = \sum_{n \notin \mathcal{M}} \alpha_n Z^n + \sum_{n \in \mathcal{M}} \alpha_n \tilde{Z}^n, \tag{16}$$

resulting in geolocation-aware SSL representations.

Given the classification loss $\mathcal{L}_{\text{class}}$ (6), downstream geolocation loss $\mathcal{L}_{\text{geo}}$ (8), and intermediate-layer geolocation losses $\mathcal{L}_{\text{geo}}^n$ for layers $n \in \mathcal{M}$ (14), the overall loss is defined as:

$$\mathcal{L}_2 = (1-\lambda)\,\mathcal{L}_{\text{class}} + \lambda \left((1-\gamma)\,\mathcal{L}_{\text{geo}} + \gamma \frac{\sum_{n \in \mathcal{M}} \mathcal{L}_{\text{geo}}^n}{|\mathcal{M}|} \right), \tag{17}$$

where $\gamma \in [0, 1]$ balances the downstream and intermediate-layer geolocation prediction losses.

IV. Experiments

A. Datasets

We primarily train our models on VoxLingua107 [7] with 6,628-hour 107-language YouTube recordings and evaluate both on the development set of VoxLingua107 and five out-of-domain datasets to show generalization capability: Babel [41], FLEURS [16], VoxPopuli [42], and the development and dialect development sets of ML-SUPERB 2.0 [15]. Table II summarizes all datasets used in our experiments. For each, we evaluate only on languages that overlap with the VoxLingua107 training set.[2] Therefore, the number of evaluated languages is often smaller than the official test set size listed in Table II. We further train our models on the combined training sets of all five datasets (9,865 hours, 157 languages) to improve domain coverage and upper-bound performance.[3]

B. Model Configuration

We use the 1B-parameter MMS model[4] as the upstream SSL encoder, which consists of 48 Transformer layers with hidden size $D = 1280$ (see (1)). The encoder is fully fine-tuned during training. The downstream ECAPA-TDNN uses channel size $C = 512$ (see (3)), and the language embedding dimension is $E = 192$ (for both downstream (5) and intermediate (10)). The AAMSoftmax loss in (6) is applied with $K = 3$ sub-centers, margin $m = 0.5$, and scaling factor $s = 30$.

To determine the optimal layers for geolocation conditioning, we experiment with four layer selection $\mathcal{M}$ strategies (see Section III-D): bottom $\{0, 4, 8, 12\}$, middle $\{16, 20, 24, 28\}$, top $\{32, 36, 40, 44\}$, and full $\{0, 4, 8, \ldots, 44\}$, denoted as 0-12, 16-28, 32-44, and 0-44, respectively. In addition, we perform ablation on the conditioning projection module in (13), comparing (i) shared vs. independent projections across layers, and (ii) frozen vs. trainable parameters.

[2]Babel: development utterances longer than 10s; FLEURS: official test split; VoxPopuli: development set of transcribed speech; ML-SUPERB 2.0: follows setup in the ML-SUPERB 2.0 challenge [43].

[3]Babel: utterances longer than 10s from the full-language-pack training set; ML-SUPERB 2.0: same processing as evaluation; VoxPopuli: transcribed training set; others use official splits.

[4]https://huggingface.co/facebook/mms-1b

TABLE II
Overview of datasets used in experiments. VL107-only: train on VoxLingua107 only; Combined: train on all training sets; (137, 8): dev and dialect-dev sets in ML-SUPERB 2.0; Seen/Unseen: whether the dataset is used during fine-tuning.

Dataset	Domain	#Langs. Train/Test	Dialect	Training Setup VL107-only	Training Setup Combined
VoxLingua107 [7]	YouTube	107/33	No	Seen	Seen
Babel [41]	Telephone	25/25	No	Unseen	Seen
FLEURS [16]	Read speech	102/102	No	Unseen	Seen
ML-SUPERB 2.0 [15]	Mixed	137/(137, 8)	Yes	Unseen	Seen
VoxPopuli [42]	Parliament	16/16	No	Unseen	Seen

C. Training Setup

For combined training, we use a tri-stage learning rate schedule [44] with warmup 5k steps from 6×10^{-6} to 1×10^{-5}, hold for 20k, then decay to 1×10^{-6} over 75k. Gradient accumulation is applied every 2 steps (VoxLingua107-only) or 4 steps (combined), with batch sizes of 3min and 1.5min, respectively. Optimization uses Adam [45] with $\beta_1 = 0.9$, $\beta_2 = 0.98$. We apply balanced data sampling [1] with upsampling factor $\beta_{\text{lang}} = 0.5$ for languages, and $\beta_{\text{dataset}} = 0.3$ for datasets in combined training. We tune λ and γ in loss $\mathcal{L}_1$ (9) and $\mathcal{L}_2$ (17) over predefined sets with 0.2 and 0.4 selected, respectively. Ablation variants include setting γ=1 (see (17)) and removing detach$(\cdot)$ (see (12)). For inference, we use the highest-accuracy checkpoint on the VoxLingua107 development set for VoxLingua107-only training, and the 62k-step checkpoint for the combined training. All experiments use ESPnet [46] with S3PRL [30] and run on one NVIDIA H200.

V. Results on VoxLingua107-only Training

Table III presents the LID accuracy of models trained on VoxLingua107 and evaluated on both in-domain and out-of-domain test sets. Three settings are compared: (i) a baseline model without geolocation supervision (Section III-A), (ii) a model with downstream geolocation prediction (Section III-C), and (iii) models with geolocation conditioning on intermediate layers (Section III-D). Overall, both geolocation prediction and conditioning models outperform the baseline (see purple-highlighted macro averages in Table III). The geolocation conditioning model with shared, trainable projections on layers 32-44 achieves the highest macro accuracy of 88.9%, outperforming both the baseline and geolocation prediction-only models. This demonstrates the effectiveness of injecting geolocation conditioning signals into intermediate representations. The most significant improvements occur on challenging sets such as ML-SUPERB 2.0 dialect and VoxPopuli, with absolute improvements of 7.3% and 5.6% respectively, suggesting that geolocation conditioning signals improve robustness to both

TABLE III

ACCURACY (%) OF MODELS TRAINED ON VOXLINGUA107 ACROSS IN-DOMAIN AND OUT-OF-DOMAIN TEST SETS. GEO PRED: DOWNSTREAM GEOLOCATION PREDICTION ONLY; GEO COND: INTERMEDIATE-LAYER GEOLOCATION CONDITIONING WITH DOWNSTREAM GEOLOCATION PREDICTION; MACRO AVG.: MACRO AVERAGE ACCURACY OVER ALL SETS; INDEP.: INDEPENDENT; TRAIN.: TRAINABLE; UNDERLINED: GROUP BEST; **BOLD**: BEST PER COLUMN; GRAY : BASELINE; PURPLE : MACRO AVG. OUTPERFORMS BASELINE; ORANGE : BEST OVERALL.

#	Model	Layers	CondProj Type	In-domain	Out-of-domain					Macro Avg.
				VoxLingua107	Babel	FLEURS	ML-SUPERB 2.0		VoxPopuli	
							Dev	Dialect		
1	Baseline	–	None	94.2	86.7	**95.8**	89.0	73.4	85.6	87.5
2	Geo Pred	–	None	94.1	86.0	95.6	89.5	73.8	88.9	88.0
	Conditioning Projection Design and Position									
3	Geo Cond	0-12	Indep. + Frozen	94.3	85.9	95.1	89.1	73.9	90.6	88.1
4	Geo Cond	0-12	Indep. + Train.	94.5	85.1	93.1	88.9	73.5	87.5	87.1
5	Geo Cond	0-12	Shared + Frozen	94.0	83.6	94.7	89.4	72.4	90.4	87.4
6	Geo Cond	0-12	Shared + Train.	94.4	85.2	93.5	88.0	71.7	88.4	86.9
7	Geo Cond	16-28	Indep. + Frozen	95.0	85.9	93.2	89.0	76.3	89.0	88.1
8	Geo Cond	16-28	Indep. + Train.	94.3	84.7	92.1	88.2	72.5	85.9	86.3
9	Geo Cond	16-28	Shared + Frozen	94.0	86.2	94.7	88.7	74.6	87.1	87.5
10	Geo Cond	16-28	Shared + Train.	94.5	86.1	94.5	89.4	71.3	88.3	87.3
11	Geo Cond	32-44	Indep. + Frozen	94.2	87.1	95.0	89.0	77.2	90.4	88.8
12	Geo Cond	32-44	Indep. + Train.	93.7	85.3	93.7	88.3	70.3	86.5	86.3
13	Geo Cond	32-44	Shared + Frozen	94.3	85.9	94.3	88.8	**80.7**	89.2	88.8
14	Geo Cond	32-44	Shared + Train.	94.9	**87.7**	93.5	89.3	78.8	89.5	**88.9**
15	Geo Cond	0-44	Indep. + Frozen	93.9	83.5	94.9	**89.7**	76.5	**91.2**	88.3
16	Geo Cond	0-44	Indep. + Train.	93.7	84.8	94.0	88.3	72.7	87.9	86.9
17	Geo Cond	0-44	Shared + Frozen	94.4	83.9	95.0	89.1	68.8	89.9	86.8
18	Geo Cond	0-44	Shared + Train.	93.9	86.5	94.3	88.2	70.4	87.3	86.8
	Remove Downstream Geolocation Loss ($\gamma = 1$)									
19	Geo Cond	32-44	Shared + Train.	**95.2**	86.8	93.2	88.5	77.4	90.2	88.6
	Remove detach$(\cdot)$									
20	Geo Cond	32-44	Shared + Train.	94.2	85.2	93.8	89.4	73.8	87.2	87.2

intra-language variations and domain shifts. Performance on FLEURS slightly declines, but remains comparable to the baseline, introducing minimal trade-off.

A. Design and Position of Conditioning Projection

Early-layer conditioning benefits from independent and frozen projection modules. Conditioning early layers (0–12, 16–28) performs best with independent and frozen projection modules. At layers 0–12, the independent frozen projection achieves up to 3.1% higher accuracy than its trainable counterpart on VoxPopuli. This result implies that frozen projection modules provide more consistent conditioning and stabilize low-level features than trainable modules. Among frozen settings, independent projections outperform shared ones (e.g., 88.1% vs. 87.5%) , highlighting the benefits of layer-specific integration of geolocation cues.

Deep-layer representations offer a stable semantic space for geolocation conditioning. Deep-layer conditioning (layers 32–44) benefits more from shared and trainable projections, which achieves the highest macro average accuracy (88.9%) among all configurations. Notably, shared and frozen projections remain competitive, especially on the ML-SUPERB 2.0 dialect development set, scoring the best accuracy of 80.7%. These results indicate that deep layers provide semantically stable representations suitable for both static and adaptive conditioning. Furthermore, shared projections consistently outperform independent ones on macro average accuracy, implying that a unified transformation better supports geolocation integration at deep layers.

Conditioning across all layers does not yield cumulative performance gains. Applying geolocation conditioning across all layers (0-44) mirrors early-layer trends: frozen projections work better than trainable ones, with the independent frozen setup achieving the highest accuracy in ML-SUPERB 2.0 development set (89.7%). However, this approach underperforms compared to deep-layer injection (32-44), and in some cases (e.g., independent trainable), even falls short of early-layer injection (e.g., macro average accuracy 86.9% vs. 87.1% at layers 0-12). This implies that broad conditioning may introduce redundancy rather than cumulative benefit.

B. Effect of Downstream Geolocation Loss and Detachment

To assess the effect of downstream geolocation loss, we remove it by setting $\gamma = 1$ in (17), while keeping intermediate-layer geolocation conditioning (experiment 19 in Table III). Compared to experiment 14, the performance drops in out-of-domain settings, despite achieving the best in-domain accuracy on VoxLingua107 (95.2%). This suggests that downstream geolocation supervision benefits cross-domain generalization.

We further examine the role of detaching the intermediate geolocation prediction before projecting it into the hidden space. Removing the detach$(\cdot)$ operation leads to a significant performance degradation (see experiment 20), especially on the ML-SUPERB 2.0 dialect development set (5.0% absolute drop compared to experiment 14). This indicates that without

TABLE IV

ACCURACY (%) OF REPRESENTATIVE LID MODELS. TYPE: SSL-BASED, ACOUSTIC FEATURE-BASED, JOINT LID-ASR, GEOLOCATION-PRETRAINED, AND OUR GEOLOCATION-CONDITIONED LID MODEL (LAYERS 32–44, SHARED TRAINABLE PROJECTION). MACRO AVG.: AVERAGE OVER ALL SETS. XEUS: ML-SUPERB 2.0 RESULTS FROM [43]. OURS: VOXLINGUA107-ONLY (VL107-ONLY) OR COMBINED TRAINING. **BOLD**: BEST OVERALL.

Model	Type	VoxLingua107	Babel	FLEURS	ML-SUPERB 2.0		VoxPopuli	Macro Avg.
					Dev	Dialect		
MMS-LID-4017 [1]	SSL	93.9	–	97.2	–	–	–	–
XLS-R-attentive [47]	SSL	**95.3**	–	–	–	–	–	–
TitaNet-LID [48]	Acoustic	94.4	–	–	–	–	–	–
XEUS [2]	LID-ASR	–	–	93.0	77.1	79.1	–	–
MMS 1B LIDCTC [23]	LID-ASR	–	–	–	86.9	74.2	–	–
OWSM v4 medium [10]	LID-ASR	–	–	95.6	–	–	–	–
Geo 1B [17]	Geo Pretrain	–	–	96.7	–	–	–	–
Ours (VL107-only)	Geo Cond	94.9	87.7	93.5	**89.3**	78.8	89.5	88.9
Ours (Combined)	Geo Cond	94.4	**95.4**	**97.7**	88.6	**86.8**	**99.0**	**93.7**

TABLE V

ACCURACY (%) ON ML-SUPERB 2.0 DIALECT DEV SET. GEO COND: LAYERS 32–44 WITH SHARED FROZEN PROJECTION; UNDERLINED: BEST ACROSS BOTH SETTINGS.

Model	ara	deu	ell	eng	guj	spa	tam	tel
Baseline	65.3	76.5	75.5	67.4	99.0	96.4	100.0	98.0
Geo Cond	61.5	88.3	83.4	76.5	97.9	98.5	100.0	98.0

detachment, gradients from the classification objective interfere with the learning of geolocation vectors, causing them to align with the classification target rather than preserving the geolocation information.

C. Improvement on Dialectal and Accented Variations

Table V presents detailed results for each language in the ML-SUPERB 2.0 dialect development set. Geolocation conditioning significantly improves or preserves accuracy on most languages with dialectal or accented variations, except for Arabic (ara). This suggests that geolocation conditioning improves the model's robustness to intra-language variations consistently across languages.

To further analyze its effect on intra-language variations, we visualize the utterance-level embeddings for English speech in ML-SUPERB 2.0 dialect development set in Fig. 2. With geolocation conditioning, the compactness score decreases from 0.71 to 0.67, indicating tighter clustering of intra-language embeddings. This demonstrates that geolocation signals, serving as a unifying constraint, guide the model to learn compact representations for intra-language variations, leading to better generalization across dialects and accents.

VI. RESULTS ON COMBINED TRAINING

Building on Section V, we expand training data from 6,628 to 9,865 hours with broader domain coverage, and train the geolocation conditioning model using shared, trainable conditioning projections on layers 32-44, achieving SOTA performance. Table IV reports the LID accuracy of our geolocation-aware LID models compared to existing SOTA systems. Our model achieves new SOTA accuracy on FLEURS (97.7%) and ML-SUPERB 2.0 (dev: 88.6%, dialect dev: 86.8%), while maintaining comparable results on VoxLingua107. Compared with Geo 1B, which relies on utterance-level geolocation pretraining, our method uses only language-level geolocation signals and achieves higher accuracy on FLEURS (97.7% vs. 96.7%). This demonstrates that estimated, language-level geolocation is sufficient to improve LID performance without requiring fine-grained utterance-level location labels. The checkpoint of our SOTA model is publicly available.

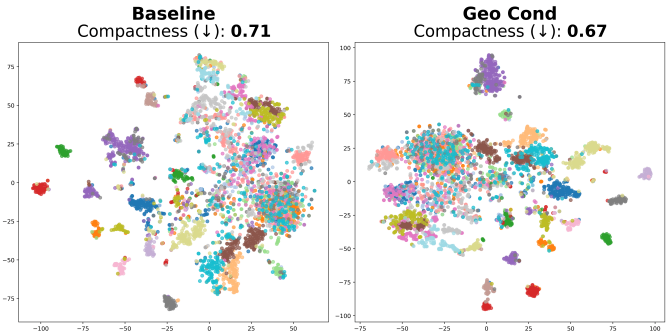

Fig. 2. t-SNE plots of English speech embeddings from ML-SUPERB 2.0 dialect dev set. Colors indicate accents within the English class. Geo Cond: geolocation-conditioned model (layers 32–44, shared frozen). Compactness: average distance to the English embedding centroid, lower indicates tighter clustering.

VII. CONCLUSION

In this paper, we propose geolocation-aware LID, a novel approach that incorporates language-level geolocation supervision and conditioning into SSL-based LID models. Using geolocation vectors from `lang2vec` project [14], we predict the language geolocation at both SSL encoder intermediate layers and the downstream embedding extractor, and inject the intermediate-layer predictions as conditioning signals into the encoder. Experiments show that our approach improves overall model performance, particularly enhancing robustness to dialectal and accented variations. Trained on a 157-language multi-domain dataset, our model achieves new SOTA results on FLEURS [16] and ML-SUPERB 2.0 [15].

ACKNOWLEDGMENT

Experiments used PSC Bridges2 and NCSA Delta via ACCESS CIS210014 and IRI120008P, supported by NSF grants #2138259, #2138286, #2138307, #2137603, #2138296.

REFERENCES

[1] V. Pratap, A. Tjandra, B. Shi, P. Tomasello *et al.*, "Scaling speech technology to 1,000+ languages," *Journal of Machine Learning Research*, vol. 25, no. 97, pp. 1–52, 2024.

[2] W. Chen, W. Zhang, Y. Peng, X. Li *et al.*, "Towards robust speech representation learning for thousands of languages," in *Proc. EMNLP*, 2024, pp. 10 205–10 224.

[3] A. Babu, C. Wang, A. Tjandra, K. Lakhotia *et al.*, "XLS-R: Self-supervised cross-lingual speech representation learning at scale," in *Proc. Interspeech*, 2021, pp. 2278–2282.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman *et al.*, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, 2023, pp. 28 492–28 518.

[5] Y. Peng, J. Tian, W. Chen, S. Arora *et al.*, "OWSM v3.1: Better and faster open Whisper-style speech models based on E-Branchformer," in *Proc. Interspeech*, 2024, pp. 352–356.

[6] Y. Zhang, W. Han, J. Qin, Y. Wang *et al.*, "Google USM: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[7] J. Valk and T. Alumäe, "VoxLingua107: a dataset for spoken language recognition," in *Proc. SLT*, 2021, pp. 652–658.

[8] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale *et al.*, "Seamless: Multilingual expressive and streaming speech translation," *arXiv preprint arXiv:2312.05187*, 2023.

[9] L. B. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale *et al.*, "SeamlessM4T: Massively multilingual & multimodal machine translation," *arXiv preprint arXiv:2308.11596*, 2023.

[10] Y. Peng, S. Muhammad, Y. Sudo, W. Chen *et al.*, "OWSM v4: Improving open Whisper-style speech models via data scaling and cleaning," in *Proc. Interspeech*, 2025.

[11] H. Liu, L. P. G. Perera, A. W. Khong, E. S. Chng *et al.*, "Efficient self-supervised learning representations for spoken language identification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1296–1307, 2022.

[12] K. Choi, A. Pasad, T. Nakamura, S. Fukayama *et al.*, "Self-supervised speech representations are more phonetic than semantic," in *Proc. Interspeech*, 2024, pp. 4578–4582.

[13] M. Yang, R. C. M. C. Shekar, O. Kang, and J. H. L. Hansen, "What can an accent identifier learn? probing phonetic and prosodic information in a wav2vec2-based accent identification model," in *Proc. Interspeech*, 2023, pp. 1923–1927.

[14] P. Littell, D. R. Mortensen, K. Lin, K. Kairis *et al.*, "URIEL and lang2vec: Representing languages as typological, geographical, and phylogenetic vectors," in *Proc. EACL (Volume 2, Short Papers)*, 2017, pp. 8–14.

[15] W. Chen, C. Meng, J. Shi, M. Bartelds *et al.*, "The ML-SUPERB 2.0 challenge: Towards inclusive ASR benchmarking for all language varieties," in *Proc. Interspeech*, 2025.

[16] A. Conneau, M. Ma, S. Khanuja, Y. Zhang *et al.*, "FLEURS: Few-shot learning evaluation of universal representations of speech," in *Proc. SLT*, 2023, pp. 798–805.

[17] P. Foley, M. Wiesner, B. Odoom, L. P. Garcia Perera *et al.*, "Where are you from? Geolocating speech and applications to language identification," in *Proc. NAACL (Long Papers)*, K. Duh, H. Gomez, and S. Bethard, Eds., 2024, pp. 5114–5126.

[18] J. Dunn and L. Edwards-Brown, "Geographically-informed language identification," in *Proc. LREC-COLING*, 2024, pp. 7672–7682.

[19] S. Cao, Y. Zhang, X. Feng, and L. Ma, "Improving speech recognition accuracy of local POI using geographical models," in *Proc. SLT*, 2021, pp. 180–185.

[20] X. Xiao, H. Chen, M. Zylak, D. Sosa *et al.*, "Geographic language models for automatic speech recognition," in *Proc. ICASSP*, 2018, pp. 6124–6128.

[21] J. Lee and S. Watanabe, "Intermediate loss regularization for CTC-based speech recognition," in *Proc. ICASSP*, 2021, pp. 6224–6228.

[22] A. Tjandra, C. Liu, F. Zhang, X. Zhang *et al.*, "DEJA-VU: Double feature presentation and iterated loss in deep transformer networks," in *Proc. ICASSP*, 2020, pp. 6899–6903.

[23] Q. Wang, J. Sun, Y. Peng, and S. Watanabe, "Improving multilingual speech models on ML-SUPERB 2.0: Fine-tuning with data augmentation and LID-aware CTC," in *Proc. Interspeech*, 2025.

[24] J. Nozaki and T. Komatsu, "Relaxing the conditional independence assumption of CTC-based ASR by conditioning on intermediate predictions," in *Proc. Interspeech*, 2021, pp. 3735–3739.

[25] W. Chen, B. Yan, J. Shi, Y. Peng *et al.*, "Improving massively multilingual ASR with auxiliary CTC objectives," in *Proc. ICASSP*, 2023, pp. 1–5.

[26] Y.-J. Lu, J. Liu, T. Thebaud, L. Moro-Velazquez *et al.*, "CA-SSLR: Condition-aware self-supervised learning representation for generalized speech processing," in *Proc. NeurIPS*, 2024.

[27] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Proc. NeurIPS*, vol. 33, pp. 12 449–12 460, 2020.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit *et al.*, "Attention is all you need," in *Proc. NeurIPS*, 2017, pp. 6000–6010.

[29] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner *et al.*, "Deep contextualized word representations," in *Proc. NAACL*, 2018, pp. 2227–2237.

[30] S. wen Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai *et al.*, "SUPERB: Speech processing universal performance benchmark," in *Proc. Interspeech*, 2021, pp. 1194–1198.

[31] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," in *Proc. Interspeech*, 2020, pp. 3830–3834.

[32] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. CVPR*, 2018, pp. 7132–7141.

[33] S.-H. Gao, M.-M. Cheng, K. Zhao, X.-Y. Zhang *et al.*, "Res2Net: A new multi-scale backbone architecture," *IEEE TPAMI*, 2019.

[34] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018, pp. 2252–2256.

[35] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456.

[36] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. CVPR*, 2019.

[37] M. Zhao, Y. Ma, M. Liu, and M. Xu, "The SpeakIn system for VoxCeleb speaker recognition challenge 2021," *arXiv preprint arXiv:2109.01989*, 2021.

[38] J.-w. Jung, W. Zhang, J. Shi, Z. Aldeneh *et al.*, "ESPnet-SPK: Full pipeline speaker embedding toolkit with reproducible recipes, self-supervised front-ends, and off-the-shelf models," *arXiv preprint arXiv:2401.17230*, 2024.

[39] H. Hammarström, R. Forkel, M. Haspelmath, and S. Bank, "Glottolog 5.2," http://glottolog.org, 2025, accessed on 2025-06-02.

[40] Á. González, "Measurement of areas on a sphere using Fibonacci and latitude–longitude lattices," *Mathematical geosciences*, vol. 42, pp. 49–64, 2010.

[41] M. J. Gales, K. M. Knill, A. Ragni, and S. P. Rath, "Speech recognition and keyword spotting for low-resource languages: Babel project research at cued," in *Proc. SLTU*, 2014, pp. 16–23.

[42] C. Wang, M. Riviere, A. Lee, A. Wu *et al.*, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc. ACL-IJCNLP (Long Papers)*, 2021, pp. 993–1003.

[43] W. Chen, J. Shi, S.-H. Wang, S. Watanabe *et al.*, "Interspeech 2025 ML-SUPERB 2.0 challenge," https://multilingual.superbbenchmark.org/challenge-interspeech2025/challenge_overview, accessed on 2025-06-02.

[44] M. Ott, S. Edunov, A. Baevski, A. Fan *et al.*, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proc. NAACL-HLT*, 2019.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR (Poster)*, 2015.

[46] S. Watanabe, T. Hori, S. Karita, T. Hayashi *et al.*, "ESPnet: End-to-end speech processing toolkit," in *Proc. Interspeech*, 2018, pp. 2207–2211.

[47] K. Kukk and T. Alumäe, "Improving language identification of accented speech," in *Proc. Interspeech*, 2022, pp. 1288–1292.

[48] F. Jia, N. R. Koluguri, J. Balam, and B. Ginsburg, "A compact end-to-end model with local and global context for spoken language identification," in *Proc. Interspeech*, 2023, pp. 5321–5325.