PAWALID: A Compact Byte-Level Language Identification Model for African and Low-Resource Languages

Anonymous ACL submission

Abstract

Language identification (LID) is a critical first step in multilingual NLP pipelines, yet existing systems struggle to adequately support African and other low-resource languages. We aim to bridge this gap by introducing **PAWALID**, a compact 7.4M-parameter byte-level LID model supporting 398 languages, including 109 African languages and complete coverage of a 49-language East African benchmark. PAWALID combines UTF-8 byte embeddings, Unicode-aware byte-class features, hashed byte n-grams, multi-scale convolutions, and lightweight self-attention, and is trained using language classification and language family objectives. We evaluate against CommonLingua, AfroLID, GlotLID, and OpenLID across seven multilingual and African-centric benchmarks using a coverage-aware protocol that reports both all-language and covered-language macro-F1. On AfriMMT-EA, PAWALID achieves 83.5% all-language macro-F1 with complete 49/49 coverage, compared with 50.6% for AfroLID at 38/49 coverage and 53.4% for GlotLID at 33/49 coverage. On CommonLID, PAWALID obtains the highest all-language macro-F1 among evaluated systems, reaching 55.6% compared with 50.8% for GlotLID. Our results show that African-language LID evaluation must account for both predictive accuracy and language coverage.

1 Introduction

Language identification (LID) is the entry point to many multilingual NLP pipelines. Before text can be translated, retrieved, filtered, or analyzed, a system must first identify language. Errors at this stage propagate downstream: misidentified text contaminates corpora, weakens multilingual models, and makes poorly supported language communities less visible in language technology. While LID is often near ceiling for high-resource languages, performance remains poor for African languages (Haas and Derczynski, 2021).

Existing LID systems were not designed with African coverage in mind. For example, FastText LID (Joulin et al., 2017) supports 176 languages with limited African representation, while CLD3 (Botha et al., 2017) remains widely used in corpus construction despite similarly limited support for underserved African languages.

Dataset audits reveal substantial labeling and quality issues in African-language corpora and web-crawled text (Kreutzer et al., 2022a; Caswell et al., 2020). These shortcomings weaken multilingual models and contribute to continued digital exclusion (Ojo et al., 2025; International Telecommunication Union, 2024; Kornai, 2013).

Recent LID systems reflect different design priorities: AfroLID (Adebara et al., 2022) focuses on African-language coverage, GlotLID (Kargaran et al., 2023) on broad multilingual coverage (1,665 languages), and OpenLID (Burchell et al., 2023) on curated open data. However, none combines compact byte-level modeling with complete East African coverage for resource-constrained settings (International Telecommunication Union, 2024). LID for African languages remains challenging because of linguistic complexity and limited data coverage (Adebara et al., 2022).

To address these gaps, we introduce PAWALID, a 7.4M-parameter byte-level LID model supporting 398 language labels, including 109 African languages and complete coverage of the 49-language AfriMMT-EA benchmark. The model operates directly on UTF-8 byte sequences and combines raw byte embeddings, Unicode-aware byte-class features, hashed byte n-grams, multi-scale convolutions, and lightweight self-attention. It is trained on a complementary mixture of AfriMMT-EA (Etori et al., 2026), CommonLingua (PleIAs, 2026), and PawaLIDAfrica. The architecture includes mixed-language and unknown-language heads, but these are not optimized in the reported checkpoint because reliable supervision for these settings re-

mains limited. We evaluate PAWALID against CommonLingua, AfroLID, GlotLID, and OpenLID on seven broad multilingual and African-centric benchmarks. On AfriMMT-EA, PAWALID achieves 83.5% all-language macro-F1 with full 49/49 coverage, compared with 53.4% for GlotLID at 33/49 coverage. On CommonLID, PAWALID achieves the highest all-language macro-F1 among evaluated systems, reaching 55.6% compared with 50.8% for GlotLID. Across broader African subsets, GlotLID remains stronger when its larger inventory covers more benchmark languages, while PAWALID remains competitive on supported languages and strongest in the East African setting.

Our contributions are threefold:

1. We introduce PAWALID, a compact 7.4M-parameter byte-level LID model supporting 398 language labels, including 109 African languages and full coverage of a 49-language East African benchmark.

2. We present a coverage-aware evaluation protocol that reports all-language macro-F1, covered-language macro-F1, and language coverage, making unsupported labels explicit rather than silently dropping them.

3. We provide a seven-benchmark evaluation showing that PAWALID is strongest in the East African setting and competitive on supported African languages, while broad-inventory systems remain stronger on global benchmarks.

2 Related Work

Early LID systems relied on character and byte n-gram statistics and lightweight methods such as Langdetect, FastText LID, and the CLD series (Cavnar et al., 1994; Nakatani, 2010; Joulin et al., 2016, 2017; McCandless, 2010; Salcianu et al., 2018). While effective, these approaches often struggle with short text and low-resource languages because of skewed training distributions and limited African-language representation (Oro et al., 2018; Sindane and Marivate, 2024). Neural approaches based on multilingual encoders such as mBERT and XLM-R improve multilingual modeling but remain dependent on tokenizers learned largely from high-resource data, leading to segmentation and out-of-vocabulary issues for underrepresented languages and scripts (Devlin et al., 2018; Conneau et al., 2020; Ndomba et al., 2024; Bhattamishra et al., 2020). Tokenizer-free byte-level models such as ByT5 and CANINE address these limitations (Xue et al., 2022; Clark et al., 2022). African-focused systems such as AfroLID (Adebara et al., 2022) prioritize African-language coverage, but do not jointly target compact deployment and complete East African language support. PAWALID builds on this direction through compact byte-level modeling with coverage-aware evaluation and stronger support for underrepresented East African languages.

3 Method

This section describes data the construction, preprocessing pipeline, model architecture, and training.

3.1 Data Sources

To construct PawaLID, we prioritize datasets with reliable language annotations in order to mitigate quality issues previously reported in automatically collected corpora (Kreutzer et al., 2022b). PawaLID uses a mixture of three complementary sources of datasets (see Appendix B), including:

AfriMMT-EA. (Etori et al., 2026), a multi-domain machine translation (MT) corpus covering 53 East African languages across five countries: Kenya (24), Uganda (15), Tanzania (11), Rwanda (1), and Ethiopia (2) across three major families: Niger-Congo, Nilo-Saharan, and Afro-Asiatic, covering several subfamilies including Bantu, Nilotic, and Cushitic. The original dataset contains 66 translation directions and approximately 714K parallel sentence pairs (See Table 9).

CommonLingua: (PleIAs, 2026), a multilingual LID corpus covering 334 languages, including 61 African languages spanning eight language families: Bantu (21), Niger-Congo/West Africa (18), Afro-Asiatic and Semitic (7), Cushitic and Chadic (4), Berber (3), Nilo-Saharan (3), pidgins and creoles (3), and two (2) languages from Malagasy, and Afrikaans((see Tables 10, 11, and 12)).

PawaLIDAfrica[1]: PawaLIDAfrica contains 341,593 examples across 32 African languages collected from multiple public sources (Table 8). Table 13 summarises the geographic distribution of PawaLID's 398 language labels. African languages constitute the largest single-region group (109

[1] https://huggingface.co/datasets/pawa-ai/PawaLIDAfrica

Check	Count	Action
Raw AfriMMT-EA evals candidates	53,923	initial held-out
Duplicate/invalid evals rows	126	removed from evals
Final AfriMMT-EA eval examples	53,797	evaluated
Intra-training duplicates	9,233	removed from training
Train examples overlaping evals text	34,197	removed from training

Table 1: Decontamination statistics for AfriMMT-EA. Additional details are provided in Appendix Section C.

labels, 27.4%), reflecting the model's design focus. A complete listing of African languages by family is provided in Table 14 in the Appendix.

3.2 Data Preparation

All datasets are normalized into a common schema consisting of text, language label, language family, mixed-language flag, and unknown-language flag. We map known ISO-code variants where appropriate, for example, mapping Swahili aliases to `swh`. We remove short examples using a minimum length threshold and deduplicate examples by text–label pair. The final training configuration contains 398 language and 48 language-family labels.

3.3 PawaLID Architecture

The PawaLID architecture (Figure 1) is a compact, byte-level classifier that operates directly on raw UTF-8 byte sequences, avoiding the need for a learned or language-specific tokenizer. Each input text is encoded as a sequence of up to 512 bytes, with three parallel feature streams extracted at every position: (i) raw byte identities from a vocabulary of 257 symbols, (ii) byte-class labels drawn from 11 Unicode-aware script categories (ASCII letter, digit, whitespace, punctuation, Latin Extended, Arabic, Ethiopic, NKo, CJK, and other), and (iii) hashed byte n-grams for $n \in \{2, 3, 4\}$, each mapped into 4,096 buckets via rolling polynomial hashing. Each stream is independently embedded—byte identities into 256-dimensional vectors, byte classes into 24-dimensional vectors, and each n-gram order into 64-dimensional vectors—and the resulting representations are concatenated at each position to form a 472-dimensional input, which is linearly projected to 256 dimensions with LayerNorm, GELU activation, and dropout. The projected sequence passes through an encoder consisting of four multi-scale depthwise-separable convolution blocks followed by two self-attention blocks. Each convolution block applies five parallel depthwise Conv1D branches with kernel sizes 3, 5, 9, 15, and 31,

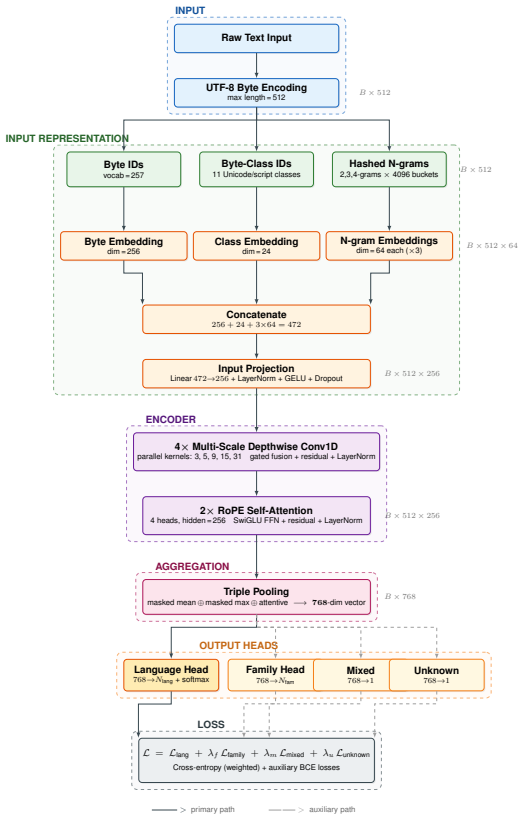

Figure 1: PawaLID-Broad architecture for byte-level language identification.

whose outputs are concatenated and fused through a gated linear unit with a residual connection and LayerNorm. This multi-scale design enables the model to capture character-level patterns (e.g., diacritics, script boundaries) alongside word-level and phrase-level n-gram features in a single pass. The self-attention blocks use Rotary Position Embeddings (RoPE; Su et al., 2024) with 4 heads and SwiGLU feed-forward networks (Shazeer, 2020), each with residual connections and LayerNorm. After encoding, a triple pooling layer aggregates the sequence into a fixed-length 768-dimensional vector by concatenating masked mean pooling, masked max pooling, and learned attentive pooling. This vector is fed to four output heads: (i) a primary language classification head with softmax over N_{lang} classes, (ii) a language-family head over N_{fam} families, (iii) a binary mixed-language detector, and (iv) a binary unknown-language detector. The entire model contains 7.4M trainable parameters.

3.4 Training Objective

Our architecture supports a four-head objective consisting of language classification, language-family prediction, mixed-language detection, and

unknown-language detection:

$$\mathcal{L} = \mathcal{L}_{\text{lang}} + \lambda_f \, \mathcal{L}_{\text{family}} + \lambda_m \, \mathcal{L}_{\text{mixed}} + \lambda_u \, \mathcal{L}_{\text{unknown}}. \tag{1}$$

In the reported checkpoint, we use weighted cross-entropy for $\mathcal{L}_{\text{lang}}$ and set $\lambda_f = 0.03$. We set $\lambda_m = \lambda_u = 0$ because the current training mixture does not contain reliable supervision for mixed-language or unknown-language detection. Thus, the final model should be interpreted as a language and language-family identifier; open-set. code-switched detection remain future work.

4 Experimental Setup

4.1 Evaluation Datasets

For external evaluation, we focus on broad coverage and African-centric benchmarks. The broad-coverage benchmarks are Common-LID(Ortiz Suarez et al., 2026), FLORES+(NLLB Team et al., 2024), and UDHR-LID (Kargaran et al., 2023). The African-centric benchmarks are African-language subsets of FLORES+, UDHR-LID, and CommonLID, together with AfriMMT-EA (Etori et al., 2026). Our evaluation design separates two questions that are often conflated in LID evaluation: whether a model performs well on the languages it supports and whether it supports the languages needed for a given deployment setting.

4.2 Baselines

We compare PAWALID against four released LID systems that collectively span the coverage–efficiency frontier. AfroLID (Adebara et al., 2022) is the strongest African-language specialist, covering 517 languages and varieties on an XLM-R backbone (200M parameters). GlotLID (Kargaran et al., 2023) provides the broadest inventory, covering 1,665 languages, using a FastText architecture with a 2–3 GB model footprint. OpenLID (Burchell et al., 2023) occupies the middle ground: 201 languages backed by a manually audited, open-data FastText model. CommonLingua (PleIAs, 2026) is the most architecturally comparable baseline to PAWALID: both are compact neural classifiers with fewer than 10M parameters trained on overlapping African-language data, making CommonLingua the natural head-to-head reference for per-language analysis (§5.2). For each system, we evaluate the released checkpoint without task-specific fine-tuning, applying the same ISO-code normalization and benchmark splits.

4.3 Evaluation Metrics

Because evaluated systems have different label inventories, we report both *all*-language and *covered*-language metrics. *All*-F1 is macro-F1 over every language in the benchmark, assigning zero performance to unsupported languages. *Covered*-F1 is macro-F1 restricted to languages that the model can predict. We also report coverage as *cov/total* and, where applicable, row-level accounting, such as the number of examples kept or dropped due to unsupported labels (see Table 4). This prevents a model from appearing strong simply because unsupported languages are silently removed before evaluation: Let $\mathcal{Y}$ denote the full set of benchmark labels and $\mathcal{Y}_m \subseteq \mathcal{Y}$ denote the subset supported by model m. For each language y, let $F1_y(m)$ be the per-language F1 score, with $F1_y(m) = 0$ when $y \notin \mathcal{Y}_m$. We define all-language macro-F1 as:

$$\text{AllF1}(m) = \frac{1}{|\mathcal{Y}|} \sum_{y \in \mathcal{Y}} F1_y(m), \tag{2}$$

and covered-language macro-F1 as:

$$\text{CovF1}(m) = \frac{1}{|\mathcal{Y}_m|} \sum_{y \in \mathcal{Y}_m} F1_y(m). \tag{3}$$

Coverage is reported as $|\mathcal{Y}_m|/|\mathcal{Y}|$. This distinction prevents systems with narrow inventories from appearing artificially strong after unsupported labels are removed.

4.4 Evaluation Protocol

We treat this evaluation as a deployment-oriented comparison rather than a single leaderboard score. A model with high covered-language F1 but low coverage may be useful for a narrow setting, but it can fail as a routing or corpus-filtering component in African multilingual pipelines. Conversely, a broad-coverage model may perform well on global benchmarks while still missing specific East African languages. We therefore interpret accuracy, macro-F1, weighted-F1, and coverage jointly, with particular attention to languages that are absent from existing high-coverage systems.

4.5 Ablation Protocol

We perform a series of controlled ablations to identify which design choices contribute to external generalization on CommonLID. The ablations target three failure modes observed during development: hash-capacity sensitivity, length-distribution mismatch, and over-confident predictions on closely related languages.

Benchmark	PawaLID			AfroLID			CommonLingua			GlotLID			OpenLID		
	all	cov	cov/total	all	cov	cov/total	all	cov	cov/total	all	cov	cov/total	all	cov	cov/total
FLORES+	67.4	85.7	171/210	15.8	77.8	49/210	60.2	79.7	165/210	**83.7**	84.6	208/210	78.4	92.4	184/210
UDHR-LID	32.2	81.1	195/418	15.2	77.0	94/418	29.1	76.9	186/418	**69.2**	79.7	368/418	24.2	90.2	155/418
CommonLID	**55.6**	64.5	95/108	9.4	60.8	22/108	50.3	59.1	94/108	50.8	62.7	96/108	46.3	73.4	74/108
AfriMMT-EA	**83.5**	83.5	**49/49**	50.6	65.7	38/49	4.8	48.2	7/49	53.4	83.9	33/49	6.3	92.7	5/49
FLORES+-Afr	73.6	85.2	37/42	71.9	80.8	38/42	56.1	76.1	33/42	**96.9**	96.9	**42/42**	87.5	98.2	38/42
UDHR-LID-Afr	69.3	90.1	37/46	74.7	81.2	43/46	53.9	84.0	32/46	**90.5**	92.6	45/46	69.3	98.9	35/46
CommonLID-Afr	61.3	64.1	22/23	52.1	64.7	20/23	57.5	60.1	22/23	**63.8**	73.4	21/23	53.3	80.2	16/23

Table 2: LID performance across broad multilingual and African-centric benchmarks. *all*: macro-F1 (%) over all benchmark languages; *cov*: macro-F1(%) over supported languages only; *cov/total*: supported languages out of total. Best *all*-F1 are in **bold**.

Hash-bucket capacity. We first test whether increasing the hashed byte n-gram capacity improves external performance. Starting from a 4,096-bucket configuration, we increase the hash space to 32,768 buckets while keeping the rest of the training setup fixed. This tests whether additional byte n-gram capacity improves discrimination enough to justify the larger feature space.

Length augmentation. We then introduce byte-level length augmentation during training. Inputs longer than 80 bytes are randomly cropped with probability 0.5 to a byte range between 20 and 400. For inputs longer than the model maximum length, we apply random-window truncation instead of always taking the first 512 bytes. Evaluation remains deterministic and uses the original text. This augmentation is motivated by the mismatch between long paragraph-style training examples and shorter, noisier CommonLID snippets.

Label smoothing. Finally, we test label smoothing ($\epsilon = 0.03$) on the language-classification head. This regularizer is intended to reduce overconfident decisions, especially among typologically or orthographically close languages.

Confusion-driven hard negatives. Separately on the main progression plot, we analyze confusion matrices, and construct targeted hard-negative groups for clusters that remain difficult after length augmentation (see Table 5). Most important clusters are Arabic such as (`arb`, `arz`, `ary`), Indonesian–Malay (`ind`, `msa`), Sinitic varieties (`zho`, `yue`, `wuu`), Lingala–Yoruba (`lin`, `yor`), and Afrikaans as a broad Latin-script absorber. Hard-negative batches are used cautiously to avoid targeted discrimination dominating the main LID objective.

Configuration	Macro-F1	Weighted-F1
Base: 4,096 hash buckets	66.44	82.77
32,768 hash buckets	66.45	82.78
+ length augmentation	69.00	87.71
+ label smoothing ($\epsilon = 0.03$)	69.07	87.80

Table 3: CommonLID ablation results under native supported-label evaluation. Length augmentation provides the largest gain, while increasing hash capacity yields negligible improvement.

5 Results

We evaluate PAWALID against the four baseline systems described in §4.2 across seven benchmarks spanning general and African-centric language identification. Table 2 reports *all*-F1, *covered*-F1, and language coverage for each model. Results indicate that no single LID system dominates across all settings: GlotLID performs strongly on broad multilingual benchmarks, whereas PAWALID is strongest on African-centric (East African).

5.1 Performance on East African Benchmark

Table 2 reports results on AfriMMT-EA, evaluated on the 49-language subset described in §B. PAWALID achieves 83.5% all-language macro-F1 with complete 49/49 language coverage. GlotLID achieves a similar covered-language F1 on the languages it supports, but covers only 33 of the 49 benchmark languages, thereby reducing its all-language macro-F1 to 53.4%. This illustrates why coverage-aware evaluation is necessary: a model can classify supported languages well while still failing as a deployment router when many target languages are outside its inventory. CommonLingua and OpenLID cover only 7 and 5 languages, respectively, making them severely limited for this East African setting. Languages such as Bukusu (`bxk`), Taveta (`tvs`), Chuka (`cuh`), and Zanaki (`zak`) are included in PAWALID's label inventory but

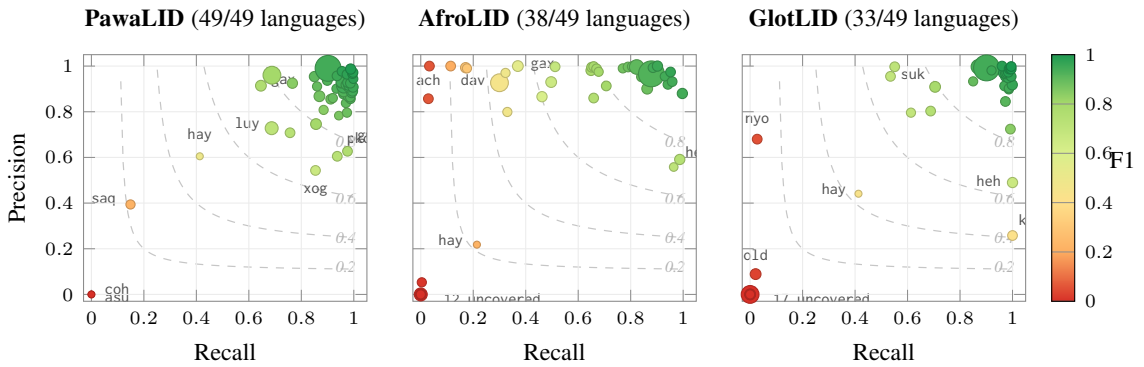

Figure 2: Precision–recall by language on AfriMMT-EA. PAWALID covers all 49 languages, with most above the F1 = 0.8 iso-curve. AFROLID covers 38/49, with many low-resource languages below F1 = 0.6. GlotLID covers 33/49; 16 unsupported languages cluster at the origin (F1 = 0). Point area denotes support and colour indicates F1.

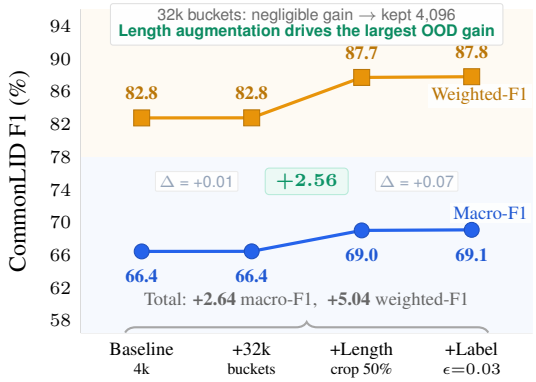

Figure 3: Ablation progression on CommonLID. Increasing hash buckets yields negligible gains, length augmentation provides the largest gain; label smoothing gives a small benefit.

Model	Kept	Dropped	Langs
PawaLID	371,291	1,939	95
CommonLingua	371,233	1,997	94
GlotLID	293,655	79,575	96
OpenLID	–	–	74

Table 4: CommonLID language coverage (ISO-normalized). PawaLID supports 95/108 benchmark languages.

absent from the evaluated baseline systems, preventing those models from predicting them. **Figure 2** visualizes this result at the per-language level. The precision–recall view is important, because aggregate macro-F1 alone does not show whether a model fails due to weak classification on covered languages or because some languages are not supported at all. In the PawaLID panel, most languages lie in the high-recall, high-precision region, with the majority above the F1 = 0.8 iso-curve. The GlotLID panel shows a visible cluster of uncovered languages at the origin, where both precision and recall are zero. Figure 2 supports two complementary observations. First, PAWALID's advantage on AfriMMT-EA arises not only from higher aggregate performance but also from broader language coverage, including languages absent from the evaluated baselines. Second, among the supported languages, most PAWALID predictions lie in the upper-right region of the plot, indicating high joint precision and recall. Together, these results suggest that improved African-language coverage substantially increases the effectiveness of East African deployment LID systems. A full per-language breakdown is provided in Table 9 (Appendix **??**). PAWALID is the only evaluated model to produce nonzero F1 for 20 of the 49 languages—including Bena, Chuka, Embu, Jita, Langi, Luhya, Taita, Taveta, Zanaki, Borana, Aringa, Kalenjin, Pokot, and Samburu–while GlotLID leads on several Nilotic languages where it has coverage (e.g., Kakwa 99.7%, Kumam 99.1%, Marakwet 99.0%). Among Bantu languages, the two models split wins: PAWALID is stronger on languages with limited external coverage, whereas GlotLID achieves higher F1 on languages such as Makonde (98.8%) and Gusii (96.5%) that appear in its training data.

5.2 African-centric Evaluation

Table 2 shows results for African-centric benchmarks, defined as African-language subsets of broader multilingual benchmarks. GlotLID leads in all-F1, due to broader language coverage (e.g., 42/42 languages on FLORES+-Afr versus 37/42 for PAWALID). However, PawaLID's cov-F1 performance shows that PAWALID remains competitive on supported languages: 85.2% vs. 96.9% on FLORES+-Afr and 90.1% vs. 92.6% on UDHR-LID-Afr. On CommonLID-Africa, the gap narrows to 2.3 points in all-F1 (61.3% vs. 63.8%), with PawaLID covering 22/23 languages compared

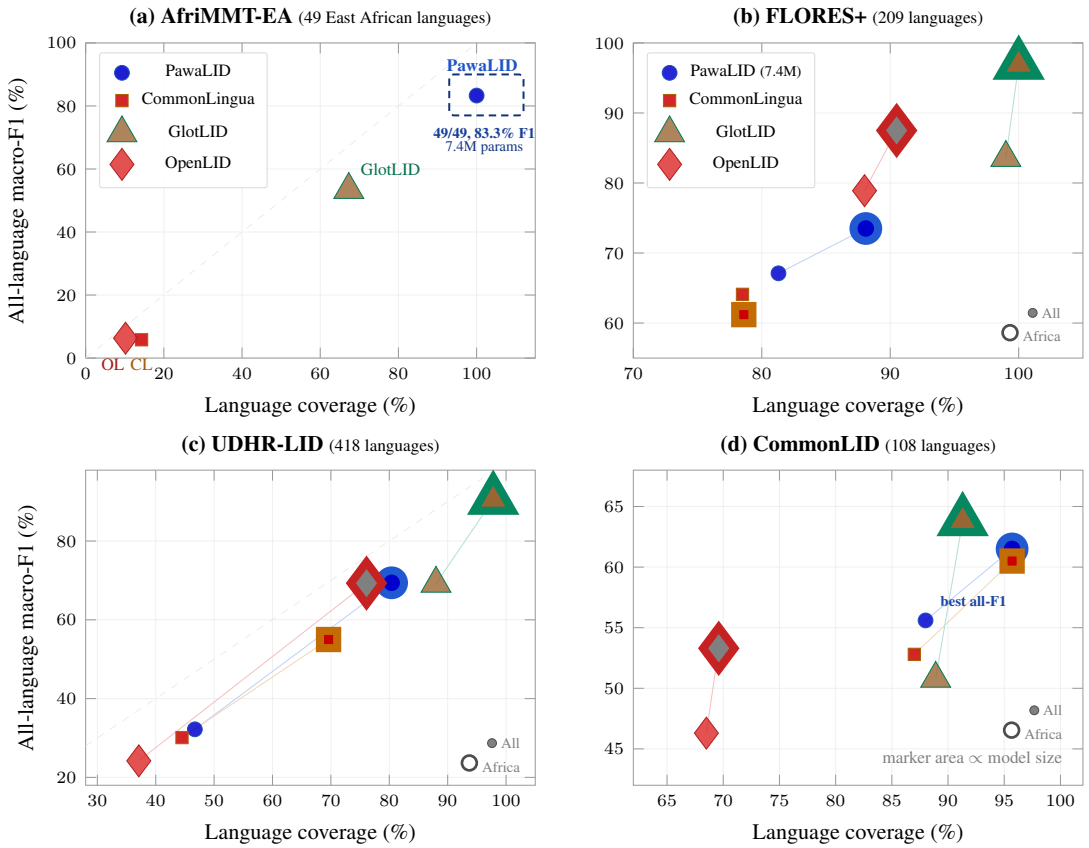

Figure 4: Coverage–accuracy Pareto view across LID benchmarks. Panel (a) shows the full benchmark set, while Panel (b) zooms into the African-centric high-coverage region. The x-axis shows language coverage, and the y-axis shows all-language macro-F1. Colours and marker shapes indicate models, and marker size approximates model footprint. Thin lines connect the same benchmark across models.

to GlotLID's 21/23. CommonLingua and OpenLID generally trail PAWALID and GlotLID, with OpenLID achieving the highest cov-F1 on several subsets. To provide a controlled per-language comparison, we pair PawaLID against CommonLingua in the African-filtered benchmarks. This pairing is motivated by architectural and scale similarity: both models are compact classifiers with fewer than 10M parameters, making them the most directly comparable systems in our evaluation. GlotLID and OpenLID, by contrast, operate at substantially different model scales and coverage breadth, making per-language comparisons less informative for architectural comparison. Full per-language F1 scores are reported in Appendix: FLORES+-Afr (Table 10), UDHR-LID-Afr (Table 11), and CommonLID-Afr (Table 12). Across all three subsets, PawaLID achieves the highest F1 for most of the languages it covers, particularly large margins for East African such as Luganda, Swahili, Kinyarwanda, and Luo. Both models struggle on Bemba (`bem`) and West Central Oromo (`gaz`), the latter likely reflecting a macrolanguage label conflict with `orm` rather than a classification failure.

5.3 General or Broad Benchmarks

Table 2 shows GlotLID leads on FLORES+ (83.7%) and UDHR-LID (69.2%), reflecting its broad language coverage (1,665). PAWALID achieves the highest all-F1 on CommonLID (55.6% vs. 50.8% for GlotLID), a benchmark characterized by domain diversity and noisy real-world text. PAWALID's lower all-F1 on FLORES+ and UDHR-LID is primarily driven by coverage rather than accuracy: its cov-F1 of 85.5% and 81.0%, respectively, shows that the model classifies covered languages well. The gap between all-F1 and cov-F1 underscores the effect of unsupported languages.

5.4 Ablation on CommonLID

Table 3 and Figure 3 summarize the main ablation progression on CommonLID. Increasing the hashed byte n-gram space from 4,096 to 32,768 buckets yields negligible benefit (+0.01 macro-F1).

Cluster	Target labels	Motivation
Arabic varieties	arb, arz, ary	Bidirectional confusion among MSA, Egyptian Arabic, and Moroccan Arabic
Austronesian	ind, msa, jav, ace	High-volume Indonesian–Malay confusion
Sinitic	zho, yue, wuu	Cantonese and Wu confused with Chinese macrolanguage label
Niger-Congo/Bantu	lin, yor, swh, sna	Lingala misclassified as Yoruba
Germanic/Latin absorber	afr, eng, deu, nld, fry	Afrikaans over-predicted on unrelated Latin-script text

Table 5: Confusion-driven hard-negative groups derived from CommonLID error analysis.

Error	Examples	Cause
Related varieties	arb/arz/ary	dialect and register overlap
Macrolanguage	fil/tgl, gaz/orm	benchmark label granularity
Lexical overlap	ind/msa	shared vocabulary and script
Domain mismatch	lin	train–test domain shift
Low support	zul, kor	few evaluation examples

Table 6: Main error categories observed in PAWALID Macrolanguage mismatch,Low support Instability.

So we retain 4,096 buckets in the final model. The largest improvement comes from byte-level length augmentation, increasing macro-F1 from 66.44 to 69.00 and weighted-F1 from 82.77 to 87.71, confirming that CommonLID exposes length and domain mismatches not captured by internal validation alone. Label smoothing provides a smaller additional gain, raising macro-F1 from 69.00 to 69.07 and weighted-F1 from 87.71 to 87.80 (See §4.5). The ablation study uses CommonLID macro-F1 as a diagnostic metric to compare configuration choices (hash capacity, length augmentation, label smoothing), but checkpoint selection itself is based on AfriMMT-EA validation macro-F1. The CommonLID ablation in Figure 3 reports native-coverage macro-F1 on the subset of examples whose labels are supported by the checkpoint, whereas Table 2 reports both all-language macro-F1, assigning zero to unsupported labels, and covered-language macro-F1 under final cross-system evaluation protocol.

5.5 Coverage–Accuracy Tradeoff

Table 2 reveals a consistent pattern across models. OpenLID achieves the highest cov-F1 on most benchmarks (73–98%) but the lowest all-F1 due to its limited language coverage. GlotLID achieves the highest all-F1 on FLORES+ and UDHR-LID due to broad language coverage (see (§5.3). PAWALID fills an important gap on this frontier: strong cov-F1 with African-centric coverage that remains underserved by existing systems. Figure 4 illustrates this tradeoff: shows broad multilingual coverage and targeted African-centric language support occupy different design space.

6 Discussion

Three findings emerge from the evaluation. First, coverage drives all-language F1 more than per-language accuracy. The gap between all-F1 and cov-F1 is consistently larger for models with narrower inventories, confirming that aggregate scores alone can be misleading when systems differ in language support. Expanding PAWALID's coverage to additional Southern and West African languages is the most direct path to improving all-F1 on FLORES+ and UDHR-LID. Second, internal validation overestimates generalization. Although we select checkpoints using AfriMMT-EA validation macro-F1, we monitor CommonLID macro-F1 per-epoch as an external diagnostic. The ablation study (§5.4) shows that length augmentation closes a substantial gap between internal and external macro-F1 without additional data, suggesting that LID systems for low-resource languages should always report out-of-domain benchmarks alongside internal validation. Third, the benchmarks themselves conflate several phenomena: genuine language confusion, macrolanguage label mismatch, short-text ambiguity, domain mismatch, and low-support instability (e.g., Korean with 8 examples, Zulu with 4). The field would benefit from African-focused LID.

7 Conclusion

PAWALID is a compact 7.4M-parameter byte-level LID model designed for African and low-resource settings. It combines UTF-8 byte embeddings, Unicode-aware byte features, hashed byte n-grams, multi-scale convolutions, and lightweight self-attention, avoiding dependence on language-specific tokenizers while remaining suitable for deployment. Results show that African-language LID should be evaluated beyond accuracy alone. On AfriMMT-EA, PAWALID is the only evaluated model covering all 49 East African languages and achieves the strongest all-language macro-F1. It also achieves the highest all-language macro-F1 on CommonLID while remaining competitive on African-focused benchmarks. PAWALID provides a compact byte-level approach tailored for African.

Limitations

PAWALID has several limitations. First, although it improves coverage for East African languages, its 398-label inventory still covers only a small fraction of the world's linguistic diversity and remains narrower than broad-coverage systems such as GlotLID. Expanding coverage to additional Southern, Central, and West African languages is therefore an important direction for future work. Second, some benchmark labels correspond to closely related varieties, macrolanguage distinctions, or partially overlapping ISO categories. In such cases, strict label-level evaluation may penalize predictions that are linguistically close or practically acceptable in some deployment settings. Future work should explore hierarchical and family-aware evaluation in addition to flat ISO-level scoring. Third, although the architecture includes mixed-language and unknown-language detection heads, these objectives are not fully exploited in the current experiments because reliable supervision for code-switched and open-set examples remains limited. Better training data for mixed-language and out-of-distribution detection would improve robustness in realistic African multilingual environments. Finally, our strongest results are obtained on sentence-level benchmark data. Further evaluation is needed on noisier sources such as web-crawled text, OCR outputs, social media, and ASR transcripts, where LID errors often have the greatest downstream impact.

Ethical Considerations

Language identification systems can improve inclusion by making underrepresented languages visible in corpus construction and multilingual NLP pipelines. However, LID errors may also exclude communities from downstream systems or misroute their text into inappropriate processing pipelines. This risk is especially important for closely related African languages and for code-switched text. We therefore report both coverage and accuracy, explicitly identify unsupported languages, and recommend human review when the model is used for high-stakes corpus filtering or institutional decision-making. The model should not be used as the sole basis for determining a speaker's identity, ethnicity, nationality, or eligibility for services.

Use of AI Assistants

We use Claude Code (Anthropic) and ChatGPT for code debugging, editing assistance, and brainstorming visualization or diagram ideas. All scientific decisions, experimental design, analysis, interpretation of results, and final manuscript content were reviewed, validated, and approved by the authors

License The dataset will be released under the Creative Commons Attribution (CC BY 4.0) license, as long as appropriate credit is given to the original creators

References

Ife Adebara, AbdelRahim Elmadany, Muhammad Abdul-Mageed, and Alcides Inciarte. 2022. AfroLID: A neural language identification tool for African languages. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1958–1981, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. 2020. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7096–7116.

Jan A. Botha, Emily Pitler, Ji Ma, Anton Bakalov, Alex Salcianu, David Weiss, Ryan McDonald, and Slav Petrov. 2017. Natural language processing with small feed-forward networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2879–2885. Association for Computational Linguistics.

Laurie Burchell, Alexandra Birch, Nikolay Bogoychev, and Kenneth Heafield. 2023. An open dataset and model for language identification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 865–879. Association for Computational Linguistics.

Isaac Caswell, Theresa Breiner, Daan van Esch, and Ankur Bapna. 2020. Language ID in the wild: Unexpected challenges on the path to a thousand-language web text corpus. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6588–6608. International Committee on Computational Linguistics.

William B Cavnar, John M Trenkle, and 1 others. 1994. N-gram-based text categorization. In *Proceedings of SDAIR-94, 3rd annual symposium on document analysis and information retrieval*, volume 161175, page 14. Las Vegas, NV.

Jonathan H Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an efficient

tokenization-free encoder for language representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 8440–8451.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Naome A Etori, Kelechi Ezema, Nathaniel Romney Robinson, Davis David, Alfred Malengo Kondoro, Elisha Ondieki Makori, Michael Samwel Mollel, and Maria Gini. 2026. AfriMMT-EA: Multi-domain machine translation for low-resource East African languages. In *Findings of the Association for Computational Linguistics: EACL 2026*, pages 3459–3492, Rabat, Morocco. Association for Computational Linguistics.

René Haas and Leon Derczynski. 2021. Discriminating between similar Nordic languages. In *Proceedings of the Eighth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 67–75, Kiyv, Ukraine. Association for Computational Linguistics.

International Telecommunication Union. 2024. Measuring digital development: Facts and figures 2024.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext. zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomáš Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th conference of the European chapter of the association for computational linguistics: volume 2, short papers*, pages 427–431.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schütze. 2023. GlotLID: Language identification for low-resource languages. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

András Kornai. 2013. Digital language death. *PLOS ONE*, 8(10):e77056.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, and 1 others. 2022a. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan Van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, and 1 others. 2022b. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Michael McCandless. 2010. Compact language detector 2.

Shuyo Nakatani. 2010. Language detection library for java. `https://github.com/shuyo/language-detection`.

Goodwill Erasmo Ndomba, Medard Edmund Mswahili, and Young-Seob Jeong. 2024. Tokenizers for african languages. *IEEE Access*, 13:1046–1054.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2024. Scaling neural machine translation to 200 languages. *Nature*, 630(8018):841–846.

Jessica Ojo, David Ifeoluwa Adelani, Pedro Araujo, and 1 others. 2025. AfroBench: How good are large language models on African languages? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17569–17608. Association for Computational Linguistics.

Ermelinda Oro, Massimo Ruffolo, and Mostafa Sheikhalishahi. 2018. Language identification of similar languages using recurrent neural networks. In *ICAART (2)*, pages 635–640.

Pedro Ortiz Suarez, Laurie Burchell, Catherine Arnett, Rafael Mosquera-Gómez, Sara Hincapie-Monsalve, Thom Vaughan, Damian Stewart, Malte Ostendorff, Idris Abdulmumin, Vukosi Marivate, and 1 others. 2026. Commonlid: Re-evaluating state-of-the-art language identification performance on web data. *arXiv preprint arXiv:2601.18026*.

PleIAs. 2026. Commonlingua: Byte-level language identification for 334 languages.

Alex Salcianu, Andy Golding, Anton Bakalov, Chris Alberti, and 1 others. 2018. Compact language detector v3.

Noam Shazeer. 2020. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Thapelo Andrew Sindane and Vukosi Marivate. 2024. From n-grams to pre-trained multilingual models for language identification. In *Proceedings of the 4th International Conference on Natural Language Processing for Digital Humanities*, pages 229–239, Miami, USA. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. Byt5: Towards a token-free future with pre-trained byte-to-byte models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

A Related Work

Early language identification (LID) systems relied on character and byte n-gram statistics with similarity measures such as rank-order distance (Cavnar et al., 1994). Systems including Langdetect (Nakatani, 2010), FastText LID (Joulin et al., 2016, 2017), and the CLD series (McCandless, 2010; Salcianu et al., 2018) expanded language coverage through lightweight statistical and subword-based approaches. However, these systems often struggle with short text and low-resource languages due to skewed training distributions and limited African-language representation (Oro et al., 2018; Sindane and Marivate, 2024). Neural approaches model LID as a downstream classification task using multilingual encoders such as mBERT (Devlin et al., 2018) and XLM-R (Conneau et al., 2020). While benefiting from large-scale multilingual pretraining, subword tokenizers learned primarily from high-resource data can produce unstable segmentation and out-of-vocabulary failures for underrepresented African languages and scripts (Ndomba et al., 2024; Bhattamishra et al., 2020). This limitation has motivated tokenizer-free byte-level models such as ByT5 (Xue et al., 2022) and CANINE (Clark et al., 2022), which reduce tokenizer dependence while preserving multilingual performance. Unlike prior approaches, PAWALID targets a complementary design point by combining compact byte-level modeling with explicit coverage-aware evaluation and stronger support for East African languages frequently underrepresented in existing LID inventories.

B PawaLID Data Construction

First, we convert the AfriMMT-EA (Etori et al., 2026) into sentence-level LID examples by treating each source and target sentence as an independent training instance labelled with its ISO language code. We then used a 49-language subset as our main source of East African languages for training. We exclude English (`eng`), Swahili (`swh`), and Kikuyu (`kik`) from the AfriMMT-EA training subset via label filtering, as these languages are overrepresented in CommonLingua and would otherwise dominate the East African training mixture. After these exclusions, 50 languages remain in principle; however, Tigrinya (`tir`) contains a single example of only 8 characters, which falls below the minimum-length threshold once (`paper_normalize_text`) strips digits and punctuation, leaving 49 languages in the final training set. The AfriMMT-EA evaluation benchmark uses the test split as provided by the original dataset, covering all 49 non-excluded languages. The goal is to provide stronger coverage for East African languages and to support the primary goal of African-language corpus filtering and routing. **Second**, we use the CommonLingua (PleIAs, 2026), to broaden coverage beyond East African languages. This is important because LID systems deployed in real-world settings frequently encounter non-African languages, mixed or code-switched web text, short inputs, and typologically related language varieties. **Third**, we include the PawaLIDAfrica dataset to boost African-language coverage beyond the original East African MT corpus.

C Train–Evaluation Separation and Decontamination

Because AfriMMT-EA is used both as a source of East African LID supervision and as the basis for the AfriMMT-EA evaluation benchmark, we explicitly separate training, validation, and test examples before model training. Splits are constructed at the sentence level, and evaluation examples are excluded from the training pool before any sampling, augmentation, or deduplication is applied. We then perform exact text–label deduplication across the merged training mixture and remove any training example whose normalized text appears in an evaluation split with the same language label. For normalization during decontamination, we lowercase text where appropriate, normalize Unicode to NFC, collapse repeated whitespace, and strip leading and trailing punctuation-only artifacts. We apply this procedure to AfriMMT-EA, CommonLingua, and PawaLIDAfrica before constructing the final training mixture. The AfriMMT-EA test split is not used for checkpoint selection; checkpoint selection is based on the AfriMMT-EA validation split macro-F1, as described in §D. This separation is important because the AfriMMT-EA evaluation is intended to measure East African language coverage and generalization rather than memorization of parallel-corpus sentences. We therefore report AfriMMT-EA results only on held-out examples that do not appear in the training mixture after normalization as shown in Table 1. The released checkpoint and deduplicated training dataset are intended to support exact reproduction of the reported experiments.

D Training Setup

Table 7 summarizes the full training configuration used for the reported PAWALID checkpoint. Unless otherwise stated, all final PAWALID models use a maximum sequence length of 512 UTF-8 bytes, 4,096 hashed byte n-gram buckets per order, a batch size of 512, an evaluation batch size of 1,024, dropout of 0.25, and bfloat16 precision. We optimize with AdamW using a learning rate of 7×10^{-5} and a warmup ratio of 0.12 for six epochs. The primary language-classification objective uses weighted cross-entropy. In the strongest configuration, we combine it with an auxiliary language-family classification loss with weight $\lambda_f = 0.03$. We set the mixed-language and unknown-language loss weights to zero, $\lambda_m = \lambda_u = 0$, because the current training data does not yet provide reliable supervision for these auxiliary heads. We select checkpoints based on macro-F1 on the AfriMMT-EA validation split evaluated after each epoch. The AfriMMT-EA test split is held out entirely and used only for final evaluation. CommonLID is evaluated per-epoch as an external diagnostic signal to monitor out-of-domain generalization and to guide ablation decisions (§5.4), but it does not influence checkpoint selection. The CommonLID test results reported in Table 2 are therefore fully held-out with respect to both training and model selection. The released checkpoint and deduplicated training dataset are intended to support exact reproduction of the reported experiments

D.1 Error Analysis

Table 6 summarizes the main error categories. Despite strong aggregate results, PAWALID exhibits several systematic error patterns. First, closely related varieties remain challenging: Egyptian Arabic, Modern Standard Arabic, and Moroccan Arabic form a bidirectional confusion cluster, and Indonesian–Malay confusion accounts for a large share of CommonLID errors. Second, some apparent errors reflect label-granularity mismatch rather than purely model failure, such as `fil` versus `tgl` and `gaz` versus `orm`. We therefore report strict ISO-based scores but also flag these cases, as practical deployments may prefer merged or hierarchical label decisions. Third, Lingala errors suggest a domain mismatch between training data and CommonLID evaluation examples, indicating that additional formal-domain Lingala data would be more useful than simply increasing model capacity.

E Datasets

F Reproducibility

Category	Details
Training Data	
Primary corpus	AfriMMT-EA (Etori et al., 2026); MT pairs converted to sentence-level LID examples (49 languages)
Broad-coverage corpus	CommonLingua (PleIAs, 2026); 334 languages
African supplement	PawaLIDAfrica; 32 African languages
Sampling caps	Max. 300k train / 100k val. examples per language per extra corpus; min. text length 5 chars
Hard-negative augmentation	GlotLID corpus for 13 targeted labels (e.g. `afr`, `arb`, `yue`, `wuu`, `rcf`); max. 3k train / 100 val. per label; min. length 80 bytes; deduplicated
Architecture	
Input representation	512 UTF-8 bytes; byte n-grams ($n \in \{2, 3, 4\}$) hashed into 4,096 buckets
Encoder	Hidden dim. 256; 4 multi-scale depthwise-separable conv. blocks; 2 self-attention blocks; 4 heads
Output heads	Language (N_{lang} classes), language family (N_{fam}), mixed-language (binary), unknown-language (binary)
Parameters	7.4M
Optimization	
Optimizer	AdamW; lr 7×10^{-5}; weight decay 0.05; cosine schedule; warmup ratio 0.12; max. grad. norm 1.0
Batch size	512 (train) / 1,024 (eval)
Training duration	6 epochs; bfloat16 precision; seed 42
Regularization	
Dropout	0.25
Label smoothing	$\epsilon = 0.03$ on language head
Length augmentation	Random byte-span crop (prob. 0.5; range 20–400 bytes; applied when input >80 bytes); random-window truncation
Loss Weights	
Language family (λ_f)	0.03
Mixed-language (λ_m)	0.0 (head present but not optimized)
Unknown-language (λ_u)	0.0 (head present but not optimized)
Hard-negative loss	Weight 0.06; applied every 2 steps; batch size 128; pool 5k samples/label
Checkpoint Selection	
Checkpoint selection	AfriMMT-EA validation split macro-F1 evaluated after each epoch
External diagnostic	CommonLID test split macro-F1 evaluated per epoch for ablation analysis (not used for selection)

Table 7: Training and reproducibility configuration for the reported PAWALID model. Mixed-language and unknown-language heads are present but receive zero loss weight in this configuration.

Language	ISO	Train	Val	Test	Total
Acholi	ach	19,930	2,491	2,491	24,912
Amharic	amh	5,075	634	634	6,343
Egyptian Arabic	arz	11,161	1,394	1,394	13,949
Bambara	bam	4,800	599	599	5,998
Ghomala	bbj	3,753	469	469	4,691
Ewe	ewe	3,988	498	498	4,984
Fon	fon	4,347	543	543	5,433
Hausa	hau	12,623	1,577	1,577	15,777
Kinyarwanda	kin	1,172	146	146	1,464
Lugbara	lgg	19,935	2,491	2,491	24,917
Lingala	lin	4,323	540	540	5,403
Luganda	lug	27,280	3,410	3,410	34,100
Luo	luo	5,795	724	724	7,243
Mossi	mos	4,442	555	555	5,552
Chichewa	nya	1,191	148	148	1,487
Runyankole	nyn	19,935	2,491	2,491	24,917
Oromo	orm	3,334	416	416	4,166
Nigerian Pidgin	pcm	7,917	989	989	9,895
Rundi	run	2,552	319	319	3,190
Shona	sna	4,914	614	614	6,142
Somali	som	2,342	292	292	2,926
Swahili	swa	757	94	94	945
Ateso	teo	19,925	2,490	2,490	24,905
Tigrinya	tir	2,169	271	271	2,711
Setswana	tsn	3,950	493	493	4,936
Rutooro	ttj	19,937	2,491	2,491	24,919
Twi	twi	5,413	676	676	6,765
Wolof	wol	5,849	731	731	7,311
isiXhosa	xho	3,501	437	437	4,375
Lusoga	xog	19,942	2,492	2,492	24,926
Yoruba	yor	11,749	1,468	1,468	14,685
Zulu	zul	9,302	1,162	1,162	11,626

Table 8: PawaLIDAfrica Dataset statistics showing train, validation, and test splits across languages.

Table 9: Per-language F1 (%) on AfriMMT-EA. Best score per language in **bold**; 0.0 = unsupported or zero F1.

ISO	Language	Family	Sup	PawaLID	AfroLID	GlotLID	CommonLingua	OpenLID
bez	Bena	Bantu	867	**94.0**	29.1	0.0	0.0	0.0
old	Chaga	Bantu	990	**79.7**	0.0	3.4	0.0	0.0
coh	Chonyi	Bantu	27	0.0	0.0	0.0	0.0	0.0
cuh	Chuka	Bantu	501	**88.5**	0.0	0.0	0.0	0.0
dug	Duruma	Bantu	501	87.6	48.5	**95.0**	0.0	0.0
ebu	Embu	Bantu	530	**89.9**	46.7	0.0	0.0	0.0
nyf	Giriama	Bantu	960	**86.8**	64.8	79.4	0.0	0.0
gog	Gogo	Bantu	3204	**93.7**	90.2	92.9	0.0	0.0
guz	Gusii	Bantu	648	94.6	94.5	**96.5**	0.0	0.0
gwr	Gwere	Bantu	624	76.3	79.8	**94.0**	0.0	0.0
hay	Haya	Bantu	126	**49.1**	21.6	42.6	0.0	0.0
heh	Hehe	Bantu	784	**92.0**	73.9	65.8	0.0	0.0
jit	Jita	Bantu	880	**89.9**	78.1	0.0	0.0	0.0
kam	Kamba	Bantu	727	73.1	20.5	74.1	0.0	**74.3**
kin	Kinyarwanda	Bantu	624	**94.7**	93.1	41.0	22.1	28.6
lag	Langi	Bantu	783	**94.0**	0.0	0.0	0.0	0.0
lug	Luganda	Bantu	15128	**94.4**	92.1	**94.4**	70.8	82.4
luy	Luhya	Bantu	1949	**70.7**	0.0	0.0	0.0	0.0
kde	Makonde	Bantu	910	93.5	94.9	**98.8**	0.0	0.0
myx	Masaba	Bantu	624	**94.2**	93.6	83.7	0.0	0.0
mer	Meru	Bantu	540	**88.9**	29.9	69.2	0.0	0.0
nuj	Nyole	Bantu	624	94.3	88.1	**96.3**	0.0	0.0
nyo	Nyoro	Bantu	624	**93.8**	74.6	5.2	0.0	0.0
asu	Pare	Bantu	100	0.0	0.0	0.0	0.0	0.0
pkb	Pokomo	Bantu	499	91.7	86.6	**94.3**	0.0	0.0
xog	Soga	Bantu	624	66.4	79.7	**90.5**	0.0	0.0
sxb	Suba	Bantu	795	95.1	93.5	**95.7**	0.0	0.0
suk	Sukuma	Bantu	635	**88.3**	6.4	68.6	0.0	0.0
dav	Taita	Bantu	5348	**80.2**	45.3	0.0	0.0	0.0
tvs	Taveta	Bantu	523	**84.5**	0.0	0.0	0.0	0.0
zak	Zanaki	Bantu	781	**98.8**	0.0	0.0	0.0	0.0
gax	Borana	Cushitic	954	**75.7**	54.0	0.0	0.0	0.0
orm	Oromo	Cushitic	333	**85.6**	70.7	0.0	26.0	0.0
som	Somali	Cushitic	499	92.0	**94.2**	94.0	91.4	86.5
ach	Acholi	Nilotic	624	91.7	5.6	**97.8**	14.7	0.0
alz	Alur	Nilotic	624	96.6	78.7	**97.4**	0.0	0.0
luc	Aringa	Nilotic	622	**98.4**	0.0	0.0	0.0	0.0
adh	Jopadhola	Nilotic	624	91.7	79.3	**97.4**	0.0	0.0
keo	Kakwa	Nilotic	624	98.1	0.0	**99.7**	0.0	0.0
kln	Kalenjin	Nilotic	758	**83.8**	60.2	0.0	0.0	0.0
ndp	Kebu	Nilotic	624	**95.5**	0.0	71.0	0.0	0.0
kdi	Kumam	Nilotic	624	96.9	96.3	**99.1**	0.0	0.0
laj	Lango	Nilotic	624	94.0	89.2	**98.0**	0.0	0.0
luo	Luo	Nilotic	946	92.3	88.1	**97.2**	10.6	38.4
mas	Maasai	Nilotic	528	**91.7**	80.0	89.0	0.0	0.0
enb	Marakwet	Nilotic	523	97.0	0.0	**99.0**	0.0	0.0
pko	Pokot	Nilotic	621	**73.5**	67.7	0.0	0.0	0.0
saq	Samburu	Nilotic	537	**21.6**	0.7	0.0	0.0	0.0
teo	Teso	Nilotic	624	95.2	89.4	**96.7**	0.0	0.0

ISO	Pawa	Afro	CL	Sup.
Bantu				
bem	0.0	**92.7**	–	997
kam	**67.8**	51.3	–	997
kik	81.6	**92.3**	76.5	997
kin	78.2	**86.4**	61.8	997
lin	92.4	**98.3**	80.4	997
lua	–	**97.1**	–	997
lug	91.2	**98.8**	61.2	997
nso	92.2	85.7	**93.9**	997
nya	85.8	**93.4**	80.5	997
run	69.3	**81.7**	26.5	997
sna	98.5	**99.0**	88.3	997
sot	**95.2**	5.3	89.7	997
ssw	93.9	**94.7**	76.6	997
swh	92.2	**92.7**	55.7	997
tsn	**92.7**	60.7	89.7	997
tso	90.6	**99.3**	86.8	997
umb	–	**77.7**	–	997
xho	84.5	**92.1**	66.2	997
zul	**86.2**	81.4	58.1	997
Niger-Congo				
bam	**77.0**	0.4	58.6	997
dyu	–	**8.9**	–	997
ewe	**99.6**	98.3	87.5	997
fon	**99.3**	93.4	64.4	997
fuv	–	**34.8**	–	997
ibo	95.7	**99.4**	94.3	997
mos	**93.2**	92.9	–	997
sag	–	**94.9**	–	997
twi	**98.9**	96.9	–	1,994
wol	**98.8**	52.0	91.7	997
yor	97.3	**98.4**	76.1	997
Semitic				
amh	99.0	**99.9**	99.4	997
arb	**42.5**	–	39.2	1,994
ary	**64.0**	–	33.8	997
arz	**59.7**	–	9.3	997
tir	99.0	**99.9**	99.4	997
Cushitic				
gaz	2.2	0.0	**20.6**	997
som	96.6	82.3	**97.0**	997
Chadic				
hau	84.2	**90.6**	84.4	997
Nilotic				
luo	96.2	**97.5**	41.7	997
Germanic				
afr	97.3	**99.6**	78.0	997
Other				
kbp	**99.9**	99.8	93.6	997
mlt	**99.4**	–	97.0	997
Macro-avg	**73.62**	71.91	56.14	

Table 10: Per-language F1 (%) on FLORES+-Afr. Pawa = PawaLID (37/42); Afro = AfroLID (38/42); CL = CommonLingua (33/42). "–" = not covered. Best per row in **bold**.

ISO	Pawa	Afro	CL	Sup.
Bantu				
bem	0.0	**91.4**	–	57
kde	88.1	**91.5**	–	51
kin	76.2	**88.7**	57.6	59
lin	92.0	**99.1**	91.0	110
lua	–	**97.3**	–	58
lug	78.4	**95.1**	35.8	49
nbl	–	**72.4**	–	58
nso	89.7	91.9	**98.2**	58
nya	**93.8**	93.1	82.2	116
nyn	**96.9**	90.1	0.0	50
run	65.2	**86.3**	38.9	58
sna	**100.0**	98.3	84.5	60
sot	**95.7**	6.8	95.0	57
ssw	95.9	**96.6**	93.2	60
suk	**88.9**	43.2	–	58
swh	94.7	**96.5**	44.4	58
tsn	92.1	63.4	**94.8**	58
tso	88.9	**93.0**	88.3	114
umb	–	**90.3**	–	107
ven	**100.0**	97.8	95.9	89
xho	76.0	**88.9**	73.3	58
zul	62.7	**78.6**	46.5	58
Niger-Congo				
bam	**94.2**	0.0	59.4	57
dyu	–	**15.1**	–	58
ewe	**100.0**	**100.0**	90.7	59
fon	72.0	**90.2**	80.8	58
fuv	–	**56.6**	–	108
ibo	**100.0**	99.1	**100.0**	58
mos	**100.0**	**100.0**	–	56
sag	–	**6.6**	–	59
twi	**99.6**	99.1	–	116
wol	81.6	56.1	**90.8**	60
yor	**98.3**	97.4	87.0	57
Semitic				
amh	99.0	**100.0**	99.0	51
arb	71.1	–	**85.1**	58
tir	99.1	**100.0**	**100.0**	58
Cushitic				
gaz	6.5	0.0	**46.2**	60
som	**96.6**	79.3	93.9	57
Chadic				
hau	95.7	**99.0**	94.1	155
Creole				
pcm	**100.0**	93.3	63.7	56
Germanic				
afr	**100.0**	**100.0**	89.9	60
Other				
kbp	**100.0**	**100.0**	81.2	59
loz	–	**99.1**	–	58
mlt	**100.0**	–	99.1	59
snk	–	–	–	58
toi	–	**96.4**	–	57
Macro-avg	69.33	**74.73**	53.93	

Table 11: Per-language F1 (%) on UDHR-LID-Afr. Pawa = PawaLID (37/46); Afro = AfroLID (43/46); CL = CommonLingua (32/46). "–" = not covered. Best per row in **bold**.

ISO	PawaLID	AfroLID	CL	Sup.
Bantu				
kik	**93.4**	90.3	53.0	62
lin	0.0	0.0	0.0	55
lug	90.5	**95.5**	83.8	1,359
nso	75.7	67.5	**78.6**	99
nyn	0.0	**50.0**	0.0	5
sna	**87.7**	85.1	81.0	1,355
sot	96.1	78.3	**96.3**	943
swh	91.2	**96.7**	92.4	16,412
xho	88.7	**92.2**	78.7	575
zul	**2.5**	0.0	0.4	4
Niger-Congo				
fuv	–	**1.4**	–	36
ibo	**91.6**	77.9	87.0	168
yor	**94.2**	91.3	87.1	2,290
Semitic				
amh	**98.8**	98.2	95.0	1,616
arb	80.5	–	**87.8**	26,148
ary	35.6	–	**56.0**	226
arz	18.5	–	**33.5**	1,102
Cushitic				
gaz	0.0	0.0	0.0	11
orm	**95.6**	9.0	75.0	1,060
Chadic				
hau	94.2	**95.0**	81.3	16,453
Creole				
pcm	**1.5**	0.0	0.0	2
Germanic				
afr	81.4	**88.0**	68.7	88
Austronesian				
mlg	**92.8**	82.4	86.1	2,152
Macro-avg	**61.33**	52.12	57.46	

Table 12: Per-language F1 (%) on CommonLID-Afr. Pawa = PawaLID (22/23); Afro = AfroLID (20/23); CL = CommonLingua (22/23). "–" = not covered. Best per row in **bold**.

Region / Category	Count	%
Africa	109	27.4
Asia	142	35.7
Europe	113	28.4
Americas	16	4.0
Oceania	9	2.3
Constructed / Other	9	2.3
Total	**398**	

Table 13: Distribution of PawaLID's 398 language labels by geographic region. African languages constitute the largest single-region group, reflecting the model's design focus.

ISO	Language	Sub-family
Bantu (38)		
asu	Asu (Pare)	E
bem	Bemba	M
bez	Bena	G
coh	Chonyi	E
cuh	Chuka	E
dav	Taita	E
dug	Duruma	E
ebu	Embu	E
gog	Gogo	G
guz	Gusii	E
gwr	Gwere	J
hay	Haya	J
heh	Hehe	G
jit	Jita	J
kde	Makonde	P
kik	Kikuyu	E
kin	Kinyarwanda	J
kon	Kongo	H
lag	Langi	F
lin	Lingala	C
lug	Luganda	J
luy	Luyia	J
mer	Meru	E
myx	Masaaba	J
nso	Northern Sotho	S
nuj	Nyole	J
nya	Chichewa	N
nyf	Giryama	E
nyn	Nyankole	J
nyo	Nyoro	J
old	Mochi (Chaga)	E
pkb	Pokomo	E
run	Rundi	J
sna	Shona	S
sot	Southern Sotho	S
ssw	Swati	S
suk	Sukuma	F
swh	Swahili	G
sxb	Suba	J
teo	Teso (Bantu)	–
tsn	Tswana	S
tso	Tsonga	S
ttj	Tooro	J
tum	Tumbuka	N
tvs	Taveta	E
ven	Venda	S
xho	isiXhosa	S
xog	Soga	J
zak	Zanaki	J
zul	isiZulu	S

ISO	Language	Sub-family
Nilotic (16)		
ach	Acholi	Western
adh	Jopadhola	Western
alz	Alur	Western
din	Dinka	Western
dip	NE Dinka	Western
enb	Markweeta	Southern
kdi	Kumam	Western
keo	Kakwa	Eastern
kln	Kalenjin	Southern
laj	Lango	Western
luc	Aringa	Central Sudanic
luo	Luo	Western
mas	Maasai	Eastern
ndp	Ndo (Kebu)	Central Sudanic
pko	Pökööt	Southern
saq	Samburu	Eastern
Niger-Congo / West Atlantic (18)		
aka	Akan	Kwa
bam	Bambara	Mande
bbj	Ghomala'	Grassfields
dag	Dagbani	Gur
dga	S. Dagaare	Gur
ewe	Ewe	Kwa
fat	Fanti	Kwa
fon	Fon	Kwa
ful	Fulah	Atlantic
gpe	Ghanaian Pidgin	Creole
gur	Farefare	Gur
guw	Gun	Kwa
ibo	Igbo	Volta-Niger
kbp	Kabiyè	Gur
kcg	Tyap	Plateau
kpe	Kpelle	Mande
mos	Mossi	Gur
nqo	N'Ko	Mande (script)
twi	Twi	Kwa
wol	Wolof	Atlantic
yor	Yoruba	Volta-Niger
Afro-Asiatic (14)		
amh	Amharic	Semitic
arq	Algerian Arabic	Semitic
ary	Moroccan Arabic	Semitic
arz	Egyptian Arabic	Semitic
gax	Borana Oromo	Cushitic
gaz	W.C. Oromo	Cushitic
gez	Ge'ez	Semitic
hau	Hausa	Chadic
kab	Kabyle	Berber
orm	Oromo	Cushitic
shi	Tachelhit	Berber
som	Somali	Cushitic
tir	Tigrinya	Semitic
zgh	Std. Moroccan Tamazight	Berber
Other African (13)		
afr	Afrikaans	Germanic
egx	Egyptian langs	Ancient
egy	Ancient Egyptian	Ancient
kam	Kamba	Bantu (E)
lgg	Lugbara	Central Sudanic
mlg	Malagasy	Austronesian
mlt	Maltese	Semitic
pcm	Nigerian Pidgin	Creole
rcf	Réunion Creole	Creole

Table 14: African languages in PawaLID's label inventory (109 of 398 total), grouped by language family. Bantu sub-family codes follow Guthrie zones. Nilotic and Niger-Congo sub-families follow standard typological groupings.