PHONOLOGICAL INTERFERENCE IN MULTILINGUAL SPEECH MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Phoneme-level speech models recognize or generate speech as a sequence of phonemes, the smallest sound units that distinguish words. These models enable fine-grained pronunciation control and understanding, yet they often fail on input that does not match any single training language, such as speech alternating between two languages, known as code-switching, or speech in low-resource languages absent from training. We identify a systematic failure mode behind this, *phonological interference*: models assume the input is in a single language and impose its phonology, overriding local phoneme-level decisions that conflict with it. We measure interference by how often a model retains phonemes that one language has but the other lacks. On code-switched input, two phone recognizers (speech-to-phoneme models) and a phoneme-conditioned text-to-speech model lose 32%-79% of these phonemes, but lose far fewer of the phonemes both languages share. On unseen languages, recognizers impose the phonology of the training language they assign to the speech: the more confident the assignment, the more they lose phonemes the unseen language has but the assigned language lacks. We probe the models' language estimate from their internal activations, and trace interference to a low dimensional subspace. On monolingual speech, steering this subspace toward a different language makes the model lose the phonemes that only the original language uses and produce those that only the target language has. We introduce *windowed language estimation* (WLE), an inference time repair that replaces the model's language estimate in this subspace with one computed from a short window around each position. On code-switched input, WLE removes 34%-58% of the interference in all three models, and in the phone recognizers it leaves monolingual performance essentially unchanged.

1 INTRODUCTION

Phoneme-level speech models operate on phonemes, the smallest sound units that distinguish words, rather than words or characters. This makes them the natural choice wherever pronunciation matters: documenting low-resource languages, assessing non-native speech, and precise speech synthesis. Yet they often fail on precisely the input they are meant to handle: speech that matches no single training language. This happens when speakers switch languages mid-sentence (code-switching), common in bilingual communities (Bullock & Toribio, 2009), or when transcribing languages unseen during training, a central use of universal phone recognizers (Li et al., 2020).

We identify a systematic pattern underlying this failure, which we term *phonological interference*: the model contextually infers which language is being processed and imposes that language's phonology (sound system), overriding local decisions about individual phonemes. This term is inspired by similar interference effects in studies of human bilingualism (Weinreich, 1953). As illustrated in Figure 1, while a model trained on monolingual English and French may pronounce each language correctly in isolation, it conflates phonemes that are distinct across the two languages in mixed utterances, such as the French and English "r" sounds, represented by [ʁ] and [ɹ], respectively, in the International Phonetic Alphabet (IPA) (International Phonetic Association, 1999).

This failure mode is a manifestation of *language confusion*, a phenomenon documented across modalities: multilingual speech models struggle with code-switched input (Liu et al., 2026; Lee et al., 2025), while text LLMs have been observed to generate text in the wrong language (Marchi-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



(a) Phonological interference (b) Windowed language estimation (WLE)

Figure 1: **Phonological interference.** Multilingual phoneme-level speech models estimate the language from the whole input and override individual phoneme predictions. (a) In a French–English code-switched utterance, a phone recognizer replaces English [ɹ] in *right* with French [ʁ]. The bar shows the model's language estimate in a low dimensional language subspace at each position, with time running left to right. The model computes this estimate from the whole utterance, so it blends French (orange) into English (blue), and the model still reads the English span partly as French. (b) Windowed language estimation (WLE) replaces the language subspace coordinates at each position with those the model computes from a short window of audio around it. The estimate is then computed from the local language context, and [ɹ] is correctly transcribed.

sio et al., 2024; Park et al., 2026). We extend this line of work to phoneme-level speech models, where we find that language confusion manifests as systematic substitution of individual phonemes. We demonstrate this phenomenon in both phone recognition (speech-to-phonemes) and synthesis (phonemes-to-speech), and provide an account of it by tracing these substitutions to a specific internal representation that causally steers them.

We measure interference with an evaluation based on *unshared phoneme recall*: how often a model preserves phonemes that occur in one language but not the other. The relative drop in phonemes which are unshared between the two languages compared to the phonemes that are shared between them separates interference from a general drop in accuracy, which would lower both alike. We evaluate two phone recognizers and a phoneme-conditioned TTS model on utterances that switch between two training languages. We further evaluate the recognizers on monolingual speech in 43 languages absent from training. In these languages, the recognizers infer one of their training languages and impose its phonology, causing interference even in the absence of code-switching.

We trace interference to a low dimensional language subspace in the model's representations, analogous to language subspaces documented in text LLMs (Chang et al., 2022; Xie et al., 2022; Dumas et al., 2025). Steering this subspace toward another language on monolingual speech makes the model drop the phonemes that only the original language uses and replace them with phonemes of the target language, while keeping most of the phonemes the two languages share. We find an analogous subspace in Whisper, a standard ASR model, and show that editing it changes the language of the output transcript, linking phonological interference to language confusion in speech recognition.

Our analysis suggests an inference-time repair. If interference comes from a language estimate the model forms over the whole input, replacing it with a local estimate should let each language span keep its own phonology. We call this *windowed language estimation* (WLE): at each position, we replace the hidden state's coordinates in the language subspace with those the model computes from a short window of input around that position, leaving all other coordinates unchanged. Because this local estimate comes from the model's own representation of the same input, WLE needs no language label or external language identifier. It reduces phonological interference by 34%-58% across all three models and leaves the phone recognizers' monolingual error rates essentially unchanged.

Our contributions are: (1) we identify phonological interference and measure it in phone recognition and synthesis, on code-switched speech and unseen languages (Section 2); (2) we trace it to a low dimensional subspace that causally steers the output in phone recognizers, a TTS model and standard ASR (Section 3); (3) we introduce WLE, an inference-time repair that takes its target from the model's own estimate on a short window and reduces interference in all three models (Section 4).

2 Measuring Phonological Interference

Definition. Phonemes are the smallest units of sound that can change a word's meaning, such as /p/ and /b/ in English *pat* versus *bat*. Every language uses a particular set of phonemes, known as its

phoneme inventory (Ladefoged & Maddieson, 1996). Some phonemes are shared across languages (e.g., /t/ occurs in both English and French), while others occur in only one of the two (e.g., the *th* sound /θ/ of English *think* is absent from French).

We define *phonological interference* as the loss of phonemes that one language has but another language lacks, in favor of that other language's phonemes; for example, English [ɪ] may become French [ʁ] (more in Table 3, Appendix A). In code-switching, the other language is the second language of the utterance. For unseen languages, it is the training language the model assigns to the speech. Bilingual speakers show interference around a code switch as small, gradient shifts toward the other language (Antoniou et al., 2011; Piccinini & Arvaniti, 2015) (Appendix K), while the models we study replace whole phonemes (Figure 2). We hypothesize that models estimate the language from context and impose its phoneme inventory, and test this in Section 3.

Measurement. We define the *unshared phonemes* of a language pair (A, B) as the phonemes of A that B lacks, and vice versa (Appendix A). We measure how often the model preserves them (*unshared phoneme recall*) and compare this against *shared phoneme recall*, over the phonemes both languages have. If B interferes with A, unshared recall should drop more than shared recall, whereas a general accuracy loss would lower both alike. An unshared phoneme of A counts as lost only if it is deleted or replaced by a phoneme of B, since substitutions by any other phoneme cannot be attributed to interference. A shared phoneme counts as recalled only on an exact match.

In code-switching, we focus on utterances consisting of one span in each of two languages. We process each span alone (SEP) and both together (TOG), and report the **interference** $\Delta = r^{\text{SEP}}_{\min} - r^{\text{TOG}}_{\min}$: the drop in unshared phoneme recall of the worse span, caused by the other language's presence. For unseen languages, we train a linear probe (Section 2.1) to identify which training language the model assigns to the recording (e.g., Teop is read as Thai in Figure 2). Unshared phonemes are then the phonemes of the unseen language that this assigned training language lacks.

2.1 MODELS

Phone recognition. We study two multilingual phone recognition models: POWSM (Li et al., 2026) and PhoneticXeus (Bharadwaj et al., 2026a). These are both encoder-based models with bidirectional attention over audio token embeddings; POWSM uses an E-Branchformer audio encoder with an autoregressive phoneme decoder, while PhoneticXeus uses the Xeus self-supervised audio encoder and non-autoregressive (CTC) phoneme prediction. POWSM is trained to predict a language tag as its first output token, and supports a `<unk>` language tag during inference for unseen or unknown languages. PhoneticXeus receives no language information during training.

Both recognizers are trained on IPAPack++ (Zhu et al., 2025), roughly 17k hours of monolingual speech with phonemic transcriptions. The seven languages of our code-switched test sets are among its training languages, but none of the test recordings are. Each span alone is therefore in distribution, and the code-switched recordings are new only in combining two seen languages. The languages of our third test set, DoReCo, are absent from it (Section 2.2).

Text-to-Speech (TTS). We use CosyVoice3 (Du et al., 2025), an LLM-based TTS model built on a Qwen2 (Yang et al., 2024) language model, adapted to generate speech tokens autoregressively followed by a flow matching module. We fine-tune the LLM part to accept IPA input on 30k monolingual IPAPack++ (Zhu et al., 2025) utterances across 16 languages via LoRA, with no language tags or language embeddings, unlike multilingual TTS models that condition on the language explicitly (Lou et al., 2025). The model receives only IPA and a fixed reference speaker audio. Each input phoneme is a token, embedded with a learned linear projection applied to its PanPhon articulatory features (Mortensen et al., 2016).

Language probe. Several analyses need the language a model assigns to its input: which training language an unseen language is read as, and how confidently (Section 2.4), and how the other language changes the estimate for a span (Section 3). We train a 16-class logistic regression probe on the intermediate representations of monolingual IPAPack++ train split recordings, taken from each recognizer's audio encoder and from the TTS model's language model at its speech token positions. The probe gives each frame a score for each language, and we average these scores over the frames of a span or recording (Appendix D; training details in Appendix A).

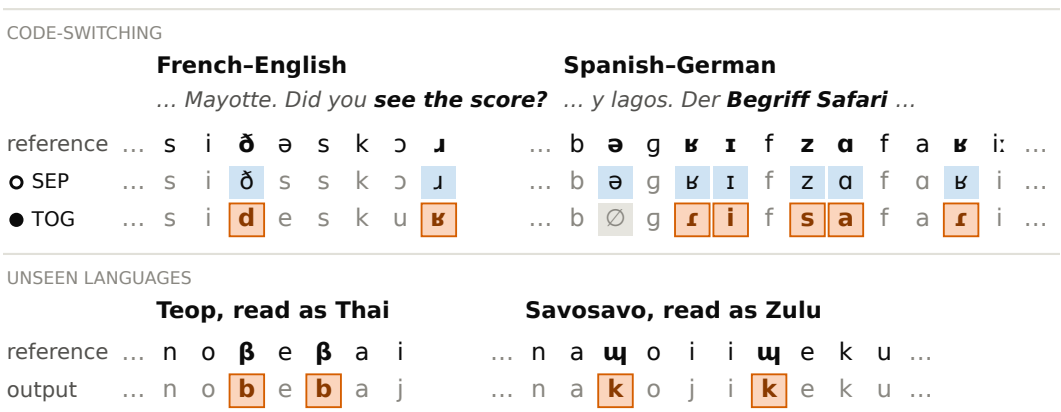


Figure 2: **Qualitative examples of interference.** PhoneticXeus transcriptions aligned to the reference, with unshared phonemes in bold. Output at unshared phonemes is shaded blue when the phoneme is kept, orange with a frame and bold type when it is replaced by the other language's phoneme, and grey for any other error (∅ marks a deletion). SEP transcribes the second span without the other language in its context, and TOG transcribes it with the other language in its context. In the French–English example, English *"the score"* takes French [d] and [ʁ]. The unseen language rows (Section 2.4) have no SEP/TOG conditions, so we list the model-assigned language instead.

2.2 Evaluation

We evaluate on two code-switched datasets (specifically containing utterances that switch once between languages), and one dataset of monolingual speech in out-of-distribution languages. For datasets not already containing IPA transcripts, we obtain these by processing orthographic text transcripts to IPA with eSpeak NG (eSpeak NG Developers, 2026).

SwitchLingua. We use 288 recordings of bilingual speakers reading code-switched sentences from SwitchLingua (Xie et al., 2026), 48 per pairing of English with German, Spanish, French, Italian, Hindi or Korean. This dataset captures natural bilingual speech, but does not control for span length or language balance (e.g., the second span is usually shorter and in English).

Spliced FLEURS. To complement SwitchLingua with a controlled set, we concatenate pairs of monolingual clips from the FLEURS test split (Conneau et al., 2023). We use the languages of SwitchLingua except French, which has no FLEURS recordings in the probe's training data, and create 10 samples for each ordered pair of six languages (300 concatenated samples total, about 8 s each). Every language appears equally in both positions and duration.

DoReCo. For unseen languages we use DoReCo (Paschen et al., 2020), a collection of documentary recordings of low-resource languages with time aligned phonemic transcriptions. We use the version from the PRiSM benchmark (Bharadwaj et al., 2026b) covering 43 unseen languages.

We apply the interference metric (Section 2) to both code-switched datasets. On SwitchLingua, SEP processes each span alone. On Spliced FLEURS, SEP replaces the other-language half with a same-language half, so SEP and TOG differ only in the other half's language, with length and speaker change held constant. Processing each half alone gives similar results (Appendix B).

To evaluate the code-switching datasets on phone recognition, we compare the model's IPA output directly to the reference. For TTS, we transcribe the synthesized audio back to IPA using PhoneticXeus. To prevent PhoneticXeus's own interference from contaminating the measurement, we split the audio at the language boundary (with forced alignment) and transcribe each language separately.

2.3 Results: Code-Switched Input

Figure 3 shows that interference is large and concentrated on unshared phonemes. Processing both languages together costs every model 32% to 79% of its unshared phoneme recall; under exact-

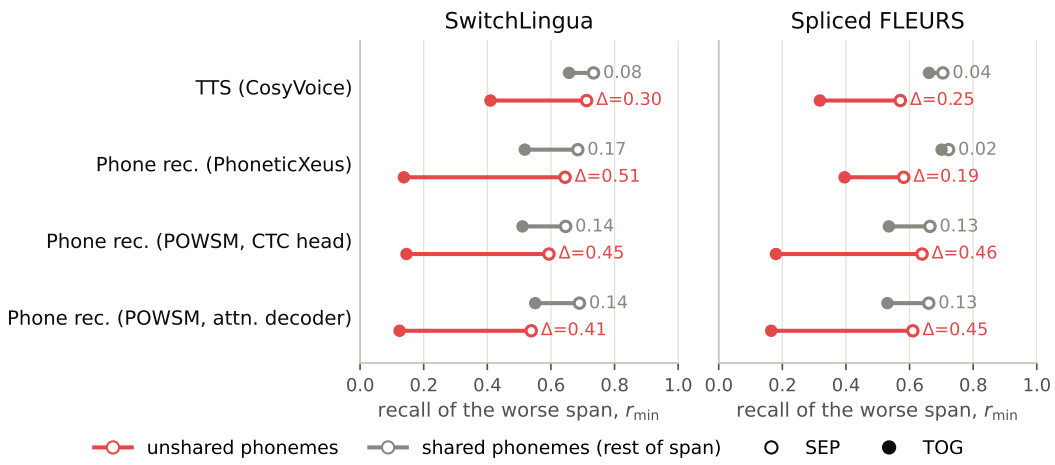


Figure 3: **Phonological interference on code-switched input.** Each bar runs from recall with each span processed separately (SEP, hollow) to recall with both languages together (TOG, filled); its length is the interference Δ. Red: unshared phonemes. Grey: shared phonemes. Unshared phoneme recall drops much more than shared phoneme recall in every model and dataset (Table 5).

match scoring, this drop is 1.9 to 8.2 times the drop in shared phonemes (Table 5). Most lost unshared phonemes are replaced by phonemes of the other language rather than by unrelated phonemes or deletions (Figure 2): on SwitchLingua, 65% of exact-match losses in TTS, 70% in PhoneticXeus and 78% in POWSM's CTC head. Some replacements could be merely notational, a similar sound written with a different symbol. Most are not: in articulatory features, they are 3 to 4 times farther from the original phoneme than the other language's closest phoneme is (Appendix B).

The same phoneme is lost more next to a language that does not have it. Whether a phoneme is unshared depends on the language pair; English /ɪ/, for example, is unshared next to Spanish and shared next to German. If the other language causes the loss, /ɪ/ should lose more recall next to Spanish than next to German. We test this for every phoneme that is unshared with some languages and shared with others, with a logistic regression that holds the phoneme and the language pair fixed and compares the drop next to languages that lack the phoneme with the drop next to languages that have it (Table 7). On Spliced FLEURS, when the other language lacks the phoneme, the odds that the model recalls it fall by an extra factor of 2.2 in POWSM and 1.3 in PhoneticXeus, compared with a language that has it. When we steer monolingual speech toward the other language (Section 3), the extra factor is 9.8 and 11.5. The TTS model shows the effect under the steering only. A phoneme's loss thus depends on whether the other language has it, not only on its difficulty.

2.4 Results: Unseen Languages

Unseen language recordings contain no second language, so the SEP/TOG comparison does not apply. Instead, we compare recordings by how strongly the model commits to one of its training languages. For each recording, we take the probe's score (Section 2.1) for the training language assigned to that language (Section 2). Within each language, we rank the recordings by this score and compare the third with the lowest scores to the third with the highest. If interference comes from the language estimate, the top third should lose more of the phonemes that the assigned language lacks, and shared phonemes should be unaffected. Figure 4 shows this pattern. From the lowest to the highest third, unshared phoneme recall falls by 0.053 (POWSM) and 0.060 (PhoneticXeus), while shared phoneme recall rises slightly (+0.015 and +0.029), which rules out a general drop in audio quality. The lost phonemes are mostly replaced by phonemes of the assigned language (Appendix C). These results match the prediction that the model's language estimate drives interference.

We evaluate only the phone recognizers here. The TTS output would have to be scored by transcribing it with PhoneticXeus, which itself shows interference on unseen languages even without code-switching. Splitting at the language boundary, which protected the code-switched evaluation, does not help on monolingual input.

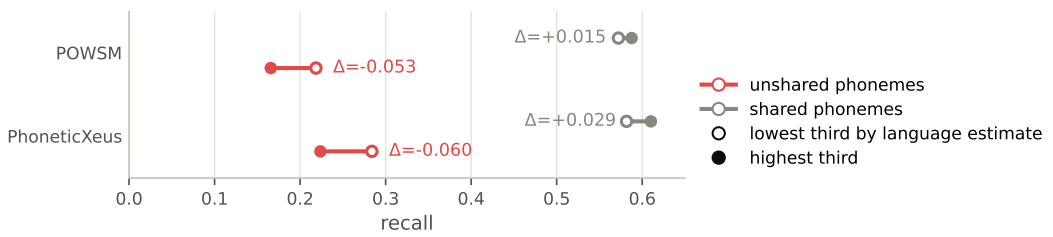
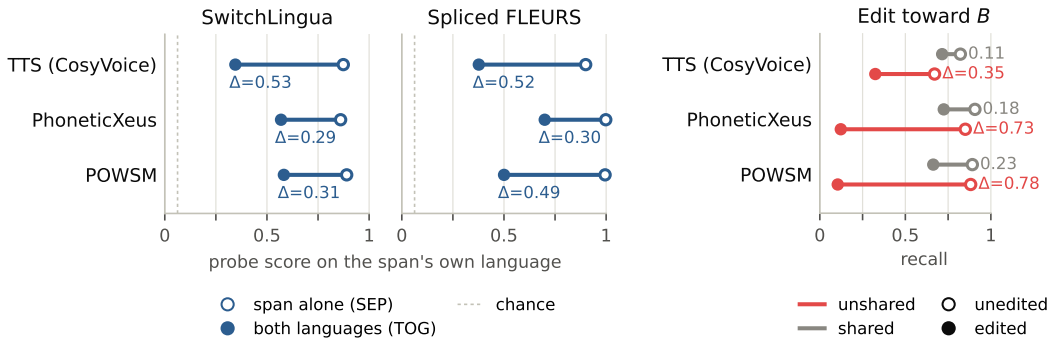


Figure 4: **Phonological interference in unseen languages.** Recall on 43 DoReCo languages, split by the model's language probe confidence: lowest third (hollow) vs. highest third (filled). Unshared phoneme recall (red) drops the more the model is confident in the estimated language, while shared phonemes (grey) remain stable, isolating interference from general difficulty of unseen languages.



(a) Language estimate on code-switched input

(b) Editing the language subspace

Figure 5: **The language estimate drops on code-switched input, and editing it reproduces interference.** (a) Probe score on each span's own language, processed separately (SEP, hollow) vs. together (TOG, filled), with the conditions of Section 2.2. (b) Recall on monolingual utterances before (hollow) and after (filled) the language subspace is set to the mean of another language B. Red: unshared phonemes. Grey: shared phonemes. Δ is the drop in each panel.

3 What Causes Phonological Interference?

On code-switched input, the probe score (Section 2.1) for a span's own language drops substantially when the other language is present (Figure 5a), even though the audio of each span is unchanged. Motivated by this, we define the *language subspace* directly from the language labels of the training data. At each block we take the centroid of each of the 16 probe languages, the mean hidden representation over that language's training frames. The 16 centered centroids span 15 directions. We whiten them by the within-language covariance Σ_w, so that the subspace separates languages relative to the variation within each language, and take the top-k singular directions $V \in \mathbb{R}^{D\times k}$ of the whitened centroids with $k = 15$, the full span, giving $\mathrm{proj}_{\mathrm{lang}}(\mathbf{h}) = \Sigma_w^{1/2} V V^\top \Sigma_w^{-1/2}\,\mathbf{h}$. This 15-dimensional subspace, under 2% of each model's hidden dimensions, is the low dimensional language subspace we refer to throughout (Appendix E).

To test whether this subspace causally drives interference, we manipulate it directly. Given a held-out monolingual utterance in language A, we edit it toward a target language B. Each block produces a representation $\mathbf{h}$ at every audio frame. At each frame and each block from a start block to the last, we replace the projection of $\mathbf{h}$ onto the language subspace with that of $\mathbf{c}_B$, the centroid representation of language B:

$$\mathbf{h} \leftarrow \mathbf{h} - \mathrm{proj}_{\mathrm{lang}}(\mathbf{h}) + \mathrm{proj}_{\mathrm{lang}}(\mathbf{c}_B), \tag{1}$$

leaving the rest of $\mathbf{h}$, $\mathbf{h} - \mathrm{proj}_{\mathrm{lang}}(\mathbf{h})$, unchanged. We apply the edit at blocks 6 to 8 of POWSM, 14 to 18 of PhoneticXeus and 14 to 22 of the TTS model, and each later block computes from the edited output. We test on monolingual IPAPack++ utterances in all 16 probe languages, from held-out speakers; the TTS model receives the corresponding IPA input.

Figure 5b shows that this edit reproduces interference. The phone recognizers lose most of their unshared phonemes (86–88% of their unedited recall) and the TTS model about half, while shared phonemes drop only a quarter to a third as much (Table 9, Appendix E). The effect lies almost entirely along the direction between the two languages: an edit along $A \to B$ alone produces 97% of the unshared phoneme recall drop of the full 15-dimensional edit.

Three controls isolate what the edit does. First, a random subspace of the same rank, shifted by the same amount, leaves recall almost unchanged: unshared phoneme Δ is 0.03–0.06 across the three models, against 0.35–0.78 for the language subspace. Second, erasing language information, by setting all frames to the mean of all 16 centroids, produces about half of the targeted edit's loss in the phone recognizers and 66% of it in the TTS model (Δ 0.23–0.42), without steering to any particular language. Third, which phonemes appear depends on the target: the edit toward B adds 7.3 (POWSM), 4.6 (PhoneticXeus) and 0.9 (TTS) phonemes per 100 that only B uses, while the same edit toward a third language C adds essentially none of B's, yet removes A's phonemes that C lacks just as much (Δ 0.35–0.78). Table 9 (Appendix E) gives the full numbers with intervals.

The mechanism extends beyond phoneme-level models. Automatic speech recognition (ASR) models such as Whisper (Radford et al., 2023) transcribe speech to text, and perform worse on code-switched speech due to language confusion (Liu et al., 2026). To test whether a similar language subspace can explain language confusion in standard ASR, we train the probe of Section 2.1 on Whisper's encoder. We find that the other language lowers its score for the span's own language by 0.28 on SwitchLingua, as in the phone recognizers, and by 0.11 on Spliced FLEURS. We then edit the encoder's language subspace toward B on monolingual speech in language A. This raises the probability of B's language token (which Whisper predicts before transcribing) by 0.20 and moves its transcript toward B, while editing toward a third language or through a random subspace does neither. The same language subspace thus controls Whisper's choice of language (Appendix F).

4 Windowed Language Estimation Mitigates Interference

To keep each language's phonemes in code-switched speech, the model must estimate the language locally, not from the whole utterance. We introduce *windowed language estimation* (WLE), which replaces the language subspace coordinates at each frame with those computed from a local window and leaves the remaining coordinates unchanged, targeting interference in code-switching.

We measure interference as in Section 2.2, applying WLE to the whole recording (TOG) and comparing it against each span decoded alone without any edit (SEP). We report SwitchLingua in Table 1 and Spliced FLEURS in Table 17 (Appendix J). We compare WLE with the same edit through a random subspace of equal rank and with a *sliding window* that replaces every coordinate with the windowed one (Appendix G). As a reference, an *oracle* applies Equation 1 toward the centroid of each frame's true language (each span's in TTS) and needs the language labels. For the recognizers, we measure the cost of WLE on monolingual speech as the change in two error rates on PRiSM (Bharadwaj et al., 2026b) (Table 2). The phone error rate (PER) counts every substitution, deletion and insertion as one error, while the phone feature error rate (PFER), which PRiSM reports, weights each error by its articulatory feature distance, so a near miss costs less than a distant substitution. For the TTS model, we measure the cost as the change in recall of shared phonemes on the code-switched sentences, relative to generating each span alone (Appendix G).

Phone recognizers. We encode the audio twice: once as a whole and once in 2 s sliding windows with a 1 s hop, each frame taken from the window in which it lies furthest from an edge. At each frame and at each block that Equation 1 edits, we replace the language subspace projection with that of the windowed encoding:

$$\mathbf{h} \leftarrow \mathbf{h} - \mathrm{proj}_{\mathrm{lang}}(\mathbf{h}) + \mathrm{proj}_{\mathrm{lang}}(\mathbf{h}_{\mathrm{win}}). \tag{2}$$

WLE reduces interference at window widths from 1 to 4 s; Appendix H gives the full ablation. The windowed language estimate is computed from the model's own representations, so WLE does not require any external language identification system.

TTS model. We apply the same edit (Equation 2) to the TTS model, with a language subspace built the same way from its own hidden representations, at rank 15 and blocks 14 to 22 of its 24

Model	Condition	$r^{\text{TOG}}_{\min}$ ↑	Δ (interference) ↓
TTS	No intervention	0.410	0.302
	WLE, fixed window	0.513	0.199
	WLE, span window	**0.619**	**0.093**
	Oracle	0.606	0.106
PhoneticXeus	No intervention	0.138	0.507
	WLE	**0.430**	**0.214**
	Oracle	0.536	0.108
POWSM	No intervention	0.146	0.448
	WLE	**0.306**	**0.288**
	Oracle	0.472	0.121

Table 1: **Effect of WLE on interference.** Best result in **bold**. WLE removes 69% of the interference in TTS with the span window and 34% with the fixed window, 58% in PhoneticXeus, and 36% in POWSM. The *Oracle* rows (gray) edit toward the true language's centroid and require language labels. POWSM rows use its CTC head. $r^{\text{TOG}}_{\min}$ and Δ as in Section 2.2.

	Model	Benchmark	ΔPER (pp) ↓		ΔPFER (pp) ↓	
			Sliding window	WLE	Sliding window	WLE
Seen	POWSM	TIMIT	+0.96	**+0.07**	+0.47	**+0.01**
		L2-ARCTIC	+1.55	**+0.10**	+0.31	**−0.00**
		GMU accent	+2.14	**−0.32**	+0.73	**−0.07**
	PhoneticXeus	TIMIT	+1.33	**+0.08**	+0.49	**+0.03**
		L2-ARCTIC	+3.22	**+0.14**	+0.50	**+0.03**
		GMU accent	+2.71	**−0.09**	+1.01	**−0.04**
Unsn.	POWSM	DoReCo	+1.97	**−0.09**	**−0.45**	+0.05
	PhoneticXeus	DoReCo	+2.08	**−0.22**	+1.06	**+0.15**

Table 2: **Performance in monolingual speech.** Change in phone error rate (ΔPER) and phone feature error rate (ΔPFER), in percentage points, on monolingual benchmarks relative to unmodified decoding. Negative = improvement. The lower (better) value in each pair is in **bold**. The sliding window degrades PER and PFER much more than WLE, except for POWSM's PFER on DoReCo.

(Appendix E). The difference is where $\mathbf{h}_{\text{win}}$ comes from. The recognizers window their input audio; the TTS model's input is a phoneme sequence, so we window that instead. We generate the sentence in chunks of about 12 phonemes. For each chunk, we take the roughly 24 input phonemes centered on it, generate from this short sequence alone, with no history and no edit, and take the mean of that run's hidden states at each edited block as $\mathbf{h}_{\text{win}}$. We then generate the chunk with the full phoneme input and all previously generated speech tokens as context, applying the edit at each of its positions (the first chunk is left unedited). We call this the *fixed window*: like the recognizers' window, it spans about 2 s of speech and needs no language label. We also test a *span window*, in which the chunks are the two language spans themselves, split at the switch point in the input text; this uses the boundary but not the identity of either language. Neither generating in chunks without the edit nor applying the edit through a random subspace reduces interference.

Results. WLE substantially reduces interference in all three models (Table 1), and in the recognizers it stays within ±0.35 pp of baseline monolingual PER and ±0.15 pp of PFER (Table 2). A naive sliding window over the full input, a less surgical approach to avoiding interference, also reduces interference but degrades monolingual PER by 1–3 pp and PFER by up to 1.1 pp. By targeting only the language subspace, WLE thus offers a favorable tradeoff, reducing interference at almost no cost to monolingual performance. The one exception is POWSM on DoReCo, where the sliding window lowers PFER by 0.45 pp while it raises PER by 2.0 pp. A random subspace of equal rank produces no meaningful change. In the TTS model the span window costs almost nothing on the shared phonemes; the fixed window costs more, since near the switch it contains phonemes of

both languages (Appendix G). Table 14 shows sentences in which WLE undoes the replacement of unshared phonemes in all three models, such as English [ɹ] becoming French [ʁ].

The oracle removes more interference than WLE in the recognizers, but its target is a training-corpus centroid rather than an encoding of the same audio, and in the TTS model the span window already matches it. Giving POWSM's attention decoder each span's own language tag, which also requires span labels, lowers Δ on SwitchLingua from 0.414 to 0.102.

5 Related Work

Language confusion in text LLMs. Multilingual language models encode language identity in a low dimensional subspace (Chang et al., 2022; Xie et al., 2022; Dumas et al., 2025), and LLMs generate text in the wrong language when this identity overrides the prompt (Marchisio et al., 2024; Choo et al., 2026). On code-switched input, their hidden states are anchored to one of the two languages (Park et al., 2026). Existing fixes steer hidden states toward a target language given as input (Yunfan et al., 2025; Sterz et al., 2025), or toward a source-language anchor extracted from the input (Park et al., 2026). In speech, a single frame or IPA symbol carries little language identity, so the model must estimate the language from context. We show that speech models have a similar subspace, that one language interferes with the other through it, and that it also drives language confusion in ASR. WLE rewrites only this subspace with the model's own estimate from a short window, with no language label, so each span keeps its own language.

Code-switching in speech models. Code-switched speech degrades both recognition and synthesis (Liu et al., 2026; Lee et al., 2025; Kline & Zellou, 2025). Liu et al. (2026) identify language confusion as a key challenge in code-switched ASR, but measure it only through word level metrics, as do broad coverage code-switched benchmarks (Yan et al., 2025). Stafford et al. (2026) document structured phoneme confusions without identifying their cause and Lee et al. (2025) propose data augmentation for recognition and translation. These works improve performance through data or model design. We measure phoneme-level interference in recognition and synthesis, on unseen languages and on code-switched input, trace its cause, and reduce it at inference time.

Interpretability of speech models. Self-supervised speech models encode phonetic and phonemic information (Pasad et al., 2021; Choi et al., 2024), including a low dimensional subspace in HuBERT for a phonological distinction (Martin et al., 2023), and show some human-like native-language perceptual biases (Millet & Dunbar, 2022; Rodriguez et al., 2024). These studies characterize what information is present without intervening on it. Like work on text LLMs that identifies and manipulates linear subspaces (Marks & Tegmark, 2024), we show that editing a language subspace causally changes the output. WLE has the form of an interchange intervention, which replaces a subspace with its value from another run. Distributed alignment search (Geiger et al., 2024) learns the subspace by optimization and copies its value from a different input to test a causal hypothesis. WLE instead uses the subspace spanned by the language centroids, which needs no optimization, and copies its value from a short window of the same input to repair the model at inference time.

6 Conclusion

We identified phonological interference in phoneme-level multilingual speech models: on code-switched input or languages absent from training, they impose the phonology of the language they estimate, replacing the phonemes it lacks with its own. We traced this to a low dimensional language subspace whose manipulation reproduces the effect and which also sets the output language of Whisper, linking interference to language confusion in ordinary speech recognition. Windowed language estimation substantially reduces interference on code-switched input without a language label, and leaves the phone recognizers' monolingual performance unchanged.

Limitations and future work. Our code-switched experiments use utterances that switch once between two languages, and extending to broader code-switching patterns remains future work. Our evidence on unseen languages compares recordings without intervening on the model. Disentangling the contributions of model architecture, training data, and labeling conventions to the emergence of interference is an important direction for future investigation.

AI USE STATEMENT

We used generative AI tools to implement methods, refine hypotheses and experiment designs, and interpret results, as well as to draft and edit the paper, prepare figures, and search the literature. We did not use them to develop theoretical models or conceptual frameworks. We reviewed all AI-assisted work and take full responsibility for the content of this paper, including any text, claims and artifacts produced with AI assistance.

ETHICS STATEMENT

Our work aims to understand and characterize how multilingual speech models fail on code-switched speech and on languages absent from their training data, and to make them more accurate for the people who speak these languages, including the endangered languages that linguists transcribe with the help of phone recognizers. The TTS model could be misused to imitate another speaker. We use only public corpora (SwitchLingua, FLEURS, DoReCo, IPAPack++) under their release terms.

REPRODUCIBILITY STATEMENT

All models are public checkpoints; we train only the TTS model's LoRA adapter and the linear probe. Appendix A gives all data, training, decoding and scoring settings, and Appendix E and Section 4 the subspace and WLE. We will release the code and models.

REFERENCES

Mark Antoniou, Catherine T. Best, Michael D. Tyler, and Christian Kroos. Inter-language interference in VOT production by L2-dominant bilinguals: Asymmetries in phonetic code-switching. *Journal of Phonetics*, 39(4):558–570, 2011. doi: 10.1016/j.wocn.2011.03.001.

Colleen Balukas and Christian Koops. Spanish-English bilingual voice onset time in spontaneous code-switching. *International Journal of Bilingualism*, 19(4):423–443, 2015. doi: 10.1177/1367006913516035.

Shikhar Bharadwaj, Chin-Jou Li, Kwanghee Choi, Eunjung Yeo, William Chen, Shinji Watanabe, and David R. Mortensen. An empirical recipe for universal phone recognition, 2026a. URL https://arxiv.org/abs/2603.29042.

Shikhar Bharadwaj, Chin-Jou Li, Yoonjae Kim, Kwanghee Choi, Eunjung Yeo, Ryan Soh-Eun Shim, Hanyu Zhou, Brendon Boldt, Karen Rosero, Kalvin Chang, et al. Prism: Benchmarking phone realization in speech models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 18093–18112, 2026b.

Barbara E Bullock and Almeida Jacqueline Toribio. *The Cambridge handbook of linguistic code-switching*. Cambridge university press, 2009.

Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 119–136, 2022.

Kwanghee Choi, Ankita Pasad, Tomohiko Nakamura, Satoru Fukayama, Karen Livescu, and Shinji Watanabe. Self-supervised speech representations are more phonetic than semantic. *arXiv preprint arXiv:2406.08619*, 2024.

Jinho Choo, JunSeung Lee, Jimyeong Kim, Yeeho Song, SK Hong, and Yeong-Dae Kwon. Tlpo: Token-level policy optimization for mitigating language confusion in large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42670–42690, 2026.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, 2020. doi: 10.18653/v1/2020.acl-main.747.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. Fleurs: Few-shot learning evaluation of universal representations of speech. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 798–805. IEEE, 2023.

Zhihao Du, Changfeng Gao, Yuxuan Wang, Fan Yu, Tianyu Zhao, Hao Wang, Xiang Lv, Hui Wang, Chongjia Ni, Xian Shi, et al. Cosyvoice 3: Towards in-the-wild speech generation via scaling-up and post-training. *arXiv preprint arXiv:2505.17589*, 2025.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31822–31841, 2025.

Jeffrey L. Elman, Randy L. Diehl, and Susan E. Buchwald. Perceptual switching in bilinguals. *The Journal of the Acoustical Society of America*, 62(4):971–974, 1977. doi: 10.1121/1.381591.

eSpeak NG Developers. espeak ng. `https://github.com/espeak-ng/espeak-ng`, 2026. Open-source text-to-speech synthesizer.

Melinda Fricke, Judith F. Kroll, and Paola E. Dussias. Phonetic variation in bilingual speech: A lens for studying the production–comprehension link. *Journal of Memory and Language*, 89:110–137, 2016. doi: 10.1016/j.jml.2015.10.001.

Maria Fernanda Gavino and Matthew Goldrick. Consequences of mixing and switching languages for retrieval and articulation. *Bilingualism: Language and Cognition*, 26(3):504–515, 2023. doi: 10.1017/S1366728922000682.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah Goodman. Finding alignments between interpretable causal variables and distributed neural representations. In *Causal Learning and Reasoning*, pp. 160–187. PMLR, 2024.

François Grosjean. The bilingual's language modes. In Janet L. Nicol (ed.), *One Mind, Two Languages: Bilingual Language Processing*, pp. 1–22. Blackwell, Oxford, 2001.

François Grosjean and Joanne L. Miller. Going in and out of languages: An example of bilingual flexibility. *Psychological Science*, 5(4):201–206, 1994. doi: 10.1111/j.1467-9280.1994.tb00501.x.

International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

Tyler Méndez Kline and Georgia Zellou. The perception of code-switched vs. monolingual sentences in tts voices. *Frontiers Comput. Sci.*, 7, 2025. URL `https://api.semanticscholar.org/CorpusID:280643658`.

P. Ladefoged and I. Maddieson. *The Sounds of the World's Languages*. Phonological Theory. Wiley, 1996. ISBN 9780631198154. URL `https://books.google.com/books?id=h1byJz_rWUcC`.

Sangmin Lee, Woojin Chung, Seyun Um, and Hong-Goo Kang. Unicom: A universal code-switching speech generator. In *EMNLP (Findings)*, pp. 13273–13288, 2025.

Chin-Jou Li, Kalvin Chang, Shikhar Bharadwaj, Eunjung Yeo, Kwanghee Choi, Jian Zhu, David R Mortensen, and Shinji Watanabe. Powsm: A phonetic open whisper-style speech foundation model. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17882–17899, 2026.

Xinjian Li, Siddharth Dalmia, Juncheng Li, Matthew Lee, Patrick Littell, Jiali Yao, Antonios Anastasopoulos, David R Mortensen, Graham Neubig, Alan W Black, et al. Universal phone recognition with a multilingual allophone system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8249–8253. IEEE, 2020.

Hexin Liu, Haoyang Zhang, Qiquan Zhang, Xiangyu Zhang, Dongyuan Shi, Eng Siong Chng, and Haizhou Li. Code-switching speech recognition under the lens: Model-and data-centric perspectives. *IEEE Transactions on Audio, Speech and Language Processing*, 2026.

Haowei Lou, Hye-young Paik, Sheng Li, Wen Hu, and Lina Yao. Generalized multilingual text-to-speech generation with language-aware style adaptation. *arXiv preprint arXiv:2504.08274*, 2025.

Kelly Marchisio, Wei-Yin Ko, Alexandre Bérard, Théo Dehaze, and Sebastian Ruder. Understanding and mitigating language confusion in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6653–6677, 2024.

Samuel Marks and Max Tegmark. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=aajyHYjjsk.

Kinan Martin, Jon Gauthier, Canaan Breiss, and Roger Levy. Probing self-supervised speech models for phonetic and phonemic information: a case study in aspiration. *arXiv preprint arXiv:2306.06232*, 2023.

Juliette Millet and Ewan Dunbar. Do self-supervised speech models develop human-like perception biases? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7591–7605, 2022.

David R Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. Panphon: A resource for mapping ipa segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3475–3484, 2016.

Daniel J Olson. Bilingual language switching and selection at the phonetic level: Asymmetrical transfer in vot production. *Journal of Phonetics*, 41(6):407–420, 2013.

Jeonghyun Park, Seunghyun Yoon, Yonghyun Jun, and Hwanhee Lee. Code-switching reveals language anchoring in multilingual llms. *arXiv preprint arXiv:2606.19668*, 2026.

Ankita Pasad, Ju-Chieh Chou, and Karen Livescu. Layer-wise analysis of a self-supervised speech representation model. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 914–921. IEEE, 2021.

Ludger Paschen, François Delafontaine, Christoph Draxler, Susanne Fuchs, Matthew Stave, and Frank Seifart. Building a time-aligned cross-linguistic reference corpus from language documentation data (doreco). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pp. 2657–2666, 2020.

Page Piccinini and Amalia Arvaniti. Voice onset time in Spanish–English spontaneous code-switching. *Journal of Phonetics*, 52:121–137, 2015. doi: 10.1016/j.wocn.2015.07.004.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 28492–28518. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/radford23a.html.

Joselyn Rodriguez, Kamala Sreepada, Ruolan Leslie Famularo, Sharon Goldwater, and Naomi Feldman. Self-supervised speech representations display some human-like cross-linguistic perceptual abilities. In *Proceedings of the 28th Conference on Computational Natural Language Learning*, pp. 458–463, 2024.

Miquel Simonet. Phonetic consequences of dynamic cross-linguistic interference in proficient bilinguals. *Journal of Phonetics*, 43:26–37, 2014. doi: 10.1016/j.wocn.2014.01.004.

Eli Stafford, Aimée Lahaussois, and Guillaume Wisniewski. What do neural speech models know about phonology? evidence from structured phoneme confusions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 25860–25873, 2026.

Peter M. Stahl. Lingua-language-detector, 2026. URL https://pypi.org/project/lingua-language-detector/.

Hannah Sterz, Fabian David Schmidt, Goran Glavas, and Ivan Vulic. Recover the target language: Language steering without sacrificing task performance. In *EMNLP (Findings)*, pp. 19390–19405, 2025.

Zirui Wang, Zachary C. Lipton, and Yulia Tsvetkov. On negative interference in multilingual models: Findings and a meta-learning treatment. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4438–4450, 2020. doi: 10.18653/v1/2020.emnlp-main.359.

Uriel Weinreich. *Languages in Contact: Findings and Problems*. Number 1 in Publications of the Linguistic Circle of New York. Linguistic Circle of New York, New York, 1953.

Peng Xie, Xingyuan Liu, Yequan Bie, Tsz Wai Chan, Yangqiu Song, Yang Wang, Hao Chen, and Kani Chen. Switchlingua: The first large-scale multilingual and multi-ethnic code-switching dataset. *Advances in Neural Information Processing Systems*, 38, 2026.

Zhihui Xie, Handong Zhao, Tong Yu, and Shuai Li. Discovering low-rank subspaces for language-agnostic multilingual representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5617–5633, 2022.

Brian Yan, Injy Hamed, Shuichiro Shimizu, Vasista Lodagala, William Chen, Olga Iakovenko, Bashar Talafha, Amir Hussein, Alexander Polok, Kalvin Chang, et al. Cs-fleurs: A massively multilingual and code-switched speech dataset. *arXiv preprint arXiv:2509.14161*, 2025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Xie Yunfan, Lixin Zou, Dan Luo, Min Tang, Chenliang Li, Xiangyang Luo, and Liming Dong. Mitigating language confusion through inference-time intervention. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 8418–8431, 2025.

Jian Zhu, Farhan Samir, Eleanor Chodroff, and David R Mortensen. Zipa: A family of efficient models for multilingual phone recognition. *arXiv preprint arXiv:2505.23170*, 2025.

A Additional Experimental Details

Phone Recognizers. We use the public checkpoints unchanged, on 16 kHz mono audio. POWSM receives the `<unk>` language tag in every condition. We report its attention decoder (ESPnet defaults: beam 5, CTC weight 0) and greedy CTC decoding from the same encoder. PhoneticXeus is decoded with greedy CTC. One encoder frame covers 40 ms of audio in POWSM (25 frames per second) and 20 ms in PhoneticXeus (50 frames per second). One speech token of the TTS model covers 40 ms (25 tokens per second).

TTS training. The base model is `Fun-CosyVoice3-0.5B-2512`, frozen. We train LoRA adapters (rank 32, $\alpha = 64$, dropout 0.05) on every attention and MLP projection of the LM, together with a linear projection of PanPhon features. The training set is 30,388 IPAPack++ utterances of 1 to 24 s with at least five phonemes, about 2,000 per language (1,150 to 2,469), in English, French, Georgian, German, Greek, Hindi, Hungarian, Italian, Kinyarwanda, Korean, Polish, Spanish, Thai, Turkish, Vietnamese and Zulu. The targets are the tokens of CosyVoice3's speech tokenizer, and the loss is cross-entropy on the speech positions. We use a batch size of 8 for a single epoch with the official hyperparameters.

TTS inference. The LM input is the start token, the projected phonemes and the task token. We sample with CosyVoice3's default repetition aware sampling (top-p 0.8, top-k 25) and cap the output at 20 speech tokens (0.8 s) per input phoneme to avoid endless decodes, and leave the flow matching module and the vocoder unchanged. The reference speaker is the 3.5 s prompt clip distributed with CosyVoice3. Each input is synthesized under four sampling seeds (0 to 3), and the recall of a span is averaged over seeds before the worse span is taken. For the TOG condition the synthesized audio (24 kHz) is force-aligned to its words and cut at the language boundary as described below.

Language estimate. The probe is a multinomial logistic regression on frames standardized with the training frames' mean and variance. It is fitted on 16,000 IPAPack++ utterances, 1,000 per language, drawn from Common Voice, an audiobook corpus (MLS, or LibriSpeech for English) and FLEURS where the language has them, and tested on 5,800 further utterances. The 2,389 training speakers and 1,112 test speakers are disjoint. Each utterance contributes at most 10 frames and each phoneme at most 2, located with the recognizer's CTC alignment, which gives up to 160,000 training frames. For the TTS a frame is one speech token position of the LM with the utterance's own speech tokens as input (teacher forcing). All audio is scaled to -23 dBFS RMS. We read the probe at layer 8 of POWSM, layer 17 of PhoneticXeus and block 7 of the TTS LM, the layers where the held-out probe score saturates (Appendix D).

Reference transcriptions. For SwitchLingua and FLEURS we transcribe the orthographic reference word by word with eSpeak NG 1.52 through `phonemizer` 3.4, each word in the voice of its own language (`en-us`, `de`, `es`, `fr-fr`, `hi`, `it`, `ko`). We remove stress marks and tie bars and apply a fixed symbol map per language that brings eSpeak's conventions to those of IPAPack++, for example German /r/ is written ʁ, and length marks are dropped in English, Italian and Korean.

SwitchLingua sample. For each language pair we shuffle the corpus recordings with a fixed seed and keep the first 48 whose sentence has 6 to 22 words and exactly one change of language between adjacent words, with at most three recordings from any one of the corpus's conversations. The 288 recordings come from 239 conversations.

Cutting at the language boundary. We align each recording to its words with `Qwen3-ForcedAligner-0.6B` and cut at the midpoint between the end of the last word of the first span and the start of the first word of the second. No audio is discarded or padded, and the same rule is applied to human recordings and synthesized audio.

Spliced FLEURS. Halves are taken from FLEURS test utterances with at least 4 s of speech, with leading and trailing silence removed (samples below 2% of the file's peak). The first half of a clip is the end of its utterance and the second half is the beginning of its utterance, so the join falls at natural utterance edges. The other side is cut at a word boundary that falls in a pause, located with the MMS forced aligner (`mms-300m-1130-forced-aligner`; Qwen3 for Korean), choosing

Language pair	Replacement	Example	Language pair	Replacement	Example
English–French	ɹ → ʁ	***r**eally*	German–Spanish	ɪ → i	*m**i**t*
English–German	w → v	***w**as*	English–German	ð → d	***th**e*
English–Spanish	ɑ → o	*pr**o**bably*	English–Spanish	z → s	***s**ince*
German–Hindi	ʁ → ɾ	*de**r***	English–Italian	æ → a	***a**nd*

Table 3: **Examples of unshared phonemes** from language pairs in our experiments. Each phoneme at the left of an arrow occurs only in the first language of the pair; the arrow shows its most common replacement when both languages are processed together, with an example word whose bold letters spell the phoneme. Table 4 lists the full sets for every pair and dataset.

the boundary that gives a half closest to 4 s within 3 to 5 s. The reference of a half is the phonemes of the words it contains. Each half receives a 5 ms fade at both ends, and the two halves are joined with no gap or crossfade at 16 kHz. We draw 10 clips for each of the 30 ordered language pairs with a fixed seed; the two halves of a clip are always different sentences. For every clip we also build the two same language splices used as SEP, in which the other half is replaced by a half in the same language from a different speaker and sentence, giving 600 control clips.

DoReCo. PRiSM's DoReCo set has 18,299 clips in 44 languages. We exclude its French, which is one of the probe's 16 languages, and report on the remaining 43. The language assigned to a DoReCo language is the one with the highest probe score after averaging over frames, then over all of that language's clips. Clips are split into thirds by the rank of their probe score for the assigned language within their own language. A phoneme is unshared when the DoReCo references of the language contain it at least three times and the probe's training references of the assigned language contain it fewer than three times.

Scoring. The unshared phonemes of a language pair are those that occur at least three times in that language's reference spans and never in the other language's, counted over the pair's recordings in the dataset. Reference and output are aligned by edit distance with unit costs. A phoneme is recalled when the aligned output phoneme is identical after normalization (Appendix B); a deletion is a loss. For unshared phonemes, a substitution by a phoneme absent from the other language's reference spans is also accepted, so the losses are deletions and substitutions by the other language's phonemes. The share of losses replaced by the other language's phonemes (Section 2) is the rise in such substitutions from SEP to TOG over the fall in exact matches. Recordings in which one span has no unshared phoneme are not scored, which leaves 266 of 288 SwitchLingua recordings (265 for PhoneticXeus, which lacks one decode) and 282 of 300 spliced clips (274 for POWSM's attention decoder, which returns 8 empty decodes). Intervals are 95% percentile bootstrap intervals from 10,000 resamples of recordings.

Table 3 shows examples and their most common replacements, and Table 4 lists the resulting sets.

B Detailed Interference Results

Table 5 gives the full numbers behind Figure 3. Before comparing reference and output phonemes, we normalize diacritics, including length, nasalization, non-syllabic marks, and any diacritic appearing fewer than three times in the reference transcriptions.

Choice of the SEP baseline on Spliced FLEURS. On Spliced FLEURS each half is processed alone, next to a same language half, and next to the other language half. Table 6 gives the worst span recall of unshared phonemes in each condition. A same language half changes recall by at most 0.012, and every interval includes 0, so Δ is nearly the same from either baseline.

The same phoneme next to different languages. Section 2.3 reports that a phoneme loses more recall next to a language that lacks it. An alternative explanation is that unshared phonemes are simply harder to recognize or synthesize, and would lose more recall next to any language. To separate the two, we use phonemes that are unshared in some language pairs and shared in others.

Pair (A–B)	Only in A	Only in B
SwitchLingua		
English–French	ɹ$_{54}$ ð$_{33}$ ɚ$_{30}$ æ$_{29}$ ʌ$_{21}$ h$_{18}$	ʁ$_{141}$ y$_{45}$ œ$_{11}$ ø$_{4}$
English–German	ɹ$_{53}$ w$_{28}$ æ$_{28}$ ð$_{28}$ ɚ$_{18}$ θ$_{8}$ ʒ$_{6}$	ʁ$_{124}$ ç$_{44}$ ɐ$_{35}$ ø$_{13}$ x$_{11}$ y$_{10}$ œ$_{5}$ ʏ$_{5}$
English–Hindi	f$_{35}$ ð$_{33}$ æ$_{31}$ ɚ$_{27}$ ʌ$_{24}$ w$_{21}$ ɑ$_{17}$ v$_{15}$ θ$_{7}$	ɾ$_{61}$ ʊ$_{19}$ k$^{h}_{18}$ b$^{h}_{12}$ t$^{h}_{11}$ ʃ$^{h}_{9}$ p$^{h}_{8}$ ʈ$_{7}$ ɖ$_{6}$ ʒ$^{h}_{3}$
English–Italian	ɪ$_{105}$ ɹ$_{54}$ ə$_{38}$ ʊ$_{30}$ ʌ$_{27}$ ð$_{18}$ ŋ$_{15}$ æ$_{11}$ ɑ$_{11}$ ɚ$_{11}$ θ$_{10}$ h$_{6}$	ɾ$_{173}$ ʎ$_{5}$
English–Korean	ə$_{63}$ ɹ$_{62}$ ʊ$_{38}$ ɚ$_{35}$ f$_{30}$ ð$_{27}$ æ$_{26}$ v$_{17}$ ɔ$_{14}$ ɑ$_{12}$ ʒ$_{5}$ θ$_{3}$	ɯ$_{119}$ ɾ$_{77}$ ɬ$_{57}$ ɕ$_{53}$ ʑ$_{47}$
English–Spanish	ɪ$_{113}$ ɹ$_{46}$ ʊ$_{33}$ z$_{23}$ e$_{22}$ ð$_{20}$ ɑ$_{20}$ ɔ$_{19}$ ʌ$_{15}$ v$_{14}$ æ$_{12}$ ɚ$_{11}$ θ$_{8}$ ʒ$_{7}$ h$_{6}$	ɾ$_{138}$ r$_{16}$ x$_{16}$ ɲ$_{5}$
Spliced FLEURS		
English–German	ɹ$_{70}$ ð$_{50}$ ɚ$_{42}$ æ$_{41}$ ʌ$_{30}$ w$_{15}$ ʒ$_{13}$ θ$_{6}$	ʁ$_{107}$ ɐ$_{38}$ ç$_{33}$ x$_{14}$ y$_{13}$ ø$_{5}$ œ$_{4}$
English–Hindi	ɹ$_{68}$ ð$_{50}$ æ$_{36}$ ʌ$_{36}$ ɚ$_{35}$ f$_{29}$ v$_{28}$ w$_{22}$ ŋ$_{19}$ ɑ$_{19}$ θ$_{10}$	ɾ$_{96}$ ʊ$_{38}$ b$^{h}_{15}$ ʈ$_{15}$ t$^{h}_{8}$ ɖ$_{7}$ d$^{h}_{6}$ ɳ$_{6}$ k$^{h}_{5}$ p$^{h}_{5}$ ʃ$^{h}_{5}$
English–Italian	ɪ$_{187}$ ə$_{106}$ ɹ$_{88}$ æ$_{47}$ ʌ$_{47}$ ð$_{44}$ ʊ$_{37}$ ɚ$_{36}$ ɑ$_{33}$ h$_{17}$	ɾ$_{111}$
English–Korean	ɹ$_{106}$ ə$_{89}$ z$_{51}$ ɚ$_{42}$ æ$_{41}$ ɔ$_{38}$ v$_{29}$ ɑ$_{28}$ ʒ$_{8}$ θ$_{4}$	ɯ$_{135}$ ɾ$_{71}$ ɬ$_{40}$ ɕ$_{30}$ ʑ$_{23}$
English–Spanish	ɪ$_{176}$ ə$_{121}$ ɹ$_{90}$ z$_{66}$ æ$_{55}$ ð$_{45}$ ʌ$_{40}$ ʊ$_{38}$ ɚ$_{36}$ e$_{26}$ ɔ$_{26}$ v$_{25}$ ɑ$_{19}$ h$_{18}$ ʒ$_{8}$ θ$_{8}$	ɾ$_{125}$ r$_{22}$ x$_{18}$ ʝ$_{5}$ ɲ$_{3}$ ʎ$_{3}$
German–Hindi	ʁ$_{125}$ v$_{44}$ ɑ$_{31}$ ç$_{25}$ ɐ$_{21}$ x$_{12}$ y$_{10}$ ø$_{10}$ ʏ$_{9}$ œ$_{4}$	ɾ$_{90}$ ʒ$_{28}$ ʊ$_{20}$ d$^{h}_{13}$ b$^{h}_{12}$ t$^{h}_{12}$ k$^{h}_{11}$ ɖ$_{10}$ ʈ$_{8}$ ɳ$_{6}$ ʂ$_{5}$ ʃ$^{h}_{3}$
German–Italian	ə$_{176}$ ɪ$_{128}$ ʁ$_{122}$ ʊ$_{68}$ ɑ$_{37}$ ɐ$_{35}$ ç$_{31}$ x$_{16}$ h$_{15}$ y$_{11}$ œ$_{9}$ ø$_{8}$ ʏ$_{8}$	ɾ$_{115}$ w$_{10}$ ʎ$_{5}$
German–Korean	ə$_{181}$ ʁ$_{129}$ z$_{51}$ ʊ$_{48}$ ɔ$_{45}$ f$_{44}$ ɐ$_{39}$ v$_{37}$ ç$_{35}$ ɑ$_{31}$ x$_{10}$ y$_{8}$ ø$_{7}$ ʏ$_{5}$ œ$_{4}$	ɯ$_{117}$ ʌ$_{90}$ ɾ$_{63}$ ɬ$_{43}$ w$_{32}$ ɕ$_{29}$ ʑ$_{28}$
German–Spanish	ə$_{148}$ ʁ$_{121}$ ɪ$_{113}$ ʊ$_{54}$ v$_{40}$ z$_{40}$ ɑ$_{37}$ ɐ$_{30}$ ç$_{29}$ e$_{27}$ ɔ$_{24}$ y$_{17}$ h$_{13}$ ø$_{6}$ ʏ$_{5}$	ɾ$_{112}$ r$_{16}$ w$_{14}$ ʝ$_{8}$ ɲ$_{5}$ ʎ$_{3}$
Hindi–Italian	ə$_{170}$ ɾ$_{119}$ ɪ$_{67}$ ʊ$_{25}$ ʊ$_{23}$ ʈ$_{17}$ ʃ$^{h}_{9}$ d$^{h}_{6}$ k$^{h}_{6}$ ɖ$_{6}$ b$^{h}_{5}$ p$^{h}_{4}$ t$^{h}_{4}$ ɳ$_{4}$ ʂ$_{3}$	ɾ$_{106}$ ʋ$_{19}$ w$_{10}$ ʎ$_{8}$ ŋ$_{4}$
Hindi–Korean	ə$_{187}$ ʒ$_{32}$ ʊ$_{26}$ ʊ$_{25}$ ɖ$_{19}$ ɔ$_{17}$ b$^{h}_{12}$ ʈ$_{12}$ f$_{10}$ ɳ$_{10}$ d$^{h}_{9}$ t$^{h}_{8}$ k$^{h}_{6}$ z$_{4}$ g$^{h}_{3}$ ʃ$^{h}_{3}$	ɯ$_{131}$ ʌ$_{126}$ ɬ$_{52}$ ʑ$_{31}$ ɕ$_{27}$
Hindi–Spanish	ə$_{178}$ e$_{95}$ ɪ$_{65}$ h$_{62}$ ʒ$_{23}$ ʊ$_{22}$ ʊ$_{20}$ ɔ$_{17}$ ʈ$_{15}$ b$^{h}_{9}$ d$^{h}_{8}$ ɖ$_{8}$ k$^{h}_{6}$ t$^{h}_{6}$ ʃ$^{h}_{4}$ z$_{3}$ ɳ$_{3}$ ʂ$_{3}$	x$_{16}$ f$_{13}$ ɾ$_{13}$ w$_{12}$ ʎ$_{4}$ ʝ$_{3}$
Italian–Korean	ɾ$_{100}$ ʋ$_{23}$ ɔ$_{20}$ f$_{15}$ ɲ$_{7}$ ʎ$_{5}$ ʒ$_{5}$	ɯ$_{119}$ ʌ$_{96}$ h$_{86}$ ɾ$_{71}$ ɬ$_{45}$ ɕ$_{38}$ ʑ$_{29}$
Italian–Spanish	e$_{134}$ ɔ$_{19}$ ʋ$_{13}$ z$_{11}$ ʒ$_{8}$	ɾ$_{106}$ x$_{23}$ ʝ$_{5}$ ɲ$_{4}$
Korean–Spanish	ɯ$_{104}$ ʌ$_{99}$ h$_{95}$ ɬ$_{35}$ ʑ$_{32}$ ɕ$_{31}$ e$_{27}$ ɪ$_{3}$	x$_{13}$ ɾ$_{10}$ ʎ$_{5}$ ʝ$_{4}$

Table 4: **The full unshared phoneme sets** of every language pair in both code-switched datasets. Since selected phonemes depend on an occurrence threshold in observed data, the exact phonemes selected may differ between the datasets for the same pair of languages (e.g., comparing English–German in both datasets). Phonemes are ordered from most to least frequent, and the subscript is the number of occurrences in those reference spans.

English /ɪ/, for example, is unshared next to Spanish, which lacks it, and shared next to German, which has it. It is the same English sound in both pairs, so if it loses more recall next to Spanish than next to German, the difference comes from the other language and not from the phoneme itself.

Only phonemes that are unshared in at least one pair and shared in at least one other contribute to this comparison. On Spliced FLEURS there are 56, about half of them vowels and taps (brief *r*-like sounds), such as English /ɪ/ and /ə/ and Hindi /ɾ/. A phoneme that no other language in the set has, such as English /ɹ/, is unshared in every pair and therefore cannot enter the comparison.

Data. For every phoneme in the reference transcript, we record whether the output reproduces it exactly, once before and once after the second language is introduced. "Before" is a half spliced next to another half in its own language, and "after" is the same half spliced next to the other language.

Model. Consider a phoneme p in a half of language A, spliced next to a half of language B. Let $u = 1$ if B lacks the phoneme p and $u = 0$ if B has it. Each reference phoneme is scored twice, and t marks the condition: $t = 0$ before the second language is introduced and $t = 1$ after. We fit one logistic regression that predicts whether p is recalled:

$$\operatorname{logit}\Pr(\text{recalled}) = \underbrace{a_{p,A} + b_{A,B} + d\,u}_{\text{before}} - t\underbrace{\big(\alpha_{p,A} + \beta_{A,B} + \gamma\,u\big)}_{\text{drop}},$$

(a) SwitchLingua (265/266 scored)

	Unshared phonemes			Shared phonemes		
Model	Δ	Δ/r^{SEP}	Δ_{exact}	Δ	Δ/r^{SEP}	Ratio
TTS (CosyVoice)	.302 [.265, .340]	.424	.224 [.186, .262]	.077 [.065, .090]	.105	2.9×
Phone rec. (PhoneticXeus)	.507 [.462, .551]	.786	.345 [.304, .387]	.167 [.145, .188]	.244	2.1×
Phone rec. (POWSM, CTC)	.448 [.404, .490]	.754	.365 [.327, .402]	.136 [.110, .161]	.210	2.7×
Phone rec. (POWSM, attn.)	.414 [.371, .456]	.769	.268 [.233, .304]	.139 [.116, .164]	.202	1.9×

(b) Spliced FLEURS (274/282 scored)

	Unshared phonemes			Shared phonemes		
Model	Δ	Δ/r^{SEP}	Δ_{exact}	Δ	Δ/r^{SEP}	Ratio
TTS (CosyVoice)	.253 [.222, .284]	.443	.198 [.169, .228]	.044 [.033, .055]	.062	4.5×
Phone rec. (PhoneticXeus)	.186 [.156, .218]	.320	.181 [.150, .213]	.022 [.009, .036]	.030	8.2×
Phone rec. (POWSM, CTC)	.459 [.425, .494]	.718	.414 [.384, .445]	.129 [.112, .148]	.195	3.2×
Phone rec. (POWSM, attn.)	.445 [.408, .482]	.730	.421 [.388, .453]	.130 [.105, .156]	.197	3.2×

Table 5: **Phonological interference on code-switched input** (full numbers behind Figure 3), with 95% bootstrap intervals. Δ: drop in worst span recall from SEP to TOG. $\Delta/r^{\text{SEP}}_{\min}$: drop as a fraction of the SEP level. Δ_{exact}: the same drop under exact match, the rule used for shared phonemes; unshared Δ also accepts substitutions by phonemes the other language lacks. Ratio: unshared Δ_{exact} divided by shared Δ.

	$r_{\min}$			Δ from	
Model	Alone	Same lang.	Other lang.	Alone	Same lang.
TTS (CosyVoice)	.570	.571	.318	.252 [.217, .287]	.253 [.222, .284]
Phone rec. (PhoneticXeus)	.591	.582	.396	.196 [.165, .228]	.186 [.156, .218]
Phone rec. (POWSM, CTC)	.636	.639	.180	.456 [.421, .491]	.459 [.425, .494]
Phone rec. (POWSM, attn.)	.598	.610	.165	.433 [.394, .472]	.445 [.408, .482]

Table 6: **Two SEP baselines on Spliced FLEURS.** Worst span recall of unshared phonemes with each half alone, next to a same language half (SEP in the main text) and next to the other language half (TOG), and the drop to TOG from each baseline, with 95% bootstrap intervals.

where $\text{logit}(x) = \log \frac{x}{1-x}$ is the log-odds. The first group of terms gives the log-odds of recall before the second language is introduced, and the second group gives how far they fall afterwards. We model log-odds rather than recall because recall is bounded: the same change in recall means more near 0 or 1 than near 0.5, and log-odds put drops from different starting points on a comparable scale.

Quantity of interest. γ is how much further the log-odds of recall fall when B lacks p than when B has it. If the drop depended only on the phoneme, and not on whether the other language has it, γ would be 0. We also report e^{γ}, which is easier to read: the odds of recall, $\Pr/(1-\Pr)$, fall by an extra factor of e^{γ} next to a language that lacks the phoneme.

Controls. The remaining terms absorb other reasons recall might differ. $a_{p,A}$ and $\alpha_{p,A}$ give each phoneme its own recall before and its own drop, separately per language (English /t/ and German /t/ get separate terms). $b_{A,B}$ and $\beta_{A,B}$ do the same for each ordered language pair, absorbing pairs that are harder overall. d lets phonemes the other language lacks start from a different recall, so that a lower starting point is not mistaken for a larger drop. Because $\alpha_{p,A}$ already captures how far p drops in general, γ is estimated only from phonemes with $u = 0$ in some pairs and $u = 1$ in others, which is why only those phonemes contribute.

The edit experiment. We fit the same regression to the edit of Section 3, which takes monolingual speech in language A and moves the model's language estimate toward another language B. Here "before" is the unedited output and "after" is the output edited toward B, and $u = 1$ if B lacks p by the training-reference definition of Appendix E. Because each of the 16 languages is edited toward

Setting	Model	γ (log-odds)	Odds ratio e^{γ}
Spliced FLEURS	POWSM, CTC head	0.77 [0.61, 0.97]	2.2 [1.8, 2.6]
	POWSM, attn. decoder	0.84 [0.66, 1.04]	2.3 [1.9, 2.8]
	PhoneticXeus	0.27 [0.10, 0.47]	1.3 [1.1, 1.6]
	TTS (CosyVoice)	0.10 [−0.02, 0.22]	1.1 [0.98, 1.3]
Edit toward B	POWSM, CTC head	2.28 [2.07, 2.53]	9.8 [7.9, 12.6]
	PhoneticXeus	2.44 [2.31, 2.64]	11.5 [10.1, 14.1]
	TTS (CosyVoice)	0.19 [0.13, 0.26]	1.2 [1.1, 1.3]

Table 7: **The same phoneme next to languages that lack it and languages that have it.** γ: additional drop in the log-odds of recalling a phoneme when the other language lacks it, with the phoneme and the language pair held fixed; 0 if a phoneme's difficulty alone decided the drop. 95% bootstrap intervals. The TTS interval on Spliced FLEURS includes 0.

the others, many more phonemes are unshared with some targets and shared with others: 262 to 300 contribute to the comparison.

Table 7 reports γ and e^{γ} for both settings, with 95% bootstrap intervals, refitting the regression on every resample. We resample the units whose outputs are correlated: clips on Spliced FLEURS; for the edit, speakers in the phone recognizers, and utterances in the TTS model, which always speaks in the same reference voice.

How far a replaced phoneme moves. Section 2.3 reports that unshared phonemes are replaced by distant phonemes of the other language. To measure this, we take each unshared reference position, find the phoneme aligned to it in the transcript (using the same recall edit distance alignment), and keep substitutions by a phoneme of the other language. We keep only the substitutions that the other language adds: the other language splice minus the same language splice on Spliced FLEURS, and TOG minus SEP on SwitchLingua. We measure distance with PanPhon's weighted feature edit distance (Mortensen et al., 2016), where a voicing change costs 0.25 and /ɹ/ to /ʁ/ costs 6.9. We compare it with the distance each substitution would have if it had gone to the most similar phoneme of the other language (Table 8; 95% bootstrap intervals over clips on Spliced FLEURS and conversations on SwitchLingua).

The added substitutions are 3.0 to 4.3 times as far as the most similar phoneme, and only 19% to 42% of them are that most similar phoneme. For example, English /ɹ/ next to French becomes /ʁ/ (6.9) rather than /j/ (1.6), and English /w/ next to German becomes /v/ (6.25) rather than /j/ (1.5). If the substitutions only reflected small articulatory conversions, where two languages write a similar sound with different symbols, they would go to the most similar phoneme.

There are two caveats to this experiment. First, deletions have no distance and are excluded; they are 8% to 35% of the added misses. PanPhon has no feature separating the tap from the trill (distance 0). Secondly, substitutions between them are 4% to 6% of the phone recognizers' added substitutions on Spliced FLEURS, none of the TTS model's, and impossible on SwitchLingua, where English has neither.

C Unseen languages: detailed results

All numbers are unweighted means over the 43 languages, with 95% bootstrap intervals from clips resampled within language.

Recall by confidence in unseen languages. From the lowest to the highest third of language estimate confidence, unshared phoneme recall falls by 0.053 on POWSM and 0.060 on PhoneticXeus (both intervals exclude zero). Shared phoneme recall rises slightly ($+0.015$ and $+0.029$), which a general drop in audio quality would not produce. The effect holds when restricted to consonants only (-0.054 and -0.067), addressing the concern that unshared vowels may reflect transcription conventions more than acoustic contrasts. The assigned language is Thai for 28–29 of the 43 languages, which likely reflects the probe's default for unfamiliar speech more than phonological similarity. The probe chooses among only 16 of the recognizers' training languages. A probe over more of

Dataset	Model	Added substitutions	Most similar phoneme	Share
Spliced FLEURS	POWSM, CTC head	2.64 [2.42, 2.87]	0.68	0.29
	POWSM, attn. decoder	2.86 [2.61, 3.10]	0.68	0.25
	PhoneticXeus	1.78 [1.46, 2.11]	0.59	0.42
	TTS (CosyVoice)	2.47 [2.18, 2.75]	0.79	0.35
SwitchLingua	POWSM, attn. decoder	3.36 [3.03, 3.72]	0.77	0.19
	PhoneticXeus	2.76 [2.45, 3.10]	0.70	0.29
	TTS (CosyVoice)	2.76 [2.36, 3.15]	0.84	0.28

Table 8: **Replacements are far from the phonemes they replace.** Added substitutions: mean PanPhon feature distance between an unshared phoneme and the other language's phoneme that replaces it, over the replacements the other language adds (95% bootstrap intervals). Most similar phoneme: the same mean if each had been replaced by the closest phoneme of the other language. Share: fraction of replacements that are that closest phoneme.

them could assign each unseen language more precisely, which we expect would increase the interference we measure.

Replacements. The share of unshared phoneme positions written as a phoneme from the assigned language's inventory rises from 0.57 (lowest tercile) to 0.63 (highest) on POWSM ($+0.054$) and from 0.58 to 0.61 on PhoneticXeus ($+0.033$).

WLE on unseen languages. WLE does not reduce interference on DoReCo. This is expected: the basis contains none of these languages, and Appendix I shows that a basis missing both languages of a pair repairs much less than the full basis. Additionally, on monolingual input the windowed and full clip language estimates largely agree, so the edit has little to correct. The sliding window also fails here and additionally degrades shared phoneme recall by 0.02–0.03.

D Language probe by layer

Figure 6 shows the language probe's accuracy at each layer, which we use to choose the layer for all language estimates in the paper. We evaluate on 5,800 held-out IPAPack++ utterances from unseen speakers (Appendix A).[1] Predictions are averaged over frames, and accuracy is averaged over the 16 languages, so chance is $1/16$. In both recognizers, accuracy plateaus by mid-network and barely changes within one or two layers of our choice. It stays high even when the probe has never seen the target language's corpus, so it is not picking up recording artifacts. The TTS behaves differently. Accuracy peaks sharply at block 7 (0.956, up from 0.694 at block 6) and is more sensitive to an unseen corpus, dropping by 0.129 compared with 0.011 for POWSM and 0.066 for PhoneticXeus. We therefore take language estimates from layer 8 of POWSM, layer 17 of PhoneticXeus, and block 7 of the TTS language model.

E Language subspace additional details

Construction. At each encoder block, we compute the mean hidden representation of each of the $K = 16$ languages over the probe's training frames. These 16 centroids, after centering, span at most $K - 1 = 15$ directions. We whiten by the average within-language covariance Σ_w (with a ridge $\lambda = 10^{-3}\,\mathrm{tr}(\Sigma_w)/D$, where D is the hidden dimension) before taking the top-k singular directions, which gives the projection $\mathrm{proj}_{\mathrm{lang}}$ used in Equation 1. The whitening ensures the edit moves along directions that separate languages after accounting for within-language variance, rather than just the directions of largest raw spread. We also tested an unwhitened variant (orthogonal projection onto the centered means directly); it removes slightly more unshared phonemes but degrades shared phonemes more, so we use the whitened version.

[1] FLEURS has no speaker labels, so we use its speaker-disjoint test split.

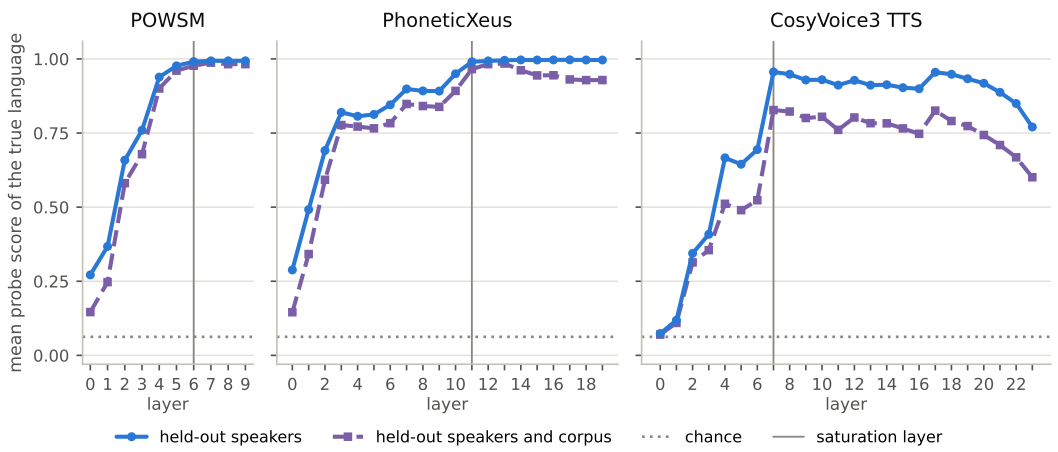


Figure 6: **Language probe accuracy by layer**, on held-out monolingual speech. **Held-out speakers**: 5,800 utterances from 1,112 unseen speakers. **Held-out speakers and corpus**: each point is from a probe trained without that language's corpus, so neither the speakers nor the recording pipeline have been seen. The vertical line marks the first layer within 0.01 of the best. For the TTS, the probe reads the language model at each speech token position, with the utterance's own speech tokens as input. We read language estimates at layer 8 (POWSM), layer 17 (PhoneticXeus) and block 7 of the TTS language model.

Evaluation data. We evaluate on held-out monolingual speech from IPAPack++ (FLEURS, Common Voice, and audiobook corpora) across all 16 languages, with no speaker overlap with the probe's training set. Each utterance in language A is paired with a target language B and a third language C (238 of the 240 ordered pairs occur).

Scoring. A phoneme of A is unshared with B if A's training references contain it at least three times and B's never. This differs from the eSpeak-based inventories used in Section 2. Comparison uses exact match after removing length, nasalization and non-syllabic marks. Δ is the per-utterance drop in recall from the unedited output, averaged within each language and then across the 16 languages. Intervals are 95% bootstrap over speaker groups within language. About 21% of the unshared positions differ from the target language only in notation: affricates written with a tie bar where the target language writes the two halves separately (English references write /t$\widehat{\text{ʃ}}$/ as two phonemes, /t/ and /ʃ/), or phonemes with a diacritic that the target language writes plainly (/k$^{\text{h}}$/ against /k/). We therefore rescore every edit after reducing each phoneme in the references and the transcripts to its base letters, splitting tie bar affricates and removing diacritics, and recompute the unshared sets from the reduced symbols. This changes Δ by at most 0.02 for every edit we report, on both recognizers and the TTS model, and by at most 0.04 for any edit on either half of the speakers. The reduced reading also removes contrasts that diacritics carry, such as aspiration in Hindi, Korean and Thai, so we keep the unreduced reading as the main one.

Dev/test split. Held-out speakers are split in two. The start block and the whitening were chosen on the development half (2,906 utterances), and the rank as described under hyperparameter selection below; all reported numbers are from the test half (2,890 utterances). Across the 22 edits shared by both halves, every Δ agrees within 0.022.

Hyperparameter selection. We select the start block where we apply the steering to balance effect size against output quality. The start block is the earliest block whose cumulative edit (through to the final block) reaches 90% of the maximum effect across all start blocks. With blocks numbered from 0, this rule selects block 6 of POWSM's 9 encoder blocks and block 14 of PhoneticXeus's 19, so the edit is applied at blocks 6 to 8 of POWSM and blocks 14 to 18 of PhoneticXeus. WLE (Section 4) edits the same blocks. Edits with PER above 0.5 relative to the unedited output are excluded as destructive. The rank is $k = 15$, the full span of the 16 centroids.

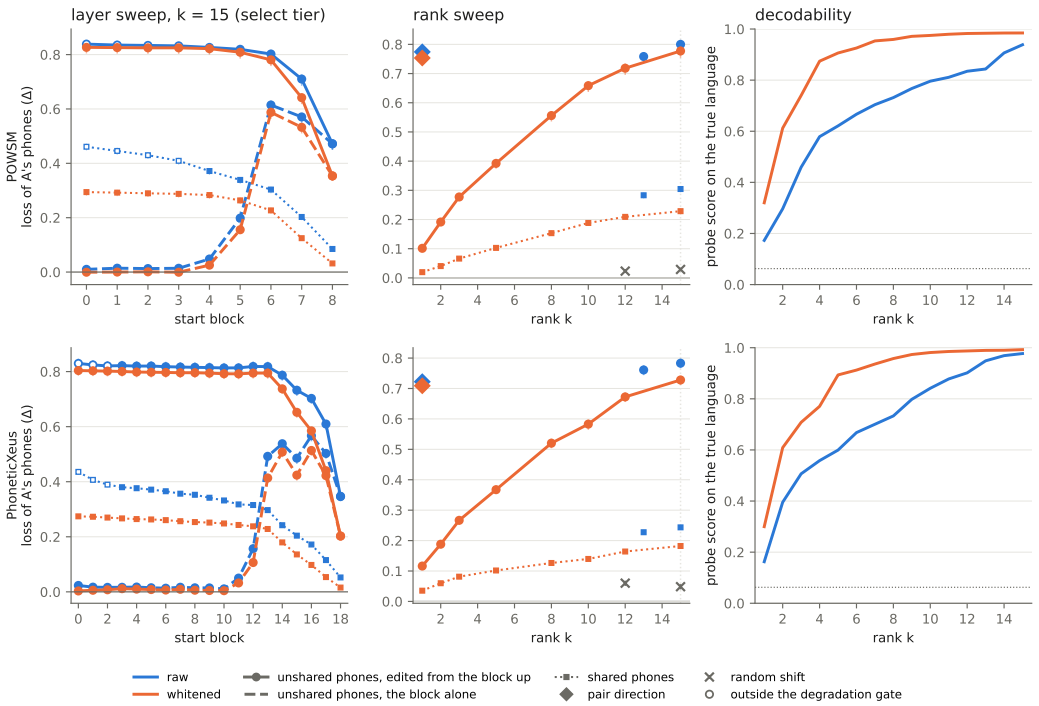


Figure 7: Subspace edit sweeps. *Left:* effect of varying the start block (rank 15, development speakers). *Middle:* effect of varying rank (at the selected start block, test speakers). *Right:* probe accuracy for the true language inside the rank-k subspace (chance dotted). Solid: cumulative edit from start block to last; dashed: single block only. Blue: unwhitened; orange: whitened. See legend for additional markers.

Rank and depth. The edit's effect grows with rank and depth (Figure 7). Rank 8 reaches 72% of the rank-15 effect. A probe refitted inside the subspace reads the correct language at $k = 7$ (probe score of ≥ 0.94), meaning fewer directions suffice for classification than for reproducing interference with steering.

In terms of depth, early blocks contribute little: editing before block 5 (POWSM) or block 11 (PhoneticXeus) changes recall by at most 0.03. No single block carries more than 75% of the full stack effect, indicating that the language estimate is recomputed across blocks. Starting the edit at block 0 adds only 0.05–0.07 to the unshared phoneme loss but increases shared phoneme damage by a similar amount.

Controls. Table 9 gives full numbers. Key findings: (1) A random subspace of equal rank shifted by the same amount removes at most 0.06 of unshared phonemes, confirming the effect is specific to the language directions. (2) Erasing language information (setting all frames to the mean of all 16 centroids) removes roughly half the unshared phonemes without imposing any particular language. (3) The phonemes that appear depend on the target language. We count the output phonemes that B uses and neither A nor C does, per 100 output phonemes. The edit toward B raises this count by 7.3 (POWSM), 4.6 (PhoneticXeus) and 0.9 (TTS), from 0.02 and 0.04 in the recognizers' unedited output. The same edit toward a third language C changes it by at most 0.06, with every interval including zero, while it removes as many of A's phonemes unshared with C as the edit toward B removes of those unshared with B (Table 9). For the TTS model the edit toward C was generated with one seed, on 1,515 utterances. (4) A single direction from A's centroid to B's captures 97% of the full edit's effect, but requires knowing the source language and so cannot be used as a practical repair. (5) The whitened construction is more precise in the phone recognizers: relative to their unedited recall, shared phonemes fall 0.29 (POWSM) and 0.23 (PhoneticXeus) as far as unshared phonemes scored by exact match, against 0.37 and 0.29 without whitening.

(a) POWSM (CTC head) — 2,890 utterances, 593 speaker groups. Unedited recall: .882 (unshared), .892 (shared).

	Δ_{unshared}	Δ_{shared}	$\Delta_{\text{sh}}/\Delta_{\text{unsh}}$
Language subspace $\to B$	.777 [.757, .791]	.229 [.220, .236]	.29
Language subspace $\to C$ (against C)	.777 [.756, .792]	—	—
One direction ($A \to B$)	.753 [.730, .767]	.228 [.220, .235]	.30
Erased to global mean	.417 [.399, .429]	.109 [.103, .114]	.26
Random shift (seed 1)	.029 [.025, .032]	.006 [.004, .009]	—
Random shift (seed 2)	.030 [.025, .035]	.006 [.005, .008]	—

(b) PhoneticXeus — 2,890 utterances, 593 speaker groups. Unedited recall: .851 (unshared), .907 (shared).

	Δ_{unshared}	Δ_{shared}	$\Delta_{\text{sh}}/\Delta_{\text{unsh}}$
Language subspace $\to B$	.728 [.714, .740]	.182 [.177, .187]	.25
Language subspace $\to C$ (against C)	.725 [.713, .737]	—	—
One direction ($A \to B$)	.709 [.695, .721]	.179 [.174, .185]	.25
Erased to global mean	.397 [.381, .408]	.089 [.085, .092]	.22
Random shift (seed 1)	.048 [.042, .054]	.015 [.013, .017]	—
Random shift (seed 2)	.062 [.054, .068]	.017 [.014, .019]	—

(c) CosyVoice3 TTS — 2,071 utterances, 4 decodes each. Unedited recall: .671 (unshared), .822 (shared).

	Δ_{unshared}	Δ_{shared}	$\Delta_{\text{sh}}/\Delta_{\text{unsh}}$
Language subspace $\to B$	.346 [.336, .357]	.107 [.101, .112]	.31
Language subspace $\to C$ (against C)	.347 [.329, .365]	—	—
One direction ($A \to B$)	.337 [.326, .348]	.105 [.100, .110]	.31
Erased to global mean	.227 [.218, .237]	.050 [.046, .053]	.22
Random shift (seed 1)	.034 [.028, .041]	.010 [.007, .013]	—
Random shift (seed 2)	.025 [.012, .037]	.006 [.001, .011]	—

Table 9: **Manipulating the language subspace reproduces interference** (full numbers behind Figure 5b). Δ: drop in recall from unedited to edited output, paired per utterance. $\Delta_{\text{sh}}/\Delta_{\text{unsh}}$: selectivity ratio; lower means the edit targets unshared phonemes more precisely. CIs are 95% bootstrap over speaker groups within language. The $\to C$ row gives the drop in recall of A's phonemes unshared with C; the other two columns are defined against B and are left empty for it, and for the TTS model it uses 1,515 utterances with one decode each. Edited transcripts retain 96–98% of unedited length in all conditions.

TTS model. The TTS model receives the same held-out utterances as IPA input. We apply the edit of Equation 1 during generation at the positions that produce speech tokens, from block 14 to block 22 of its 24, using the whitened subspace of rank 15 built from the TTS model's own training representations. The start block was selected by the same rule as for the phone recognizers. The last block, 23, is left unedited. Its representations pass through a final normalization before we read them, so an edit there would act on different values than those the subspace was fitted on. Output speech is transcribed by PhoneticXeus and compared to the input IPA. Each utterance is generated with four random seeds; edited outputs are compared against unedited outputs at the same seed.

Excluded target languages. Editing toward four target languages (Korean, Zulu, Kinyarwanda, Vietnamese) caused the TTS output to degenerate: the edited audio was 1.4–1.9 times as long as the original (vs. at most 1.25 for other targets), and PER rose to 0.51 (vs. 0.34) and PFER to 0.23 (vs. 0.12), from 0.20 and 0.07 unedited. We report these four separately. Figure 5b and Table 9 use the remaining 2,071 utterances. On the excluded four, the edit still removes far more unshared than shared phonemes (84% vs. 32%), but with more overall degradation. We leave open an investigation into why these four behave differently, which is not clearly explained by training data volume or pretraining language coverage.

	SwitchLingua	Spliced FLEURS
WER, SEP	.160	.152
WER, TOG	.259	.540
Δ WER	+.099 [.071, .126]	+.388 [.356, .423]
Δ WER, span in the picked language	−.010 [−.027, .006]	+.045 [.007, .097]
Δ WER, the other span	+.223 [.168, .282]	+.741 [.695, .784]

Table 10: **Whisper loses the span that is not in the language it picks.** WER rises when the other language is present. The span in the picked language is transcribed about as well as it is alone, and almost all of the rise comes from the other span, which Whisper translates or drops. WER is the number of word errors divided by the number of reference words, over both spans. Δ is TOG minus SEP on the same recordings, with 95% bootstrap intervals over recordings. As in Section 2.2, SEP is the span alone on SwitchLingua, and the span next to another speaker of the same language on Spliced FLEURS. The bottom rows use the 274 recordings and 286 clips where Whisper picks one of the two languages.

F LANGUAGE CONFUSION IN WHISPER

In this section we show that the same language subspace also explains a failure at the level of words: language confusion in ASR, extending beyond phoneme-level models. Whisper large-v3 (Radford et al., 2023) is a widely used ASR model that outputs words. For every 30 s of audio, its decoder first outputs one language token and then writes the transcript. We call the language of this token the *picked language*. We use the public checkpoint unchanged and let the model pick the language itself (temperature 0, beam size 5).

Whisper outputs words, so it cannot show interference as defined in Section 2. On code-switched speech it shows language confusion (Marchisio et al., 2024): part of the transcript is in the wrong language or missing. We show here that this failure has the same cause as interference, in three steps. First, on code-switched speech Whisper loses the words of the span that is not in the language it picks. Second, its encoder carries a language estimate that shifts with context, as in the phone recognizers. Third, editing the language subspace of the encoder changes the language Whisper picks, and the picked language decides the language of the transcript. Whisper is a useful test for two reasons. It is widely used, so the failure affects ordinary speech recognition. It also states its language choice itself, so the test does not depend on our phoneme scoring.

Word error rate. We transcribe SwitchLingua and Spliced FLEURS under the SEP and TOG conditions of Section 2.2. We measure word error rate (WER, lower is better) against the orthographic reference, after removing case and punctuation. In TOG, we align the whole transcript to the whole reference and count the errors of each span separately. WER rises when the other language is present (Table 10): from 0.16 to 0.26 on SwitchLingua and from 0.15 to 0.54 on Spliced FLEURS. On Spliced FLEURS the rise appears in both positions (first half $+0.42$, second half $+0.36$), in the 200 clips without English ($+0.40$) and in all 15 language pairs.

Which span loses words? In each TOG recording, one span is in the picked language and the other span is not. We leave out the recordings where Whisper picks a third language (14 in each dataset, mostly Urdu for Hindi). The span in the picked language is transcribed almost as well as it is alone. Almost all of the rise in WER comes from the other span (Table 10, bottom rows). On SwitchLingua, Whisper usually translates the other span into the picked language. On Spliced FLEURS it usually drops it: in 218 of 286 clips it writes nothing for that span.

Language estimate. We train the probe of Section 2.1 on Whisper's encoder, with the same utterances, frames and held-out speakers as for POWSM. On held-out speakers, the probe score of the true language is 0.99 at layer 22. We select this layer as in Appendix D.

On SwitchLingua, the probe score of a span's own language is 0.89 for the span alone and 0.61 inside the whole recording. This drop of 0.28 [.26, .30] is close to the drop in POWSM (0.31) and PhoneticXeus (0.29). On Spliced FLEURS the drop is 0.11 [.06, .21], smaller than in POWSM (0.49) and PhoneticXeus (0.30).

Edit	$\Delta P(B)$	Δ transcript in B	$\Delta P(A)$
Toward B	$+.201$ [.184, .212]	$+.221$ [.195, .238]	$-.447$ [$-.465, -.427$]
Toward C	$+.007$ [.005, .008]	$+.002$ [.000, .003]	$-.449$ [$-.468, -.429$]
Mean of all centroids	$+.008$ [.006, .010]	$+.003$ [.001, .005]	$-.276$ [$-.298, -.261$]
Random subspace	$+.001$ [.000, .001]	$+.000$ [.000, .000]	$-.013$ [$-.019, -.010$]

Table 11: **Whisper transcribes in the language its encoder is steered toward.** Every edit of the language subspace lowers the probability of A, and only the edit toward B raises the probability of B and moves the transcript to B. A is the language of the utterance, B is the target language and C is a third language. $P(A)$ and $P(B)$ are the probabilities Whisper gives to the language tokens of A and B. "Transcript in B" is the text language identifier's confidence that the transcript is written in B. Each entry is the change from the unedited model, where $P(A)$ is 0.963. Entries are averaged within each source language and then over the 14 languages, with 95% bootstrap intervals over speaker groups. The edit toward C uses the 2,027 utterances whose C has a language token.

Editing the language subspace. We build the language subspace on Whisper's encoder as in Appendix E. We apply the edit of Equation 1 to the frames that contain audio, from block 21 to the last block (31). The start block is selected on the development speakers as in Appendix E. Whisper has no language token for Kinyarwanda or Zulu, which leaves 2,345 test utterances in 14 languages. We measure two things. The first is the probability that Whisper gives to the language token of B. The second is whether the transcript is written in B, according to a text language identifier (Stahl, 2026). Table 11 shows how both change from the unedited model.

Editing the subspace changes the language Whisper picks, and the transcript follows the pick. The edit toward B lowers the probability of A by 0.45 and raises the probability of B by 0.20. The identifier's confidence that the transcript is written in B rises by a similar amount (0.22). After the edit, Whisper picks B in 28% of the utterances, and in those utterances it writes the transcript in B. Some of these transcripts are translations. Others spell the sounds of A in the script of B.

The edit moves Whisper to the language we aim at. The edit toward a third language C lowers the probability of A just as much, but it raises C ($+0.19$) and leaves B unchanged. When we move the frames to the mean of all 16 language centroids, which points at no language, the probability of A falls by less (0.28) and B does not gain. A random subspace of the same rank, shifted by the same amount, has almost no effect.

Finally, we give the decoder the language token of A after the edit toward B. The transcript then stays in A, and its character error rate rises by only 0.02. The edit therefore changes the language Whisper writes in and leaves the content it hears almost intact. Together these results give a chain: the language subspace of the encoder sets the language token, and the language token sets the language of the transcript.

Where the confusion arises. These results locate the confusion. On Spliced FLEURS, the encoder still gives most frames of both halves to their own language when the other language is present (probe score 0.88). The decoder picks one language for the whole clip, so only one half can be in the picked language. The other half is translated or dropped. When we give the decoder the correct language token, the transcript is nearly correct even after the edit. The encoder's language estimate therefore shifts with context, as in the phone recognizers, and the decoder's single choice turns this shift into the loss of a whole span.

G WLE: CONFIDENCE INTERVALS AND RANDOM SUBSPACE CONTROL

Any edit of the hidden states at these blocks could reduce interference simply by disrupting the model, so we test whether the repair depends on the language subspace itself. Table 12 gives the intervals behind Table 1. The control applies the same edit, at the same blocks and with the same targets, through a random subspace of equal rank.

The random subspace does not reduce interference in any model. In the recognizers it changes Δ by at most 0.02 on SwitchLingua, and on Spliced FLEURS it stays within 0.02 of no repair (Table 17). Paired per recording, WLE leaves 0.16 to 0.28 less interference than the random subspace in every

Model	Condition	$r_{\min}^{\text{TOG}} \uparrow$	Δ (interference) $\downarrow$
TTS	No intervention	.410	.302 [.265, .340]
	In chunks, no edit	.351	.361 [.328, .394]
	WLE, fixed window	.513	.199 [.175, .223]
	WLE, fixed window, random subspace	.332	.380 [.350, .411]
	Span by span, no edit	.383	.329 [.291, .368]
	WLE, span window	.619	.093 [.068, .120]
	WLE, span window, random subspace	.383	.330 [.292, .367]
	Oracle (known span language)	.606	.106 [.079, .135]
	Oracle, random subspace	.388	.324 [.286, .363]
PhoneticXeus	No intervention	.138	.507 [.462, .551]
	Sliding window (2 s)	.443	.202 [.160, .243]
	WLE	.430	.214 [.171, .257]
	WLE, random subspace	.150	.494 [.449, .539]
POWSM	No intervention	.146	.448 [.404, .490]
	Sliding window (2 s)	.441	.153 [.120, .187]
	WLE	.306	.288 [.249, .327]
	WLE, random subspace	.143	.451 [.407, .494]

Table 12: Interference on SwitchLingua with 95% bootstrap intervals over recordings. In all three models the random subspace leaves interference at the level of the same decoding without the edit.

model and window, and every interval excludes zero. On monolingual speech, the two edits cost the same: the random subspace also changes PER and PFER by at most 0.15 pp (Table 13). Both edits disturb the model by the same amount, and only the edit through the language subspace removes interference.

In the TTS model, generating in 12-phoneme chunks without the edit slightly increases interference, by 0.059 [0.025, 0.093], and the random subspace does not undo this. The edit removes a further 0.162 [0.131, 0.194] beyond the chunked decode.

The span window outperforms the fixed window on both interference and cost. It leaves 0.106 [0.074, 0.137] less interference and barely lowers shared phoneme recall: relative to generating each span alone, shared recall drops by 0.075 without intervention, 0.086 with the span window and 0.122 with the fixed window. The gap comes from how often each window's target reflects the right language. We label each target by its nearest language centroid at the last edited block. The span window's target is the chunk's own language on 88% of chunks. The fixed window's target is on only 79% of chunks whose window stays within one language, and on 65% of chunks whose window crosses the switch and so mixes both languages.

TTS details. The edit is applied at blocks 14 to 22 of the language model at rank 15, the blocks and rank of the steering edit (Appendix E). The sentence is generated in chunks of about 12 phonemes, each closed at the next word boundary and generated in the full context of the phonemes and speech tokens before it; the first chunk is never edited, since nothing precedes it. The fixed window is the chunk plus about six phonemes on either side, cut at word boundaries. It is generated with no history and no edit, and the mean of its hidden states at each edited block is the target for every position of the chunk. In the span window arm the chunks are the two language spans. Each sentence is generated with four sampling seeds, which are averaged before scoring. The basis and the oracle's language centroids are fitted on real audio read teacher-forced, while the windowed target and the edit act on the model's own generated states.

Examples. Table 14 shows two sentences in which all three models replace unshared phonemes with the other language's counterparts when they process the whole utterance, and WLE restores them. In (a), the French context turns English [ɪ] into [ʁ] and [ʊ] into [y]. In (b), the English context turns Spanish [ɾ] into [ɹ], and the TTS model also turns [x] into [h]. In the TTS model, these substitutions appear in three or four of the four sampling seeds without the edit and in none with WLE. These are the clearest cases on SwitchLingua: in each model, $r_{\min}$ falls by at least 0.5 from SEP to TOG, and WLE brings it back to within 0.1 of SEP. Seven recordings meet both conditions in

Model	Benchmark	Sliding window	WLE	WLE, random subspace
ΔPER (pp)				
POWSM	TIMIT	$+0.96$ [$+0.85, +1.07$]	$+0.07$ [$+0.02, +0.11$]	$+0.01$ [$-0.02, +0.04$]
	L2-ARCTIC	$+1.55$ [$+1.37, +1.74$]	$+0.10$ [$+0.01, +0.19$]	$+0.02$ [$-0.02, +0.07$]
	GMU accent	$+2.14$ [$+2.00, +2.29$]	-0.32 [$-0.39, -0.26$]	-0.04 [$-0.07, -0.01$]
	DoReCo	$+1.97$ [$+1.77, +2.19$]	-0.09 [$-0.19, +0.02$]	-0.13 [$-0.18, -0.09$]
PhoneticXeus	TIMIT	$+1.33$ [$+1.22, +1.44$]	$+0.08$ [$+0.05, +0.11$]	$+0.01$ [$-0.01, +0.03$]
	L2-ARCTIC	$+3.22$ [$+3.01, +3.43$]	$+0.14$ [$+0.06, +0.23$]	$+0.05$ [$-0.00, +0.10$]
	GMU accent	$+2.71$ [$+2.55, +2.87$]	-0.09 [$-0.17, -0.02$]	-0.02 [$-0.05, +0.01$]
	DoReCo	$+2.08$ [$+1.88, +2.30$]	-0.22 [$-0.32, -0.13$]	-0.15 [$-0.19, -0.10$]
ΔPFER (pp)				
POWSM	TIMIT	$+0.47$ [$+0.40, +0.53$]	$+0.01$ [$-0.02, +0.04$]	$+0.01$ [$-0.01, +0.03$]
	L2-ARCTIC	$+0.31$ [$+0.20, +0.42$]	-0.00 [$-0.06, +0.05$]	$+0.00$ [$-0.03, +0.04$]
	GMU accent	$+0.73$ [$+0.66, +0.81$]	-0.07 [$-0.11, -0.03$]	-0.04 [$-0.05, -0.02$]
	DoReCo	-0.45 [$-0.61, -0.28$]	$+0.05$ [$-0.02, +0.11$]	-0.15 [$-0.18, -0.12$]
PhoneticXeus	TIMIT	$+0.49$ [$+0.43, +0.55$]	$+0.03$ [$+0.01, +0.04$]	$+0.01$ [$-0.01, +0.02$]
	L2-ARCTIC	$+0.50$ [$+0.40, +0.61$]	$+0.03$ [$-0.01, +0.08$]	-0.02 [$-0.05, +0.02$]
	GMU accent	$+1.01$ [$+0.94, +1.08$]	-0.04 [$-0.07, -0.00$]	-0.01 [$-0.02, +0.01$]
	DoReCo	$+1.06$ [$+0.94, +1.20$]	$+0.15$ [$+0.11, +0.20$]	$+0.05$ [$+0.03, +0.08$]

Table 13: ΔPER (top) and ΔPFER (bottom), in percentage points, behind Table 2, with 95% bootstrap intervals over clips.

Model	Word	Reference	SEP	TOG	WLE
(a) English words in a French–English sentence: *Le Président américain a annoncé une décision importante aujourd'hui. Can you believe he **revoked** Biden's **security** clearance?*					
TTS	*revoked*	**ɹ**ɪvoʊkt	ɹɪvoʊkt	ʁivɔk	ɹɪvoʊkt
	security	sɪkjʊ**ɹ**ɪti	sɪkjʊɹɪdi	sikjuʁiti	sɪkjʊɹɪdi
PhoneticXeus	*revoked*	**ɹ**ɪvoʊkt	ɹɪvoʊkt	ʁivɔkt	ɹɪvoʊkt
	security	sɪkjʊ**ɹ**ɪti	sɪkjʊɹəti	skyʁiti	skjʊɹiti
POWSM	*revoked*	**ɹ**ɪvoʊkt	ɹɪvoʊktu	ʁəvɔkt	ɹəvokt
	security	sɪkjʊ**ɹ**ɪti	sɪkjʊɹəti	skyʁiti	sɪkʊɹəti
(b) Spanish words in a Spanish–English sentence: *Swindon Borough Council está **trabajando** en este proyecto. It's a big step toward preserving our paleontological heritage.*					
TTS	*trabajando*	t**r**aba**x**ando	traβaxando	tɹɑpəhũndoʊ	trabaxando
PhoneticXeus	*trabajando*	t**r**aba**x**ando	trabaxando	tɹəbəæ̃ndoʊ	trabaxando
POWSM	*trabajando*	t**r**aba**x**ando	traβaxando	tɹəbəkæ̃ndoʊ	trabaxando

Table 14: **Examples of WLE.** Each model's transcript of the words in bold when it processes each span alone (SEP), the whole utterance (TOG), and the whole utterance with WLE. For the TTS model, WLE uses the span window, and the transcript is PhoneticXeus's reading of the synthesized span (first sampling seed). In the reference, the unshared phonemes of the word's own language are in bold. In the transcripts, phonemes that only the other language uses are in red.

all three models, and we chose these two from them to cover both directions. Among all recordings with a lower $r_{\min}$ under TOG than under SEP, WLE recovers at least half of the drop in 79% for the TTS model, 57% for PhoneticXeus and 43% for POWSM.

H WINDOW WIDTH

Table 15 repeats the WLE repair with windows of 1, 2, 3 and 4 s, each with a hop of half its width, on both recognizers and both datasets. WLE reduces interference at every width, and the random subspace leaves it at its unrepaired level at every width (Δ within 0.02 of no intervention). No single width gives the least interference everywhere. Under WLE, PhoneticXeus on SwitchLingua does

Condition	1 s	2 s	3 s	4 s
PhoneticXeus, SwitchLingua (no intervention: .507)				
Sliding window	.223 [.188, .260]	**.202** [.160, .243]	.207 [.163, .250]	.246 [.203, .290]
WLE	**.185** [.146, .222]	.214 [.171, .257]	.259 [.216, .304]	.306 [.263, .349]
POWSM, SwitchLingua (no intervention: .448)				
Sliding window	.245 [.203, .287]	**.153** [.120, .187]	.184 [.151, .218]	.211 [.171, .251]
WLE	.321 [.282, .359]	**.288** [.249, .327]	.294 [.255, .333]	.330 [.289, .371]
PhoneticXeus, Spliced FLEURS (no intervention: .186)				
Sliding window	.108 [.082, .135]	.037 [.016, .059]	.039 [.017, .062]	**.022** [.001, .042]
WLE	.075 [.050, .101]	**.047** [.023, .073]	.052 [.027, .077]	.054 [.030, .079]
POWSM, Spliced FLEURS (no intervention: .459)				
Sliding window	.214 [.184, .243]	.081 [.060, .102]	.048 [.029, .068]	**.036** [.017, .055]
WLE	.322 [.293, .351]	.234 [.205, .264]	.198 [.170, .227]	**.185** [.158, .215]

Table 15: Interference Δ ($\downarrow$) by window width, with 95% bootstrap intervals over recordings or clips. The unrepaired Δ is in parentheses, and the lowest value in each row is shown in **bold**.

	SwitchLingua		Spliced FLEURS	
Basis	PhoneticXeus	POWSM	PhoneticXeus	POWSM
No repair, Δ	.507	.448	.186	.459
Sliding window (2 s), Δ	.202	.153	.037	.081
Share of the window's reduction				
All 16 languages, rank 15 (WLE)	.96	.54	.93	.60
7 code-switched languages, rank 6	.88	.52	.92	.55
8 Indo-European languages, rank 7	.92	.52	.92	.50
All 16 minus one pair language, rank 14	.83 – .94	.43 – .53	.77 – .92	.51 – .59
All 16 minus both pair languages, rank 13	.24 – .34	.00 – .05	.50 – .67	.29 – .35
8 non-Indo-European languages, rank 7	.05	$-.01$	$-.01$	.09
9 languages outside all evaluated pairs, rank 8	.03	$-.04$	$-.03$	.03
Random basis, ranks 6–15	.01 – .04	$-.03$ – .00	$-.01$ – .08	.01 – .05

Table 16: Share of the sliding window's interference reduction recovered by WLE with each basis. Each value is $(\Delta_{\text{no repair}} - \Delta_{\text{WLE}}) \,/\, (\Delta_{\text{no repair}} - \Delta_{\text{window}})$. The full 16-language basis recovers most of the window's effect. Bases missing one language of the evaluated pair show partial repair; on SwitchLingua, whose pairs all contain English, bases missing both languages recover a quarter to a third of the window's reduction in PhoneticXeus and no more than a random basis in POWSM.

best at 1 s and POWSM on Spliced FLEURS does best at 3–4 s, while the other two settings change little between 2 and 3 s. We report 2 s throughout.

The monolingual cost of WLE stays small at the widths we measured. On the benchmarks of Table 13, WLE changes PER by -0.48 to $+0.76$ pp at 1 s and by -0.40 to $+0.18$ pp at 3 s (the latter on clips of at least 4.5 s), and PFER by -0.08 to $+0.40$ pp at 1 s and by -0.08 to $+0.13$ pp at 3 s. The sliding window degrades PER by 4.9 to 13.5 pp at 1 s and by 0.4 to 2.7 pp at 3 s. It degrades PFER by 1.9 to 7.0 pp at 1 s, and at 3 s it changes PFER by -0.9 to $+0.6$ pp. A narrow window is therefore only usable when it is restricted to the language subspace.

I WLE WITH DIFFERENT LANGUAGE SUBSETS

Setup. We test whether WLE requires all 16 training languages in its basis. We refit the subspace (Appendix E) on various subsets of the 16 languages and measure how much interference each basis removes, expressed as a share of the 2 s sliding window's reduction (1.0 = matches the window, 0.0 = no effect). Table 16 shows results across both datasets and both recognizers.

Key findings. The table reveals a clear pattern: WLE works when the basis contains at least one language of the pair being evaluated. Specifically:

	SwitchLingua		Spliced FLEURS	
Condition	PhoneticXeus	POWSM	PhoneticXeus	POWSM
No intervention	.507 [.462, .551]	.448 [.404, .490]	.186 [.156, .218]	.459 [.425, .494]
Sliding window (2 s)	.202 [.160, .243]	.153 [.120, .187]	.037 [.016, .059]	.081 [.060, .102]
WLE	.214 [.171, .257]	.288 [.249, .327]	.047 [.023, .073]	.234 [.205, .264]
WLE, random subspace	.494 [.449, .539]	.451 [.407, .494]	.178 [.148, .211]	.440 [.406, .475]
Oracle	.108 [.073, .144]	.121 [.080, .162]	.028 [.005, .050]	.143 [.118, .169]
Oracle, random subspace	.505 [.460, .549]	.452 [.409, .493]	.185 [.155, .216]	.448 [.414, .483]

Table 17: Interference Δ (lower is better) with 95% intervals. The oracle reduces interference beyond WLE on all four cells, and beyond the sliding window on three. The same edits through a random subspace have no effect, confirming the reduction depends on the language subspace.

(1) The full basis and the 7 code-switched languages perform similarly, showing that languages outside the evaluation pairs contribute little.

(2) Removing one language of the pair degrades performance partially. Removing both degrades it further: on SwitchLingua the basis recovers 0.24–0.34 of the window's reduction in PhoneticXeus and 0.00–0.05 in POWSM, the level of a random basis.

(3) Bases containing none of the evaluation languages (the 9 uninvolved languages, the 8 non-Indo-European languages on non-Korean pairs) perform no better than random, despite displacing representations by a comparable amount. The subspace directions must include the relevant languages to be effective.

(4) The 16 language centroids sit in near-general position: removing both languages of a pair leaves only 4–15% of their pairwise direction in the remaining span, comparable to a random subspace of the same rank (1–3%).

J ORACLE REPAIR (KNOWN LANGUAGE)

Setup. WLE estimates the local language from a windowed encoding. As a reference, we test an oracle that knows the true language boundary and sets each frame's subspace coordinates to the centroid of its actual language:

$$\mathbf{h} \leftarrow \mathbf{h} - \mathrm{proj}_{\mathrm{lang}}(\mathbf{h}) + \mathrm{proj}_{\mathrm{lang}}(\mathbf{c}_{\ell(t)}), \tag{3}$$

where $\ell(t)$ is the language of frame t. All other settings (basis, rank, blocks, scoring) match WLE in Table 1. A random subspace of equal rank with the same per-frame targets serves as a control. Both run on SwitchLingua and Spliced FLEURS with both recognizers. In the TTS model the oracle sets each span's coordinates to the centroid of the span's known language, with the same random subspace control; Table 12 gives its rows.

Results. The oracle removes more interference than WLE in all four cells, with the largest gap on POWSM (Table 17). It does not remove all of it: Δ stays between 0.108 and 0.143, except for PhoneticXeus on Spliced FLEURS (0.028). To check whether the remaining interference is phoneme-specific, we subtract the shared-phoneme drop from the unshared-phoneme drop (exact match) for each recording or clip (Table 5). On SwitchLingua, this gap is 0.178 (PhoneticXeus) and 0.230 (POWSM) with no intervention and 0.058 and 0.144 under WLE. Under the oracle it is close to zero (0.002 and 0.011, both CIs include zero), so the remaining interference affects shared and unshared phonemes equally. On Spliced FLEURS, the gap also disappears for PhoneticXeus (0.018 $[-0.005, 0.039]$). POWSM, however, keeps 0.086 of its 0.285, so on POWSM these 15 dimensions do not explain the whole effect. In the TTS model, the oracle and WLE with the span window leave similar interference ($\Delta = 0.106$ vs. 0.093; paired difference -0.013 $[-0.031, 0.006]$). The two methods differ only in their target, so the language estimate the model forms from the span alone works as well as the true language's centroid.

Caveats. The oracle edit is not a fair comparison with WLE, for two reasons. First, it requires knowing the language boundary. Second, the oracle subtracts a training corpus centroid from test

Language tag	SwitchLingua	Spliced FLEURS
<unk>	.414 [.371, .456]	.445 [.408, .482]
Each span's own language	.102 [.070, .134]	.248 [.213, .284]
Difference	.312 [.263, .360]	.197 [.160, .235]

Table 18: **POWSM's attention decoder with each span's own language tag.** Interference Δ (lower is better), the drop in worst span recall of unshared phonemes from SEP to TOG, with 95% bootstrap intervals, on the recordings (266) and clips (274) that both decodes can score. Difference: Δ under <unk> minus Δ under the span's own tag, paired per recording or clip.

audio, so the edit also moves each frame toward the training distribution, not just toward the correct language. WLE avoids this because it subtracts two encodings of the same audio. We therefore treat the oracle as a loose upper bound, not a target for WLE to match. The second reason applies to the TTS oracle as well, whose centroids come from real audio while the span window's target comes from the model's own generated states.

Language tag as an oracle. When POWSM's attention decoder receives each span's own language tag, it shows much less interference, and the remaining interference stays above zero on both sets (Table 18). POWSM takes one language tag per utterance, as the first token of its decoder prompt, so we decode each span alone under its own language's tag and the whole recording once under each of its two languages' tags, and we read each span from the decode under its own tag. Like the oracle edit, this needs the language of each span and the switch point. On Spliced FLEURS the decoder often ends the transcription right after the tag; in that case we take the best non-empty hypothesis of the same beam. Paired per recording or clip, the tag lowers Δ by 0.312 [0.263, 0.360] on SwitchLingua and by 0.197 [0.160, 0.235] on Spliced FLEURS. The CTC head of the same encoder receives no tag and shows as much interference as the attention decoder under <unk> (Table 5).

K Interference in bilingual speech

Weinreich (1953) describes interference as deviation from the norms of a language that bilinguals produce because of their" familiarity with more than one language." and Grosjean (2001) calls the momentary form of this, caused by the language not currently being spoken, *dynamic interferences*. Studies of code-switched speech measure this influence mostly through voice onset time (VOT). Stops near a switch shift toward the other language, by about 28 ms in a laboratory switching task (Antoniou et al., 2011) and by 2 to 6 ms in spontaneous conversation (Piccinini & Arvaniti, 2015; Fricke et al., 2016), and a vowel contrast that only Catalan has becomes slightly more similar to Spanish when Spanish words are mixed in (Simonet, 2014). The shift often affects only one of the two languages (Olson, 2013; Balukas & Koops, 2015), and some studies find none (Grosjean & Miller, 1994; Gavino & Goldrick, 2023). On the perception side, the language context shifts where bilingual listeners place a category boundary (Elman et al., 1977). In people, interference is therefore small and gradient, and the models we study show a much larger form of it, replacing whole phonemes in both recognition and synthesis. In multilingual machine learning, *negative interference* names a different effect: languages that share model capacity during training lower each other's performance (Wang et al., 2020; Conneau et al., 2020). The interference we study arises at inference, from the context of a single input.