Linear Script Representations in Speech Foundation Models Enable Zero-Shot Transliteration

Ryan Soh-Eun Shim[1,2], Kwanghee Choi[3], Kalvin Chang[5], Ming-Hao Hsu[6], Florian Eichin[1,2], Zhizheng Wu[6], Alane Suhr[5], Michael A. Hedderich[1,2], David Harwath[3], David R. Mortensen[4], Barbara Plank[1,2]

[1]LMU Munich, Germany, [2]Munich Center for Machine Learning, Germany,
[3]University of Texas at Austin, USA, [4]Carnegie Mellon University, USA,
[5]University of California, Berkeley, USA, [6]Chinese University of Hong Kong, Shenzhen, China

Correspondence: s.shim@lmu.de

Abstract

Multilingual speech foundation models such as Whisper are trained on web-scale data, where data for each language consists of a myriad of regional varieties. However, different regional varieties often employ different scripts to write the same language, rendering speech recognition output also subject to non-determinism in the output script. To mitigate this problem, we show that script is linearly encoded in the activation space of multilingual speech models, and that modifying activations at inference time enables direct control over output script. We find the addition of such script vectors to activations at test time can induce script change even in unconventional language-script pairings (e.g. Italian in Cyrillic and Japanese in Latin script). We apply this approach to inducing post-hoc control over the script of speech recognition output, where we observe competitive performance across all model sizes of Whisper.[1]

1 Introduction

Multilingual Automatic Speech Recognition (ASR) models such as Whisper (Radford et al., 2023) map speech in different languages to their corresponding written form, where such written language is represented in various scripts, such as Cyrillic, Chinese, or Latin characters. Similar to how one can translate between languages, it is also possible to write the same spoken word in different scripts via *transliteration*. For instance, a Russian audio sample of the word meaning *computer*, typically written in Cyrillic script as компьютер, can be written in Latin characters as *kompyuter* instead.

Moreover, many languages are written in multiple scripts. For example, the pluricentric language of Serbo-Croatian encompasses multiple mutually-intelligible varieties and multiple orthographic standards. Within Serbo-Croatian, Serbian, for instance, is written in both Cyrillic and Latin characters, while Croatian leans heavily towards the usage of only the Latin script. Similarly, Mandarin is written in Traditional and Simplified Chinese characters, where the former is primarily used in Taiwan and the latter in China. Thus models trained on web-scale data have been trained on speech within the same language that has transcriptions in different scripts, leading to non-determinism. As scripts carry sociopolitical connotations, finer control over the output script may help better adapt the output to different scripts (e.g. Latin script in Croatian contexts (Jovanović, 2018); Traditional Chinese in Taiwanese contexts (Su and Chun, 2021)). As such, post-hoc control over the script of the output is desirable.

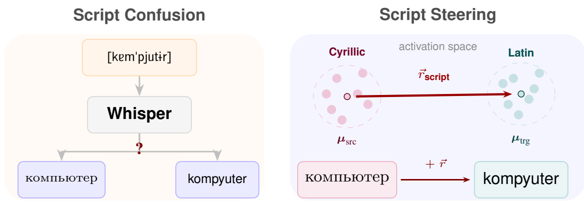

Figure 1: **Left:** Multilingual ASR models like Whisper exhibit *script confusion*: the same speech may be transcribed in different scripts non-deterministically. **Right:** We find script is encoded as a linear direction $\vec{r}_{\text{script}}$ in Whisper's activation space. Adding this vector at inference time induces the target script, enabling transliteration without any training.

For inducing post-hoc control, an increasing body of work in text-based models has found high-level concepts such as refusal (Arditi et al., 2024; Wang et al., 2025b) and truthfulness (Li et al., 2024; Marks and Tegmark, 2024) to be represented as a linear direction in the activation space of neural models (Park et al., 2024). Such findings have been used to induce test-time interventions that steer generation output to the specified concept direction with simple vector arithmetic, where directions can be added to activations at test time to induce behavior (Turner et al., 2023a). In this work, we

[1]Code is at: https://github.com/mainlp/transliteration

Findings of the Association for Computational Linguistics: ACL 2026, pages 9532–9544
July 2-7, 2026 ©2026 Association for Computational Linguistics

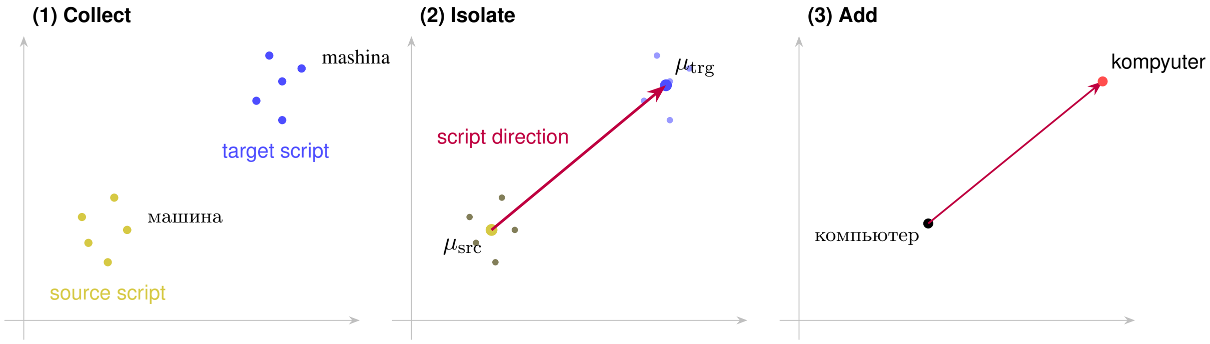

Figure 2: Illustration of our method for extracting script vectors. (1) For each decoder layer, we collect activations in the source (yellow) and target (blue) script. (2) We isolate a script direction by subtracting the mean of the source activations from the mean of the target activations for each layer. (3) At test time, we add the script direction to the activations to induce the transcription to be in the target script.

leverage this strategy *to induce script change in speech foundation models* by way of such steering. Our findings are as follows:

Script confusion hinders control over ASR output. We propose script vectors as a way to address the problem of *script confusion*, where languages written in multiple scripts suffer from lack of control over the script of speech recognition outputs.

Script is a linear direction in activation space. We find *script* to be represented linearly in multilingual ASR models, enabling post-hoc control over the script of the ASR output by way of adding script vectors to the hidden states. We show that such an approach can induce transliterations in novel directions (e.g. Italian in Cyrillic characters) and mitigate script confusion without any training, requiring as few as 1 good quality example pair.

In-depth analysis of script vectors. We perform an in-depth analysis of the layer-wise presence and linguistic characteristics of transliterations induced with script vectors, where we find script to be linearly represented in all decoder layers of an encoder-decoder model, and that the induced transliterations tend to be sound-based, reflecting actual pronunciation.

2 Methodology

We seek to isolate specific script directions in a speech foundation model's activation space with *script vectors* such that the addition of the found script vector to the activations can control the script of the generated text at test time. In this work, we focus on encoder-decoder speech foundation models, where the encoder processes speech and the decoder is an auto-regressive language model that processes text.

2.1 Script Steering

We obtain one steering vector per decoder layer by computing the difference in mean activations (Arditi et al., 2024; Marks and Tegmark, 2024) between samples of two given scripts. Our methodology consists of three steps: activation collection and filtering, direction isolation, and activation addition, where Figure 2 provides an overview.

Activation collection Let D be the hidden size and L the number of decoder layers. For each audio sample a, we decode two transcriptions with different text prompts (fed into the decoder): a *source* prompt p^{SRC} that contextually biases the transcription towards the source script and a *target* prompt p^{TRG} that biases towards the target script (where the scripts are in the same language for this work).[2] For a text sequence $z(p, a)$ decoded from audio a given prompt p, let $\mathbf{x}_\ell(p^{SRC}, a)_t \in \mathbb{R}^D$ be the activation (output) of decoder layer ℓ at token position t given source prompt p^{SRC}, and $\mathbf{x}_\ell(p^{TRG}, a)_t \in \mathbb{R}^D$, the corresponding activation under the target prompt p^{TRG}. For each layer ℓ, we average activations per example via a mean pool over the token positions:

$$\bar{\mathbf{x}}_\ell(p, a) := \frac{1}{|z|} \sum_{t \in z(p,a)} \mathbf{x}_\ell(p, a)_t \quad (1)$$

Given a training set $\mathcal{D}$, we then compute the mean layer-wise activations using the source prompt $\mathcal{D}_{\text{SRC}} := \{\bar{\mathbf{x}}_\ell(p^{SRC}, a) | a \in \mathcal{D}\}$ and the

[2]We provide the prompts we use in Appendix A.

Language	Predicted romanization (ours)	Deterministic romanization	Original transcript
Russian	Del Patro imel rane **premuscestvo** vo vtorom seti, no **eta** takzhe potrebvala taj brejka posle dostizzenia 6-6.	del potro imel rannee **preimushchestvo** vo vtorom sete no **eto** takzhe potrebovalo tai-breika posle dostizheniya 6-6	дель потро имел раннее преимущество во втором сете но это также потребовало тай-брейка после достижения 6-6
Hindi	Kuch anvon me aar sthir kendrak hota hai jiska matlab yaha ki **unme** thode ya bina kisi jhatke se **tutne** ki **pravati** hoti hai.	kuch annuom mem asthir kemdrak hotaa hai jisakaa matalab yah hai ki **unamem** thodde yaa binaa kisii jhattake se **ttuuttane** kii **pravrtti** hotii hai	कउचहह अब्ओमर्मँएम् असतहइर कएम्दरअक होता हअइ जइसका मअतअलअब यअह हअइ कइ उनअमएम् तहोङए या बइना कइसई जहअथअकए सए थऊथअनए कई परअवङततइ होतई हअइ
Greek	**Skeftite ti diadromi** me ski os mia **omia kalipsi apostasis** pezoporondas	**skefteite ten diadrome** me ski os mia **omoia kalypse apostases** pezoporondas	σκεφτείτε την διαδρομή με σκι ως μια όμοια κάλυψη απόστασης πεζοπορώντας
Japanese	Hong Kong no chiheisen wo egaite tachinarabu biru-gun **wa**, victoria harbour no mizu no sonzai ni yote kirabiyakana bougurafu ni tatoerarete imasu.	honkon no chiheisen wo egai te tachi narabu biru gun **ha** bikutoria haabaa no mizu no sonzai niyotte kirabiyakana bou gurafu ni tatoe rareteimasu	香港の地平線を描いて立ち並ぶビル群は ビクトリアハーバーの水の存在によって きらびやかな棒グラフに例えられています
Korean	Internet은 집tanjeogin communicationgwa saramgwa saramgane communicationyosoreul modu **gachugo itda**.	inteoneseun jibdanjeogin keomyunikeisyeongwa saramgwa saram ganyi keomyunikeisyeon yosoreul modu **gajcugo issda**	인터넷은 집단적인 커뮤니케이션과 사람과 사람 간의 커뮤니케이션 요소를 모두 갖추고 있다

Table 1: Sample predictions of our romanization with Whisper large-v2. Parts where deterministic romanization output does not reflect actual pronunciation but our prediction does are typeset in bold.

target prompt $\mathcal{D}_{\text{TRG}} := \{\bar{\mathbf{x}}_\ell(p^{TRG}, a) | a \in \mathcal{D}\}$, respectively:

$$\mathbf{v}_\ell^{\text{SRC}} := \frac{1}{|\mathcal{D}_{\text{SRC}}|} \sum_{\bar{\mathbf{x}} \in \mathcal{D}_{\text{SRC}}} \bar{\mathbf{x}}, \quad \mathbf{v}_\ell^{\text{TRG}} := \frac{1}{|\mathcal{D}_{\text{TRG}}|} \sum_{\bar{\mathbf{x}} \in \mathcal{D}_{\text{TRG}}} \bar{\mathbf{x}}. \tag{2}$$

To avoid noise due to unsuccessful prompting (*i.e.*, the model outputting the non-target script), we additionally apply filtering while collecting and averaging activations. We retain activations where the transcriptions are likely to be in the target script, which results in cleaner activations from which the vector can be isolated. We use the normalized Levenshtein edit distance $\bar{d}_{\text{edit}}$ between the transcription prediction $z(p, a)$ and the ground truth $\hat{z}(p)$ given script prompt p:

$$\bar{d}_{\text{edit}}(z, \hat{z}) := \frac{d_{\text{edit}}(z, \hat{z})}{\max(|\text{z}|, |\hat{\text{z}}|)}. \tag{3}$$

Based on the threshold hyperparameter $\theta \in \mathbb{R}$, we only keep the activations satisfying:

$$\bar{d}_{\text{edit}}(z(p^{\text{SRC}}, a), \hat{z}(p^{\text{SRC}})) < \theta, \quad \bar{d}_{\text{edit}}(z(p^{\text{TRG}}, a), \hat{z}(p^{\text{TRG}})) < \theta \tag{4}$$

i.e. the samples for which the two prompted predictions match source and target script respectively.

Direction isolation Based on the linear representation hypothesis (Park et al., 2024) where high-level concepts are represented as directions in activation space, we use $\mathbf{v}_\ell^{\text{SRC}}$ and $\mathbf{v}_\ell^{\text{TRG}}$ to obtain the *steering vector* $\mathbf{r}_\ell$ for layer ℓ:

$$\mathbf{r}_\ell := \mathbf{v}_\ell^{\text{TRG}} - \mathbf{v}_\ell^{\text{SRC}} \in \mathbb{R}^D. \tag{5}$$

Activation addition During transcription, at each step of decoding, we edit the *current last-token* activation $\mathbf{h}_{t,\ell} \in \mathbb{R}^D$ at layer ℓ by adding the script vector to the activation:

$$\mathbf{h}_{t,\ell}^{\text{steered}} := \mathbf{h}_{t,\ell} + \sigma \mathbf{r}_\ell, \tag{6}$$

where σ controls the strength of the direction. In our experiments, we follow Wang et al. (2025a) in steering all layers at once, using the same sigma value for all layers. We show in Section 5 that steering all layers is arguably preferable, due to script being linearly represented in all layers. In our experiments, we tune sigma on a validation set, the details of which are described in Section 3.4.

3 Experiments

We evaluate our setup on two use cases: (1) mitigating script confusion, where multiple scripts occur within a single language's speech recognition output, and (2) transliteration, where we transcribe speech from a source language in the orthography (script) of a target language, where the target script is not associated with the source language (e.g. transcribing Italian in Cyrillic script). We focus our transliteration experiments on romanization and cyrillization, due to the prevalence of languages written in these two scripts. We provide further details below.

3.1 Script confusion

The goal of the script confusion setting is twofold: to measure the extent to which transcriptions are unstable in script, and whether our script steering mitigates such an instability. We focus our experiments on two languages that are written in multiple scripts: Serbian (written in Latin and Cyrillic script) and Mandarin (written in Traditional and Simplified Chinese). For generating ground truths for these script pairs, we employ deterministic, language-specific tools: cyrtranslit (Labrèche, 2025) for Serbian and OpenCC (Kuo et al., 2024) for Mandarin. We compare the following approaches:

no-prompt We empirically measure the likelihood of speech recognition being in a given script, by directly running the model without any modifications, and measuring its output against the gold transcription in the target script. This provides a notion of how much speech recognition in a given language is likely to be unstable in the script. In this setup, the only input is the audio itself.

prompt For our primary baseline, we measure the performance of contextually biasing the output script by prepending the decoder with text in the target script.[3]

steer (ours) Finally, we measure the extent to which our proposed activation steering method (Section 2) can mitigate script confusion, where we isolate directions in activation space that encode script, and add them to activations at test time. The same prompts as the **prompt** baseline are used to obtain script vectors. In addition, we include results of learning script vectors using a single example, which we label as **one-shot**. We describe the one-shot setup in more detail in Section 5.

3.2 Transliteration

In the transliteration setting, we evaluate romanization and cyrillization, for which deterministic tools are available (Hermjakob et al., 2018; Labrèche, 2025). For romanization, we evaluate on five languages with diverse scripts that are supported by the tools uroman (Hermjakob et al., 2018) and pykakasi (Miura et al., 2024), which include Japanese, Korean, Russian, Greek, and Hindi. For cyrillization, we rely on International Components for Unicode[4] (ICU) to transliterate the same languages as above into Cyrillic, but instead of Russian we evaluate on Italian, as the former is already in Cyrillic. For source-target script pairs not directly implemented in ICU, we use the deterministic romanization tools above to generate romanization then pass through ICU to generate Cyrillic.

In addition to **no-prompt** and **prompt** from section 3.1, we evaluate steering with script vectors in two different ways:

zero-shot (ours) We empirically found that script vectors obtained for a specific source-target pair of scripts generalize zero-shot to other languages and scripts to a reasonable extent. As such, we measure the extent to which the romanization and cyrillization directions (obtained from the Serbian script confusion experiments) when applied to other languages successfully induces the corresponding scripts.

pseudo label (ours) As the zero-shot approach has shown to induce transliterations with varying degrees of success, we hypothesize that the transliterations induced in such a manner can be used as pseudo transcriptions to learn script vectors better suited for the new language. As such, we repeat the script steering procedure using the steered model to induce script vectors, where we collect activations on the steered model and learn a script vector on these newly collected activations.

3.3 Evaluation

The goal of both settings is to transcribe speech using the correct target script. To quantify the degree to which the predicted text matches the ground

[3] Based on: https://github.com/openai/whisper/discussions/277

[4] https://github.com/unicode-org/icu

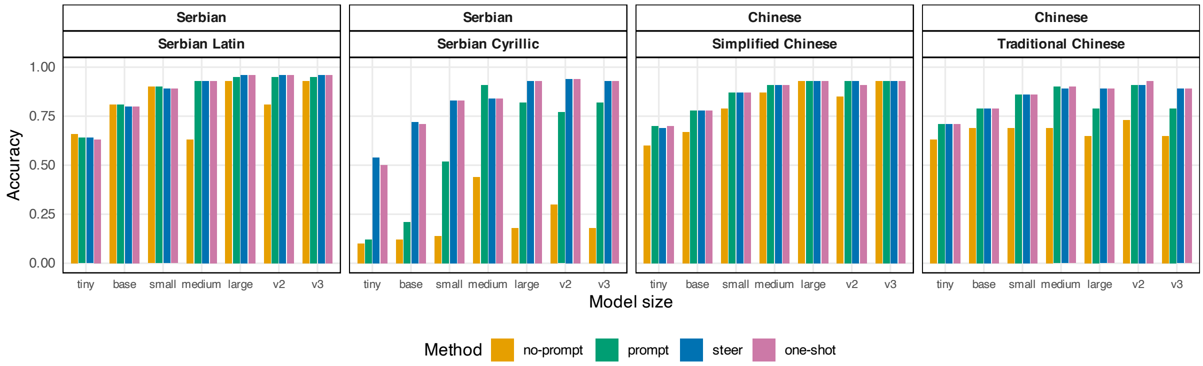

Figure 3: Script confusion mitigation accuracy. Line color shows method. The x-axis shows model size, while the y-axis shows normalized edit similarity to ground truth in target script (Equation 3).

truth, we compute the edit distance of our transcription against such ground truth at the character level, normalized by the length of the reference or hypothesis (whichever is longer). We then subtract the resulting value from 1 to convert the distance into a similarity metric for ease of interpretation (Equation 3). In our results, we term this *accuracy*, where 1.0 is text that fully matches the ground truth. For preprocessing, as our goal is to quantify the degree to which correct transliteration occurs, we remove punctuation, spacing, and portions of the text that are not in the target script.

3.4 Dataset and Hyperparameters

We perform our experiments in both settings with the FLEURS (Conneau et al., 2023) dataset, a dataset with parallel speech and text transcriptions in 100+ languages.[5] For each source and target script, we collect activations on 10 samples respectively from the train set, where the first 10 audio samples that exceed the θ threshold in both source and target scripts are collected. For θ that serves as the threshold for filtering activations, we use $\theta = 0.4$ for script confusion and romanization. For cyrillization, due to lower recognition performance than romanization, we increase to $\theta = 0.8$. For σ that controls the strength of the script vector, we perform a grid search over $\sigma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ on the validation set in FLEURS. We report our results on the test sets of each language using the best σ.[6]

[5]We include dataset statistics in Table 11.

[6]For script confusion, best here means the sigma resulting in the highest mean accuracy. For transliteration, due to zero-shot accuracy being unevenly distributed over samples, we use the sigma resulting in the single highest (i.e. max) accuracy. For script confusion, the best σ under all settings is 0.1. For transliteration, we report σ in Table 12.

As for the speech foundation model, we leverage Whisper (Radford et al., 2023), where we evaluate our approach on all model sizes: tiny, base, small, medium, large, v2, and v3. for the script confusion setting. For transliteration, we focus on large-v2, the best performing model size in script confusion, which prior work has also found to perform the best across all model sizes (Ma et al., 2025).

4 Results

We next present results for our two key setups.

4.1 Script Confusion

Script vectors induce the correct target script in smaller models even when prompting does not. Figure 3 shows our results on mitigating script confusion. We observe that the effectiveness of prompting for controlling the output script tends to be more limited in smaller models, but improves with the size of the model. We see this particularly for inducing Cyrillic in Serbian. For instance, in Whisper tiny, prompting only slightly induces more Cyrillic script than running inference directly, whereas activation steering significantly increases Cyrillic script accuracy.

Script confusion mitigation effectively induces the less frequent scripts of a language. For both Chinese and Serbian, we observe that prompting and steering both induce improvements over the no-prompt baseline primarily for the less frequent (and thus lower resource) script of a given language (*i.e.*, Serbian in Cyrillic and Traditional Chinese). For Serbian in Latin script and Simplified Chinese, the improvements are less pronounced, due to inference already producing higher proportions of transcriptions in these scripts.

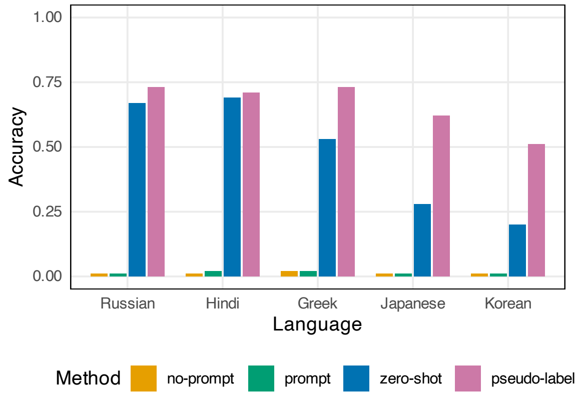

Figure 4: Romanization accuracy across different languages for Whisper large-v2. The x-axis shows languages, while the y-axis shows normalized edit similarity against ground truth in target script (Equation 3).

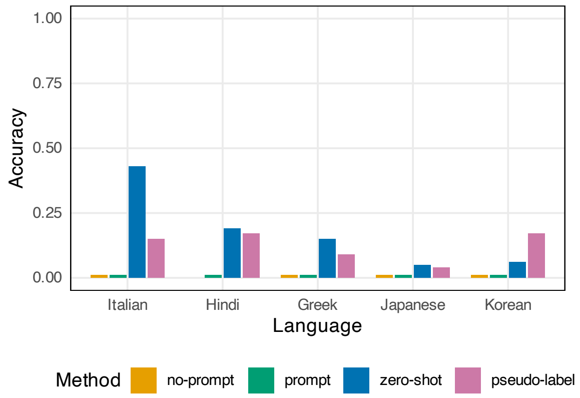

Figure 5: Cyrillization accuracy across different languages for Whisper large-v2. The x-axis shows languages, while the y-axis shows normalized edit similarity against ground truth in target script (Equation 3).

4.2 Transliteration

Script vectors can generalize across languages. Figure 4 shows the results of directly applying our romanization vector derived from Serbian on different languages, along with results from using the induced transcriptions as pseudo labels to learn a direction adapted to the specific language. We observe that for Hindi and Russian, zero-shot performance is already at 69 and 67%, respectively, suggesting romanization direction to be broadly similar across languages. The further adaptation step with pseudo labels boosts performance across all five languages, further supporting the hypothesis of similarity between romanization direction across languages.

Script vectors can induce transliterations in unconventional language-script pairs. Figure 5 shows our results for cyrillization. In comparison with romanization, cyrillization observes more limited performance across most languages, with the exception of Italian, which reaches 43% accuracy. We provide examples of our Cyrillic transliterations in Table 2. We hypothesize the exceptional performance of Italian in cyrillization to be due to it being written in the high-resource Latin script, which the majority of Whisper's training data is in (Radford et al., 2023). This is also empirically supported by the general success of romanization for all source languages examined in Section 4.2.

5 Analysis

In this section, we present further analyses on our methodology and results.

Script vectors can be learned from one example. As our method does not rely on any finetuning, we hypothesize that few high quality activation pairs should already be able to approximate the script direction–and find one pair is sufficient. As such, we repeat our script confusion experiments using a single sample, where we select the first example that passes our filtering threshold described in Section 2.1. Figure 3 shows our results. We observe that one sample is already enough to induce script vectors, making our steering approach very sample-efficient. This potentially implies that cyrillization performance in the pseudo-label setup has potential for improvement, if one isolates the script vector on a single high quality example, as opposed to the current few-shot setup that potentially introduces noise with a larger number of samples.

Script is linearly separable across all decoder layers. We analyze the layer-wise presence of script-related information using Marks and Tegmark (2024)'s linear probe. Let $\mathbf{r}_l \in \mathbb{R}^D$ be the script vector for layer ℓ as defined in Section 2.1 and $\mathbf{x}_\ell \in \mathbb{R}^D$ an activation in the same layer. The probe is then $\sigma(\mathbf{r}_l^T \mathbf{x})$, where σ is the sigmoid function. Using the train split of FLEURS, we collect 50 activations per script for Serbian in Cyrillic and Latin script, and Mandarin in Traditional and Simplified script, yielding 100 activations in total for each language.[7] On the FLEURS test split, we perform the same collection procedure and subtract the mean of the training split activations as a preprocessing step to center each activation. The

[7]We use $\theta = 0.1$ here to ensure the cleanliness of the labels.

Language	Prediction (ours)	Deterministic	Original
Italian	Нелло специфико си состене ке е посибиле скоприре су уна персона ста ментэндо интерпретандо ле микроэспрессиони ин модо корpsтто.	нелло специфицо си состиене цхе е поссибиле сцоприре се уна персона ста ментендо интерпретандо ле мицроеспрессиони ин модо цорретто	nello specifico si sostiene che è possibile scoprire se una persona sta mentendo interpretando le microespressioni in modo corretto
Hindi	Вахан коу уси дин такребен 1200 ге МТ ке саме паре дургатана стал се дур лејаја геа.	ваахан ко усии дин такариибан 1200 гмт ке самаы пар дургхаттанаастхал се дуур ле йааыаа гаыаа	वाहअन को उसई दइन तअकरईबअन १२०० गमत कए सअमअय पअर दउरगहअथनासतहअल सए दऊर लए जाया गअया
Greek	То «Теолака Жинк» γήιϑκε στο τραγούδι των Бадзан.	тоте о лакка синнг егетхеке сто трагоуди тон батзан	τότε ο λάκκα σινγκ ηγήϑηκε στο τραγούδι των μπατζάν
Japanese	ワイルドカードを購入するとおтокуна бае га аримасу. Минами Африканоу омонаа коујн матауа Минами Африканоу сваутеу на коујну ни нужоу дъјуудщу.	уаирудокаадо уо коуныуу сурутоо ена бааи гааримасу минами афурика но омона коуен матаха минами афурика носубетено кокуритсукоуен ни ныууйоу декимасу	ワイルドカードを購入するとお得な場合があります 南アフリカの主な公園または南アフリカのすべての国立公園に入場できま
Korean	그해에 가장 규모가 큰 토너먼트 경기는 12월에 лас Канитасее фолоу генгианг изо ерлимьимида.	геу хаеыи гайанг гыумога кеун тонеомеонтеу гыеонгинеун 12уеоле расеу канитасеуыи полро гыеонггийангесео ыеолрибнида	그 해의 가장 규모가 큰 토너먼트 경기는 12월에 라스 카니타스의 폴로 경기장에서 열립니다

Table 2: Sample predictions of our cyrillization with Whisper large-v2. Italian observes the strongest performance, although other languages also observe sensible cyrillization in parts.

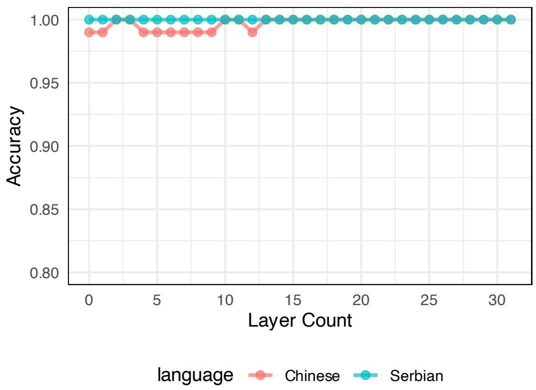

Figure 6: Script probing accuracy for Whisper large-v2 on Chinese and Serbian (Section 5).

probe then predicts the script (e.g., Simplified or Traditional Chinese) and measures accuracy.

Figure 6 shows our results. We observe that for both Serbian and Mandarin, the scripts produced are separable across all layers. One implication of this finding is that, as already in the first decoder layer it is predictable what script a given sample will decode to, we hypothesize one can rely on such probes to selectively apply either prompting or activation steering, instead of waiting until the full forward pass to produce the output in practical scenarios. As we observe in Section 4.1 that steering occasionally underperforms prompting for the more frequent script of a given language, this would potentially yield further performance gains. Another implication is that, since script is linearly separable in all layers, it is intuitive then to also apply steering in all layers, instead of selecting only one layer.

Script vectors induce sound-based transliterations. In Table 1, we observe that in comparison with the deterministic transliteration outputs, our prediction appears to reflect the actual pronunciation in many cases. We put in bold parts where the deterministic transliteration is induced by character mappings but does not actually reflect the pronunciation, where steering does. Taking as example the case of Korean, a deterministic transliteration based on character mapping renders the output *gajcugo issda* for the Korean tokens 갖추고 있다, deviating significantly from the pronunciation /kat͡ɕʰugoit̚ta/, which the steered transliteration *gachugo itda* arguably reflects better. Similarly, for the Greek example, modern Greek has gone through vowel mergers that cause η, ι, υ, ει, οι to be pronounced as /i/. The original pronunciation in earlier stages of Greek is preserved in the orthography, which the deterministic system relies on for mapping, but is not reflective of how they are actually pronounced in modern Greek (Holton et al., 2012). In contrast, our steering results consistently render these as *i*, in line with the pronuncia-

Script	Method	σ	Accuracy
Serbian Latin	baseline	0.0	0.81
	steer (tuned)	0.3	**0.86**
	steer (fixed)	1.0	0.0
Serbian Cyrillic	baseline	0.0	0.3
	steer (tuned)	0.5	0.4
	steer (fixed)	1.0	**0.57**
Traditional Chinese	baseline	0.0	0.73
	steer (tuned)	0.5	0.73
	steer (fixed)	1.0	**0.74**
Simplified Chinese	baseline	0.0	0.85
	steer (tuned)	0.5	0.85
	steer (fixed)	1.0	**0.86**

Table 3: Activation steering results for Whisper's encoder.

Script	Method	σ	Accuracy
Serbian Latin	baseline	0.0	0.18
	steer	0.3	**0.8**
Serbian Cyrillic	baseline	0.0	**0.74**
	steer	0.1	0.54
Traditional Chinese	baseline	0.0	**0.52**
	steer	0.1	**0.52**
Simplified Chinese	baseline	0.0	**0.75**
	steer	0.1	**0.75**

Table 4: Activation steering results on OWSM-CTC v4.

tion. Similar examples are also found in Hindi and Russian where orthographic cues mislead the deterministic system. Finally, in the case of Japanese, the character は is normally pronounced as *ha* as the deterministic system transliterates, but in the example above serves as a topic particle, in which case the correct pronunciation would be *wa* (Martin, 2003). Our steering correctly reflects this.

Can we steer only the encoder instead? Our experiments thus far focus on script representations in the decoder. However, as Whisper's language token is obtained by way of cross-attention to the encoder representations (Radford et al., 2023), it is possible that script information, similar to the case of language, exists in the encoder representations. We therefore empirically test this hypothesis by steering all layers and time steps of the encoder representations, repeating our experiments on script confusion. As prompting does not affect the encoder, we diverge from Section 2.1 here in collecting script activations without prompting the decoder, to ensure the generated script originates from encoder-based signals.[8] As we empirically find σ to reach the max value of 0.5 during hyperparameter tuning in some cases, we additionally also experiment with a larger σ of 1.0. Table 3 shows our results. We observe that for the case of script confusion, encoder steering indeed results in some control over script across all directions examined, although with noticeably weaker effect than decoder steering in Section 4.1. For all script directions that reached the max σ value of 0.5, using the larger σ of 1.0 all resulted in improved performance. For Serbian romanization, as it is the only case in which σ had an optimal σ below 0.5, the use of a 1.0 σ resulted in no text being generated. These results imply script information may indeed be represented in Whisper's encoder to some extent.

Does script steering work on other ASR model architectures? Our main experiments focus on the encoder-decoder architecture due to its widespread usage, but equally important in speech model architectures are models trained on a Connectionist Temporal Classification (CTC) loss. We therefore perform our steering experiments on a conventional CTC model — OWSM-CTC v4 (Peng et al., 2025b) — to observe whether our findings hold for architectures beyond the encoder-decoder. Table 4 shows our results for the script confusion setup. We observe that activation steering results on CTC-based models are more imbalanced compared to the Whisper-based results. For script confusion in Chinese, performance remains mostly unaffected under the current setup. For Serbian, romanization observes a substantial performance increase from 18% to 80%, whereas cyrilization (the dominant script) observes a performance drop from 74% to 54% after steering compared to the baseline. The observed success in the case of romanization implies the linear representation of script may partially generalize also to CTC-based models, although potentially to a weaker extent than in Whisper. We leave a more thorough investigation of script representation across different model architectures to future work.

Why are some named entities transcribed in English? In Table 1, we observe that some named entities such as *victoria harbour* and *Hong Kong*

[8]As we cannot prompt the same audio to obtain different scripts, we average instead over non-paired script activations in our encoder-based experiments, where the filtering is done per script individually.

are written directly in their English forms, as opposed to being transcribed phonetically the way they are pronounced in the original language (*bikutoria haabaa* and *hon kon* in Japanese). We also observe a related phenomenon where named entities tend to be the most easily steerable portions of a given transcription when applying smaller sigma values. We hypothesize this phenomenon to be due to some named entities being cross-lingually similar to English in pronunciation, such that they share stronger cross-script alignment. As the majority of Whisper's training data is in English (Radford et al., 2023), it is likely that this strongly affects the representation of such named entities to also be in English for Latin script languages. This is further evidenced by terms such as *Internet* and *communication* in the Korean transliteration in Table 1, which are loanwords in Korean that are phonetically similar to their English sources. We hypothesize that using data with many named entities or cross-lingually similar sounding words can ease the process of extracting script vectors.

6 Related Work

Script confusion There is significant work on the related problem of *language confusion* wherein language models (Marchisio et al., 2024) or code-switched ASR (Liu et al., 2023) struggles to output the correct language. In contrast, we focus on script confusion in speech foundation models, which is the problem of not just outputting the correct language but also the correct script.

Omnilingual et al. (2025) tackled script confusion in multilingual ASR by modifying the language token to include the script. This approach may require retraining or finetuning for models already trained without script tokens like Whisper. Furthermore, as script is hard-coded in the language tokens, it is not obvious whether transliteration in unseen directions can be induced. In contrast, we provide an inference-time, backpropagation-free mechanism for script control that allows transliteration even in unconventional language-script pairs.

Transliteration in NLP and ASR Work on transliteration has primarily focused on the text domain for use cases such as Latin script keyboards. This involves training task-specific models to perform transliteration (Ryskina et al., 2020; Roark et al., 2020) and finetuning LLMs (Purkayastha et al., 2023; Kirov et al., 2024). Within multilingual NLP and ASR, romanization (transliteration into Latin scripts) has provided a crosslingual phonetic space, enabling "zero-shot" adaptation to new languages (Liu et al., 2025; Jaavid et al., 2024; Saji et al., 2025; Jung et al., 2025; Lee et al., 2025; Zhu et al., 2024; Yan et al., 2023; Nigatu and Aldarmaki, 2025). None has explored activation steering for transliteration.

Activation steering Activation steering is a technique for controlling the outputs of a language model at inference time by extracting steering vectors and adding them to the language model's activations (Turner et al., 2023b). This method can control model outputs' topic, sentiment, toxicity (Turner et al., 2023b), truthfulness (Li et al., 2024; Marks and Tegmark, 2024; Ravfogel et al., 2025) and refusal of safety-related queries (Wang et al., 2025b; Marshall et al., 2024). This works because many of the aforementioned concepts can be linearly represented in an embedding space (Park et al., 2024; Modell et al., 2025) and thus controlled with a single vector. Such methods can be used in speech language models as well (Lin et al., 2025a; Xie et al., 2025; Lin et al., 2025b). No prior work on activation steering, however, has focused on test-time control of subword properties like sounds, much less orthography (script).

7 Conclusion and Future Work

In this work, we show that script information is linearly represented in the activation space of speech foundation models. Such directions enable control over the script of the transcription output and generalize cross-lingually, enabling transliteration in novel directions. One implication of our work is that speech foundation models may already acquire cross-script alignment even without an explicit transliteration step, as recent work has shown transliterating all scripts to a unified script to help speech recognition performance (Lee et al., 2025). Future work may therefore look into the training dynamics of speech foundation models to investigate how such an alignment is acquired throughout training.

Limitations

We acknowledge that there exist multiple possible transliterations for a given word: ones that lean more semantic (Li et al., 2007) and ones that are more phonetically faithful, among other criteria (Demirsahin et al., 2022). In this work, we focused only on phonetically faithful transliterations. Furthermore, while we tested various models in preliminary experiments, our experiments focus mostly on the Whisper series due to their widespread usage and influence (Ma et al., 2025; Peng et al., 2025a). Future work can analyze the training dynamics of open-data speech foundation models (Peng et al., 2025a; Li et al., 2025) and encoder-only models as well, where the additional task of phoneme recognition in Li et al. (2025) may potentially induce even stronger cross-script alignment. Finally, in this work we study only the settings of X $\rightarrow$ Cyrillic and X $\rightarrow$ Latin transliteration, along with transliteration between traditional and simplified Chinese. Future work can expand this investigation to more transliteration directions.

Ethical Considerations

All data used in this work are publicly available. While our experiments focus on understanding the representation of script information in speech foundation models, we acknowledge that activation steering and probing may be misapplied to either generate or detect content associated with certain demographic information. Our study is intended towards the goal of interpretable speech recognition, and we encourage its use towards building systems that embrace empirical linguistic diversity and the self-expression of its speakers.

Use of AI Assistants

The authors acknowledge the usage of ChatGPT as an assistant tool in part of the source code's development, in assisting the creation of figures, and in enhancing the coherence of parts of the manuscript.

Acknowledgements

We would like to thank the members of the MaiNLP lab and the anonymous reviewers for their invaluable feedback. Specifically, we thank Silvia Casola and Rob van der Goot for comments on an earlier draft. We would also like to thank Chin-Jou Li for valuable advice in refining this manuscript. Lastly, we recognize the support for Ryan Soh-Eun Shim and Barbara Plank through the ERC Consolidator Grant 101043235.

References

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

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

Isin Demirsahin, Cibu Johny, Alexander Gutkin, and Brian Roark. 2022. Criteria for useful automatic Romanization in South Asian languages. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 6662–6673, Marseille, France. European Language Resources Association.

Ulf Hermjakob, Jonathan May, and Kevin Knight. 2018. Out-of-the-box universal Romanization tool uroman. In *Proceedings of ACL 2018, System Demonstrations*, pages 13–18, Melbourne, Australia. Association for Computational Linguistics.

David Holton, Peter Mackridge, Irene Philippaki-Warburton, and Vassilios Spyropoulos. 2012. *Greek: A comprehensive grammar of the modern language*. Routledge.

J Jaavid, Raj Dabre, M Aswanth, Jay Gala, Thanmay Jayakumar, Ratish Puduppully, and Anoop Kunchukuttan. 2024. Romansetu: Efficiently unlocking multilingual capabilities of large language models via romanization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15593–15615.

Srđan Mladenov Jovanović. 2018. Assertive discourse and folk linguistics: Serbian nationalist discourse about the cyrillic script in the 21st century. 17(4):611–631.

Haeji Jung, Jinju Kim, Kyungjin Kim, Youjeong Roh, and David R Mortensen. 2025. Happiness is sharing a vocabulary: A study of transliteration methods. *arXiv preprint arXiv:2510.10827*.

Christo Kirov, Cibu Johny, Anna Katanova, Alexander Gutkin, and Brian Roark. 2024. Context-aware transliteration of romanized south asian languages. *Computational Linguistics*, 50(2):475–534.

Carbo Kuo and 1 others. 2024. Open Chinese convert (opencc).

Georges Labrèche. 2025. Cyrtranslit. A Python package for bi-directional transliteration of Cyrillic script to Latin script and vice versa. Supports transliteration for Belarusian, Bulgarian, Greek, Montenegrin, Macedonian, Mongolian, Russian, Serbian, Tajik, and Ukrainian.

Sangmin Lee, Woojin Chung, and Hong-Goo Kang. 2025. Lama-ut: Language agnostic multilingual asr through orthography unification and language-specific transliteration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24393–24401.

Chin-Jou Li, Kalvin Chang, Shikhar Bharadwaj, Eunjung Yeo, Kwanghee Choi, Jian Zhu, David Mortensen, and Shinji Watanabe. 2025. Powsm: A phonetic open whisper-style speech foundation model. *Preprint*, arXiv:2510.24992.

Haizhou Li, Khe Chai Sim, Jin-Shea Kuo, and Minghui Dong. 2007. Semantic transliteration of personal names. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 120–127, Prague, Czech Republic. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Preprint*, arXiv:2306.03341.

Tsung-En Lin, Kuan-Yi Lee, and Hung-Yi Lee. 2025a. Adaptive vector steering: A training-free, layer-wise intervention for hallucination mitigation in large audio and multimodal models. *arXiv preprint arXiv:2510.12851*.

Weilin Lin, Jianze Li, Hui Xiong, and Li Liu. 2025b. Sarsteer: Safeguarding large audio language models via safe-ablated refusal steering. *arXiv preprint arXiv:2510.17633*.

Hexin Liu, Haihua Xu, Leibny Paola Garcia, Andy WH Khong, Yi He, and Sanjeev Khudanpur. 2023. Reducing language confusion for code-switching speech recognition with token-level language diarization. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Yihong Liu, Mingyang Wang, Amir Hossein Kargaran, Ayyoob ImaniGooghari, Orgest Xhelili, Haotian Ye, Chunlan Ma, François Yvon, and Hinrich Schütze. 2025. How transliterations improve crosslingual alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2417–2433.

Rao Ma, Mengjie Qian, Yassir Fathullah, Siyuan Tang, Mark Gales, and Kate Knill. 2025. Cross-lingual transfer learning for speech translation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 33–43, Albuquerque, New Mexico. Association for Computational Linguistics.

Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677, Miami, Florida, USA. Association for Computational Linguistics.

Samuel Marks and Max Tegmark. 2024. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. In *First Conference on Language Modeling*.

Thomas Marshall, Adam Scherlis, and Nora Belrose. 2024. Refusal in llms is an affine function. *arXiv preprint arXiv:2411.09003*.

Samuel E Martin. 2003. *A reference grammar of Japanese*. University of Hawaii Press.

Hiroshi Miura and 1 others. 2024. pykakasi: Kana kanji simple inversion library.

Alexander Modell, Patrick Rubin-Delanchy, and Nick Whiteley. 2025. The origins of representation manifolds in large language models. *arXiv preprint arXiv:2505.18235*.

Hellina Hailu Nigatu and Hanan Aldarmaki. 2025. Exploring transliteration-based zero-shot transfer for amharic asr. In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 64–73.

ASR Omnilingual, Gil Keren, Artyom Kozhevnikov, Yen Meng, Christophe Ropers, Matthew Setzler, Skyler Wang, Ife Adebara, Michael Auli, Can Balioglu, and 1 others. 2025. Omnilingual asr: Open-source multilingual speech recognition for 1600+ languages. *arXiv preprint arXiv:2511.09690*.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Yifan Peng, Shakeel Muhammad, Yui Sudo, William Chen, Jinchuan Tian, Chyi-Jiunn Lin, and Shinji Watanabe. 2025a. Owsm v4: Improving open whisper-style speech models via data scaling and cleaning. *Preprint*, arXiv:2506.00338.

Yifan Peng, Muhammad Shakeel, Yui Sudo, William Chen, Jinchuan Tian, Chyi-Jiunn Lin, and Shinji Watanabe. 2025b. OWSM v4: Improving Open Whisper-Style Speech Models via Data Scaling and Cleaning. In *Interspeech 2025*, pages 2225–2229.

Sukannya Purkayastha, Sebastian Ruder, Jonas Pfeiffer, Iryna Gurevych, and Ivan Vulić. 2023. Romanization-based large-scale adaptation of multilingual language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7996–8005, Singapore. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Shauli Ravfogel, Gilad Yehudai, Tal Linzen, Joan Bruna, and Alberto Bietti. 2025. Emergence of linear truth encodings in language models. *Advances in Neural Information Processing Systems*, 38.

Brian Roark, Lawrence Wolf-Sonkin, Christo Kirov, Sabrina J Mielke, Cibu Johny, Isin Demirsahin, and Keith Hall. 2020. Processing south asian languages written in the latin script: the dakshina dataset. *arXiv preprint arXiv:2007.01176*.

Maria Ryskina, Matthew R. Gormley, and Taylor Berg-Kirkpatrick. 2020. Phonetic and visual priors for decipherment of informal Romanization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8308–8319, Online. Association for Computational Linguistics.

Alan Saji, Jaavid Aktar Husain, Thanmay Jayakumar, Raj Dabre, Anoop Kunchukuttan, and Ratish Puduppully. 2025. Romanlens: The role of latent romanization in multilinguality in llms. *arXiv preprint arXiv:2502.07424*.

Hsi-Yao Su and Chen-Cheng Chun. 2021. Chineseness, Taiwaneseness, and the traditional and simplified Chinese scripts: Tourism, identity, and linguistic commodification. *Language & Communication*, 77:35–45.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023a. Activation addition: Steering language models without optimization. *arXiv e-prints*, pages arXiv–2308.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023b. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2025a. Bridging the language gaps in large language models with inference-time cross-lingual intervention. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5418–5433, Vienna, Austria. Association for Computational Linguistics.

Xinpeng Wang, Mingyang Wang, Yihong Liu, Hinrich Schütze, and Barbara Plank. 2025b. Refusal direction is universal across safety-aligned languages. *arXiv preprint arXiv:2505.17306*.

Tianxin Xie, Shan Yang, Chenxing Li, Dong Yu, and Li Liu. 2025. Emosteer-tts: Fine-grained and training-free emotion-controllable text-to-speech via activation steering. *arXiv preprint arXiv:2508.03543*.

Brian Yan, Matthew Wiesner, Ondřej Klejch, Preethi Jyothi, and Shinji Watanabe. 2023. Towards zero-shot code-switched speech recognition. In *ICASSP 2023-2023 IEEE International Conference On Acoustics, Speech And Signal Processing (ICASSP)*, pages 1–5. IEEE.

Han Zhu, Gaofeng Cheng, Qingwei Zhao, and Pengyuan Zhang. 2024. Transliterated zero-shot domain adaptation for automatic speech recognition. *arXiv preprint arXiv:2412.11185*.

A Appendix

Script	Prompt
Simplified Chinese	这是一句普通话
Traditional Chinese	這是一句普通話
Serbian (Latin)	Ovo je srpska rečenica
Serbian (Cyrillic)	Ово је српска реченица

Table 5: Script confusion prompts. All prompts have the meaning of "this is a sentence in X", where X is the actual language.

Language	**Prompt**
Russian	eto russkoye predlozheniye
Greek	auti einai mia elliniki protasi
Hindi	yah ek hindi vaakya hai
Korean	igeoseun hangugeo munjangibnida
Japanese	kore wa nihongo no bun desu

Table 6: Romanization prompts. All prompts have the meaning of "this is a sentence in X", where X is the actual language.

Language	**Prompt**
Italian	куэста э уна фразе итальяна
Greek	афти инэ мия эллиники протаси
Hindi	йе эк хинди вакья хэ
Korean	игосын хангуго мунджанимнида
Japanese	корэ ва нихонго но бун десу

Table 7: Cyrillization prompts. All prompts have the meaning of "this is a sentence in X", where X is the actual language.

Script	Method	Model Size						
		tiny	base	small	medium	large	v2	v3
sr-latn	no-prompt	**0.66**	**0.81**	**0.9**	0.63	0.93	0.81	0.93
	prompt	0.64	**0.81**	**0.9**	**0.93**	0.95	0.95	0.95
	steer	0.64	0.8	0.89	**0.93**	**0.96**	**0.96**	**0.96**
sr–cyrl	no-prompt	0.1	0.12	0.14	0.44	0.18	0.3	0.18
	prompt	0.12	0.21	0.52	**0.91**	0.82	0.77	0.82
	steer	**0.54**	**0.72**	**0.83**	0.84	**0.93**	**0.94**	**0.93**
zh–sim	no-prompt	0.6	0.67	0.79	0.87	**0.93**	0.85	**0.93**
	prompt	**0.7**	**0.78**	**0.87**	**0.91**	**0.93**	**0.93**	**0.93**
	steer	0.69	**0.78**	**0.87**	**0.91**	**0.93**	**0.93**	**0.93**
zh–trad	no-prompt	0.63	0.69	0.69	0.69	0.65	0.73	0.65
	prompt	**0.71**	**0.79**	**0.86**	**0.9**	0.79	**0.91**	0.79
	steer	**0.71**	**0.79**	**0.86**	0.89	**0.89**	**0.91**	**0.89**

Table 8: Script confusion across Whisper models.

Script	Method	Model Size						
		tiny	base	small	medium	large	v2	v3
sr-latn	no-prompt	**0.66**	**0.81**	**0.9**	0.63	0.93	0.81	0.93
	prompt	0.64	**0.81**	**0.9**	**0.93**	0.95	0.95	0.95
	steer	0.63	0.8	0.89	**0.93**	**0.96**	**0.96**	**0.96**
sr–cyrl	no-prompt	0.1	0.12	0.14	0.44	0.18	0.3	0.18
	prompt	0.12	0.21	0.52	**0.91**	0.82	0.77	0.82
	steer	**0.5**	**0.71**	**0.83**	0.84	**0.93**	**0.94**	**0.93**
zh–sim	no-prompt	0.6	0.67	0.79	0.87	**0.93**	0.85	**0.93**
	prompt	**0.7**	**0.78**	**0.87**	**0.91**	**0.93**	**0.93**	**0.93**
	steer	**0.7**	**0.78**	**0.87**	**0.91**	**0.93**	0.91	**0.93**
zh–trad	no-prompt	0.63	0.69	0.69	0.69	0.65	0.73	0.65
	prompt	**0.71**	**0.79**	**0.86**	**0.9**	0.79	0.91	0.79
	steer	**0.71**	**0.79**	**0.86**	**0.9**	**0.89**	**0.93**	**0.89**

Table 9: One-shot script steering results across Whisper models.

Script	Method	Romanization	Cyrillization
Hindi	no-prompt	0.01	0.00
	prompt	0.02	0.01
	zero-shot	0.69	**0.19**
	pseudo-label	**0.71**	0.17
Greek	no-prompt	0.02	0.01
	prompt	0.02	0.01
	zero-shot	0.53	**0.15**
	pseudo-label	**0.73**	0.09
Japanese	no-prompt	0.01	0.01
	prompt	0.01	0.01
	zero-shot	0.28	**0.05**
	pseudo-label	**0.62**	0.04
Korean	no-prompt	0.01	0.01
	prompt	0.01	0.01
	zero-shot	0.2	0.06
	pseudo-label	**0.51**	**0.17**
Russian	no-prompt	0.01	*NA*
	prompt	0.01	*NA*
	zero-shot	0.67	*NA*
	pseudo-label	**0.73**	*NA*
Italian	no-prompt	*NA*	0.01
	prompt	*NA*	0.01
	zero-shot	*NA*	**0.43**
	pseudo-label	*NA*	0.15

Table 10: Romanization and cyrillization performance for Whisper large-v2.

Language	Train	Validation	Test
Mandarin	3246	409	945
Serbian	2944	290	700
Russian	2562	356	775
Italian	3030	391	865
Hindi	2120	239	418
Greek	3215	271	650
Japanese	2292	266	650
Korean	2307	226	382

Table 11: Statistics of the train, validation, and test data in the FLEURS (Conneau et al., 2023) dataset we employ.

Script	Method	Romanization	Cyrillization
Hindi	zero-shot	0.2	0.3
	pseudo-label	0.1	0.1
Greek	zero-shot	0.3	0.3
	pseudo-label	0.1	0.1
Japanese	zero-shot	0.3	0.3
	pseudo-label	0.2	0.5
Korean	zero-shot	0.3	0.3
	pseudo-label	0.1	0.5
Russian	zero-shot	0.2	*NA*
	pseudo-label	0.1	*NA*
Italian	zero-shot	*NA*	0.2
	pseudo-label	*NA*	0.1

Table 12: Best sigmas for transliteration.