Analyzing Multilingual Parametric Knowledge Retrieval with Sparse Autoencoders

Marc E. Canby **Ikhyun Cho** **Julia Hockenmaier**
The Grainger College of Engineering
Siebel School of Computing and Data Science
University of Illinois Urbana-Champaign
Urbana, IL, USA
{marcec2, ihcho2, juliahmr}@illinois.edu

Abstract

Multilingual language models can answer factual questions across languages, but their answers are not always consistent across languages, and the internal pathways used to retrieve this knowledge remain unclear. A model may rely on language-specific shortcuts, shared multilingual representations, or another language as an internal pivot. We investigate these possibilities in Llama-3.1-8B and Gemma-2-9B with sparse autoencoders (SAEs), identifying, analyzing, and causally intervening on language-associated features during factual recall. On CUS-QA (Libovický et al., 2026), a parallel QA dataset spanning Czech, Slovak, Ukrainian, and English, we identify SAE features associated with particular subsets of languages. We then decompose each question's activations into these subsets to measure reliance on input-language, multilingual, or pivot-like representations across layers. Across both models, factual recall is heterogeneous: different input languages and questions exhibit different language-associated pathways. Steering substantially changes cross-lingual answer alignment and sometimes increases correctness in Llama, while it produces only small changes in Gemma. These results show that multilingual factual recall is mediated by identifiable SAE features that can be used to diagnose and control how models access knowledge across languages.[1]

1 Introduction

Language models can store and retrieve factual knowledge from their parameters (Petroni et al., 2019; Jiang et al., 2020b; Geva et al., 2021). In multilingual models, this ability extends across languages, but unevenly: factual recall varies substantially with the query language (Jiang et al., 2020a; Kassner et al., 2021; Xu et al., 2023). This raises a mechanistic question: when a model answers the same factual question in different languages, does it access a shared multilingual representation, or rely on different language-associated pathways? Output-level evaluations can measure correctness or cross-lingual consistency (Qi et al., 2023), but they cannot determine whether aligned outputs arise from shared internal computation or from separate computations that converge on the same answer. Understanding multilingual factual recall therefore requires connecting cross-lingual behavior to internal representations.

	Unsteered				Steered toward SK+UK			
	EN	CS	SK	UK	EN	CS	SK	UK
CS	26.5	26.5	21.2	9.7	**28.3**	**28.3**	20.4	8.8
SK	18.1	12.4	14.3	7.6	**23.8**	15.2	21.0	7.6
UK	29.3	18.3	13.4	22.0	**31.7**	19.5	14.6	28.0
All	24.3	19.3	16.7	12.3	**27.7**	21.3	19.0	13.7

Table 1: **Performance of Llama 3.1-8B on CUS-QA dataset.** Accuracy (%) of Llama 3.1-8B when asked country-specific questions (rows) in specific input languages (columns) on 300 test examples of the CUS-QA dataset (Libovický et al., 2026). English performs best on all countries' questions, with the "home" language performing second best; clearly multilingual fact retrieval is uneven. Steering Llama's sparse autoencoder activations toward features associated with both SK and UK provide an accuracy gain for all input languages.

We study this question using CUS-QA (Libovický et al., 2026), a parallel QA dataset spanning Czech, Slovak, Ukrainian, and English. Because each fact is queried across languages, CUS-QA lets us compare behavior while holding the target knowledge fixed. Unlike much prior multilingual factual probing, which uses cloze-style or otherwise constrained formats, CUS-QA requires free-form answers. This makes the task closer to realistic model use and lets us study both correctness and semantic alignment among generated answers. As Table 1 shows, factual recall is highly uneven

[1] Our code is provided at https://github.com/marccanby/multi-mechinterp.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3250–3264
November 6-10, 2026 ©2026 Association for Computational Linguistics

across input languages: on Llama 3.1-8B, English performs best overall, while each country's "home" language is typically second-best.

We frame this behavior with three hypotheses. Under the *multilingual representation* hypothesis, factual recall is mediated by features shared across languages. Under the *shortcut* hypothesis, the model relies primarily on features associated with the prompt language. Under the *pivot* hypothesis, the model accesses a fact through features associated with another language or language subset, without necessarily performing explicit translation.

We test these hypotheses using sparse autoencoders (SAEs), which decompose activations into sparse feature activations (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024). First, we identify SAE features whose activation profiles correspond to *language subsets*, from single-language features to features shared by groups such as CS+SK, SK+UK, or CS+SK+UK. This subset labeling is central: a single-language taxonomy would miss features that may act as cross-lingual bridges. Second, we use these features to decompose each question's activations, yielding a layerwise measure of which language-associated representations the model relies on. Third, we steer the model toward selected feature subsets, testing how this changes downstream answers.

Our choice of SAEs is motivated primarily by interpretability: their sparse, individually identifiable features let us label components by language association and measure their contributions across layers and questions (Cunningham et al., 2023; Gao et al., 2024; Lieberum et al., 2024). Dense language means or LDA-style directions can support steering, but do not provide the same feature-level decomposition. We use steering to test whether the identified features influence behavior, rather than to establish superiority over other steering methods.

Our results show that multilingual factual recall is heterogeneous. Some questions remain anchored to the input language, consistent with shortcut-like processing; others shift toward non-input language profiles; and aligned answers tend to have more similar language-reliance profiles, especially in middle layers. Moreover, steering toward selected subsets changes both internal reliance and behavior. In particular, steering toward SK+UK on Llama-3.1-8B improves accuracy across all input languages in our test set, and subset steering can substantially increase cross-lingual answer alignment. Together, these findings suggest that multilingual factual recall is mediated by identifiable SAE features, and that subset-level features provide a useful interface for interpreting and controlling how models retrieve facts across languages.

2 Related Work

2.1 Parametric Fact Retrieval

Prior work has studied language models as stores of parametric factual knowledge. The LAMA probe introduced cloze-style factual queries for testing whether pretrained language models encode relational knowledge without task-specific supervision (Petroni et al., 2019). Subsequent work showed that factual recall is highly sensitive to how knowledge is queried, including prompt wording, paraphrase choice, relation type, and answer frequency (Jiang et al., 2020b; Elazar et al., 2021; Kassner et al., 2021). This has motivated work on localizing and editing factual associations, including ROME and MEMIT (Meng et al., 2023b,a), and studies on how subject and relation information flow through the model during inference (Geva et al., 2023).

Multilingual factual retrieval adds another layer of complexity: the same fact may be accessible in some languages but not others. X-FACTR and mLAMA extend factual probing to multilingual settings, showing that factual recall varies substantially across languages and that evaluation is complicated by multi-token answers, morphology, and language-specific answer forms (Jiang et al., 2020a; Kassner et al., 2021). More recent work studies cross-lingual transfer and consistency, asking whether models retrieve compatible answers when the same fact is queried in different languages (Xu et al., 2023; Qi et al., 2023). Our work builds on this line of research, but moves from output-level evaluation to internal mechanism: we ask which language-associated SAE features support free-form factual answers, and whether intervening on those features changes retrieval behavior.

2.2 Mechanistic Analysis of Knowledge and Language

Mechanistic interpretability work has investigated how factual information is represented and retrieved inside transformers. Knowledge-neuron methods identify neurons associated with specific relational facts (Dai et al., 2022), while work on feed-forward layers interprets transformer MLPs as key-value memories that map input patterns to output distributions (Geva et al., 2021). Causal tracing

and model-editing work further suggest that factual associations can be localized to particular mid-layer computations and directly modified (Meng et al., 2023a). Other tools, such as the logit lens and tuned lens, decode intermediate representations to study how predictions evolve layer by layer (Nguyen, 2024; Belrose et al., 2025). Together, these approaches support the view that factual recall can be analyzed as an internal computation rather than only as an input-output behavior.

Recent work has applied mechanistic tools to multilingual processing. Wendler et al. (2024) ask whether multilingual transformers use English as an internal pivot language, finding evidence that non-English prompts can pass through intermediate representations closer to English before returning to the input language's output space. Other work identifies language-specific neurons and shows that manipulating them can steer the output language of multilingual models (Tang et al., 2024). Andrylie et al. (2026) uses sparse autoencoders to find language-specific features. Our work is closely related, but focuses on factual recall rather than output-language control. Also, our method enables us to identify and intervene on features associated not only with single languages, but also with language subsets such as CS+SK or SK+UK.

2.3 Sparse Autoencoders

Sparse autoencoders (SAEs) have become a popular way to interpret model internals (Bricken et al., 2023; Cunningham et al., 2023; Gao et al., 2024). For a fixed transformer layer l, let $h^{(l)} \in \mathbb{R}^d$ denote the residual stream activation at a single token position. An SAE learns a linear decomposition of $h^{(l)}$ into a sparse feature representation. Let $W_{\text{enc}}^{(l)} \in \mathbb{R}^{k \times d}$ and $W_{\text{dec}}^{(l)} \in \mathbb{R}^{d \times k}$ denote the encoder and decoder matrices, respectively. Feature activations are computed as

$$z^{(l)} = \text{ReLU}\Big(W_{\text{enc}}^{(l)} h^{(l)} + b\Big),$$

and the reconstruction is given by $\hat{h}^{(l)} = W_{\text{dec}}^{(l)} z^{(l)}$. The SAE is trained to minimize reconstruction error $\|h^{(l)} - \hat{h}^{(l)}\|_2^2$ subject to a sparsity constraint on $z^{(l)}$. After training, each coordinate of $z^{(l)}$ is interpreted as a feature capturing a pattern in the residual stream.

3 Methodology

3.1 Extracting Feature Activations for Multilingual Question Answering

Each prompt consists of an *instruction section* followed by a *question section*. For analysis, we extract feature activations only from tokens belonging to the question section. Let the question section of an input $x = q(L)$ consist of tokens $t_1, \ldots, t_T$. For each layer l and token position i, we compute the residual stream activation $h_i^{(l)}$ and obtain sparse feature activations

$$z_i^{(l)} = \text{ReLU}\Big(W_{\text{enc}}^{(l)} h_i^{(l)} + b\Big).$$

Let $I_f^{(l)}(x)$ be the set of tokens in layer l on which feature f has non-zero activation. To obtain a question-level representation, we average feature activations across this set:

$$a_f^{(l)}(x) = \frac{1}{|I_f^{(l)}(x)|} \sum_{i \in I_f^{(l)}(x)} z_{i,f}^{(l)}$$

This produces one activation value per feature per layer per question, which is used in subsequent multilingual reliance analyses. We restrict our analysis to question encoding activations; incorporating generation-time activations is left for future work.

3.2 Measuring a feature's language profile

Goal For each SAE feature f (at each layer), we aim to characterize the extent to which it is shared across languages versus specific to a single language. In particular, given a fixed set of languages $\mathcal{L} = \{\text{cs}, \text{sk}, \text{uk}, \text{en}\}$, we classify each feature into one of four high-level categories: (i) **None** (inactive / unclassified), if the feature does not activate sufficiently often or strongly to support reliable estimation; (ii) **ℓ-specific**, if its activation is concentrated in a single language $\ell \in \mathcal{L}$; (iii) **subset-shared**, if it is consistently active in a proper subset $S \subset \mathcal{L}$ with $2 \leq |S| < |\mathcal{L}|$; or (iv) **panlingual**, if it is broadly active across all languages in $\mathcal{L}$.

Method Fix a layer and let $a_f(x) \geq 0$ denote the activation of SAE feature f on an example x. For each language $\ell \in \mathcal{L}$, we compute the expected activation magnitude

$$\mu_{f,\ell} = \mathbb{E}[a_f(x) \,|\, \text{lang}(x) = \ell],$$

estimated in practice by the sample mean over all examples in language ℓ. We then normalize these

per-language activations into a language-mixture profile

$$p_{f,\ell} = \frac{\mu_{f,\ell}}{\sum_{\ell' \in \mathcal{L}} \mu_{f,\ell'} + \epsilon},$$

where ϵ is a small constant for numerical stability. Intuitively, $p_{f,\ell}$ represents the fraction of feature f's activation mass attributable to language ℓ.

To identify features that are effectively inactive in our setting, we define the total activation mass

$$m_f = \sum_{\ell \in \mathcal{L}} \mu_{f,\ell}.$$

Features with m_f below a small threshold $m_{\min}$ are excluded from subsequent multilinguality analyses.

Subset membership from soft activations We seek a measure of subset structure that captures which languages primarily account for a feature's activation. Concretely, for any subset $S \subseteq \mathcal{L}$, we define a similarity score $\text{score}_f(S) \in [0, 1]$ that quantifies how closely p_f matches the idealized pattern of an S-type feature. We require such a score to satisfy the following properties: (i) it attains its maximum when p_f matches the ideal subset distribution (uniform over S and zero elsewhere); (ii) it penalizes probability mass assigned to languages outside S; and (iii) it penalizes imbalance within S, favoring distributions that are evenly spread across the languages in S.

First, for any nonempty subset $S \subseteq \mathcal{L}$, define the ideal distribution

$$u_S(\ell) = \begin{cases} \frac{1}{|S|} & \ell \in S \\ 0 & \ell \notin S \end{cases}$$

We measure similarity using the total variation (TV) distance between the observed distribution and the subset prototype:

$$\text{TV}(p_f, u_S) = \frac{1}{2} \sum_{\ell \in \mathcal{L}} \left| p_{f,\ell} - u_S(\ell) \right|$$

We then define the subset similarity score as

$$\text{score}_f^{\text{TV}}(S) = 1 - \text{TV}(p_f, u_S)$$

This score lies in $[0, 1]$ and admits a direct geometric interpretation: the total variation distance corresponds to the amount of probability mass that must be redistributed to transform p_f into the ideal subset distribution u_S. Consequently, the score penalizes both leakage outside the subset and imbalance within the subset in a transparent and easily interpretable manner.

Assigning hard subset labels Here, we wish to assign a hard subset label $S_f \subseteq \mathcal{L}$ to each feature. Given a subset similarity score $\text{score}_f^{\text{TV}}(S)$, we assign a label based on three criteria: selectivity, overall fit, and distinctiveness relative to competing subsets. Features that don't fit all criteria are labeled `None`.

First, we only consider features that are sufficiently selective: we discard features that fire on more than an α fraction of all sentences (regardless of language). In practice, this removes broad "spray" features that activate nearly everywhere and would otherwise be misleadingly assigned to the pan-lingual subset $\mathcal{L}$, not because they reflect shared multilingual structure, but simply because they are active across almost all inputs. We also remove features that have total activation less than 0.01.

Next, we only consider labeling feature f with subset $S_f^\star$, where

$$S_f^\star = \arg\max_{S \subseteq \mathcal{L}} \text{score}_f(S)$$

denotes the subset with highest similarity score. We require that this subset provides a sufficiently good fit: $\text{score}_f^{\text{TV}}(S_f^\star) \geq \tau$. Finally, to ensure that the assignment is unambiguous, we compare $S_f^\star$ against all competing subsets; this is necessary because the subsets are not arbitrary classification labels since they often share the same language(s). Thus, for each $T \neq S_f^\star$, we define a normalized margin

$$h_f(S_f^\star, T) = \frac{\text{score}_f(S_f^\star) - \text{score}_f(T)}{1 - \text{score}(u_{S_f^\star}, u_T)},$$

where u_S denotes the prototype distribution for subset S, and $\text{score}(u_{S_f^\star}, u_T)$ measures the similarity between the corresponding subset prototypes.[2] We then define the overall margin as the worst-case normalized separation:

$$h_f(S_f^\star) = \min_{T \neq S_f^\star} h_f(S_f^\star, T).$$

We assign the hard subset label $S_f^\star$ if $h_f(S_f^\star) \geq \delta$ and `None` otherwise.

Figure 1 shows the number of features at each layer of Llama labeled with each label.[3] We find

[2] The normalization ensures that the margin accounts for intrinsic overlap between subsets, measuring the observed separation relative to the maximum achievable separation from each competing subset.

[3] Figure 5 in Appendix B shows the corresponding figure for Gemma.

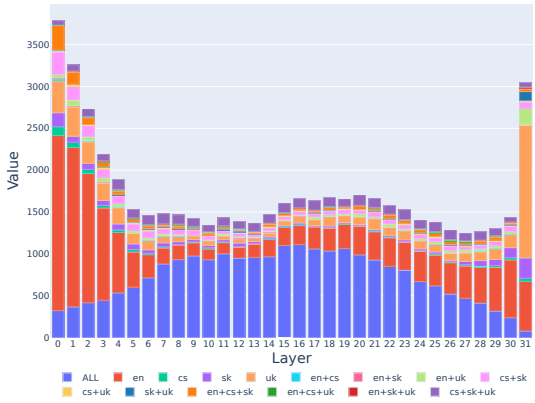


Figure 1: **Number of features in Llama with each subset label.** For each layer, shows the number of features labeled $S \subseteq \mathcal{L}$, where ALL represents the full set of languages $\mathcal{L}$.

that ALL features are most prevalent, especially in middle layers, and that EN features are common at the beginning and end layers. It is also notable that a Slavic feature CS+SK+UK is identified clearly in all layers. The UK feature is especially common in the last layer, likely due to the need for special unembedding representations of Ukrainian, which has different alphabet than the other languages.

3.3 Measuring a Question's Reliance on Language-Specific and Multilingual Information

We now quantify, for a given question instance $x = q(L_x)$, the extent to which its internal representation relies on language-specific versus multilingual information. All quantities are defined separately at each layer.

Method 1: Soft Activations Let $a_f^{(l)}(x) \geq 0$ denote the activation of feature f at layer l on input x, and define the total activation mass

$$A^{(l)}(x) = \sum_f a_f^{(l)}(x)$$

We assume the language-mixture profiles $p_{f,\ell}$ defined in §3.2 are available for each feature. We attribute the activation of each feature proportionally to its language-mixture profile. For each language $\ell \in \mathcal{L}$, define the attributed activation mass

$$M_\ell^{(l)}(x) = \sum_f a_f^{(l)}(x)\, p_{f,\ell}$$

Normalizing gives a language-reliance distribution:

$$R_\ell^{(l)}(x) = \frac{M_\ell^{(l)}(x)}{A^{(l)}(x)}$$

This represents the proportion of the representation at layer l attributable to each language. This method captures the aggregate contribution of each language to the representation, but does not distinguish whether that contribution arises from language-specific features or from features shared across multiple languages.

Method 2: Hard Feature Labels We complement the soft attribution above with a feature-level decomposition based on hard subset labels. Each feature f is assigned a subset $S_f \subseteq \mathcal{L}$ (or None if no unambiguous assignment is possible), as described in §3.2. We then attribute the activation of each feature entirely to its assigned subset.

For each subset $S \subseteq \mathcal{L}$, define the attributed activation mass

$$M_S^{(l)}(x) = \sum_{f:S_f=S} a_f^{(l)}(x)$$

Since not all features admit a reliable subset label, we restrict attention to the labeled features. Let

$$A_{\text{lab}}^{(l)}(x) = \sum_{f:S_f\neq\text{None}} a_f^{(l)}(x)$$

denote the total activation mass of labeled features. We normalize to obtain a reliance distribution:

$$R_S^{(l)}(x) = \frac{M_S^{(l)}(x)}{A_{\text{lab}}^{(l)}(x)}$$

The distribution over subsets characterizes how the interpretable portion of the representation at layer l decomposes across language-specific and multilingual feature types.

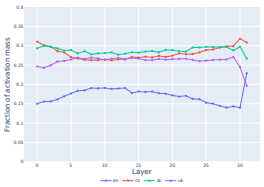
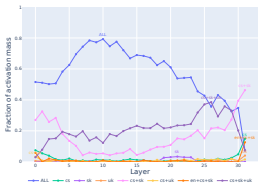
3.4 Steering: Enhancing Multilinguality

We investigate whether multilingual behavior can be causally influenced by intervening on SAE features associated with particular language subsets. All steering interventions are applied at a single transformer layer l.

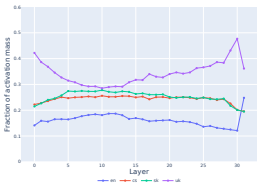
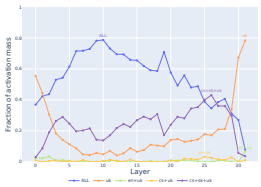
Subset-conditioned feature selection Recall that each SAE feature f is assigned a hard subset label $S_f \subseteq \mathcal{L}$. When we wish to steer toward a target subset $S^\star$, we first collect all features f such that $S_f = S^\star$. Within this subset, features are ranked according to their total activation mass $m_f^{(l)}$, and we select the top-K highest-mass features:

$$F_{S^\star}^{(l)} = \text{TopK}_{f:S_f=S^\star}\ m_f^{(l)}.$$

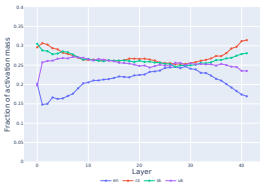
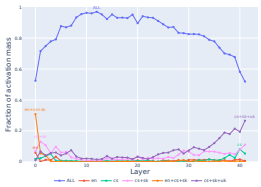
Thus, steering directions are chosen from the most active features associated with the target subset.

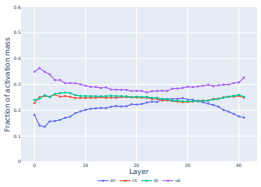
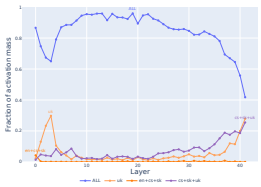
(a) Llama: Question `uk-164` in Czech

(b) Llama: Question `uk-165` in Ukrainian

(c) Gemma: Question `uk-164` in Czech

(d) Gemma: Question `uk-164` in Ukrainian

Figure 2: **Layerwise language attribution for specific questions.** In each pair, the left panel shows the soft language-reliance scores $R_\ell^{(l)}(x)$, while the right panel shows the hard-label subset attribution scores $R_S^{(l)}(x)$; in the latter, only subsets whose score exceeds 0.01 at any layer are shown. **(a) and (c):** The left plots shows that for this question, even though the input language is Czech, Slovak dominates all other languages at most intermediate layers for Llama and multiple languages essentially tie at many intermediate layers for Gemma; the right plots shows that the `ALL` feature is generally most prevalent. **(b):** The left plots shows that Ukrainian dominates other languages at all layers for this question, a property of *all* Ukrainian questions across both models. The right plots shows that `ALL`, `cs+sk+uk`, and `uk` features are most prevalent.

Constructing the steering vector Each selected SAE feature corresponds to a decoder direction from the SAE decoder matrix $W_{\text{dec}}^{(l)} \in \mathbb{R}^{d\times k}$. Let $w_f^{(l)} \in \mathbb{R}^d$ denote the decoder vector corresponding to feature f. To prevent decoder vectors with larger norms from dominating the intervention, we first normalize each decoder vector independently: $\tilde{w}_f^{(l)} = w_f^{(l)}/\|w_f^{(l)}\|_2$. The final steering vector for subset $S^\star$ at layer l is then constructed by summing the normalized decoder directions:

$$v_{S^\star}^{(l)} = \sum_{f\in F_{S^\star}^{(l)}} \tilde{w}_f^{(l)}$$

Applying the intervention Let $h_i^{(l)} \in \mathbb{R}^d$ denote the residual stream activation at layer l and token position i during inference. Given steering strength $\alpha \in \mathbb{R}$, we apply the intervention by modifying the residual stream as

$$\hat{h}_i^{(l)} = h_i^{(l)} + \alpha \cdot v_{S^\star}^{(l)}$$

Interventions are applied through a forward hook on the residual stream corresponding to layer l. For all experiments, steering is applied only at the final token position of the input prompt.

4 Experimental Setup

Dataset We use the CUS-QA (Libovický et al., 2026) dataset, a parallel dataset of 1,408 diverse, open-ended fact-based questions about three countries (Czechia, Slovakia, and Ukraine) available in English, Czech, Slovak, and Ukrainian; unlike many multilingual benchmarks, its questions are neither multiple-choice nor generated from a small set of templates. We split the dataset into 1,008 training, 100 validation, and 300 test questions; the split is performed over underlying questions, with all four parallel-language versions of each question kept in the same partition. We identify features using the train set and report results on the test set.

LLMs & SAEs We study Llama 3.1-8B (Grattafiori et al., 2024) and Gemma 2-9B (Gemma Team, 2024), both pretrained dense decoder-only transformers. We select these models because they have publicly available SAEs across all layers, enabling us to analyze multilingual processing throughout the transformers. For Llama, we use the `llamascope-res-32k` SAEs from Llama Scope (He et al., 2024), with 32K features per layer. For Gemma, we use the 16K-width canonical SAEs from `gemma-scope-9b-pt-res` in Gemma Scope (Lieberum et al., 2024). Both suites are trained on



(a) **Llama alignment using** $R_\ell^{(l)}(x)$ (b) **Llama alignment using** $R_S^{(l)}(x)$ (c) **Gemma alignment using** $R_\ell^{(l)}(x)$ (d) **Gemma alignment using** $R_S^{(l)}(x)$

Figure 3: **Multilingual alignment.** For each question, the average TV distance between pairs of languages with the same (aligned) outputs is computed, as well for pairs of languages with different (non-aligned) outputs. These values are then averaged over all questions in the test set; the results show that language profiles are more similar among pairs of languages when outputs are aligned than when they are not aligned.

residual-stream activations.

Evaluation Because the CUS-QA questions are free-response (not multiple choice or cloze), we use model-based evaluation to score outputs.[4] We choose GPT-4o mini (OpenAI, 2024) for model based evaluation. We evaluate along two dimensions: **correctness** (is the model's output correct compared to the gold answer?) and **alignment** (do different languages give the same answer, regardless of output language and correctness?).[5] Ideally, a good model would produce correct answers regardless of input language, meaning it should score high on both of these metrics. A study comparing GPT-4o mini's evaluations with three human evaluations is provided in Appendix A.

Steering We use a steering strength of $\alpha = 10$ for Llama and $\alpha = 20$ for Gemma, and $K = 1$ for both models; we always apply steering at the final token of the prompt in the middle layer (15 for Llama and 20 for Gemma).[6]

5 Main Findings

5.1 Analyzing Reliance on Multilinguality Information

We rank questions x by the strongest sustained pivot away from the input language L_x. For each layer l, we define the input margin

$$m^{(l)}(x) = R_{L_x}^{(l)}(x) - \max_{L \neq L_x} R_L^{(l)}(x)$$

Negative values indicate that some non-input language receives more attributed activation mass than the input language. The sustained pivot area is the most negative contiguous segment across layers:

$$\text{PivotArea}(x) = \min_{a \leq b} \sum_{l=a}^{b} m^{(l)}(x)$$

This score captures both the strength and duration of a pivot away from the input; if it is ≥ 0 then no layer exhibits a pivot away from the input language.

We compute this score separately for each input language and question on the test set. For Llama, 70% of Czech questions and 1.3% of English questions pivot away from the input language, while Slovak and Ukrainian never pivot at any layer. For Gemma, the corresponding rates are 24% for Czech, 19% for English, 0.6% for Slovak, and 0% for Ukrainian. Interestingly, Llama pivots substantially more than Gemma on Czech while Gemma exhibits more pivoting for English; Slovak is input-dominant on nearly all questions, while Ukrainian is input-dominant on all questions in both models. Figure 2 illustrates these patterns through layerwise soft language-reliance scores $R_\ell^{(l)}(x)$ and hard-label subset attribution scores $R_S^{(l)}(x)$, with Llama examples in the top row and Gemma examples in the bottom row.

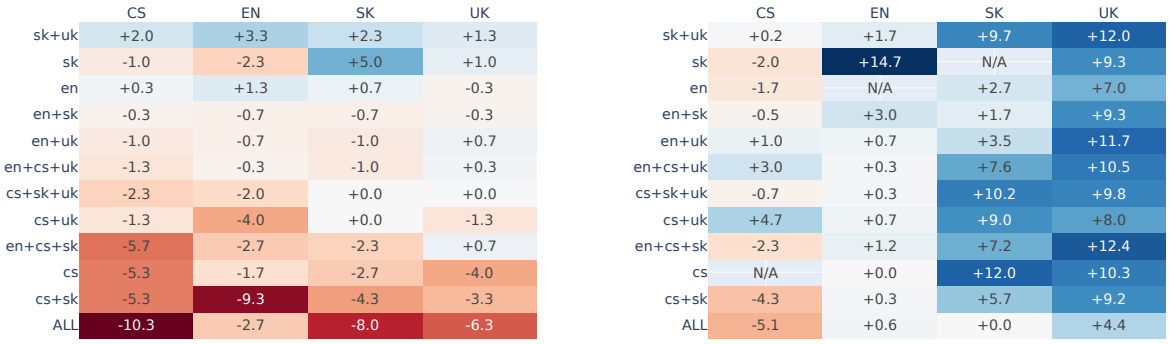
5.2 Multilingual Alignment

For each question, we determine which input languages produce the same answers via model-based evaluation (see §4). For each pair of languages, we then determine the average TV-distance of the

[4] Model-based evaluation works well in this setting because most outputs are short-response facts (e.g. "On the Teteriv River" or "1603 meters") that are easy for an LLM to verify against gold answers.

[5] In early experiments, we also analyzed **language** (does the model return the answer in the same language as the input?), but we abandoned this because it is not well-defined (e.g. does including diacritic marks mean an answer is not in English?); we instead only analyze the semantic content of the answer, using the correctness and alignment metrics.

[6] We find this setting works reasonably well, though during our preliminary investigations we explored far more values for α and K as well as steering locations.



(a) **Accuracy change** (b) **Alignment change**

Figure 4: **Performance change when steering on Llama.** Shows the change in accuracy and change in alignment scores when steering each input language (columns) to each subset (rows) on the entire test set, as compared to the non-steered baseline. When steering language ℓ toward subset S, we compute the alignment score over $\ell \cup S$, reflecting the expectation that these languages should become increasingly aligned after such steering.

language profiles of each layer, and determine the average among answers that are aligned (regardless of correctness) and the average among those that are not aligned. Figure 3 depicts these values averaged over all questions for Llama and Gemma. The results show that when two languages produce the same answer on a question, the language profiles tend to be more similar (lower TV-distance) than when two languages produce different answers. We also see that language profiles tend to be more similar in middle layers for both Llama and Gemma.

5.3 Steering

Figure 4 shows the results of steering Llama toward various subsets for the four input languages. In Figure 4a, we see that steering toward SK+UK gives observed accuracy increases for all input languages, while all other subsets are more varied. Further, Ukrainian inputs benefit from steering toward many subsets; this is likely due to Ukrainian's linguistic distance from the other languages. We evaluate the SK+UK intervention using the exact paired-permutation test of (Zmigrod et al., 2022), with Holm correction across the four input languages, finding that the correctness increases are not statistically significant.

We also explore whether steering a language ℓ toward a subset S in Llama leads to more alignment among the languages $\{\ell\} \cup S$; this is shown in Figure 4b. Strikingly, Ukrainian increases its alignment scores dramatically when steering, as does Slovak; English is largely unchanged, and Czech tends to have decreased alignment. For SK+UK steering specifically, alignment increases are statistically significant at $p < 0.01$ for English ($+1.67$ points), Slovak ($+9.67$), and Ukrainian ($+12.00$), but not Czech ($+0.17$). Thus, the statistical evidence is stronger for alignment, the intervention's more direct target, than for correctness. In §5.1, we found that Slovak and Ukrainian never pivot toward another language during their computation in Llama, relying more on a language internal "shortcut"; the results here suggest that steering these languages brings them closer to the outputs of other languages, mitigating the shortcut effect. English and Czech have less of an alignment increase after steering.

For Gemma, when we steer toward SK+UK, we find that correctness changes by $+0.7$, -0.3, $+0.7$, and 0.0 percentage points for English, Czech, Slovak, and Ukrainian, respectively; alignment changes by 0.0, -0.3, -1.0, and $+1.0$ points. Unlike Llama, Gemma shows neither consistent correctness gains nor substantial alignment gains. Thus, while the contrast in language-reliance representations recurs across models, steering effects appear more model-dependent.

6 Discussion

Our results suggest that multilingual factual recall is not governed by a single mechanism. Some inputs remain anchored to the prompt language, while others shift toward non-input language profiles. In both Llama and Gemma, Czech and English exhibit more pivoting than Slovak and Ukrainian, although the rates differ somewhat (English rarely pivots in Llama but does so more fre-

quently in Gemma, and Czech pivots substantially more in Llama than Gemma). This contrast between the processing of Czech/English and of Slovak/Ukrainian in both models suggests that input language systematically influences which representations are accessed during factual recall. Variation across questions within a language further indicates that these patterns depend on the question as well as the language. We therefore do not interpret the results as selecting one hypothesis, but as evidence that language-associated representations varies across input languages and questions.

The steering results are more model-dependent. In Llama, steering toward SK+UK yields observed accuracy increases across all input languages, while Ukrainian and Slovak show large alignment gains when steered toward other subsets. This suggests that inputs that remain language-anchored can be pushed toward more cross-lingually aligned behavior. However, Gemma's SK+UK intervention produces only small, mixed changes in correctness and alignment. Thus, the recurring representational patterns do not imply that the same steering methods will have comparable effects across models.

More broadly, these findings support labeling SAE features by language subsets. A single-language taxonomy would miss features shared across related linguistic contexts. This helps explain why subset-level steering is useful: it can target shared cross-lingual directions, not merely single-language features. SAE features do not by themselves prove that the model translates, shortcuts, or uses a shared multilingual representation, but they provide a concrete interface for measuring and causally manipulating the language-associated pathways used during factual recall.

7 Conclusion & Future Work

We studied multilingual factual recall in Llama 3.1-8B and Gemma 2-9B through the lens of SAE features, asking whether models retrieve facts through shared multilingual representations, input-language shortcuts, or pivot-like pathways associated with other languages. Language-reliance profiles vary across input languages and across questions within a language, while aligned answers tend to have more similar profiles. Both models exhibit more pivoting for Czech and English than for Slovak and Ukrainian, although the rates differ substantially. Steering effects are more model-dependent: Gemma's SK+UK intervention produces small, mixed effects while steering Llama toward various subsets produces substantial alignment increases (particularly on Ukrainian). These findings support subset-labeled SAE features as an interface for interpreting multilingual knowledge access and testing its response to intervention. A natural next step is to extend the analysis to *generated* text, tracking how language-subset reliance evolves across tokens, differs between correct and incorrect continuations, and responds to steering. Applying the framework to broader multilingual QA resources, such as MKQA (Longpre et al., 2021), and additional language combinations would test how far these findings extend beyond CUS-QA.

Limitations

This study examines only two models and SAE families: Llama 3.1-8B with Llama Scope 32K SAEs and Gemma 2-9B with Gemma Scope 16K SAEs. Although some language-reliance patterns recur across both models, their steering responses differ, and generalization to other model scales, architectures, and SAE training setups remains uncertain. Our steering experiments also explore a limited region of the intervention space. Broader evaluation across layers, feature-set construction, steering strengths, token positions, and generation-time interventions would clarify when subset steering is beneficial, when it degrades behavior, and how robust its effects are.

Our empirical claims are also limited to CUS-QA: open-ended factual questions about Czechia, Slovakia, and Ukraine, asked in Czech, Slovak, Ukrainian, and English. Its parallel questions let us hold the underlying fact fixed while varying languages, but the observed language-reliance patterns may differ for other topics, question styles, and language combinations. We cannot infer that the same subsets or pivot patterns will appear across multilingual factual QA more generally.

Acknowledgments

This research used the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta is a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications.

We used AI assistants for coding, proofreading, and LaTeX formatting. The authors reviewed all outputs and take responsibility for the final content.

References

Lyzander Marciano Andrylie, Inaya Rahmanisa, Mahardika Krisna Ihsani, Alfan Farizki Wicaksono, Haryo Akbarianto Wibowo, and Alham Fikri Aji. 2026. Sparse autoencoders can capture language-specific concepts across diverse languages. *Preprint*, arXiv:2507.11230.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2025. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024. Scaling and evaluating sparse autoencoders. *Preprint*, arXiv:2406.04093.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235, Singapore. Association for Computational Linguistics.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024. Llama Scope: Extracting millions of features from Llama-3.1-8B with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Zhengbao Jiang, Antonios Anastasopoulos, Jun Araki, Haibo Ding, and Graham Neubig. 2020a. X-FACTR: Multilingual factual knowledge retrieval from pretrained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5943–5959, Online. Association for Computational Linguistics.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020b. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438.

Nora Kassner, Philipp Dufter, and Hinrich Schütze. 2021. Multilingual LAMA: Investigating knowledge in multilingual pretrained language models. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3250–3258, Online. Association for Computational Linguistics.

Jindřich Libovický, Jindřich Helcl, Andrei Manea, and Gianluca Vico. 2026. CUS-QA: Local-knowledge-oriented open-ended question answering dataset. *Preprint*, arXiv:2507.22752.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma Scope: Open sparse autoencoders everywhere all at once on Gemma 2. *Preprint*, arXiv:2408.05147.

Shayne Longpre, Yi Lu, and Joachim Daiber. 2021. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Transactions of the Association for Computational Linguistics*, 9:1389–1406.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in GPT. *Preprint*, arXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Thong Nguyen. 2024. Logit Prisms: Decomposing Transformer Outputs for Mechanistic Interpretability.

OpenAI. 2024. GPT-4o mini: advancing cost-efficient intelligence. https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Jirui Qi, Raquel Fernández, and Arianna Bisazza. 2023. Cross-lingual consistency of factual knowledge in multilingual language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10650–10666, Singapore. Association for Computational Linguistics.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15366–15394, Bangkok, Thailand. Association for Computational Linguistics.

Shaoyang Xu, Junzhuo Li, and Deyi Xiong. 2023. Language representation projection: Can we transfer factual knowledge across languages in multilingual language models? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3692–3702, Singapore. Association for Computational Linguistics.

Ran Zmigrod, Tim Vieira, and Ryan Cotterell. 2022. Exact paired-permutation testing for structured test statistics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4894–4902, Seattle, United States. Association for Computational Linguistics.

A Human Validation of Model-Based Evaluation

We validated the model-based correctness and alignment judgments of GPT-4o mini (on the outputs of the CUS-QA dataset on Llama 3.1-8B) against three human annotators (the authors). For correctness, the unit of evaluation is a question–language response. All three annotators judged 616 responses independently and gave the same correctness judgment for 610 of them (99.03%). The model-based judge agreed with the human majority on 615 of 616 responses (99.84%; 95% Wilson score confidence interval: [99.09%, 99.97%]).

For alignment, the unit of evaluation is the output of a given question in each of the 4 languages (English, Czech, Slovak, and Ukrainian). Each annotator partitioned the 4 answers into groups with the same semantic content. We report results for all subsets of two, three, or four languages in Table 2, showing the percentage of cases in which all three humans agreed and the proportion in which the LLM-based judge agreed with *any* human partition's. In order to "agree", we require that the judges' partitions match *exactly* on each language, making the metric very strict.

The LLM-judge's alignment judgments are less reliable than its correctness ones, particularly when comparing all four languages; results based on that measure should therefore be interpreted with this limitation in mind.

B Additional Figures

Figure 5 shows the number of features in Gemma with each subset label at each layer.

Languages	Human agreement	LLM-Judge agreement	95% Wilson Score Interval
EN+CS	96.6%	91.2%	[85.5%, 94.8%]
EN+SK	94.6%	88.4%	[82.3%, 92.7%]
EN+UK	94.6%	94.6%	[89.6%, 97.2%]
CS+SK	94.6%	76.9%	[69.4%, 83.0%]
CS+UK	95.2%	87.1%	[80.7%, 91.6%]
SK+UK	94.6%	84.4%	[77.6%, 89.3%]
EN+CS+SK	90.5%	66.0%	[58.0%, 73.2%]
EN+CS+UK	89.8%	77.6%	[70.2%, 83.5%]
EN+SK+UK	88.4%	70.1%	[62.2%, 76.9%]
CS+SK+UK	88.4%	63.3%	[55.2%, 70.6%]
EN+CS+SK+UK	83.7%	54.4%	[46.4%, 62.3%]

Table 2: **Human validation of answer-alignment judgments by language subset.** Each row evaluates 147 questions for the indicated set of input languages. Human agreement is the percentage for which **all** three annotators grouped the answers in the same way. Judge agreement is the percentage for which the LLM judge grouped the answers in the same way as **at least one** human annotator. The final column gives 95% Wilson score confidence intervals for judge agreement.

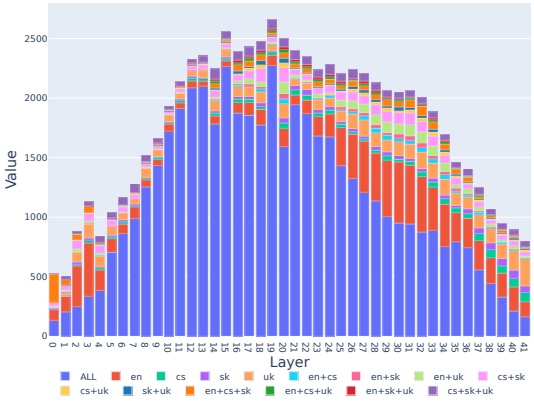


Figure 5: **Number of features in Gemma with each subset label.** For each layer, shows the number of features labeled $S \subseteq \mathcal{L}$, where ALL represents the full set of languages $\mathcal{L}$.

C Prompts

Example question in English

```
You are an experienced trivia game contestant. Answer the following question
↪ correctly and as concisely as possible without giving extra information.
Question: [QUESTION]
Answer:
```

Correctness

```
You are a strict judge for a multilingual trivia task.
Your job is to determine whether the contestant's answer is correct based on
↪ meaning.
Return only YES or NO.

### Instructions
Decide whether the contestant's answer is correct.

- The contestant's answer does NOT need to exactly match any reference answer
- It is correct if it expresses the same meaning as any valid answer
- Ignore language differences
- Ignore explanation; but if the explanation contradicts the answer, return NO
- If the contestant's response is hedging or not giving a clear answer, return
↪ NO
- If the contestant provides lots of repeating text that does not include the
↪ answer, return NO
- Minor spelling differences are fine
- Return only YES or NO without any extra text

### Example

Questions:
- en: What is the capital of France?
- cs: Jaké je hlavní město Francie?
- sk: Aké je hlavné mesto Francúzska?
- uk: Яка столиця Франції?

Valid (Gold) answers in some languages:
- (en) Paris
- (cs) Paříž
- (sk) Paríž
- (uk) Париж

Contestant answer: Париж

Output: YES

### Your task:

Questions:
- en: [Q_EN]
- cs: [Q_CS]
- sk: [Q_SK]
```

```
- uk: [Q_UK]

Valid (Gold) answers in some languages:
- (en) [GOLD_EN]
- (cs) [GOLD_CS]
- (sk) [GOLD_SK]
- (uk) [GOLD_UK]

Contestant answer: [PRED_UK]

### Output (only return YES or NO without any extra text):
```

Alignment

```
You are a strict judge of answer equivalence.
Return only valid JSON matching the schema.

### Instructions
Group the answers into equivalence classes based on their specific meaning.

Rules:
- Two answers belong in the same group only if they express the same specific
↪  underlying answer
- This is regardless of language
- Ignore wording differences, formatting, and extra explanation
- Do NOT group answers together merely because they are all incorrect
- Two incorrect answers should be grouped together only if they refer to the
↪  same specific entity, title, phrase, or fact
- If one answer says "I don't know" or gives no real answer, do not group it
↪  with other incorrect answers unless they also give no answer
- Use as few groups as possible, but do not merge distinct answers

### Example 1

Questions:
- en: What is the capital of France?
- cs: Jaké je hlavní město Francie?
- sk: Aké je hlavné mesto Francúzska?
- uk: Яка столиця Франції?

Answers:
- en: Paris
- cs: München
- sk: Париж
- uk: Munich

Output: {"en": "G1", "cs": "G2", "sk": "G1", "uk": "G2"}

### Example 2

Questions:
- en: What is the capital of France?
- cs: Jaké je hlavní město Francie?
```

```
- sk: Aké je hlavné mesto Francúzska?
- uk: Яка столиця Франції?

Answers:
- en: London
- cs: Prague
- sk: Londýn
- uk: I don't know

Output: {"en": "G1", "cs": "G2", "sk": "G1", "uk": "G3"}

### Your task:

Questions:
- en: [Q_EN]
- cs: [Q_CS]
- sk: [Q_SK]
- uk: [Q_UK]

Answers:
- en: [PRED_EN]
- cs: [PRED_CS]
- sk: [PRED_SK]
- uk: [PRED_UK]

### Output (only return the JSON in the proper format without any extra text):
```