Translation Circuit Discovery and Targeted Optimization for Low-Resource Multilingual Translation

Atai Waris Khan, Abdullah Mazhar, Raghu Kumar, Md Shad Akhtar
Indraprastha Institute of Information Technology Delhi;
{ataik,abdullahm,raghu22386,shad.akhtar}@iiitd.ac.in

Abstract

Recent multilingual large language models achieve strong translation performance, yet the internal mechanisms responsible for low-resource multilingual translation remain insufficiently understood. In this paper, we study translation circuits in multilingual decoder-only LLMs for Hindi→Urdu and Hindi→Sindhi (Devanagari). We propose TITAN, a mechanistic framework that combines causal intervention, sparse feature decomposition, and targeted preference optimization to identify and optimize translation-crucial transformer components. Unlike conventional full-parameter fine-tuning, TITAN restricts optimization to sparse causally important circuits, enabling scalable and parameter-efficient adaptation while updating fewer than 5% of model parameters. Experimental results show that TITAN achieves BLEU improvements of +22.67% on Hindi→Urdu and +45.86% on Hindi→Sindhi over the multilingual baselines. Human expert evaluation further confirms substantial gains in adequacy and fluency across both language pairs. Additional analyses reveal that the identified translation circuits for the studied Indic language pairs operate over a shared semantic subspace rather than through an English-centric pivot representation. Overall, our findings demonstrate that mechanistic circuit discovery can directly improve low-resource multilingual translation while providing interpretable insights into multilingual transformer behavior.

1 Introduction

Recent multilingual large language models have demonstrated strong translation capabilities across many high-resource language pairs (Zhu et al., 2024; Xu et al., 2024; Alves et al., 2024). At the same time, mechanistic interpretability studies have begun analyzing how translation behavior emerges inside transformer models through circuit analysis and causal intervention (Zhang et al., 2026). However, existing work remains largely focused on English-centric and high-resource settings, leaving the mechanisms underlying low-resource multilingual translation insufficiently explored. This limitation is particularly important for South Asian languages, which often exhibit strong lexical overlap, shared grammatical structure, script variation, and phonological divergence (Subbarao, 2012). Consequently, current fine-tuning approaches optimize model parameters without identifying which components are causally responsible for translation behavior.

In this work, we study multilingual translation circuits in decoder-only LLMs for two low-resource South Asian directions: Hindi→Urdu and Hindi→Sindhi (Devanagari). These language pairs provide complementary linguistic settings: Hindi and Urdu share substantial lexical and grammatical overlap but differ primarily in script, whereas Hindi and Sindhi additionally differ in phonology and lexical composition due to Sindhi-specific implosive consonants and Persian-Arabic lexical influence (Raza et al., 2004). This setting enables controlled analysis of whether related low-resource translations rely on shared or specialized computational mechanisms.

To address this problem, we propose TITAN (**T**ranslation-aware **I**nterpretability through **T**argeted **A**ctivation **N**etworks), a mechanistic framework for parameter-efficient multilingual translation. TITAN combines CIRCE (**C**ausal **I**ntervention for **R**epresentation **C**ircuit **E**ngineering), sparse feature decomposition, and targeted preference optimization to identify and optimize translation-crucial transformer components. Unlike conventional full-parameter fine-tuning, TITAN restricts optimization to sparse causally important circuits, improving both interpretability and parameter efficiency.

Translation Circuit. We define a translation circuit as a sparse set of transformer components

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 28626–28643
October 24-29, 2026 ©2026 Association for Computational Linguistics

that causally contribute to a specific translation behavior. These components are identified through causal intervention by measuring their effect on translation-relevant outputs, yielding a circuit comprising causally important attention heads and MLP blocks for the studied translation task.

Our analysis reveals that the identified translation circuits for the studied low-resource Indic language pairs are distributed across attention and MLP components. We further show that the identified translation circuits for the studied Indic translation directions operate through a shared semantic subspace rather than through the English-centric pivot mechanisms reported in prior work (Zhang et al., 2026). Experimentally, TITAN achieves BLEU improvements of +33.75% on Hindi→Urdu and +45.86% on Hindi→Sindhi while updating fewer than 5% of model parameters. Human expert evaluation further confirms substantial improvements in adequacy and fluency across both language pairs. We additionally validate the generalisability of the discovered sparse-circuit behavior on English↔French and English↔Hindi translation directions.

Contributions. Our contributions are summarized as follows:

- We propose TITAN, a mechanistic framework that combines causal intervention, sparse feature decomposition, and targeted preference optimization for parameter-efficient multilingual translation.
- We show that the identified translation circuits for the studied low-resource Indic language pairs operate through a shared semantic subspace rather than an English-centric pivot mechanism.
- We demonstrate that restricting optimization to causally identified translation circuits substantially improves translation quality while updating fewer than 5% of model parameters.

Reproducibility. Code and dataset is open-sourced on GitHub.

2 Related Work

Low-resource neural machine translation, mechanistic interpretability, and multilingual large language models have emerged as closely connected research directions. Prior work has improved multilingual transfer and translation quality for under-represented languages, while recent mechanistic interpretability research enables causal analysis of transformer computations through activation intervention and sparse feature decomposition.

Low-Resource Neural Machine Translation. Low-resource neural machine translation (NMT) remains challenging due to limited parallel corpora and linguistic diversity across Indic languages. Recent work has explored multilingual transfer learning, back-translation, transliteration, and pivot-based modeling to improve translation quality for low-resource Indian languages (Wei et al., 2024; P M et al., 2024; Bala Das et al., 2024). Recent parameter-efficient approaches further investigate language-specific adaptation, including language-specific LoRA, intrinsic language-specific subspaces, and language-aware parameter selection for multilingual knowledge transfer. These methods improve language-specific adaptation but primarily focus on parameter specialization rather than identifying components that are causally responsible for translation behavior.

Mechanistic Interpretability and Multilingual LLMs. Mechanistic interpretability seeks to reverse-engineer transformer computations through causal interventions and circuit analysis. Prior work has used activation and path patching to identify circuits underlying reasoning and attention. In multilingual neural machine translation, studies have identified language-specific subspaces emerging during multilingual fine-tuning (Cao et al., 2024), as well as language-aware parameters for targeted cross-lingual knowledge transfer (Dong et al., 2025). Language-specific fine-tuning has further been explored for adapting LLMs to low-resource translation settings (Liang et al., 2025). Recent circuit-discovery work, such as Emotion Circuits (Wang et al., 2025), combines contrastive examples with causal analysis to identify functionally relevant representations. Unlike these approaches, TITAN focuses on discovering translation-specific circuits and leveraging them for targeted multilingual translation optimization.

Preference Optimization for Translation Alignment. Preference optimization methods such as Direct Preference Optimization have recently been used to improve multilingual generation and translation quality through pairwise preference learning. Existing approaches generally optimize model parameters without identifying which components are causally responsible for translation behavior.

Pair	Split	#Sent	Hindi		Target Language		Avg. Len.
			#Tokens	#Types	#Tokens	#Types	
Hi–Sd	SFT Train	20K	404.6k	36.1K	398.2K	49.7K	20.1
	DPO Train	15K	302.8K	30.5K	297.8K	41.6K	20.0
	Contrastive-From SFT	10K	220.6K	15.8K	217.6K	20.5K	21.9
Hi–Ur	SFT Train	30K	562.3K	47.3K	599.2K	43.7K	19.4
	DPO Train	30K	567.1K	47.4K	603.1K	43.8K	19.5
	Contrastive-From SFT	10K	186.9K	15.6K	198.9K	15.1K	19.3

Table 1: Corpus statistics for the Hindi–Sindhi (Devanagari) and Hindi–Urdu parallel corpora used in TITAN. Each corpus is partitioned into SFT, DPO, and contrastive analysis splits. We sample a 10K contrastive subset from the SFT split of each language pair to construct contrastive-based intervention prompts for causal translation circuit identification. **Hi–Sd**: Hindi–Sindhi (Devanagari); **Hi–Ur**: Hindi–Urdu. Avg. Len. denotes the mean number of whitespace-tokenized tokens per sentence.

In contrast, TITAN combines mechanistic circuit discovery with targeted optimization to selectively optimize translation-relevant components.

3 Dataset

We evaluate TITAN on two low-resource South Asian translation pairs: Hindi–Urdu and Hindi–Sindhi (Devanagari). These language pairs provide complementary multilingual settings involving script divergence, lexical overlap, and phonological variation. Corpus statistics are summarized in Table 1.

3.1 Translation Corpora

We use the Hindi–Urdu and Hindi–Sindhi subsets of the Bharat Parallel Corpus Collection (BPCC) (Gala et al., 2023). Hindi and Urdu share substantial grammatical and lexical overlap but differ primarily in script, with Hindi written in Devanagari and Urdu in Arabic-Nastaliq. In contrast, although Hindi and Sindhi both use the Devanagari script, Sindhi additionally contains implosive consonants and stronger Persian-Arabic lexical influence.

After preprocessing and deduplication, the Hindi–Urdu corpus contains 60K parallel sentence pairs, while the Hindi–Sindhi corpus contains 35K sentence pairs. Both corpora are partitioned into supervised fine-tuning (SFT), Direct Preference Optimization (DPO), and contrastive analysis splits.

3.2 Contrastive Translation Data

To support mechanistic analysis, we construct contrastive translation pairs consisting of valid translation prompts and counterfactual prompts where translation intent is removed while preserving grammatical structure. We sample 10K sentence pairs from the SFT split of each language pair and generate counterfactual variants using semantic perturbation strategies. For Hindi–Sindhi, sampling is additionally stratified to ensure sufficient coverage of Sindhi-specific implosive phoneme markers. Detailed prompt templates, perturbation taxonomies, and validation protocols are provided in Appendix Section A.

3.3 Preprocessing

We apply uniform preprocessing across both corpora, including Unicode normalization, duplicate removal, and sentence-length filtering. Hindi and Sindhi samples are validated using Devanagari Unicode ranges, whereas Urdu samples are verified using Arabic-script code points. For Hindi–Sindhi, we additionally preserve Sindhi-specific implosive phoneme markers to support phonological circuit analysis.

4 Methodology

We propose **TITAN** (**T**ranslation-aware **I**nterpretability through **T**argeted **A**ctivation **N**etworks), a mechanistic interpretability-guided framework for parameter-efficient multilingual translation. Rather than updating all transformer parameters, TITAN identifies the sparse set of components causally responsible for translation behavior and restricts optimization to the discovered translation circuits. The framework is built upon CIRCE (**C**ausal **I**ntervention for **R**epresentation **C**ircuit **E**ngineering), which combines subspace intervention with sparse feature decomposition to isolate translation-relevant computation. All CIRCE and TITAN experiments use Gemma-2 9B

as the underlying base model.

Hindi–Urdu and Hindi–Sindhi provide complementary multilingual settings for circuit analysis. Hindi–Urdu exhibits strong grammatical overlap with script divergence, whereas Hindi–Sindhi shares script similarity but differs substantially in phonology and lexical composition due to Sindhi-specific implosive phonemes and Persian-Arabic lexical influence. This setting enables TITAN to analyze how translation circuits specialize across distinct forms of linguistic variation.

4.1 Contrastive Translation Construction

To isolate translation-specific computation from general language modeling behavior, we construct contrastive translation pairs consisting of a valid translation prompt X^+ and a counterfactual prompt X^-:

$$X^+ = \text{“[SRC]: [text] - [TGT]:”} \quad (1)$$

Counterfactual prompts preserve grammatical structure while removing translation intent through target nullification, semantic distortion, and contradictory task descriptors. This design allows TITAN to separate activations associated with translation generation from those associated with general multilingual processing.

Each sample additionally contains translation direction labels, vocabulary overlap annotations, and phoneme-level markers. For Hindi–Sindhi, we explicitly annotate samples containing Sindhi-specific implosive phonemes to support phonological circuit analysis.

4.2 CIRCE: Translation Circuit Identification

TITAN identifies translation-relevant transformer components using CIRCE, a subspace-constrained extension of activation patching. For a transformer component c at layer ℓ, we compute activation differences between positive and counterfactual prompts:

$$\Delta \mathbf{a}_c^{(i)} = \mathbf{a}_c(X_i^+) - \mathbf{a}_c(X_i^-) \quad (2)$$

The resulting activation differences are aggregated into a component-specific activation matrix $\mathbf{M}_c$ and decomposed into translation-relevant structure and residual variation:

$$\min_{\mathbf{S}_c, \mathbf{E}_c, \mathbf{\Gamma}} \left\| \mathbf{M}_c - \mathbf{S}_c \mathbf{1}^\top - \mathbf{E}_c \mathbf{\Gamma}^\top \right\|_F \quad \text{s.t.} \quad \text{Span}(\mathbf{S}_c) \perp \text{Span}(\mathbf{E}_c) \quad (3)$$

Here, $\mathbf{S}_c$ denotes a single translation-relevant direction, while $\mathbf{E}_c$ captures residual variation; this direction defines the intervention basis $\mathbf{U}_c$. Instead of replacing the full activation, CIRCE intervenes only along the translation-relevant direction:

$$\tilde{\mathbf{a}}_c = \mathbf{U}_c \mathbf{U}_c^\top \mathbf{a}_c(X^-) + (\mathbf{I} - \mathbf{U}_c \mathbf{U}_c^\top) \mathbf{a}_c(X^+) \quad (4)$$

Component importance is measured using intervention-induced translation logit changes. Rather than using fixed thresholds, TITAN calibrates component selection from the empirical intervention distribution using a P99 noise-floor criterion, yielding an operating threshold near 0.25%. This preserves sparse but reproducible low-resource translation signals while filtering negligible intervention effects. Components producing substantial translation degradation under intervention are retained as translation-crucial, forming the translation circuit used for optimization; CIRCE applies targeted SFT to these components while freezing all others.

For interpretability, discovered attention heads are further categorized into source, indicator, and positional heads based on their dominant routing behavior.

4.3 Sparse Feature Decomposition

Transformer activations frequently exhibit feature superposition, where multiple linguistic properties are encoded within the same representation. To disentangle these representations, TITAN trains Sparse Autoencoders (SAEs) exclusively on residual activations extracted from CIRCE-identified translation-crucial layers. Restricting SAE training to CIRCE-identified layers ensures that the discovered sparse features remain grounded in causally relevant translation circuitry rather than generic multilingual representations.

For an activation vector $\mathbf{h}^{(\ell)}$, the encoder computes a sparse latent representation:

$$\mathbf{z}^{(\ell)} = \text{ReLU}\left((\mathbf{W}_\ell)^\top \mathbf{h}^{(\ell)} + \mathbf{b}_\ell \right) \quad (5)$$

The original activation is reconstructed through:

$$\hat{\mathbf{h}}^{(\ell)} = \mathbf{W}_\ell \mathbf{z}^{(\ell)} \quad (6)$$

Training minimizes a joint reconstruction and sparsity objective:

$$\mathcal{L}_{\text{SAE}} = \|\mathbf{h}^{(\ell)} - \hat{\mathbf{h}}^{(\ell)}\|_2^2 + \lambda \|\mathbf{z}^{(\ell)}\|_1 \quad (7)$$

The resulting sparse representations expose disentangled translation features that are substantially easier to interpret than raw transformer activations.

4.4 Feature-Level Causal Scoring

We evaluate each SAE feature z_j using language selectivity and causal lift. To isolate language identity from semantic variation, selectivity is computed using semantically matched bilingual prompts expressing the same meaning in different languages:

$$\widetilde{\text{Sel}}_j = \frac{\mathbb{E}[z_j(x_{\text{tgt}})] - \mathbb{E}[z_j(x_{\text{src}})]}{\text{Std}[z_j(x_{\text{tgt}})] + \text{Std}[z_j(x_{\text{src}})] + \epsilon} \quad (8)$$

To estimate causal relevance beyond correlation, we intervene on feature j during decoding and measure the induced translation logit change:

$$\text{Lift}_j^{(\ell)} = \mathbb{E}_{x\sim D}\left[\frac{1}{|T|}\sum_{t\in T}\Delta M_\alpha^{(\ell,j)}(x,t)\right] \quad (9)$$

where T denotes decoding steps and $\Delta M_\alpha^{(\ell,j)}$ represents the translation logit change induced by intervention magnitude α.

We compute a joint feature score:

$$\text{Score}_j = \max(\widetilde{\text{Sel}}_j, 0) \cdot \max(\text{Lift}_j, 0) \quad (10)$$

Features exhibiting both high selectivity and positive causal lift are retained as translation-relevant features, while low-scoring features are discarded. This filtering stage provides a finer-grained characterization of the CIRCE-identified translation circuit by distinguishing causally relevant features from language-selective but causally weak features.

4.5 Targeted Translation Optimization

Using the discovered translation circuit, CIRCE performs targeted SFT on the identified components while freezing all others, whereas TITAN further applies DPO to the same components. Training proceeds hierarchically from word-level translation to phrase-level and finally sentence-level translation, allowing the model to gradually acquire increasingly complex translation structure.

To preserve the general language capabilities of the base model, optimization is regularized using a KL divergence constraint:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MT}} + \lambda_{\text{KL}}\text{KL}(p_\theta \| p_{\text{base}}) \quad (11)$$

where $\mathcal{L}_{\text{MT}}$ denotes the standard translation objective.

After supervised fine-tuning, TITAN further refines translation quality using Direct Preference Optimization. Preference pairs are constructed by treating the gold reference translations from the parallel corpus as preferred outputs, while translations generated by the supervised fine-tuned model are treated as rejected outputs.

5 Experiments and Results

Here, we compare TITAN against multilingual machine translation systems and decoder-only large language models across Hindi-Urdu and Hindi-Sindhi (Devanagari) translation tasks. We further analyze the effectiveness of targeted circuit optimization under both SFT and DPO settings. In addition to quantitative evaluation, we provide mechanistic analysis to study the role of sparse translation circuits in low-resource multilingual translation.

Baselines. We compare TITAN against both multilingual neural machine translation systems and instruction-tuned decoder-only LLMs under a unified evaluation setup. The evaluated baselines include NLLB-200 and IndicTrans2, which represent strong multilingual translation models specifically designed for low-resource settings, as well as Gemma-2, Qwen-3.5, LLaMA-3.1, and Mistral3, which represent recent decoder-only multilingual LLMs. All models are evaluated under zero-shot, supervised fine-tuning (SFT), and DPO settings. CIRCE denotes targeted SFT on the causally identified translation-crucial components, while TITAN further applies DPO to the same components.

Performance Comparison. Table 2 reports translation performance across Hindi–Urdu and Hindi–Sindhi translation directions. Zero-shot performance remains consistently weak across both multilingual MT systems and decoder-only LLMs, particularly for Hindi–Sindhi due to additional phonological and lexical divergence. Supervised fine-tuning and DPO improve translation quality, with Gemma-2 and NLLB-200 emerging as the strongest multilingual baselines across most directions.

TITAN consistently outperforms all baselines across both language pairs. The gains are most pronounced for Hindi–Sindhi, where TITAN achieves relative BLEU improvements of +45.86% and +48.98% for H→S and S→H respectively over

		Hindi–Urdu								Hindi–Sindhi (Devanagari)							
		H→U				U→H				H→S				S→H			
	Models	**BL**	**CF2**	**TER**	**COM**	**BL**	**CF2**	**TER**	**COM**	**BL**	**CF2**	**TER**	**COM**	**BL**	**CF2**	**TER**	**COM**
ZS	NLLB-200	0.01	0.09	105.65	57.08	0.31	0.61	108.63	55.39	0.44	1.05	101.23	41.34	2.31	10.58	91.36	43.08
	IndicTrans2	4.33	25.19	80.69	62.85	2.12	23.63	87.16	60.57	0.59	3.08	119.88	41.76	1.15	7.38	109.90	55.16
	Mistral-3	0.32	2.40	103.46	33.70	0.19	2.56	102.95	30.12	0.17	1.92	296.8	33.43	0.68	5.21	238.11	27.63
	LLaMA-3.1	1.46	3.78	117.57	43.83	0.11	4.20	100.79	44.42	0.51	3.63	273.22	37.52	1.21	6.93	250.01	33.43
	Qwen-3.5	0.21	3.02	99.23	40.30	0.53	3.30	99.02	41.22	0.21	3.02	99.23	34.80	0.53	3.30	109.02	31.20
	Gemma-2	5.93	28.23	69.81	59.05	6.40	25.58	79.30	53.29	0.62	5.93	187.74	39.87	2.26	8.42	79.30	36.41
SFT	NLLB-200	23.39	56.61	41.70	72.76	22.03	51.20	44.77	73.73	15.89	33.42	88.22	56.83	17.33	31.29	78.05	65.89
	IndicTrans2	23.63	41.43	76.24	70.87	25.36	50.33	49.36	74.38	14.83	34.57	78.65	60.2	18.66	48.96	81.23	67.14
	LLaMA-3.1	10.59	36.58	59.87	66.8	9.59	31.32	70.21	65.35	9.81	23.22	95.99	58.98	10.25	18.59	96.76	57.38
	Qwen-3.5	8.89	33.90	73.27	64.05	7.25	30.86	73.15	51.63	5.63	17.10	110.99	38.52	7.21	21.85	99.02	45.31
	Gemma-2	23.79	56.66	40.98	74.16	25.89	54.69	43.75	75.33	15.94	34.10	86.50	59.86	17.38	47.56	75.28	72.77
	Mistral-3	9.47	31.94	72.57	69.16	12.00	36.78	69.28	67.22	5.44	20.38	102.90	55.77	5.58	23.63	95.25	51.93
DPO	NLLB-200	25.86	59.36	40.14	74.98	23.51	55.52	43.00	75.15	17.00	40.92	58.22	58.39	18.34	50.15	51.13	70.64
	IndicTrans2	24.68	58.25	50.22	71.52	24.96	56.30	40.58	74.84	17.32	40.30	70.60	61.9	20.63	50.91	72.55	68.56
	LLaMA-3.1	11.29	39.18	53.63	68.37	9.93	34.50	57.70	65.83	10.21	27.66	72.36	59.89	12.11	23.32	70.39	61.85
	Qwen-3.5	8.90	33.70	71.64	64.98	7.32	30.90	71.21	52.74	8.98	29.31	96.44	44.96	8.36	30.02	87.37	50.04
	Gemma-2	25.94	60.10	42.37	75.88	25.78	56.10	44.39	77.37	18.01	43.68	84.71	62.81	19.89	51.20	69.25	74.55
	Mistral-3	9.29	30.61	73.60	68.84	11.22	36.10	68.28	70.01	6.22	25.58	97.50	56.88	7.76	32.84	88.25	54.69
	CIRCE	**28.03**	**62.47**	**58.56**	**79.33**	**26.23**	**53.88**	**71.36**	**80.52**	**22.19**	**52.90**	**45.21**	**65.76**	**25.47**	**61.48**	**28.36**	**76.5**
	TITAN	**31.82**	**65.93**	**52.30**	**80.52**	**27.01**	**55.44**	**72.10**	**81.28**	**23.25**	**55.92**	**42.52**	**70.17**	**27.80**	**65.33**	**25.54**	**78.95**
	$\Delta_{\text{SFT}}(\%)$	↑ 33.75	↑ 5.77	↓ 27.62	↑ 6.11	↑ 4.33	↑ 1.37	↓ 60.23	↑ 5.05	↑ 45.86	↑ 61.76	↑ 45.94	↑ 11.72	↑ 48.98	↑ 33.44	↑ 66.07	↑ 5.90

Table 2: Translation performance comparison across Hindi-Urdu and Hindi-Sindhi (Devanagari) under zero-shot (ZS), supervised fine-tuning (SFT), and Direct Preference Optimization (DPO) settings. TITAN consistently outperforms multilingual translation baselines across both language pairs, demonstrating the effectiveness of sparse causal translation circuits for low-resource multilingual translation. All CIRCE and TITAN results use Gemma-2 9B as the underlying base model. **BL**: BLEU, **CF2**: ChrF2, **TER**: Translation Edit Rate, **COM**: COMET.

the strongest baselines. Similar improvements are observed for Hindi–Urdu, demonstrating that restricting optimization to causally identified translation circuits substantially improves low-resource multilingual translation. Notably, the CIRCE-only variant already surpasses several fully optimized baselines despite updating only sparse translation-relevant components. Overall, the results show that TITAN provides more scalable and parameter-efficient alternative to conventional full-model optimization.

For a parameter-controlled comparison, we match LoRA SFT+DPO to TITAN at the same 3.14% trainable-parameter budget. Despite the identical budget, LoRA SFT+DPO achieves 26.37 BLEU, compared with 31.82 BLEU for TITAN (+5.45 BLEU), indicating that the gain comes from circuit-based component selection rather than parameter count.

Generalizability. We evaluate TITAN beyond the low-resource Hindi–Urdu and Hindi–Sindhi settings in two complementary ways. For each evaluation direction, the translation circuit is independently identified using CIRCE on that language pair before targeted optimization. We evaluate TITAN beyond the low-resource Hindi–Urdu and Hindi–Sindhi settings in two complementary ways.

First, we assess high-resource cross-lingual generalization on Hindi–English and English–French. Table 3 reports the results. For Hindi→English, TITAN improves BLEU from 29.33 with DPO to 34.21, while English→Hindi improves from 28.71 to 31.94. On English→French, BLEU increases from 30.58 to 33.57, and French→English from 33.09 to 36.40. Similar improvements are observed in ChrF++ and TER, indicating consistent gains across evaluation metrics.

Second, to test whether the observed behavior extends beyond closely related Indic languages, we evaluate TITAN on two typologically distinct low-resource directions: Hindi→Tamil and English→Irish. Table 4 reports the re-

Setting	Direction	BLEU	chrF++	TER	COMET
Hindi–English					
SFT	Hi→En	27.96	52.36	70.24	73.71
DPO	Hi→En	29.33	54.37	66.35	75.43
CIRCE	Hi→En	33.48	58.11	66.34	76.39
TITAN	Hi→En	**34.21**	**60.04**	**60.22**	**76.72**
SFT	En→Hi	26.08	50.88	70.49	78.15
DPO	En→Hi	28.71	52.34	65.91	79.6
CIRCE	En→Hi	31.03	56.99	65.16	82.36
TITAN	En→Hi	**31.94**	**57.51**	**62.21**	**83.95**
English–French					
SFT	En→Fr	29.64	52.88	60.44	80.01
DPO	En→Fr	30.58	55.24	58.63	80.88
CIRCE	En→Fr	32.43	55.40	59.55	84.20
TITAN	En→Fr	**33.57**	**57.90**	**54.35**	**85.51**
SFT	Fr→En	31.02	54.89	54.17	82.5
DPO	Fr→En	33.09	56.51	49.63	83.41
CIRCE	Fr→En	34.87	55.72	50.13	85.98
TITAN	Fr→En	**36.40**	**59.25**	**46.08**	**86.77**

Table 3: Cross-lingual generalization on unseen high-resource Hindi–English and English–French translation tasks. TITAN consistently outperforms SFT and DPO baselines, using Gemma-2 9B as the base model; CIRCE applies targeted SFT, while TITAN further applies DPO to the same identified components.

sults. TITAN improves BLEU from 23.01 to 26.71 on Hindi→Tamil and from 29.52 to 33.19 on English→Irish over the DPO baseline, corresponding to relative improvements of 16.08% and 12.43%, respectively. The consistent performance ordering of SFT < DPO < CIRCE < TITAN across language pairs further supports the generality of circuit-based optimization beyond the closely related Indic language pairs used for circuit discovery.

6 Behavioral Pattern Analysis

We now investigate the internal behavioral patterns learned by multilingual decoder-only LLMs during low-resource translation. Specifically, we analyze whether TITAN identifies genuinely causal translation circuits, how translation responsibilities are distributed across transformer components, and how multilingual representations evolve across layers. Unless otherwise stated, analyses are performed on the strongest TITAN checkpoint for both Hindi→Urdu and Hindi→Sindhi (Devanagari).

Setting	Direction	BLEU	chrF++	TER	COMET
Hindi–Tamil					
SFT	Hi→Ta	22.61	40.66	81.66	77.20
DPO	Hi→Ta	23.01	42.87	77.20	79.00
CIRCE	Hi→Ta	25.04	49.46	75.80	81.20
TITAN	Hi→Ta	**26.71**	**50.60**	**72.20**	**82.70**
English–Irish					
SFT	En→Ga	26.39	51.71	84.35	70.60
DPO	En→Ga	29.52	53.71	80.26	72.80
CIRCE	En→Ga	30.84	54.42	76.90	74.60
TITAN	En→Ga	**33.19**	**57.82**	**66.70**	**75.70**

Table 4: Performance on additional typologically diverse low-resource translation tasks. TITAN consistently outperforms SFT, DPO, and CIRCE on Hindi–Tamil and English–Irish, demonstrating that the observed gains extend beyond the closely related Indic language pairs used for primary evaluation.

6.1 Causal Validation via Progressive Knockout

We evaluate whether CIRCE-identified components are causally necessary for translation behavior through progressive knockout experiments, where identified attention heads and MLP blocks are sequentially ablated using mean-activation replacement. As shown in Figure 1, ablating CIRCE-identified components produces substantially larger degradation than ablating randomly selected components. To rule out circularity from targeted optimization, we additionally evaluate the same CIRCE-identified components in a general LoRA SFT+DPO model trained without restricting optimization to these components. Ablating eight CIRCE-identified components reduces BLEU from 25.53 to 16.54 (35.2%), whereas ablating eight randomly selected components reduces BLEU only from 25.53 to 25.41 (0.45%). For Hindi→Sindhi, removing a single key MLP reduces translation accuracy from 100% to 54.4%, whereas random MLP ablation causes only minor degradation. We further observe that attention-head ablation leads to gradual degradation, indicating distributed routing behavior, while MLP ablation causes sharper collapse, suggesting concentrated semantic transformation functions. Similar trends for Hindi→Urdu are provided in Appendix Section C.

6.2 Attention Routing Behavior

We next analyze the most causally important attention heads identified by CIRCE to understand how

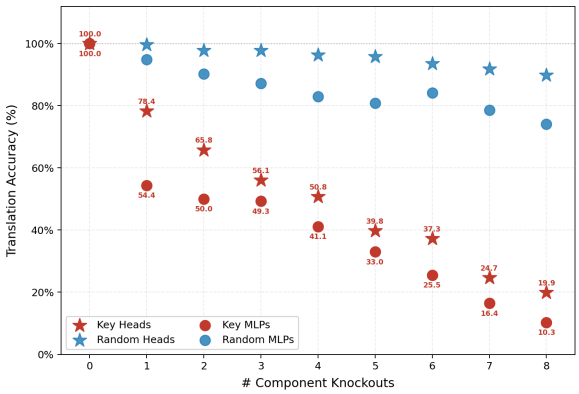

Figure 1: Progressive knockout analysis for Hindi→Sindhi translation. CIRCE-identified components produce substantially larger degradation than randomly selected components.

multilingual translation routing is implemented. As shown in Appendix Figures 3 and 4, both translation directions exhibit a consistent routing pipeline consisting of structural anchoring, source-language identification, translation-direction routing, and final target-language commitment.

Early-layer heads primarily attend to formatting boundaries and language identifiers, whereas later layers shift toward translation-direction routing and target-language commitment. Unlike prior English-centric analyses where attention heads predominantly align to source content tokens, the discovered Indic translation circuits rely heavily on explicit language-label anchoring. We attribute this behavior to the strong lexical and subword overlap between closely related Indic languages, where language identifiers provide more discriminative routing signals than raw lexical content. Additional head visualizations and layer-wise routing analyses are provided in Appendix Section D.

6.3 MLP Representation Dynamics

To analyze semantic processing within the translation circuit, we probe MLP hidden states using cosine similarity with source-language, target-language, and translation-indicator token embeddings. As shown in Appendix Figures 5 and 6, source-language alignment gradually increases through middle layers, whereas target-language alignment emerges primarily in the final layers, indicating delayed target-language realization during multilingual translation. ; We further observe consistently negative English alignment across all layers for both translation directions, contradicting previously reported English-centric pivot mechanisms. Instead, dominant alignment emerges along shared Indic semantic directions, suggesting that closely related Indic translation operates through a shared semantic subspace rather than an explicit pivot language. Additional layer-wise analyses are provided in Appendix Section E.

6.4 Sparse Feature Behavior

Finally, we analyze Sparse Autoencoder (SAE) features extracted from the residual stream to study how translation-relevant representations are distributed across layers. As shown in Appendix Figures 7 and 8, both translation directions exhibit strongly layer-localized feature distributions, where target-language-specific features dominate early layers while Hindi-specific features emerge more prominently in later layers.

Importantly, highly selective features do not necessarily exhibit strong causal influence. Instead, translation behavior is governed by a sparse subset of high-lift features distributed across both source and target language populations. Together with the attention and MLP analyses, these findings suggest that multilingual translation emerges through sparse routing circuits combined with shared semantic computation rather than through explicit pivot-language translation. Additional feature-level analyses are provided in Appendix Section F.

7 Discussion

Expert Assessment. To complement automatic metrics, we conduct human evaluation using native bilingual experts proficient in Hindi–Urdu and Hindi–Sindhi (Devanagari). Experts independently evaluate randomly sampled translations from the SFT and DPO stages of both Gemma-2 baselines and the TITAN pipeline using adequacy and fluency scores on a 1–5 scale. Table 5 reports the percentage of ratings ≥ 4. Across both language pairs, expert judgements consistently confirm the effectiveness of TITAN over full-parameter baselines. For Hi–Ur, the final TITAN configuration improves adequacy from 36.7% to 76.7% and fluency from 53.3% to 86.7% relative to the Gemma-2 DPO baseline. Similar gains are already visible at the CIRCE-guided SFT stage, indicating that improvements arise from sparse-circuit optimization rather than DPO alone.

Hi–Sd exhibits the largest qualitative improvements, with TITAN achieving 93.5% adequacy and 61.3% fluency compared to 51.3% and 25.8% for

Direction	System	Adequacy	Fluency
Hindi → Urdu			
Gemma-2 (SFT)	Baseline	33.3	20.0
Gemma-2 (DPO)	Baseline	36.7	53.3
CIRCE	Ours	56.7	56.7
TITAN	Ours	**76.7**	**86.7**
Hindi → Sindhi			
Gemma-2 (SFT)	Baseline	45.2	12.9
Gemma-2 (DPO)	Baseline	51.3	25.8
CIRCE	Ours	64.8	39.4
TITAN	Ours	**93.5**	**61.3**

Table 5: Human expert evaluation (% of ratings ≥ 4 on a 1–5 scale) for adequacy and fluency across Hindi–Urdu and Hindi–Sindhi translation tasks. Best results per direction are highlighted in bold.

the Gemma-2 DPO baseline. Experts further observed that TITAN better preserves semantic meaning and produces more morphologically appropriate translations for technical terminology.

Several limitations remain. Sindhi outputs occasionally retain untranslated Hindi loanwords and exhibit minor Unicode inconsistencies around implosive phoneme characters. Performance also degrades on long compound sentences with multiple embedded clauses, where fluency decreases despite largely preserved adequacy.

Error Analysis. Table 2 reveals a consistent asymmetry in TITAN's residual errors. Across all evaluation settings, TITAN underperforms the strongest baseline in only two cases, both involving TER for Hindi–Urdu translation directions. Despite higher TER values, TITAN simultaneously achieves stronger BLEU, ChrF2, and COMET scores on the same directions, indicating that the discovered translation circuits prioritize semantic fidelity over minimum-edit lexical reconstruction. For example, on U→H, TITAN improves BLEU from 25.78 to 27.01 and COMET from 77.37 to 81.28, while producing higher TER relative to the reference translation. Error patterns are strongest for Urdu-as-source directions, whereas the largest gains emerge for shared-script Hindi–Sindhi translation (+45.86% H→S and +48.98% S→H). This suggests that sparse circuit transfer is more effective under shared orthographic structure and becomes more challenging under Devanagari–Nastaliq script conversion.

8 Conclusion

We introduced TITAN, a mechanistic interpretability-guided framework for parameter-efficient multilingual translation that identifies and optimizes sparse translation-crucial circuits instead of full-parameter adaptation. Experiments on Hindi–Urdu and Hindi–Sindhi show substantial improvements over multilingual baselines while updating fewer than 5% of parameters. Analyses further suggest that low-resource multilingual translation relies on sparse structured circuits within a shared semantic subspace rather than explicit English-centric pivot representations.

9 Ethics and Limitations

While TITAN performs strongly across low-resource and typologically diverse settings, broader validation across additional languages, scripts, and domains remains necessary. Although sparse circuit optimization reduces trainable parameters, causal intervention and sparse feature decomposition remain computationally expensive during circuit discovery. Occasional errors include untranslated loanwords, Unicode inconsistencies, and reduced fluency for long compound sentences. The current analysis also focuses on the identified translation circuits rather than the model's complete multilingual representation space, limiting the scope of conclusions about global multilingual organization. TITAN does not guarantee factually or culturally perfect translations, particularly for rare expressions or specialized domains, and may inherit biases from its underlying pretraining and translation data. Our causal findings should therefore be interpreted within the evaluated models and language settings rather than as universal translation circuits. Further evaluation with broader domains, independent annotators, and additional language families would strengthen these conclusions. All datasets are publicly available research corpora, no personally identifiable information was collected, and human evaluation was conducted by bilingual experts. Real-world deployment should include appropriate human oversight and linguistic validation.

Acknowledgments

The work is partially supported by a research project COIL-D at IIIT Delhi, funded by MeitY, Govt of India. The authors also acknowledge the support of the Infosys Foundation through Center of AI (CAI) at IIIT-Delhi.

References

Duarte M. Alves, José Pombal, Nuno M. Guerreiro, Pedro H. Martins, João Alves, Amin Farajian, Ben Peters, Ricardo Rei, Patrick Fernandes, Sweta Agrawal, Pierre Colombo, José G. C. de Souza, and André F. T. Martins. 2024. Tower: An open multilingual large language model for translation-related tasks. *Preprint*, arXiv:2402.17733.

Sudhansu Bala Das, Divyajyoti Panda, Tapas Kumar Mishra, Bidyut Kr. Patra, and Asif Ekbal. 2024. Multilingual neural machine translation for indic to indic languages. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 23(5).

Zhe Cao, Zhi Qu, Hidetaka Kamigaito, and Taro Watanabe. 2024. Exploring intrinsic language-specific subspaces in fine-tuning multilingual neural machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21142–21157.

Tianyu Dong, Bo Li, Jinsong Liu, Shaolin Zhu, and Deyi Xiong. 2025. Mlas-lora: language-aware parameters detection and lora-based knowledge transfer for multilingual machine translation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15645–15660.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Xiao Liang, Yen-Min Jasmina Khaw, Soung-Yue Liew, Tien-Ping Tan, and Donghong Qin. 2025. Toward low-resource languages machine translation: a language-specific fine-tuning with lora for specialized large language models. *IEEE Access*, 13:46616–46626.

Abhinav P M, Ketaki Shetye, and Parameswari Krishnamurthy. 2024. MTNLP-IIITH: Machine translation for low-resource Indic languages. In *Proceedings of the Ninth Conference on Machine Translation*, pages 751–755, Miami, Florida, USA. Association for Computational Linguistics.

Sarfraz Raza, FZ Agha, and R Usman. 2004. Phonemic inventory of sindhi and acoustic analysis of voiced implosives. *Center for Research in Urdu language Processing (CRULP)*.

Karumuri Venkata Subbarao. 2012. *South Asian languages: A syntactic typology*. Cambridge University Press. A modern structural overview focusing on the high degree of syntactic convergence, grammar structures, and parametric variation across South Asian language families.

Chenxi Wang, Yixuan Zhang, Ruiji Yu, Yufei Zheng, Lang Gao, Zirui Song, Zixiang Xu, Gus Xia, Huishuai Zhang, Dongyan Zhao, and 1 others. 2025. Do llms" feel"? emotion circuits discovery and control. *arXiv preprint arXiv:2510.11328*.

Bin Wei, Zheng Jiawei, Zongyao Li, Zhanglin Wu, Jiaxin Guo, Daimeng Wei, Zhiqiang Rao, Shaojun Li, Yuanchang Luo, Hengchao Shang, Jinlong Yang, Yuhao Xie, and Hao Yang. 2024. Machine translation advancements of low-resource Indian languages by transfer learning. In *Proceedings of the Ninth Conference on Machine Translation*, pages 775–780, Miami, Florida, USA. Association for Computational Linguistics.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. *Preprint*, arXiv:2309.11674.

Hongbin Zhang, Kehai Chen, Xuefeng Bai, Xiucheng Li, Yang Xiang, and Min Zhang. 2026. Exploring the translation mechanism of large language models. *Preprint*, arXiv:2502.11806.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. *Preprint*, arXiv:2304.04675.

Appendix

A Translation Task Templates and Examples

A.1 Dataset Construction

We construct contrastive translation datasets for two directions: Hindi→Sindhi (Hi→Sd, Devanagari) and Hindi→Urdu (Hi→Ur). The two directions dissociate factors that are typically conflated in multilingual circuit analysis. Hi→Sd involves no script boundary, thereby isolating cross-lingual semantic mapping from orthographic transduction while preserving a non-trivial transfer challenge through Sindhi-specific implosives and Persian–Arabic lexical borrowing. In contrast, Hi→Ur crosses a script boundary (Devanagari → Perso-Arabic Nastaliq) and, given the strong grammatical overlap between Hindi and Urdu, tests whether the identified circuit encodes semantic content independently of surface orthography.

Contrastive prompts are sampled from the SFT splits. We sample 10K pairs per direction.

A.2 Prompt Format

The general template, introduced in Section 4.1 (Eq. 1), is:

$$X^+ = \texttt{[SRC]: [text]} \rightarrow \texttt{[TGT]:} \tag{12}$$

where `[SRC]` and `[TGT]` are language-name tokens, and the trailing colon serves as the generation trigger and query position for attention analyses. We use the arrow glyph → because it tokenizes as a single token under the Gemma-2 SentencePiece tokenizer. By contrast, the hyphen (`-`) was empirically verified to fragment at the directive position, destabilizing query-position alignment.

A.3 Contrastive Pair Construction

Each source sentence s yields one positive prompt X^+ and four counterfactual prompts $\{X_k^-\}_{k=1}^4$, differing from X^+ only in the directive region. Construction follows two principles:

1. **Structural isomorphism**: identical token-position layout, punctuation, and source-sentence span.

2. **Targeted lexical substitution**: a single semantic component of the directive is replaced.

Each perturbation type is instantiated with three lexical variants (e.g., Target Nullification: `Nothing` / `Null` / `Void`); reported results are averaged across variants.

Table 6 illustrates the four perturbation types with representative examples from both translation directions.

A.4 Validation Protocol

Templates are verified along two axes: *surface fluency* and *task-level disruption*. Because the four perturbations alter the directive at different structural depths, a uniform fluency threshold is inappropriate; accordingly, we apply differentiated validation criteria.

Fluency-preserving perturbations (Semantic Obfuscation and Target Nullification) substitute a single target-slot token and are required to preserve surface fluency. Measured perplexity deviations relative to positive prompts confirm that these perturbations alter directive semantics without disrupting grammatical structure.

Structurally invasive perturbations (Paradox Insertion and Action Distortion) modify the directive at deeper structural levels and are therefore expected to induce higher perplexity by design. For these perturbations, validity is evaluated through task disruption rather than a fluency threshold.

For task-level disruption, all four perturbation types are required to yield zero translation accuracy on a held-out evaluation set, together with strong inter-annotator agreement between two annotators regarding whether each counterfactual successfully removes the translation directive. Templates failing the task-disruption criterion are discarded.

A.5 Token Taxonomy

Each token position is assigned one of four functional roles for attention and SAE analyses (Section 4.2).
IND tokens form the circuit's structural scaffold and draw spike attention from Indicator Heads. SRC tokens are the primary attention targets for Source Heads, while the TGT position serves as the readout site for logit-difference and unembedding analyses. For Hi→Sd, samples containing Sindhi-specific implosive phonemes are additionally tagged at the SRC level to support phonological circuit analysis.

Perturbation Type	Direction	X^+ / X^- Example	Ablated Component
Target Nullification	Hi→Sd	`Hindi: ... → Sindhi:` `Hindi: ... → Nothing:`	Target language identity
	Hi→Ur	`Hindi: ... → Urdu:` `Hindi: ... → Nothing:`	Target language identity
Action Distortion	Hi→Sd	`Hindi: ... → Sindhi:` `Eat the following: ... → Sindhi:`	Source-side instruction frame
	Hi→Ur	`Hindi: ... → Urdu:` `Eat the following: ... → Urdu:`	Source-side instruction frame
Semantic Obfuscation	Hi→Sd	`Hindi: ... → Sindhi:` `Hindi: ... → Color:`	Translation-task semantics
	Hi→Ur	`Hindi: ... → Urdu:` `Hindi: ... → Color:`	Translation-task semantics
Paradox Insertion	Hi→Sd	`Hindi: ... → Sindhi:` `Hindi: ... → Silent Rock:`	Coherent target slot
	Hi→Ur	`Hindi: ... → Urdu:` `Hindi: ... → Silent Rock:`	Coherent target slot

Table 6: Contrastive perturbation taxonomy with representative examples. Each X^- alters a single semantic component of the directive while preserving structural isomorphism with X^+.

Role	Tokens	Function
IND (Indicator)	`Hindi`, `Sindhi`, `Urdu`, →, `:`	Translation direction and structural scaffold
SRC (Source)	Content tokens of the source sentence	Semantic payload
TGT (Target)	First generated token(s) at query position	Translation output signal
RAND (Random)	Position-shuffled baseline	Control for positional effects

Table 7: Token role definitions used in attention and SAE analyses.

B Experimental Setup

All Baseline fine-tuned models use identical training splits with LoRA rank 16 and batch size 8, with Gemma-2 9B serving as the underlying base model for CIRCE and `TITAN`. SFT models are trained for 3 epochs, while DPO optimization is performed for 2 epochs on NVIDIA A100 40GB GPUs. `TITAN` further incorporates Sparse Autoencoder decomposition with latent dimension 14336 for translation-specific feature discovery prior to targeted optimization.

C Causal Validation via Progressive Knockout

In addition to the Hindi→Sindhi results presented in the main paper, Figure 2 presents the corresponding progressive knockout analysis for Hindi→Urdu translation. Across both translation directions, ablating CIRCE-identified components consistently produces substantially larger degradation than ablating randomly selected attention heads and MLP blocks, confirming that `TITAN` identifies causally meaningful translation circuits rather than incidentally activated neurons.

For Hindi→Urdu, removing the first key MLP reduces translation accuracy from 100% to approximately 87%, whereas random MLP ablation causes only marginal degradation. Similar to Hindi→Sindhi, attention-head ablation results in gradual performance decline, indicating distributed routing behavior across multiple heads, while MLP ablation causes sharper early-stage collapse, suggesting concentrated semantic transformation functions.

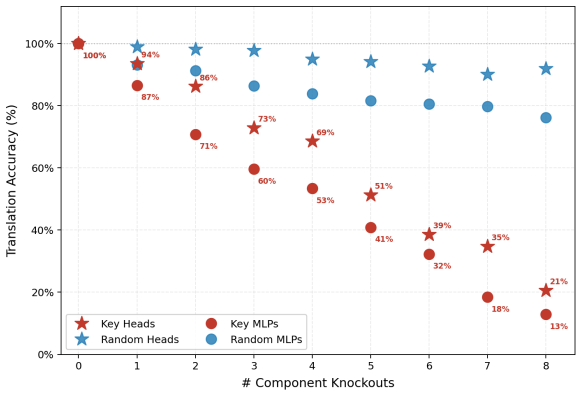

Figure 2: Progressive knockout analysis for Hindi→Urdu translation. CIRCE-identified components produce substantially larger degradation than randomly selected components.

Across both language pairs, the gap between CIRCE-identified and random components widens monotonically as additional components are removed. Even after multiple interventions, randomly selected components preserve most translation capability, whereas CIRCE-identified circuits rapidly collapse toward near-chance performance. Together, these findings further support that multilingual translation behavior is highly localized within sparse transformer subcircuits.

D Attention Routing Behavior

Figures 3 and 4 present the dominant CIRCE-identified attention heads for Hindi→Sindhi and Hindi→Urdu translation respectively. Across both language pairs, we observe a highly structured routing pipeline in which early-layer heads primarily attend to prompt separators and formatting boundaries, followed by mid-layer heads specializing in source-language identification through explicit language labels. Later layers progressively shift toward translation-direction routing and final target-language commitment.

In both translation directions, the strongest causal heads emerge near the final generation layers, particularly around the terminal translation commitment stage. Notably, several heads consistently attend to language-indicator tokens rather than source content tokens, suggesting that routing decisions are strongly conditioned on explicit language identifiers. This behavior differs substantially from previously reported English-centric translation circuits, where attention heads predominantly align to source semantic content.

Although both translation directions exhibit similar routing structure, the Hindi→Urdu circuit displays broader and more distributed mid-layer routing behavior compared to Hindi→Sindhi. We attribute this difference to the additional script-conversion complexity introduced by Urdu Nastaliq generation. Overall, these findings suggest that low-resource Indic translation relies heavily on label-aware routing mechanisms operating at prompt boundaries before semantic transformation occurs within downstream MLP representations.

E MLP Representation Dynamics

Figures 5 and 6 present the layer-wise MLP representation dynamics for Hindi→Sindhi and Hindi→Urdu translation respectively. To analyze semantic processing within the translation circuit, we project MLP hidden states onto source-language, target-language, and translation-indicator token embeddings using cosine similarity.

Across both translation directions, source-language alignment gradually increases through middle layers and peaks near the later network stages, indicating progressive encoding of source-language semantics within MLP representations. In contrast, target-language alignment remains weak for most layers before sharply increasing near the final generation layers, suggesting that multilingual translation postpones target-language realization until late-stage decoding.

We further analyze whether translation proceeds through an implicit English pivot, as suggested in prior multilingual interpretability studies. Projecting MLP transformation vectors onto English, Hindi, Urdu, and Sindhi representation directions reveals consistently negative English alignment across nearly all layers for both language pairs. This strongly contradicts previously reported English-centric multilingual pivot mechanisms.

Instead, the dominant latent alignment emerges along shared Indic semantic directions, particularly Hindi-centered representations, suggesting that closely related Indic translation directions operate through a shared Hindustani semantic subspace rather than through any explicit pivot language. In this shared latent space, semantic computation becomes approximately orthogonal to explicit language identifiers, enabling language-neutral processing prior to final target-language commitment.

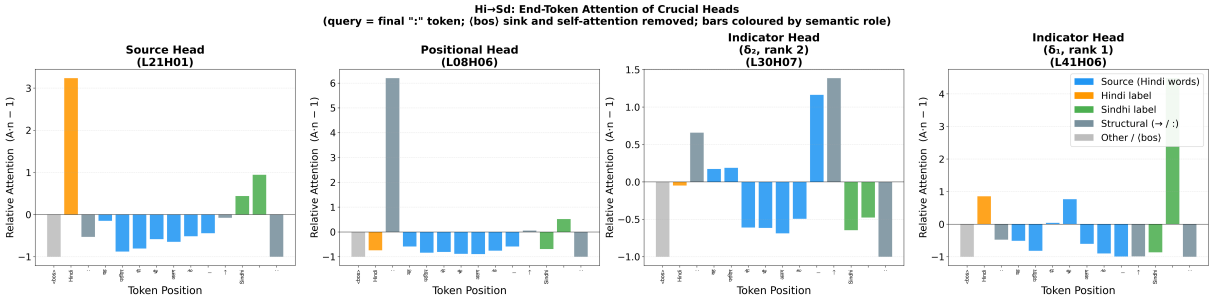

Figure 3: Dominant CIRCE-identified attention heads for Hindi→Sindhi translation. Different heads specialize in structural anchoring, source-language identification, translation-direction routing, and target-language commitment.

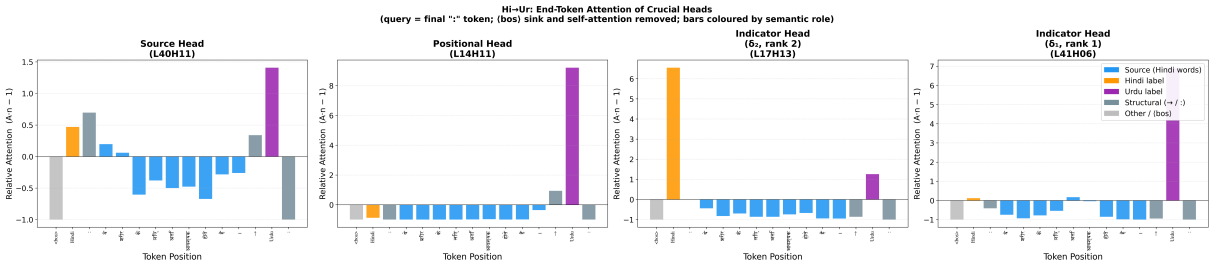

Figure 4: Dominant CIRCE-identified attention heads for Hindi→Urdu translation. Compared to Hindi→Sindhi, the Urdu circuit exhibits broader mid-layer routing behavior associated with script-conversion complexity.

F Sparse Feature Behavior

Figures 7 and 8 present the layer-wise Sparse Autoencoder (SAE) feature distributions for Hindi→Sindhi and Hindi→Urdu translation respectively. To analyze translation-relevant representations, we classify SAE features jointly using selectivity and causal-lift statistics in order to distinguish routing-oriented features from causally important translation features.

Across both language pairs, we observe strongly layer-localized feature distributions in which target-language-specific features dominate earlier layers while Hindi-specific features emerge more prominently in later layers. Between these regions lies a middle-layer transition zone where neither language exhibits strong dominance, consistent with the shared latent-space behavior observed in the MLP analysis.

Importantly, highly selective features do not necessarily exhibit strong causal influence. Most language-specific features produce minimal causal lift despite strong selectivity scores, indicating that activation specialization alone is insufficient to explain translation behavior. Instead, translation is governed by a sparse subset of high-lift features distributed across both source and target language populations.

Combining the SAE findings with the attention and MLP analyses yields a unified mechanistic interpretation of low-resource Indic translation. Attention heads primarily implement language-aware routing and prompt-boundary coordination, whereas MLP layers perform semantic transformation within a shared Hindustani latent subspace. Translation behavior therefore emerges through sparse routing circuits coupled with language-neutral semantic computation rather than through explicit pivot-language translation.

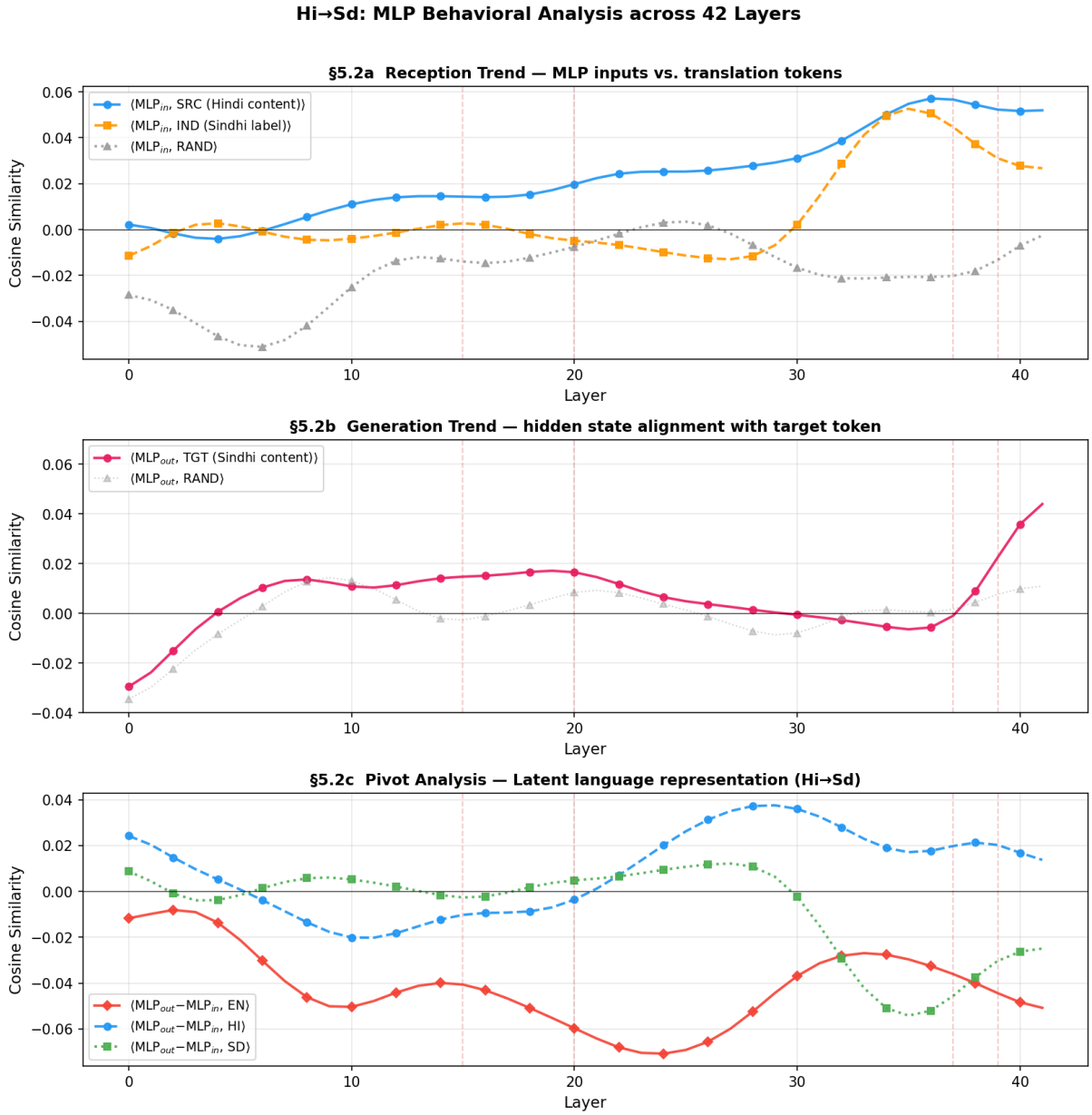

Figure 5: MLP representation dynamics for Hindi→Sindhi translation. Top: source-language reception behavior. Middle: target-language generation behavior. Bottom: latent pivot analysis across Hindi, Sindhi, and English representation directions.

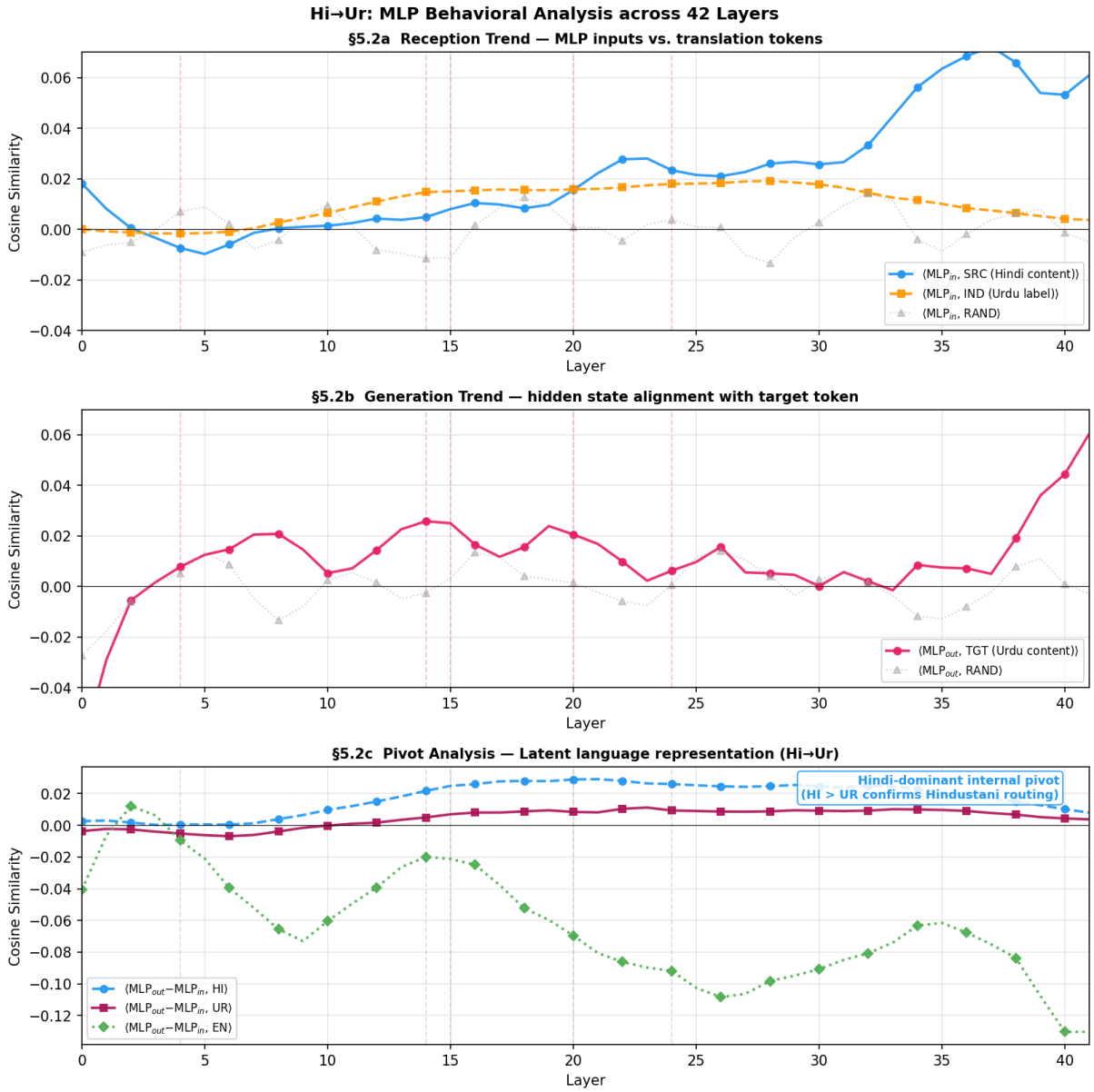

Figure 6: MLP representation dynamics for Hindi→Urdu translation. Similar to Hindi→Sindhi, translation behavior evolves through shared Indic semantic directions rather than explicit English pivots.

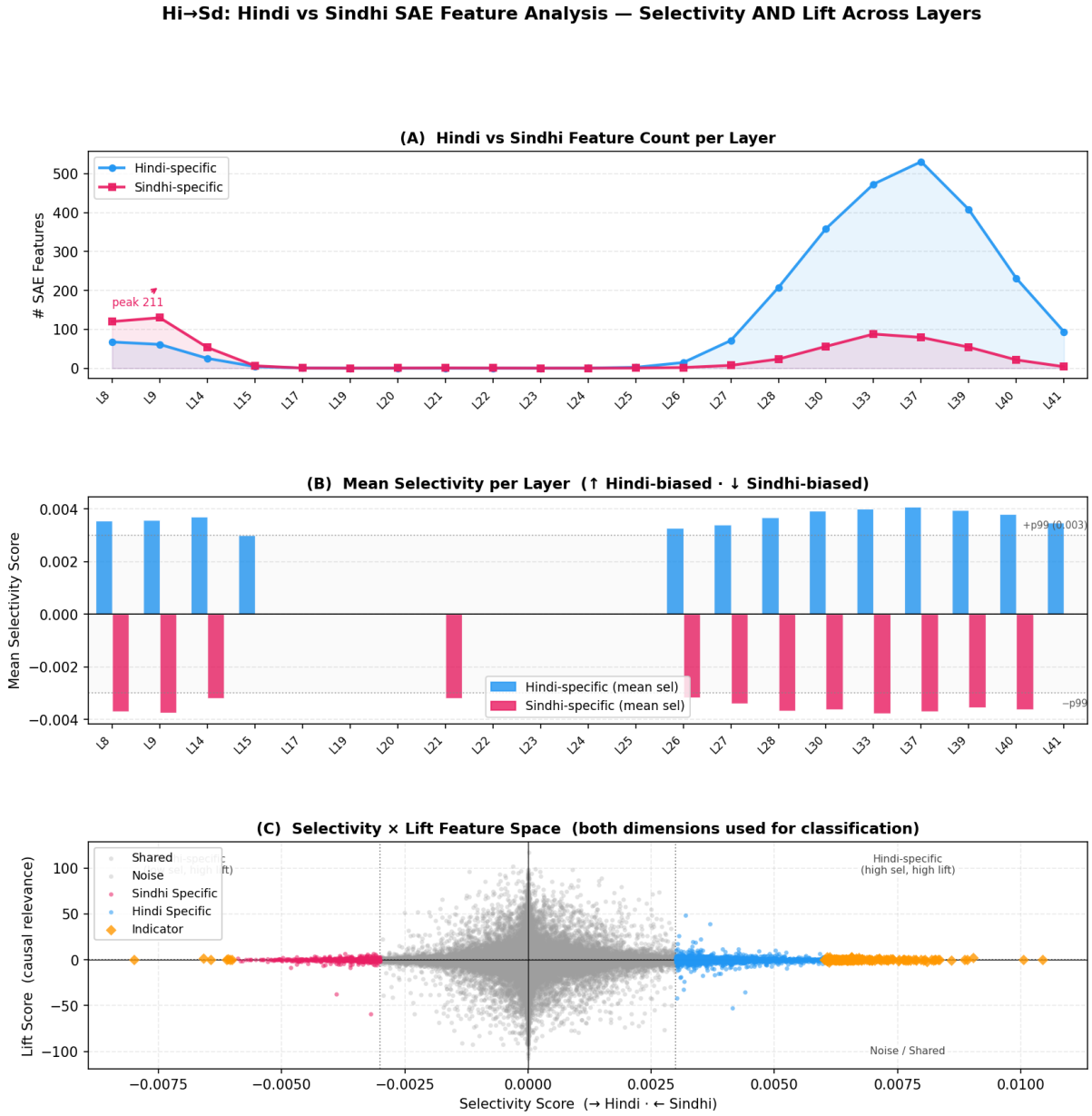

Figure 7: Sparse feature analysis for Hindi→Sindhi translation. Top: layer-wise distribution of Hindi-specific and Sindhi-specific SAE features. Middle: mean feature selectivity across layers. Bottom: joint selectivity–lift feature space highlighting sparse causally relevant translation features.

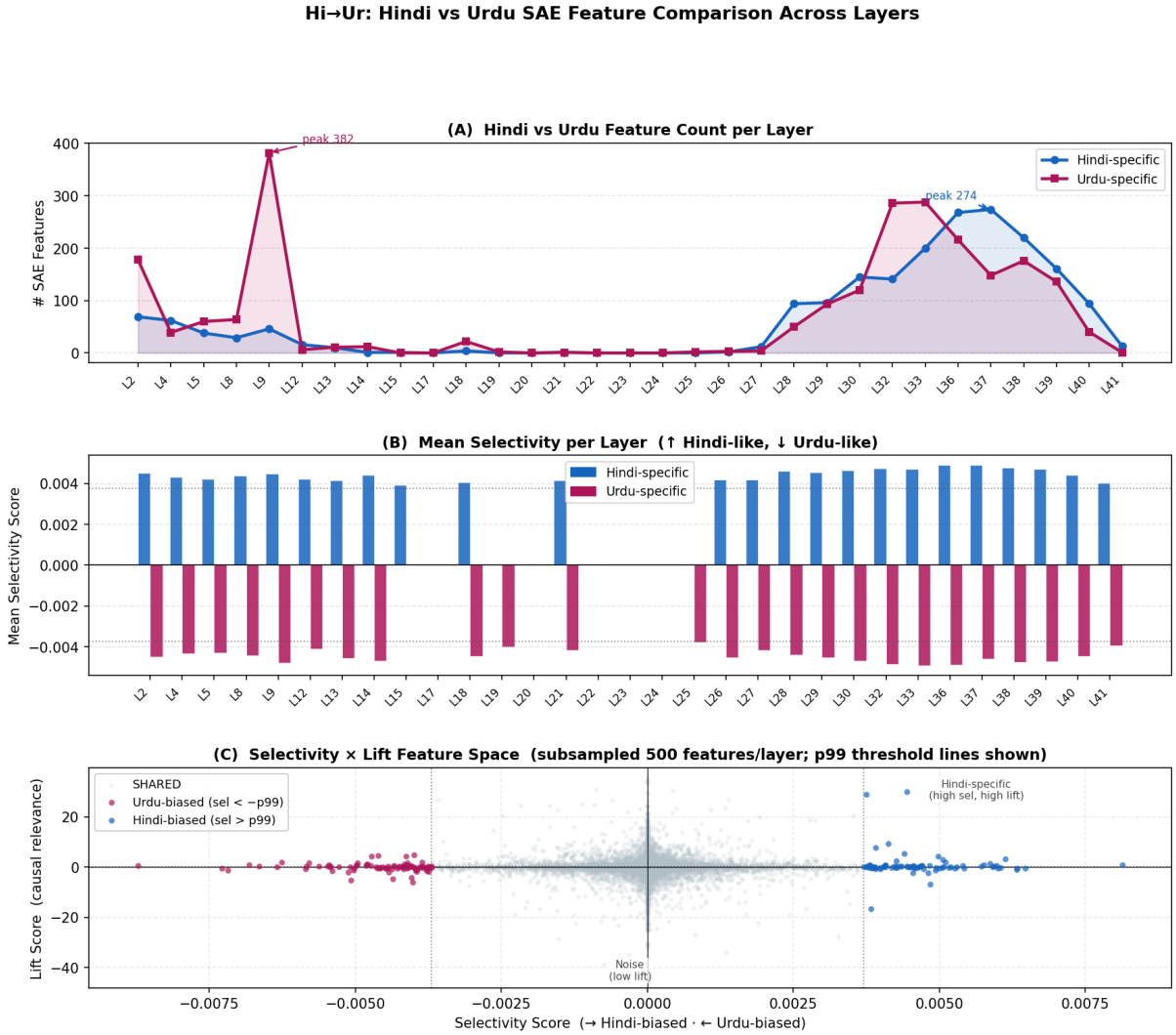

Figure 8: Sparse feature analysis for Hindi→Urdu translation. Similar to Hindi→Sindhi, causally important translation behavior is concentrated within a sparse subset of high-lift features.