Code-Switching Reveals Language Anchoring in Multilingual LLMs

Anonymous ACL submission

Abstract

Multilingual Large Language Models (MLLMs) are increasingly expected to handle Code-Switched (CS) inputs, yet mixing languages frequently degrades performance relative to source- or target-language monolingual counterparts. To understand this degradation, we use grammar-forced CS as a controlled diagnostic setting for locating CS representations relative to their source and target counterparts. We introduce **Anchor Bias**, a geometric measure that quantifies *language anchoring*—whether a CS hidden state aligns closer to its source or target language counterpart. Across diverse MLLMs, Anchor Bias reveals a consistent grammar-frame effect: source-framed CS stays source-anchored, whereas target-framed CS shifts target-ward and shows larger Question Answering (QA) degradation. Motivated by this representational pattern, we propose **CANVAS** (**C**ontextual **A**nchor-based **N**eural **V**ector **A**lignment **S**teering), an inference-time intervention that extracts a source-side canvas from the input and softly steers target-language hidden states toward the source anchor during prefill. CANVAS consistently recovers QA F1 across MLLMs and CS conditions, showing that internal anchoring signals provide an actionable target for mitigating CS inference failures.

1 Introduction

Multilingual Large Language Models (MLLMs) are increasingly expected to handle Code-Switched (CS) inputs, where users mix multiple languages within an interaction. However, standard evaluations (Hu et al., 2020; Liang et al., 2020) largely assume monolingual contexts, overlooking this pervasive practice. Crucially, CS is not random lexical noise but a structured linguistic phenomenon governed by grammatical organization (Doğruöz et al., 2021; Winata et al., 2023). Consequently, an MLLM competent on monolingual inputs can still face unique processing failures when an information need is expressed through a structured mixed-language form, frequently degrading performance compared to monolingual counterparts.

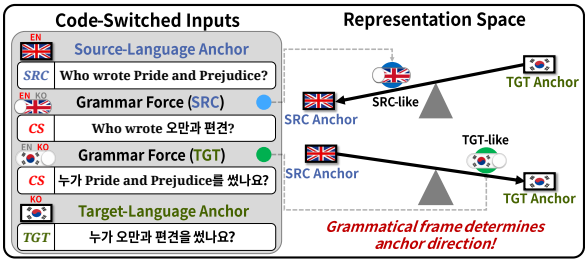

Figure 1: **Locating code-switched inputs in the multilingual representation space.** Source- and target-framed CS variants preserve different grammatical structures for the same information need, plotted alongside their matched monolingual anchors.

Prior work evaluates CS across robustness, translation, generation, and multilingual generalization (Laureano De Leon et al., 2024; Huzaifah et al., 2024; Abdaljalil et al., 2025; Mohamed et al., 2025; Heredia et al., 2025; Winata et al., 2026). While these studies show that mixed-language inputs expose fragilities hidden by monolingual benchmarks, and existing MLLM analyses suggest latent language preferences or English-centered processing pathways (Wendler et al., 2024; Zhao et al., 2024; Sharma et al., 2025), neither explains how a CS instance is geometrically positioned relative to its matched monolingual counterparts.

Understanding the representational pattern behind this performance drop requires mapping where the mixed-language hidden state lies relative to its monolingual references: source-aligned, target-shifted, or intermediate. Since identical QA failures can stem from different internal representational causes, we probe the geometric layout of the model's representation space rather than rely-

ing only on answer-level analysis.

To this end, we study code-switched QA in a controlled setting designed for representation-level diagnosis. For each CS question, we use semantically matched source- and target-language counterparts as monolingual references. Following CodeMixQA (Winata et al., 2026), we leverage grammar-forced CS variants to isolate the *grammatical frame*. By holding the question fixed while isolating which language organizes the utterance, we test whether the CS state stays between monolingual references or moves toward the frame language's representation.

Figure 1 illustrates this framework with an English–Korean example. A source-framed question (e.g., *"Who wrote 오만과 편견?"*) embeds target lexical items into a source syntactic backbone, while a target-framed question (e.g., *"누가 Pride and Prejudice를 썼나요?"*) does the reverse. Together with the matched source-only and target-only questions, which serve as the monolingual anchors, this paired construction isolates the frame variable and reveals which monolingual subsystem the MLLM internally follows when interpreting the same cross-lingual intent.

To quantify this representation-level phenomenon, we introduce **Anchor Bias**, a geometric measure assessing whether a CS hidden state aligns closer to its source- or target-language counterpart. Our analysis across diverse MLLMs reveals a consistent, frame-dependent pattern: source-framed CS remains close to the source anchor, whereas target-framed CS exhibits a sharp representational shift target-ward, coinciding with larger performance drops. This suggests that CS inputs tend to degrade more when their representations move away from the source-side trajectory that better preserves answer quality.

Motivated by these geometric findings, we propose **CANVAS** (**C**ontextual **A**nchor-based **N**eural **V**ector **A**lignment **S**teering) to test whether a source-ward representation correction can mitigate this shift via an inference-time intervention. CANVAS extracts a source-side canvas from the in-context input and softly steers target-language hidden states toward that canvas during prefill.

Across MLLMs and CS conditions, CANVAS consistently improves QA F1 without parameter updates. Its largest gains appear under target-framed CS, suggesting that source-canvas alignment can mitigate the target-ward anchoring pattern identified by our analysis, a pattern that reflects realistic bilingual usage such as native-language sentence frames with embedded English technical terms (Bali et al., 2014; Winata et al., 2023; Yong et al., 2023).

2 Experimental Setup

2.1 Experimental Framework

Task setup. We analyze how MLLMs internally process mixed-language intent using code-switched QA as a controlled setting. For a given information need, we evaluate a matched quadrant of question variants ($\mathcal{Q}(x)$) across a source language, a target language, and two distinct CS forms, as illustrated in Figure 1. Following common English–target-language evaluation conventions (Laureano De Leon et al., 2024; Winata et al., 2023, 2026), we treat English as the source language and the paired non-English language as the target; Appendix E tests source-language generalization. Each instance keeps the factual query fixed and varies only the linguistic frame, enabling direct representational and behavioral comparison.

Dataset and Variant Construction. To align our evaluation, we utilize the fact-seeking queries from SimpleQA Verified (Haas et al., 2026) as the source language questions (q^{SRC}) and map them to the corresponding grammar-forced CS pairs from CodeMixQA (Winata et al., 2026). Each pair provides two structural alignments: GF-SRC (English structure with target lexical inserts, q^{SRC}) and GF-TGT (target structure with English lexical inserts, q^{TGT}). For the target-language counterpart (q^{TGT}), we translate the source question via Qwen3-235B-A22B (Yang et al., 2025). This pipeline yields a complete matched quadruplet $\mathcal{Q}(x) = \{q^{\mathrm{SRC}}, q^{\mathrm{TGT}}, q^{\mathrm{GF\text{-}SRC}}, q^{\mathrm{GF\text{-}TGT}}\}$ evaluated against a unified reference answer. We pair q^{SRC} with nine target languages (Bengali, Spanish, French, Hindi, Korean, Marathi, Toba Batak, Urdu, Chinese), evaluating 955 unique source queries across 2,880 matched comparisons (see Appendix F.1 for detailed statistics).

Target-Language Anchors. Because the GF-TGT condition adheres to the target-language syntactic frame, word-level translation largely preserves its surface backbone. Consequently, q^{TGT} serves as a natural target-language anchor for representation-level analysis. Appendix F.2 provides further discussion on alignment.

2.2 Implementation Details

Models. To examine whether language anchoring appears across architectures and model families, we evaluate various MLLMs. We use Aya-Expanse-8B (Dang et al., 2024), Qwen3.5-4B, Qwen3.5-9B, Qwen3.5-27B (Yang et al., 2025), Llama-3.1-70B-Instruct, Llama-3.3-70B-Instruct (Meta AI, 2024), Mixtral-8x7B-Instruct (Jiang et al., 2024), Phi-3.5-MoE-Instruct (Abdin et al., 2024), Qwen3-30B-A3B (Yang et al., 2025), and Qwen3.6-35B-A3B (Yang et al., 2025).

Inference Setups. We use greedy decoding throughout our experiments. Translations are generated with Qwen3-235B-A22B-2507 (Yang et al., 2025) via OpenRouter; all other models in the pool are run locally. Details are reported in Appendix C (with prompt templates in Appendix F.4).

2.3 Evaluation Metrics

Evaluation Metric. The model receives one question variant at a time and generates an answer. We score each generated answer against the SimpleQA reference answer using normalized token-overlap F1. We report F1 per condition and summarize CS degradation as the gap between F1 on q^{SRC} and F1 on q^{CS}.

Internal Hidden Signals. Output F1 establishes whether CS changes answer quality, but it does not reveal how the model internally interprets the mixed-language question. We therefore extract hidden states from the question region for matched source, target, and CS variants and compare the CS representation against both monolingual counterparts. This representation comparison asks where CS lies relative to the two reference language states and forms the basis of the anchor-bias analysis in Section 3.

3 Language Anchoring of Code-Switched Inputs

We investigate where a code-switched hidden state lies in the multilingual representation space relative to its matched monolingual counterparts. Geometrically, if a CS input is processed as a balanced bilingual construction, its representation should occupy an intermediate space. Conversely, if the mixed input adheres to a dominant language processing pathway, its hidden state will skew toward either the source or target anchor. To formally quantify this position, we introduce *anchor bias*, a geometric metric that reveals a consistent, frame-dependent language anchoring pattern across diverse MLLM architectures and transformer layers.

3.1 Formulating Anchor Bias

Metric Derivation. For a matched question set, let $q_s = q^{\mathrm{SRC}}$, $q_t = q^{\mathrm{TGT}}$, and $q_c = q^{\mathrm{CS}}$. For a model M and question variant q, let $H_M(q) \in \mathbb{R}^{L\times T\times d}$ denote the hidden states over layers, tokens, and dimensions. We mean-pool the question-content token span $C(q)$ (excluding shared instruction and template markers) to obtain the layer-wise context vector $\mathbf{r}_\ell(q) \in \mathbb{R}^d$:

$$\mathbf{r}_\ell(q) = \frac{1}{|C(q)|} \sum_{t\in C(q)} H_M(q)[\ell, t, :]. \quad (1)$$

We define the layer-wise similarity as $s_\ell(q_i, q_j) = \cos(\mathbf{r}_\ell(q_i), \mathbf{r}_\ell(q_j))$. The raw layer-wise anchor bias of a CS question is then defined as $\mathrm{AB}_\ell(q_c) = s_\ell(q_s, q_c) - s_\ell(q_t, q_c)$, where positive values indicate closer proximity to the source anchor and negative values signal a target-ward skew.

Anisotropy Normalization. Raw cosine similarities are not directly comparable across model families due to shifting background anisotropies and scale variations in multilingual hidden spaces (Ethayarajh, 2019; Timkey and van Schijndel, 2021). In highly anisotropic spaces, unrelated input representations can exhibit artificially high baseline similarity, obscuring relative cross-lingual distances (see Appendix G.1 for baseline diagnostics). We counteract this by applying a layer-specific random cosine normalization. We estimate μ_ℓ^M as the mean cosine over random source question pairs with non-overlapping answers, capturing pure directional bias rather than semantic overlap. The normalized similarity is computed as:

$$\tilde{s}_\ell(q_i, q_j) = \frac{s_\ell(q_i, q_j) - \mu_\ell^M}{1 - \mu_\ell^M}. \quad (2)$$

Using this normalized similarity, we define normalized layer-wise anchor bias as $\widetilde{\mathrm{AB}}_\ell(q_c) = \tilde{s}_\ell(q_s, q_c) - \tilde{s}_\ell(q_t, q_c)$. To obtain a stable model-level estimate, we average $\widetilde{\mathrm{AB}}_\ell(q_c)$ over the upper half of transformer layers ($\mathcal{L}_{\mathrm{upper}}$):

$$\mathrm{AB}_{\mathrm{nrm}}^{\mathrm{upper}}(q_c) = \frac{1}{|\mathcal{L}_{\mathrm{upper}}|} \sum_{\ell\in\mathcal{L}_{\mathrm{upper}}} \widetilde{\mathrm{AB}}_\ell(q_c). \quad (3)$$

Model	AB^{upper}_{nrm}			QA behavior (F1)				SRC-anchored layers (%)		
	GF-SRC	GF-TGT	Δ_{AB}	SRC	GF-SRC	GF-TGT	TGT	GF-SRC	GF-TGT	Δ_{layer}
Aya-Expanse-8B	+0.300	-0.220	+0.520	9.3	7.0 (-2.3)	4.4 (-4.9)	2.4	85.6	42.8	+42.8
Qwen3.5-4B	+0.169	-0.262	+0.431	10.0	8.2 (-1.8)	7.2 (-2.8)	4.8	70.7	29.2	+41.5
Mixtral-8x7B	+0.273	-0.452	+0.725	10.8	6.7 (-4.1)	5.8 (-5.0)	4.5	74.6	33.2	+41.4
Qwen3-30B-A3B	+0.393	-0.081	+0.474	11.2	8.8 (-2.4)	6.5 (-4.7)	7.7	78.5	42.1	+36.4
Phi-3.5-MoE	+0.248	-0.300	+0.548	13.1	9.6 (-3.5)	6.5 (-6.6)	3.0	64.8	31.7	+33.1
Qwen3.5-9B	+0.185	-0.256	+0.441	13.1	11.9 (-1.2)	8.1 (-5.0)	8.7	72.9	32.5	+40.4
Qwen3.5-27B	+0.154	-0.197	+0.351	16.6	13.3 (-3.3)	10.4 (-6.2)	11.8	73.5	34.4	+39.1
Llama3.3-70B	+0.267	-0.191	+0.458	18.9	13.7 (-5.2)	11.6 (-7.3)	6.5	81.4	39.5	+41.9
Qwen3.6-35B-A3B	+0.215	-1.011	+1.226	21.4	17.0 (-4.4)	13.5 (-10.3)	11.1	76.2	38.1	+38.1
Llama3.1-70B	+0.257	-0.225	+0.482	23.1	16.9 (-6.2)	13.5 (-9.6)	6.2	80.6	38.0	+42.6
Mean	+0.246	-0.320	+0.566	14.8	11.3 (-3.5)	8.7 (-6.2)	6.7	75.9	36.2	+39.7

Table 1: **Effect of grammatical frame across models.** Upper-layer anchor bias, QA F1 on source, code-switched, and target-language inputs, and the percentage of source-anchored layers. Red values indicate F1 drops relative to SRC; TGT is reported as a monolingual reference.

This restriction follows prior findings that upper layers encode more abstract, task-aligned semantic representations than lower ones (Peters et al., 2018; Tenney et al., 2019), a property extending to modern generative LLMs (Marks and Tegmark, 2024), thereby yielding a single robust anchor-bias score per condition.

3.2 Frame-Dependent Anchoring Analysis

Frame-Dependent Language Anchoring. As shown in Table 1, internal anchor bias consistently follows the input's grammatical backbone rather than lexical mix. GF-SRC (source-framed) exhibits a positive mean anchor bias of $+0.246$, confirming orientation toward the source anchor. Conversely, GF-TGT (target-framed) demonstrates a distinct target-ward shift with a negative mean bias of -0.320. This swing ($\Delta_{AB} = +0.566$) firmly establishes that the grammatical frame is a strong factor shaping the direction of internal language anchoring. Order-control experiments further show that changing the grammatical/order frame while holding the lexical side fixed shifts the representation anchor (Appendix G.5)

Behavioral QA Alignment of Anchor Bias. This frame-dependent representational shift corresponds to QA performance. As shown in Table 1, GF-SRC produces a smaller average F1 degradation relative to the monolingual source input (-3.5) than GF-TGT (-6.2), suggesting that closer proximity to the source anchor tends to preserve task performance better. This behavioral alignment is consistent across individual model–condition pairs, where AB^{upper}_{nrm} correlates positively with ΔF1 ($\rho = +0.485$, $p < 0.01$).

Layer-Wise Robustness. We examine whether this anchoring shift persists across model depth. Specifically, we quantify the percentage of layers exhibiting positive AB^{upper}_{nrm} (i.e., source-anchored layers) for each model and grammar-forced condition. As shown in Table 1, GF-SRC yields a predominantly source-heavy trajectory across 75.9% of layers on average, whereas GF-TGT drops sharply to 36.2%. This substantial gap demonstrates that the structural frame influences internal representation trajectories robustly throughout the model's depth rather than just at the final layers.

Per-layer anchor trajectory. We further visualize how anchor bias evolves across model depth. Table 1 reports normalized upper-layer anchor bias for cross-model comparison, whereas Figure 2 plots raw AB_ℓ at each layer for the two grammar-forced CS conditions: GF-SRC and GF-TGT. We use raw AB_ℓ for this layer-wise visualization because normalization rescales AB_ℓ by the positive factor $1/(1 - \mu_\ell)$: it preserves the sign and within-layer ordering, but can visually over-amplify lower layers when μ_ℓ is close to 1 due to anisotropy. Figure 2 confirms the same frame-dependent pattern observed in Table 1. GF-SRC stays mostly source-oriented, while GF-TGT shifts target-ward. This contrast is small in early layers and becomes clearer toward the top of the model, with the strongest target-ward anchoring concentrated in the final 30% of layers. This depth profile motivates applying intervention at later layers, where the source–target anchoring contrast is more pronounced, rather than during earlier representation formation.

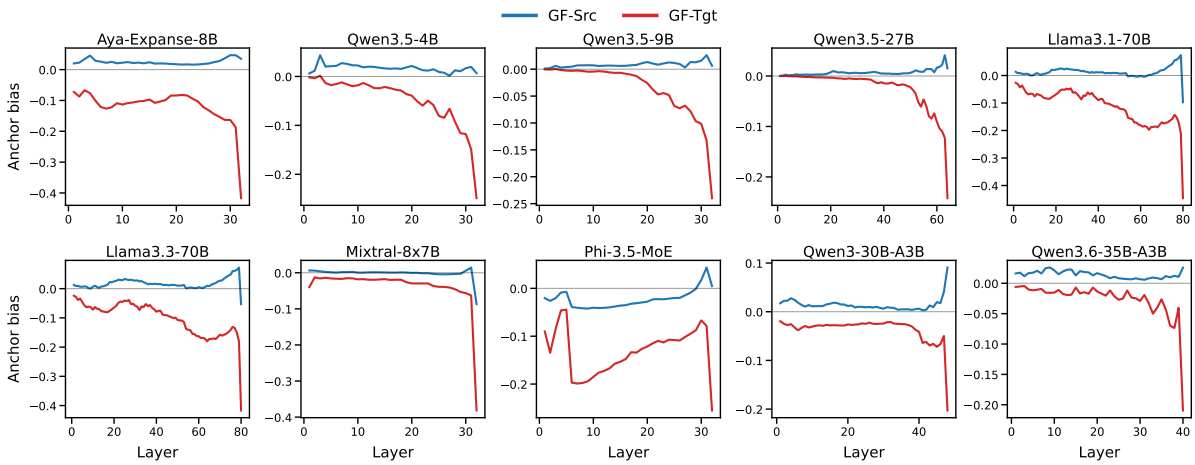

Figure 2: **Per-layer anchor bias across model depth.** Raw AB_ℓ on question-content tokens, averaged across examples, plotted against layer index for each model. Positive: source-anchored; negative: target-anchored.

4 Aligning Code-Switched Representations with CANVAS

Motivated by our finding that target-ward anchoring becomes increasingly target-anchored in the upper layers of target-framed code-switched inputs (Section 3), we propose a precisely targeted intervention for CS inputs: **CANVAS** (Contextual Anchor-based Neural Vector Alignment Steering). CANVAS leverages this late-stage geometric window to extract an in-context source canvas from the input itself and softly steers target-language hidden states toward it during prefill. We apply this steering only in the designated upper layers, where the source–target anchoring contrast is most pronounced, and set its strength using the instance-specific controller introduced below.

Stage 1: Token Partition. CANVAS starts by separating the CS input into source, target, and shared spans, which are then used to estimate a source canvas and set the interpolation strength. Given a CS input q^{CS}, we partition the input span as $C(q^{\mathrm{CS}}) = C_{\mathrm{src}} \cup C_{\mathrm{tgt}} \cup C_{\mathrm{oth}}$, where C_{src} contains source-language tokens, C_{tgt} contains target-language tokens, and C_{oth} contains punctuation, numbers, whitespace artifacts, or other language-neutral tokens. We partition input tokens using lightweight token-level language tagging; implementation details are provided in Appendix H.4.

Stage 2: Source Canvas Construction. Using this partition, CANVAS runs an initial forward prefill pass to extract baseline hidden states $H_M(q^{\mathrm{CS}}) \in \mathbb{R}^{L\times T\times d}$ from the templated user-query span (see Appendix H.5 for the extraction protocol). We restrict intervention to the upper layers $\mathcal{L}_{\mathrm{ctrl}} = \{\lfloor 0.7L \rfloor, \ldots, L-1\}$, isolating the final 30% of model depth where abstract semantic anchoring stabilizes. These later layers provide more robust contextual targets for alignment than lower layers tied to surface forms. For each layer $\ell \in \mathcal{L}_{\mathrm{ctrl}}$, we compute two layer-specific anchors—the *source canvas* ($\mathbf{c}_\ell^{\mathrm{src}}$) and the *target anchor* ($\mathbf{a}_\ell^{\mathrm{tgt}}$)—via mean-pooling over their respective token sets:

$$\mathbf{c}_\ell^{\mathrm{src}} = \frac{1}{|C_{\mathrm{src}}|} \sum_{t \in C_{\mathrm{src}}} H_M(q^{\mathrm{CS}})[\ell, t, :], \quad (4)$$

and the *target anchor*,

$$\mathbf{a}_\ell^{\mathrm{tgt}} = \frac{1}{|C_{\mathrm{tgt}}|} \sum_{t \in C_{\mathrm{tgt}}} H_M(q^{\mathrm{CS}})[\ell, t, :]. \quad (5)$$

The source canvas subsequently acts as the dynamic, in-context reference representation for aligning target-language hidden states.

Stage 3: Canvas Alignment Scoring. To quantify the directional drift, we define the layer-wise *canvas direction* as the normalized vector from the target anchor to the source canvas: $\mathbf{u}_\ell = (\mathbf{c}_\ell^{\mathrm{src}} - \mathbf{a}_\ell^{\mathrm{tgt}})/\|\mathbf{c}_\ell^{\mathrm{src}} - \mathbf{a}_\ell^{\mathrm{tgt}}\|_2$. Let $C_{\mathrm{mix}} = C_{\mathrm{src}} \cup C_{\mathrm{tgt}}$. We compute the aggregated mixed representation $\mathbf{m}_\ell$ over all source- and target-language tokens:

$$\mathbf{m}_\ell = \frac{1}{|C_{\mathrm{mix}}|} \sum_{t \in C_{\mathrm{mix}}} H_M(q^{\mathrm{CS}})[\ell, t, :]. \quad (6)$$

The overall *canvas alignment* score $\gamma(q^{\mathrm{CS}})$ is defined as the upper-layer average of the cosine similarity between the mixed representation and the canvas direction:

$$\gamma(q^{\mathrm{CS}}) = \frac{1}{|\mathcal{L}_{\mathrm{ctrl}}|} \sum_{\ell \in \mathcal{L}_{\mathrm{ctrl}}} \cos(\mathbf{m}_\ell, \mathbf{u}_\ell)\,. \quad (7)$$

Model	L	$\mathcal{L}_{\text{ctrl}}$	Avg. F1				ΔF1 by CS condition			
			BASE	Rand	Adapt	q	GF-SRC	GF-TGT	Random	Selective
Llama3.2-1B	16	(11, 15)	2.57	2.68	**3.20**	**	+0.08	+0.82	**+1.01**	+0.62
Mistral-7B	32	(22, 31)	4.53	4.96	**5.06**	**	+0.14	**+0.80**	+0.65	+0.55
Aya-Expanse-8B	32	(22, 31)	5.99	5.99	**6.88**	**	+0.62	**+1.41**	+0.63	+0.91
Llama3.1-8B	32	(22, 31)	6.21	6.72	**7.79**	**	+1.21	+0.64	+2.15	**+2.34**
Mixtral-8x7B	32	(22, 31)	6.77	6.82	**6.91**	–	+0.16	+0.29	**+0.38**	-0.29
Qwen3-30B-A3B	48	(33, 47)	6.91	6.93	**8.11**	**	+0.06	**+2.48**	+0.71	+1.55
Qwen3.5-4B	36	(25, 35)	7.58	7.35	**8.14**	**	+0.37	+0.70	+0.33	**+0.86**
Phi-3.5-MoE	32	(22, 31)	7.99	7.93	**8.82**	**	+0.12	**+1.31**	+0.61	+1.26
Qwen3.5-9B	40	(28, 39)	9.89	10.01	**10.24**	*	-0.20	**+0.77**	+0.22	+0.61
Llama3.1-70B	80	(56, 79)	12.89	12.98	**13.20**	–	-0.52	+0.67	**+0.73**	+0.35
Llama3.3-70B	80	(56, 79)	12.96	13.27	**13.61**	*	+0.29	**+1.66**	+0.61	+0.04
Qwen3.5-27B	48	(33, 47)	14.14	14.22	**15.40**	**	+0.28	+1.74	+0.96	**+2.03**
Qwen3.6-35B-A3B	40	(28, 39)	15.80	16.07	**17.53**	**	+0.66	**+3.58**	+1.41	+1.28

Table 2: **CANVAS intervention results.** Avg. F1 is averaged over four CS conditions; condition-wise columns report Adapt–Base ΔF1. Rand/Adapt denote random-direction/adaptive CANVAS. Bold/underline mark best/second-best values; $^{*}q < 0.05$, $^{**}q < 0.01$ after BH–FDR correction.

A positive γ means that the CS representation is aligned with the source canvas direction, while a negative γ indicates target-side anchoring.

Stage 4: Adaptive Interpolation. Using the instance-specific score $\gamma_{\text{CS}} = \gamma(q^{\text{CS}})$, CANVAS dynamically calibrates the steering intensity α:

$$\alpha = \text{clip}_{[\alpha_{\min}, \alpha_{\max}]}(\alpha_0 - \lambda \gamma_{\text{CS}}), \tag{8}$$

where α_0 represents the base steering intensity at neutral alignment, $\lambda > 0$ controls the sensitivity to the alignment score, and $\alpha_{\min}, \alpha_{\max}$ define the clipped operating range. The negative-slope rule implements adaptive mitigation: if $\gamma_{\text{CS}} > 0$ (source-canvas aligned), the steering intensity decreases; if $\gamma_{\text{CS}} < 0$ (target-oriented), the steering intensity increases. This design applies stronger correction to more target-skewed CS states while keeping the update bounded under a single global schedule across all evaluated models and conditions. We provide hyperparameter validation and sensitivity analyses in Appendices H.1 and H.15.

Stage 5: Online source-canvas interpolation. Following α determination, CANVAS initiates a second prefill pass using upper-layer activation hooks. For each target-language token position $t \in C_{\text{tgt}}$ at layer $\ell \in \mathcal{L}_{\text{ctrl}}$, we inject the source canvas via soft interpolation:

$$\widetilde{\mathbf{h}}_{\ell,t} = (1 - \alpha)\, \mathbf{h}_{\ell,t} + \alpha\, \mathbf{c}_{\ell}^{\text{src}}. \tag{9}$$

This operation runs online, modifying only the latent vectors at target-token positions while leaving the surface text completely untouched. The adjusted hidden states are saved directly into the KV cache, from which the model greedily decodes the final answer.

4.1 Intervention Protocol and Results

To evaluate the practical effectiveness of CANVAS, we compare its downstream CS performance against standard greedy decoding (**BASE**).

Evaluation protocol. We evaluate **BASE** decoding against two CANVAS configurations: (1) **Rand**, a baseline that injects an identical interpolation but utilizes a randomly sampled steering direction; and (2) **Adapt**, our full framework that dynamically sets the steering intensity α via the instance-specific score γ. Beyond the grammar-forced conditions (GF-SRC, GF-TGT), we expand our evaluation to the *Random* and *Selective* lexical switching conditions from CodeMixQA (Winata et al., 2026), yielding four structural environments to test CANVAS's robustness.

Consistent QA Enhancement Across Conditions. As shown in Table 2, CANVAS (**Adapt**) improves average F1 across all evaluated MLLMs, supporting the view that source-canvas alignment can recover part of the CS performance loss. The most substantial accuracy gains concentrate under the GF-TGT condition—the setting where our probing finds the strongest target-ward representation drift and the largest F1 degradation. Furthermore, consistent gains under *Random* and *Selective* conditions demonstrate that source-canvas alignment operates as a generalizable linguistic correction mechanism. CANVAS further generalizes to a Spanish source anchor, shows significant paired gains for most models, and adds only modest overhead without external model calls (Appendices E.2, H.13, and H.10). It therefore remains a lightweight method for broad CS settings.

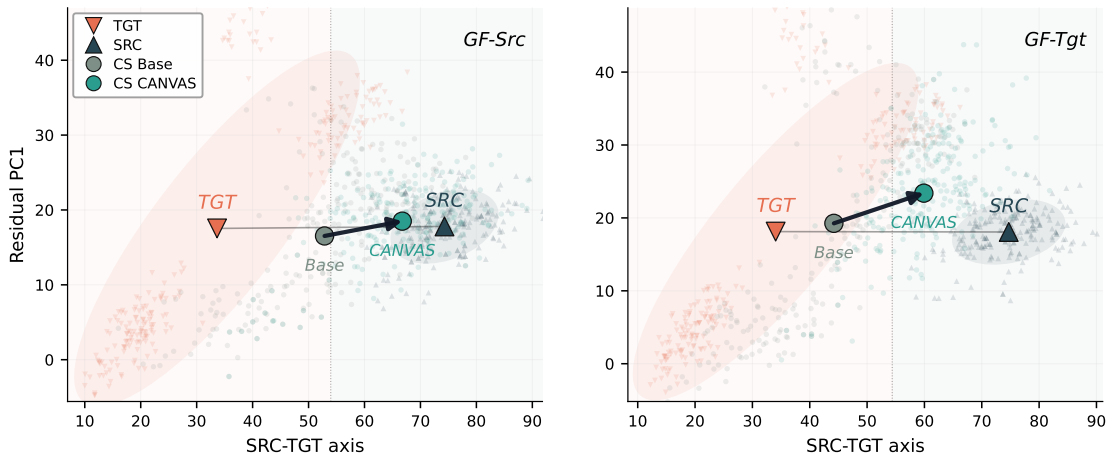

Figure 3: **Latent displacement along the SRC–TGT axis.** We project token states onto a 2D subspace where the horizontal axis captures the global SRC–TGT direction. Faded points indicate individual instances, large markers denote condition centroids, and arrows show the directional shift induced by CANVAS.

4.2 Analysis and Geometric Diagnostics

Geometric Verification of Representational Steering. To check whether the intervention produces the intended geometric correction, we visualize the latent trajectories of Qwen3.5-27B before and after the intervention. We project hidden states onto the primary SRC–TGT population vector, derived as the macro-average difference between clean source and target representations. For the vertical axis, we first remove this SRC–TGT direction from each representation and then take the first principal component of the remaining residual variation. As shown in Figure 3, the representational centroids for both GF-SRC and GF-TGT undergo a substantial, rightward vector shift post-intervention, moving toward the clean source population space. This geometric relocation supports the interpretation that CANVAS shifts the latent processing path toward the source-side anchor.

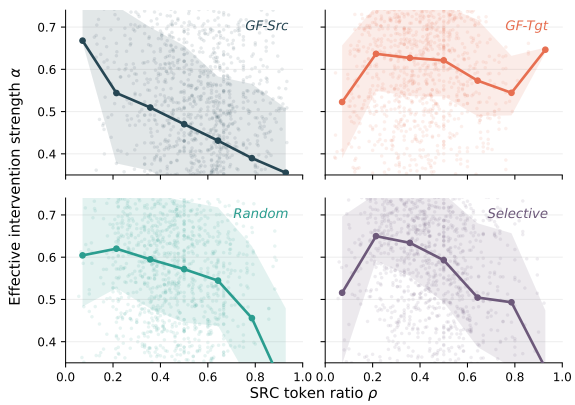

Figure 4: **Adaptive intervention strength.** CANVAS assigns larger interpolation strength α to more target-heavy inputs, measured by lower source-token ratio ρ.

Adaptive Response to Structural Skew. To verify whether CANVAS responds to the degree of cross-lingual drift estimated from internal representations and adjusts its interpolation strength accordingly, we analyze the operational sensitivity of the adaptive controller. Aggregating data across all models, we compute the instance-level source token ratio $\rho = |C_{\text{src}}|/(|C_{\text{src}}| + |C_{\text{tgt}}|)$ and track its correlation with the runtime steering intensity α. As illustrated in Figure 4, CANVAS exhibits tight, deliberate sensitivity to the input's structural imbalance without manual thresholding: GF-SRC queries naturally preserve a high source ratio and trigger mild alignment ($\rho = 0.5936$, $\gamma = -0.0070$, $\alpha = 0.4533$), whereas target-heavy GF-TGT inputs automatically elicit more aggressive geometric adjustments ($\rho = 0.4044$, $\gamma = -0.1394$, $\alpha = 0.6083$). This monotonic pattern suggests that CANVAS allocates stronger intervention to inputs with greater target-ward drift, where source-side correction is more needed.

Robustness on Monolingual Inputs. Although CANVAS targets language-mixed inputs, it should remain stable in ordinary monolingual or source-dominant contexts. Our adaptive controller supports this by reducing the effective interpolation strength when the input shows little target-ward drift, making the update close to an identity operation rather than applying a fixed intervention. We empirically validate this safety property on pure monolingual and source-heavy inputs ($\rho_{\text{SRC}} \geq 0.80$) (detailed in Appendix H.3), where CANVAS preserves standard parametric performance.

Model	Bilingual CS					Trilingual CS				
	ID-EN		HA-EN		Bi. Mean	JV-ID-EN		SU-ID-EN		Tri. Mean
	BASE	CANVAS	BASE	CANVAS	ΔAcc	BASE	CANVAS	BASE	CANVAS	ΔAcc
Mistral-7B	25.45	**26.67**	39.90	**40.40**	+0.86	**29.09**	28.28	56.57	56.57	−0.41
Llama-3.1-8B	20.81	**23.64**	60.61	**65.66**	+3.94	37.37	**43.64**	78.38	**79.39**	**+3.64**
Aya-Expanse-8B	16.16	**18.59**	44.44	**50.00**	**+3.99**	28.28	**30.91**	84.65	**85.25**	+1.62
Qwen3.5-27B	25.25	**28.28**	90.40	90.40	+1.52	**59.60**	58.38	86.46	**86.67**	−0.51
Mean	21.92	**24.30**	58.84	**61.62**	+2.58	38.59	**40.30**	76.52	**76.97**	+1.09

Table 3: **CANVAS on PingPong multi-turn CS QA.** Five-option multiple-choice accuracy (%) for BASE and CANVAS. Bi. / Tri. Mean report the average ΔAcc for bilingual and trilingual CS settings.

4.3 Generalization to Multi-Turn CS Dialog.

We test whether CANVAS generalizes beyond tightly controlled, single-turn prompts to noisy, long-context multi-turn environments. We evaluate on the answerable QA subset of PingPong (Farhansyah et al., 2026), a human-authored multi-party dialog benchmark. The evaluated settings cover Indonesian–English (id-en), Hausa–English (ha-en), Javanese–Indonesian–English (jv-id-en), and Sundanese–Indonesian–English (su-id-en). Using the same 30% upper-layer schedule, CANVAS transfers effectively to this setting (Table 3). It yields a +2.58 accuracy gain on bilingual conversation pairs (id-en, ha-en) across all tested architectures. Even under trilingual settings (jv-id-en, su-id-en), where forcing a binary partition is naturally lossy, CANVAS secures a positive average improvement of +1.09 Acc. Appendix H.16 further shows positive gains on retrieval-augmented QA and topic classification. Together, these results suggest that in-context source alignment generalizes beyond controlled short-answer QA to more realistic CS settings.

5 Related Work

5.1 Multilingual Consistency in LLMs

Prior work shows that multilingual LLMs do not behave uniformly across languages, reporting fairness gaps, hallucination gaps, retrieval-augmented generation failures, and cross-language preferences (Piqueras and Søgaard, 2022; Ramesh et al., 2023; Chataigner et al., 2024; Wu et al., 2024; Park and Lee, 2025). Other studies examine cross-lingual factual consistency and representation transfer, showing that knowledge can be unevenly represented or accessed across languages (Qi et al., 2023; Xu et al., 2023). Closely related work further suggests that multilingual models may rely on latent language preferences or English-centered internal processing (Wendler et al., 2024; Zhao et al., 2024; Sharma et al., 2025). Our work extends this line of analysis to code-switched inputs, asking whether a single mixed-language question is internally anchored toward the source or target side.

5.2 Code-Switching in LLMs

Code-switching has long been studied as a structured linguistic phenomenon shaped by grammatical, discourse-level, and interactional constraints, rather than as random multilingual noise (Doğruöz et al., 2021; Winata et al., 2023). Recent NLP work studies CS as an evaluation setting for robustness, translation, generation, understanding, and reasoning (Laureano De Leon et al., 2024; Huzaifah et al., 2024; Abdaljalil et al., 2025; Mohamed et al., 2025; Heredia et al., 2025; Winata et al., 2026). Controlled CS studies further show that code-switched inputs can reveal syntactic generalization, language-specific knowledge access, output-language alignment, and cross-lingual consistency effects in LLMs (Sterner and Teufel, 2025b,a; Kim et al., 2025; Oh et al., 2026). Our work follows this probing perspective, but focuses on internal language anchoring: we measure whether a CS representation becomes more source-like or target-like, and whether this anchor predicts QA degradation.

6 Conclusion

We study how MLLMs process code-switched inputs through the lens of internal language anchoring. We introduce anchor bias to measure whether a CS representation is closer to its source or target-language counterpart, and show that grammatical frames consistently shift CS representations toward the corresponding language anchor. We further propose CANVAS, an inference-time intervention that uses the same anchor signal to softly align target-language token representations toward an in-context source canvas.

Limitations

Our analysis is limited in that it models code-switched processing through paired source and target anchors. This controlled axis enables matched comparisons with monolingual siblings and links representation shifts to downstream behavior, but it also compresses richer multilingual dynamics into a source–target contrast. Natural code-switching may involve local syntax, borrowed terms, named entities, discourse context, and language-specific factual familiarity that cannot be fully captured by a single anchor direction. We therefore treat anchor bias as a measurable and actionable component of CS processing, rather than a complete account of multilingual degradation.

Ethics Statement

We conduct all experiments using publicly available multilingual datasets and models that are distributed under documented research licenses, and we follow the usage terms set by the original providers. Our short-answer factoid QA setting is based on the CodeMixQA construction protocol, and our multi-turn QA setting is based on PingPong. Target-language counterparts are produced by a strong neural translator (Qwen3-235B-A22B), and grammar-forced code-switched variants are utilized following the CodeMixQA protocol from prior work. No new human-subject data are collected, and no personally identifying information is introduced by our pipeline.

References

Samir Abdaljalil, Erchin Serpedin, Khalid Qaraqe, and Hasan Kurban. 2025. Evaluating multilingual and code-switched alignment in LLMs via synthetic natural language inference. *arXiv preprint arXiv:2508.14735*.

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Kalika Bali, Jatin Sharma, Monojit Choudhury, and Yogarshi Vyas. 2014. "i am borrowing ya mixing?" an analysis of english-hindi code mixing in facebook. In *Proceedings of the first workshop on computational approaches to code switching*, pages 116–126.

Lucas Bandarkar, Chenyuan Yang, Mohsen Fayyaz, Junlin Hu, and Nanyun Peng. 2026. Multilingual routing in mixture-of-experts. *Preprint*, arXiv:2510.04694.

Cléa Chataigner, Afaf Taïk, and Golnoosh Farnadi. 2024. Multilingual hallucination gaps in large language models. *arXiv preprint arXiv:2410.18270*.

Yuxin Chen, Zhengzhou Cai, Xiangtian Ji, Weixiang Zhao, An Zhang, Xiang Wang, and Tat-Seng Chua. 2026. Understanding multilingualism in mixture-of-experts llms: Routing mechanism, expert specialization, and layerwise steering. *arXiv preprint arXiv:2601.14050*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

John Dang, Shivalika Singh, Daniel D'Souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, and 1 others. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

A. Seza Doğruöz, Sunayana Sitaram, Barbara E. Bullock, and Almeida Jacqueline Toribio. 2021. A survey of code-switching: Linguistic and social perspectives for language technologies. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1654–1666, Online. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? comparing the geometry of bert, elmo, and gpt-2 embeddings. 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 55–65.

Mohammad Rifqi Farhansyah, Hanif Muhammad Zhafran, Farid Adilazuarda, Shamsuddeen Hassan Muhammad, Maryam Ibrahim Mukhtar, Nedjma Ousidhoum, Genta Indra Winata, Ayu Purwarianti, and Alham Fikri Aji. 2026. Pingpong: A natural benchmark for multi-turn code-switching dialogues. *Preprint*, arXiv:2601.17277.

Lukas Haas, Gal Yona, Giovanni D'Antonio, Sasha Goldshtein, and Dipanjan Das. 2026. Simpleqa verified: A reliable factuality benchmark to measure parametric knowledge. *Preprint*, arXiv:2509.07968.

Maite Heredia, Gorka Labaka, Jeremy Barnes, and Aitor Soroa. 2025. Conditioning LLMs to generate code-switched text. *arXiv preprint arXiv:2502.12924*.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalization. *Preprint*, arXiv:2003.11080.

Muhammad Huzaifah, Weihua Zheng, Nattapol Chanpaisit, and Kui Wu. 2024. Evaluating code-switching translation with large language models. In *Proceedings of LREC-COLING*, pages 6381–6394.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Seoyeon Kim, Huiseo Kim, Chanjun Park, Jinyoung Yeo, and Dongha Lee. 2025. Can code-switched texts activate a knowledge switch in LLMs? a case study on English-Korean code-switching. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 22303–22327, Suzhou, China. Association for Computational Linguistics.

Frances Adriana Laureano De Leon, Harish Tayyar Madabushi, and Mark Lee. 2024. Code-mixed probes show how pre-trained models generalise on code-switched text. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3457–3468, Torino, Italia. ELRA and ICCL.

Yaobo Liang, Nan Duan, Yeyun Gong, Ning Wu, Fenfei Guo, Weizhen Qi, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, Xiaodong Fan, Ruofei Zhang, Rahul Agrawal, Edward Cui, Sining Wei, Taroon Bharti, Ying Qiao, Jiun-Hung Chen, Winnie Wu, and 5 others. 2020. XGLUE: A new benchmark dataset for cross-lingual pre-training, understanding and generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6008–6018, Online. Association for Computational Linguistics.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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

Meta AI. 2024. Introducing Meta Llama 3: The most capable openly available LLM to date. https://ai.meta.com/blog/meta-llama-3/. Accessed: 2026-05-26.

Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. 2025. Lost in the mix: Evaluating LLM understanding of code-switched text. *arXiv preprint arXiv:2506.14012*.

Juhyun Oh, Haneul Yoo, Faiz Ghifari Haznitrama, and Alice Oh. 2026. Ola: Output language alignment in code-switched llm interactions. *Preprint*, arXiv:2601.03589.

Jeonghyun Park and Hwanhee Lee. 2025. Investigating language preference of multilingual rag systems. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5647–5675.

Matthew E Peters, Mark Neumann, Luke Zettlemoyer, and Wen-tau Yih. 2018. Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 1499–1509.

Laura Cabello Piqueras and Anders Søgaard. 2022. Are pretrained multilingual models equally fair across languages? In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3597–3605.

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.

Krithika Ramesh, Sunayana Sitaram, and Monojit Choudhury. 2023. Fairness in language models beyond english: Gaps and challenges. In *Findings of the association for computational linguistics: EACL 2023*, pages 2106–2119.

Nikhil Sharma, Kenton Murray, and Ziang Xiao. 2025. Faux polyglot: A study on information disparity in multilingual large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8090–8107, Albuquerque, New Mexico. Association for Computational Linguistics.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Igor Sterner and Simone Teufel. 2025a. Code-switching and syntax: A large-scale experiment. In *Findings of ACL*, pages 11526–11533.

Igor Sterner and Simone Teufel. 2025b. Minimal pair-based evaluation of code-switching. In *Proceedings of ACL*, pages 18575–18598.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. Bert rediscovers the classical nlp pipeline. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4593–4601.

William Timkey and Marten van Schijndel. 2021. All bark and no bite: Rogue dimensions in transformer language models obscure representational quality. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4527–4546, Online and Punta Cana, Dominican Republic. 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.

Genta Winata, Alham Fikri Aji, Zheng Xin Yong, and Thamar Solorio. 2023. The decades progress on code-switching research in NLP: A systematic survey on trends and challenges. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2936–2978, Toronto, Canada. Association for Computational Linguistics.

Genta Indra Winata, David Anugraha, Patrick Amadeus Irawan, Anirban Das, Haneul Yoo, Paresh Dashore, Shreyas Kulkarni, Ruochen Zhang, Haruki Sakajo, Frederikus Hudi, and 1 others. 2026. Can large language models understand, reason about, and generate code-switched text? *arXiv preprint arXiv:2601.07153*.

Suhang Wu, Jialong Tang, Baosong Yang, Ante Wang, Kaidi Jia, Jiawei Yu, Junfeng Yao, and Jinsong Su. 2024. Not all languages are equal: Insights into multilingual retrieval-augmented generation. *arXiv preprint arXiv:2410.21970*.

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.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zheng-Xin Yong, Ruochen Zhang, Jessica Zosa Forde, Skyler Wang, Arjun Subramonian, Holy Lovenia, Samuel Cahyawijaya, Genta Indra Winata, Lintang Sutawika, Jan Christian Blaise Cruz, Yin Lin Tan, Long Phan, Rowena Garcia, Thamar Solorio, and Alham Fikri Aji. 2023. Prompting multilingual large language models to generate code-mixed texts: The case of south East Asian languages. In *Proceedings of the 6th Workshop on Computational Approaches to Linguistic Code-Switching*, pages 43–63, Singapore. Association for Computational Linguistics.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *Preprint*, arXiv:2402.18815.

Kening Zheng, Wei-Chieh Huang, Jiahao Huo, Zhonghao Li, Henry Peng Zou, Yibo Yan, Xin Zou, Jungang Li, Junzhuo Li, Hanrong Zhang, Xuming Hu, and Philip S. Yu. 2026. Unveiling language routing isolation in multilingual moe models for interpretable subnetwork adaptation. *Preprint*, arXiv:2604.03592.

A Appendix Table of Contents

1. Section B describes the use of LLMs during manuscript preparation.
2. Section C provides **implementation details**, including the OpenRouter pipeline, local decoding setup, hidden-state extraction, quantization fallback, and hardware resources.
3. Section D provides **complementary analyses for MoE models**, including MoE routing extraction, routing-profile similarity, routing configuration, Cohen's d for routing separation, and the relationship between expert routing and representation-level anchor bias.
4. Section E tests **source-language generalization with Spanish as the source anchor**, covering both the Spanish-pivot anchor-bias replication and Spanish-anchor CANVAS results.
5. Section F describes the **dataset construction and matched comparison statistics**, caveats on translated target-language questions, additional code-switching conditions, and prompt templates.
6. Section G provides additional **anchor-bias** analyses, including anisotropy diagnostics, language-wise breakdowns, metric variants, frontier-model scaling validation, and same-lexicon order counterfactuals that disentangle lexical language from word order.
7. Section H provides detailed analyses of **CANVAS**, including the adaptive α schedule, layer-wise effects, source-heavy checks, token-level language tagging, hidden-state extraction, layer-span ablations, language-wise mitigation, case studies, model-wise movement diagnostics, cost analysis, additional movement analyses, cumulative recovery trajectories, statistical tests, detector sensitivity, hyperparameter sensitivity, and task extensions to retrieval-augmented QA, multi-turn dialog QA, and topic classification.

B The Use of Large Language Models

We write the manuscript ourselves, and an LLM (ChatGPT) is used solely for refinement—style, clarity, and grammar. It is not used for ideation or content generation.

C Implementation Details

OpenRouter pipeline. Translation of SimpleQA questions and answer generation by the high-capacity reference model Qwen3-235B-A22B-2507 (Yang et al., 2025), accessed via the OpenRouter OpenAI-compatible chat-completion API, with `DeepInfra` as the sole underlying provider for all calls (selected for consistent latency, to avoid provider switching, and to improve reproducibility). The maximum translation length is 256 tokens and the maximum answer length is 64 tokens; sampling temperature is set to 0 (greedy).

Local model decoding. All dense models (Aya-Expanse-8B, Qwen3.5-4B/9B/27B, Llama-3.1-70B-Instruct, Llama-3.3-70B-Instruct) and all MoE models (Mixtral-8x7B-Instruct, Phi-3.5-MoE-Instruct, Qwen3-30B-A3B, Qwen3.6-35B-A3B) are run locally with deterministic greedy decoding. Specifically, we set `do_sample=False` and use each model's default end-of-sequence token. For Qwen-3 family models, we set `enable_thinking=False` where the tokenizer supports it.

Hidden-state extraction. For representation analysis, we apply each model's chat template to the question, run a forward pass with `output_hidden_states=True`, and pool representations over the question-content region. We exclude shared prompt tokens (system instruction, assistant prefix, chat-template markers) from the pooled representation so that cosine-based comparisons focus on the question itself rather than on identical instruction text.

Quantization fallback. Models are loaded with automatic device mapping in bfloat16 where possible. If bf16 OOMs on the available hardware, we fall back to 4-bit NF4 quantization via `bitsandbytes` (used for all 70B-class dense models and the largest MoE checkpoints) or, for intermediate cases, to 8-bit. The CANVAS hidden-state interpolation is applied after dequantization at each control layer, so the intervention does not interact with the quantization scheme.

Hardware. All local inference runs use a single SLURM node with $4\times$ NVIDIA RTX 6000 Ada Generation GPUs (48 GB each), connected via PCIe. Per-model runtime is dominated by the prefill pass with hidden-state extraction. A full eval-

Model	Cosine similarity over expert-frequency vectors						Jaccard overlap over active expert sets					
	cos(SRC, CS)			cos(TGT, CS)			Jacc(SRC, CS)			Jacc(TGT, CS)		
	GF-SRC	GF-TGT	d	GF-SRC	GF-TGT	d	GF-SRC	GF-TGT	d	GF-SRC	GF-TGT	d
Qwen3-30B-A3B	0.924	0.879	+0.64	0.912	0.943	-0.49	0.910	0.886	+0.94	0.897	0.917	-0.68
Qwen3.6-35B-A3B	0.933	0.905	+0.63	0.925	0.944	-0.40	0.646	0.586	+0.56	0.596	0.653	-0.53
Phi-3.5-MoE	0.896	0.843	+0.55	0.905	0.942	-0.51	0.969	0.960	+0.65	0.960	0.967	-0.42

Table 4: **Routing similarity under grammar-forced CS.** We compare CS routing profiles against source and target routes. Highlighted columns indicate the frame-matched route.

uation pass for one model and one CS condition completes within roughly 0.05–0.3 h for models up to 8B parameters, 0.4–0.8 h for 27–35B models, and 0.6–1.1 h for 70B-class dense models and the largest MoE checkpoints (Mixtral-8x7B), measured as the sum of per-example base, oracle, and CANVAS generation wall-clock times in the result logs.

D Code-Switching for MoE Models

D.1 Routing Similarity in MoE Models

The representation-level results show where grammar-forced CS inputs are placed in the hidden-state space. However, hidden-state geometry is only one axis of how MLLMs process language. In mixture-of-experts (MoE) models, language processing can also be examined through the sparse computational pathway selected by the router: each token is assigned to a small subset of feed-forward experts. Recent work on multilingual MoE models shows that different languages can induce distinct routing patterns and partially specialized experts (Bandarkar et al., 2026; Chen et al., 2026; Zheng et al., 2026). This makes routing a complementary axis for analyzing language understanding in MLLMs. While prior analyses mainly focus on monolingual or language-separated inputs, CS provides a more fine-grained setting: a single question contains material from two languages, but its grammatical frame may be dominated by one of them. We therefore ask whether the language anchor observed in hidden-state geometry also appears in the model's expert-routing profile.

Routing profiles. We collect top-k expert selections from router or gate modules during the same forward pass for MoE models. We aggregate expert selections over the question-content region and compare whether the CS routing profile is closer to the source or target-language routing profile. For each matched question set, we run separate forward passes for the source, CS, and target-language variants under the same prompt format. At each MoE layer, the router selects top-k experts for every question-content token. Let $z_{\ell,t,j}(q)$ denote the j-th selected expert for token t at layer ℓ when processing question variant q, where $j \in \{1, \ldots, k\}$. We aggregate these selections into a layer-wise expert-frequency vector $\mathbf{f}_\ell(q)$, whose e-th component is

$$f_{\ell,e}(q) = \frac{1}{k|C(q)|} \sum_{t \in C(q)} \sum_{j=1}^{k} \mathbb{I}[z_{\ell,t,j}(q) = e] \,. \tag{10}$$

This vector records how often each expert is selected for a given input condition. Thus, source, CS, and target-language questions each induce their own routing-frequency profile.

Routing similarity. We compare each CS routing profile against the source and target-language routing profiles. First, we compute cosine similarity between expert-frequency vectors:

$$S_\ell^{\cos}(q_i, q_j) = \cos(\mathbf{f}_\ell(q_i), \mathbf{f}_\ell(q_j)) \,. \tag{11}$$

Second, we compute the Jaccard overlap between active expert sets. Let

$$A_\ell(q) = \{e \mid f_{\ell,e}(q) > 0\} \tag{12}$$

be the set of experts activated by question q at layer ℓ. We define

$$S_\ell^{\text{jac}}(q_i, q_j) = \frac{|A_\ell(q_i) \cap A_\ell(q_j)|}{|A_\ell(q_i) \cup A_\ell(q_j)|}. \tag{13}$$

For each CS condition, we report its routing alignment with the source route, $S(q_s, q_c)$, and with the target route, $S(q_t, q_c)$, averaging each score over matched examples and MoE layers. We additionally report Cohen's d as a standardized effect size between the GF-SRC and GF-TGT routing-score distributions; positive values indicate larger scores for GF-SRC, while negative values indicate larger scores for GF-TGT. Details are provided in Appendix D.2.

Routing shifts toward the frame language. Table 4 shows that the grammatical frame also shapes the sparse routing path selected by MoE models. Source-framed CS inputs are routed more like the source-language question: their expert-frequency vectors and active expert sets are closer to the SRC routing profile. Conversely, target-framed CS inputs are routed more like the target-language question, showing higher alignment with the TGT routing profile. This pattern holds consistently across both cosine similarity over expert-frequency vectors and Jaccard overlap over active expert sets. The Cohen's d values further show that the two grammar-forced conditions induce clearly separated routing distributions. Beyond this directional pattern, routing behavior also aligns with the representation-level anchor signal. At the example level, routing bias, defined as source-route alignment minus target-route alignment, correlates strongly with anchor bias for cosine routing profiles ($\rho = +0.703$, $p < 0.001$) and moderately for Jaccard routing profiles ($\rho = +0.464$, $p < 0.001$). This suggests that expert-routing shifts track the same internal anchoring reflected in hidden states and are behaviorally relevant.

D.1.1 MoE Routing Configuration

We report the routing configuration for the MoE models used in the expert-selection analysis. These details are important because routing similarity depends on the number of available experts, the number of experts selected per token, and the number of MoE layers from which routing traces are collected. Models with different expert counts and top-k values can have different baseline overlap patterns, so these metadata help interpret the routing-similarity results in Table 4.

Model	Experts	top-k	Routing layers
Phi-3.5-MoE	16	2	32
Qwen3-30B-A3B	128	8	48
Qwen3.6-35B-A3B	256	8	40

Table 5: **MoE routing configuration.** We compute routing similarity from the top-k expert selections over question-content tokens at each MoE layer.

For each model, we collect expert selections over the question-content region after applying the same prompt format used in the representation analysis. We then aggregate these token-level selections into layer-wise expert-frequency vectors. This makes the routing analysis parallel to the hidden-state analysis: both compare matched EN, TGT, and CS variants over the same question region.

D.2 Cohen's d for Routing Separation

We use Cohen's d to quantify how strongly the routing-score distributions differ between GF-SRC and GF-TGT. For a given routing metric, let a and b denote the per-example routing scores under GF-SRC and GF-TGT, respectively. We compute

$$d = \frac{\mu(a) - \mu(b)}{\sqrt{(\sigma_a^2 + \sigma_b^2)/2}}, \tag{14}$$

where $\mu(\cdot)$ and $\sigma^2(\cdot)$ denote the mean and variance of each distribution. The sign of d gives the direction of the effect: $d > 0$ means that GF-SRC has larger routing scores, while $d < 0$ means that GF-TGT has larger routing scores. The magnitude $|d|$ measures standardized separation; as a common rule of thumb, values around 0.2, 0.5, and 0.7 are often interpreted as small, medium, and large effects. We use d only as an effect-size summary, not as a statistical significance test.

D.3 Expert Routing Mirrors Anchor Bias

Table 4 in the main text shows that grammar-forced CS inputs route toward the frame-matched language route. Here, we provide a compact mechanism check linking this expert-level signal to representation-level anchor bias, without repeating the main routing table.

Each point in Figure 5 is an example-level CS input from an MoE model. The x-axis is routing bias, computed as the SRC-vs-TGT difference in CS routing similarity, and the y-axis is representation anchor bias computed from hidden states. Positive values indicate SRC-oriented behavior, whereas negative values indicate TGT-oriented behavior. Thus, points in the upper-right and lower-left quadrants indicate examples where expert routing and representation geometry agree in their language orientation. The positive association suggests that language anchoring is reflected not only in hidden-state geometry but also in sparse expert selection.

E Source-Language Generalization: Spanish Pivot

E.1 Anchor Bias

Experimental setup. We replace English with Spanish as the source language and build eight CS

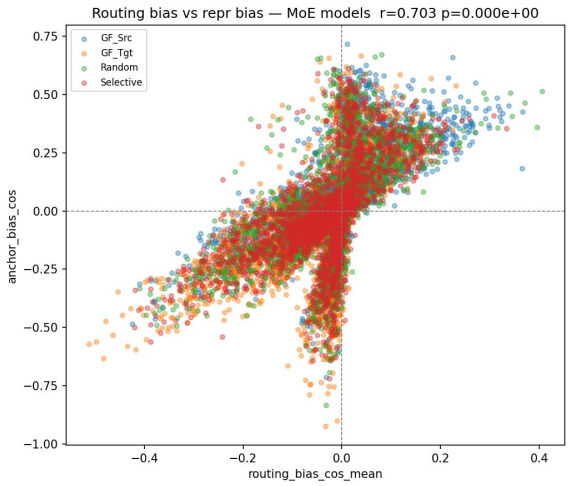

Figure 5: **Routing bias tracks representation anchor bias.** Each point is an example-level CS input from an MoE model. The horizontal axis reports SRC-vs-TGT expert-routing bias, and the vertical axis reports representation anchor bias. The positive association indicates that expert allocation shifts in the same SRC/TGT direction as representation-space anchor bias.

pairs: ES↔{Toba Batak, Bengali, French, Hindi, Korean, Marathi, Urdu, Chinese}. Questions, system prompts, and gold answers are in Spanish; CS inputs follow the same GF-Src / GF-Tgt grammar-forcing protocol as the main experiments. We evaluate Aya-Expanse-8B, Qwen3.5-4B, Qwen3.5-9B, and Qwen3.5-27B. Anchor bias uses the same upper-half aggregation and per-layer random cosine normalization as Table 1.

Results. Table 6 shows the same frame-dependent split with Spanish as the source. GF-Src is consistently ES-anchored ($+0.213$ to $+0.358$), while GF-Tgt is consistently TGT-anchored (-0.190 to -0.337). The mean gap is $+0.541$, larger than the English-pivot upper-half gap of $+0.384$ computed over the same four models. QA follows the same ordering: GF-Src has a smaller mean F1 drop than GF-Tgt (-3.4 vs. -5.9 pp), and SRC-anchored layer ratios are much higher for GF-Src ($\sim 82\%$) than GF-Tgt ($\sim 35\%$).

Interpretation. The Spanish pivot rules out an English-specific explanation: the model tracks the language that supplies the grammatical frame, not English identity itself. The effect is also not weakened when the source language is less dominant in pre-training. We therefore interpret anchor bias as a general source-language anchoring phenomenon: upper-layer representations follow the grammatical backbone of the input, whether that source is English or Spanish. MoE routing diagnostics are not repeated for this pivot.

E.2 CANVAS

Setup. We test whether CANVAS generalizes beyond English as the source anchor. We construct a Spanish-pivot CS dataset by pairing each TGT language with Spanish in place of English and translating the English gold answers to Spanish via Qwen3.6-35B-A3B. We evaluate four conditions per example: ES-ORACLE (pure-Spanish question, upper bound), BASE (Spanish–TGT CS, no intervention), CANVAS (CS with steering toward the Spanish anchor), and TGT-ORACLE (pure-TGT question, lower bound). We pair Spanish with eight TGT languages, giving $1{,}280$ examples per model (8 pairs $\times$ 2 CS conditions $\times$ 80 items).

Pipeline. We keep the adaptive interpolation schedule used in the main runs ($\alpha = \text{clip}_{[0.05,0.75]}(0.45 - 1.5\gamma)$, 30% layer budget) and replace only the SRC anchor with Spanish. For non-Latin TGT scripts (Bengali, Devanagari, Korean, Chinese), we tag tokens by character-level script: Latin-script tokens are treated as SRC and non-Latin tokens as TGT. For Latin-script TGT languages (French, Toba Batak), we fall back to the `lingua` language detector. The system prompt declares Spanish as the answer language.

Result. We report results in Table 7. We observe a positive $\Delta\text{F1}_{\text{CANVAS}}$ on all five models, with a mean improvement of $+0.67$ F1 over direct CS answering. Llama-3.1-8B benefits the most ($+1.47$), and the two smaller Qwen variants gain $+0.75$ and $+0.85$. Even Mistral-7B ($+0.15$) and Qwen3.5-27B ($+0.13$) show no regression and stay well above their corresponding TGT-oracle baselines.

Script-dependent behaviour. We find a clear script-dependent pattern when aggregating per-pair gains across the five models. The six non-Latin TGT pairs (BEN, HIN, KOR, MAR, URD, ZHO) all gain ΔF1 in the range $+0.52$ to $+1.35$ (mean $+0.95$), while the two Latin-script TGT pairs (BBC-ESP $+0.03$ and FRA-ESP -0.37) average -0.17. This pattern is consistent with the tagger design: script-based tagging is unambiguous for non-Latin TGT languages, whereas the `lingua` fallback used for Latin-script TGT languages can struggle to separate Spanish from mor-

Model	AB^{upper}_{nrm} (*ES as source*)			QA behavior (F1)			SRC-anchored layers (%)		
	GF-SRC	GF-TGT	Δ_{AB}	SRC	GF-SRC	GF-TGT	GF-SRC	GF-TGT	Δ_{layer}
Aya-Expanse-8B	+0.358	-0.337	+0.695	14.8	11.2 (-3.6)	8.3 (-6.5)	83.8	40.5	+43.3
Qwen3.5-4B	+0.261	-0.256	+0.517	7.4	4.7 (-2.7)	2.5 (-4.9)	79.8	32.7	+47.1
Qwen3.5-9B	+0.281	-0.270	+0.550	12.4	9.5 (-2.9)	6.5 (-5.9)	80.7	33.5	+47.2
Qwen3.5-27B	+0.213	-0.190	+0.403	17.1	12.7 (-4.4)	10.7 (-6.4)	82.4	35.0	+47.4
Mean	+0.278	-0.263	+0.541	12.9	9.5 (-3.4)	7.0 (-5.9)	81.7	35.4	+46.3

Table 6: **Source-language anchor bias with Spanish (ES) as source.** Metrics follow Table 1: AB^{upper}_{nrm} averages normalized anchor bias over the upper half of layers, and SRC-anchored layers count the fraction of layers with $AB_{nrm} > 0$.

Model	ES-ora	Base	CANVAS	TGT-ora	$\Delta F1_{CANVAS}$
Llama3.1-8B	10.54	7.06	8.53	2.57	+1.47
Mistral-7B	8.00	3.46	3.61	1.16	+0.15
Qwen3.5-4B	7.36	3.60	4.35	2.17	+0.75
Qwen3.5-9B	12.39	8.03	8.88	3.85	+0.85
Qwen3.5-27B	17.10	11.68	11.81	9.17	+0.13
Mean	11.08	6.77	7.44	3.78	+0.67

Table 7: **CANVAS with a Spanish anchor.** F1 (%) on the Spanish-pivot CS dataset. ES-ora and TGT-ora are pure-Spanish and pure-target baselines, respectively. CANVAS improves over direct CS answering for all five models, showing that the source-anchor intervention is not English-specific.

phologically similar Romance-language tokens, most notably French. The non-Latin sub-pool, therefore, provides the cleanest evaluation of the language-agnostic claim, and its Spanish-anchor gains match the magnitude of the English-anchor CANVAS improvements reported in the main paper.

Takeaway. CANVAS produces positive $\Delta F1$ across all five evaluated models when Spanish replaces English as the source anchor, with the strongest effect on non-Latin TGT languages where script-based tagging is most reliable. This result argues against the alternative explanation that the English-anchor gains depend on a privileged role of English in the representation space: the same steering mechanism transfers to another high-resource source language with only the runtime language-detection backend changed. The Latin-script TGT sub-pool further suggests that the main practical bottleneck for non-English anchors is SRC/TGT tagging precision, rather than the steering mechanism itself.

F Dataset Details

F.1 Dataset Statistics

We provide dataset statistics to clarify the scale and balance of the controlled CS evaluation. Our setup starts from English SimpleQA questions and pairs each question with a target-language counterpart and a code-switched variant. Because each CS example is evaluated together with its matched EN and TGT forms, the resulting evaluation is organized around EN–TGT–CS comparisons rather than independent monolingual examples.

Item	Count
Unique English source questions	955
Target languages	9
CS examples per language	320
Examples per language–condition	80
Matched EN–TGT–CS comparisons	2,880
CS conditions	4
Evaluated variants per CS example	3
Total evaluated input instances	8,640

Table 8: **Dataset statistics.** Each CS example is evaluated with matched EN, TGT, and CS inputs under the same reference answer. The four CS conditions are balanced within each language pair.

As shown in Table 8, each target language contributes 320 CS examples, with 80 examples for each CS condition. Since each CS example is evaluated with its matched English and target-language counterparts, the full evaluation contains 8,640 input instances. This balanced construction ensures that differences between GF-SRC and GF-TGT are not driven by unequal language or condition coverage. We use nine target languages: Bengali, Spanish, French, Hindi, Korean, Marathi, Toba Batak, Urdu, and Chinese.

F.2 Caveat on Translated Target Questions

In our matched set $\mathcal{Q}(x) = \{q^{\text{SRC}}, q^{\text{TGT}}, q^{\text{GF-SRC}}, q^{\text{GF-TGT}}\}$, q^{SRC} and the two grammar-forced CS variants come from CodeMixQA (Winata et al., 2026), but q^{TGT} is generated by Qwen3-235B-A22B from q^{SRC} using the prompt in Section F.4. For short, factoid SimpleQA queries (e.g., *"Who wrote Pride and Prejudice?"* → *"누가 오만과 편견을 썼나요?"*) the translation direction is essentially deterministic and the resulting q^{TGT} has the same syntactic frame as $q^{\text{GF-TGT}}$. For more complex queries, however, the translator can pick a paraphrase whose syntactic frame differs from the CodeMixQA CS construction (e.g., choosing an active vs. passive frame, or a different connective ordering). Because of this, we treat q^{TGT} as a target-language reference for representation-level anchor comparison rather than as a strict word-order-matched twin of the CS variants; all behavioral claims in the paper are reported either per condition or as deltas relative to q^{SRC}, so they are unaffected by this mismatch. The cosine-based anchor-bias measure is also robust to surface-syntactic variation because it pools over question-content tokens within each variant separately rather than aligning tokens across variants.

F.3 Additional Code-Switching Conditions

In addition to the two grammar-forced conditions used in the main analysis, we also examine random and selective code-switching conditions from CodeMixQA (Winata et al., 2026). These conditions provide useful controls because they introduce bilingual lexical mixing without imposing the same explicit grammatical-frame contrast as GF-SRC and GF-TGT. Random switching replaces tokens without a strong linguistic selection criterion, while selective switching targets more informative lexical items. As a result, these conditions help separate the general effect of lexical mixing from the more specific effect of grammatical frame.

As shown in Table 9, both random and selective switching reduce QA performance, confirming that code-switched lexical mixing itself can be challenging for MLLMs. However, their anchor-bias values remain much closer to zero than the grammar-forced contrast reported in the main results. This suggests that arbitrary or lexically selective mixing does not by itself create the strong directional anchoring observed under GF-SRC and GF-TGT. Instead, the main contrast is driven by the grammatical frame that organizes the mixed-language question.

Condition	Anchor bias	ΔF1
RANDOM	−0.026	−5.19
SELECTIVE	−0.103	−4.37

Table 9: **Additional CS controls.** Random and selective switching degrade performance even without imposing the same grammar-frame contrast as GF-SRC/GF-TGT. These controls show that arbitrary lexical mixing can be harmful, while the grammar-forced pair isolates the stronger structural anchoring effect.

F.4 Prompt Templates

We provide the prompt templates used for constructing target-language questions and generating short answers. The templates are intentionally restrictive: the translation prompt preserves the original information needed, and the answer prompt asks for a short English answer without explanations.

Target-language question construction. For each English SimpleQA question, we use Qwen3-235B-A22B to obtain a target-language counterpart. The prompt asks the model to preserve the original information needed and output only the translated question.

Target-language question construction

System: You are a professional translator.

User: Translate the following English question into {target language}.

Constraints

- Preserve the meaning exactly.
- Preserve named entities unless they have a standard form in the target language.
- Do not add explanations.
- Output only the translated question.

Input question

{question}

English short-answer generation. For answer generation, the model receives one question at a time and is instructed to return a short English answer. We use deterministic decoding and a maximum output length of 64 tokens.

English short-answer generation

System: You are a short-answer QA system. Return only the final answer as a short phrase. Answer must be in English. No explanation.

User: {question}

G Anchor Bias Details

G.1 Anisotropy Diagnostics

Cosine similarities in hidden-state spaces can be inflated by representation anisotropy: even unrelated inputs may occupy similar directions and therefore show high background cosine similarity. This is especially important for cross-model comparisons, because different model families can have very different random-cosine baselines. To quantify this effect, we estimate a random baseline cosine for each model from unrelated question pairs.

Model	Type	Random cos.	$1-\mu$
Aya-Expanse-8B	Dense	0.7944	0.2056
Llama3.3-70B	Dense	0.8729	0.1271
Llama3.1-70B	Dense	0.8775	0.1225
Qwen3.5-4B	Dense	0.9112	0.0888
Qwen3.5-27B	Dense	0.9145	0.0855
Qwen3.5-9B	Dense	0.9320	0.0680
Mixtral-8x7B	MoE	0.8862	0.1138
Phi-3.5-MoE	MoE	0.8698	0.1302
Qwen3.6-35B-A3B	MoE	0.9379	0.0621
Qwen3-30B-A3B	MoE	0.9791	0.0209

Table 10: **Random baseline cosine for anisotropy correction.** We estimate the average cosine similarity between representations of unrelated questions for each model. High random cosine values indicate strong background alignment in the hidden-state space; $1-\mu$ is the denominator used in the random cosine normalization.

As shown in Table 10, random baseline cosine values are high across all models and exceed 0.90 for several Qwen-family models. This means that raw cosine similarity can overstate representational closeness even for unrelated inputs. We therefore subtract the model- and layer-specific random baseline before comparing EN–CS and TGT–CS similarities in the main anchor-bias. We use a random-pair baseline rather than the SRC–TGT cosine because SRC–TGT cosine is itself a content-anchored quantity we want to measure relative to chance; layer-wise min–max scaling would similarly entangle normalization with the very anchor distance under study.

G.2 Language-wise Anchor Bias

Setup. We group the matched SRC/CS/TGT triples by target language and CS condition, then compute the same normalized anchor-bias metric used in Table 1. For each language–condition cell, we average over examples and models. We also report CS F1 in the same point scale as the main tables so that representation movement and QA behavior can be read together.

Results. Table 11 shows that the anchor signal is not only a model-level effect. The GF-Src column is generally more SRC-oriented than GF-Tgt, but the strength of this split differs by language. This language-level variation explains why the aggregate AB–F1 pattern should be interpreted as a structured multilingual phenomenon rather than as a single global scalar.

G.3 Additional Anchor-Bias Variants

Table 20 reports four variants of the anchor-bias metric to verify that the GF-Src >0 / GF-Tgt <0 sign pattern is not an artefact of a specific pooling or normalization choice. AB_{raw} is the raw question-content cosine difference; AB_{nrm} normalizes by $1-b_m$ (the model's random-cosine dynamic range, shown in the second column); AB_{tok} applies per-token L2 normalization before pooling; and $\mathrm{AB}_{tok,nrm}$ combines both. All four variants agree on the sign for every model–condition pair: GF-Src is positive (SRC-anchored) for models with positive main-table bias, GF-Tgt is universally negative (TGT-anchored), and all Random and Selective conditions are negative. The normalized variants (AB_{nrm}, $\mathrm{AB}_{tok,nrm}$) amplify absolute magnitudes substantially for models with small dynamic range (e.g., Llama3.2-1B, $1-b_m$=0.0041) but do not change the ordering or direction of any entry. We use AB_{nrm} (normalized mean-pool) as the primary metric.

G.4 Frontier Model Scaling Validation

Setup. The Layer-AB and Mitigation pools used in the main paper are restricted to models whose weights are publicly available, so that the anchor-bias analysis and CANVAS mitigation can be performed on internal hidden states. A natural concern is whether code-switched degradation also persists at the closed-source frontier scale, where models substantially outperform our open pool on general QA. We address this concern by evaluating three frontier API models on the same

Language	AB^{upper}_{nrm} by CS condition				CS F1 by CS condition			
	GF-SRC	GF-TGT	Random	Selective	GF-SRC	GF-TGT	Random	Selective
Toba Batak	+4.40	+4.00	+4.03	+4.02	10.1	10.6	8.9	12.6
Hindi	+2.30	+0.53	+1.11	+0.57	15.2	14.1	16.0	15.4
Chinese (Mandarin)	+1.00	-1.65	-0.05	-0.21	13.9	9.0	13.7	16.1
Spanish	+0.21	-1.22	-0.58	-0.78	15.8	12.2	16.7	16.5
Urdu	+0.57	-1.56	-0.74	-0.77	12.8	13.5	13.1	14.8
French	+0.18	-1.25	-0.91	-0.96	13.5	12.7	10.3	9.9
Bengali	-0.28	-2.04	-1.06	-1.51	8.9	9.1	13.6	8.9
Korean	-0.26	-1.97	-1.68	-1.44	15.6	8.6	10.5	11.0
Marathi	-0.29	-2.16	-1.27	-1.68	13.6	12.0	10.0	15.3
Mean	+0.87	-0.81	-0.13	-0.31	13.3	11.3	12.5	13.4

Table 11: **Language-wise anchor bias by CS condition.** We group examples by target language and CS condition and report the same normalized anchor-bias family used in Table 1. F1 is shown on the same point scale as the main tables.

Language	Avg. F1		ΔF1 by CS condition			
	BASE	CANVAS	GF-SRC	GF-TGT	Random	Selective
Hindi	10.47	**11.43**	+0.65	+1.51	+1.23	+0.44
Marathi	8.98	**11.25**	+1.43	+3.28	+2.26	+2.13
Spanish	10.37	**10.55**	-0.10	+0.20	+0.69	-0.06
Urdu	9.49	**10.22**	+0.15	+0.96	+1.01	+0.81
Simplified Chinese	9.44	**9.91**	+0.18	+0.98	+0.29	+0.42
Korean	7.83	**9.48**	+1.49	+2.10	+1.51	+1.52
French	8.69	**8.79**	+0.02	-0.50	+0.39	+0.51
Bengali	7.37	**8.61**	+0.65	+1.52	+1.23	+1.54
Toba Batak	6.94	**7.01**	-0.33	-0.05	+0.37	+0.31
Mean	8.84	**9.70**	+0.46	+1.11	+1.00	+0.85

Table 12: **Language-wise CANVAS mitigation.** We use the same reporting convention as Table 2: Avg. F1 is averaged over four CS conditions, and condition-wise columns report CANVAS–Base F1 changes.

2,879-row CS dataset used throughout the paper: DeepSeek-V3.2 (Liu et al., 2024), Gemini 2.5 Flash (Comanici et al., 2025), and GPT-5.4-mini (Singh et al., 2025). All three models are queried via OpenRouter under identical decoding settings (greedy, `max_new_tokens` = 48) and the same system prompt as the open pool. Each model produces predictions for the EN, CS, and TGT conditions; we report F1 against the same gold answers.

Model	EN	CS	TGT	$\Delta F1_{CS-EN}$	$\Delta F1_{TGT-EN}$
DeepSeek-V3.2	19.97	16.86	11.28	−3.11	−8.69
Gemini 2.5 Flash	28.42	20.78	9.34	−7.64	−19.07
GPT-5.4-mini	19.59	17.71	16.05	−1.88	−3.54

Table 13: **Frontier model scaling check.** Three closed-source frontier models evaluated via OpenRouter on the same 2,879-row CS dataset. CS degradation ($\Delta F1_{CS-EN} < 0$) and the source-language preference ordering $F1_{TGT} < F1_{CS} < F1_{EN}$ both hold for every frontier model.

Frontier CS degradation. Table 13 reports aggregate F1 across the three conditions. Two findings stand out. First, all three frontier models exhibit CS degradation: $\Delta F1_{CS-EN}$ ranges from −1.88 (GPT-5.4-mini) to −7.64 (Gemini 2.5 Flash). Although the magnitude varies, the direction is consistent — *none* of the frontier models is immune to code-switching. Second, the source-language anchoring ordering $F1_{TGT} < F1_{CS} < F1_{EN}$ that we report for the open pool is preserved in every frontier model. The TGT−EN gap ranges from −3.54 to −19.07, indicating that source-language preference is not a small-model artifact but a property that scales with model capacity.

Open vs. frontier CS performance. A useful cross-check is the actual CS performance gap relative to model scale. Our largest open model, Qwen3.6-35B-A3B (35B parameters, ∼ 3B active per token), reaches CS F1 15.80 under direct CS answering and 17.53 with CANVAS applied (Table 19). The frontier results in Table 13 sit at 16.86 (DeepSeek-V3.2), 17.71 (GPT-5.4-mini), and 20.78 (Gemini 2.5 Flash). CANVAS therefore brings the open MoE model within 0.18 F1 of GPT-5.4-mini and ahead of DeepSeek-V3.2; only Gemini 2.5 Flash retains a meaningful 3.25 F1 lead. Given that DeepSeek-V3.2 is roughly an order of magnitude larger in total parameter count

than Qwen3.6-35B-A3B, the practical CS F1 gap is small relative to the underlying capacity difference: scaling alone does not close the CS gap, while a representation-level intervention applied to a moderate-capacity open model achieves comparable CS performance.

G.5 Lexical Language versus Word Order

Setup. We test whether word order alone can move an input across the source–target anchor axis. Starting from the Korean-target examples, we construct two same-lexicon order diagnostics using OpenRouter Qwen3-235B-A22B only to change the order of the existing lexical material. SRC-TGT$_{\text{order}}$ keeps source-language lexical material but arranges it in target-like order, while TGT-SRC$_{\text{order}}$ keeps target-language lexical material but arranges it in source-like order. We compare these probes with pure source text (SRC), pure target text (TGT), and the two grammar-forced CS conditions. We evaluate three local 7–8B-level models and report normalized anchor bias, where positive values indicate source-oriented representations and negative values indicate target-oriented representations.

Results. Table 14 shows that word order does affect the strength of anchoring, even though it does not fully determine the anchor direction. When we keep source-language words but impose target-like order, SRC-TGT$_{\text{order}}$ moves away from the pure SRC endpoint: its mean AB_{nrm} drops from $+1.000$ to $+0.647$. This means that changing only the order of the source sentence already reduces source anchoring and shifts the representation toward the target side. Conversely, when we keep target-language words but impose source-like order, TGT-SRC$_{\text{order}}$ moves slightly away from the pure TGT endpoint: its mean AB_{nrm} increases from -1.000 to -0.876. This movement is smaller than the source-to-target-order case on average, although Aya-Expanse-8B shows a relatively large shift.

The two order-counterfactual probes therefore occupy intermediate positions rather than collapsing exactly to either monolingual endpoint. SRC-TGT$_{\text{order}}$ stays between SRC and the CS conditions on the source side, while TGT-SRC$_{\text{order}}$ stays between TGT and the CS conditions on the target side. This confirms that grammatical frame is not a superficial formatting detail: changing only the order of the same lexical material is enough to move the internal representation away from its original monolingual anchor.

H CANVAS Details

H.1 Adaptive Alpha Schedule and Clip Range

Setup. CANVAS uses the adaptive interpolation rule from the main method section:

$$\alpha = \text{clip}_{[\alpha_{\min},\alpha_{\max}]}\left(\alpha_0 - \lambda\gamma_{\text{CS}}\right),$$

where γ_{CS} is the representation-alignment score computed from the clean prefill over the question span. The term $-\lambda\gamma_{\text{CS}}$ is intentional: if the CS state is already source-canvas aligned ($\gamma_{\text{CS}} > 0$), CANVAS applies a weaker correction; if it is target-oriented ($\gamma_{\text{CS}} < 0$), CANVAS increases the interpolation strength. The main paper setting is $(\alpha_0, \lambda, \alpha_{\min}, \alpha_{\max}) = (0.45, 1.5, 0.05, 0.75)$.

Schedule choice. We use this single global schedule because it balances three constraints: it improves average CS F1, preserves the intended monotonic behavior, and keeps the intervention bounded. The center $\alpha_0 = 0.45$ gives a moderate interpolation when the alignment score is near neutral, while $\lambda = 1.5$ creates a visible separation between source-aligned and target-oriented inputs before clipping. A smaller λ would flatten that adaptivity; a much larger λ would make moderate alignment changes saturate the clip bounds more often. We choose the clip range $[0.05, 0.75]$ to keep the lower regime close to an identity update and the upper regime short of target-state replacement. We do not tune language-specific, condition-specific, or model-specific constants.

Parameter interpretation. The four values control different aspects of the same bounded controller. Raising α_0 strengthens the default update for inputs with $\gamma_{\text{CS}} \approx 0$, whereas lowering it makes neutral cases closer to direct CS answering. Raising $\alpha_{\min}$ forces more interpolation even for strongly source-aligned cases, while lowering it makes the low end more nearly identity-like. Raising $\alpha_{\max}$ permits stronger source-canvas mixing for target-oriented cases, while lowering it caps those corrections earlier; at our upper bound, at least one quarter of the original target-token representation is preserved. Table 15 summarizes these roles.

Model	SRC	SRC-TGT$_{\text{order}}$	CS GF-SRC	CS GF-TGT	TGT-SRC$_{\text{order}}$	TGT
Aya-Expanse-8B	1.000	0.445	0.313	-0.606	-0.765	-1.000
Llama3.1-8B	1.000	0.692	0.032	-0.641	-0.910	-1.000
Qwen3.5-4B	1.000	0.804	0.076	-0.627	-0.954	-1.000
Mean	1.000	0.647	0.140	-0.625	-0.876	-1.000

Table 14: **Lexical language versus word order on Korean-target examples.** We report normalized anchor bias. SRC-TGT$_{\text{order}}$ keeps source-language words but places them in target-like order; TGT-SRC$_{\text{order}}$ keeps target-language words but places them in source-like order. Positive values indicate source-oriented representations; negative values indicate target-oriented representations.

Component	Value	Interpretation
Base strength α_{base}	0.45	Sets α when $\gamma_{\text{CS}} \approx 0$. Larger values strengthen neutral cases.
Adaptivity scale λ	1.5	Scales $-\lambda\gamma_{\text{CS}}$. Larger values separate SRC-/TGT-oriented cases more sharply.
Lower clip $\alpha_{\min}$	0.05	Keeps strongly SRC-aligned cases near identity. Larger values force a stronger minimum update.
Upper clip $\alpha_{\max}$	0.75	Prevents full target-state replacement. Larger values allow stronger source-canvas mixing.

Table 15: **CANVAS adaptive-α schedule.** We use one global bounded controller for all languages, conditions.

Observed distribution. We visualize the effective α distribution from the CANVAS representation-analysis dump in Figure 6. We find that the learned behavior matches the intended controller: GF-SRC has a higher source-token ratio and receives a weaker intervention on average ($\alpha = 0.453$), while GF-TGT is more target-heavy and receives a stronger intervention ($\alpha = 0.608$). We also observe that random and selective switching falls between these two endpoints.

H.2 Layer-wise CANVAS Analysis

Setup. We analyze where the CANVAS-induced representation shift appears across depth. CANVAS intervenes only after the control-layer boundary, so the expected pattern is layer-local: lower layers should remain close to the base CS trajectory, while upper layers should show an increasing source-directed shift after entering $\mathcal{L}_{\text{ctrl}}$.

Results. Figure 7 shows this trajectory as a heatmap over the relative position inside $\mathcal{L}_{\text{ctrl}}$. The left panel reports $\Delta\cos_{\text{SRC}}$, and the right panel reports $-\Delta\cos_{\text{TGT}}$, so darker colors in both panels indicate a stronger source-ward correction. The shift becomes larger toward the later part of the control interval, supporting the mechanism implied by the anchor-bias analysis: CANVAS redirects the late-layer CS state rather than globally rewriting the entire prompt representation.

H.3 Source-heavy Checks

CANVAS preserves source-heavy inputs. We next check cases that are already mostly source-language text. We select examples with a detected source-token ratio $\rho_{\text{SRC}} \geq 0.80$ from the full CANVAS runs and split them into three diagnostic rows. "All source-heavy" is the full selected set. "Steered source-heavy" keeps only examples where the detector still finds at least one target-language token span ($|C_{\text{tgt}}| > 0$), so CANVAS actually has hidden states to modify. "Identity source-heavy" contains examples with no detected target span ($|C_{\text{tgt}}| = 0$), where CANVAS should leave the prompt effectively unchanged. Table 16 shows that the full source-heavy set uses a moderate mean α (0.445) and preserves performance: average F1 changes from 9.08 to 9.12, with 97.3% of examples showing no F1 drop. Even in the stricter steered subset, where CANVAS actually intervenes, the average change remains small (ΔF1=-0.21), and 96.0% of examples show no drop.

H.4 Token-Level Language Tagging for CANVAS

CANVAS requires identifying which question tokens belong to the source-language span and which belong to the target-language span. We therefore apply lightweight token-level language tagging before computing the source canvas and target anchor. Each token in the question-content span is assigned one of three labels: SRC, TGT, or OTHER. SRC denotes source-language to-

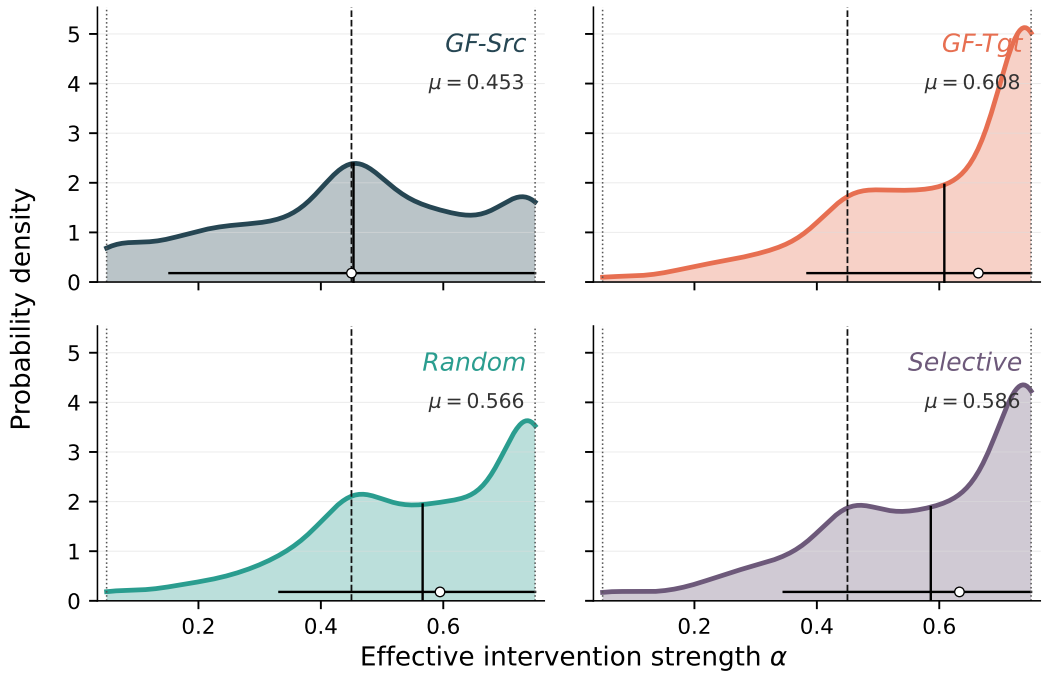

Figure 6: **Effective α distribution under the validation-selected CANVAS rule.** We plot the area-normalized probability density of the effective interpolation strength used by CANVAS. We mark the validation-selected center $\alpha_0 = 0.45$ and clip range $[0.05, 0.75]$ with dashed and dotted vertical lines, respectively. We assign weaker updates to SRC-heavy GF-SRC examples and stronger updates to TGT-heavy GF-TGT examples, while random and selective switching remain between these two endpoints.

Subset	Definition	Mean ρ_{SRC}	Mean α	Base F1	CANVAS F1	ΔF1	No-drop (%)	Fixed / Lost
All source-heavy	$\rho_{\text{SRC}} \geq 0.80$	0.991	0.445	9.08	9.12	+0.04	97.3	265/246
Steered source-heavy	$\rho_{\text{SRC}} \geq 0.80,\ \lvert C_{\text{tgt}} \rvert > 0$	0.861	0.366	15.04	14.83	-0.21	96.0	18/26
Identity source-heavy	$\rho_{\text{SRC}} \geq 0.80,\ \lvert C_{\text{tgt}} \rvert = 0$	1.000	0.450	8.68	8.74	+0.06	97.4	247/220

Table 16: **CANVAS on source-heavy CS inputs.** We split high-source-ratio examples into all cases, cases with at least one target-token span that CANVAS can steer, and cases with no detected target span where CANVAS should behave as an identity operation. No-drop reports the percentage of examples where CANVAS F1 is at least direct CS F1. Fixed/Lost counts base-zero to CANVAS-positive cases and base-positive to CANVAS-zero cases.

kens, which correspond to English tokens in our main English–target-language setting. TGT denotes target-language tokens, and OTHER denotes punctuation, numbers, whitespace artifacts, or language-neutral tokens.

Because chat templates include system instructions and assistant-prefix tokens that are shared across all inputs, we first locate the raw user question inside the full templated prompt. When tokenizer offset mappings are available, we use them to identify the token positions overlapping with the question span. Tokens outside this span are not used for canvas construction, alignment estimation, or interpolation. This ensures that the CANVAS signal is computed from the code-switched question itself rather than from shared prompt text. For non-Latin target languages, we use Unicode script information. Tokens containing only Hangul (U+AC00–U+D7A3), CJK (U+4E00–U+9FFF), Devanagari (U+0900–U+097F), Arabic, or Bengali-script characters are labeled as TGT. Tokens containing only Latin-script alphabetic characters are labeled as SRC, except for Latin-script target languages. For Spanish, French, and Toba Batak, script information alone is insufficient because both English and the target language use Latin characters. In these cases, we use the Lingua probabilistic language detector* restricted to the {English, target} language pair to assign each Latin-script token to SRC or TGT. Tokens with mixed Latin and non-Latin characters are conservatively labeled as TGT, while punctuation and empty or non-alphabetic surfaces are labeled as OTHER.

*https://github.com/pemistahl/lingua-py

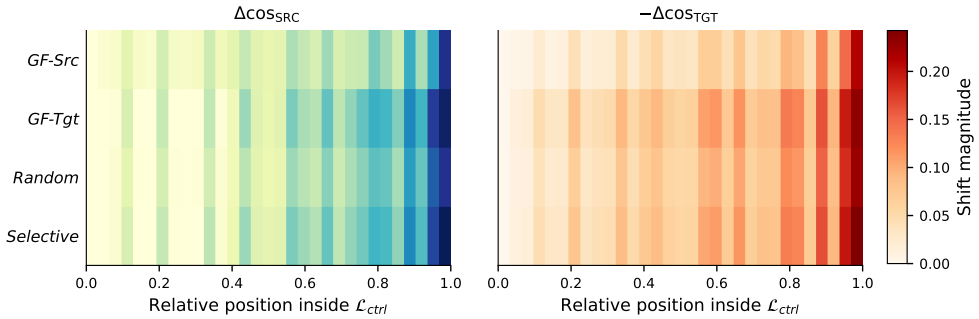

Figure 7: **Layer-wise CANVAS effect.** We plot the magnitude of the source-ward shift across the relative position inside $\mathcal{L}_{\text{ctrl}}$. Darker cells indicate larger $\Delta\cos_{\text{SRC}}$ or larger $-\Delta\cos_{\text{TGT}}$. The effect is strongest near the later control layers, matching the layer-wise anchoring analysis.

For BPE tokenizers, raw token strings may not directly reflect the original Unicode surface form. We therefore decode each token ID back into its surface string before applying script-based tagging. If either the source or target span contains too few tokens, CANVAS skips the intervention for that example and falls back to ordinary decoding.

H.5 Hidden-State Extraction for CANVAS

CANVAS computes its source canvas, target anchor, and alignment score from the hidden states of the user question, not from the full chat-templated input. For each example, the model input is the concatenation of system-prompt tokens, user-question tokens, and assistant-prefix tokens:

$$\mathbf{x} = [\mathbf{x}_{\text{sys}}; \mathbf{x}_{\text{q}}; \mathbf{x}_{\text{asst}}], \tag{15}$$

where $\mathbf{x}_{\text{sys}}$ contains the shared QA instruction, $\mathbf{x}_{\text{q}}$ contains the actual code-switched question, and $\mathbf{x}_{\text{asst}}$ contains model-specific generation-template tokens such as assistant role markers. Since $\mathbf{x}_{\text{sys}}$ and $\mathbf{x}_{\text{asst}}$ are shared across inputs and can obscure the representation of the CS question itself, all CANVAS signals are computed only from token positions overlapping with $\mathbf{x}_{\text{q}}$.

Concretely, we run a clean forward pass with hidden-state outputs enabled. For a model with L layers, sequence length T, and hidden size d, this gives

$$H_M(q^{\text{CS}}) \in \mathbb{R}^{L\times T\times d}. \tag{16}$$

We then identify the token positions corresponding to the user question using tokenizer offsets when available. These question positions are further tagged as source, target, or other, producing C_{src}, C_{tgt}, and C_{oth}.

For each selected upper layer $\ell \in \mathcal{L}_{\text{ctrl}}$, we gather the hidden states at source-language positions and average them along the token dimension to obtain one layer-specific source canvas:

$$\mathbf{c}_\ell^{\text{src}} = \frac{1}{|C_{\text{src}}|} \sum_{t\in C_{\text{src}}} H_M(q^{\text{CS}})[\ell, t, :]. \tag{17}$$

The target anchor is computed analogously:

$$\mathbf{a}_\ell^{\text{tgt}} = \frac{1}{|C_{\text{tgt}}|} \sum_{t\in C_{\text{tgt}}} H_M(q^{\text{CS}})[\ell, t, :]. \tag{18}$$

Thus, each selected layer has its own source canvas and target anchor. For example, if two source-token hidden states at a layer are

$$[a, b, c, d] \quad \text{and} \quad [e, f, g, h],$$

their mean source canvas is

$$\left[\frac{a+e}{2}, \frac{b+f}{2}, \frac{c+g}{2}, \frac{d+h}{2}\right].$$

This layer-specific averaging is repeated independently for every $\ell \in \mathcal{L}_{\text{ctrl}}$, so CANVAS aligns target-token states to the source canvas of the same layer rather than to a single global vector.

H.6 CANVAS Layer-span Ablation

Evaluation protocol. We compare direct code-switched answering (BASE) with CANVAS on the same short-answer QA setting used in the previous sections. For the intervention study, we evaluate MLLMs across four CS conditions: GF-SRC, GF-TGT, random switching, and selective switching. We report average F1 over the four CS conditions and condition-wise F1 changes relative to BASE. We also include a random-direction baseline, $\text{CANVAS}_{\text{rand}}$, which keeps the same adaptive α controller and 30% layer budget as CANVAS but replaces the source–target representation direction with a random interpolation direction.

Model	CANVAS-10	CANVAS-30	CANVAS-50	CANVAS-70	CANVAS-90	$\text{CANVAS}_{\text{rand}}$
Aya-Expanse-8B	6.03	6.30	**7.11**	6.70	4.91	5.99
Llama3.1-8B	7.11	7.68	**8.03**	7.25	5.28	6.72
Mistral-7B	4.92	5.13	**5.15**	4.27	3.38	4.96
Phi-3.5-MoE	7.96	8.58	**8.81**	8.30	6.17	7.93
Qwen3.5-27B	14.93	**15.03**	14.45	12.85	8.95	14.22
Qwen3.5-4B	7.80	**7.98**	7.81	7.03	4.83	7.35
Qwen3.5-9B	10.04	**10.24**	9.52	8.72	5.56	10.01
Qwen3-30B-A3B	7.17	**8.13**	8.10	7.33	6.00	6.93
Mean	8.24	**8.64**	8.62	7.81	5.64	8.01

Table 17: **Layer-span ablation for CANVAS.** We report average F1 over the four CS conditions. CANVAS-k applies the adaptive intervention to the last $k\%$ of layers. $\text{CANVAS}_{\text{rand}}$ uses the same adaptive α and the same 30% layer span as CANVAS-30, but replaces the source–target representation direction with a random interpolation direction.

Setup. We vary the intervention span while keeping the current adaptive schedule fixed, $\alpha = \text{clip}_{[0.05,0.75]}(0.45 - 1.5\gamma_{\text{CS}})$. CANVAS-$k$ applies the intervention only to the last $k\%$ of layers. We choose the 30% span as the main setting because it matches the upper-layer region where the anchor-bias analysis shows stable language-frame commitment, while still leaving the lower and middle layers to encode the original CS question without intervention. Thus, the ablation tests whether CANVAS should act as a late anchor correction or as a broad representation rewrite.

Results. Table 17 supports this choice. CANVAS-30 gives the best average F1, while CANVAS-50 performs nearly on par and can be viewed as a stronger, wider-span variant. We use CANVAS-30 as the main setting because it achieves the same level of improvement with a narrower intervention span, making it the minimal effective intervention: it targets the late anchor-commitment region while leaving more of the lower and middle-layer CS composition untouched. In contrast, performance drops for CANVAS-70 and collapses for CANVAS-90. This pattern suggests that increasing the span beyond the late control region does not simply strengthen the same correction. Once the intervention reaches too far into lower layers, it perturbs lexical and compositional processing before the model has formed a stable high-level CS state; the source-direction update then competes with question encoding rather than correcting a late anchor. The random-direction baseline is also lower than CANVAS-30 for every evaluated model, so the gain does not come from arbitrary hidden-state mixing; it depends on applying the source–target anchor direction in the late control region identified by the analysis.

H.7 Language-wise CANVAS Mitigation

Setup. We reuse the main CANVAS outputs and group examples by target language. Following Table 2, Avg. F1 is the mean over the four CS conditions, and each condition column reports CANVAS–Base F1 change.

Results. Table 12 shows that CANVAS improves most language groups, with the largest gains on languages where the direct CS baseline is weaker. The gains are not restricted to a single CS construction: improvements appear across grammar-forced, random, and selective switching, although some high-resource or already-stable cases show smaller or slightly negative changes.

H.8 CANVAS Case Studies

Overview. We pair one recovery with one failure for the same Llama-3.1-8B model. Each figure combines the CS question and answer with layer-wise cosine similarity to the SRC and TGT anchors and the resulting anchor bias. The pair shows what CANVAS changes internally and also bounds the claim: source-ward movement is useful on average, but it is not an answer verifier.

Success Case The success case uses a Korean–English CS question with 11 Korean tokens and 6 English tokens. Its alignment score is strongly target-oriented ($\gamma = -0.396$), so CANVAS selects $\alpha = 0.75$ and steers Korean-token hidden states in layers 22–31. Figure 8 shows that direct CS answering returns **2019**, while CANVAS recovers the gold year **1660**.

The lower panels show the internal correction behind the answer change. Similarity to the EN anchor rises under CANVAS, similarity to the KO anchor falls across the steered layers, and anchor bias flips from negative to positive from layer 23

Success case: CANVAS fixes a CS answer (Llama-3.1-8B, Korean–English)

Model	Llama-3.1-8B-Instruct
CS input	Süleymaniye는 대화재에서 몇 년도에 damage되었나요? *("In what year was the Süleymaniye [Mosque] damaged in the Great Fire?")*
Gold answer	**1660**

Base CS answer	CANVAS answer	Diagnostics
2019	1660	α=0.75; γ=−0.396; EN tokens = 6; KO tokens = 11; steered layers = 10 (layers 22–31 of 32)

Token breakdown. The question contains 11 Korean tokens (TGT) and only 6 English tokens (SRC), giving γ=−0.396 (strongly TGT-anchored). CANVAS detects this imbalance and applies α=0.75 interpolation toward the English anchor in the top 10 layers.

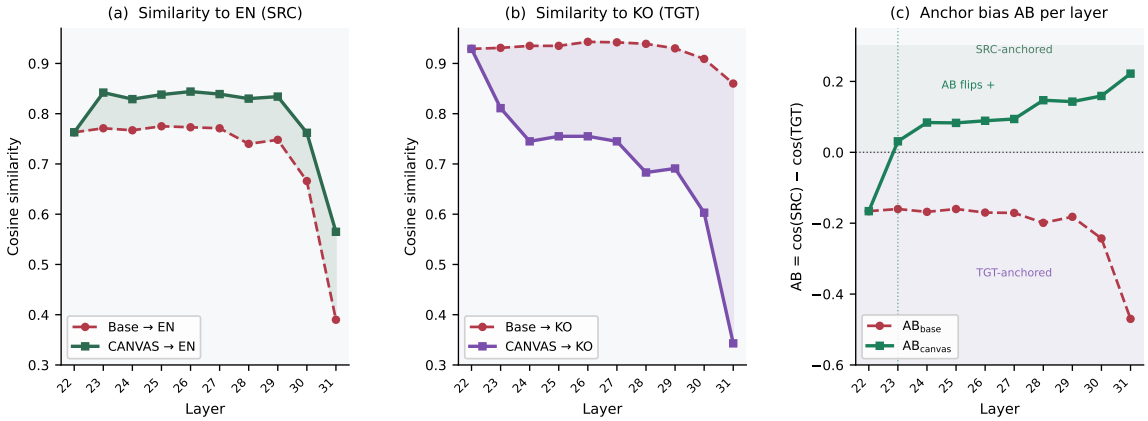

Figure 8: **CANVAS success case: representation trajectory.** *Top*: qualitative example showing the wrong base answer (2019) and the correct CANVAS answer (1660) for a Korean–English CS question. *Bottom (a–c)*: layer-by-layer hidden-state cosine similarity to the EN anchor (SRC) and KO anchor (TGT) for the CS prompt, and the resulting anchor bias $\mathrm{AB} = \cos(\mathbf{h}_{\mathrm{EN}}, \mathbf{h}_{\mathrm{CS}}) - \cos(\mathbf{h}_{\mathrm{KO}}, \mathbf{h}_{\mathrm{CS}})$. In the base run, AB stays negative throughout (KO-anchored). CANVAS flips AB to positive from layer 23 onward (EN-anchored), which is sufficient for the model to recover the correct answer.

onward. This example, therefore, connects the intended source-ward intervention to a concrete answer recovery.

Failure Case The failure case uses a Spanish–English GF-Tgt question with 14 Spanish tokens and 6 English tokens. CANVAS sees a target-oriented alignment score (γ=−0.234) and applies its maximal interpolation (α=0.75). In Figure 9, direct CS answering already returns the gold year **2018**, but CANVAS changes the answer to **2017**.

The layer trace explains the failure boundary. CANVAS reduces Spanish similarity and pushes anchor bias toward the source side, but the shift is incomplete at the top of the stack: anchor bias briefly crosses zero in the early steered layers and returns negative by the final layer. At the same time, the direct CS state is already sufficient for the correct answer. The strong source-ward correction, therefore, perturbs an already correct answer state without producing a stable final source-oriented state. This case shows why source-anchor correction improves CS answering on average, but is not a correctness certificate for every example.

H.9 Model-wise CANVAS Analysis

Setup. We group the CANVAS representation analyses by model. For each model, we compare the code-switched representation before and after intervention against two reference anchors: the source-language question representation and the target-language question representation. We compute the metrics over the control-layer interval $\mathcal{L}_{\mathrm{ctrl}}$ and mean-pool question-content tokens, so the comparison focuses on the question state rather than the generated answer.

Metrics. Table 18 reports the model-wise movement signals. $\Delta\mathrm{AB}$ measures whether CANVAS increases source-oriented anchor bias. $\Delta\cos_{\mathrm{SRC}}$ measures the change in similarity to the SRC and TGT anchors. The projection ratio η measures how much of the intervention displacement lies along the SRC–TGT axis; a positive η means that

Failure case: CANVAS changes a correct CS answer (Llama-3.1-8B, Spanish–English)

Model	Llama-3.1-8B-Instruct
CS input	En qué año la diseñadora de moda Mowalola Ogunlesi abandonó Central Saint Martins? *("In what year did fashion designer Mowalola Ogunlesi drop out of Central Saint Martins?")*
Gold answer	**2018**

Base CS answer	CANVAS answer	Diagnostics
2018	2017	α=0.75; γ=−0.234; EN tokens = 6; ES tokens = 14; steered layers = 10 (layers 22–31 of 32)

Token breakdown. The question contains 14 Spanish tokens (TGT) and 6 English tokens (SRC), so CANVAS applies its strongest source-ward interpolation. Here the direct CS answer is already correct, but the intervention changes the generated year.

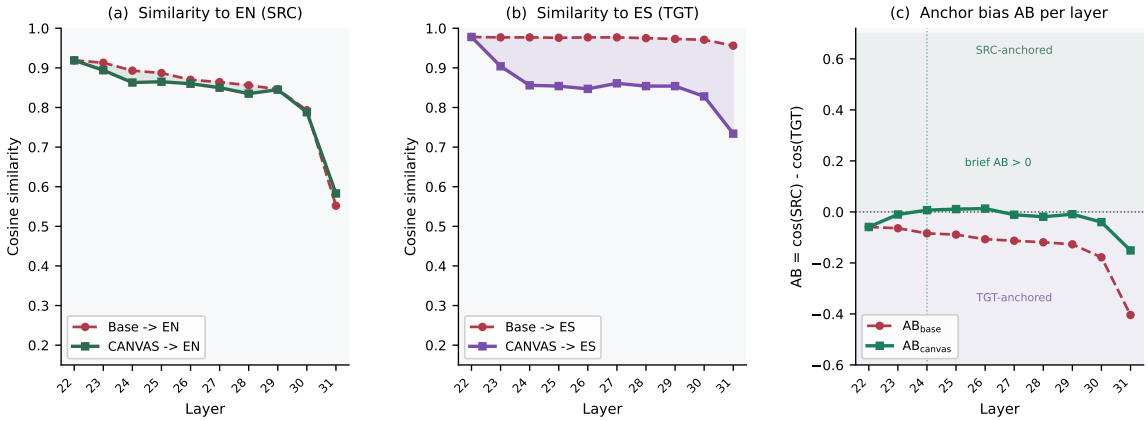

Figure 9: **CANVAS failure case: representation trajectory.** *Top*: qualitative example where direct CS answering returns the gold year (2018), while CANVAS outputs a wrong year (2017). *Bottom (a–c)*: layer-by-layer cosine similarity to the EN anchor (SRC), similarity to the Spanish anchor (TGT), and anchor bias $\mathrm{AB} = \cos(\mathbf{h}_{\mathrm{EN}}, \mathbf{h}_{\mathrm{CS}}) - \cos(\mathbf{h}_{\mathrm{ES}}, \mathbf{h}_{\mathrm{CS}})$. CANVAS moves the CS state toward the SRC side but does not produce a stable final source-oriented state, while also changing an already correct answer.

the movement is in the SRC direction.

Results. Across analyzed models, CANVAS consistently increases AB, increases cosine similarity to the SRC anchor, and yields positive projection ratios on the SRC–TGT axis. Figure 11 shows the same analysis as a model-wise representation grid. Each row corresponds to one model, and each column corresponds to a grammar-forced CS condition. The arrows summarize the movement from the base CS centroid to the CANVAS centroid, making it possible to compare whether the same source-directed shift appears across model families.

Model-wise diagnostics. We provide model-wise versions of the two main CANVAS diagnostics. Figure 12 decomposes the signed projection ratio η by model and CS condition. The mass of each density remains mostly to the right of zero, and $\Pr(\eta > 0)$ stays above 90% for every analyzed model. Figure 13 decomposes the adaptive controller by model. Within each model, a lower source-token ratio ρ generally receives a larger effective interpolation strength α, showing that the validation-selected schedule remains adaptive.

H.10 CANVAS Cost Analysis

Setup. We measure the wall-clock cost of CANVAS on the same full CS evaluation set used for the main intervention results. For each model and example, we record direct CS answering latency and CANVAS latency, then compute the empirical cost ratio by dividing CANVAS latency by direct CS latency. This is an end-to-end latency measurement, not a per-token FLOP estimate.

Results. Table 19 reports the model-wise cost. CANVAS improves average F1 from 8.79 to 9.61 over the 13-model pool while using 1.13× the latency of direct CS answering on average. In absolute time, CANVAS takes 866.9 ms per example on average, i.e., less than one second per question. This is only 24.1 ms above the direct CS answering average in the same model pool. The overhead remains small because CANVAS reuses the same model and applies a hidden-state intervention dur-

Model	ΔAB	$\Delta\cos_{\text{SRC}}$	η	$\Pr(\eta > 0)$	Mean α
Aya-Expanse-8B	0.143	0.036	0.160	92.0	0.542
Llama3.1-8B	0.159	0.059	0.214	93.9	0.562
Mistral-7B	0.224	0.087	0.171	92.1	0.613
Phi-3.5-MoE	0.071	0.026	0.121	90.4	0.548
Qwen3.5-27B	0.111	0.032	0.248	93.8	0.523
Qwen3.5-4B	0.156	0.050	0.243	94.1	0.594
Qwen3.5-9B	0.139	0.047	0.238	94.3	0.576
Qwen3-30B-A3B	0.088	0.029	0.192	93.6	0.469

Table 18: **Model-wise representation movement under CANVAS.** ΔAB measures the increase in source-oriented anchor bias, and $\Delta\cos_{\text{SRC}}$ measures the change in similarity to the source-language anchor. η measures the signed projection of the displacement on the SRC–TGT axis, and $\Pr(\eta > 0)$ reports the percentage of examples whose displacement has a source-ward component.

Model	Avg. F1			Mean latency			Generation diagnostics			Tail latency	
	BASE	CANVAS	ΔF1	Base ms	CANVAS ms	Cost$\times$	Base len.	CANVAS len.	Faster ex.	Base p95	CANVAS p95
Aya-Expanse-8B	5.99	6.88	+0.89	334.8	404.2	1.21	4.3	5.0	31.1%	1216	1554
Mistral-7B	4.53	5.06	+0.54	644.9	583.2	0.90	8.1	7.6	37.5%	1633	1576
Qwen3.5-4B	7.58	8.14	+0.56	444.3	528.7	1.19	2.6	2.4	26.7%	862	888
Qwen3.5-9B	9.89	10.24	+0.35	442.1	544.7	1.23	2.4	2.3	18.7%	777	898
Qwen3.5-27B	14.14	15.40	+1.25	975.1	1166.4	1.20	2.8	2.7	17.1%	1855	2100
Llama3.2-1B	2.57	3.20	+0.63	137.4	138.3	1.01	3.3	3.1	38.9%	460	399
Llama3.1-8B	6.21	7.79	+1.59	279.2	288.6	1.03	3.5	3.1	27.8%	755	700
Llama3.1-70B	12.89	13.20	+0.31	1207.7	1518.0	1.26	2.4	2.4	13.8%	2212	2622
Llama3.3-70B	12.96	13.61	+0.65	1157.1	1466.7	1.27	2.3	2.3	13.3%	1974	2345
Mixtral-8x7B	6.77	6.91	+0.13	3167.0	1853.3	0.59	13.3	12.6	86.2%	5762	3467
Phi-3.5-MoE	7.99	8.82	+0.83	757.0	824.6	1.09	4.2	4.1	11.5%	2337	2331
Qwen3-30B-A3B	6.91	8.11	+1.20	653.2	905.0	1.39	2.9	2.9	5.4%	1416	1784
Qwen3.6-35B-A3B	15.80	17.53	+1.73	756.7	1047.7	1.38	2.3	2.2	2.3%	1177	1619
Mean	8.79	9.61	+0.82	842.8	866.9	1.13	4.2	4.1	25.4%	1726	1714

Table 19: **CANVAS cost analysis.** We measure empirical wall-clock latency per example for direct CS answering and CANVAS on the same evaluation set. Cost$\times$ is CANVAS latency divided by direct CS latency. Length is a lightweight token proxy computed from generated answer strings, Faster ex. is the percentage of examples where CANVAS is faster than direct CS answering, and p95 reports the 95th-percentile latency in milliseconds.

ing prefill, instead of calling an additional model.

Why some ratios are below one. Some models show empirical cost ratios below 1.0 even though CANVAS adds a clean prefill and a steered prefill. This happens because the reported latency includes autoregressive answer generation. If CANVAS changes the hidden state so that the model emits a shorter answer or reaches EOS earlier, the shorter decoding phase can offset the extra prefill pass. The integrated diagnostics in Table 19 support this interpretation: Mistral-7B and Mixtral-8x7B have cost ratios below one, shorter CANVAS outputs on average, and lower 95th-percentile latency under CANVAS. Mixtral-8x7B is especially clear: CANVAS is faster on 86.2% of examples and reduces p95 latency from 5762 ms to 3467 ms. We therefore interpret ratios below one as an empirical decoding-length effect, not as a claim that CANVAS is cheaper per forward pass.

Interpretation. The main takeaway is that CANVAS provides a consistent F1 gain with modest wall-clock overhead. Since the method adds sub-second latency per example on average and does not require an external model call, we view it as practical for real-time or interactive CS question answering settings.

H.11 Additional Movement Analysis

We first test whether CANVAS changes the internal representation in the intended direction. We aggregate examples across the eight models used for the representation analysis. For each example, we extract four question-level representations from the control layers: the source-only representation, the target-only representation, the CS representation without intervention, and after CANVAS. We mean-pool question-content token states within each layer and then average over the selected control layers, so the metric reflects the question representation rather than the generated answer. We measure the signed projection ratio $\eta = \langle \mathbf{r}_{\text{CS}}^{\text{CANVAS}} - \mathbf{r}_{\text{CS}}^{\text{BASE}}, \mathbf{r}_{\text{SRC}} - \mathbf{r}_{\text{TGT}} \rangle / \|\mathbf{r}_{\text{SRC}} - \mathbf{r}_{\text{TGT}}\|_2^2$, where $\eta > 0$ indicates source-ward movement and $\eta < 0$ indicates target-ward movement. As shown in Figure 10, the distribution lies

Table 20: **Anchor-bias metric variants.** AB_{raw} is the question-content-only cosine anchor bias ($\cos(\mathbf{h}_{\mathrm{SRC}}, \mathbf{h}_{\mathrm{CS}}) - \cos(\mathbf{h}_{\mathrm{TGT}}, \mathbf{h}_{\mathrm{CS}})$, mean-pooled over question tokens). AB_{nrm} divides AB_{raw} by $1-b_m$, the model-specific random-cosine dynamic range. AB_{tok} L2-normalizes each token vector before pooling; $\mathrm{AB}_{tok,nrm}$ additionally applies the same normalization. Main tables report AB_{nrm}. All values are verified against source data.

Model	$1-b_m$	Cond.	Mean-pool		Token-norm	
			AB_{raw}	AB_{nrm}	AB_{tok}	$\mathrm{AB}_{tok,nrm}$
Qwen3.5-4B	0.0636	GF-SRC	+0.038	+0.59	+0.042	+0.66
		GF-TGT	−0.279	−4.39	−0.276	−4.33
		RAND	−0.142	−2.24	−0.138	−2.17
		SEL	−0.196	−3.08	−0.193	−3.03
Mistral-7B	0.0454	GF-SRC	−0.175	−3.86	−0.160	−3.51
		GF-TGT	−0.450	−9.90	−0.440	−9.69
		RAND	−0.342	−7.52	−0.328	−7.23
		SEL	−0.366	−8.06	−0.354	−7.79
Llama3.1-8B	0.0254	GF-SRC	−0.011	−0.43	−0.015	−0.60
		GF-TGT	−0.381	−14.97	−0.385	−15.12
		RAND	−0.235	−9.25	−0.239	−9.40
		SEL	−0.277	−10.87	−0.281	−11.03
Aya-Expanse-8B	0.0719	GF-SRC	+0.030	+0.42	+0.041	+0.57
		GF-TGT	−0.434	−6.04	−0.427	−5.94
		RAND	−0.273	−3.79	−0.262	−3.65
		SEL	−0.316	−4.40	−0.308	−4.28
Qwen3.5-9B	0.0211	GF-SRC	+0.036	+1.72	+0.037	+1.75
		GF-TGT	−0.272	−12.93	−0.267	−12.68
		RAND	−0.138	−6.57	−0.135	−6.40
		SEL	−0.187	−8.86	−0.183	−8.69
Qwen3.5-27B	0.0206	GF-SRC	+0.018	+0.87	+0.019	+0.94
		GF-TGT	−0.290	−14.10	−0.284	−13.76
		RAND	−0.155	−7.54	−0.151	−7.34
		SEL	−0.202	−9.78	−0.197	−9.55
Llama3.1-70B	0.0203	GF-SRC	−0.104	−5.15	−0.104	−5.13
		GF-TGT	−0.472	−23.29	−0.469	−23.14
		RAND	−0.338	−16.66	−0.335	−16.53
		SEL	−0.374	−18.44	−0.371	−18.31
Llama3.3-70B	0.0186	GF-SRC	−0.060	−3.24	−0.061	−3.27
		GF-TGT	−0.446	−23.99	−0.443	−23.83
		RAND	−0.295	−15.90	−0.293	−15.79
		SEL	−0.337	−18.11	−0.334	−17.99

mostly on the positive side for all CS conditions. Across the 32 model–condition cells, CANVAS shifts the anchor bias *toward the source side* in every cell and increases SRC similarity in every cell. Using the same upper-layer anchor bias defined in Section 3, CANVAS moves $\mathrm{AB}^{\mathrm{upper}}$ source-ward by $+0.136$ on average (range $+0.071$ to $+0.224$ across the eight models; per-model values in Table 18), so the shift is consistently in the SRC direction. The projection metric also follows the intended direction: mean η is positive across model–condition cells ($+0.1983$), and 93.0% of examples have $\eta > 0$, indicating that most CANVAS updates move CS representations source-ward. These results show that CANVAS not only improves QA scores, but also moves CS hidden states along the intended internal anchor direction.

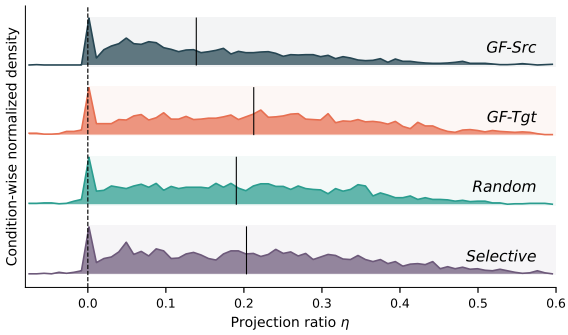

Figure 10: **Source-ward representation movement.** Signed projection ratio η of the CANVAS displacement onto the SRC–TGT axis ($\eta > 0$: source-ward; $\eta < 0$: target-ward).

H.12 Cumulative Recovery Trajectory

Setup. The per-layer projection η_ℓ measures the fraction of the SRC–TGT representational gap that CANVAS closes at control layer ℓ. Reading these layer-wise contributions cumulatively answers a

Relative position r	Mean $S(r)$
0.0	0.000
0.1	0.127
0.2	0.306
0.3	0.516
0.4	0.749
0.5 (midpoint)	1.001
0.6	1.261
0.7	1.536
0.8	1.819
0.9	2.096
1.0 (final)	2.359

Table 21: **Aggregate cumulative recovery along the control window.** $r \in [0, 1]$ is the relative position within each model's control-layer interval, and $S(r)$ is the cumulative projection up to that position averaged across 7,195 examples from eight models. The mean cumulative shift reaches exactly 1.0 at the window midpoint and continues to grow to 2.36 by the final control layer.

complementary question: *how far along the SRC–TGT axis does the intervention carry the CS representation by each step?* For a model with control window $\mathcal{L}_{\text{ctrl}} = [\ell_0, \ell_0 + N - 1]$, we define the cumulative recovery at relative position $r \in [0, 1]$ as

$$S(r) = \sum_{\ell=\ell_0}^{\ell_0+\lfloor r(N-1)\rfloor} \eta_\ell .$$

A value $S(r) \geq 1$ means that the intervention has accumulated a shift equal to or larger than the full SRC–TGT representational distance by relative position r. Using relative position rather than absolute layer count makes the trajectory comparable across models whose control windows have different lengths (N=10 for the 32-layer models, N=14 for Qwen3-30B-A3B, and N=19 for Qwen3.5-27B).

Aggregate trajectory. Table 21 reports $\bar{S}(r)$ across all 7,195 examples from the eight models used in the representation analysis. The mean cumulative shift grows monotonically and reaches exactly $\bar{S}(0.5)$=1.001 at the window midpoint, then continues to $\bar{S}(1.0)$=2.36 by the final control layer. Per-layer contributions therefore *compound* rather than cancel: by the time the intervention has used half of its allotted layers, the average representation has on aggregate already traversed the full SRC–TGT gap.

Crossing distribution. 71.7% of examples cross S=1 at some point inside the control window. Among those, the median first-crossing position is r^*=0.46, and the mean is 0.53 — closely aligned with the aggregate midpoint behaviour. Only 6.4% cross within the first 30% of the window and 37.7% within the first 50%, so most trajectories use the second half of the control window to complete the traversal. This justifies the 30% layer budget: a shorter span would leave the majority of examples short of full traversal, while a longer span risks perturbing lower-layer composition (see Appendix H.6).

Model variance. Table 22 reports the same statistics per model. Trajectory length varies widely with capacity: Qwen3.5-27B accumulates the largest final shift ($\bar{S}(1.0)$=4.72) with 71% of inputs crossing by the midpoint, while Phi-3.5-MoE accumulates the smallest ($\bar{S}(1.0)$=1.21) with only 9% crossing that early. Larger Qwen models show both higher cumulative magnitudes and earlier crossings; Llama-3.1-8B reaches the gap relatively quickly (r^*=0.44) but at a smaller magnitude than the 27B Qwen. Despite this spread, every model except Phi-3.5-MoE drives more than 65% of its examples past full traversal.

Condition dependence. Table 23 shows the same trajectory broken down by CS condition. Cumulative shift is largest for GF-TGT ($\bar{S}(1.0)$=2.58, 76% ever cross) and smallest for GF-SRC ($\bar{S}(1.0)$=1.96, 63% ever cross), with random and selective switching in between. This ordering mirrors the adaptive α schedule: target-heavy inputs receive stronger interventions, and the stronger interventions translate directly into longer cumulative trajectories along the SRC–TGT axis.

Interpretation. The cumulative analysis supports three conclusions. First, per-layer interventions compound rather than cancel: by the midpoint of the control window, the mean shift already matches the full SRC–TGT gap. Second, the 30% layer budget is not redundant: only 37.7% of examples cross S=1 within the first half of the window, so the remaining layers are needed to bring the bulk of examples past full traversal. Third, the condition-level differences confirm that the adaptive controller produces *representation-level* effects rather than nominal α adjustments alone — target-heavy inputs end the window further along the SRC–TGT axis than source-heavy inputs do, by an amount consistent with their adaptive α.

Model	$\bar{S}(0.5)$	$\bar{S}(1.0)$	%cross by 0.5	%ever cross	median r^*
Aya-Expanse-8B	0.748	1.601	23.1%	65.2%	0.556
Llama3.1-8B	1.034	2.140	40.7%	69.1%	0.444
Mistral-7B	0.770	1.710	26.1%	68.8%	0.556
Phi-3.5-MoE	0.422	1.212	8.8%	53.0%	0.778
Qwen3-30B-A3B	1.001	2.687	40.7%	77.3%	0.462
Qwen3.5-4B	1.080	2.427	46.1%	76.2%	0.444
Qwen3.5-9B	1.067	2.376	44.9%	74.7%	0.444
Qwen3.5-27B	1.886	4.719	71.4%	89.1%	0.333
Mean	1.001	2.359	37.7%	71.7%	0.462

Table 22: **Cumulative recovery per model.** $\bar{S}(r)$ is the mean cumulative projection at relative position r within the control window. "%cross by 0.5" is the fraction of examples whose cumulative S first reaches 1.0 at or before the midpoint. "%ever cross" is the fraction that crosses $S{=}1$ anywhere in the window. r^* is the median relative position of the first crossing among examples that cross.

CS condition	$\bar{S}(0.5)$	$\bar{S}(1.0)$	%ever cross
GF-SRC	0.769	1.964	63.4%
GF-TGT	1.142	2.584	75.9%
RANDOM	0.996	2.350	72.6%
SELECTIVE	1.095	2.536	74.5%

Table 23: **Cumulative recovery by CS condition.** Target-heavy conditions (GF-TGT, SELECTIVE) accumulate a larger cumulative shift and cross $S{=}1$ more often than source-heavy GF-SRC, mirroring the adaptive α schedule.

H.13 CANVAS Statistical Analysis

We assess the statistical reliability of the QA gains in Table 2 using paired per-example F1 scores from the six MLLMs in our primary statistical cohort in Table 24 — those with BASE / $\text{CANVAS}_{\text{adapt}}$ predictions on the full 2880-question evaluation set and diverse MLLMs (Aya-Expanse-8B, Qwen3.5-{4B, 9B, 27B}, Llama-3.1-8B, Mistral-7B). For each model M and CS condition c we form the paired difference $\delta_{M,c}(x) = \text{F1}_{\text{CANVAS}}(M, c, x) - \text{F1}_{\text{BASE}}(M, c, x)$ over the $\sim$720 shared instances per condition.

Tests. For each of the $6 \times 4 = 24$ *(model, condition)* cells we run a one-sided Wilcoxon signed-rank test against $H_0 : \text{median}(\delta_{M,c}) \leq 0$ and control the false discovery rate using the Benjamini–Hochberg procedure at $q = 0.05$. To separate per-model noise from a population-level effect, we treat the per-model means $\bar{\delta}_{M,c}$ as the unit of analysis: for each condition we report the cross-model mean, a model-level bootstrap 95 % CI (10,000 resamples of the six models with replacement), and a one-sided binomial sign test against $H_0 : p = 0.5$ on the number of models with $\bar{\delta}_{M,c} > 0$.

Results. All six MLLMs show a positive overall $\bar{\delta}_M$ (one-sided sign test, $p = 1.6 \times 10^{-2}$), with an overall mean lift of $+0.73$ F1 (model-level 95 % CI $[+0.41, +1.10]$). The same $6/6$ all-positive sign-test outcome ($p = 1.6 \times 10^{-2}$) holds for GF-SRC, GF-TGT, and SELECTIVE; model-level bootstrap CIs in every one of the four CS conditions: SELECTIVE $+0.95$ ($[+0.50, +1.49]$), RANDOM $+0.80$ ($[+0.22, +1.35]$), GF-TGT $+0.61$ ($[+0.32, +0.90]$), and GF-SRC $+0.56$ ($[+0.20, +0.99]$). After BH–FDR correction across the 24 cells, eight are individually significant at $q < 0.05$. The strongest per-cell evidence is concentrated in LLAMA-3.1-8B (three of four cells significant, with $q = 0.004$ on both RANDOM and SELECTIVE), MISTRAL-7B (two of four, including GF-TGT with $\Delta\text{F1} = +0.62$, $q = 0.011$, and SELECTIVE at $q = 0.018$), and AYA-EXPANSE-8B (GF-TGT $q = 0.039$, RANDOM $q = 0.031$). Across all 24 cells the point estimate is positive in 23 — a one-sided binomial test against $p{=}0.5$ gives $p{=}1.5{\times}10^{-6}$, ruling out a chance pattern of cell-level lifts.

Anchor signal validity. The upper-layer anchor bias used by CANVAS separates GF-SRC from GF-TGT at large effect sizes at the per-example level: on `anchor_bias_cos_qonly` the per-model Cohen's d values are $+0.77$ (Qwen3.5-4B), $+0.79$ (Qwen3.5-27B), $+0.81$ (Qwen3.5-9B), and $+1.09$ (Aya-Expanse-8B), all at or above the $|d| \geq 0.7$ conventional threshold for a "large" separation. The Spearman correlation between this anchor signal and $\Delta\text{F1}_{\text{CS vs EN}}$ aggregated to the (model, condition) level is $\rho = +0.524$ with bootstrap 95 % CI $[+0.07, +0.81]$ ($p = 0.037$, $n = 16$ cells). Together, the all-positive sign tests under three of four CS conditions and overall, the $23/24$ positive cell directions, and the large Cohen's d on the anchor signal itself indicate that CANVAS's gains are a robust population-level effect rather

GF-SRC

Model	ΔF1	95% CI	d_z	q
Aya-Expanse-8B	+0.94	[−0.19, +2.13]	0.06	0.071
Qwen3.5-4B	+0.36	[−0.42, +1.15]	0.03	0.288
Qwen3.5-9B	+0.14	[−0.66, +0.93]	0.01	0.367
Qwen3.5-27B	+0.20	[−0.82, +1.20]	0.01	0.367
Llama3.1-8B	**+1.50**	[+0.39, +2.64]	0.10	**0.018**
Mistral-7B	+0.20	[−0.48, +0.85]	0.02	0.170
Mean	+0.56	[−0.36, +1.48]	0.04	—

GF-TGT

Model	ΔF1	95% CI	d_z	q
Aya-Expanse-8B	**+1.17**	[+0.16, +2.18]	0.08	**0.039**
Qwen3.5-4B	+0.02	[−1.15, +1.17]	0.00	0.367
Qwen3.5-9B	+0.49	[−0.12, +1.17]	0.06	0.170
Qwen3.5-27B	+0.95	[−0.16, +2.09]	0.06	0.071
Llama3.1-8B	+0.42	[−0.80, +1.65]	0.03	0.217
Mistral-7B	**+0.62**	[+0.01, +1.23]	0.07	**0.011**
Mean	+0.61	[−0.34, +1.58]	0.05	—

Random

Model	ΔF1	95% CI	d_z	q
Aya-Expanse-8B	**+1.22**	[+0.12, +2.34]	0.08	**0.031**
Qwen3.5-4B	+0.05	[−0.75, +0.85]	0.00	0.470
Qwen3.5-9B	−0.18	[−0.97, +0.63]	−0.02	0.773
Qwen3.5-27B	**+1.28**	[+0.21, +2.38]	0.09	**0.031**
Llama3.1-8B	**+1.85**	[+0.71, +2.98]	0.12	**0.004**
Mistral-7B	+0.56	[−0.05, +1.23]	0.06	0.132
Mean	+0.80	[−0.12, +1.74]	0.06	—

Selective

Model	ΔF1	95% CI	d_z	q
Aya-Expanse-8B	+1.13	[+0.04, +2.29]	0.07	0.052
Qwen3.5-4B	+0.91	[−0.16, +2.01]	0.06	0.140
Qwen3.5-9B	+0.58	[−0.26, +1.45]	0.05	0.106
Qwen3.5-27B	+0.13	[−1.12, +1.40]	0.01	0.449
Llama3.1-8B	**+2.17**	[+0.91, +3.49]	0.12	**0.004**
Mistral-7B	**+0.79**	[+0.23, +1.44]	0.10	**0.018**
Mean	+0.95	[−0.06, +2.01]	0.07	—

Table 24: **Statistical analysis of CANVAS, per condition.** Each panel reports the per-model paired difference $\Delta\text{F1} = \text{CANVAS}_{\text{adapt}} - \text{BASE}$ (F1 × 100) on the full evaluation set. 95 % CIs are from 10,000 paired bootstrap resamples of question examples; d_z is the Hedges-corrected paired effect size; q is the Benjamini–Hochberg-adjusted Wilcoxon signed-rank p-value (one-sided, alternative > 0) across all 6×4=24 cells. Bold entries are significant at $q < 0.05$. The *Mean* row averages the per-model statistics within each condition.

than the consequence of a few favourable cells.

H.14 CANVAS Detector Sensitivity

Motivation. CANVAS uses token-level language tags to decide which question tokens form the source canvas and which target-language token states are interpolated during prefill. The detector does not change the model, the decoding objective, the control layers, or the adaptive rule

$$\alpha = \text{clip}_{[0.05,0.75]}(0.45 - 1.5\gamma_{\text{CS}}).$$

It only changes the SRC/TGT/OTHER token partition. We therefore evaluate how much the final intervention result depends on the language tagger.

Detector variants. We compare four taggers while keeping all other CANVAS components fixed. CURRENT is the tagger used in the main experiments: Unicode script rules for non-Latin target languages and Lingua for Latin-script target languages. UNICODE-ONLY removes Lingua and labels Latin alphabetic tokens as source-language tokens; this is a lower-dependency stress test that should mainly hurt Latin-script target pairs. SOURCE-WORDSET uses the English source question as a lexical reference: Latin tokens that appear in the English source question are labeled SRC, and remaining Latin content tokens are treated as TGT. This variant is analysis-only because it uses the source reference. REFERENCE-ORACLE uses the available source and target reference questions to approximate a perfect token-level partition; it is not a deployable inference-time method, but it gives an upper bound on how much better token tagging could improve CANVAS. The runner also exposes optional `langid` and `langdetect` backends when those packages are installed, but the default comparison avoids adding new dependencies and focuses on the current tagger, a no-detector stress test, and two reference-aided upper-bound settings.

Evaluation protocol. We run each detector on the same full CS examples and generation settings used for the main evaluation, using four models: Qwen3.5-4B, Llama3.1-8B, Mistral-7B, and Aya-Expanse-8B. We report macro-average F1 over GF-SRC, GF-TGT, random switching, and selective switching, along with condition-wise F1, mean α, and the percentage of examples where CANVAS skips intervention because either the source or target span is empty. The key comparison is between CURRENT and REFERENCE-ORACLE. If the oracle row improves substantially, better token-level language identification offers remaining headroom. If the oracle row is close to the current tagger, the main CANVAS gains are not strongly limited by detector quality.

Results. Table 25 shows that detector choice has only a small effect on aggregate CANVAS perfor-

Detector	Avg. F1	Δ vs. current	GF-Src	GF-Tgt	Random	Selective	Mean α	Skip (%)
Current tagger	6.98	+0.00	7.15	6.09	7.16	7.51	0.574	5.1
Unicode only	6.96	-0.01	7.15	6.14	7.09	7.48	0.542	38.4
Source-wordset	6.97	-0.00	7.18	6.00	7.18	7.53	0.521	6.2
Reference oracle	7.00	+0.03	7.27	6.00	7.15	7.58	0.512	9.6

Table 25: **Detector sensitivity for CANVAS.** We keep the CANVAS steering rule fixed and vary only the token-level language tagger. Avg. F1 is macro-averaged over the four CS conditions. Δ is measured against the current Unicode+Lingua tagger. The reference-oracle row is an analysis-only upper bound and is not used as the deployable method.

mance. The current tagger obtains 6.98 average F1, while the reference oracle obtains 7.00 average F1, a gain of only +0.03. This means that a nearly perfect reference-based token partition does not materially change the intervention result in this model pool. The unicode-only stress test remains close in F1 (6.96), but it skips many more examples (38.4%) because it cannot reliably split source and target spans when both languages use Latin script. The source-wordset variant also stays close to the current tagger, which suggests that the current detector already captures most of the useful source/target partition for CANVAS.

Interpretation. We therefore treat CANVAS as weakly dependent on the particular detector used here. Better token-level tagging may still help individual Latin-script language pairs, but the full-run oracle comparison shows limited remaining headroom at the aggregate level. This also supports the main intervention claim: the improvement is not primarily caused by a fragile language-detector choice, but by the representation-space source–target correction applied after token partitioning.

H.15 CANVAS Hyperparameter Sensitivity

We test whether the adaptive controller depends on a narrow hyperparameter optimum. The main paper setting instantiates the bounded schedule $\alpha = \mathrm{clip}_{[\alpha_{\min},\alpha_{\max}]}(\alpha_0 - \lambda\gamma_{\mathrm{CS}})$ with $(\alpha_0, \lambda, \alpha_{\min}, \alpha_{\max}) = (0.45, 1.5, 0.05, 0.75)$. We report the surrounding negative-slope neighbourhood on the same two-model panel (Qwen3.5-4B and Llama3.1-8B, two 100-example seeds per model, macro-averaged over four CS conditions). The sweep varies α_0 (0.40–0.65) and the clipping range while keeping the effective slope $-\lambda = -1.5$, and Table 26 lists variants whose Mean F1 lies at or below the main setting. All listed variants still improve over direct CS answering, with ΔF1 confined to a narrow $[+0.92, +1.04]$ band. The values used in the main paper are therefore a conservative bounded operating point in a flat negative-slope neighbourhood rather than a fragile optimum.

H.16 Task Extensions

RAG setup. We use retrieval to test whether the CANVAS gain is limited to parametric factual recall. We evaluate Llama-3.1-8B on two QA settings: the CodeMixQA short-answer task and PingPong multi-turn dialog QA. The no-retrieval condition ($K = 0$) answers from the original task input, whereas the top-one retrieval condition ($K = 1$) prepends one retrieved passage before answering so that the model can use external evidence in addition to its stored knowledge. For CodeMixQA, retrieval uses the CS question itself as the search query. For PingPong, we evaluate two retrieval queries: a question-only query and a dialog-aware query that concatenates the question with the full multi-turn dialog. The latter tests retrieval when the evidence-bearing entity or relation is only recoverable from the surrounding conversation rather than the final question alone. CodeMixQA uses token F1 against short gold answers, while PingPong QA follows the five-option multiple-choice protocol and reports parsed-letter accuracy.

Topic-classification setup. We also test a non-QA dialog task using the PingPong topic-classification split. Each example contains a code-switched multi-turn conversation and one topic label from five categories. We provide the dialog and the label options to Llama-3.1-8B, parse the selected label from the model output, and report five-class accuracy for direct inference and CANVAS.

Results. Table 27 collects the extension results. On CodeMixQA, retrieval raises the direct-answer baseline substantially, and CANVAS still improves the retrieved condition from 19.71 to 20.60 F1. The gain is larger in the matched no-retrieval condition, where CANVAS improves

Adaptive schedule				Avg. F1 (%)			
center	*slope*	$\alpha_{\min}$	$\alpha_{\max}$	Qwen3.5-4B	Llama3.1-8B	Mean	Δ
0.45	-1.5	**0.05**	**0.75**	**7.98**	**7.68**	**7.83**	**+1.04** ★
0.45	-1.5	0.05	0.80	7.93	7.76	7.82	+1.02
0.45	-1.5	0.10	0.80	7.91	7.74	7.82	+1.02
0.45	-1.5	0.10	0.85	7.89	7.73	7.81	+1.01
0.40	-1.5	0.05	0.75	7.95	7.63	7.79	+0.99
0.40	-1.5	0.05	0.80	7.91	7.65	7.78	+0.98
0.40	-1.5	0.05	0.70	7.89	7.64	7.76	+0.97
0.50	-1.5	0.10	0.75	7.94	7.69	7.82	+1.02
0.55	-1.5	0.10	0.85	7.70	7.92	7.81	+1.01
0.60	-1.5	0.15	0.85	7.81	7.80	7.80	+1.01
0.55	-1.5	0.15	0.85	7.70	7.88	7.79	+0.99
0.65	-1.5	0.15	0.90	7.82	7.73	7.78	+0.98
0.60	-1.5	0.15	0.80	7.77	7.76	7.76	+0.97
0.55	-1.5	0.10	0.80	7.66	7.86	7.76	+0.96
0.55	-1.5	0.15	0.80	7.66	7.82	7.74	+0.95
0.55	-1.5	0.10	0.90	7.75	7.73	7.74	+0.94
0.60	-1.5	0.15	0.90	7.83	7.60	7.72	+0.92
Base	—	—	—	7.41	6.19	6.80	—

Table 26: **CANVAS hyperparameter sensitivity.** Schedule $\alpha = \mathrm{clip}_{[\alpha_{\min},\alpha_{\max}]}(\textit{center} - 1.5\gamma_{\mathrm{CS}})$. The starred row (★) is the main paper setting; we list only variants whose Mean F1 falls at or below it. All variants improve over BASE, with ΔF1 in $[+0.92, +1.04]$.

Dataset	Task	Search Query	Retrieval	Metric	Items	BASE	CANVAS	Δ
CodeMixQA	Short-answer QA	–	$K = 0$	Token F1	2880	5.81	**7.51**	+1.70
CodeMixQA	Retrieval-augmented QA	Question	$K = 1$	Token F1	2880	19.71	**20.60**	+0.89
PingPong	Multi-turn dialog QA	–	$K = 0$	Accuracy	1683	46.76	**48.48**	+1.72
PingPong	Retrieval-augmented dialog QA	Question	$K = 1$	Accuracy	1683	44.86	**46.64**	+1.78
PingPong	Retrieval-augmented dialog QA	Question + dialog	$K = 1$	Accuracy	1683	46.23	**48.48**	+2.26
PingPong	Topic classification	–	–	Accuracy	396	49.75	**51.26**	+1.52

Table 27: **Retrieval and task extensions for CANVAS.** We evaluate Llama-3.1-8B with direct inference (BASE) and the same CANVAS intervention used in the main experiments. $K = 0$ uses no retrieved passage, whereas $K = 1$ prepends one retrieved passage for the QA tasks. For PingPong RAG, we compare retrieval from the question alone with retrieval from the question concatenated with its dialog context.

short-answer QA by +1.70 F1. On PingPong, CANVAS improves dialog QA without retrieval from 46.76 to 48.48 accuracy (+1.72) and remains beneficial in both top-one RAG settings. The question-only retrieval setting improves from 44.86 to 46.64 accuracy (+1.78), while the dialog-aware retrieval setting improves from 46.23 to 48.48 accuracy (+2.26). These RAG gains show that the effect is not limited to recovering source-language parametric knowledge: when retrieved evidence is available, source-canvas alignment still helps the model integrate the CS query with external context. Dialog-aware retrieval gives a stronger retrieved baseline and the larger CANVAS gain, showing that the surrounding dialog can improve retrieval when the final question alone underspecifies the evidence request. The same intervention improves topic classification from 49.75 to 51.26 accuracy, extending the effect beyond QA.

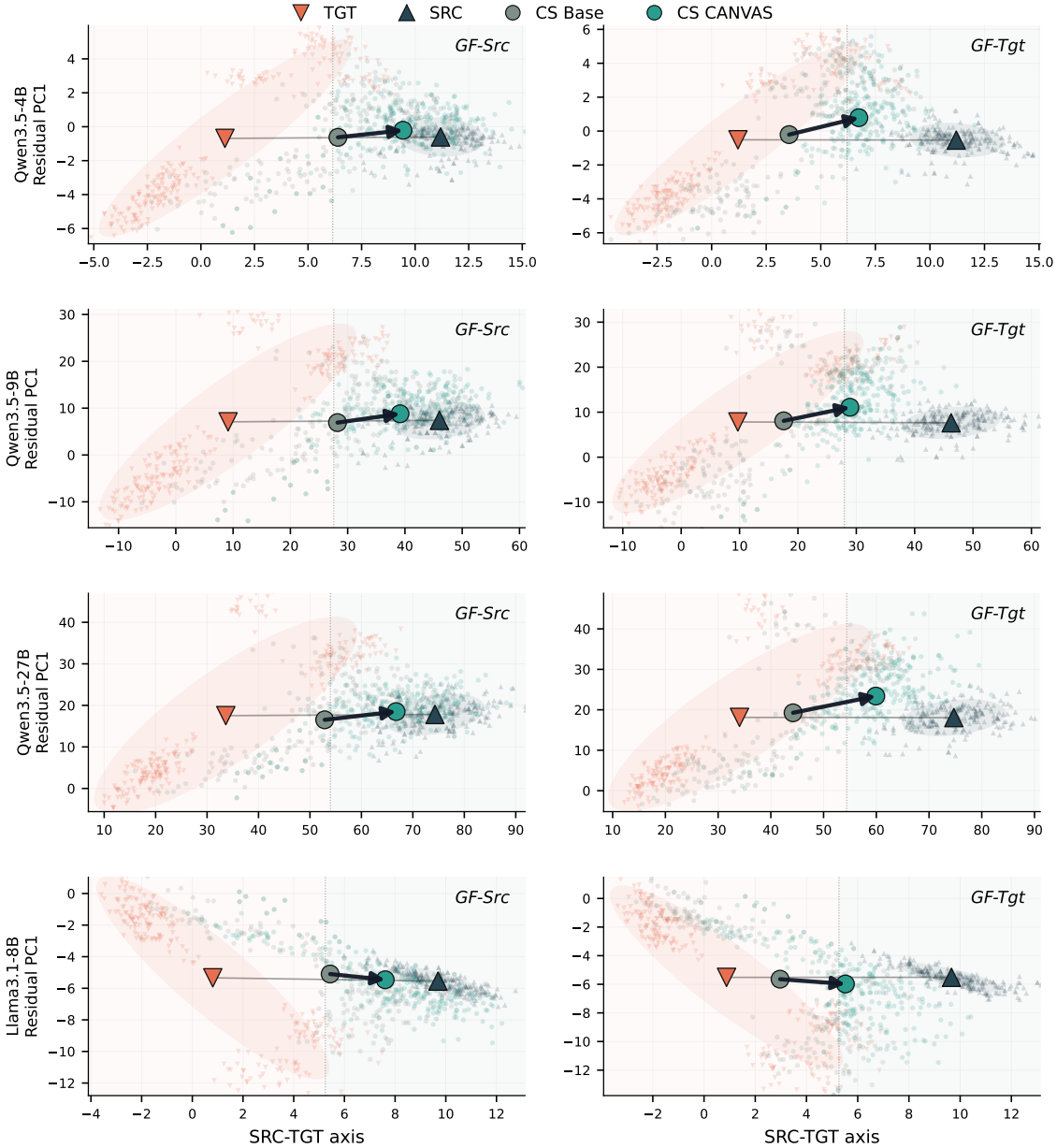

Figure 11: **Model-wise CANVAS representation movement.** We visualize the same representation analysis for four representative models. Each row is a model and each column is a grammar-forced CS condition. Triangles mark SRC/TGT anchors, gray points mark base CS states, teal points mark CANVAS states, and arrows connect the base and CANVAS centroids. Across models, CANVAS shifts the CS centroid toward the source-side region of the SRC–TGT axis.

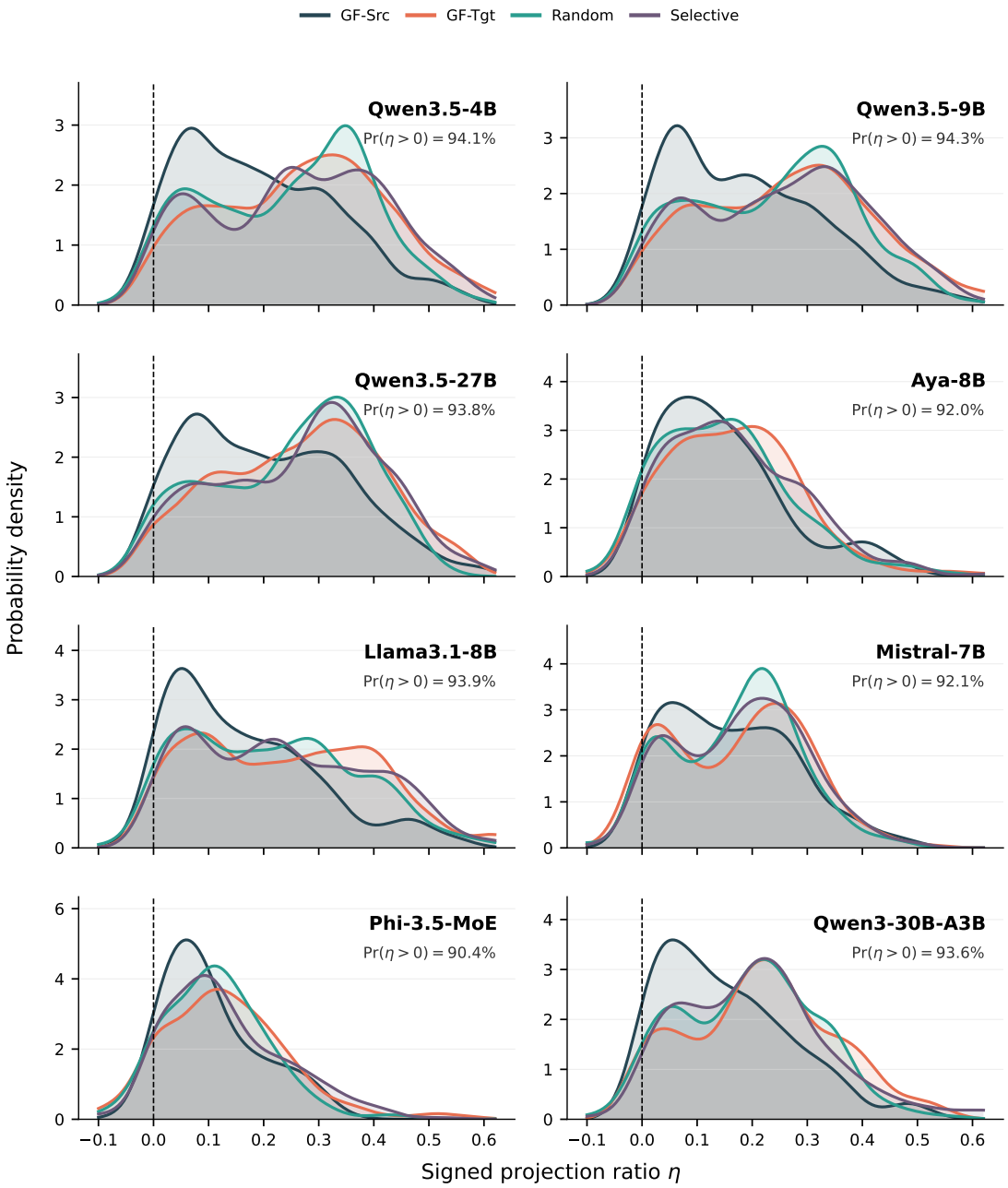

Figure 12: **Model-wise source-ward representation movement.** We plot the signed projection ratio η of the CANVAS displacement onto the SRC–TGT anchor axis for each analyzed model. Positive values indicate movement toward the source anchor. Each panel reports the percentage of examples with $\eta > 0$.

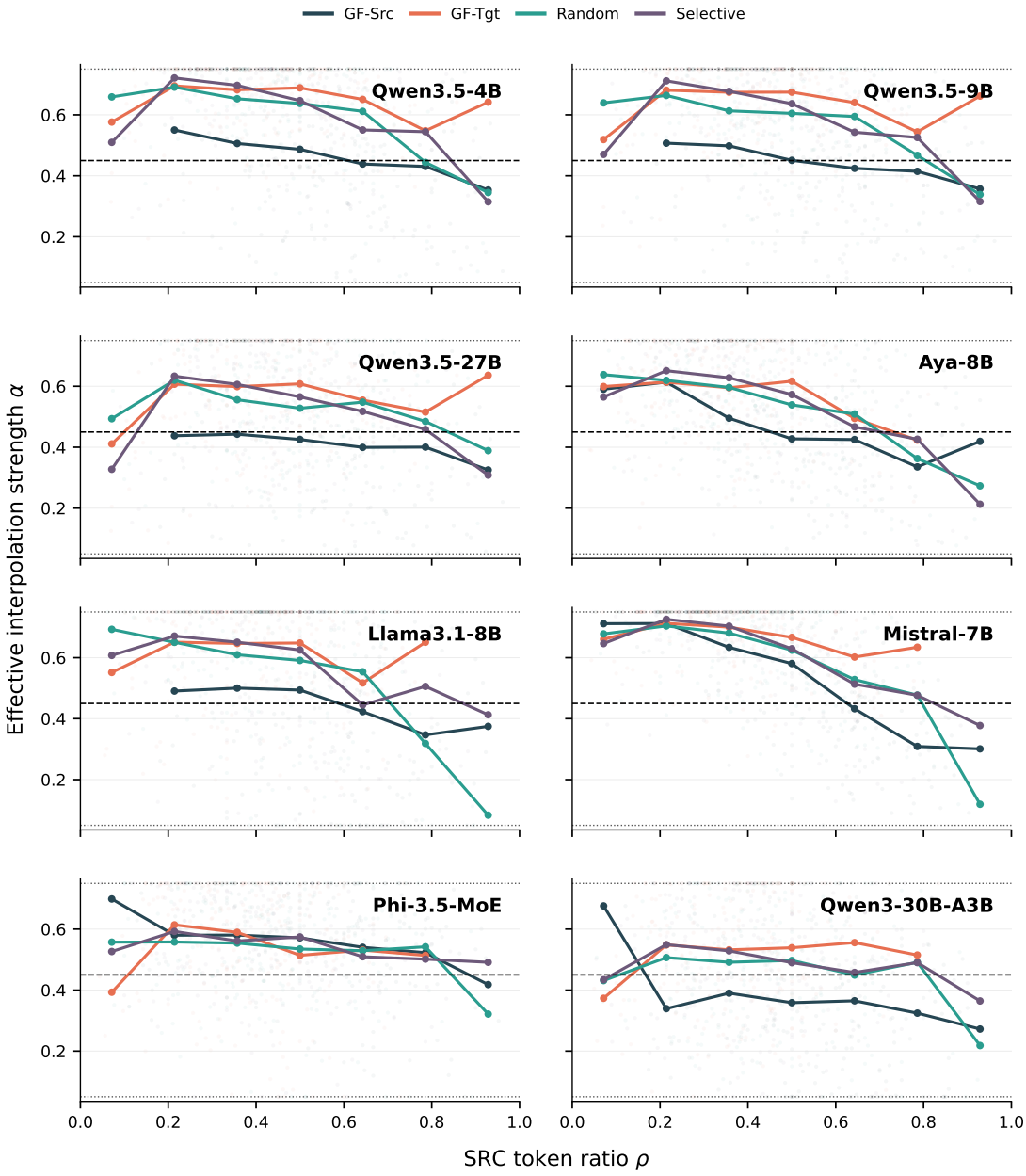

Figure 13: **Model-wise adaptive intervention strength.** We compare the source-token ratio ρ with the effective interpolation strength α selected by CANVAS for each analyzed model. Lower ρ corresponds to more target-heavy inputs, and the binned trend lines show how the adaptive rule changes intervention strength within each model.