Format Before Math: A Behavioral and Mechanistic Study of LLMs' Language-Induced Failures in Arithmetic

Anonymous ACL submission

Abstract

Cross-lingual inconsistency in LLMs is typically attributed to differences in multilingual knowledge or representation. We challenge this assumption by studying LLM behavior on arithmetic, a task where language is irrelevant by design, across 30 models, 9 languages, and 5 prompt styles. We find that base model failures are dominated by *format failures*: certain (language, prompt) combinations trigger a behavioral mode in which the model fails to produce a valid numeric answer (producing instead prompt repetitions, unresolved expression loops, or empty outputs) despite having the correct answer encoded in its representations, accounting for up to 98% of accuracy variance across conditions. Logit lens analysis localizes the format decision to the latter layers of the forward pass, where the first generated token is a near-perfect classifier of output mode (99.7% precision in Gemma). Steering vector interventions causally confirm this localization, recovering over 90% accuracy in format-failure conditions. Our findings suggest that cross-lingual inconsistency in base models is largely a generation-phase phenomenon susceptible to targeted intervention, distinct from the knowledge-level inconsistency studied in prior work.

1 Introduction

Large language models are increasingly deployed in multilingual settings, yet their performance varies significantly depending on the input language (Novikova et al., 2025; Huang et al., 2026), with a positive bias toward high-resource languages. This phenomenon, known as *cross-lingual inconsistency*, damages the reliability and usefulness of LLMs for global applications (Jin et al., 2024; Agarwal et al., 2025), and has been linked to language-based jailbreaks (Shen et al., 2024) and biases in output generation (Zhang et al., 2024; Zhou and Zhang, 2024; Lee et al., 2025).

Prior work studying cross-lingual inconsistency has typically focused on factual knowledge tasks (Ai et al., 2025; Qi et al., 2023; Lim et al., 2025), where language carries semantic content, making it difficult to isolate whether inconsistency reflects differences in world knowledge, task comprehension, or output generation. Interpretability work has characterized how language is organized internally in multilingual models (Wendler et al., 2024; Zhao et al., 2024; Zeng et al., 2025), but has not localized specific behavioral failures to particular network components or confirmed their causal role. These two literatures have rarely been connected.

We study cross-lingual inconsistency through a controlled arithmetic benchmark across 30 models, 9 languages, and 5 prompt styles. Arithmetic provides a language-agnostic test bed: operations are expressed in non-linguistic symbols, the expected output is always a number, and the only varying element is the minimal linguistic prompt. Any behavioral difference across languages must therefore reflect how the model processes the linguistic surface of the prompt alone, isolating format sensitivity from world knowledge and task comprehension.

We find that base model failures are dominated by *format failures*: certain (language, prompt) combinations trigger a behavioral mode in which the model fails to produce a numeric answer (producing instead prompt repetitions, unresolved expression loops, or empty outputs) despite having the correct answer encoded in its representations, accounting for up to 98% of accuracy variance. We term this the **format gate**, localize it to the latter layers of the forward pass through logit lens analysis, and causally confirm it through steering vector

interventions that recover over 90% accuracy in format-failure conditions. Code, data, and results are available at [ANONYMIZED_LINK].

Our contributions are:

- A controlled multilingual arithmetic benchmark spanning 9 languages, 5 prompt styles, and 3 evaluation conditions, designed to isolate format sensitivity from computational competence.
- A behavioral decomposition showing format failures dominate cross-lingual variance in base models, with few-shot prompting and instruction tuning addressing them through distinct mechanisms.
- Mechanistic localization and causal verification of the format gate via logit lens and steering vectors, showing bidirectional control of output mode and confirming pre-encoded correct answers in format-failure cases.

2 Previous Work

2.1 Cross-lingual inconsistency

Cross-lingual inconsistency, understood as the tendency of LLMs to produce different outputs for semantically equivalent inputs across languages, has been identified as a critical reliability concern (Novikova et al., 2025). Prior work has studied it through factual knowledge tasks, proposing metrics over candidate answer probabilities (Qi et al., 2023), semantic similarity (Xing et al., 2024), or logit lens analysis (Lim et al., 2025), consistently finding performance gaps that persist with scale. Lim et al. (2025) show inconsistency arises from language-specific representational subspaces, with larger models more likely to use language-specific subspaces and smaller ones defaulting to a shared English-proximal representation, and that steering toward shared representations reduces inconsistency. Shen et al. (2024) show that alignment does not resolve inconsistency in safety-critical low-resource settings, where models produce both more unsafe and irrelevant responses (a format-like failure). However, all these studies use tasks where language carries semantic content, preventing isolation of whether inconsistency reflects world knowledge, task comprehension, or output generation, and do not decompose inconsistency into its underlying sources. We address both gaps through a language-agnostic task design.

2.2 Multilingual representations

Interpretability work consistently finds that input and output representations are language-specific (first and last layers), while intermediate layers are more language-agnostic (Wendler et al., 2024; Kojima et al., 2024; Zeng et al., 2025). Semantic decisions are predominantly made in an English-proximal space (Wendler et al., 2024; Schut et al., 2025), with Zhao et al. (2024) characterizing this as a multilingual workflow in which models convert inputs to English for intermediate reasoning before generating in the target language. Language-specific neurons are concentrated in early and final layers (Tang et al., 2024; Kojima et al., 2024). Wang et al. (2025a) show through logit lens and cosine similarity that factual errors arise from failures during the language transition in final layers, but restrict their analysis to factual tasks. We extend this to a language-agnostic setting, enabling causal confirmation of format failure localization through activation steering.

2.3 Mathematics across languages

Shi et al. (2022) introduce MGSM, the first multilingual arithmetic reasoning benchmark, finding that performance varies across languages. Wang et al. (2025b) confirm these effects persist in state-of-the-art models, further finding that input-output language consistency is low and thinking length differs significantly across languages. Christ et al. (2025) show that math-specific parameters can be isolated from language-processing ones, suggesting computation and output generation are distinct. Unlike prior work, we use simple arithmetic as a controlled probe for format sensitivity rather than a reasoning challenge.

3 Experimental Setup

Benchmark. We construct a controlled arithmetic benchmark of 600 problems spanning six operation types: summation, subtraction, multiplication, division, two-operator expressions (mix), and linear equations (eq). Problem operands are chosen to be non-trivial (at least two or three digits depending on the

operation type) and the gold answer never appears as an operand in the expression, to avoid trivial pattern matching during evaluation.

Languages and prompts. Each problem is presented with 5 prompt variants in each of 9 languages: Arabic (ar), Catalan (ca), German (de), English (en), Spanish (es), Basque (eu), French (fr), Russian (ru), and Chinese (zh), plus empty (no linguistic prefix) and fake (random Unicode) control conditions. Prompt variants range from minimal one-word instructions (e.g. "Answer:") to longer phrasings (e.g. "Provide the numeric answer to the following operation:"), automatically translated and author-verified (Appendix A.1). We include prompt variation to increase behavioral range for mechanistic analysis.

Evaluation conditions. We evaluate under three conditions: (1) zero-shot with base pretrained models, (2) five-shot with base pretrained models, and (3) zero-shot with instruction-tuned variants. For five-shot, we generate five fixed additional problems per type under the same constraints as the evaluation set, formatted with the same language and prompt style as the target problem. All generations use greedy decoding for determinism and reproducibility.

Output taxonomy. Model outputs are classified into three categories using a regex-based extractor: ANSWER-CORRECT (a numeric answer matching the gold label is produced), ANSWER-WRONG (a numeric answer is produced but is incorrect), and NO-ANSWER (no numeric answer is extracted). NO-ANSWER cases are further subdivided into prompt repetition loops, unresolved operation loops, and no digits generated. Classification adopts a permissive upper-bound criterion: any output containing the gold answer in a non-operator position is considered correct, making format failure estimates conservative (see Appendix A.2 for details and Appendix A.3 for examples).

Models. We evaluate 30 models spanning multiple families and scales: Gemma-3 (Team et al., 2025) (4B, 12B, base and instruct), Llama3.2 (1B, 3B) and Llama-3.1 (Grattafiori et al., 2024) (8B, base and instruct), Qwen3 (Yang et al., 2025) (0.6B, 1.7B, 4B, 8B, 14B, base and instruct), and three European multilingual models: EuroLLM (Martins et al., 2024) (1.7B, 9B, base and instruct), Salamandra (Gonzalez-Agirre et al., 2025) (2B, 7B, base and instruct), and Teuken7B (Ali et al., 2025) (base and instruct). Full model identifiers and HuggingFace links are provided in Appendix A.4. We focus our analysis on three representative models: Gemma-3-4b-pt, Meta-Llama-3.2-3B, and Qwen3-1.7B-Base, selected as archetypes of format-sensitive, intermediate, and format-robust behavioral profiles at comparable scales.[1]

4 Behavioral-Level Analysis

We analyze model behavior under zero-shot base, five-shot base, and zero-shot instruction-tuned conditions, using accuracy variance across (language, prompt) as our measure of cross-lingual inconsistency (Xing et al., 2024). We additionally evaluate constrained decoding as a targeted intervention that forces numeric output, testing whether correct answers are encoded in the model's representations independently of format. Full results for all 30 models are in Appendix B.1.

4.1 Zero-shot base models: format failures dominate

Under zero-shot, base models exhibit substantial accuracy variance across (language, prompt) conditions, with distribution shapes varying dramatically across model families (Figure 1). Gemma-3-4B shows a clearly bimodal distribution reflecting two distinct behavioral regimes; Llama-3.2-3B shows a wide unimodal distribution capable of near-zero accuracy; Qwen3-1.7B-Base shows the tightest distribution with minimal failures.

Figure 2 reveals that this variance is driven by format failures: in Gemma, 37% of zero-shot outputs are NO-ANSWER, dropping to 0% with five-shot while accuracy recovers from 47% to 82%. Llama shows a similar pattern: NO-ANSWER drops from 18% to 0%, yet ANSWER-WRONG remains at 23%. Qwen has almost no NO-ANSWER (5%) even in zero-shot. NO-ANSWER rate and accuracy

[1] Qwen3-1.7B-Base is selected despite its smaller size as it already surpasses the other families in accuracy and robustness at this scale range.

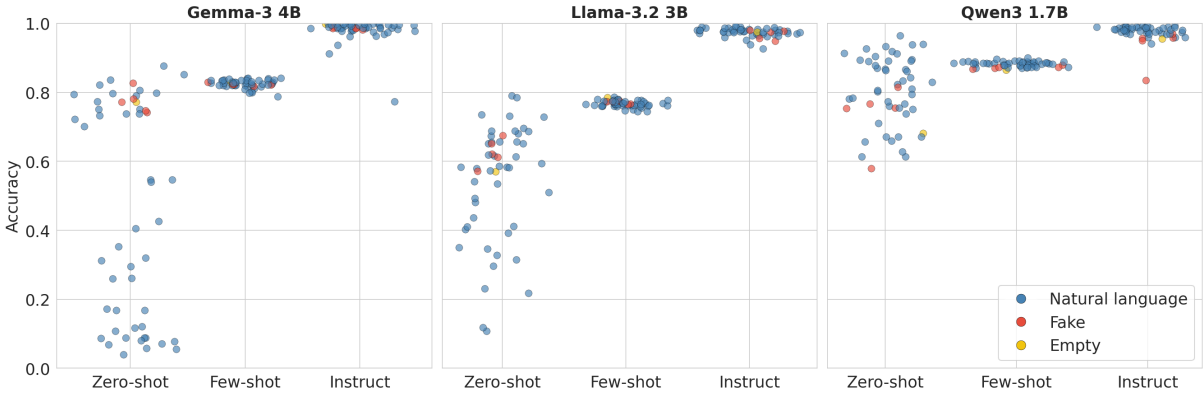


Figure 1: Accuracy distribution across (language, prompt) conditions under zero-shot base, few-shot base, and zero-shot instruct settings. Each point represents one (language, prompt) combination. Natural language conditions are shown in blue, the *fake* prompt in orange, and the *empty* prompt in yellow.

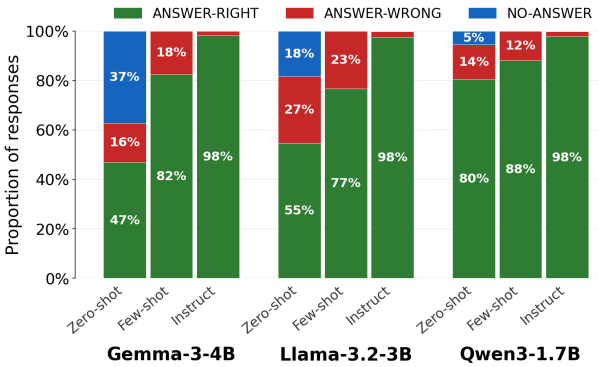


Figure 2: Distribution of ANSWER-CORRECT, ANSWER-WRONG, and NO-ANSWER outputs across evaluation conditions.

are strongly negatively correlated in format-sensitive models (mean r=−0.97 for Gemma and Llama) but weakly in Qwen (r=−0.67), pointing to format failure as the primary driver of cross-lingual variance (Appendix B.4).

Several additional patterns emerge. Scaling improves mean accuracy but only modestly reduces variance (Appendix B.1). Fake and empty prompts show consistently low NO-ANSWER rates in format-sensitive models, often outperforming real language prompts, suggesting recognizable linguistic content triggers format failure (a pattern absent in Qwen). Finally, equations show near-zero NO-ANSWER rates universally while sum and sub are most vulnerable (Appendix B.3).

4.2 Few-shot: isolating computation errors

Five-shot prompting eliminates format failures almost entirely across all model families. The wide distributions observed in zero-shot collapse to narrow bands concentrated near each model's ceiling accuracy (Figure 1, center column), with NO-ANSWER rates dropping to near zero regardless of language or prompt style (Figure 2). This variance collapse indicates that format failures, not computation errors, were responsible for the cross-lingual inconsistency observed in zero-shot.

The residual errors that remain after format is stabilized are small and largely language-invariant, with standard deviations below 0.05 across all models (Appendix B.1). Notably, the proportion of ANSWER-WRONG responses remains similar between zero-shot and five-shot: five-shot anchors the model in answer mode but does not reduce computation errors, which were already present in zero-shot ANSWER cases. These errors are not uniformly distributed across operation types: sum, sub, mul, and div reach near-ceiling accuracy, while mix and eq remain substantially below ceiling (Appendix B.2), confirming that residual errors concentrate in multi-step operations requiring sequential reasoning incompatible with the fixed direct-answer format.

4.3 Instruction tuning: reasoning unlocks computation

Instruction-tuned variants of Gemma, Llama, and Qwen reach near-ceiling accuracy (97-98%) with minimal variance across (language, prompt) conditions (Figure 2, right column) and residual format failures concentrated mainly in Basque and fake with the longest prompt variant (*long_numeric*) (Ap-

	Gemma3-4B		Llama3.2-3B		Qwen3-1.7B	
	$\Delta\mu$	$\Delta\sigma$	$\Delta\mu$	$\Delta\sigma$	$\Delta\mu$	$\Delta\sigma$
empty	-0.11	0.00	-0.02	0.00	-0.19	0.00
ar	**0.29**	-0.28	-0.01	-0.08	-0.19	0.01
ca	**0.20**	-0.34	**0.03**	-0.09	-0.25	0.10
de	**0.28**	-0.31	**0.21**	-0.14	-0.22	0.04
en	**0.35**	-0.25	-0.07	-0.01	-0.39	0.11
es	**0.44**	-0.13	**0.08**	-0.14	-0.30	0.06
eu	-0.02	-0.16	**0.06**	-0.08	-0.06	-0.06
fr	**0.32**	-0.25	**0.14**	-0.10	-0.41	0.06
ru	**0.14**	-0.32	**0.28**	-0.12	-0.32	0.08
zh	-0.10	0.00	-0.09	-0.01	-0.19	0.08
fake	-0.15	0.04	-0.03	-0.01	-0.14	0.06
Avg.	**0.17**	-0.23	**0.06**	-0.12	-0.24	0.05

Table 1: $\Delta\mu$ and $\Delta\sigma$ show the change in mean accuracy and cross-prompt standard deviation under constrained vs. open-ended decoding; negative $\Delta\sigma$ indicates more consistent behavior across prompts.

pendix B.1). Crucially, this improvement goes beyond format compliance: unlike five-shot prompting, which forces direct answers and leaves multi-step operations (mix, eq) unsolved, instruction-tuned models employ step-by-step reasoning that resolves these operations. The gap between five-shot and instruction-tuned accuracy is concentrated precisely in mix and eq (the operation types that require intermediate steps) indicating that reasoning capability, not format compliance alone, accounts for the performance recovery (Appendix B.2; see Appendix B.5 for output examples).

This pattern does not hold for European multilingual models. The smaller instruct variants of Salamandra and EuroLLM reach at most 0.27, with few-shot doing slightly better (up to 0.36), while the largest models in these families and Teuken reach at most 0.86. Unlike Gemma, Llama, and Qwen, accuracy variance shows mixed patterns across error types (Appendix B.4), suggesting residual format failures alongside capacity-related errors.

4.4 Constrained decoding: computation is pre-encoded

If format failure prevents the model from producing an answer rather than reflecting an inability to compute one, then blocking non-numeric generation should recover correct answers directly from the model's representations. We test this through constrained decoding, restricting generation to digit tokens, whitespace, minus sign, and EOS. Table 1 shows the change in accuracy (Δ acc) and cross-prompt standard deviation ($\Delta\ \sigma$) relative to open-ended generation. In Gemma, constrained decoding substantially improves accuracy in high-NO-ANSWER languages (e.g. es: Δacc = +0.44, fr: +0.32) while simultaneously reducing prompt sensitivity ($\Delta\ \sigma < 0$ across all real languages), indicating that correct answers are already encoded in the model's representations and that format failure occurs in the generation phase rather than the comprehension phase. The effect is absent or negative for Qwen (Δacc = -0.24 globally), consistent with its format robustness, since Qwen base models produce correct answers through intermediate generation steps that are blocked by constrained decoding. Constrained decoding is not universally effective: some conditions show high variance or degenerate outputs (e.g. repetitive digit sequences), particularly when the prompt does not naturally lead to a direct numeric answer. Full results and examples are in Appendix B.6.

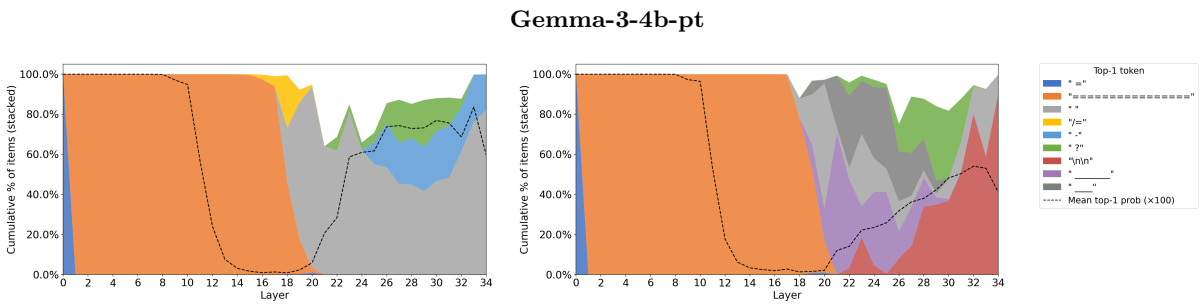
Notably, constrained decoding recovers accuracy on single-step operations (sum, sub, mul, div) but fails on mix and eq (Appendix B.2), mirroring the five-shot pattern and further confirming that multi-step failures reflect a reasoning limitation rather than a format problem.

5 Mechanistic-Level Results

All prompts in our benchmark end with the arithmetic expression followed by `=`, making this the final input token before generation begins, and thus the natural position to investigate format-related processing. Despite this token being identical across all languages, the hidden state at the `=` position retains language-specific information in final layers (Appendix C.2), suggesting that linguistic context in the prompt accumulates in the residual stream and influences what follows. We now investigate where and how this linguistic trace triggers the format gate.

5.1 Localizing the format gate

We find that the first token generated after `=` is a near-perfect predictor of the output mode in Gemma and Llama, but not in Qwen (Table C.1). In Gemma, `\n\n` predicts NO-ANSWER with 99.7% precision and 93.9% recall, while numeric prefix tokens (`' '`, `' -'`) predict ANSWER with over 95% precision and



(a) ANSWER (12225) (b) NO-ANSWER (5175)

Figure 3: Distribution of top-1 token predictions across layers by answer category. Areas represent the percentage of instances predicting each token as top-1 at each layer (filtered to tokens reaching ≥20% in at least one layer). The dashed line shows mean top-1 softmax probability.

99.6% recall. Llama shows a similar but weaker pattern, with question-mark variants as NO-ANSWER tokens (precision 76–99%, combined recall 83.5%) and numeric prefix tokens predicting ANSWER with 96.1% precision and 94.3% recall. In Qwen, no token reliably signals NO-ANSWER, consistent with its lack of a sharp behavioral gate.

This raises the question of where in the network this routing decision is formed. We apply logit lens (nostalgebraist, 2020), which projects the residual stream at each layer through the final unembedding matrix to obtain a vocabulary distribution, revealing what token the model would predict at each layer. By applying this at the `=` position, we trace how the model's implicit prediction evolves across layers and identify where the format decision crystallizes. Figure 3 shows the distribution of top-1 predictions per layer, split by ANSWER and NO-ANSWER cases for Gemma; results for Llama and Qwen are in Appendix C.3.

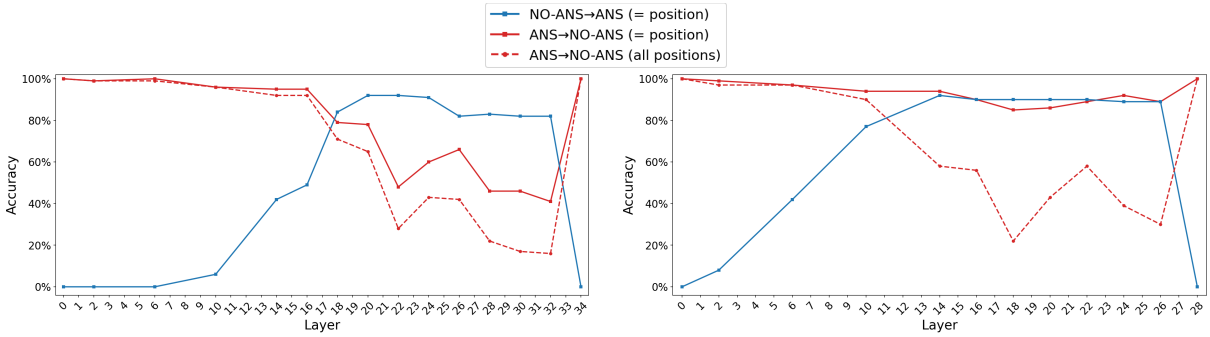
In Gemma, both output modes produce identical top-1 predictions through early layers: all examples predict `================`, with no differentiation between ANSWER and NO-ANSWER conditions. Top-1 probability drops sharply around layers 8–14, indicating a transition. Around layers 18-20, `' '` emerges as the dominant prediction in ANSWER cases and remains so thereafter, partially converting to `' -'` around layer 24 for negative answers. In NO-ANSWER cases, `\n\n` emerges around layer 22 and grows to dominate final layers. The two modes begin to diverge from layer 18, with NO-ANSWER showing lower final-layer probability (∼0.40 vs ∼0.60), indicating a weaker, later commitment.

In Llama and Qwen (Appendix C.3), `' ='` dominates only the first few layers before giving way to diverse predictions. In Llama, divergence sharpens around layer 16. Then, unlike Gemma, the main NO-ANSWER token `?\n` emerges as the dominant prediction in both output modes around layer 22, but only ANSWER cases gradually replace it with numeric tokens in final layers, suggesting a late resolution mechanism. Qwen shows a similar but more diffuse trajectory, with the two modes converging at the final layer, consistent with the absence of a reliable NO-ANSWER token. Qwen's intermediate predictions tend toward English, consistent with prior work on English as an internal pivot (Wendler et al., 2024).

These findings reveal a two-stage process: format-relevant tokens emerge in the second half of the network (layers 18–22 in Gemma, 16–22 in Llama), then stabilize as the winning token commits. In Gemma, the lower final-layer probability of NO-ANSWER predictions confirms that format failure reflects a weak and late commitment relative to answer mode. The contrast between models (abrupt divergence in Gemma, late resolution in Llama, absent in Qwen) mirrors the gradient of format sensitivity identified in §4.

5.2 Causal intervention

We compute steering vectors using contrastive activation addition (Rimsky et al., 2024) to test whether the format gate identified in §5.1 can be causally controlled. For each model, the steering vector is computed as the mean difference of residual stream represen-



(a) Gemma-3-4b-pt (b) Meta-Llama-3.2-3B

Figure 4: Accuracy across layers under causal intervention via steering vectors for (a) Gemma-3-4b-pt (b) and Meta-Llama-3.2-3B. **Blue solid** (NO-ANS →ANS): positive steering at = position, baseline accuracy 0%. **Red solid** (ANS →NO-ANS): reverse direction at = position, baseline accuracy 100%. **Red dashed**: reverse direction applied at all generation steps.

	Gemma			Llama		
Experiment	**Steering vec**	**Eval on**	**Acc (layer)**	**Steering vec**	**Eval on**	**Acc (layer)**
Base case	`(ca,ans)-(es,ans)`	`(es,ans)`	92% (20)	`(en,ans)-(ru,ans)`	`(ru,ans)`	92% (14)
Within language	`(es,det)-(es,ans)`	`(es,ans)`	93% (24)	`(ru,long_num)-(ru,ans)`	`(ru,ans)`	99% (18)
Reverse direction	`(ca,ans)-(es,ans)`	`(ca,ans)`	41% (32) *16% (32)	`(en,ans)-(ru,ans)`	`(en,ans)`	85% (18) *22% (18)
Prompt generalization	`(ca,ans)-(es,ans)`	`(es,comp)`	83% (20)	`(en,ans)-(ru,ans)`	`(ru,sol)`	81% (14)
Language generalization	`(ca,ans)-(es,ans)`	`(ar,ans)`	85% (16)	`(en,ans)-(ru,ans)`	`(de,ans)`	77% (26)
FS vector	`(fs,es,ans)-(es,ans)`	`(es,ans)`	86% (18)	`(fs,ru,ans)-(ru,ans)`	`(ru,ans)`	93% (10)
IT vector	`(it,es,ans)-(es,ans)`	`(es,ans)`	× 5% (24)	`(it,ru,ans)-(ru,ans)`	`(ru,ans)`	88% (18)
	`(it,ar,ans)-(ar,ans)`	`(ar,ans)`	80% (24)	`(it,de,ans)-(de,ans)`	`(de,ans)`	85% (24)

Table 2: Steering vector generalization results for Gemma-3-4b-pt and Meta-Llama-3.2-3B. Numbers in parentheses indicate the layer at which peak accuracy was achieved. All experiments use coefficient +1 and intervention at the = position only, except the reverse direction experiments which use coefficient -1. * indicates the reverse direction result with intervention at all generation steps. ×indicates the vector failed to rescue answer mode.

tations at the = position, taken at the output of each transformer block, between 100 ANSWER-CORRECT instances [ANS] and 100 NO-ANSWER instances [NO-ANS], computed independently at each layer. At inference time, the vector is added to the residual stream at the output of the target transformer block, applied only at the = token position, and its effects are evaluated on a separate held-out set of 100 instances. We apply this to Gemma and Llama; Qwen is excluded due to its near-zero NO-ANSWER rate.

Figure 4 shows accuracy per layer for three conditions: positive steering (NO-ANS →ANS, = only), negative steering (ANS →NO-ANS, = only), and negative steering applied at every generated token. The vector is computed from (ca, ans) vs. (es, ans) for Gemma and (en, ans) vs. (ru, ans) for Llama, as these represent the most behaviorally contrasting conditions under the *answer* prompt. Positive steering recovers over 90% accuracy in both models, with onset at layer 20 in Gemma and layer 14 in Llama. The steeper rise in Gemma is consistent with its abrupt logit lens transition; Llama's more gradual rise mirrors its smoother format commitment. The reverse direction induces NO-ANSWER, though with a weaker effect when applied only at = and stronger when applied at all generation steps, and Llama shows two peaks (layers 18 and 26). This asymmetry between positive and negative directions is consistent with the logit lens finding that answer-mode tokens commit earlier and more confidently than NO-ANSWER tokens in Gemma, while in Llama both modes share an intermediate token before resolving differently. Results for a second language pair and all-positions steering are in Appendix C.4.

Table 2 generalizes these results. The steering vector transfers across prompt styles and languages, confirming it captures a model-

internal format representation. Vectors computed from same-language conditions differing only in prompt style ("within-language" in the table) achieve the strongest results (93% in Gemma, 99% in Llama), ruling out the interpretation that the vector captures language identity. Few-shot vectors succeed (86–93%), and IT vectors work in most conditions (ar: 80% in Gemma, ru: 88% in Llama), though the es IT vector fails for Gemma (5%), a result that warrants further investigation.

6 Discussion

Reframing cross-lingual inconsistency. Our results suggest that, even when language plays no key semantic role in the task, inconsistency across linguistic surface variations arises and decomposes into qualitatively distinct failure types. The dominant failure in base models is format-based: the format decision crystallizes in the final layers of the forward pass, and NO-ANSWER rate alone explains the vast majority of accuracy variance (r,=,−0.991 in Gemma, r,=,−0.949 in Llama). Notably, fake and empty prompts outperform real language prompts in format-sensitive models, and equations bypass the format gate universally, suggesting the gate is activated by learned associations between specific linguistic patterns and non-answer continuation behaviors in pretraining data. Format failures respond to few-shot prompting; reasoning failures require step-by-step instruction tuning; capacity-related failures persist regardless, as European multilingual models illustrate: IT reduces format failures but cannot supply the reasoning capabilities that well-resourced families develop. This mirrors Shen et al. (2024)'s finding that alignment training yields minimal improvement for low-resource languages, suggesting some failures are rooted in pretraining and are not yet addressed through standard post-training. How format failures interact with factual inconsistencies in tasks where language carries semantic content remains an open question, as disentangling the two failure modes in naturalistic settings would clarify their relative contribution to cross-lingual performance gaps.

Format robustness as a training-recipe property. Format sensitivity varies substantially across families in ways not explained by scale: Qwen base models are robust even at 0.6B while Gemma shows dramatic sensitivity at 4B. The gradient of gate sharpness (abrupt in Gemma, gradual in Llama, absent in Qwen) suggests format robustness is shaped by pretraining data composition or training objectives rather than being an inevitable consequence of multilingual representation, though the specific pretraining factors remains unclear.

Implications for intervention. Steering vector results suggest format failures can be addressed through lightweight inference-time interventions as an alternative to full instruction tuning. The effectiveness of few-shot vectors (86–93% accuracy recovery) is particularly promising, and a small number of examples may suffice to extract a usable format gate direction. Whether this generalizes beyond arithmetic remains open for future work. However, inference-time steering may affect model behavior on other tasks, and its broader impact remains to be evaluated before deployment.

7 Conclusion

Through an arithmetic task with a minimal linguistic prefix, where language is irrelevant by design, we reveal a previously overlooked source of cross-lingual inconsistency in base LLMs: format failure (rather than computational inability), a failure mode distinct from the knowledge-level inconsistency studied in prior work on factual tasks. Certain (language, prompt) combinations trigger a behavioral mode in which the model fails to produce a numeric answer despite having the correct answer encoded in its representations. This format gate crystallizes in the latter layers of the forward pass, manifests as a near-deterministic first-token decision, and can be causally controlled through targeted activation steering. Few-shot prompting and instruction tuning address format failures through distinct mechanisms, while constrained decoding confirms that computation is intact even when format fails. Format sensitivity varies substantially across model families and is not resolved by scaling alone, suggesting it is a training-recipe property amenable to targeted intervention.

Limitations

Our mechanistic analysis focuses on three model families and one task domain; whether the format gate generalizes to other tasks and architectures remains open. We decompose failures into format and computation errors but do not fully characterize the latter; wrong answers vary in nature (rounding errors, sign errors, etc.) and their origins are unexplored.

Although our arithmetic setup is designed as the minimal case where language affects behavior, language and prompt variation cannot be fully disentangled: prompts are always instantiated in a specific language, and observed variance reflects both cross-lingual and paraphrase sensitivity jointly. Similarly, the boundary between format robustness and other behavioral properties (e.g. instruction following, output style) is not always sharp.

Our regex-based extractor may miss valid answers in unusual formats, and the permissive classification criterion may count some ambiguous cases as correct, leading to slightly optimistic accuracy estimates. A more precise extractor could tighten these estimates.

Logit lens projects intermediate representations through the final unembedding matrix, but there is no guarantee that intermediate layers produce meaningful vocabulary-space predictions: the method reveals tendencies rather than definitive layer-level computations (nostalgebraist, 2020).

Finally, we can only speculate on why specific (language, prompt) combinations trigger the format gate: the unknown composition of pretraining data makes it difficult to explain why particular linguistic associations were learned.

References

Amit Agarwal, Hansa Meghwani, Hitesh Laxmichand Patel, Tao Sheng, Sujith Ravi, and Dan Roth. 2025. Aligning LLMs for multilingual consistency in enterprise applications. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 117–137, Suzhou (China). Association for Computational Linguistics.

Xi Ai, Mahardika Krisna Ihsani, and Min-Yen Kan. 2025. Are knowledge and reference in multilingual language models cross-lingually consistent? In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 4975–5011, Suzhou, China. Association for Computational Linguistics.

Mehdi Ali, Michael Fromm, Klaudia Thellmann, Jan Ebert, Alexander Arno Weber, Richard Rutmann, Charvi Jain, Max Lübbering, Daniel Steinigen, Johannes Leveling, Katrin Klug, Jasper Schulze Buschhoff, Lena Jurkschat, Hammam Abdelwahab, Benny Jörg Stein, Karl-Heinz Sylla, Pavel Denisov, Nicolo' Brandizzi, Qasid Saleem, and 22 others. 2025. Teuken-7b-base & teuken-7b-instruct: Towards european llms. *Preprint*, arXiv:2410.03730.

Bryan R Christ, Zachary Gottesman, Jonathan Kropko, and Thomas Hartvigsen. 2025. Math neurosurgery: Isolating language models' math reasoning abilities using only forward passes. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24803–24840, Vienna, Austria. Association for Computational Linguistics.

Aitor Gonzalez-Agirre, Marc Pàmies, Joan Llop, Irene Baucells, Severino Da Dalt, Daniel Tamayo, José Javier Saiz, Ferran Espuña, Jaume Prats, Javier Aula-Blasco, Mario Mina, Iñigo Pikabea, Adrián Rubio, Alexander Shvets, Anna Sallés, Iñaki Lacunza, Jorge Palomar, Júlia Falcão, Lucía Tormo, and 5 others. 2025. Salamandra technical report. *Preprint*, arXiv:2502.08489.

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

Kaiyu Huang, Fengran Mo, Xinyu Zhang, Hongliang Li, You Li, Yuanchi Zhang, Weijian Yi, Yulong Mao, Jinchen Liu, Yuzhuang Xu, Jinan Xu, Jian-Yun Nie, and Yang Liu. 2026. A survey on large language models with multilingualism: recent advances and new frontiers. *Artificial Intelligence Review*.

Yiqiao Jin, Mohit Chandra, Gaurav Verma, Yibo Hu, Munmun De Choudhury, and Srijan Kumar. 2024. Better to ask in english: Cross-lingual evaluation of large language models for healthcare queries. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2627–2638, New York, NY, USA. Association for Computing Machinery.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pretrained language models: Finding and controlling language-specific neurons. In *Proceedings of the*

2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Nahyun Lee, Yeongseo Woo, Hyunwoo Ko, and Guijin Son. 2025. Controlling language confusion in multilingual LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1026–1035, Vienna, Austria. Association for Computational Linguistics.

Zheng Wei Lim, Alham Fikri Aji, and Trevor Cohn. 2025. Language-specific latent process hinders cross-lingual performance. *Preprint*, arXiv:2505.13141.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *Preprint*, arXiv:2409.16235.

nostalgebraist. 2020. interpreting GPT: the logit lens. `https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens`. LessWrong blog post.

Jekaterina Novikova, Carol Anderson, Borhane Blili-Hamelin, Domenic Rosati, and Subhabrata Majumdar. 2025. Consistency in language models: Current landscape, challenges, and future directions. *Preprint*, arXiv:2505.00268.

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.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

Lisa Schut, Yarin Gal, and Sebastian Farquhar. 2025. Do multilingual llms think in english? *Preprint*, arXiv:2502.15603.

Lingfeng Shen, Weiting Tan, Sihao Chen, Yunmo Chen, Jingyu Zhang, Haoran Xu, Boyuan Zheng, Philipp Koehn, and Daniel Khashabi. 2024. The language barrier: Dissecting safety challenges of LLMs in multilingual contexts. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2668–2680, Bangkok, Thailand. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2022. Language models are multilingual chain-of-thought reasoners. *Preprint*, arXiv:2210.03057.

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

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Mingyang Wang, Heike Adel, Lukas Lange, Yihong Liu, Ercong Nie, Jannik Strötgen, and Hinrich Schuetze. 2025a. Lost in multilinguality: Dissecting cross-lingual factual inconsistency in transformer language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5075–5094, Vienna, Austria. Association for Computational Linguistics.

Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, Qiqian Cang, Yichang Zhang, Fei Huang, Junyang Lin, Fei Huang, and Jingren Zhou. 2025b. Polymath: Evaluating mathematical reasoning in multilingual contexts. *Preprint*, arXiv:2504.18428.

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.

Xiaolin Xing, Zhiwei He, Haoyu Xu, Xing Wang, Rui Wang, and Yu Hong. 2024. Evaluating knowledge-based cross-lingual inconsistency in large language models. *Preprint*, arXiv:2407.01358.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Hongchuan Zeng, Senyu Han, Lu Chen, and Kai Yu. 2025. Converging to a lingua franca: Evolution of linguistic regions and semantics alignment in multilingual large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10602–10617, Abu Dhabi, UAE. Association for Computational Linguistics.

Liang Zhang, Qin Jin, Haoyang Huang, Dongdong Zhang, and Furu Wei. 2024. Respond in my language: Mitigating language inconsistency in response generation based on large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4177–4192, Bangkok, Thailand. Association for Computational Linguistics.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA. Curran Associates Inc.

Di Zhou and Yinxian Zhang. 2024. Political biases and inconsistencies in bilingual gpt models—the cases of the u.s. and china. *Scientific Reports*, 14.

A Benchmark and Evaluation Details

A.1 Prompts per Language

Table 3 shows the five prompt variants used across the languages evaluated in this work. Prompts were automatically translated and verified by the authors. The *answer*, *compute*, *solve*, and *determine* variants are short imperative instructions of one to two tokens in most languages, while *long_numeric* provides a full descriptive sentence. The *fake* condition replaces the linguistic prompt with random Unicode characters drawn from diverse scripts, serving as a control that preserves the structural format of the prompt (prefix followed by the arithmetic expression) while removing all interpretable linguistic content. The *empty* condition (not shown) omits the prefix entirely, presenting only the arithmetic expression.

A.2 Output Classification and Extractor Details

Model outputs are classified into four categories, described in Table 4. The primary distinction is between outputs that contain a numeric response (Answer-Correct, Answer-Incorrect) and those that do not (No-Answer, Unknown). The Unknown category captures edge cases where neither a numeric answer nor a recognized failure pattern is detected; these represent less than 1% of outputs across all conditions and are excluded from accuracy calculations.

No-Answer extractors. When no numeric answer is detected, we apply a set of pattern-based extractors to identify the failure mode, described in Table 5. The most frequent patterns are *prompt_repetition_drift* (the model echoes the prompt and expression in a loop) and *unresolved_operation_loop* (the model generates a sequence of arithmetic expressions without resolving any of them). These two patterns account for the vast majority of No-Answer cases in format-sensitive models.

Answer extractors. When a numeric response is detected, we apply a prioritized sequence of extractors to identify the candidate answer, described in Table 6. Extractors are applied in order of specificity — more constrained patterns (e.g. *mix_first_line_final_eq*) take priority over more general ones (e.g. *first_line_equal*). The extracted number is then compared to the gold answer for exact match.

A.3 Output Examples

Tables 8, and 9 show representative output examples for Gemma-3-4b-pt, Meta-Llama-3.2-3B, and Qwen3-1.7B-Base respectively, under zero-shot base conditions. Each table illustrates the four output classes defined in Table 4: ANSWER-CORRECT, ANSWER-WRONG, NO-ANSWER (prompt repetition), and NO-ANSWER (unresolved operation loop), across different languages and prompt styles.

Output classification is not always straightforward. Consider the following output for "Ответь: 27 + 14 * 5 =" (ru, answer prompt):

```
107.\n\n1. 14 * 5 = 70.\n2.
27 + 70 = 97.\nОтветь:
```

	answer	compute	determine	solve	long_numeric
ar	أجب:	احسب:	حدد:	حل:	أعطِ الناتج العددي للعملية التالية:
ca	Respon:	Calcula:	Determina:	Resol:	Proporciona la resposta numèrica de la següent operació:
de	Antworte:	Berechne:	Bestimme:	Löse:	Gib die numerische Antwort für die folgende Operation an:
en	Answer:	Compute:	Determine:	Solve:	Proporciona la respuesta numérica de la siguiente operación:
es	Responde:	Calcula:	Determina:	Resuelve:	Proporciona la respuesta numérica...
eu	Erantzun:	Kalkulatu:	Zehaztu:	Ebatzi:	Eman hurrengo eragiketaren emaitza zenbakizkoa:
fr	Réponds :	Calcule :	Détermine :	Résous :	Donne la réponse numérique à l'opération suivante :
ru	Ответь:	Вычисли:	Определи:	Реши:	Дай числовой ответ для следующей операции:
zh	回答:	计算:	求:	解:	给出下列运算的数值结果:
fake	xfxt\\:	pp夠:	CNgM:	BЯ:	JqFW๐C:

Table 3: Prompt variants used across languages. Each prompt instructs the model to solve the following arithmetic expression, varying in length and phrasing from minimal one-word instructions to full descriptive sentences. The *fake* condition uses random Unicode characters with no linguistic meaning, serving as a control for the absence of interpretable linguistic signal.

Class	Condition
ANSWER-CORRECT	The output contains the gold answer as a response (not as an operand within an unanswered arithmetic expression).
ANSWER-INCORRECT	A numeric response is detected but does not match the gold answer.
NO-ANSWER	No numeric response is detected and no recognized failure pattern is identified. These cases are excluded from accuracy calculations (<1% of outputs).
UNKNOWN	There is no response number detected nor any no-answer pattern detected.

Table 4: Output classification taxonomy. Each model generation is assigned to one of four classes based on the presence and correctness of a numeric response.

```
97.\n\nОтветь:     12 * 12 - 12
= 108...
```

The model first outputs an incorrect answer (107), then self-corrects through step-by-step reasoning to arrive at the correct answer (97), before entering an unresolved operation loop. This case is classified as ANSWER-CORRECT since the gold answer appears in a non-operator position. Our extractor adopts a permissive upper-bound criterion, counting any output containing the gold answer outside of operator position as correct regardless of what precedes it. This conservative choice means our accuracy estimates a lower bound.

A different failure mode appears in the following output for "Ebatzi: 258 + 308 =" (eu, solve prompt):

```
566\nEbatzi: 258 + 308 =
566\nPost by Ebatzi » Mon Mar
12, 2018 9:00 am\nI have a
question about the 566th day
of the year. I have been told
that it is the day that the
world will end...
```

The model produces the correct answer (566) immediately, then drifts into English forum-style text apparently triggered by associating the number with pretraining data. This case is correctly classified as ANSWER-CORRECT since the gold answer appears first in a non-operator position. This example also illustrates the language drift phenomenon, where the model begins in Basque but continues in English, and the influence of pretraining data associations on generation behavior even after a correct answer has been produced.

A.4 Models

Table 10 lists all 30 models evaluated in this work, including HuggingFace model identifiers, parameter counts, and model type (base or instruct). Models are grouped by family. All models were evaluated using greedy decoding with the default tokenizer and chat template where applicable.

Extractor	Triggered when
empty	Model output is literally empty
no_digit_gate	Output contains no digits at all, or all digits come from prompt/expression echoes
prompt_repetition_drift	Output is drifting by repeating prompt numbers or echoing expressions without resolving them
unresolved_operation_loop	Output keeps repeating unanswered a op b = operations and never resolves
other_repetition_drift	Text repetition drift (not number-echo, not prompt-echo)

Table 5: No-Answer extractor patterns. When no numeric response is detected, outputs are further classified by failure mode using these pattern-based extractors. Extractors highlighted in red identify cases where digits are present but originate from the prompt or expression rather than a genuine model response.

Extractor	What it matches
mix_first_line_final_eq	mix-type problems: first-line final = <number>
prompt_expression_eq_number	Exact prompt expression followed by = <number>
expression_eq_number_answer_variant	Same but specific to answer prompt variant
operation_followed_by_number	Original operation followed by a number (with optional unit/symbol)
single_number_output	Entire output contains exactly one number not in the prompt
leading_standalone_number	First line is a standalone number (non-variable problems)
leading_prefix_number	First line has a prefix + number (e.g. = 42)
leading_number_unit_expl	First-line number with a unit or explanation
probable_text_answer	CJK equality-style sentence ending in a number
leading_number_expl	First-line number with non-math explanation text
mcq	Multiple-choice question pattern
xslot_leading_and_assignment	x-slot problems: leading standalone number AND explicit x = <n>both found
xslot_assignment_only	x-slot problems: explicit variable assignment only
leading_standalone_number_xslot	x-slot problems: leading standalone number only (no assignment)
boxed / bold	\boxed{…} or **…** formatted answer
answer_label	Explicit answer label like "Answer: 1728" or single-line colon/equals number
sentence_prompt_numbers_answer	First/last sentence that uses prompt numbers and ends in an answer
last_line_eq_number_eos_guard	Last line = <number>only when generation ended with EOS
last_standalone_number_line	Last standalone number line (e.g. after </think>)
first_line_equal	First-line operation ending with = <number>(low-priority fallback)

Table 6: Answer extractor patterns applied when a numeric response is detected. Extractors are applied in decreasing order of specificity; the first matching extractor determines the candidate answer, which is then compared to the gold label by exact match.

Class	Lang	Prompt + expression	Output (truncated)
NO-ANSWER (UNRESOLVED OP LOOP)	es	`Responde: 216 + 305 =`	`\n\nResponde: 125 + 125 =\n\nResponde: 125 + 125 + 125 =\n\n(…)`
NO-ANSWER (PROMPT REP.)	fr	`Résous : 174 - 28 =`	`\n\nRésous : 174 - 28 =\n\nRésous : 174 - 28 =\n\n(…)`
ANSWER-CORRECT	ru	Ответь: 14 * 26 =	364\n14 * 26 = 364\n(...)
ANSWER-WRONG	zh	回答： 10 + 27 * 7 =	201\n\n 回答： 10 + 27 * 7 = 201\n\n(⋯)

Table 7: Example outputs for each output class from **Gemma-3-4B-pt**.

Class	Lang	Prompt + expression	Output (truncated)
NO-ANSWER (UNRESOLVED OP LOOP)	de	`Berechne: 186 + 427 =`	`?\nBerechne: 1 + 1 = ?\nBerechne: 1 + 1 = ?\n(…)`
NO-ANSWER (PROMPT REP.)	ca	`Resol: 10 + 27 * 7 =`	`10 + 27 * 7 = 10 + 27 * 7 = 10 + 27 * 7 = 10 + 27 * 7 = (…)`
ANSWER-CORRECT	fake	`CNgM: 174 - 28 =`	`146\nCNgM: 174 - 28 = 146\n(…)`
ANSWER-WRONG	eu	`Ebatzi: -x - 7 = -57\nx =`	`-7\nEbatzi: -x - 7 = -57\nx = -7\n(…)`

Table 8: Example outputs for each output class from **Llama-3.2-3B**.

Class	Lang	Prompt + expression	Output (truncated)
NO-ANSWER (UNRESOLVED OP LOOP)	fr	`Détermine : 12 * 22 =`	`24 * 12 = 12 * 24 = 22 * 12 = 22 * 22 = 12 * 12 = 22 * 24 = (…)`
NO-ANSWER (PROMPT REP.)	es	`Calcula: 219 + 299 =`	`219 + 299 = 219 + 299 = 219 + 299 = 219 + 299 = (…)`
ANSWER-CORRECT	ar	حل: 14 * 26 =	364\n الحل: 14 * 26 = 364\n الحل: 14 * 26 = 364\n (…)
ANSWER-WRONG	en	`Answer: 22 * 19 - 15 =`	`397\nAnswer: 22 * 19 - 15 = 397\n(…)`

Table 9: Example outputs for each output class from **Qwen3-1.7B-Base**.

Gemma-3 (Team et al., 2025) is Google's open model series with strong multilingual coverage. Llama-3 (Grattafiori et al., 2024) is Meta's open model series covering Llama-3.1 (8B) and Llama-3.2 (1B, 3B) variants. Qwen3 (Yang et al., 2025) is Alibaba's model series with strong multilingual and mathematical pretraining, particularly for Chinese. EuroLLM (Martins et al., 2024), Salamandra (Gonzalez-Agirre et al., 2025), and Teuken-7B (Ali et al., 2025) are European multilingual models; Salamandra additionally emphasizes Catalan and other Iberian languages, and Teuken-7B focuses on German.

Qwen3 instruct models support a reasoning mode that generates explicit chain-of-thought tokens (`<think>`); we evaluate with reasoning disabled (`enable_thinking=False`) for comparability with Gemma and Llama instruct variants, which do not have an equivalent reasoning scaffold. We verified that this yields comparable accuracy to the default reasoning-enabled mode (global accuracy 0.97 in both cases for Qwen3-1.7B).

Family	HuggingFace identifier	Params	Type
Gemma-3	google/gemma-3-4b-pt	4B	base
	google/gemma-3-4b-it	4B	instruct
	google/gemma-3-12b-pt	12B	base
	google/gemma-3-12b-it	12B	instruct
Llama-3	meta-llama/Llama-3.2-1B	1B	base
	meta-llama/Llama-3.2-1B-Instruct	1B	instruct
	meta-llama/Llama-3.2-3B	3B	base
	meta-llama/Llama-3.2-3B-Instruct	3B	instruct
	meta-llama/Llama-3.1-8B	8B	base
	meta-llama/Llama-3.1-8B-Instruct	8B	instruct
Qwen3	Qwen/Qwen3-0.6B-Base	0.6B	base
	Qwen/Qwen3-0.6B	0.6B	instruct
	Qwen/Qwen3-1.7B-Base	1.7B	base
	Qwen/Qwen3-1.7B	1.7B	instruct
	Qwen/Qwen3-4B-Base	4B	base
	Qwen/Qwen3-4B	4B	instruct
	Qwen/Qwen3-8B-Base	8B	base
	Qwen/Qwen3-8B	8B	instruct
	Qwen/Qwen3-14B-Base	14B	base
	Qwen/Qwen3-14B	14B	instruct
EuroLLM	utter-project/EuroLLM-1.7B	1.7B	base
	utter-project/EuroLLM-1.7B-Instruct	1.7B	instruct
	utter-project/EuroLLM-9B	9B	base
	utter-project/EuroLLM-9B-Instruct	9B	instruct
Salamandra	BSC-LT/salamandra-2b	2B	base
	BSC-LT/salamandra-2b-instruct	2B	instruct
	BSC-LT/salamandra-7b	7B	base
	BSC-LT/salamandra-7b-instruct	7B	instruct
Teuken	openGPT-X/Teuken-7B-base-v0.6	7B	base
	openGPT-X/Teuken-7B-instruct-v0.6	7B	instruct

Table 10: All models evaluated in this work. HuggingFace identifiers are provided for reproducibility.

B Behavioral Results

B.1 Detailed Accuracy Results

Table 11 reports mean accuracy and standard deviation across languages and prompts for all 30 models under zero-shot base, few-shot base, and zero-shot instruction-tuned conditions. Table 12 provides a summary of key metrics per model including global accuracy, cross-language and cross-prompt standard deviation, empty and fake prompt accuracy, and NO-ANSWER rate. Table 13 lists the top-3 and bottom-3 (language, prompt) setups per model.

Although instruct models show near-zero NO-ANSWER rates overall, a small number of residual failures persist in specific edge cases. These are concentrated in Basque with the *long_numeric* prompt across all instruct model families, and in fake prompts with *long_numeric* for some Llama and Qwen instruct variants, suggesting that low-resource languages with complex prompt styles remain challenging even after instruction tuning (under 5% of conditions affected). Table 14 shows the (lang, prompt) with the highest no-answer rate across the instructed Gemma, Llama, and Qwen models evaluated.

B.2 Accuracy per Problem Type

Table 15 reports mean accuracy per problem type (sum, sub, mul, div, mix, eq) for each representative model under all four evaluation conditions. This supports the claims in §4.2 and §4.3 regarding the concentration of residual errors in multi-step operations (mix, eq) under few-shot and constrained decoding conditions.

B.3 No-answer rates per language and problem type

Figure 5 shows NO-ANSWER rates per language and problem type, aggregated across all prompt styles, for the three representative models under zero-shot base conditions. This supports the claim in §4.1 that equations universally bypass the format gate while simple

Model	Cond.	empty	ar	ca	de	en	es	eu	fr	ru	zh	fake	GLOBAL
Llama-3.2-1B	ZS	0.38±0.00	0.58±0.05	0.45±0.07	0.49±0.05	0.38±0.24	0.43±0.11	0.49±0.06	0.47±0.06	0.39±0.09	0.54±0.10	0.48±0.05	0.47±0.11
	FS	0.69±0.00	0.68±0.01	0.67±0.01	0.68±0.01	0.68±0.01	0.68±0.02	0.67±0.01	0.67±0.02	0.67±0.01	0.67±0.01	0.67±0.00	0.67±0.01
	IT	0.92±0.00	0.90±0.04	0.88±0.05	0.92±0.03	0.92±0.01	0.92±0.03	0.90±0.04	0.92±0.05	0.93±0.03	0.93±0.02	0.93±0.01	0.92±0.03
Meta-Llama-3.2-3B	ZS	0.57±0.00	0.59±0.12	0.59±0.12	0.39±0.18	0.65±0.13	0.53±0.18	0.56±0.09	0.47±0.16	0.31±0.16	0.71±0.05	0.63±0.04	0.54±0.17
	FS	0.79±0.00	0.77±0.01	0.77±0.01	0.76±0.01	0.77±0.01	0.77±0.01	0.76±0.01	0.77±0.01	0.77±0.01	0.77±0.01	0.77±0.00	0.77±0.01
	IT	0.97±0.00	0.97±0.01	0.97±0.01	0.98±0.01	0.98±0.01	0.98±0.01	0.96±0.02	0.99±0.00	0.97±0.00	0.98±0.01	0.97±0.01	0.97±0.01
Meta-Llama-3.1-8B	ZS	0.82±0.00	0.70±0.14	0.77±0.06	0.52±0.16	0.68±0.14	0.77±0.05	0.57±0.25	0.68±0.20	0.46±0.27	0.80±0.04	0.75±0.07	0.67±0.18
	FS	0.85±0.00	0.84±0.01	0.84±0.01	0.84±0.01	0.85±0.01	0.84±0.01	0.83±0.01	0.85±0.01	0.84±0.01	0.84±0.02	0.84±0.01	0.84±0.01
	IT	0.98±0.00	0.96±0.02	0.98±0.01	0.99±0.00	0.98±0.01	0.99±0.00	0.95±0.01	0.98±0.01	0.99±0.00	0.97±0.01	0.96±0.03	0.98±0.02
Qwen3-0.6B	ZS	0.47±0.00	0.74±0.09	0.78±0.07	0.75±0.13	0.82±0.10	0.77±0.04	0.68±0.02	0.70±0.17	0.74±0.14	0.71±0.11	0.67±0.01	0.73±0.11
	FS	0.80±0.00	0.80±0.00	0.80±0.01	0.80±0.00	0.79±0.01	0.80±0.00	0.80±0.01	0.80±0.00	0.80±0.01	0.81±0.01	0.80±0.01	0.80±0.01
	IT	0.95±0.00	0.96±0.01	0.95±0.01	0.95±0.03	0.97±0.01	0.95±0.01	0.94±0.02	0.97±0.01	0.97±0.01	0.97±0.01	0.95±0.02	0.96±0.02
Qwen3-1.7B	ZS	0.68±0.00	0.85±0.06	0.75±0.10	0.82±0.04	0.91±0.03	0.79±0.13	0.72±0.07	0.87±0.10	0.82±0.11	0.78±0.08	0.73±0.09	0.80±0.10
	FS	0.86±0.00	0.89±0.01	0.88±0.01	0.89±0.00	0.89±0.00	0.89±0.01	0.88±0.01	0.88±0.01	0.89±0.01	0.89±0.01	0.87±0.00	0.88±0.01
	IT	0.95±0.00	0.98±0.01	0.98±0.01	0.98±0.01	0.98±0.01	0.98±0.01	0.97±0.02	0.98±0.01	0.98±0.01	0.98±0.00	0.93±0.06	0.97±0.02
Qwen3-4B	ZS	0.83±0.00	0.93±0.03	0.95±0.03	0.94±0.01	0.94±0.03	0.97±0.02	0.89±0.04	0.95±0.02	0.94±0.01	0.93±0.02	0.89±0.04	0.93±0.04
	FS	0.95±0.00	0.97±0.00	0.96±0.00	0.97±0.00	0.97±0.01	0.97±0.00	0.96±0.00	0.96±0.00	0.97±0.00	0.96±0.00	0.97±0.01	0.97±0.00
	IT	0.99±0.00	1.00±0.00	0.99±0.00	1.00±0.00	1.00±0.00	0.99±0.00	0.98±0.01	1.00±0.00	1.00±0.00	1.00±0.00	0.99±0.01	0.99±0.01
Qwen3-8B	ZS	0.88±0.00	0.95±0.04	0.94±0.02	0.96±0.02	0.95±0.03	0.94±0.03	0.93±0.03	0.95±0.01	0.95±0.03	0.96±0.02	0.89±0.08	0.94±0.04
	FS	0.96±0.00	0.97±0.00	0.97±0.01	0.98±0.00	0.97±0.01	0.98±0.00	0.97±0.00	0.98±0.00	0.97±0.00	0.97±0.00	0.97±0.00	0.97±0.00
	IT	0.99±0.00	1.00±0.00	1.00±0.00	1.00±0.00	0.99±0.00	1.00±0.00	1.00±0.00	1.00±0.00	0.99±0.02	1.00±0.00	1.00±0.00	1.00±0.01
Qwen3-14B	ZS	0.88±0.00	0.91±0.08	0.91±0.05	0.94±0.06	0.97±0.02	0.93±0.06	0.92±0.01	0.97±0.01	0.94±0.05	0.95±0.02	0.91±0.03	0.93±0.05
	FS	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.98±0.00	0.97±0.01	0.97±0.00	0.98±0.00
	IT	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.01	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00
gemma-3-4b-pt	ZS	0.77±0.00	0.35±0.31	0.47±0.37	0.42±0.34	0.31±0.31	0.19±0.20	0.65±0.20	0.32±0.30	0.52±0.36	0.64±0.22	0.77±0.03	0.47±0.31
	FS	0.83±0.00	0.83±0.01	0.83±0.01	0.82±0.02	0.81±0.02	0.82±0.01	0.83±0.00	0.82±0.01	0.83±0.01	0.82±0.01	0.82±0.00	0.82±0.01
	IT	1.00±0.00	0.98±0.00	0.99±0.01	0.99±0.01	0.98±0.04	0.99±0.01	0.97±0.02	0.99±0.01	0.95±0.10	0.99±0.02	0.99±0.00	0.98±0.03
gemma-3-12b-pt	ZS	0.88±0.00	0.46±0.34	0.45±0.42	0.36±0.36	0.37±0.34	0.26±0.22	0.48±0.37	0.25±0.17	0.40±0.29	0.85±0.07	0.87±0.04	0.48±0.34
	FS	0.94±0.00	0.94±0.01	0.93±0.01	0.93±0.01	0.93±0.01	0.94±0.01	0.93±0.01	0.93±0.01	0.93±0.01	0.93±0.00	0.93±0.01	0.93±0.01
	IT	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.00	1.00±0.01	1.00±0.00	1.00±0.00	1.00±0.00	0.99±0.01	1.00±0.00	1.00±0.01	1.00±0.00
Teuken-7B	ZS	0.52±0.00	0.53±0.02	0.54±0.05	0.46±0.15	0.56±0.08	0.57±0.04	0.54±0.03	0.48±0.17	0.48±0.09	0.59±0.03	0.54±0.02	0.53±0.09
	FS	0.69±0.00	0.69±0.01	0.67±0.01	0.67±0.01	0.68±0.01	0.67±0.01	0.68±0.01	0.67±0.01	0.67±0.01	0.68±0.01	0.68±0.01	0.68±0.01
	IT	0.84±0.00	0.63±0.09	0.81±0.06	0.78±0.07	0.86±0.03	0.80±0.04	0.79±0.07	0.80±0.03	0.79±0.05	0.68±0.12	0.81±0.06	0.78±0.09
salamandra-2b	ZS	0.13±0.00	0.20±0.02	0.20±0.07	0.20±0.08	0.11±0.11	0.13±0.09	0.16±0.07	0.20±0.10	0.19±0.08	0.19±0.06	0.18±0.02	0.17±0.08
	FS	0.32±0.00	0.37±0.01	0.37±0.02	0.37±0.02	0.34±0.01	0.37±0.01	0.37±0.01	0.38±0.01	0.37±0.02	0.34±0.01	0.36±0.02	0.36±0.02
	IT	0.13±0.00	0.10±0.02	0.20±0.06	0.16±0.04	0.18±0.02	0.20±0.02	0.12±0.04	0.17±0.03	0.14±0.05	0.06±0.03	0.12±0.05	0.15±0.06
salamandra-7b	ZS	0.24±0.00	0.24±0.12	0.24±0.12	0.24±0.06	0.20±0.11	0.20±0.11	0.25±0.02	0.22±0.10	0.24±0.10	0.25±0.05	0.31±0.05	0.24±0.09
	FS	0.61±0.00	0.63±0.01	0.63±0.02	0.61±0.02	0.58±0.05	0.63±0.01	0.63±0.01	0.62±0.01	0.62±0.01	0.63±0.01	0.62±0.01	0.62±0.02
	IT	0.34±0.00	0.32±0.05	0.40±0.06	0.37±0.01	0.45±0.07	0.39±0.03	0.36±0.03	0.39±0.03	0.34±0.02	0.34±0.05	0.30±0.02	0.37±0.06
EuroLLM-1.7B	ZS	0.16±0.00	0.18±0.02	0.20±0.02	0.16±0.08	0.23±0.02	0.20±0.04	0.19±0.03	0.15±0.09	0.20±0.04	0.21±0.04	0.19±0.02	0.19±0.05
	FS	0.32±0.00	0.36±0.02	0.36±0.01	0.36±0.02	0.36±0.02	0.36±0.01	0.36±0.02	0.36±0.02	0.35±0.01	0.36±0.02	0.34±0.01	0.35±0.02
	IT	0.10±0.00	0.23±0.07	0.21±0.11	0.25±0.08	0.35±0.04	0.26±0.08	0.31±0.03	0.30±0.03	0.29±0.04	0.32±0.02	0.17±0.06	0.27±0.08
EuroLLM-9B	ZS	0.71±0.00	0.61±0.16	0.56±0.13	0.60±0.11	0.69±0.08	0.55±0.13	0.62±0.05	0.56±0.13	0.52±0.19	0.64±0.05	0.67±0.04	0.61±0.12
	FS	0.71±0.00	0.71±0.01	0.71±0.01	0.72±0.02	0.71±0.02	0.71±0.01	0.71±0.01	0.70±0.01	0.71±0.01	0.71±0.02	0.72±0.01	0.71±0.01
	IT	0.83±0.00	0.88±0.03	0.86±0.02	0.89±0.04	0.88±0.04	0.88±0.04	0.78±0.11	0.88±0.04	0.89±0.05	0.85±0.03	0.86±0.06	0.86±0.06

Table 11: Mean accuracy ± standard deviation per language and condition (ZS = zero-shot base, FS = five-shot base, IT = instruction-tuned) for all models.

model	Mean ACC	SD Lang	SD Prompt	Acc Empty	Mean Acc Fake	SD Fake	No-Answer rate
Llama-3.2-1B	0.468	0.066	0.052	0.383	0.477	0.049	0.134
Meta-Llama-3.2-3B	0.534	0.127	0.050	0.570	0.626	0.040	0.182
Meta-Llama-3.1-8B	0.662	0.118	0.075	0.818	0.754	0.068	0.160
Llama-3.2-1B-Instruct	0.914	0.017	0.014	0.915	0.928	0.015	0.002
Llama-3.2-3B-Instruct	0.975	0.009	0.005	0.975	0.968	0.014	0.002
Meta-Llama-3.1-8B-Instruct	0.979	0.012	0.004	0.982	0.959	0.029	0.002
Qwen3-0.6B-Base	0.743	0.042	0.051	0.473	0.668	0.013	0.052
Qwen3-1.7B-Base	0.813	0.061	0.052	0.682	0.734	0.090	0.055
Qwen3-4B-Base	0.937	0.020	0.008	0.830	0.892	0.037	0.005
Qwen3-8B-Base	0.948	0.010	0.008	0.883	0.892	0.082	0.011
Qwen3-14B-Base	0.937	0.023	0.014	0.880	0.913	0.026	0.016
Qwen3-0.6B	0.959	0.012	0.007	0.952	0.946	0.019	0.001
Qwen3-1.7B	0.980	0.005	0.002	0.955	0.934	0.056	0.002
Qwen3-4B	0.995	0.004	0.003	0.992	0.990	0.006	0.000
Qwen3-8B	0.997	0.003	0.002	0.995	0.996	0.005	0.000
Qwen3-14B	0.999	0.001	0.001	0.998	1.000	0.000	0.000
gemma-3-4b-pt	0.427	0.158	0.239	0.772	0.774	0.034	0.370
gemma-3-12b-pt	0.427	0.179	0.249	0.883	0.874	0.044	0.439
gemma-3-4b-it	0.980	0.015	0.019	0.998	0.986	0.003	0.000
gemma-3-12b-it	0.998	0.002	0.003	1.000	0.997	0.007	0.000
Teuken-7B-base-v0.6	0.528	0.045	0.062	0.517	0.536	0.024	0.029
Teuken-7B-instruct-v0.6	0.772	0.071	0.045	0.837	0.809	0.056	0.021
salamandra-2b	0.174	0.035	0.050	0.128	0.178	0.019	0.054
salamandra-7b	0.230	0.022	0.073	0.242	0.313	0.052	0.104
salamandra-2b-instruct	0.149	0.047	0.027	0.128	0.118	0.049	0.044
salamandra-7b-instruct	0.374	0.039	0.029	0.345	0.302	0.019	0.024
EuroLLM-1.7B	0.191	0.025	0.022	0.162	0.191	0.016	0.066
EuroLLM-9B	0.596	0.052	0.075	0.715	0.666	0.044	0.085
EuroLLM-1.7B-Instruct	0.281	0.045	0.020	0.100	0.167	0.060	0.032
EuroLLM-9B-Instruct	0.866	0.034	0.038	0.835	0.860	0.061	0.004

Table 12: Summary statistics per model: mean accuracy, standard deviation across languages (SD Lang) and across prompts (SD Prompt), accuracy on empty and fake prompts, and overall NO-ANSWER rate under zero-shot base conditions.

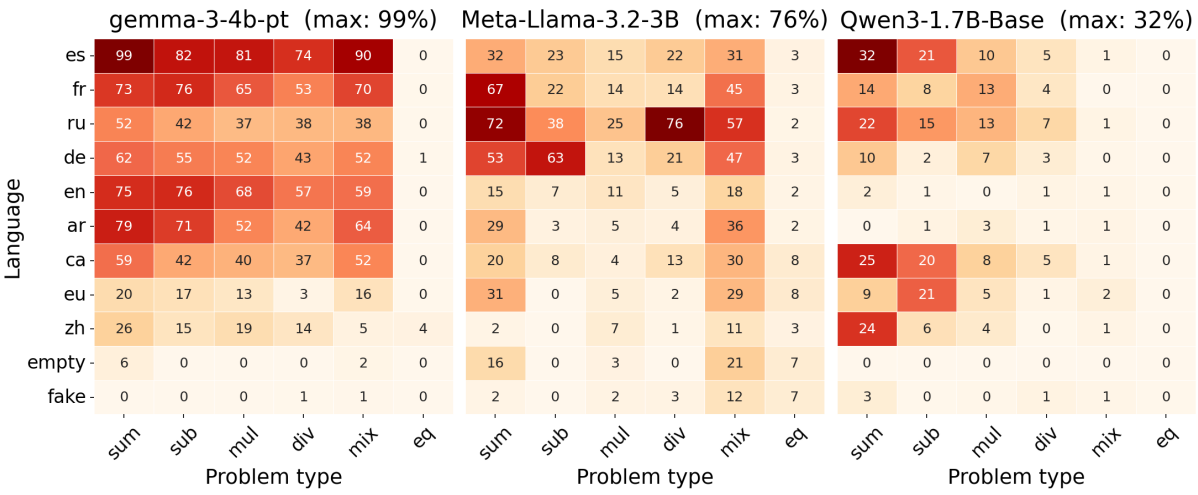


Figure 5: No-answer rates per language and problem type accumulated for all prompt styles.

Model	Top-3 setups	Bottom-3 setups
Llama-3.2-1B	(ar, long_numeric, 0.643); (zh, compute, 0.607); (ar, solve, 0.598)	(en, determine, 0.102); (en, solve, 0.150); (ru, long_numeric, 0.242)
Meta-Llama-3.2-3B	(en, answer, 0.790); (zh, compute, 0.785); (en, compute, 0.735)	(de, determine, 0.108); (ru, solve, 0.118); (ru, determine, 0.218)
Meta-Llama-3.1-8B	(ar, solve, 0.873); (zh, compute, 0.867); (en, answer, 0.848)	(ru, long_numeric, 0.180); (ru, answer, 0.190); (eu, compute, 0.202)
Llama-3.2-1B-Instruct	(es, long_numeric, 0.963); (ru, compute, 0.963); (ru, solve, 0.953)	(ca, solve, 0.822); (fr, compute, 0.827); (ca, long_numeric, 0.830)
Llama-3.2-3B-Instruct	(fr, solve, 0.992); (en, solve, 0.990); (fr, answer, 0.990)	(eu, long_numeric, 0.927); (eu, determine, 0.938); (fake, solve, 0.948)
Meta-Llama-3.1-8B-Instruct	(de, solve, 0.995); (ca, determine, 0.993); (de, long_numeric, 0.993)	(fake, determine, 0.923); (fake, answer, 0.933); (ar, long_numeric, 0.938)
Qwen3-0.6B-Base	(fr, long_numeric, 0.935); (en, long_numeric, 0.933); (en, compute, 0.908)	(empty, answer, 0.473); (fr, determine, 0.507); (de, solve, 0.558)
Qwen3-1.7B-Base	(fr, long_numeric, 0.965); (ru, long_numeric, 0.940); (en, compute, 0.938)	(fake, answer, 0.580); (ca, compute, 0.613); (es, compute, 0.613)
Qwen3-4B-Base	(es, long_numeric, 0.990); (ca, long_numeric, 0.987); (es, answer, 0.977)	(empty, answer, 0.830); (fake, answer, 0.832); (eu, answer, 0.867)
Qwen3-8B-Base	(zh, answer, 0.982); (zh, long_numeric, 0.980); (en, determine, 0.978)	(fake, answer, 0.760); (fake, long_numeric, 0.862); (empty, answer, 0.883)
Qwen3-14B-Base	(en, compute, 0.990); (es, solve, 0.990); (de, solve, 0.988)	(ar, compute, 0.785); (ca, determine, 0.825); (es, determine, 0.825)
Qwen3-0.6B	(zh, compute, 0.983); (zh, determine, 0.982); (eu, answer, 0.980)	(de, answer, 0.895); (fake, long_numeric, 0.917); (eu, determine, 0.927)
Qwen3-1.7B	(fr, solve, 0.993); (en, long_numeric, 0.992); (ca, long_numeric, 0.992)	(fake, answer, 0.835); (eu, long_numeric, 0.942); (fake, solve, 0.950)
Qwen3-4B	(ar, compute, 1.000); (de, answer, 1.000); (ca, answer, 1.000)	(eu, compute, 0.972); (eu, solve, 0.977); (fake, answer, 0.982)
Qwen3-8B	(ar, answer, 1.000); (ar, compute, 1.000); (ar, solve, 1.000)	(ru, compute, 0.963); (eu, answer, 0.988); (fake, determine, 0.988)
Qwen3-14B	(ar, answer, 1.000); (ar, compute, 1.000); (ca, compute, 1.000)	(eu, determine, 0.987); (eu, long_numeric, 0.993); (es, answer, 0.995)
gemma-3-4b-pt	(zh, solve, 0.877); (ca, answer, 0.852); (en, answer, 0.837)	(de, long_numeric, 0.040); (ar, long_numeric, 0.055); (en, long_numeric, 0.058)
gemma-3-12b-pt	(fake, solve, 0.937); (en, answer, 0.933); (zh, solve, 0.923)	(de, long_numeric, 0.010); (fr, long_numeric, 0.020); (ca, long_numeric, 0.067)
gemma-3-4b-it	(en, solve, 1.000); (ca, determine, 0.998); (empty, answer, 0.998)	(ru, long_numeric, 0.773); (en, long_numeric, 0.912); (eu, long_numeric, 0.937)
gemma-3-12b-it	(ar, answer, 1.000); (ar, compute, 1.000); (ar, determine, 1.000)	(ru, long_numeric, 0.973); (en, long_numeric, 0.985); (fake, answer, 0.985)
Teuken-7B-base-v0.6	(en, determine, 0.633); (fr, compute, 0.622); (zh, solve, 0.613)	(fr, long_numeric, 0.182); (de, long_numeric, 0.198); (ru, long_numeric, 0.385)
Teuken-7B-instruct-v0.6	(en, solve, 0.897); (ca, solve, 0.892); (en, long_numeric, 0.885)	(ar, long_numeric, 0.492); (zh, long_numeric, 0.552); (ar, solve, 0.608)
salamandra-2b	(ru, compute, 0.273); (de, solve, 0.263); (zh, compute, 0.258)	(en, solve, 0.012); (fr, long_numeric, 0.017); (en, determine, 0.018)
salamandra-7b	(fake, solve, 0.397); (en, answer, 0.360); (ca, answer, 0.360)	(ar, long_numeric, 0.033); (fr, long_numeric, 0.047); (es, long_numeric, 0.053)
salamandra-2b-instruct	(ca, solve, 0.237); (ca, long_numeric, 0.228); (ca, compute, 0.220)	(fake, compute, 0.035); (zh, compute, 0.038); (zh, solve, 0.040)
salamandra-7b-instruct	(en, long_numeric, 0.557); (ca, long_numeric, 0.502); (en, determine, 0.458)	(ar, answer, 0.252); (fake, answer, 0.278); (fake, solve, 0.292)
EuroLLM-1.7B	(zh, compute, 0.262); (en, answer, 0.260); (en, determine, 0.250)	(fr, long_numeric, 0.012); (de, long_numeric, 0.017); (es, solve, 0.142)
EuroLLM-9B	(ar, solve, 0.802); (en, long_numeric, 0.760); (en, solve, 0.752)	(ru, compute, 0.273); (ca, compute, 0.350); (es, compute, 0.350)
EuroLLM-1.7B-Instruct	(en, solve, 0.400); (en, determine, 0.368); (en, answer, 0.360)	(fake, answer, 0.082); (ca, solve, 0.093); (empty, answer, 0.100)
EuroLLM-9B-Instruct	(ru, solve, 0.953); (fr, compute, 0.932); (en, compute, 0.928)	(eu, long_numeric, 0.670); (eu, answer, 0.688); (fake, answer, 0.753)

Table 13: Top-3 and bottom-3 (language, prompt) setups per model ranked by accuracy under zero-shot base conditions.

Model	Top failing condition	NO-ANS	Rate
gemma-3-4b-it	eu, long_numeric	7	1.2%
gemma-3-12b-it	fake, answer	1	0.2%
Llama-3.2-1B-Instruct	eu, long_numeric	25	4.2%
Llama-3.2-3B-Instruct	fake, solve	15	2.5%
Meta-Llama-3.1-8B-Instruct	fake, determine	24	4.0%
Qwen3-0.6B	fake, long_numeric	9	1.5%
Qwen3-1.7B	eu, long_numeric	26	4.3%
Qwen3-4B	eu, determine	7	1.2%
Qwen3-8B	eu, answer	2	0.3%
Qwen3-14B	ar, long_numeric	1	0.2%

Table 14: Most frequent residual NO-ANSWER condition per instruct model. Most of them are concentrated in Basque, a low-resource language, and fake prompts.

	Gemma-3-4B				Meta-Llama-3.2-3B				Qwen3-1.7B			
Type	ZS	Const.	FS	IT	ZS	Const.	FS	IT	ZS	Const.	FS	IT
sum	0.442	0.857	0.999	0.992	0.669	0.965	1.000	0.991	0.823	0.675	1.000	0.936
sub	0.515	0.936	0.988	0.997	0.838	0.995	1.000	0.998	0.874	0.786	0.993	0.985
mul	0.563	0.905	0.995	0.999	0.624	0.636	0.999	0.994	0.888	0.652	0.998	0.995
div	0.455	0.795	1.000	0.926	0.639	0.874	1.000	0.904	0.751	0.617	1.000	0.952
mix	0.350	0.205	0.457	0.990	0.317	0.069	0.298	0.995	0.813	0.252	0.551	0.984
eq	0.483	0.138	0.510	0.984	0.174	0.082	0.305	0.964	0.667	0.363	0.749	0.996

Table 15: Accuracy by problem type and evaluation condition (ZS = zero-shot base, Const. = constrained decoding, FS = five-shot base, IT = instruction-tuned) for three representative models.

additive operations are most vulnerable, and that the pattern is consistent across languages.

B.4 Correlations between accuracy and error modes

Table 16 reports Pearson correlations between accuracy and NO-ANSWER rate, and between accuracy and ANSWER-WRONG rate, across all 51 (language, prompt) conditions for all 30 models.

Base models show a consistent pattern: strong negative r(acc, no-ans) indicates format failure dominates accuracy variance (Gemma: −0.991, −0.996; Llama: −0.932 to −0.976). The positive r(acc, wrong-ans) in base models reflects format gate dominance: in conditions where the model attempts an answer, both accuracy and wrong-answer rate are higher. Instruct models show the inverse pattern: near-zero r(acc, no-ans) and strong negative r(acc, wrong-ans), confirming that instruction tuning eliminates format failures and exposes computation errors as the remaining source of variance.

European multilingual models (EuroLLM, Salamandra, Teuken) show a different profile: instruct variants have moderate correlations on both dimensions, with neither failure type clearly dominating. This reflects consistently low accuracy across conditions rather than a clean format vs. computation distinction, consistent with the knowledge or capacity limitations discussed in §4.3.

Instruct model correlations for r(acc,no-ans)r(acc, no-ans) r(acc,no-ans) should be interpreted cautiously: NO-ANSWER rates are near-zero (typically under 0.1% of conditions), meaning correlations are computed from very few cases and may not be statistically reliable.

B.5 Examples of step-by-step reasoning in instruction-tuned models

Tables 17 and 18 show representative outputs for mix and eq operations under zero-shot instruction-tuned conditions for each model family, all using the English *answer* prompt. All three models employ step-by-step reasoning to resolve both operation types, in contrast to five-shot base models which produce direct answers incompatible with multi-step resolution. These examples illustrate why instruction tuning, unlike five-shot prompting, recovers accuracy on mix and eq operations. The reason-

Model	N	r(acc, no-ans)	r(acc, wrong-ans)
Llama-3			
Llama-3.2-1B	51	−0.932	0.381
Meta-Llama-3.2-3B	51	−0.949	0.200
Meta-Llama-3.1-8B	51	−0.976	0.451
Llama-3.2-1B-Instruct	51	−0.323	−0.970
Llama-3.2-3B-Instruct	51	−0.545	−0.912
Meta-Llama-3.1-8B-Instruct	51	−0.580	−0.886
Qwen3			
Qwen3-0.6B-Base	51	−0.533	−0.679
Qwen3-1.7B-Base	51	−0.671	−0.593
Qwen3-4B-Base	51	−0.153	−0.941
Qwen3-8B-Base	51	−0.393	−0.882
Qwen3-14B-Base	51	−0.733	−0.535
Qwen3-0.6B	51	−0.317	−0.942
Qwen3-1.7B	51	−0.579	−0.760
Qwen3-4B	51	−0.535	−0.866
Qwen3-8B	51	−0.186	−0.983
Qwen3-14B	51	−0.021	−0.990
Gemma-3			
gemma-3-4b-pt	51	−0.991	0.683
gemma-3-12b-pt	51	−0.996	0.572
gemma-3-4b-it	51	−0.174	−0.996
gemma-3-12b-it	51	−0.263	−0.942
EuroLLM			
EuroLLM-1.7B	51	−0.718	0.486
EuroLLM-9B	51	−0.885	0.161
EuroLLM-1.7B-Instruct	51	−0.338	−0.384
EuroLLM-9B-Instruct	51	−0.344	−0.899
Teuken			
Teuken-7B-base	51	−0.887	0.221
Teuken-7B-instruct	51	−0.705	−0.653
Salamandra			
salamandra-2b	51	−0.664	0.687
salamandra-7b	51	−0.773	0.637
salamandra-2b-instruct	51	−0.623	0.186
salamandra-7b-instruct	51	−0.537	−0.610

Table 16: Pearson correlation between accuracy and error type rates across 51 (language, prompt) conditions. r(acc, no-ans): correlation with NO-ANSWER rate; r(acc, wrong-ans): correlation with ANSWER-WRONG rate.

ing style differs substantially across families. Gemma produces compact inline reasoning: for the mix problem it resolves the expression in a single chained line without explicit narration, while for eq it uses structured markdown steps with bold formatting. Llama produces verbose natural language reasoning, explicitly naming the mathematical operations (order of operations, isolating the variable) before each step. Qwen3, evaluated here with reasoning mode disabled (enable_thinking set to False), produces the most structured output, using markdown headers (`### Step`), LaTeX-formatted equations (`$$...$$`), and a boxed final answer (`\boxed`), suggesting strong pretraining on mathematical document formats. The presence of markdown and LaTeX in Qwen's output without reasoning mode enabled indicates that this structured formatting is part of its instruction-following behavior rather than its reasoning scaffold.

B.6 Constrained decoding results

Table 19 reports mean accuracy and cross-prompt standard deviation under constrained decoding for the three representative models across all languages. These results complement the delta table shown in the main text (Table 1).

Constrained decoding is not universally effective. Some (language, prompt) conditions show high variance or degenerate outputs, particularly when the prompt does not naturally anchor numeric generation. For example, for

Prompt: `Answer: (18 + 22) * 20 =`	
Model	**Output (truncated)**
gemma-3-4b-it	`(18 + 22) * 20 = 40 * 20 = 800\n\n`
Llama-3.2-3B-Instruct	`To solve the equation, we need to follow the order of operations (PEMDAS):\n\n1. Evaluate the expression inside the parentheses: 18 + 22 = 40\n2. Multiply 40 by 20: 40 * 20 = 800\n\nThe answer is 800.`
Qwen3-1.7B	`To solve the expression:\n\n$(18 + 22) \times 20$\n\n1. Add 18 and 22:\n $18 + 22 = 40$\n\n2. Multiply the result by 20:\n $40 \times 20 = 800$\n\n ###Final Answer:\n$\boxed{800}$`

Table 17: Example outputs for a mix-type problem across three instruction-tuned models.

Prompt: `Answer: -5x - 11 = 179, x =`	
Model	**Output (truncated)**
gemma-3-4b-it	`Here's how to solve for x:\n\n1.**Add 11 to both sides:**\n -5x - 11 + 11 = 179 + 11\n -5x = 190\n\n2.**Divide both sides by -5:**\n -5x / -5 = 190 / -5\n x = -38\n\nTherefore, x = **-38**` (…)
Llama-3.2-3B-Instruct	`To solve for x, we need to isolate x.\n\n-5x - 11 = 179\n\nFirst, add 11 to both sides:\n\n-5x = 179 + 11\n-5x = 190\n\nNext, divide both sides by -5:\n\nx = -190 / 5\nx = -38\n\nSo, the value of x is -38.`
Qwen3-1.7B	`We are given the equation:\n\n$$\n-5x - 11 = 179\n$$\n\n### Step 1: Add 11 to both sides to isolate the term with $ x $\n\n$$\n-5x - 11 + 11 = 179 + 11\n$$\n\n$$\n-5x = 190\n$$\n\n### Step 2: Divide both sides by -5\n\n$$\nx = \\frac{190}{-5}\n$$\n\n$$\nx = -38\n$$\n\n###`✅`Final Answer:\n\n$$\n\\boxed{-38}\n$$`

Table 18: Example outputs for an equation-type problem across three instruction-tuned models.

the prompt 求: `81 / 27 =`, Gemma under (zh, *determine*) produces a repetition of one of the operands (`27 27 27 27 27`), while under (zh, *answer*) it produces `3 81 - 27 27 27`, a partially degenerate output whose first token (3) happens to be the correct answer and is therefore classified as ANSWER-CORRECT by the permissive extractor. These cases contribute to the residual standard deviation visible in some conditions in Table 19.

C Mechanistic Analysis

C.1 First token as an answer-mode classifier

Table 20 reports precision and recall for the primary ANSWER and NO-ANSWER tokens identified for each representative model. Precision measures how reliably a given first token predicts a specific output mode; recall measures what proportion of that output mode starts with that token. The gradient from Gemma (near-perfect classifier) to Llama (reliable but imperfect) to Qwen (no reliable NO-ANSWER token) directly reflects the behavioral gradient in format sensitivity identified in 4.1.

C.2 Cosine Curves

Figure 6 shows mean cosine similarity between hidden states at the `=` position across layers, for three condition pair types: within-language different-prompt, cross-lingual same-prompt, and cross-lingual different-prompt. High similarity in early and intermediate layers indicates that the `=` token is processed largely independently of linguistic context. Divergence in final layers, where cross-lingual pairs show lower similarity, confirms that language-specific information accumulates toward the end of the forward pass, motivating the logit lens analysis in Section 5.1.

Model	empty	ar	ca	de	en	es	eu	fr	ru	zh	fake	GLOBAL
gemma-3-4b-pt	0.66	0.64±0.03	0.67±0.03	0.70±0.03	0.66±0.06	0.63±0.07	0.63±0.04	0.64±0.05	0.66±0.04	0.54±0.22	0.62±0.07	0.64±0.08
Meta-Llama-3.2-3B	0.55	0.58±0.04	0.62±0.03	0.60±0.04	0.58±0.12	0.61±0.04	0.62±0.01	0.61±0.06	0.59±0.04	0.62±0.04	0.60±0.03	0.60±0.05
Qwen3-1.7B-Base	0.49	0.66±0.07	0.50±0.20	0.60±0.08	0.52±0.14	0.49±0.19	0.66±0.01	0.46±0.16	0.50±0.19	0.59±0.16	0.59±0.15	0.56±0.15

Table 19: Mean accuracy ± standard deviation per language for constrained decoding.

Model	ANS tokens	Precision (ANS)	Recall (ANS)	NO-ANS tokens	Precision (NO-ANS)	Recall (NO-ANS)
gemma-3-4b	" ", " -"	95.8/99.7%	99.80%	\n\n	99.7	93.90%
llama-3-3b	" ", " -"	96.1/96.1%	94.30%	"?\n", "?", " "	98.6/86.1/76.1%	83.50%
qwen-3-1.7b	" ", " -"	93.0/98.0%	74.40%	" ?"	14.90%	17.60%

Table 20: Precision and recall of the first generated token as a classifier of output mode (ANSWER vs. NO-ANSWER) for each representative model under zero-shot base conditions.

C.3 Logit lens: Llama and Qwen

Figure 7 shows logit lens results for Meta-Llama-3.2-3B and Qwen3-1.7B-Base, complementing the Gemma analysis in Figure 3. In Llama, `=` dominates only the first few layers before giving way to diverse predictions; divergence between ANSWER and NO-ANSWER modes begins around layer 16, with `?\n` emerging as a shared dominant prediction around layer 22 before ANSWER cases transition to numeric tokens. In Qwen, predictions are more diffuse throughout with no clear bifurcation, and the two output modes converge rather than diverge at the final layer — consistent with Qwen's near-zero NO-ANSWER rate and absence of a reliable NO-ANSWER token.

C.4 Steering vectors: full results

Tables 21 and 22 report per-layer accuracy for all steering vector conditions at the `=` position for Gemma-3-4b-pt (layers 2–34) and Meta-Llama-3.2-3B (layers 2–28) respectively, including a second language pair for each model (en,ans)–(es,ans) for Gemma and (zh,ans)–(ru,ans) for Llama. Tables 23 and 24 report the corresponding results when the steering vector is applied at all generation steps rather than only at `=`. Results are consistent across language pairs, confirming that the steering direction captures a model-internal format representation rather than language-specific features. The all-positions negative direction produces stronger effects than `=`-only intervention in both models, as discussed in §5.2.

Model	Steering vec	Eval on	Coeff	L2	L6	L10	L14	L16	L18	L20	L22	L24	L26	L28	L30	L32	L34
Base case	(ca,ans)-(es,ans)	(es,ans)	1	0	0	0.06	0.42	0.49	0.84	0.92	0.92	0.91	0.82	0.83	0.82	0.82	0
	(en,ans)-(es,ans)	(es,ans)	1	0	0	0.27	0.51	0.60	0.85	0.87	0.87	0.80	0.79	0.75	0.78	0.79	0
Within language	(es,determine)-(es,ans)	(es,ans)	1	0	0	0.18	0.45	0.47	0.64	0.81	0.92	0.93	0.82	0.82	0.83	0.81	0
	(ar,solve)-(ar,ans)	(ar,ans)	1	0	0.05	0.24	0.52	0.65	0.86	0.89	0.86	0.86	0.82	0.82	0.82	0.83	0
Reverse direction	(ca,ans)-(es,ans)	(ca,ans)	−1	0.99	1	0.96	0.95	0.95	0.79	0.78	0.48	0.60	0.66	0.46	0.46	0.41	1
	(en,ans)-(es,ans)	(en,ans)	−1	0.98	0.98	0.99	0.99	0.98	0.98	0.99	0.97	0.93	0.97	0.96	0.94	0.95	1
Prompt generalization	(ca,ans)-(es,ans)	(es,compute)	1	0.01	0	0.12	0.62	0.62	0.81	0.83	0.83	0.82	0.79	0.80	0.81	0.79	0
	(en,ans)-(es,ans)	(es,compute)	1	0.01	0	0.24	0.55	0.65	0.81	0.83	0.83	0.80	0.78	0.79	0.79	0.79	0
Language generalization	(ca,ans)-(es,ans)	(ar,ans)	1	0.04	0	0.30	0.69	0.85	0.83	0.83	0.83	0.82	0.80	0.81	0.81	0.81	0.01
	(en,ans)-(es,ans)	(ar,ans)	1	0.01	0.03	0.40	0.71	0.85	0.84	0.84	0.83	0.81	0.80	0.80	0.80	0.81	0.01
FS vector	(fs,es,ans)-(es,ans)	(es,ans)	1	0	0	0.08	0.53	0.51	0.86	0.84	0.83	0.83	0.73	0.75	0.72	0.74	0
	(fs,ar,ans)-(ar,ans)	(ar,ans)	1	0.15	0.13	0.69	0.90	0.89	0.88	0.89	0.88	0.84	0.84	0.84	0.84	0.86	0
IT vector	(it,es,ans)-(es,ans)	(es,ans)	1	0	0	0	0	0	0	0	0.03	0.05	0.02	0.03	0.03	0.04	0
	(it,ar,ans)-(ar,ans)	(ar,ans)	1	0.06	0.01	0	0.01	0.09	0.13	0.40	0.74	0.80	0.69	0.77	0.55	0.03	0.02

Table 21: Per-layer accuracy for Gemma-3-4B steering vector experiments across layers 2-34.

Experiment	Steering vec	Eval on	Coeff	L2	L6	L10	L14	L16	L18	L20	L22	L24	L26	L28
Base case	(en,ans)-(ru,ans)	(ru,ans)	1	0.08	0.42	00.77	0.92	0.90	0.90	0.90	0.90	0.89	0.89	0
	(zh,ans)-(ru,ans)	(ru,ans)	1	0.01	0.43	0.61	0.35	0.84	0.84	0.83	0.97	0.97	0.96	0
Within language	(ru,long_num)-(ru,ans)	(ru,ans)	1	0.30	0.67	0.69	0.45	0.67	0.97	0.88	0.97	0.97	0.97	0
	(de,long_num)-(de,ans)	(de,ans)	1	0.32	0.67	0.69	0.45	0.67	0.97	0.88	0.97	0.97	0.97	0
Reverse direction	(en,ans)-(ru,ans)	(en,ans)	−1	0.99	0.97	0.94	0.94	0.90	0.85	0.86	0.89	0.92	0.89	1
	(zh,ans)-(ru,ans)	(zh,ans)	−1	0.99	0.77	0.78	0.93	0.92	0.91	0.90	0.95	0.95	0.95	1
Prompt generalization	(en,ans)-(ru,ans)	(ru,solve)	1	0.02	0.08	0.43	0.81	0.80	0.74	0.77	0.77	0.77	0.76	0.01
	(zh,ans)-(ru,ans)	(ru,solve)	1	0.02	0.31	0.46	0.27	0.73	0.78	0.66	0.75	0.73	0.73	0.01
Language generalization	(en,ans)-(ru,ans)	(de,ans)	1	0.07	0.41	0.62	0.76	0.74	0.74	0.75	0.76	0.73	0.77	0
	(zh,ans)-(ru,ans)	(de,ans)	1	0.03	0.54	0.55	0.29	0.67	0.77	0.71	0.77	0.77	0.78	0
FS vector	(fs,ru,ans)-(ru,ans)	(ru,ans)	1	0.39	0.87	0.93	0.93	0.91	0.91	0.91	0.90	0.90	0.90	0
	(fs,de,ans)-(de,ans)	(de,ans)	1	0.46	0.92	0.95	0.93	0.93	0.91	0.92	0.91	0.92	0.93	0
IT vector	(it,de,ans)-(de,ans)	(de,ans)	1	0.01	0.51	0.78	0.44	0.47	0.84	0.82	0.84	0.85	0.13	0.01
	(it,ru,ans)-(ru,ans)	(ru,ans)	1	0.11	0.39	0.72	0.75	0.63	0.88	0.88	0.88	0.86	0.86	0.01

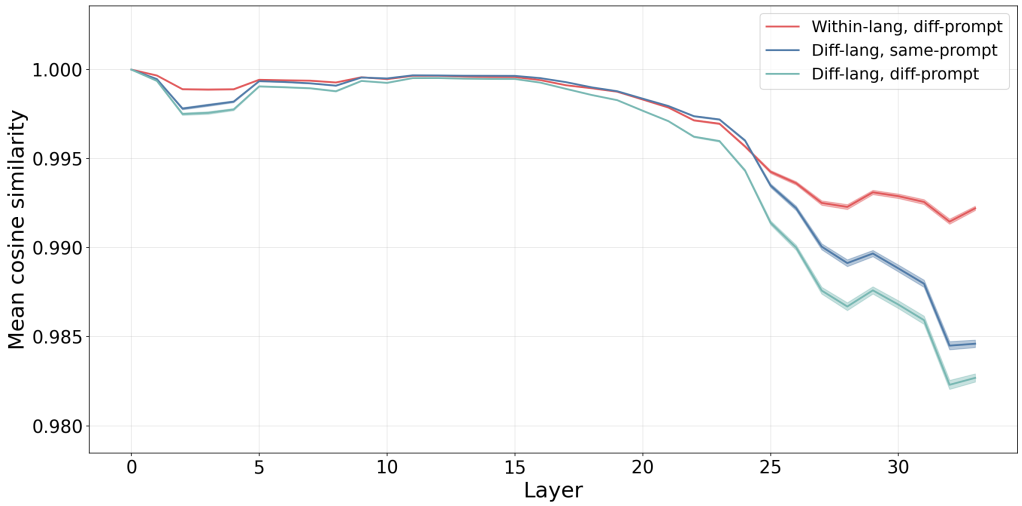
Table 22: Per-layer accuracy for Llama steering vector experiments across layers 2-28.

Experiment	Steering vec	Eval on	Coeff	L2	L6	L10	L14	L16	L18	L20	L22	L24	L26	L28	L30	L32	L34
Base case	(ca,ans)-(es,ans)	(es,ans)	1	0	0	0.06	0.46	0.51	0.79	0.85	0.77	0.83	0.81	0.80	0.80	0.79	0
	(en,ans)-(es,ans)	(es,ans)	1	0	0	0.26	0.49	0.55	0.78	0.77	0.69	0.71	0.71	0.73	0.68	0.71	0
Within language	(es,determine)-(es,ans)	(es,ans)	1	0	0	0.18	0.47	0.49	0.61	0.79	0.88	0.92	0.80	0.77	0.78	0.74	0
	(ar,solve)-(ar,ans)	(ar,ans)	1	0.01	0.05	0.25	0.52	0.65	0.82	0.88	0.69	0.82	0.85	0.87	0.82	0.85	0
Reverse direction	(ca,ans)-(es,ans)	(ca,ans)	−1	0.99	0.99	0.96	0.92	0.92	0.71	0.65	0.28	0.43	0.42	0.22	0.17	0.16	1
	(en,ans)-(es,ans)	(en,ans)	−1	0.97	0.96	0.95	0.94	0.94	0.71	0.77	0.24	0.10	0.13	0.10	0.08	0.09	0.97
Prompt generalization	(ca,ans)-(es,ans)	(es,compute)	1	0	0	0.12	0.63	0.58	0.77	0.77	0.73	0.74	0.76	0.76	0.75	0.74	0
	(en,ans)-(es,ans)	(es,compute)	1	0.01	0	0.23	0.51	0.62	0.72	0.75	0.71	0.66	0.78	0.73	0.70	0.71	0
Language generalization	(ca,ans)-(es,ans)	(ar,ans)	1	0.04	0	0.31	0.70	0.78	0.68	0.72	0.63	0.70	0.72	0.78	0.71	0.77	0
	(en,ans)-(es,ans)	(ar,ans)	1	0.01	0.03	0.40	0.68	0.76	0.70	0.65	0.59	0.67	0.70	0.77	0.69	0.71	0
FS vector	(fs,es,ans)-(es,ans)	(es,ans)	1	0	0	0.07	0.52	0.49	0.72	0.69	0.57	0.73	0.71	0.73	0.75	0.76	0
	(fs,ar,ans)-(ar,ans)	(ar,ans)	1	0.15	0.11	0.66	0.85	0.85	0.76	0.63	0.44	0.70	0.73	0.79	0.80	0.83	0
IT vector	(it,es,ans)-(es,ans)	(es,ans)	1	0	0.01	0	0	0	0	0	0.05	0.03	0.03	0.03	0.04	0	0
	(it,ar,ans)-(ar,ans)	(ar,ans)	1	0.04	0.01	0	0	0.07	0.09	0.35	0.72	0.80	0.69	0.78	0.57	0.01	0.02

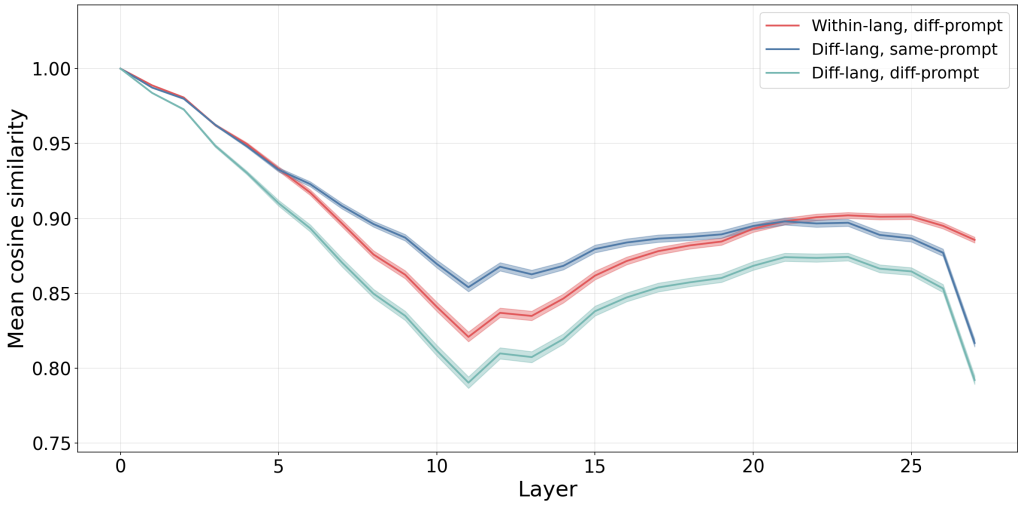
Table 23: Per-layer accuracy for Gemma-3-4B steering vector at all generation positions.

Model	Steering vec	Eval on	Coeff	L2	L6	L10	L14	L16	L18	L20	L22	L24	L26	L28
Base case	(en,ans)-(ru,ans)	(ru,ans)	1	0.12	0.86	0.89	0.88	0.88	0.87	0.90	0.85	0.89	0.92	0
	(zh,ans)-(ru,ans)	(ru,ans)	1	0.01	0.41	0.60	0.35	0.89	0.85	0.88	0.92	0.93	0.96	0
Within language	(ru,long_num)-(ru,ans)	(ru,ans)	1	0.44	0.42	0.46	0.48	0.55	0.96	0.85	0.97	0.99	0.97	0
	(de,long_num)-(de,ans)	(de,ans)	1	0.37	0.67	0.69	0.42	0.62	0.92	0.86	0.94	0.96	0.96	0
Reverse direction	(en,ans)-(ru,ans)	(en,ans)	−1	0.97	0.97	0.90	0.58	0.56	0.22	0.43	0.58	0.39	0.31	1
	(zh,ans)-(ru,ans)	(zh,ans)	−1	0.98	0.81	0.78	0.90	0.54	0.83	0.82	0.79	0.87	0.87	1
Prompt generalization	(en,ans)-(ru,ans)	(ru,solve)	1	0.02	0.46	0.73	0.82	0.81	0.76	0.75	0.74	0.78	0.81	0.01
	(zh,ans)-(ru,ans)	(ru,solve)	1	0.02	0.29	0.44	0.27	0.73	0.76	0.69	0.77	0.76	0.81	0.01
Language generalization	(en,ans)-(ru,ans)	(de,ans)	1	0.07	0.71	0.73	0.77	0.74	0.77	0.81	0.77	0.78	0.79	0
	(zh,ans)-(ru,ans)	(de,ans)	1	0.03	0.58	0.57	0.29	0.68	0.75	0.70	0.78	0.77	0.74	0
FS vector	(fs,ru,ans)-(ru,ans)	(ru,ans)	1	0.37	0.89	0.92	0.85	0.83	0.83	0.88	0.79	0.90	0.92	0
	(fs,de,ans)-(de,ans)	(de,ans)	1	0.47	0.92	0.79	0.85	0.78	0.83	0.89	0.92	0.92	0.89	0
IT vector	(it,de,ans)-(de,ans)	(de,ans)	1	0.02	0.50	0.76	0.42	0.47	0.83	0.83	0.81	0.84	0.15	0.01
	(it,ru,ans)-(ru,ans)	(ru,ans)	1	0.11	0.38	0.69	0.75	0.62	0.88	0.82	0.76	0.85	0.89	0.01

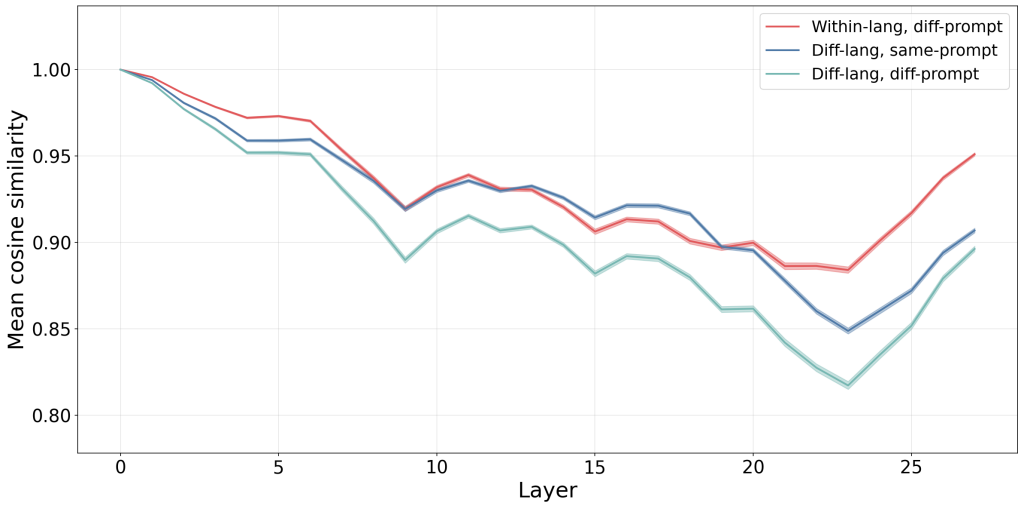
Table 24: Per-layer accuracy for Llama-3.2-3B steering vector at all generation positions.



(a) Gemma-3-4B



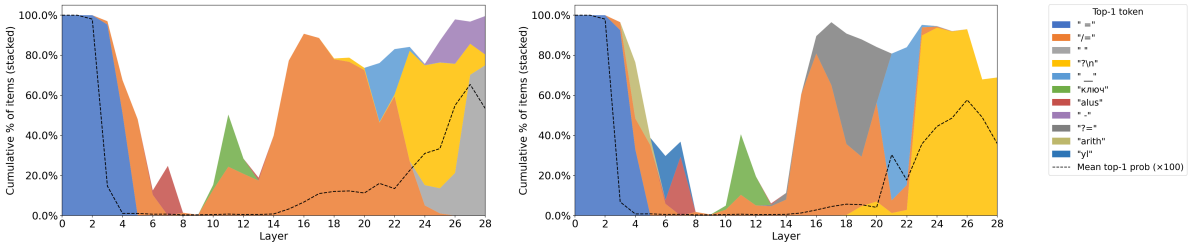
(b) Llama-3.2-3B



(c) Qwen3-1.7B-Base

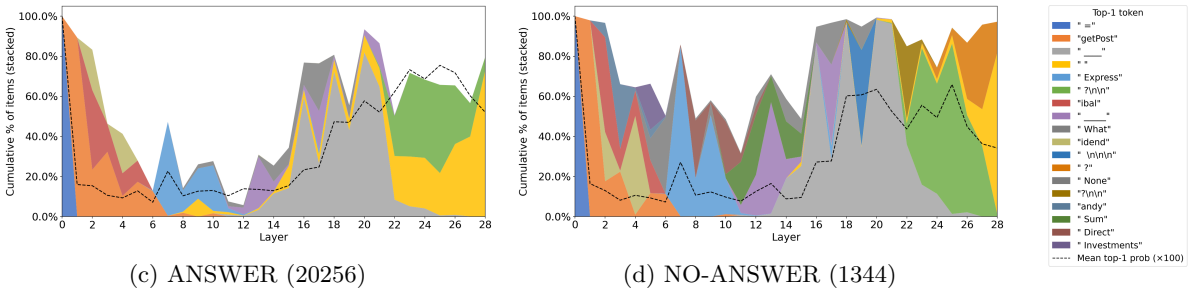
Figure 6: Mean cosine similarity between hidden states at the = position across layers, computed for within-language different-prompt (blue), cross-lingual same-prompt (orange), and cross-lingual different-prompt (green) condition pairs.

Llama-3-3b-base



(a) ANSWER (18148) (b) NO-ANSWER (3452)

Qwen-3-1.7B-Base



(c) ANSWER (20256) (d) NO-ANSWER (1344)

Figure 7: Per-layer top-1 token frequency (20% peak threshold), split by answer category. Coloured areas indicate the share of examples predicting each token as top-1 at each layer; the dashed line shows mean top-1 softmax probability (×100).