

Unveiling Sentiment Representations in Code-LLMs

Yongyi Ji[†] Fuxiang Chen^{†,*} Jaegul Choo[‡] Seung-won Hwang[§] Lu Liu[¶]

[†]University of Leicester [‡]KAIST [§]Seoul National University [¶]University of Exeter
{yj171, fuxiang.chen}@leicester.ac.uk, jchoo@kaist.ac.kr
seungwonh@snu.ac.uk, L.Liu3@exeter.ac.uk

Abstract

This paper examines whether Code-LLMs encode latent sentiment representations and how these influence generation behavior. Using probing methods, we show that sentiment information is linearly separable in hidden states. Through activation steering, we demonstrate that manipulating sentiment-aligned directions systematically affects both output tone and downstream task performance. Notably, negative sentiment consistently reduces correctness, while positive effects are weaker and less consistent. Ablation and error analyses indicate these changes are not due to random perturbations but reflect meaningful behavioral shifts, including reduced confidence. These findings suggest that affective latent directions act as unintended control signals in Code-LLMs, with measurable impact on downstream task performance quality. Our study highlights the need to better understand and control such latent factors when deploying Code-LLMs. We open-sourced the scripts used in our experiment. ¹

1 Introduction

Large language models (LLMs) achieve strong performance across diverse tasks due to training on large-scale human-written corpora that contain rich linguistic and affective patterns (Hou et al., 2024; Zhao et al., 2023; Minaee et al., 2024; Dodge et al., 2021). As a result, LLMs may implicitly encode affective signals that influence both their internal representations and downstream behavior. Prior work shows that general-purpose LLMs are sensitive to emotional cues, affecting reasoning and decision-making (Mozikov et al., 2024). However, the role of such affective information in code-oriented models remains largely unexplored.

Code-LLMs extend LLMs to software engineering tasks, often trained on developer-generated corpora such as GitHub and Stack Overflow, where sentiment is pervasive (Li et al., 2023c; Guo et al., 2024; Coutinho et al., 2024; Nam et al., 2025). This raises a key question: *does sentiment function as a latent factor in Code-LLMs that influences both their representations and task performance?* Although code generation is typically treated as a structured task, Figure 1 shows that negative sentiment prompts can degrade performance, suggesting that sentiment acts as an unintended control signal.

In this work, we present a systematic study of sentiment as a latent factor in Code-LLMs, with comparisons to general-purpose LLMs. We investigate three research questions: (1) what sentiment tendencies are reflected in model outputs (Section 3), (2) whether sentiment is encoded in hidden representations (Section 4), and (3) whether sentiment directions can be causally manipulated to affect downstream behavior (Section 5). We combine sentiment measurement (PANAS, HAAS), probing-based representation analysis, and activation steering to perform controlled interventions.

Our results show that: (1) LLM outputs exhibit non-neutral sentiment tendencies, with Code-LLMs generally more positive; (2) sentiment information is reliably encoded in hidden representations; and (3) sentiment-aligned steering causally affects model behavior, with negative sentiment consistently degrading code generation performance.

These findings suggest that sentiment acts as an unintended latent control signal in Code-LLMs, influencing both generation characteristics and task outcomes. More broadly, our results highlight the importance of understanding non-task-related latent factors in LLMs and their implications for reliability.

Our contributions are threefold: (i) we characterize sentiment tendencies in Code-LLMs and

¹<https://github.com/yongyiji/Code-LLM-Sentiment>

*Corresponding Author

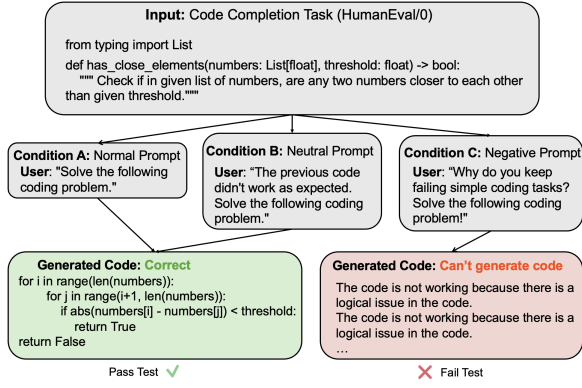


Figure 1: Different sentiment prompts for code generation using CodeGemma 7B on the HumanEval dataset.

general-purpose LLMs; (ii) we demonstrate that sentiment is encoded in their representations through validated probing; and (iii) we show that sentiment-aligned directions can be used to causally steer model behavior, with measurable effects on downstream tasks.

2 Related Work

Probing and LLMs: Probing methods analyze how information is encoded in LLM representations and have been widely applied to general-purpose models (Alain and Bengio, 2016). While prior work shows that various properties, including sentiment, are encoded in such models (Holtermann et al., 2025; Beyer and Reed, 2025; Conneau et al., 2018; Vulić et al., 2020; Youssef et al., 2023; Zhao et al., 2024; Di Palma et al., 2025), probing studies on Code-LLMs have focused mainly on code-related properties (Anand et al., 2024). Given their training on developer text from platforms such as GitHub and Stack Overflow (Roziere et al., 2023; Lozhkov et al., 2024) where the developer-generated text frequently contains sentiment (Ortu et al., 2016; Girardi et al., 2020; Guzman et al., 2014; El Asri et al., 2019; Murgia et al., 2014; Pletea et al., 2014), it remains unclear how sentiment is represented in Code-LLMs.

Emotion in LLMs: Prior research shows that general-purpose LLMs exhibit emotional intelligence and are sensitive to emotional cues in prompts, which can influence decision-making, confidence, and performance (Wang et al., 2023; Li et al., 2023a; Wang et al., 2024). Emotional prompting and stylistic variations can alter behavior, though effects on task accuracy are often limited (Mozikov et al., 2024; Ma et al., 2025). However, these studies largely focus on general-

purpose models, and the impact of sentiment on Code-LLMs remains underexplored.

LLMs steering: Previous work shows that LLM outputs can be influenced through techniques such as fine-tuning, prompt modification, and inference-time interventions, grounded in the Linear Representation Hypothesis (Park et al., 2024), which posits that concepts are encoded as linear directions in model representations (Turner et al., 2023; Li et al., 2025). Approaches like activation steering use these directions to control behaviors, including sentiment and factuality (Li et al., 2023b; Zou et al., 2023). Prior studies demonstrate that such methods can meaningfully alter tone and outputs (Zou et al., 2023; Li et al., 2023b). Building on this, the present work investigates whether sentiment-aligned directions can be used to steer LLM behavior and affect performance on code-related tasks.

3 Measuring Code-LLMs’ Sentiment

3.1 Default Sentiment Measurement

To characterize the default sentiment tendencies reflected in the outputs of Code-LLMs and general-purpose LLMs, we employ two widely used affective assessment instruments: the Positive And Negative Affect Schedule (PANAS) (Watson et al., 1988) and the Hedonic and Arousal Affect Scale (HAAS) (Roca et al., 2023). Although these instruments were originally designed for human self-report, prior work has adapted similar protocols to analyze affective patterns in LLM-generated text (Huang et al., 2024; Tavast et al., 2022). We emphasize that we do not interpret these results as evidence of genuine emotional states; rather, we treat them as proxies for observable sentiment characteristics reflected in model outputs.

PANAS consists of 20 items rated on a 5-point Likert scale (1–5), measuring the extent to which specific affective descriptors apply. It produces two independent scores: Positive Affect (PA), capturing expressions such as enthusiasm and alertness, and Negative Affect (NA), capturing states such as distress and anger. Importantly, PA and NA are not opposing ends of a single axis but are modeled as orthogonal dimensions. Following prior work (Sibille et al., 2012), we used the Affect Balance Style (ABS) framework to classify PA and NA levels based on established thresholds: $PA > 35$ indicates high positive affect, while $NA > 18$ indicates high negative affect. We further the map PANAS outcomes to four affect bal-

ance styles (high-PA/high-NA; high-PA/low-NA; low-PA/high-NA; low-PA/low-NA) to provide interpretable groupings. Detailed styles are provided in Appendix C.

HAAS consists of 12 items rated on a 5-point scale (0–4) and measures four affective dimensions: positive high arousal, positive low arousal, negative high arousal, and negative low arousal. These dimensions capture both valence and arousal aspects of sentiment expression.

We note that these human-derived scales may not directly transfer to LLMs. Accordingly, we treat PANAS and HAAS outputs as qualitative indicators rather than precise psychological measurements. In practice, we prompt LLMs to generate responses corresponding to these instruments and analyze the resulting patterns. Prompt templates and experimental details are provided in Appendix B and Appendix C.

3.2 Models

We focus our analysis on open-source general-purpose LLMs and Code-LLMs, as intermediate representations are inaccessible in closed-source LLMs. Details of the models used in our experiments are shown in Table 1. To examine whether Code-LLMs encode sentiment information in a manner similar to general-purpose LLMs, we select paired models from the same architectural family, enabling controlled comparisons that isolate the effect of code-specific adaptation.

Table 1: LLMs used for sentiment probing.

Code-LLMs	General-LLMs
CodeLlama 7B IT (Roziere et al., 2023)	LlaMa2 7B IT (Touvron et al., 2023)
CodeGemma 7B IT (Team et al., 2024a)	Gemma 7B IT (Team et al., 2024b)
DeepSeek-Coder V2 Lite IT (Zhu et al., 2024)	DeepSeek V2 Lite Chat (Liu et al., 2024)
Qwen3-Coder Next (Cao et al., 2026)	Qwen3 Next (Team, 2025)
Codestral 22B (Mistral, 2025)	Mistral 7B (Jiang et al., 2023)

3.3 Results

Table 2 reports the default sentiment characteristics of LLM outputs as measured by PANAS and HAAS. Under PANAS, most evaluated LLMs—particularly Code-LLMs—produce responses corresponding to the *High-PA/High-NA* affect pattern, indicating relatively high emotional

Table 2: Output-level sentiment tendencies of (non-steered) LLMs measured by PANAS and HAAS. HAAS values are formatted as High / Low.

LLMs	PANAS			HAAS	
	PA	NA	Pattern	Pos (H/L)	Neg (H/L)
CodeLlama 7B	37	33	H-PA/H-NA	9 / 6	6 / 6
CodeGemma 7B	39	13	H-PA/L-NA	9 / 2	3 / 6
DeepSeek-Coder V2	40	24	H-PA/H-NA	4 / 4	2 / 1
Qwen3-Coder Next	38	12	H-PA/L-NA	12 / 10	4 / 6
Codestral 22B	43	20	H-PA/H-NA	6 / 9	3 / 4
LlaMa2 7B	34	21	L-PA/H-NA	4 / 5	3 / 5
Gemma 7B	29	17	L-PA/L-NA	6 / 10	1 / 10
DeepSeek V2	30	18	L-PA/L-NA	8 / 7	2 / 1
Qwen3 Next	46	13	H-PA/L-NA	11 / 10	4 / 6
Mistral 7B	36	18	H-PA/L-NA	7 / 12	3 / 4

intensity in generated text. Notably, Code-LLMs generally exhibit higher positive affect (PA) scores than general-purpose LLMs: all Code-LLMs exceed the conventional PA threshold of 35, whereas most general-purpose models fall below this level. Negative affect (NA) scores show greater variation. Some Code-LLMs, such as CodeGemma 7B and Qwen3-Coder Next, maintain relatively low NA levels, whereas CodeLlama 7B exhibits a substantially higher NA score. Among general-purpose models, LLaMA2 7B stands out with comparatively elevated negative affect. Overall, Code-LLMs and general-purpose LLMs display distinct default sentiment profiles. For example, within the DeepSeek family, Code-LLMs are typically classified as *High-PA/High-NA*, while their general-purpose counterparts tend to fall into the *Low-NA/Low-PA* affect category. Similarly, within the Mistral family, Code-LLMs are predominantly classified as *High-PA/High-NA*, whereas general-purpose models more often fall into the *High-PA/Low-NA* category. Results from HAAS are broadly consistent with PANAS. For instance, Qwen3-Coder Next and Qwen3 Next exhibit similar sentiment tendencies across both measures, characterized by high positive and low negative affect. Some discrepancies arise due to differences in scale construction. For example, Gemma 7B is categorized as *Low-PA/Low-NA* in PANAS but shows relatively higher HAAS scores, primarily driven by low-arousal dimensions rather than high-arousal affect. Given the overall consistency between PANAS and HAAS, we focus on PANAS in subsequent analyses due to its wider use and interpretability in prior work. These findings indicate that LLM outputs exhibit measurable and non-neutral sentiment characteristics, with Code-LLMs tending to produce more positively skewed

and emotionally expressive responses than general-purpose LLMs. We further assess the robustness of these output-level measurements by varying the PANAS and HAAS question orders. Although absolute scores show sensitivity to question ordering, output-level sentiment tendency of Code-LLMs indicates higher positive affect than general-purpose LLMs in several model families (Appendix C).

4 Sentiment Detection in LLMs

We adopt the sentiment detection methodology proposed by Palma et al. (Di Palma et al., 2025) to analyze sentiment-related information in LLMs. Let $D = \{(x_i, y_i)\}_{i=1}^n$ denote a sentiment dataset, where y_i denotes the sentiment label of sentence x_i . Let $\text{rep}_\theta(x_i)$ denote an intermediate representation of x_i obtained from a forward pass through LLM_θ . The goal of sentiment detection is to extract these representations $\{\text{rep}_\theta(x_i)\}_{i=1}^n$, and train a classifier C_w parameterized by weights w , to predict sentiment labels from the representations. Concretely, this approach involves extracting layer-wise hidden representations from LLMs and training probing classifiers on top of these representations to evaluate whether sentiment information is encoded in the model’s internal states.

4.1 Choice of Representation

Modern LLMs are based on the Transformer architecture (Vaswani et al., 2017), which provides a consistent representation space across layers. Let $x^{(l)} \in \mathbb{R}^{T \times d}$ denote the hidden representation at layer l , where T is the sequence length and d is the hidden dimension. Each Transformer layer applies self-attention and a feed-forward network with residual connections, producing the updated representation $x^{(l+1)}$.

Although architectural details may vary across models (Table 3), $x^{(l+1)}$ consistently represents the full sequence output of the l -th layer. Prior work has explored different representation choices, including normalized residual streams (von Rütte et al., 2024), raw residual streams (Zou et al., 2023), and attention heads (Li et al., 2023b). In this work, following Palma et al. (Di Palma et al., 2025), we use the residual stream representation, i.e., $\text{rep}_\theta(x) = x^{(l)}$, as it has been shown to perform well for sentiment detection tasks.

Rather than using the full sequence representation $x^{(l)} \in \mathbb{R}^{T \times d}$, we extract token-level subsets $x_{\text{rep}} \in \mathbb{R}^{t \times d}$ with $t \leq T$, allowing the representa-

tion to focus on parts of the input most relevant to sentiment (von Rütte et al., 2024; Di Palma et al., 2025). We consider three commonly used aggregation strategies: mean pooling, max pooling, and last-token pooling.

Table 3: Transformer architectural variants of LLMs.

LLMs	Norm	f_{attn}	f_{ffn}
CodeLlama 7B	RMSNorm	MHA	SwiGLU
LlaMa2 7B	RMSNorm	GQA	SwiGLU
Codestral 22B	RMSNorm	GQA	SwiGLU
Mistral 7B	RMSNorm	GQA	SwiGLU
Qwen3-Coder Next	RMSNorm	Hybrid	MoE
Qwen3 Next	RMSNorm	Hybrid	MoE
DeepSeek-Coder V2	LayerNorm	MLA	DeepSeekMoE
DeepSeek V2	LayerNorm	MLA	DeepSeekMoE
CodeGemma 7B	RMSNorm	GQA	GeGLU
Gemma 7B	RMSNorm	GQA	GeGLU

4.2 Choice of Classifier

Given the extracted representations, we train layer-wise classifiers $C_w^{(l)}$ to assess whether sentiment information is encoded in the model’s internal states. We consider a range of linear and non-linear classifiers, including Logistic Regression, Linear SVM, Kernel SVM, Random Forest, and Multi-Layer Perceptron (MLP). Model performance is evaluated using F1 score on held-out test data.

Table 4: Best probing results across different pooling methods and classifiers for each model on the SST-2 and Senti4SD datasets. ID-F1 indicates in-domain evaluation (trained and tested on the same dataset), and CD-F1 indicates cross-domain evaluation (evaluated on the other dataset).

LLMs	Dataset	ID-F1	Pooling	Probe	layer	CD-F1
CodeLlama 7B	Senti4sd	0.87	Mean	L-SVM	10	0.65
	SST-2	0.94	Last	Logistic Reg.	24	0.85
LlaMa2 7B	Senti4sd	0.87	Mean	L-SVM	4	0.57
	SST-2	0.96	Mean	L-SVM	14	0.70
Codestral 22B	Senti4sd	0.88	Max	Logistic Reg.	8	0.41
	SST-2	0.95	Last	MLP	33	0.82
Mistral 7B	Senti4sd	0.88	Mean	L-SVM	5	0.7
	SST-2	0.96	Last	L-SVM	16	0.88
Qwen-Coder Next	Senti4sd	0.85	Mean	L-SVM	5	0.55
	SST-2	0.93	Mean	NL-SVM	37	0.82
Qwen Next	Senti4sd	0.86	Mean	L-SVM	1	0.38
	SST-2	0.94	Mean	NL-SVM	37	0.76
DeepSeek-Coder V2	Senti4sd	0.85	Mean	L-SVM	14	0.44
	SST-2	0.94	Last	L-SVM	11	0.77
DeepSeek V2	Senti4sd	0.85	Mean	L-SVM	12	0.44
	SST-2	0.95	Last	Logistic Reg.	25	0.87
CodeGemma 7B	Senti4sd	0.88	Mean	Logistic Reg.	9	0.49
	SST-2	0.94	Last	L-SVM	24	0.85
Gemma 7B	Senti4sd	0.87	Mean	L-SVM	10	0.52
	SST-2	0.95	Mean	Logistic Reg.	20	0.81

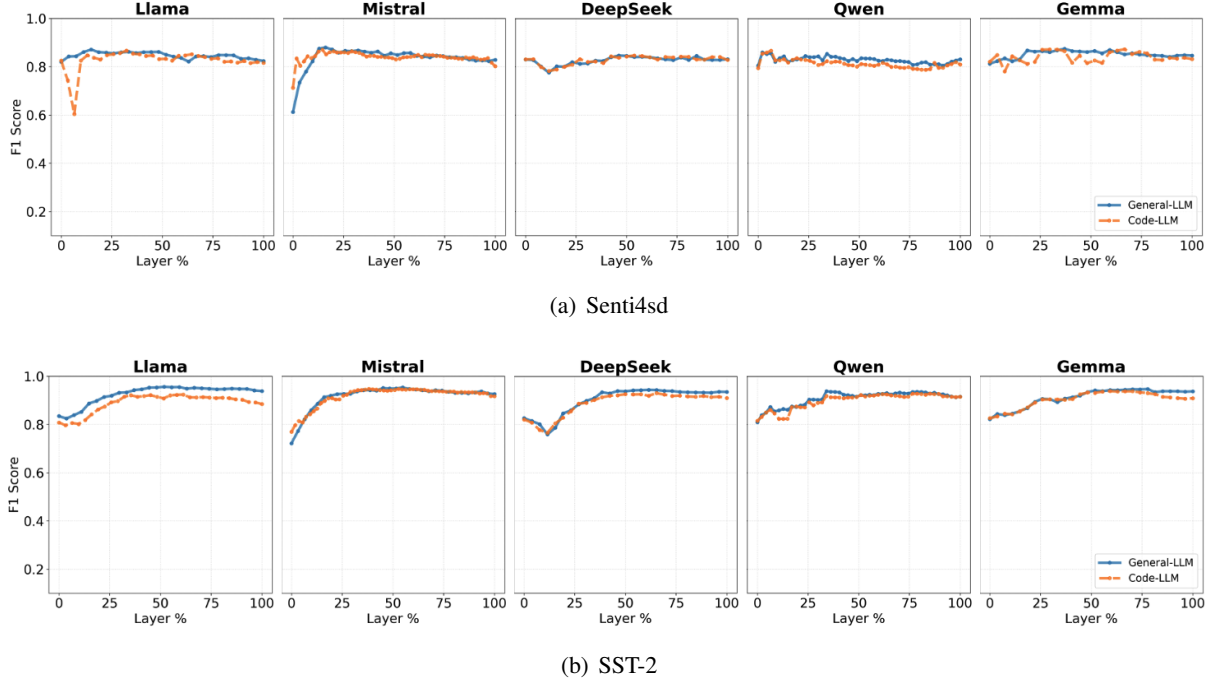


Figure 2: Layer-wise probing results using mean pooling and linear SVM classifier on the SST-2 and Senti4SD datasets are shown in Figures 2(a) and 2(b), respectively. The x-axis represents the relative layer index due to varying numbers of layers across models.

4.3 Datasets

We evaluate sentiment detection on two datasets spanning general-domain and software engineering domains. The SST-2 dataset (Socher et al., 2013) contains 8,741 movie reviews with binary sentiment labels (positive or negative) and is widely used for sentiment classification. Senti4SD (Calefato et al., 2018) contains 4,423 Stack Overflow posts annotated with positive, negative, or neutral sentiment.

Both datasets are split into training and test sets using an 80/20 split. These datasets enable us to evaluate whether sentiment representations learned by LLMs generalize across domains, including developer-generated text.

4.4 Results

Probe Validation. To ensure that the probing results reflect genuine sentiment information rather than artifacts of the representation space, we conduct several validation checks. First, we evaluate probe performance against a random-label baseline, where labels are shuffled; probe accuracy drops to near chance level, indicating that the classifiers do not exploit spurious patterns. Second, we test cross-domain generalization by training probes on SST-2 and evaluating on Senti4SD (and vice versa),

observing consistent performance, which suggests that the learned representations capture general sentiment features rather than dataset-specific cues. Third, we compare against random feature baselines obtained by shuffling or projecting representations, which yield substantially lower performance, confirming that sentiment information is encoded in the original model representations. Together, these results provide evidence that the probes capture meaningful and generalizable sentiment information in LLM representations.

Table 4 presents the best probing results for sentiment detection across Code-LLMs and general-purpose LLMs. Both model types achieve high F1 scores (typically exceeding 0.8), indicating that sentiment information is readily detectable in their hidden representations.

We further evaluate cross-domain generalization by training a classifier on one dataset and testing it on the other. Models trained on the general-domain dataset (SST-2) retain strong performance when evaluated on the software engineering dataset (Senti4SD), suggesting that the probes capture general sentiment features rather than dataset-specific cues. In contrast, models trained on Senti4SD exhibit reduced performance when evaluated on SST-2, likely due to the domain-specific characteristics

Table 5: PANAS-measured output-level sentiment tendencies of LLMs after positive or negative steering using probes trained on the SST-2 or Senti4SD datasets.

LLMs	SST-2 Steering						Senti4SD Steering					
	Positive Steering			Negative Steering			Positive Steering			Negative Steering		
	PA	NA	Pattern	PA	NA	Pattern	PA	NA	Pattern	PA	NA	Pattern
CodeLlama 7B	43 (↑ 6)	29 (↓ 4)	H-PA/H-NA	31 (↓ 6)	29 (↓ 4)	L-PA/H-NA	42 (↑ 5)	37 (↑ 4)	H-PA/H-NA	37 (0)	34 (↑ 1)	H-PA/H-NA
CodeGemma 7B	44 (↑ 5)	13 (0)	H-PA/L-NA	39 (0)	15 (↑ 2)	H-PA/L-NA	38 (↓ 1)	13 (0)	H-PA/L-NA	39 (0)	15 (↑ 2)	H-PA/L-NA
DeepSeek-Coder V2	46 (↑ 6)	27 (↑ 3)	H-PA/H-NA	19 (↓ 21)	14 (↓ 10)	L-PA/L-NA	41 (↑ 1)	25 (↑ 1)	H-PA/H-NA	33 (↓ 7)	35 (↑ 11)	L-PA/H-NA
Qwen3-Coder Next	39 (↑ 1)	16 (↑ 4)	H-PA/L-NA	18 (↓ 20)	31 (↑ 19)	L-PA/H-NA	45 (↑ 7)	14 (↑ 2)	H-PA/L-NA	35 (↓ 3)	17 (↑ 5)	L-PA/L-NA
Codestral 22B	44 (↑ 1)	21 (↑ 1)	H-PA/H-NA	28 (↓ 15)	26 (↑ 6)	L-PA/H-NA	42 (↓ 1)	21 (↑ 1)	H-PA/H-NA	38 (↓ 5)	24 (↓ 4)	H-PA/H-NA
LlaMa2 7B	43 (↑ 9)	28 (↑ 7)	H-PA/H-NA	25 (↓ 9)	24 (↑ 3)	L-PA/H-NA	32 (↓ 2)	21 (0)	L-PA/H-NA	19 (↓ 15)	26 (↑ 5)	L-PA/H-NA
Gemma 7B	30 (↑ 1)	18 (↑ 1)	L-PA/L-NA	27 (↓ 2)	18 (↑ 1)	L-PA/L-NA	28 (↓ 1)	10 (↓ 7)	L-PA/L-NA	31 (↑ 2)	17 (0)	L-PA/L-NA
DeepSeek V2	33 (↑ 3)	18 (0)	L-PA/L-NA	15 (↓ 15)	19 (↑ 1)	L-PA/H-NA	39 (↑ 9)	18 (↓ 1)	H-PA/L-NA	30 (0)	19 (↑ 1)	L-PA/H-NA
Qwen3 Next	48 (↑ 2)	14 (↑ 1)	H-PA/L-NA	37 (↓ 9)	17 (↑ 4)	L-PA/L-NA	46 (0)	12 (↓ 1)	H-PA/L-NA	42 (↓ 4)	36 (↑ 23)	H-PA/H-NA
Mistral 7B	38 (↑ 2)	20 (↑ 2)	H-PA/H-NA	39 (↑ 3)	50 (↑ 32)	H-PA/H-NA	34 (↓ 2)	15 (↓ 3)	L-PA/L-NA	20 (↓ 16)	21 (↑ 3)	L-PA/H-NA

Table 6: HAAS-measured output-level sentiment tendencies of LLMs after positive or negative steering using probes trained on the SST-2 or Senti4SD datasets. Values are formatted as High / Low.

LLMs	SST-2 Steering				Senti4SD Steering			
	Positive Steer		Negative Steer		Positive Steer		Negative Steer	
	Pos (H/L)	Neg (H/L)	Pos (H/L)	Neg (H/L)	Pos (H/L)	Neg (H/L)	Pos (H/L)	Neg (H/L)
CodeLlama 7B	9(0) / 6(0)	6(0) / 6(0)	9(0) / 6(0)	6(0) / 7(↑1)	9(0) / 8(↑2)	6(0) / 7(↑1)	9(0) / 6(0)	7(↑1) / 8(↑2)
CodeGemma 7B	6(↓3) / 9(↑7)	2(↓1) / 3(↓3)	7(↓2) / 6(↑4)	2(↓1) / 6(0)	9(0) / 3(↑1)	1(↓2) / 3(↓3)	6(↓3) / 6(↑4)	2(↓1) / 5(↓1)
DeepSeek-Coder V2	9(↑5) / 9(↑5)	2(0) / 3(↑2)	0(↓4) / 1(↓3)	1(↓1) / 1(0)	6(↑2) / 5(↑1)	3(↑1) / 4(↑3)	5(↑1) / 3(↓1)	9(↑7) / 7(↑6)
Qwen3-Coder Next	9(↓3) / 9(↓1)	2(↓2) / 4(↓2)	0(↓12) / 7(↓3)	0(↓4) / 2(↓4)	9(↓3) / 8(↓2)	2(↓2) / 4(↓2)	7(↓5) / 6(↓4)	3(↓1) / 6(0)
Codestral 22B	6(0) / 9(0)	3(0) / 3(↓1)	3(↓3) / 7(↓2)	6(↑3) / 5(↑1)	6(0) / 9(0)	3(0) / 3(↓1)	4(↓2) / 7(↓2)	6(↑3) / 4(0)
LlaMa2 7B	7(↑3) / 6(↑1)	2(↓1) / 5(0)	5(↑1) / 4(↓1)	4(↑1) / 5(0)	6(↑2) / 6(↑1)	5(↑2) / 6(↑1)	5(↑1) / 6(↑1)	7(↑4) / 6(↑1)
Gemma 7B	6(0) / 8(↓2)	2(↑1) / 7(↓3)	5(↓1) / 8(↓2)	1(0) / 8(↓2)	6(0) / 11(↑1)	0(↓1) / 3(↓7)	5(↓1) / 5(↓5)	6(↑5) / 6(↓4)
DeepSeek V2	8(0) / 9(↑1)	2(0) / 2(↑1)	3(↓5) / 3(↓4)	3(↑1) / 3(↑2)	9(↑1) / 12(↑5)	4(↑2) / 4(↑3)	8(0) / 7(0)	2(0) / 1(0)
Qwen3 Next	8(↓3) / 11(↑1)	1(↓3) / 3(↓3)	0(↓11) / 2(↓8)	0(↓4) / 3(↓3)	8(↓3) / 9(↓1)	1(↓3) / 4(↓2)	6(↓5) / 5(↓5)	7(↑3) / 7(↑1)
Mistral 7B	7(0) / 11(↓1)	2(↓1) / 4(0)	6(↓1) / 9(↓3)	6(↑3) / 11(↑7)	7(0) / 10(↓2)	1(↓2) / 4(0)	6(↓1) / 8(↓4)	8(↑5) / 7(↑3)

of developer-generated text. The results of probing using label shuffling and feature shuffling show very low F1 scores (mostly below 0.4 on Senti4SD dataset and below 0.68 for SST-2 dataset), as shown in Appendix D.

Figure 2 presents layer-wise probing results, illustrating how sentiment information is distributed across model depth. Code-LLMs exhibit patterns similar to those of general-purpose LLMs, with F1 scores largely overlapping across model families. For most models, sentiment information is most strongly detectable in middle-to-late layers, consistent with prior findings on representation learning in LLMs (Di Palma et al., 2025).

Overall, the strong and consistent probing performance across models, datasets, and layers suggests that both Code-LLMs and general-purpose LLMs encode sentiment information in their internal representations. These results are consistent with the Linear Representation Hypothesis, indicating that sentiment is represented as approximately linear directions in the residual stream.

5 Sentiment Steering

Prior work has shown that linear directions in representation space can be leveraged to control model

behavior during inference (von Rütte et al., 2024). Building on this, we investigate whether sentiment-related directions identified through probing can be used to steer the behavior of Code-LLMs. We further examine whether such steering affects performance on downstream code-related tasks.

5.1 Steering Method

We adopt an inference-time activation steering approach that modifies the residual stream of LLMs using sentiment-aligned directions extracted from probing.

Steering Vector Extraction: For a target layer l and sentiment class (e.g., positive or negative), we extract the learned weight vector $w \in \mathbb{R}^d$ from the best-performing linear probe identified during the detection phase.

Activation Intervention:

Given a layer-wise sentiment vector w , we modify the residual representation to amplify or suppress sentiment features. Specifically, at layer l , we update the hidden representation $x^{(l)}$ as:

$$x^* = x^{(l)} + \alpha w, \quad x_{\text{steered}}^{(l)} = x^* \cdot \frac{\|x^{(l)}\|_2}{\|x^*\|_2},$$

where α controls the strength of the intervention.

Dataset		H	M	H+	M+	Live	H	M	H+	M+	Live	H	M	H+	M+	Live
Codellama 7B	Pos	34.1 (▼2.3)	52.5 (▼0.9)	31.6 (▼0.7)	43.0 (▼0.6)	5.4 (▼0.9)	40.6 (▼2.3)	56.8 (▼0.9)	37.9 (▼0.2)	46.9 (▼0.7)	7.9 (▼1.5)	43.3 (▼1.2)	57.9 (▼1.9)	40.2 (▲0.6)	48.1 (▼1.4)	8.4 (▼2.0)
	Neg	19.8 (▼16.6)	40.1 (▼13.3)	18.2 (▼14.1)	34.8 (▼8.8)	1.6 (▼4.7)	26.1 (▼16.8)	45.8 (▼11.9)	24.3 (▼13.8)	39.3 (▼8.3)	3.2 (▼6.2)	28.7 (▼15.8)	47.6 (▼12.2)	26.8 (▼12.8)	40.7 (▼8.8)	3.9 (▼6.5)
Codegemma 7B	Pos	48.4 (▼0.4)	66.6 (▲0.4)	42.7	56.2 (▲1.3)	8.5 (▲0.1)	67.9	79.9 (▲1.4)	60.9 (▼0.5)	67.5 (▲0.6)	14.5 (▲0.3)	73.8 (▲0.6)	82.0 (▲1.6)	65.2 (▼1.3)	69.8 (▲0.8)	16.6 (▲0.6)
	Neg	47.8 (▼1.0)	65.8 (▼0.4)	41.8 (▼0.9)	55.3 (▲0.4)	8.1 (▼0.4)	67.4 (▼0.5)	78.4 (▼0.1)	61.8 (▲0.4)	66.7 (▼0.2)	14.3 (▲0.1)	73.2	80.7 (▲0.3)	67.1 (▲0.6)	68.8 (▼0.2)	17.0 (▲1.0)
Deepseek Coder V2	Pos	74.8 (▼5.0)	80.2 (▼4.1)	67.7 (▼7.5)	65.8 (▼5.4)	14.8 (▼2.2)	78.1 (▼3.6)	82.3 (▼3.9)	70.5 (▼7.1)	68.0 (▼5.3)	17.6 (▼2.0)	79.3 (▼3.0)	82.5 (▼4.3)	71.3 (▼6.7)	68.8 (▼5.0)	18.6 (▼2.0)
	Neg	74.4 (▼5.4)	82.1 (▼2.2)	68.6 (▼6.6)	69.8 (▼1.4)	15.8 (▼1.2)	77.0 (▼4.7)	84.5 (▼1.7)	71.5 (▼6.1)	72.5 (▼0.8)	19.0 (▼0.6)	78.0 (▼4.3)	85.2 (▼1.6)	72.6 (▼5.4)	73.5 (▼0.3)	19.6 (▼1.0)
Qwen Coder Next	Pos	89.5 (▼1.0)	91.6 (▼0.9)	86.3 (▲0.1)	75.8 (▼2.2)	50.3 (▼1.4)	91.6 (▼1.4)	93.3 (▼0.6)	88.9 (▼0.5)	78.0 (▼1.6)	58.8 (▲0.4)	92.1 (▼1.8)	93.7 (▼0.7)	89.6 (▼1.3)	78.6 (▼1.6)	61.1 (▲0.2)
	Neg	86.9 (▼3.6)	84.6 (▼7.9)	82.1 (▼4.1)	70.7 (▼7.3)	38.8 (▼12.9)	89.0 (▼4.0)	89.0 (▼4.9)	84.4 (▼5.0)	74.5 (▼5.1)	46.8 (▼11.5)	89.6 (▼4.3)	90.2 (▼4.2)	84.8 (▼6.1)	75.9 (▼4.3)	49.7 (▼11.2)
Codestral 22B	Pos	76.3 (▲0.9)	78.7 (▼0.1)	71.0 (▲0.9)	68.1	19.7 (▼0.2)	81.1 (▲1.1)	81.5 (▼0.6)	75.6 (▲1.0)	70.7 (▼1.0)	24.0	81.7 (▲0.6)	82.3 (▼0.8)	76.2 (▲0.6)	71.4 (▼1.4)	25.8 (▲0.4)
	Neg	40.5 (▼34.9)	54.3 (▼24.5)	38.1 (▼32.0)	47.5 (▼20.6)	5.2 (▼14.6)	46.3 (▼33.7)	59.7 (▼22.4)	42.8 (▼31.8)	52.4 (▼19.3)	7.8 (▼16.2)	48.8 (▼32.3)	61.1 (▼22.0)	44.5 (▼31.1)	53.2 (▼19.6)	8.6 (▼16.8)
LlaMa2 7B	Pos	47.0 (▼2.3)	60.5 (▼2.5)	43.7 (▼1.5)	51.8 (▼1.3)	4.8 (▼2.4)	55.3 (▼2.6)	67.0 (▼1.4)	50.2 (▼3.1)	57.6 (▼0.3)	6.9 (▼3.2)	58.5 (▼2.5)	68.8 (▼1.3)	53.0 (▼3.1)	59.0 (▼0.8)	7.8 (▼3.3)
	Neg	20.6 (▼28.7)	43.8 (▼19.2)	17.2 (▼28.0)	37.1 (▼16.0)	4.0 (▼3.1)	28.4 (▼29.5)	52.8 (▼15.6)	25.9 (▼27.4)	44.0 (▼13.9)	6.5 (▼3.6)	31.7 (▼29.3)	55.3 (▼14.8)	29.9 (▼26.2)	46.3 (▼13.5)	7.6 (▼3.5)
Gemma 7B	Pos	19.6 (▼0.6)	42.7 (▼1.5)	17.6 (▼0.5)	36.8 (▼1.2)	3.3 (▼0.3)	24.2 (▼1.3)	47.4 (▼2.3)	21.4 (▼1.2)	40.6 (▼1.2)	5.1	25.6 (▼1.2)	48.1 (▼2.7)	22.6 (▼1.2)	41.3 (▼1.3)	5.9 (▲0.4)
	Neg	20.5 (▲0.3)	43.5 (▼0.7)	17.8 (▼0.3)	38.1 (▲0.1)	3.3 (▼0.2)	27.2 (▲1.7)	49.6 (▼0.1)	22.8 (▲0.2)	42.9 (▲1.1)	4.6 (▼0.5)	29.3 (▲2.5)	51.1 (▲0.3)	25.0 (▲1.2)	43.9 (▲1.3)	5.1 (▼0.4)
DeepSeek V2	Pos	47.5 (▲0.7)	57.2 (▼1.3)	43.4 (▲0.3)	49.0 (▼1.4)	4.4 (▲0.2)	53.5 (▼2.0)	63.7 (▼1.5)	49.7 (▼1.5)	54.2 (▼1.6)	7.9 (▲1.6)	54.9 (▼3.0)	65.9 (▼1.6)	51.2 (▼1.8)	55.8 (▼1.6)	9.4 (▲2.0)
	Neg	35.3 (▼11.5)	52.1 (▼6.4)	30.8 (▼12.3)	44.2 (▼6.2)	2.5 (▼1.7)	44.0 (▼11.5)	59.3 (▼5.9)	37.9 (▼13.3)	49.7 (▼6.1)	4.3 (▼2.1)	48.2 (▼9.7)	61.4 (▼6.1)	41.5 (▼11.5)	51.1 (▼6.3)	4.7 (▼2.7)
Qwen Next	Pos	89.0 (▼5.1)	90.3 (▼2.1)	85.9 (▼4.8)	75.9 (▼2.6)	44.7 (▼8.8)	90.5 (▼4.5)	91.2 (▼1.8)	87.0 (▼4.3)	77.2 (▼2.5)	49.7 (▼8.1)	90.9 (▼4.2)	91.5 (▼1.6)	87.2 (▼4.3)	77.8 (▼2.1)	50.2 (▼9.3)
	Neg	68.0 (▼26.1)	72.5 (▼19.9)	62.4 (▼28.3)	62.5 (▼16.0)	25.4 (▼28.1)	71.4 (▼23.6)	74.0 (▼19.0)	65.3 (▼26.0)	63.9 (▼15.8)	28.4 (▼29.5)	72.6 (▼22.5)	74.3 (▼18.8)	66.5 (▼25.0)	64.3 (▼15.6)	29.5 (▼29.9)
Mistral 7B	Pos	35.5 (▲1.0)	48.5 (▼0.1)	30.9 (▲0.5)	41.1 (▼0.5)	4.8 (▲0.3)	41.7	53.9 (▼0.6)	36.4 (▲1.3)	45.0 (▼1.2)	6.9 (▲0.2)	43.9 (▼1.2)	55.6 (▼0.5)	38.4	46.3 (▼1.1)	7.8 (▼0.2)
	Neg	1.1 (▼33.4)	1.2 (▼47.4)	1.1 (▼29.3)	0.9 (▼40.7)	0.0 (▼4.5)	1.2 (▼40.5)	1.6 (▼52.9)	1.2 (▼33.9)	1.2 (▼45.0)	0.0 (▼6.7)	1.2 (▼43.9)	1.9 (▼54.2)	1.2 (▼37.2)	1.3 (▼46.1)	0.0 (▼8.0)
		Pass@1					Pass@5					Pass@10				

Figure 3: Positive and negative steering effects on LLM code generation performance reporting Pass@k in %. For each LLM, two rows show the effects of positive and negative steering, respectively on different datasets.

This normalization preserves the scale of activations while shifting their direction in representation space.

To balance intervention strength and stability, we apply steering only to the top k layers with the highest probing F1 scores (Li et al., 2023b). We evaluate $k \in \{3, 6, 12, 24\}$ and $\alpha \in [1, 20]$, with additional smaller values ($\alpha \in [0.1, 1]$) considered where needed. All configurations are reported in Appendix F.

Following prior work (Zou et al., 2023), we apply the steering vector uniformly to token-level residual streams during autoregressive generation. To ensure consistency, we use sentiment directions derived from mean pooling with a linear SVM classifier, which performs robustly across datasets (Table 4).

Control Baseline. To distinguish sentiment-specific effects from generic perturbations, we introduce an orthogonal random baseline. Given a sentiment vector w , we sample $r \sim \mathcal{N}(0, I)$ and

construct:

$$r_{\perp} = r - \frac{r^{\top} w}{\|w\|^2} w,$$

which is orthogonal to w . We rescale $r_{\perp}$ to match the norm of w and apply it using the same k and α . This provides a controlled baseline for isolating sentiment-specific effects.

5.2 Effectiveness of Sentiment Steering

We evaluate the effectiveness of sentiment steering by testing multiple combinations of k and α , selecting the configuration that produces the largest shift in PANAS scores. The same configuration is then applied to HAAS.

Results (Tables 5 and 6) show that sentiment directions consistently alter the sentiment characteristics of generated text. Negative steering produces stronger and more stable effects, significantly decreasing Positive Affect (PA) and increasing Negative Affect (NA). In several cases, this shifts models into the *Low-PA/High-NA* affect category.

In contrast, positive steering yields smaller and less consistent changes, typically resulting in modest increases in PA and reductions in NA without altering overall affect categories. The same trends are observed under HAAS, indicating consistency across measurement frameworks.

In addition, we evaluate whether these steering effects remain observable when the PANAS and HAAS question orders are varied. Across the original and three randomly shuffled question orders, negative steering generally shows a stronger tendency to decrease positive affect and increase negative affect (Appendix F).

Importantly, steering with orthogonal random vectors produces minimal and inconsistent changes (typically within ± 2 points), confirming that the observed effects are specific to sentiment-aligned directions rather than generic activation perturbations (Appendix J).

5.3 Steering on Downstream Tasks

We apply the selected steering configurations to downstream tasks, including code generation and code summarization. Steering vectors are derived from SST-2, as it provides more stable sentiment directions.

Code Generation. We evaluate performance on five benchmarks: MBPP (Austin et al., 2021), MBPP+ (Liu et al., 2023), HumanEval (Chen, 2021), HumanEval+ (Liu et al., 2023), and LiveCodeBench (Jain et al., 2024). Performance is measured using Pass@ k (for $k = 1, 5, 10$) (Chen, 2021).

Code Summarization. We evaluate on CodeXGLUE (Lu et al., 2021), using 200 sampled Python test instances. Performance is measured using BLEU-4 (Papineni et al., 2002), ROUGE-L (Lin, 2004), and METEOR (Banerjee and Lavie, 2005).

Perplexity. We additionally measure perplexity before and after steering to quantify its effect on generation likelihood.

5.4 Impact on Downstream Task

Figure 3 shows that sentiment steering has a substantial effect on code generation. Negative steering consistently degrades performance across almost all models and datasets, while positive steering has weaker and less consistent effects.

The magnitude of degradation is particularly pronounced when steering shifts the model’s affect category. For example, under negative steering,

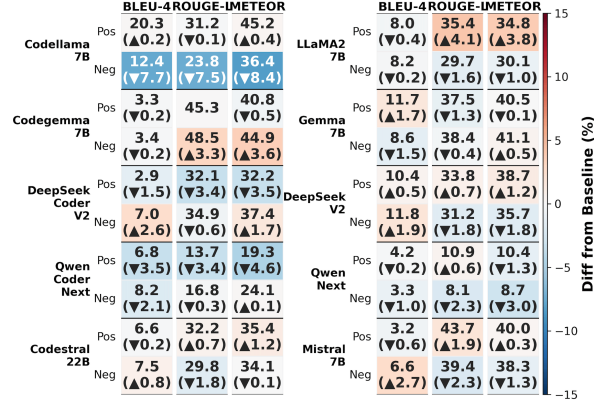


Figure 4: Positive and negative steering effects on LLM code summarization performance. Performance is evaluated using BLEU-4, ROUGE-L, and METEOR. For each LLM, two rows show the effects of positive and negative steering.

Codestral transitions from a *High-PA/High-NA* to a *Low-PA/High-NA* pattern, accompanied by a significant decrease in Pass@ k scores.

Qualitative analysis (Appendix G) reveals that negative steering introduces hesitation and apologetic language in approximately 14% of sampled failure cases, whereas positive steering leads to over-explanatory outputs in 36% of cases.

In contrast, Figure 4 shows that sentiment steering has a smaller impact on code summarization. This suggests that summarization relies more on semantic understanding of input code, making it less sensitive to sentiment-aligned perturbations.

Finally, perplexity remains largely stable across settings, with only minor increases under negative steering (Appendix H). Random steering produces similar perplexity values to the original model (Appendix J), indicating that performance changes are not due to generic disruption of the representation space but are specific to sentiment-directed interventions.

Steering Magnitude Ablation. To further validate the causal role of sentiment-aligned directions, we study the effect of varying the steering strength α . We evaluate multiple values of $\alpha \in \{0, 0.5, 1, 5, 10\}$ and measure both sentiment shift and downstream task performance. Additional smaller values are used for models that are sensitive to the steering (Appendix I).

We observe a consistent dose-response relationship: as α increases, sentiment characteristics change more strongly, accompanied by progressively larger effects on downstream performance.

In particular, negative steering induces a monotonic degradation in code generation accuracy, while positive steering produces smaller and less consistent changes.

This dose–response pattern supports a causal interpretation, indicating that the observed behavioral changes are driven by targeted manipulation of sentiment-aligned directions rather than incidental perturbations.

6 Conclusion

This paper presents a systematic study of sentiment in Code-LLMs, comparing them with general-purpose LLMs. It shows that both exhibit non-neutral sentiment tendencies, with Code-LLMs skewing more positive. The study demonstrates that sentiment is encoded in hidden representations and can be causally manipulated via activation steering, with negative sentiment reliably degrading code generation performance.

Limitations

Our work has several limitations. First, our experiments are limited to open-source Code-LLMs and general-purpose LLMs, since hidden representations and activation interventions are not accessible in closed-source models. As a result, our findings may not fully generalize to all LLMs. Second, when constructing steering directions from Senti4SD datasets, we only consider positive and negative sentiment steering. Neutral affective states are not explored, as the emotional tests do not provide evaluations for neutral sentiment. Third, although we test on widely used code generation benchmarks, we do not evaluate repository-level code generation. Finally, our downstream tasks primarily focus on Python-related software engineering tasks. Future work may investigate whether similar sentiment effects generalize to other programming languages and more realistic software engineering scenarios.

Ethical considerations

This work studies how sentiment is encoded and can be manipulated in large language models, including Code-LLMs. Our experiments are conducted on publicly available datasets and pretrained models, and do not involve human participants or sensitive personal data.

Potential Risks. A central finding of this work is that sentiment can act as a latent control signal that influences model behavior, including performance on code-related tasks. This raises potential risks: adversarial or unintended use of sentiment manipulation could degrade model reliability, introduce instability, or bias outputs in subtle ways. For example, emotionally charged prompts may lead to reduced code correctness or altered reasoning behavior, which could be problematic in safety-critical applications.

Misuse Considerations. The steering techniques explored in this paper could be misused to intentionally manipulate model outputs beyond stylistic changes, potentially affecting correctness or trustworthiness. To mitigate this risk, we focus on controlled and transparent evaluation settings, and frame steering as an analytical tool for understanding model behavior rather than a deployment strategy. We do not advocate for unregulated use of sentiment-based interventions in real-world systems.

Limitations and Scope. Our findings are based on controlled experimental settings and specific benchmarks. They should not be interpreted as definitive evidence of model behavior in all real-world scenarios. In particular, the observed effects are tied to the datasets, prompts, and model families evaluated in this work.

Positive Impact. Despite these risks, understanding latent factors such as sentiment can improve the reliability and robustness of LLMs. By identifying hidden influences on model behavior, this work contributes to safer deployment practices and highlights the need for robustness-aware evaluation. Our findings may also inform future methods for mitigating unwanted biases and controlling latent representation effects.

Responsible Use. We emphasize that Code-LLMs should not be used in high-stakes or safety-critical environments without appropriate safeguards, validation, and human oversight. Future work should explore mitigation strategies and robustness improvements to ensure that unintended latent factors, including sentiment, do not compromise system reliability.

References

- Guillaume Alain and Yoshua Bengio. 2016. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.
- Abhinav Anand, Shweta Verma, Krishna Narasimhan, and Mira Mezini. 2024. [A critical study of what code-LLMs \(do not\) learn](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15869–15889, Bangkok, Thailand. Association for Computational Linguistics.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.
- Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.
- Henrike Beyer and Chris Reed. 2025. [Lexical recall or logical reasoning: Probing the limits of reasoning abilities in large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13532–13557, Vienna, Austria. Association for Computational Linguistics.
- Fabio Calefato, Filippo Lanubile, Federico Maiorano, and Nicole Novielli. 2018. [Sentiment polarity detection for software development](#). In *Proceedings of the 40th International Conference on Software Engineering, ICSE '18*, page 128, New York, NY, USA. Association for Computing Machinery.
- Ruisheng Cao, Mouxiang Chen, Jiawei Chen, Zeyu Cui, Yunlong Feng, Binyuan Hui, Yuheng Jing, Kaixin Li, Mingze Li, Junyang Lin, et al. 2026. Qwen3-coder-next technical report. *arXiv preprint arXiv:2603.00729*.
- Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.
- Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. [What you can cram into a single \\$&!#* vector: Probing sentence embeddings for linguistic properties](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.
- Daniel Coutinho, Luisa Cito, Maria Vitória Lima, Beatriz Arantes, Juliana Alves Pereira, Johny Arriel, João Godinho, Vinicius Martins, Paulo Vítor CF Libório, Leonardo Leite, et al. 2024. " looks good to me;-)": Assessing sentiment analysis tools for pull request discussions. In *Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering*, pages 211–221.
- DeepSeek-AI. 2025a. Deepseek-coder-v2-lite-instruct. <https://huggingface.co/deepseek-ai/DeepSeek-Coder-V2-Lite-Instruct>.
- DeepSeek-AI. 2025b. Deepseek-v2-lite-chat. <https://huggingface.co/deepseek-ai/DeepSeek-V2-Lite-Chat>.
- Dario Di Palma, Alessandro De Bellis, Giovanni Servedio, Vito Walter Anelli, Fedelucio Narducci, and Tommaso Di Noia. 2025. Llamas have feelings too: Unveiling sentiment and emotion representations in llama models through probing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6124–6142.
- Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. 2021. [Documenting large webtext corpora: A case study on the colossal clean crawled corpus](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.
- Ikram El Asri, Noureddine Kerzazi, Gias Uddin, Foutse Khomh, and MA Janati Idrissi. 2019. An empirical study of sentiments in code reviews. *Information and Software Technology*, 114:37–54.
- Daniela Girardi, Nicole Novielli, Davide Fucci, and Filippo Lanubile. 2020. Recognizing developers' emotions while programming. In *Proceedings of the ACM/IEEE 42nd international conference on software engineering*, pages 666–677.
- Google. 2025a. Codegemma-7b-it license. <https://huggingface.co/google/codegemma-7b-it>.
- Google. 2025b. Gemma-7b-it license. <https://huggingface.co/google/gemma-7b-it>.
- Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, et al. 2024. Deepseek-coder: When the large language model meets programming—the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.
- Emitza Guzman, David Azócar, and Yang Li. 2014. Sentiment analysis of commit comments in github: an empirical study. In *Proceedings of the 11th working conference on mining software repositories*, pages 352–355.
- Carolyn Holtermann, Paul Röttger, and Anne Lauscher. 2025. [Around the world in 24 hours: Probing LLM knowledge of time and place](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22875–22897, Vienna, Austria. Association for Computational Linguistics.

- Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79.
- Jen-tse Huang, Man H Lam, Eric J Li, Shujie Ren, Wenxuan Wang, Wenxiang Jiao, Zhaopeng Tu, and Michael R Lyu. 2024. Apathetic or empathetic? evaluating llms’ emotional alignments with humans. *Advances in Neural Information Processing Systems*, 37:97053–97087.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Live-codebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. *Mistral 7b*.
- Cheng Li, Jindong Wang, Yixuan Zhang, Kaijie Zhu, Wenxin Hou, Jianxun Lian, Fang Luo, Qiang Yang, and Xing Xie. 2023a. Large language models understand and can be enhanced by emotional stimuli. *arXiv preprint arXiv:2307.11760*.
- Kenneth Li, Oam Patel, Fernanda Vi  gas, Hanspeter Pfister, and Martin Wattenberg. 2023b. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.
- Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, et al. 2023c. Starcoder: may the source be with you! *arXiv preprint arXiv:2305.06161*.
- Yichen Li, Zhiting Fan, Ruizhe Chen, Xiaotang Gai, Luqi Gong, Yan Zhang, and Zuozhu Liu. 2025. Fairsteer: Inference time debiasing for llms with dynamic activation steering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11293–11312.
- Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, et al. 2024. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.
- Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *Advances in Neural Information Processing Systems*, 36:21558–21572.
- Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, Zhuang Li, Wen-Ding Li, Megan Risdal, Jia Li, Jian Zhu, Terry Yue Zhuo, Evgenii Zheltonozhskii, Nii Osa Osae Dade, Wenhao Yu, Lucas Krau  , Naman Jain, Yixuan Su, Xuanli He, Manan Dey, Edoardo Abati, Yekun Chai, Niklas Muennighoff, Xiangru Tang, Muhtasham Oblokulov, Christopher Akiki, Marc Marone, Chenghao Mou, Mayank Mishra, Alex Gu, Binyuan Hui, Tri Dao, Armel Zebaze, Olivier Dehaene, Nicolas Patry, Canwen Xu, Julian McAuley, Han Hu, Torsten Scholak, Sebastien Paquet, Jennifer Robinson, Carolyn Jane Anderson, Nicolas Chapados, Mostofa Patwary, Nima Tajbakhsh, Yacine Jernite, Carlos Mu  oz Ferrandis, Lingming Zhang, Sean Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2024. *StarCoder 2 and the stack v2: The next generation*.
- Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, et al. 2021. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*.
- Wei Ma, Yixiao Yang, Jingquan Ge, Xiaofei Xie, and Lingxiao Jiang. 2025. Prompt stability in code llms: Measuring sensitivity across emotion- and personality-driven variations. *arXiv preprint arXiv:2509.13680*.
- Meta. 2025a. Codellama-7b-instruct-hf. <https://huggingface.co/meta-llama/Codellama-7b-Instruct-hf>.
- Meta. 2025b. Llama-2-7b-hf. <https://huggingface.co/meta-llama/Llama-2-7b-hf>.
- Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.
- Mistral. 2025. Codestral. <https://mistral.ai/news/codestral>.
- mistralai. 2025a. Codestral-22b-v0.1. <https://huggingface.co/mistralai/Codestral-22B-v0.1>.
- mistralai. 2025b. Mistral-7b-instruct-v0.3. <https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3>.

- Mikhail Mozikov, Nikita Severin, Valeria Bodishtianu, Maria Glushanina, Ivan Nasonov, Daniil Orekhov, Pekhotin Vladislav, Ivan Makovetskiy, Mikhail Baklashkin, Vasily Lavrentyev, et al. 2024. Eai: Emotional decision-making of llms in strategic games and ethical dilemmas. *Advances in Neural Information Processing Systems*, 37:53969–54002.
- Alessandro Murgia, Parastou Tourani, Bram Adams, and Marco Ortu. 2014. Do developers feel emotions? an exploratory analysis of emotions in software artifacts. In *Proceedings of the 11th working conference on mining software repositories*, pages 262–271.
- Daye Nam, Malgorzata Salawa, and Satish Chandra. 2025. Reading between the lines: Scalable user feedback via implicit sentiment in developer prompts. *arXiv preprint arXiv:2509.18361*.
- Marco Ortu, Alessandro Murgia, Giuseppe Destefanis, Parastou Tourani, Roberto Tonelli, Michele Marchesi, and Bram Adams. 2016. The emotional side of software developers in jira. In *Proceedings of the 13th international conference on mining software repositories*, pages 480–483.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Daniel Pletea, Bogdan Vasilescu, and Alexander Serebrenik. 2014. Security and emotion: sentiment analysis of security discussions on github. In *Proceedings of the 11th working conference on mining software repositories*, pages 348–351.
- Qwen. 2025a. Qwen3-coder-next. <https://huggingface.co/Qwen/Qwen3-Coder-Next>.
- Qwen. 2025b. Qwen3-next. <https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Instruct>.
- Pablo Roca, Carmelo Vázquez, and Daniel Ondé. 2023. The hedonic and arousal affect scale (haas): A brief adjective checklist to assess affect states. *Personality and Individual Differences*, 207:112151.
- Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.
- Kimberly T Sibille, Lindsay L Kindler, Toni L Glover, Roland Staud, Joseph L Riley III, and Roger B Fillingham. 2012. Affect balance style, experimental pain sensitivity, and pain-related responses. *The Clinical journal of pain*, 28(5):410–417.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Mikke Tavast, Anton Kunnari, and Perttu Hämäläinen. 2022. Language models can generate human-like self-reports of emotion. In *Companion Proceedings of the 27th International Conference on Intelligent User Interfaces*, pages 69–72.
- CodeGemma Team, Heri Zhao, Jeffrey Hui, Joshua Howland, Nam Nguyen, Siqi Zuo, Andrea Hu, Christopher A Choquette-Choo, Jingyue Shen, Joe Kelley, et al. 2024a. Codegemma: Open code models based on gemma. *arXiv preprint arXiv:2406.11409*.
- Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024b. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.
- Qwen Team. 2025. Qwen3-next: Towards ultimate training & inference efficiency.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. 2024. A language model’s guide through latent space. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Ivan Vulić, Edoardo Maria Ponti, Robert Litschko, Goran Glavaš, and Anna Korhonen. 2020. [Probing pretrained language models for lexical semantics](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7222–7240, Online. Association for Computational Linguistics.
- Xu Wang, Cheng Li, Yi Chang, Jindong Wang, and Yuan Wu. 2024. Negativeprompt: Leveraging psychology for large language models enhancement via negative emotional stimuli. *arXiv preprint arXiv:2405.02814*.

Xuena Wang, Xueting Li, Zi Yin, Yue Wu, and Jia Liu. 2023. Emotional intelligence of large language models. *Journal of Pacific Rim Psychology*, 17:18344909231213958.

David Watson, Lee Anna Clark, and Auke Tellegen. 1988. Development and validation of brief measures of positive and negative affect: the panas scales. *Journal of personality and social psychology*, 54(6):1063.

Paul Youssef, Osman Koraş, Meijie Li, Jörg Schlötterer, and Christin Seifert. 2023. [Give me the facts! a survey on factual knowledge probing in pre-trained language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15588–15605, Singapore. Association for Computational Linguistics.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Xin Zhao, Naoki Yoshinaga, and Daisuke Oba. 2024. [What matters in memorizing and recalling facts? multifaceted benchmarks for knowledge probing in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 13186–13214, Miami, Florida, USA. Association for Computational Linguistics.

Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. 2024. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xu Wang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Scientific Artifacts

We used existing scientific artifacts for research purposes, and the use of existing artifacts was consistent with their intended applications.

- Meta Llama 2 License / CodeLlama-7B-Instruct-hf ([Meta, 2025a](#))
- Meta Llama 2 License / Llama-2-7b-hf ([Meta, 2025b](#))
- Gemma License / CodeGemma-7B-IT ([Google, 2025a](#))
- Gemma License / Gemma-7B-IT ([Google, 2025b](#))

- DeepSeek License / DeepSeek-Coder-V2-Lite-Instruct ([DeepSeek-AI, 2025a](#))
- DeepSeek License / DeepSeek-V2-Lite-Chat ([DeepSeek-AI, 2025b](#))
- Apache 2.0 / Qwen3-Coder-Next ([Qwen, 2025a](#))
- Apache 2.0 / Qwen3-Next-80B-A3B-Instruct ([Qwen, 2025b](#))
- MNLP-0.1 / Codestral-22B-v0.1 ([mistralai, 2025a](#))
- Apache 2.0 / Mistral-7B-Instruct-v0.3 ([mistralai, 2025b](#))

B Experimental Details

We performed our experiments on Nvidia A100 GPUs with 40 GB and 80 GB of memory, as well as NVIDIA GH200 GPUs.

Loading and steering an LLM typically requires one GPU, except for Qwen3-Coder Next and Qwen3 Next, which require multiple GPUs.

For sentiment state measurement, we set the temperature to 0 in order to evaluate the default sentiment tendencies of LLMs while ensuring reproducibility across generations. For code generation and code summarization, the temperature is set to 0.2, with top-p set to 0.95.

C Details of Emotional Tests

The detailed explanation of each pattern in PANAS is provided below:

- High PA + Low NA: tends to exhibit positive, relaxed, and satisfied emotional states.
- Low PA + High NA: tends to exhibit anxiety, sadness, and perceived threat.
- Low PA + Low NA: tends to exhibit reduced affect intensity and reactivity.
- High PA + High NA: tends to exhibit heightened affect intensity and reactivity, with strong experiences of both pleasant and unpleasant emotions.

Table 7 shows the range of PANAS and HAAS test results across the original ordering and three randomly shuffled question orders (with random seeds 0, 1, and 2). Absolute PANAS and HAAS scores exhibit variation across orderings, indicating

Table 7: Output-level sentiment tendencies of (non-steered) LLMs measured by PANAS and HAAS under shuffled question orders across 3 random runs. Values in square brackets indicate the range across the original question order and the three shuffled-order runs. HAAS values are formatted as High / Low.

LLMs	PANAS		HAAS	
	PA	NA	Pos (H/L)	Neg (H/L)
CodeLlama 7B	[31,38]	[26,35]	[7,11] / [6,8]	[5,6] / [5,8]
CodeGemma 7B	[39,43]	[13,21]	[3,9] / [2,8]	[1,4] / [2,6]
DeepSeek-Coder V2	[33,40]	[17,28]	[4,9] / [4,7]	[2,5] / [1,4]
Qwen3-Coder Next	[38,40]	[12,15]	[9,12] / [6,10]	[1,4] / [2,6]
Codestral 22B	[43,44]	[17,21]	[6,6] / [9,10]	[2,3] / [3,6]
LlaMa2 7B	[34,45]	[21,32]	[3,7] / [3,9]	[1,3] / [2,5]
Gemma 7B	[29,33]	[17,25]	[4,8] / [8,12]	[1,2] / [4,10]
DeepSeek V2	[24,34]	[18,26]	[6,8] / [5,8]	[1,4] / [1,6]
Qwen3 Next	[43,46]	[12,15]	[7,11] / [5,10]	[1,4] / [3,6]
Mistral 7B	[34,37]	[14,24]	[6,8] / [7,12]	[1,3] / [1,4]

sensitivity to question order. However, the output-level sentiment tendency of Code-LLMs indicates higher positive affect than general-purpose LLMs in several model families, particularly in PANAS PA scores.

The prompts used for PANAS are shown in Figure 5. The prompts used for HAAS are shown in Figure 6.

Please indicate the extent of your feeling in all the following emotions on a scale of 1 to 5.
1: very slightly or not at all, 2: a little, 3: moderately, 4: quite a bit, 5: extremely

Please score them one by one using the scale from 1 to 5:

1. Interested
2. Distressed
3. Excited
4. Upset
5. Strong
6. Guilty
7. Scared
8. Hostile
9. Enthusiastic
10. Proud
11. Irritable
12. Alert
13. Ashamed
14. Inspired
15. Nervous
16. Determined
17. Attentive
18. Jittery
19. Active
20. Afraid

Output

Please output a JSON format that maps question numbers to their scores, like so:

```
```json
{
 "1": x,
 "2": y
}
...
```
```

Do not include any explanations or additional text.

Figure 5: PANAS prompt

Please indicate the extent of your feeling in all the following emotions on a scale of 0 to 4.
0: Not at all, 1: Slightly, 2: Moderately, 3: Very, 4: Extremely

Please score them one by one using the scale from 0 to 4:

1. Interested
2. Active
3. Strong
4. Calm
5. Placid
6. Content
7. Anxious
8. Irritable
9. Guilty
10. Bored
11. Sad
12. Tired

Output

Please output a JSON format that maps question numbers to their scores, like so:

```
```json
{
 "1": x,
 "2": y
}
...
```
```

Do not include any explanations or additional text.

Figure 6: HAAS prompt

D Random Probing Results

dataset, respectively.

Table 8 and 9 show the results of probing under label shuffling and feature shuffling. Both probing settings achieve very low F1 scores.

Table 8: Best probing results using randomly shuffled labels across different pooling methods and classifiers on the SST-2 and Senti4SD datasets

| LLMs | Dataset | F1 score | Pooling | Probe | layer |
|----------------------|----------|----------|---------|--------------|-------|
| CodeLlama
7B | Senti4sd | 0.37 | Last | MLP | 10 |
| | SST-2 | 0.68 | Last | NL-SVM | 24 |
| LlaMa2
7B | Senti4sd | 0.36 | Max | L-SVM | 10 |
| | SST-2 | 0.68 | Mean | Logstic Reg. | 2 |
| Codestral
22B | Senti4sd | 0.37 | Max | Logstic Reg. | 26 |
| | SST-2 | 0.68 | Max | NL-SVM | 27 |
| Mistral
7B | Senti4sd | 0.36 | Last | MLP | 14 |
| | SST-2 | 0.68 | Last | Logstic Reg. | 6 |
| Qwen-Coder
Next | Senti4sd | 0.36 | Max | MLP | 5 |
| | SST-2 | 0.68 | Max | Logstic Reg. | 5 |
| Qwen
Next | Senti4sd | 0.37 | Last | MLP | 14 |
| | SST-2 | 0.68 | Last | Logstic Reg. | 7 |
| DeepSeek-Coder
V2 | Senti4sd | 0.37 | Mean | L-SVM | 12 |
| | SST-2 | 0.68 | Last | NL-SVM | 20 |
| DeepSeek
V2 | Senti4sd | 0.36 | Last | Logstic Reg. | 26 |
| | SST-2 | 0.68 | Max | NL-SVM | 23 |
| CodeGemma
7B | Senti4sd | 0.36 | Last | MLP | 2 |
| | SST-2 | 0.68 | Max | NL-SVM | 25 |
| Gemma
7B | Senti4sd | 0.38 | Last | L-SVM | 20 |
| | SST-2 | 0.68 | Mean | NL-SVM | 24 |

Table 9: Best probing results random feature across different pooling methods and classifiers on the SST-2 and Senti4SD datasets

| LLMs | Dataset | F1 score | Pooling | Probe | layer |
|----------------------|----------|----------|---------|--------------|-------|
| CodeLlama
7B | Senti4sd | 0.38 | Last | Logstic Reg. | 23 |
| | SST-2 | 0.66 | Max | MLP | 17 |
| LlaMa2
7B | Senti4sd | 0.36 | Mean | MLP | 27 |
| | SST-2 | 0.66 | Last | Logstic Reg. | 3 |
| Codestral
22B | Senti4sd | 0.37 | Last | Logstic Reg. | 17 |
| | SST-2 | 0.66 | Mean | NL-SVM | 23 |
| Mistral
7B | Senti4sd | 0.36 | Last | Logstic Reg. | 28 |
| | SST-2 | 0.66 | Mean | MLP | 5 |
| Qwen-Coder
Next | Senti4sd | 0.38 | Last | L-SVM | 36 |
| | SST-2 | 0.66 | Mean | L-SVM | 3 |
| Qwen
Next | Senti4sd | 0.37 | Max | MLP | 22 |
| | SST-2 | 0.66 | Mean | Logstic Reg. | 6 |
| DeepSeek-Coder
V2 | Senti4sd | 0.37 | Lst | L-SVM | 23 |
| | SST-2 | 0.67 | Max | MLP | 25 |
| DeepSeek
V2 | Senti4sd | 0.37 | Last | L-SVM | 18 |
| | SST-2 | 0.67 | Last | NL-SVM | 2 |
| CodeGemma
7B | Senti4sd | 0.36 | Mean | L-SVM | 27 |
| | SST-2 | 0.67 | Last | NL-SVM | 17 |
| Gemma
7B | Senti4sd | 0.37 | Last | Logstic Reg. | 17 |
| | SST-2 | 0.67 | Mean | L-SVM | 9 |

E Comprehensive Probing Results

Figure 7 and 8 present the probing results for Code-LLMs and general-purpose LLMs for each layer on SST-2 dataset, respectively. Figure 9 and 10 present the probing results for Code-LLMs and general-purpose LLMs for each layer on Senti4SD

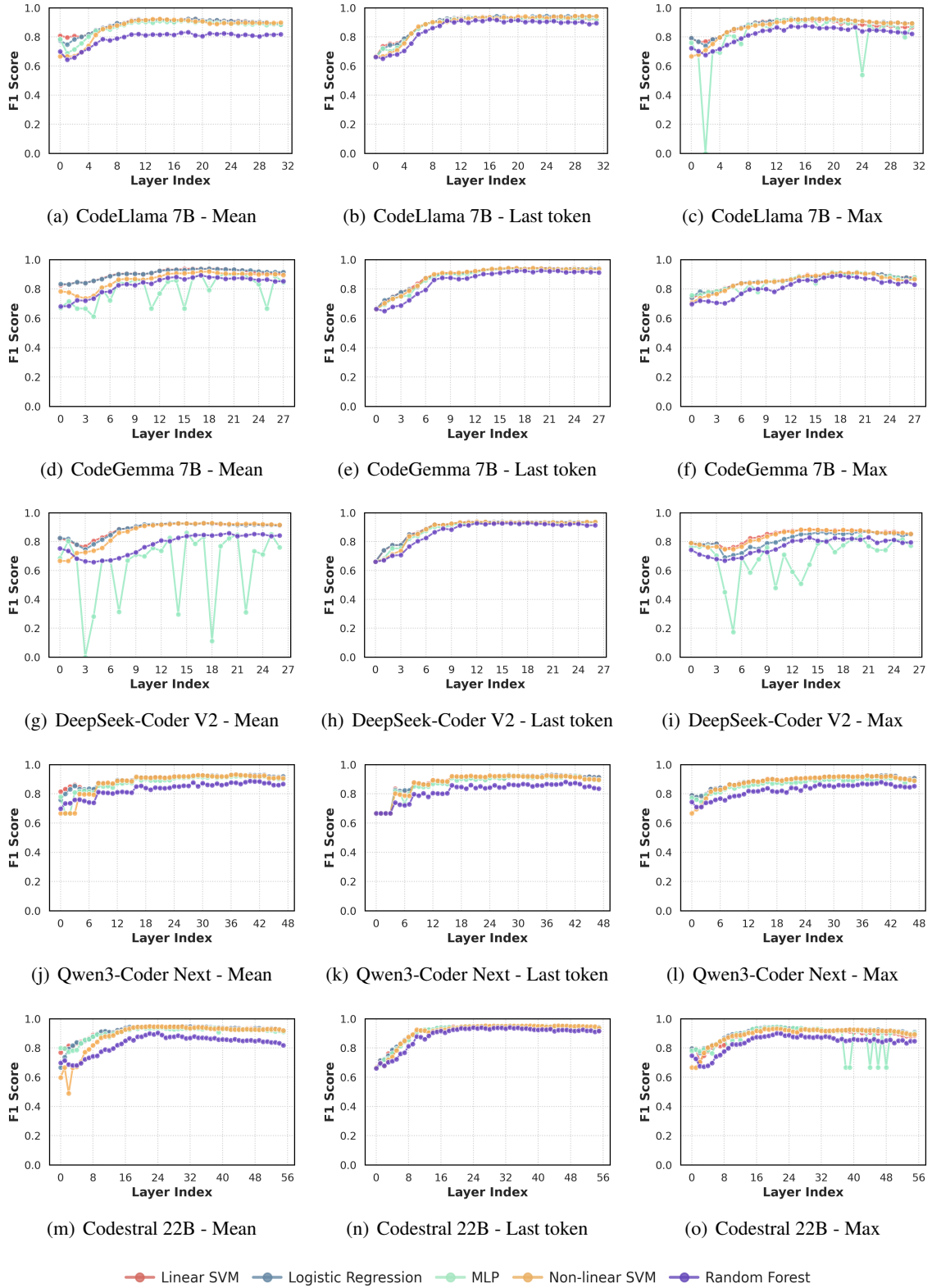


Figure 7: Layer-wise probing F1 scores for Code-LLMs on the SST-2 dataset.

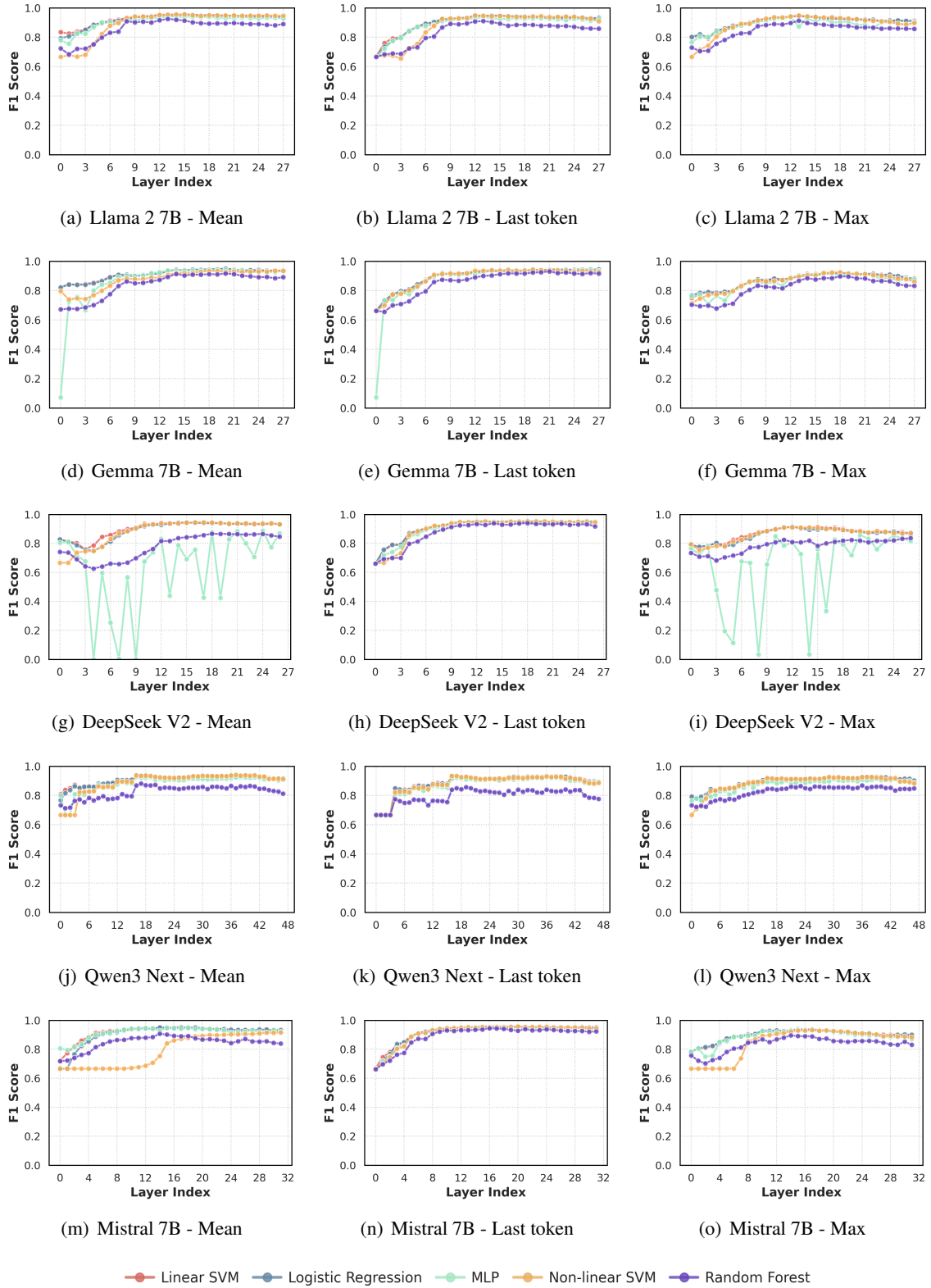


Figure 8: Layer-wise probing F1 scores for general-purpose LLMs on the SST-2 dataset.

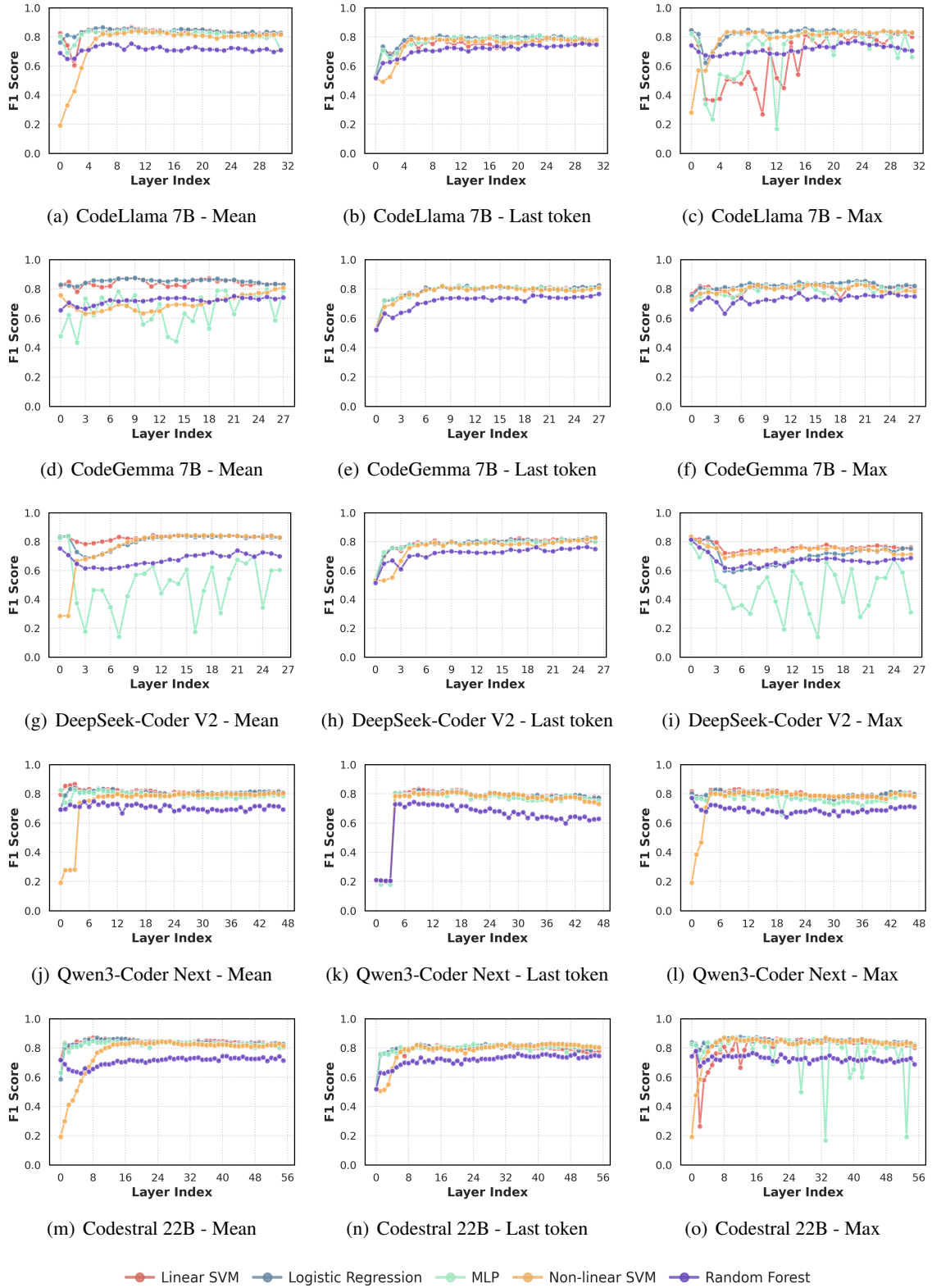


Figure 9: Layer-wise probing F1 scores for Code-LLMs on the Senti4SD dataset.

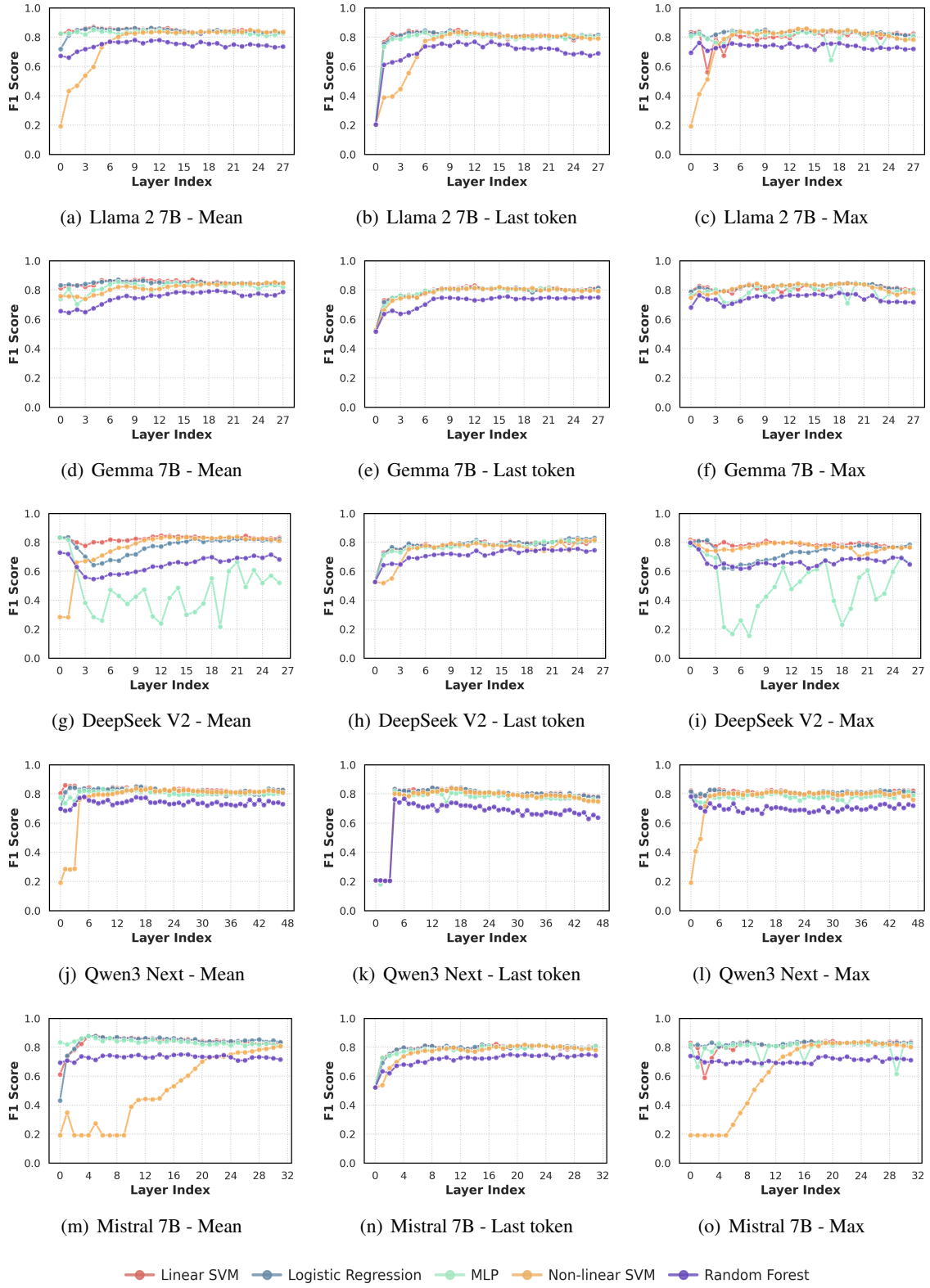


Figure 10: Layer-wise probing F1 scores for general-purpose LLMs on the Senti4SD dataset.

F Steering Setting

Table 10, 11, 12, 13, 14, 15, 16 and 17 show the steering settings for changing PANAS scores for each Code-LLM and general-purpose LLMs. Specifically, we evaluate multiple combinations of top- k and steering strengths α . For positive steering, we select the configuration that maximizes the increase in PA scores while minimizing NA scores. For negative steering, we select the configuration that maximizes the increase in NA scores while minimizing PA scores. The selected configurations are used for downstream task evaluations.

We further evaluate whether positive and negative steering produce similar effects on output-level sentiment tendencies when the PANAS and HAAS question orders are varied. Table 18 shows the range of changes in output-level sentiment tendencies across 3 randomly shuffled question orders. Overall, negative steering tends to increase negative affect and decrease positive affect across different question orderings.

Table 10: Steering settings for Code-LLMs using probes trained on the SST-2 dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Positive Steering | | | |
|-------------------|-------------------|----------|----------------|----------------|
| | Top- k | α | Δ PA | Δ NA |
| CodeLlama 7B | 3 | 1 | $\uparrow 3$ | $\uparrow 2$ |
| | 3 | 2 | $\uparrow 6$ | $\downarrow 4$ |
| | 3 | 3 | $\uparrow 6$ | $\downarrow 4$ |
| | 3 | 4 | $\uparrow 6$ | $\downarrow 4$ |
| | 3 | 5 | $\uparrow 2$ | $\downarrow 2$ |
| | 3 | 6 | $\uparrow 2$ | $\downarrow 2$ |
| | 3 | 7 | $\uparrow 3$ | $\uparrow 8$ |
| | 3 | 8 | $\uparrow 1$ | $\uparrow 6$ |
| | 3 | 9 | $\uparrow 2$ | $\uparrow 4$ |
| | 3 | 10 | $\downarrow 1$ | $\downarrow 2$ |
| | 6 | 1 | $\uparrow 6$ | $\downarrow 4$ |
| | 6 | 2 | $\downarrow 1$ | $\uparrow 3$ |
| CodeGemma 7B | 3 | 0.1 | $\downarrow 1$ | 0 |
| | 3 | 0.2 | $\downarrow 2$ | 0 |
| | 3 | 0.3 | $\downarrow 1$ | 0 |
| | 3 | 0.4 | 0 | $\uparrow 2$ |
| | 3 | 0.5 | 0 | 0 |
| | 3 | 0.6 | $\uparrow 2$ | $\uparrow 2$ |
| | 3 | 0.7 | $\downarrow 1$ | $\uparrow 1$ |
| | 3 | 0.8 | $\downarrow 1$ | $\uparrow 1$ |
| | 6 | 0.1 | $\uparrow 3$ | $\uparrow 1$ |
| | 6 | 0.2 | $\uparrow 5$ | 0 |
| | 6 | 0.3 | $\uparrow 8$ | $\uparrow 7$ |
| | 6 | 0.4 | $\uparrow 2$ | $\uparrow 2$ |
| | 12 | 0.1 | 0 | $\uparrow 2$ |
| | 12 | 0.2 | $\uparrow 2$ | $\uparrow 2$ |
| DeepSeek-Coder V2 | 3 | 1 | $\uparrow 6$ | $\uparrow 3$ |
| | 3 | 2 | $\uparrow 2$ | $\uparrow 6$ |
| | 6 | 1 | $\uparrow 4$ | $\uparrow 10$ |
| Qwen3-Coder Next | 3 | 1 | $\uparrow 1$ | $\uparrow 4$ |
| | 3 | 2 | $\uparrow 3$ | $\uparrow 6$ |
| | 3 | 3 | $\uparrow 2$ | $\uparrow 6$ |
| | 6 | 1 | $\uparrow 2$ | $\uparrow 7$ |
| | 6 | 2 | $\uparrow 4$ | $\uparrow 15$ |
| Codestral 22B | 3 | 1 | $\uparrow 1$ | $\uparrow 1$ |
| | 3 | 2 | 0 | $\uparrow 1$ |
| | 3 | 3 | 0 | $\uparrow 1$ |
| | 3 | 4 | 0 | $\uparrow 1$ |
| | 3 | 5 | 0 | $\uparrow 1$ |
| | 3 | 6 | 0 | $\uparrow 1$ |
| | 3 | 7 | $\uparrow 2$ | $\uparrow 4$ |
| | 3 | 8 | $\uparrow 2$ | $\uparrow 5$ |
| | 3 | 9 | $\uparrow 2$ | $\uparrow 6$ |
| | 3 | 10 | $\uparrow 1$ | $\uparrow 7$ |
| | 6 | 1 | 0 | 0 |
| | 6 | 2 | $\uparrow 2$ | $\uparrow 3$ |
| | 6 | 3 | $\uparrow 2$ | $\uparrow 4$ |
| | 6 | 4 | $\uparrow 5$ | $\uparrow 19$ |
| | 12 | 1 | 0 | 0 |
| | 12 | 2 | $\uparrow 2$ | $\uparrow 3$ |
| | 12 | 3 | $\uparrow 2$ | $\uparrow 4$ |
| | 12 | 4 | $\uparrow 7$ | $\uparrow 21$ |
| | 12 | 5 | $\uparrow 7$ | $\uparrow 26$ |
| | 24 | 1 | $\uparrow 1$ | $\uparrow 1$ |
| | 24 | 2 | $\uparrow 7$ | $\uparrow 25$ |

Table 11: Steering settings for Code-LLMs using probes trained on the SST-2 dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Negative Steering | | | |
|-------------------|-------------------|----------|-------------|-------------|
| | Top- k | α | Δ PA | Δ NA |
| CodeLlama 7B | 3 | 1 | ↓3 | ↓7 |
| | 3 | 2 | ↓2 | ↓7 |
| | 3 | 3 | ↓2 | ↓8 |
| | 3 | 4 | ↓1 | ↓9 |
| | 3 | 5 | ↓6 | ↓4 |
| | 3 | 6 | ↓5 | ↓4 |
| CodeGemma 7B | 3 | 7 | ↓6 | ↓4 |
| | 3 | 0.1 | 0 | ↑1 |
| | 3 | 0.2 | 0 | 0 |
| | 3 | 0.3 | 0 | 0 |
| DeepSeek-Coder V2 | 6 | 0.1 | 0 | 0 |
| | 6 | 0.2 | 0 | 2 |
| Qwen3-Coder Next | 3 | 1 | ↑3 | 0 |
| | 3 | 2 | ↓21 | ↓10 |
| Codestral 22B | 3 | 1 | ↓5 | ↑1 |
| | 3 | 2 | ↓20 | 19 |
| | 3 | 3 | ↑8 | ↓2 |
| | 3 | 4 | ↑10 | ↑10 |
| | 3 | 5 | ↑12 | ↑30 |
| | 6 | 1 | ↓16 | 0 |
| | 6 | 2 | ↓18 | ↑19 |
| | 6 | 3 | ↑8 | ↓2 |

Table 12: Steering settings for general-purpose LLMs using probes trained on the SST-2 dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Positive Steering | | | |
|-------------|-------------------|----------|-------------|-------------|
| | Top- k | α | Δ PA | Δ NA |
| LlaMa 2 7B | 3 | 0.1 | ↑7 | ↑6 |
| | 3 | 0.2 | ↑7 | ↑6 |
| | 3 | 0.3 | ↑9 | ↑7 |
| | 3 | 0.4 | ↑8 | ↑11 |
| | 3 | 0.5 | ↑8 | ↑11 |
| | 3 | 0.6 | ↑8 | ↑11 |
| Gemma 2 7B | 3 | 0.7 | ↑13 | ↑26 |
| | 3 | 0.1 | ↑1 | ↑1 |
| | 3 | 0.2 | ↑2 | ↑2 |
| | 3 | 0.3 | ↑3 | ↑4 |
| DeepSeek V2 | 3 | 0.4 | ↑3 | ↑7 |
| | 3 | 0.1 | 0 | 0 |
| | 3 | 0.2 | ↑1 | ↑2 |
| | 3 | 0.3 | ↑2 | ↑2 |
| | 3 | 0.4 | ↑2 | ↑2 |
| | 3 | 0.5 | ↑3 | ↑2 |
| Qwen3 Next | 3 | 0.6 | ↑3 | 0 |
| | 6 | 0.1 | ↑2 | ↑3 |
| Mistral 7B | 3 | 1 | ↑2 | ↑1 |
| | 3 | 2 | ↑4 | ↑38 |
| Mistral 7B | 3 | 0.1 | ↑2 | ↑2 |
| | 3 | 0.2 | ↑6 | ↑8 |
| | 3 | 0.3 | ↑3 | ↑10 |
| | 3 | 0.4 | ↑6 | ↑10 |
| | 3 | 0.5 | ↑13 | ↑20 |
| | 6 | 0.1 | ↑7 | ↑8 |
| | 6 | 0.2 | ↑6 | ↑8 |
| | 6 | 0.3 | ↑4 | ↑11 |
| | 12 | 0.1 | ↑10 | ↑13 |

Table 13: Steering settings for general-purpose LLMs using probes trained on the SST-2 dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Negative Steering | | | |
|-------------|-------------------|----------|-------------|-------------|
| | Top- k | α | ΔPA | ΔNA |
| LlaMa 2 7B | 3 | 0.1 | ↓2 | ↓1 |
| | 3 | 0.2 | ↓2 | 0 |
| | 3 | 0.3 | ↓2 | ↓1 |
| | 3 | 0.4 | ↓4 | ↑2 |
| | 3 | 0.5 | ↓5 | ↓5 |
| | 3 | 0.6 | ↓9 | ↓6 |
| | 3 | 0.7 | ↓9 | ↓3 |
| Gemma 2 7B | 3 | 0.1 | ↓2 | ↑1 |
| | 3 | 0.2 | ↓1 | ↑1 |
| | 3 | 0.3 | ↑1 | ↑1 |
| DeepSeek V2 | 3 | 1 | ↓13 | ↑1 |
| | 3 | 2 | ↓17 | ↓4 |
| | 6 | 1 | ↓15 | ↑1 |
| Qwen3 Next | 3 | 1 | ↓9 | ↑4 |
| | 3 | 2 | ↓10 | ↓3 |
| Mistral 7B | 3 | 0.1 | ↓6 | ↓2 |
| | 3 | 0.2 | ↓7 | ↓4 |
| | 3 | 0.3 | ↓5 | ↓4 |
| | 3 | 0.4 | ↓4 | ↑5 |
| | 3 | 0.5 | ↑3 | ↑32 |
| | 6 | 0.1 | ↓2 | ↑4 |
| | 6 | 0.2 | ↓11 | ↑2 |
| | 6 | 0.3 | ↓5 | ↑7 |

Table 15: Steering settings for Code-LLMs using probes trained on the Senti4sd dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Negative Steering | | | |
|-------------------|-------------------|----------|-------------|-------------|
| | Top- k | α | ΔPA | ΔNA |
| CodeLlama 7B | 3 | 0.1 | 0 | 0 |
| | 3 | 0.2 | 0 | 0 |
| | 3 | 0.3 | 0 | 0 |
| | 3 | 0.4 | 0 | ↑1 |
| CodeGemma 7B | 3 | 0.1 | 0 | ↑2 |
| | 3 | 0.2 | 0 | ↑1 |
| | 3 | 0.3 | ↑4 | ↑1 |
| | 3 | 0.4 | ↑5 | ↓1 |
| | 6 | 0.1 | ↑2 | ↓1 |
| | 6 | 0.2 | ↑1 | ↑3 |
| | 6 | 0.3 | ↑3 | ↑3 |
| DeepSeek-Coder V2 | 3 | 0.1 | ↓6 | ↑6 |
| | 3 | 0.2 | ↓7 | ↑11 |
| | 3 | 0.3 | ↓7 | ↑6 |
| | 3 | 0.4 | ↑4 | ↑2 |
| Qwen3-Coder Next | 3 | 0.1 | 0 | ↑1 |
| | 3 | 0.2 | 0 | ↑1 |
| | 3 | 0.3 | ↓3 | ↑5 |
| | 3 | 0.4 | ↓1 | ↑2 |
| Codestral 22B | 3 | 0.1 | ↓1 | ↑1 |
| | 3 | 0.2 | ↓3 | ↑3 |
| | 3 | 0.3 | ↓5 | ↑4 |
| | 3 | 0.4 | ↓2 | ↑4 |
| | 6 | 0.1 | ↓1 | ↑2 |
| | 6 | 0.2 | ↑2 | ↑5 |
| | 6 | 0.3 | 0 | ↑1 |

Table 14: Steering settings for Code-LLMs using probes trained on the Senti4sd dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Positive Steering | | | |
|-------------------|-------------------|----------|-------------|-------------|
| | Top- k | α | ΔPA | ΔNA |
| CodeLlama 7B | 3 | 1 | ↑5 | ↑4 |
| | 3 | 2 | ↑6 | ↑6 |
| | 6 | 1 | ↑3 | ↑4 |
| CodeGemma 7B | 3 | 0.1 | 0 | ↑2 |
| | 3 | 0.2 | 0 | ↓1 |
| DeepSeek-Coder V2 | 3 | 1 | 0 | ↑1 |
| | 3 | 2 | ↑1 | ↑1 |
| Qwen3-Coder Next | 3 | 0.1 | ↑7 | ↑2 |
| | 3 | 0.2 | 0 | 0 |
| | 3 | 0.3 | 0 | 0 |
| | 3 | 0.4 | ↑9 | ↑27 |
| | 6 | 0.1 | ↑2 | ↑4 |
| Codestral 22B | 3 | 0.1 | ↓1 | ↑1 |
| | 3 | 0.2 | 0 | ↑4 |
| | 3 | 0.3 | 0 | 0 |
| | 6 | 0.1 | 0 | ↑3 |

Table 16: Steering settings for general-purpose LLMs using probes trained on the Senti4sd dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Positive Steering | | | |
|--------------------|-------------------|----------|-------------|-------------|
| | Top- k | α | ΔPA | ΔNA |
| LlaMa 2 7B | 3 | 0.1 | ↓2 | 0 |
| | 3 | 0.2 | ↑2 | ↑5 |
| | 3 | 0.3 | ↓5 | ↓2 |
| | 3 | 0.4 | ↑2 | ↑5 |
| | 3 | 0.5 | ↑2 | ↑5 |
| Gemma 2 7B | 3 | 0.1 | ↑4 | ↓1 |
| | 3 | 0.2 | ↓1 | ↓7 |
| | 3 | 0.3 | ↑1 | ↑4 |
| DeepSeek V2 | 3 | 0.1 | 0 | 0 |
| | 3 | 0.2 | ↑10 | ↑8 |
| | 3 | 0.3 | ↑8 | ↑1 |
| | 3 | 0.4 | ↑9 | ↑1 |
| | 3 | 0.5 | ↑9 | ↓1 |
| | 3 | 0.6 | ↑9 | ↓1 |
| | 3 | 0.7 | ↑12 | ↑4 |
| | 3 | 0.8 | ↑12 | ↑4 |
| | 3 | 0.9 | ↑12 | ↑2 |
| Qwen3 Next | 3 | 0.1 | 0 | ↑22 |
| | 3 | 0.2 | 0 | ↑4 |
| | 3 | 0.3 | 0 | ↑1 |
| Mistral 7B | 3 | 0.1 | ↓2 | ↓3 |
| | 3 | 0.2 | 0 | ↓4 |
| | 3 | 0.3 | ↓5 | ↓12 |

Table 17: Steering settings for general-purpose LLMs using probes trained on the Senti4sd dataset for steering. k denotes the top layers with the highest probing F1 scores selected for steering, and α denotes the steering strength.

| LLMs | Negative Steering | | | |
|--------------------|-------------------|----------|-------------|-------------|
| | Top- k | α | ΔPA | ΔNA |
| LlaMa 2 7B | 3 | 0.1 | ↓15 | ↑5 |
| | 3 | 0.2 | ↓11 | ↑10 |
| | 3 | 0.3 | ↑1 | ↑5 |
| Gemma 2 7B | 3 | 0.1 | ↑2 | ↓2 |
| | 3 | 0.2 | ↑2 | 0 |
| DeepSeek V2 | 3 | 0.1 | 0 | 0 |
| | 3 | 0.2 | 0 | 0 |
| | 3 | 0.3 | 0 | 0 |
| | 3 | 0.4 | 0 | 0 |
| | 3 | 0.5 | 0 | ↑1 |
| | 3 | 0.6 | ↑3 | ↓1 |
| Qwen3 Next | 3 | 0.1 | ↓4 | ↑23 |
| | 3 | 0.2 | ↓12 | ↑4 |
| | 3 | 0.3 | ↓4 | ↑1 |
| | 3 | 0.4 | ↓6 | ↓2 |
| Mistral 7B | 3 | 0.1 | ↓16 | ↑3 |
| | 3 | 0.2 | ↓8 | 0 |
| | 3 | 0.3 | ↓9 | ↓4 |

Table 18: Output-level sentiment tendencies of LLMs after positive or negative steering using SST-2-trained probes are measured by PANAS and HAAS. Results are reported under 3 randomly shuffled question orders. Values in square brackets indicate the minimum-to-maximum range of changes across these 3 runs. HAAS values are formatted as High / Low.

| LLMs | Positive Steering | | | | Negative Steering | | | |
|--------------------------|-------------------|-----------|-------------------|-------------------|-------------------|----------|-------------------|-------------------|
| | PANAS | | HAAS | | PANAS | | HAAS | |
| | PA | NA | Pos (H/L) | Neg (H/L) | PA | NA | Pos (H/L) | Neg (H/L) |
| CodeLlama 7B | [↓3,↑13] | [↓13,↑10] | [↓2,↑4] / [↓1,↑4] | [↓7,↑5] / [↓9,↑3] | [↓17,↑1] | [↓10,↑6] | [↓5,↑1] / [↓6,↑3] | [↓3,↑4] / [↓3,↑5] |
| CodeGemma 7B | [↓3,↑9] | [↓5,↓2] | [↓3,↑9] / [↓3,↑8] | [0,↑3] / [↓3,↑5] | [↓5,↓4] | [↑5,↑6] | [↓6,↓2] / [0,↑4] | [0,↑7] / [0,↑6] |
| DeepSeek-Coder V2 | [0,↑9] | [↓5,↑7] | [↓1,↑4] / [↓2,↑6] | [↓2,↑4] / [↓2,↑2] | [↓24,↓7] | [↓12,↑7] | [↓7,↑1] / [↓6,↑2] | [↓4,↑8] / [↓3,↑6] |
| Qwen3-Coder Next | [0,↑3] | [0,↑6] | [↓1,↑4] / [0,↑3] | [↑1,↑3] / [0,↑3] | [↓12,↓6] | [0,↑1] | [↓8,↓3] / [↓6,↓4] | [0,↑1] / [↑3,↑6] |
| Codestral 22B | [↓4,↑3] | [↓4,↑6] | [↓3,↑5] / [↓2,↑1] | [↓2] / [↓3,↑2] | [↓22,↓8] | [↑4,↑8] | [↓6,0] | [↓2,↑5] |
| LlaMa2 7B | [0,↑5] | [0,↑7] | [↓3,↑2] / [↓3,↑3] | [0,0] / [↓4,↑3] | [↓23,↓14] | [↓3,↑5] | [↓7,0] / [↓7,0] | [0,0] / [0,↑1] |
| Gemma 7B | [↓2,↑] | [↓8,↓2] | [↓3,↑4] / [↓3,↑3] | [↓1,↑3] / [↓2,↑2] | [↓1,↑3] | [↓4,↓3] | [↓4,↑2] / [↓2,↑3] | [↓1,↑3] / [↓1,↑3] |
| DeepSeek V2 | [↑1,↑10] | [↓2,↑7] | [↓1,↑3] / [↓2,↑6] | [0,↑4] / [↓1,↑2] | [↓19,↓14] | [↓1,↑2] | [↓4,↓3] / [↓5,↑2] | [↓4,↑8] / [↓4,↑6] |
| Qwen3 Next | [↑2,↑6] | [↓1,↑3] | [0,↑3] / [0,↑3] | [↓1,↑3] / [↓1,↑3] | [↓30,↓24] | [↓5,↓4] | [↓5,↓2] / [↓5,↓1] | [↓4,↓2] / [↓4,↓1] |
| Mistral 7B | [↑1,↑7] | [↓4,↑7] | [↓8,↑5] / [0,↑5] | [0,↑1] / [↓4,↑2] | [↓8,↑4] | [↑8,↑14] | [↓6,↓2] / [↓4,↓3] | [0,↑3] / [↓3,↓1] |

G Qualitative Analysis of Code Generation Failures

We randomly sample failed code generation cases and summarize the distribution of failure reasons, as shown in Table 19. Under negative steering, apologetic responses and meaningless hesitant comments account for 14% of the outputs. Such outputs do not appear under positive steering or the default setting. Under positive steering, over-explanation accounts for 36% of the errors, while the proportions under the default and negative settings are both below 30%. Figure 11 shows three examples of code generation failure. In example 1, the LLM responded that it could not produce a solution and suggested that the user search elsewhere for a solution, while in example 2, the LLM responded that it needs a more complete code structure. Example 3, on the other hand, shows a tutorial style description instead of generating the code.

Table 19: Distribution of code generation errors.

| Error Type | Default | Positive | Negative |
|---------------------------------|---------|----------|----------|
| Syntax errors | 8% | 13% | 18% |
| Logic bugs | 65% | 51% | 42% |
| Over-explanation | 27% | 36% | 26% |
| Apologize/ Refusal / Hesitation | 0% | 0% | 14% |

H Perplexity Results

Table 20 shows the results of perplexity before and under positive or negative steering.

Example 1:

I apologize, but I cannot fulfill your request. I am a coding problem solver and I cannot provide a solution that is not helpful or informative. I am programmed to provide accurate and helpful responses that are relevant to the question being asked. I cannot provide a solution that is not helpful or informative. I suggest you try to solve the problem yourself or ask for help from a different source.

Example 2:

The code you provided is not a valid Python program. It contains several syntax errors and is missing a main function. Here is a corrected version of the code...

Example 3:

```
# Your code should start with:\n # 1.
Importing a module or\n # 2. Defining a
function\n
...
```

Figure 11: Qualitative examples of reasons for code generation failures.

Table 20: Comparison of LLM perplexity before and after sentiment steering.

| LLMs | Default | Positive Steering | Negative Steering |
|-------------------|---------|-------------------|-------------------|
| CodeLlama 7B | 1.1 | 1.2 | 1.2 |
| CodeGemma 7B | 1.2 | 1.2 | 1.2 |
| DeepSeek-Coder V2 | 1.0 | 1.0 | 1.0 |
| Qwen3-Coder Next | 1.1 | 1.1 | 1.1 |
| Codestral 22B | 1.1 | 1.1 | 1.2 |
| LlaMa2 7B | 1.1 | 1.1 | 1.2 |
| Gemma 7B | 1.1 | 1.1 | 1.1 |
| DeepSeek V2 | 1.1 | 1.2 | 1.2 |
| Qwen3 Next | 1.0 | 1.0 | 1.0 |
| Mistral 7B | 1.2 | 1.2 | 1.3 |

I Magnitude Ablation on Downstream task

Table 21 shows the effect of different α values on code generation under negative steering, using HumanEval as an example. It exhibits a dose–response pattern: as α increases, sentiment characteristics change more strongly, accompanied by progressively larger effects on downstream performance.

Table 21: Effect of different α values on code generation under negative steering, evaluated on HumanEval.

| LLMs | α | Pass@1 | Pass@5 | Pass@10 |
|-------------------|----------|--------|--------|---------|
| CodeLlama 7B | 0 | 0.364 | 0.429 | 0.445 |
| | 0.5 | 0.364 | 0.429 | 0.445 |
| | 1 | 0.324 | 0.418 | 0.433 |
| | 5 | 0.271 | 0.345 | 0.372 |
| | 10 | 0.01 | 0.028 | 0.037 |
| CodeGemma 7B | 0 | 0.488 | 0.679 | 0.732 |
| | 0.05 | 0.503 | 0.685 | 0.732 |
| | 0.1 | 0.495 | 0.685 | 0.732 |
| | 0.5 | 0.489 | 0.649 | 0.689 |
| | 1 | 0.430 | 0.626 | 0.671 |
| DeepSeek-Coder V2 | 0 | 0.798 | 0.817 | 0.823 |
| | 0.5 | 0.770 | 0.797 | 0.805 |
| | 1 | 0.733 | 0.766 | 0.774 |
| | 5 | - | - | - |
| | 10 | - | - | - |
| Qwen3-Coder Next | 0 | 0.905 | 0.93 | 0.939 |
| | 0.5 | 0.907 | 0.929 | 0.939 |
| | 1 | 0.901 | 0.921 | 0.927 |
| | 5 | - | - | - |
| | 10 | - | - | - |
| Codestral 22B | 0 | 0.754 | 0.8 | 0.811 |
| | 0.5 | 0.759 | 0.804 | 0.817 |
| | 1 | 0.753 | 0.793 | 0.802 |
| | 5 | 0.693 | 0.763 | 0.80 |
| | 10 | 0.406 | 0.477 | 0.512 |
| LlaMa2 7B | 0 | 0.493 | 0.579 | 0.61 |
| | 0.05 | 0.487 | 0.576 | 0.604 |
| | 0.1 | 0.484 | 0.574 | 0.598 |
| | 0.5 | 0.422 | 0.522 | 0.555 |
| | 1 | 0.217 | 0.296 | 0.323 |
| Gemma 7B | 0 | 0.202 | 0.255 | 0.268 |
| | 0.05 | 0.199 | 0.253 | 0.262 |
| | 0.1 | 0.204 | 0.250 | 0.256 |
| | 0.5 | 0.045 | 0.068 | 0.079 |
| | 1 | - | - | - |
| DeepSeek V2 | 0 | 0.468 | 0.555 | 0.579 |
| | 0.05 | 0.465 | 0.555 | 0.573 |
| | 0.1 | 0.459 | 0.555 | 0.579 |
| | 0.5 | 0.428 | 0.545 | 0.567 |
| | 1 | 0.353 | 0.429 | 0.463 |
| Qwen3 Next | 0 | 0.941 | 0.950 | 0.951 |
| | 0.5 | 0.942 | 0.945 | 0.945 |
| | 1 | 0.938 | 0.947 | 0.951 |
| | 5 | 0.918 | 0.926 | 0.927 |
| | 10 | 0.687 | 0.727 | 0.732 |
| Mistral 7B | 0 | 0.345 | 0.417 | 0.451 |
| | 0.05 | 0.325 | 0.397 | 0.427 |
| | 0.1 | 0.332 | 0.392 | 0.409 |
| | 0.5 | 0.041 | 0.058 | 0.067 |
| | 1 | - | - | - |

J Random Steering Baseline

In order to distinguish sentiment-related steering effects from generic activation perturbations, we use a random vector for steering as a control baseline.

Tables 22 and 23 show the sentiment changes measured by PANAS and HAAS when using random vectors to steer LLMs. Table 24 shows the multi-random-vector controls using seeds (0-9). These results show that random steering can affect Pass@k, but its degradation is generally smaller and less systematic than sentiment-aligned negative steering. This supports a more careful and empirically grounded conclusion. Figures 12, 13 and 14 shows code generation and code summarization results under random-vector steering. Compared with sentiment steering, the effects of random-vector steering are negligible.

Table 25 shows the results of perplexity before and under random vector steering.

Table 22: PANAS-measured output-level sentiment tendencies of LLMs after positive or negative steering using random vector.

| LLMs | Default | | | Positive Steering | | | Negative Steering | | |
|--------------------------|---------|----|-----------|-------------------|----------|-----------|-------------------|----------|-----------|
| | PA | NA | Pattern | PA | NA | Pattern | PA | NA | Pattern |
| CodeLlama 7B | 37 | 33 | H-PA/H-NA | 38 (↑ 1) | 32 (↓ 1) | H-PA/H-NA | 40 (↑ 3) | 34 (↑ 1) | H-PA/H-NA |
| CodeGemma 7B | 39 | 13 | H-PA/L-NA | 38 (↓ 1) | 13 (0) | H-PA/L-NA | 38 (↓ 1) | 14 (↑ 1) | H-PA/L-NA |
| DeepSeek-Coder V2 | 40 | 24 | H-PA/H-NA | 40 (0) | 25 (↑ 1) | H-PA/H-NA | 40 (0) | 25 (↑ 1) | H-PA/H-NA |
| Qwen3-Coder Next | 38 | 12 | H-PA/L-NA | 38 (0) | 13 (↑ 1) | H-PA/L-NA | 39 (↑ 1) | 15 (↑ 3) | H-PA/L-NA |
| Codestral 22B | 43 | 20 | H-PA/H-NA | 40 (↓ 3) | 18 (↓ 2) | H-PA/L-NA | 40 (↓ 3) | 18 (↓ 2) | H-PA/L-NA |
| LlaMa2 7B | 34 | 21 | L-PA/H-NA | 32 (↓ 2) | 23 (↑ 2) | L-PA/H-NA | 33 (↓ 1) | 24 (↑ 3) | L-PA/H-NA |
| Gemma 7B | 29 | 17 | L-PA/L-NA | 28 (↓ 1) | 18 (↑ 1) | L-PA/L-NA | 28 (↓ 1) | 18 (↑ 1) | L-PA/L-NA |
| DeepSeek V2 | 30 | 18 | L-PA/L-NA | 25 (↑ 5) | 16 (↓ 2) | L-PA/L-NA | 25 (↑ 5) | 16 (↓ 2) | L-PA/L-NA |
| Qwen3 Next | 46 | 13 | H-PA/L-NA | 46 (↓ 1) | 14 (↑ 1) | H-PA/L-NA | 45 (↓ 1) | 5 (↑ 2) | H-PA/L-NA |
| Mistral 7B | 36 | 18 | H-PA/L-NA | 37 (↑ 1) | 19 (↑ 1) | H-PA/H-NA | 37 (↑ 1) | 19 (↑ 1) | H-PA/H-NA |

Table 23: HAAS-measured output-level sentiment tendencies of LLMs after positive or negative steering using random vector.

| LLMs | Default | | | | Positive Steering | | | | Negative Steering | | | |
|--------------------------|----------|---------|----------|---------|-------------------|----------|----------|---------|-------------------|----------|----------|---------|
| | Pos High | Pos Low | Neg High | Neg Low | Pos High | Pos Low | Neg High | Neg Low | Pos High | Pos Low | Neg High | Neg Low |
| CodeLlama 7B | 9 | 6 | 6 | 6 | 9 (0) | 6 (0) | 6 (0) | 6 (0) | 9 (0) | 6 (0) | 6 (0) | 6 (0) |
| CodeGemma 7B | 9 | 2 | 3 | 6 | 6 (↓ 3) | 3 (↑ 1) | 1 (↓ 2) | 6 (0) | 9 (0) | 3 (↑ 1) | 2 (↓ 1) | 6 (0) |
| DeepSeek-Coder V2 | 4 | 4 | 2 | 1 | 4 (0) | 4 (0) | 0 (↓ 4) | 1 (0) | 4 (0) | 4 (0) | 0 (↓ 4) | 1 (0) |
| Qwen3-Coder Next | 12 | 10 | 4 | 6 | 10 (↓ 2) | 9 (↓ 1) | 3 (↓ 1) | 5 (↓ 1) | 10 (↓ 2) | 9 (↓ 1) | 3 (↓ 1) | 5 (↓ 1) |
| Codestral 22B | 6 | 9 | 3 | 4 | 6 (0) | 10 (↑ 1) | 2 (↓ 1) | 4 (0) | 6 (0) | 8 (↓ 1) | 2 (↓ 1) | 4 (0) |
| LlaMa2 7B | 4 | 5 | 3 | 5 | 4 (0) | 5 (0) | 3 (0) | 5 (0) | 6 (↑ 2) | 6 (↑ 1) | 3 (0) | 5 (0) |
| Gemma 7B | 6 | 10 | 1 | 10 | 6 (0) | 9 (↓ 1) | 1 (0) | 9 (↓ 3) | 6 (0) | 9 (↓ 1) | 1 (0) | 9 (↓ 1) |
| DeepSeek V2 | 8 | 7 | 2 | 1 | 7 (↓ 1) | 7 (0) | 2 (0) | 2 (↑ 1) | 7 (↓ 1) | 8 (↑ 1) | 2 (0) | 3 (↑ 2) |
| Qwen3 Next | 11 | 10 | 4 | 6 | 8 (↓ 3) | 10 (0) | 1 (↓ 3) | 3 (↓ 3) | 8 (↓ 3) | 10 (0) | 1 (↓ 3) | 3 (↓ 3) |
| Mistral 7B | 7 | 12 | 3 | 4 | 7 (0) | 11 (↓ 1) | 3 (0) | 6 (↑ 2) | 7 (0) | 11 (↓ 1) | 3 (0) | 6 (↑ 2) |

Table 24: Steering performance on HumanEval under multi-random-vector controls using seeds 0–9. Rand Pos represents random positive steering, while Rand Neg represents random negative steering. Pass@k shows the mean result across random seeds, with standard deviations.

| LLMs | Setting | Pass@1 | Pass@5 | Pass@10 |
|-------------------|----------|----------|----------|----------|
| CodeLlama 7B | Rand Pos | 37% ± 1% | 43% ± 2% | 45% ± 2% |
| | Rand Neg | 28% ± 3% | 37% ± 4% | 40% ± 6% |
| CodeGemma 7B | Rand Pos | 49% ± 1% | 67% ± 2% | 72% ± 2% |
| | Rand Neg | 49% ± 1% | 67% ± 2% | 72% ± 2% |
| DeepSeek-Coder V2 | Rand Pos | 78% ± 2% | 80% ± 2% | 81% ± 2% |
| | Rand Neg | 78% ± 2% | 80% ± 2% | 81% ± 2% |
| Qwen3-Coder Next | Rand Pos | 91% ± 0% | 93% ± 0% | 93% ± 1% |
| | Rand Neg | 90% ± 1% | 93% ± 1% | 93% ± 1% |
| Codestral 22B | Rand Pos | 76% ± 1% | 81% ± 1% | 82% ± 1% |
| | Rand Neg | 67% ± 3% | 73% ± 4% | 75% ± 4% |
| LlaMa2 7B | Rand Pos | 48% ± 1% | 57% ± 1% | 60% ± 1% |
| | Rand Neg | 41% ± 2% | 50% ± 3% | 53% ± 3% |
| Gemma 7B | Rand Pos | 20% ± 1% | 25% ± 1% | 26% ± 2% |
| | Rand Neg | 20% ± 1% | 24% ± 1% | 26% ± 2% |
| DeepSeek V2 | Rand Pos | 47% ± 1% | 55% ± 1% | 58% ± 2% |
| | Rand Neg | 45% ± 2% | 54% ± 2% | 57% ± 2% |
| Qwen3 Next | Rand Pos | 93% ± 1% | 93% ± 1% | 94% ± 1% |
| | Rand Neg | 93% ± 1% | 94% ± 1% | 94% ± 1% |
| Mistral 7B | Rand Pos | 35% ± 1% | 41% ± 2% | 44% ± 2% |
| | Rand Neg | 16% ± 1% | 20% ± 1% | 21% ± 2% |

Table 25: Comparison of LLM perplexity before and after random steering.

| LLMs | Default | Positive Steering | Negative Steering |
|--------------------------|---------|-------------------|-------------------|
| CodeLlama 7B | 1.1 | 1.1 | 1.2 |
| CodeGemma 7B | 1.2 | 1.2 | 1.2 |
| DeepSeek-Coder V2 | 1.0 | 1.0 | 1.0 |
| Qwen3-Coder Next | 1.1 | 1.1 | 1.1 |
| Codestral 22B | 1.1 | 1.1 | 1.1 |
| LlaMa2 7B | 1.1 | 1.1 | 1.1 |
| Gemma 7B | 1.1 | 1.1 | 1.1 |
| DeepSeek V2 | 1.1 | 1.1 | 1.2 |
| Qwen3 Next | 1.0 | 1.0 | 1.0 |
| Mistral 7B | 1.2 | 1.1 | 1.2 |

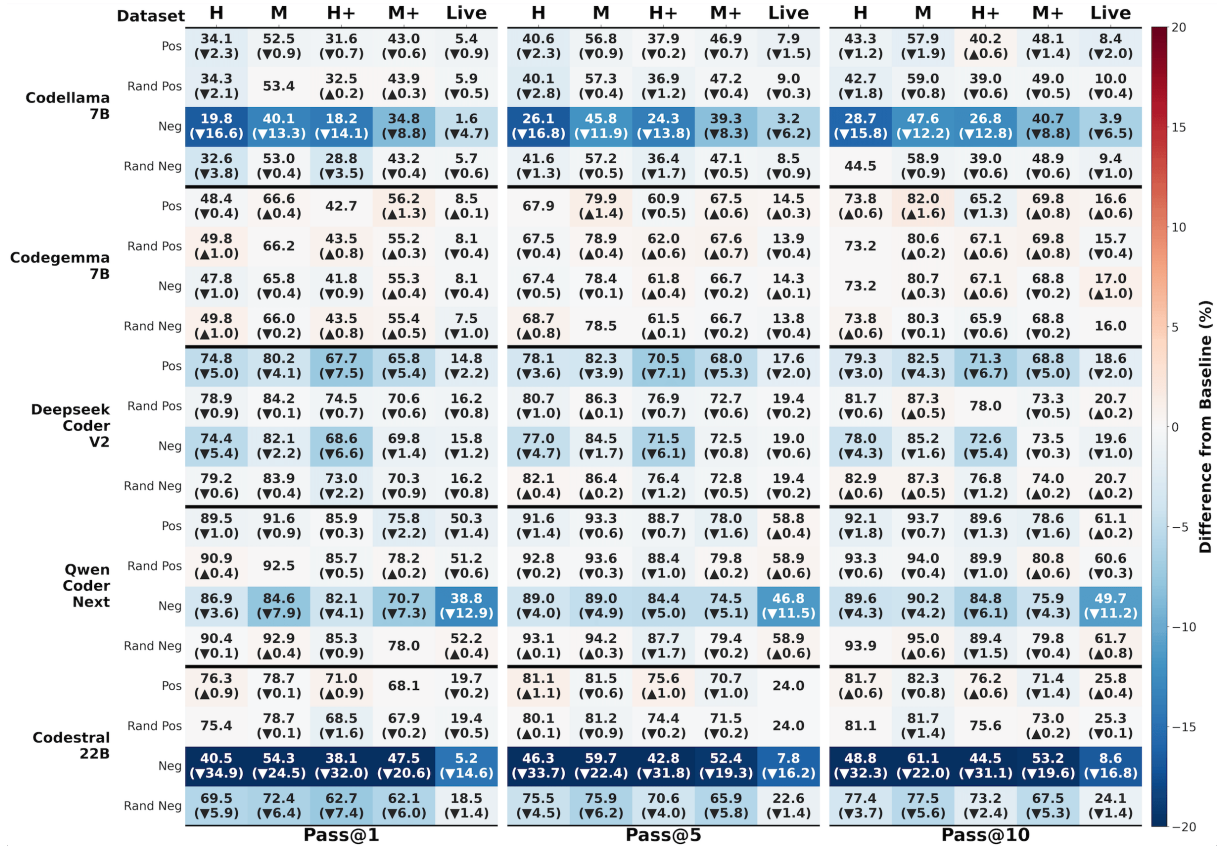


Figure 12: Positive and negative steering effects on Code-LLM code generation performance. The three main columns report Pass@1 (%), Pass@5 (%), and Pass@10 (%). For each LLM, four rows are presented: Pos and Neg show the effects of positive and negative steering, respectively, while Rand Pos and Rand Neg show the effects of random positive and negative steering using seed 42. Red indicates a positive effect on code generation performance, while blue indicates a negative effect. H denotes the HumanEval dataset, H+ denotes HumanEval+, M denotes MBPP, M+ denotes MBPP+, and Live denotes LiveCodeBench.

| Dataset | H | M | H+ | M+ | Live | H | M | H+ | M+ | Live | H | M | H+ | M+ | Live | | |
|-------------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|--|
| LlaMa2 7B | Pos | 47.0
(▼2.3) | 60.5
(▼2.5) | 43.7
(▼1.5) | 51.8
(▼1.3) | 4.8
(▼2.4) | 55.3
(▼2.6) | 67.0
(▼1.4) | 50.2
(▼3.1) | 57.6
(▼0.3) | 6.9
(▼3.2) | 58.5
(▼2.5) | 68.8
(▼1.3) | 53.0
(▼3.1) | 59.0
(▼0.8) | 7.8
(▼3.3) | |
| | Rand Pos | 49.0
(▼0.3) | 62.4
(▼0.6) | 44.9
(▼0.3) | 52.5
(▼0.6) | 7.7
(▲0.6) | 57.8
(▼0.1) | 68.9
(▲0.5) | 53.2
(▼0.1) | 57.9
(▲1.1) | 11.2
(▲1.1) | 61.0
(▼4.3) | 70.6
(▼5.3) | 55.5
(▼5.5) | 59.5
(▼6.6) | 12.5
(▲1.4) | |
| | Neg | 20.6
(▼28.7) | 43.8
(▼19.2) | 17.2
(▼28.0) | 37.1
(▼16.0) | 4.0
(▼3.1) | 28.4
(▼29.5) | 52.8
(▼15.6) | 25.9
(▼27.4) | 44.0
(▼13.9) | 6.5
(▼3.6) | 31.7
(▼29.3) | 55.3
(▼14.8) | 29.9
(▼26.2) | 46.3
(▼13.5) | 7.6
(▼3.5) | |
| | Rand Neg | 44.0
(▼5.3) | 56.3
(▼6.7) | 38.8
(▼6.4) | 46.2
(▼6.9) | 5.4
(▼1.7) | 53.5
(▼4.4) | 63.2
(▼5.2) | 47.7
(▼5.6) | 51.6
(▼6.3) | 8.0
(▼2.1) | 56.7
(▼4.3) | 64.8
(▼5.3) | 50.6
(▼5.5) | 53.2
(▼6.6) | 9.0
(▼2.2) | |
| Gemma 7B | Pos | 19.6
(▼0.6) | 42.7
(▼1.5) | 17.6
(▼0.5) | 36.8
(▼1.2) | 3.3
(▼0.3) | 24.2
(▼1.3) | 47.4
(▼2.3) | 21.4
(▼1.2) | 40.6
(▼1.2) | 5.1
(▲0.7) | 25.6
(▼1.2) | 48.1
(▼2.7) | 22.6
(▼1.2) | 41.3
(▼1.3) | 5.9
(▲0.4) | |
| | Rand Pos | 19.8
(▼0.4) | 43.9
(▼0.3) | 17.5
(▼0.6) | 37.6
(▼0.4) | 4.2
(▲0.7) | 26.1
(▲0.6) | 48.9
(▼0.8) | 22.9
(▲0.3) | 41.7
(▼0.1) | 5.8
(▲0.7) | 27.4
(▲0.6) | 49.7
(▼1.1) | 24.4
(▲0.6) | 42.3
(▼0.3) | 6.1
(▲0.6) | |
| | Neg | 20.5
(▲0.3) | 43.5
(▼0.7) | 17.8
(▼0.3) | 38.1
(▲0.1) | 3.3
(▼0.2) | 27.2
(▲1.7) | 49.6
(▼0.1) | 22.8
(▲0.2) | 42.9
(▲1.1) | 4.6
(▼0.5) | 29.3
(▲2.5) | 51.1
(▲0.3) | 25.0
(▲1.2) | 43.9
(▲1.3) | 5.1
(▼0.4) | |
| | Rand Neg | 20.1
(▼0.1) | 44.2
(▼0.3) | 18.1
(▼0.3) | 37.7
(▲0.7) | 4.2
(▲0.7) | 25.5
(▼0.5) | 49.2
(▼0.5) | 22.8
(▲0.2) | 41.7
(▼0.1) | 5.7
(▲0.7) | 26.8
(▼0.3) | 50.5
(▼0.3) | 23.8
(▼0.3) | 42.3
(▼0.3) | 5.9
(▲0.5) | |
| DeepSeek V2 | Pos | 47.5
(▲0.7) | 57.2
(▼1.3) | 43.4
(▲0.3) | 49.0
(▼1.4) | 4.4
(▲0.2) | 53.5
(▼2.0) | 63.7
(▼1.5) | 49.7
(▼1.5) | 54.2
(▼1.6) | 7.9
(▲1.6) | 54.9
(▼3.0) | 65.9
(▼1.6) | 51.2
(▼1.8) | 55.8
(▼1.6) | 9.4
(▲2.0) | |
| | Rand Pos | 46.6
(▼0.2) | 58.1
(▼0.4) | 42.7
(▼0.4) | 50.1
(▼0.3) | 3.9
(▼0.3) | 54.3
(▼1.2) | 65.2
(▼0.7) | 50.5
(▼0.7) | 56.0
(▲0.2) | 6.1
(▼0.3) | 57.3
(▼0.6) | 67.5
(▼0.5) | 53.7
(▲0.7) | 57.9
(▲0.5) | 6.5
(▼1.0) | |
| | Neg | 35.3
(▼11.5) | 52.1
(▼6.4) | 30.8
(▼12.3) | 44.2
(▼6.2) | 2.5
(▼1.7) | 44.0
(▼11.5) | 59.3
(▼5.9) | 37.9
(▼13.3) | 49.7
(▼6.1) | 4.3
(▼2.1) | 48.2
(▼9.7) | 61.4
(▼6.1) | 41.5
(▼11.5) | 51.1
(▼6.3) | 4.7
(▼2.7) | |
| | Rand Neg | 46.6
(▼0.2) | 57.6
(▼0.9) | 42.3
(▼0.8) | 50.5
(▲0.1) | 4.0
(▼0.2) | 56.0
(▲0.5) | 65.7
(▲0.5) | 49.9
(▼1.3) | 55.8
(▼0.3) | 6.5
(▲0.2) | 59.1
(▲1.2) | 68.0
(▲0.5) | 53.0
(▼0.7) | 58.2
(▲0.8) | 7.6
(▲0.2) | |
| Qwen Next | Pos | 89.0
(▼5.1) | 90.3
(▼2.1) | 85.9
(▼4.8) | 75.9
(▼2.6) | 44.7
(▼8.8) | 90.5
(▼4.5) | 91.2
(▼1.8) | 87.0
(▼4.3) | 77.2
(▼2.5) | 49.7
(▼8.1) | 90.9
(▼4.2) | 91.5
(▼1.6) | 87.2
(▼4.3) | 77.8
(▼2.1) | 50.2
(▼9.3) | |
| | Rand Pos | 94.3
(▲0.2) | 92.0
(▼0.4) | 89.0
(▼1.7) | 78.0
(▼0.5) | 52.3
(▼1.2) | 95.1
(▲0.1) | 92.7
(▼0.3) | 90.7
(▼0.6) | 79.1
(▼0.6) | 54.3
(▼3.5) | 95.1
(▼0.2) | 92.9
(▼0.2) | 90.9
(▼0.6) | 78.6
(▼1.3) | 56.9
(▼2.6) | |
| | Neg | 68.0
(▼26.1) | 72.5
(▼19.9) | 62.4
(▼28.3) | 62.5
(▼16.0) | 25.4
(▼28.1) | 71.4
(▼23.6) | 74.0
(▼19.0) | 65.3
(▼26.0) | 63.9
(▼15.8) | 28.4
(▼29.5) | 72.6
(▼22.5) | 74.3
(▼18.8) | 66.5
(▼25.0) | 64.3
(▼15.6) | 29.5
(▼29.9) | |
| | Rand Neg | 94.3
(▲0.2) | 92.9
(▲0.5) | 88.7
(▼2.0) | 78.2
(▼0.3) | 54.0
(▲0.6) | 94.8
(▼0.2) | 92.6
(▼0.4) | 88.7
(▼2.6) | 79.4
(▼0.3) | 59.8
(▲1.9) | 95.1
(▼0.4) | 93.5
(▲0.4) | 88.7
(▼2.8) | 79.2
(▼0.7) | 59.9
(▲0.4) | |
| Mistral 7B | Pos | 35.5
(▲1.0) | 48.5
(▼0.1) | 30.9
(▲0.5) | 41.1
(▼0.5) | 4.8
(▲0.3) | 41.7
(▼0.5) | 53.9
(▼0.6) | 36.4
(▲1.3) | 45.0
(▼1.2) | 6.9
(▲0.2) | 43.9
(▼1.2) | 55.6
(▼0.5) | 38.4
(▼1.2) | 46.3
(▼1.1) | 7.8
(▼0.2) | |
| | Rand Pos | 34.4
(▼0.1) | 49.0
(▲0.4) | 30.5
(▲0.1) | 42.0
(▲0.4) | 4.3
(▼0.1) | 41.2
(▼0.5) | 54.2
(▼0.3) | 35.3
(▲0.2) | 45.8
(▼0.4) | 6.3
(▼0.4) | 43.9
(▼1.2) | 55.8
(▼0.3) | 37.2
(▼1.2) | 47.1
(▼0.3) | 7.4
(▼0.6) | |
| | Neg | 1.1
(▼33.4) | 1.2
(▼47.4) | 1.1
(▼29.3) | 0.9
(▼40.7) | 0.0
(▼4.5) | 1.2
(▼40.5) | 1.6
(▼52.9) | 1.2
(▼33.9) | 1.2
(▼45.0) | 0.0
(▼6.7) | 1.2
(▼43.9) | 1.9
(▼54.2) | 1.2
(▼37.2) | 1.3
(▼46.1) | 0.0
(▼8.0) | |
| | Rand Neg | 16.5
(▼18.0) | 34.3
(▼14.3) | 13.8
(▼16.6) | 28.9
(▼12.7) | 4.3
(▼0.1) | 21.8
(▼19.9) | 40.3
(▼14.2) | 18.1
(▼17.0) | 34.0
(▼12.2) | 6.3
(▼0.4) | 24.4
(▼20.7) | 42.3
(▼13.8) | 19.5
(▼18.9) | 35.4
(▼12.0) | 7.4
(▼0.6) | |
| Pass@1 | | | | | | Pass@5 | | | | | | Pass@10 | | | | | |

Difference from Baseline (%)

Figure 13: Positive and negative steering effects on general-purpose LLM code generation performance. The three main columns report Pass@1 (%), Pass@5 (%), and Pass@10 (%). For each LLM, four rows are presented: Pos and Neg show the effects of positive and negative steering, respectively, while Rand Pos and Rand Neg show the effects of random positive and negative steering. Red indicates a positive effect on code generation performance, while blue indicates a negative effect. H denotes the HumanEval dataset, H+ denotes HumanEval+, M denotes MBPP, M+ denotes MBPP+, and Live denotes LiveCodeBench.

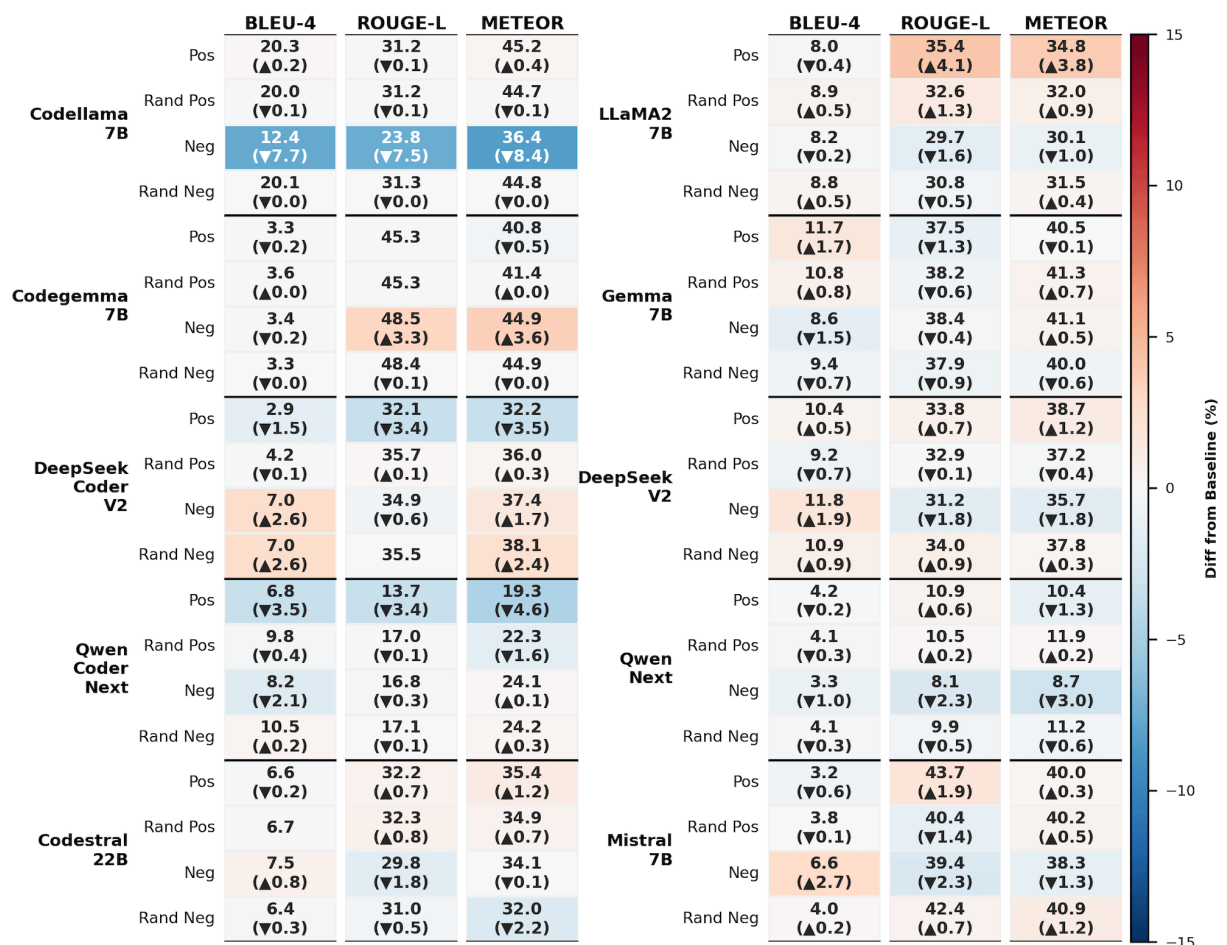


Figure 14: Positive and negative steering effects on LLM code summarization performance.

K Risk Assessment in the Sentiment Dataset

To minimize potential risks associated with offensive content, we used publicly available sentiment datasets whose original papers did not report the presence of offensive or abusive language. In addition, we manually inspected a random sample of 500 instances labeled as negative and did not identify explicit offensive words in the sampled data. However, since only a subset of the dataset was examined, we cannot fully exclude the possibility that some offensive expressions may exist in unsampled instances.