How LLMs Follow Instructions: Skillful Coordination, Not a Universal Mechanism

Anonymous ACL submission

Abstract

Instruction tuning is commonly assumed to endow language models with a domain-general ability to follow instructions, yet the underlying mechanism remains poorly understood. Does instruction-following rely on a universal mechanism or compositional skill deployment? We investigate this through diagnostic probing across nine diverse tasks in three instruction-tuned models.

Our analysis provides converging evidence against a universal mechanism. First, general probes trained across all tasks show selective rather than uniform deficits relative to task-specific specialists, inconsistent with a shared representation. Second, cross-task transfer is weak and clustered by skill similarity. Third, causal ablation reveals sparse asymmetric dependencies rather than shared representations. Tasks also stratify by complexity across layers, with structural constraints emerging early and semantic tasks emerging late. Finally, temporal analysis shows constraint satisfaction operates as dynamic monitoring during generation rather than pre-generation planning.

These findings indicate that instruction-following is better characterized as skillful coordination of diverse linguistic capabilities rather than deployment of a single abstract constraint-checking process.

1 Introduction

The remarkable ability of Large Language Models (LLMs) to follow diverse user instructions has driven much of their recent success. However, while benchmarks (Zhou et al., 2023; Qin et al., 2024b) have quantified what models can achieve, the internal mechanisms governing how they maintain compliance remain poorly understood. Recent evidence suggests that instruction following may not be a fragmented set of behaviors, but rather a property encoded within a dedicated "instruction-following dimension" in the model's latent space (Heo et al., 2025; Stolfo et al., 2025).

This raises a key architectural question: Is instruction following driven by task-specific skills, or by a general, task-independent representation of rule-following? In this work, we investigate this question by proposing a framework that disentangles two distinct capabilities: (1) **task-specific skills**, the localized linguistic knowledge required to execute a command, such as identifying sentiment or formatting text; and (2) **constraint satisfaction**, a universal, task-invariant cognitive state of actively adhering to a requested instruction.

This distinction leads to our central research question: Does a model encode a representation of *constraint satisfaction* that is distinct from its *task-specific skills*? To explore this, we investigate the potential representation through two lenses: its **scope** and its **temporal dynamics**.

RQ1: What is the scope of the constraint satisfaction representation? This question breaks down into two parts. *Universality*: to what extent does the representation generalize across different, unrelated tasks? *Specificity*: does this signal activate only when an explicit constraint is given, or does it also appear in open-ended generation?

RQ2: What are the temporal dynamics of this representation? We investigate when the signal is active to understand its role in the cognitive process. *Planning*: when does the signal first become detectable? *Monitoring*: how long does the signal persist during generation?

To address these questions, we employ a diagnostic framework combining linear probing, cross-task transfer analysis, and causal intervention. We train *specialist* probes (task-specific) and *general* probes (all tasks) to distinguish successful from failed constraint satisfaction across nine diverse instruction-following tasks. By comparing probe accuracy and rowspace information, we quantify

representational sharing. Temporal analysis extracts representations at multiple generation stages to trace when constraint satisfaction signals emerge and persist. Finally, PWCCA-based dendograms reveals the organizazional structure of cross-task information similarity.

The remainder of this paper proceeds as follows. §2 reviews related work and our contributions. §3 describes our diagnostic probing framework and analysis methods. §4 details our experimental setup. §5 presents our findings. §6 concludes with implications for future research.

2 Related work

Benchmarking instruction-tuned LLMs. Evaluation methodologies have shifted from subjective assessments to verifiable constraints. Benchmarks like IFEval and InFoBench employ objective criteria to automate compliance verification (Qin et al., 2024b; Zhou et al., 2023), while FollowBench and SysBench extend evaluations to multi-turn dialogues and system-level instructions (Jiang et al., 2024; Qin et al., 2024a). These reveal that modern LLMs struggle with multiple fine-grained constraints despite handling simple tasks well.

Interpreting instruction-following mechanisms. To study instruction tuning, researchers probe LLM internals (Alain and Bengio, 2017; Belinkov, 2022), mapping surface syntax (Tenney et al., 2019) to detect high-level cognitive states like truthfulness (Li et al., 2023) and confidence (Kadavath et al., 2022). Recent work analyzes how instruction tuning reshapes representations: Wu et al. (2024) found that tuning shifts focus toward instruction-specific verbs and rotates knowledge representations toward user-oriented tasks, while Heo et al. (2025) identified an "instruction-following dimension" serving as an internal compliance predictor. He et al. (2025); Stolfo et al. (2025) demonstrate these signals can be manipulated via steering vectors to improve adherence without additional fine-tuning.

Mechanistic analyses of fine-tuning. Prior work suggests that fine-tuning is governed by low-dimensional structure: fine-tuning has a low intrinsic dimension (Aghajanyan et al., 2021), task-specific behavior can be isolated and composed as task vectors (Ilharco et al., 2023), and different tasks appear already integrated during pre-training (Qin et al., 2022). Related work studies how fine-tuning displaces representations (Zhou and Srikumar, 2022) and how such displacements can be inferred from activation differences for model diffing (Minder et al., 2025). Unlike this line of work, which asks how weights or representations are displaced by fine-tuning, we ask whether instruction-following itself is mediated by a single task-invariant mechanism. The diagnostic tools we employ are nonetheless directly applicable to model diffing, and whether the compositional, low-dimensional organization we report is specific to instruction tuning or a more general property of fine-tuning remains open.

2.1 Contributions

Our research extends the frameworks of Heo et al. (2025) and Stolfo et al. (2025) in three ways. First, rather than adopting IFEval's split between base queries and instructions—which yields labels that can ignore total prompt fidelity—we evaluate adherence over the entire input as a single task, across tasks spanning a broader range of linguistic complexity than IFEval's heuristics cover. Second, while Stolfo et al. (2025) suggest that first-token representations suffice to predict adherence, we trace the longitudinal dynamics across the full generation process. Our contributions are the following:

1. We develop an extensible and comprehensive diagnostic framework integrating specialist and general probing, cross-task transfer, causal ablation, and temporal analysis across nine diverse instruction-following tasks[1].

2. We provide converging evidence against a universal constraint satisfaction mechanism, revealing instead compositional skill deployment with sparse asymmetric dependencies.

3. We discover that constraint satisfaction operates as dynamic monitoring during generation rather than pre-generation planning.

3 Methodology

We use diagnostic probing (Alain and Bengio, 2017) to classify model responses as *Success* or *Failure* based on task adherence. Our investigation of RQ1 (§3.2 and §3.3) is structured as a coherent orchestration of analyses, where each step strengthens and validates the preceding conclusions.

[1]Our code and datasets will be made publicly available upon acceptance. More details about the implementation design are in Appendix B.

Task	Prompt	Option	Correct	Incorrect
Character count	Generate a sentence with *option* chars	10	Bird sings	Bird sings high.
Word count	Generate a sentence with *option* words	4	The sky is blue.	I love music.
JSON format	Describe *option* as a JSON object	an animal	`{ "fur": "black" }`	`"Fur": black`
Word inclusion	Generate a sentence with the word *option*	house	I live in a tiny house.	The rent is too high.
Word exclusion	Generate a sentence without the word *option*	house	The rent is too high.	I live in a tiny house.
Topic	Write a news text about *option*	sports	Today we discuss the football game...	This artist's paintings are...
Sentiment	Write an *option* review	positive	I really liked this product.	I hate this product.
Toxicity	Generate a *option* comment	non-toxic	I do not appreciate your content	I cannot stand your ****** face
Register	Generate a *option* text	formal	The ancient scrolls whispered forgotten...	I'm going to make a giant batch of...

Table 1: Overview of the nine experimental tasks. Includes prompt templates and illustrative correct/incorrect examples used to test a range of linguistic instruction-following competencies.

3.1 Tasks

We designed nine tasks spanning four linguistic dimensions: structural (e.g., word counts), lexical (word inclusion), semantic (topic/sentiment), and stylistic (formality). As shown in Table 1, datasets use fluent but "incorrect" responses to distinguish constraint adherence from mere linguistic well-formedness, pairing multiple prompt templates with task-specific options via a balanced mix of LLM-generated and existing data (see Appendix A).

3.2 Universality (RQ1)

We investigate whether a shared, task-agnostic representation for constraint satisfaction exists, using three complementary methods that each build on the previous and avoid the same confounds.

***General* vs. *specialist* probes** We compare task-specific *specialist* probes against a single *general* probe trained on all tasks. This comparison is preliminary rather than decisive: a specialist fits a single distribution while the general probe accommodates a mixture, so a uniform *specialist* advantage is expected under *any* account. What is diagnostic is not the performance gap but its *structure*. We then build on this comparison with the OOD generalization and cross-task ablation analyses below.

Out-of-Distribution (OOD) generalization For each *specialist* probe, we evaluate its accuracy on all other unseen tasks. Unlike the previous comparison, this avoids the optimization-asymmetry confound: it asks whether any representation encodes the broad, approximately uniform cross-task structure a universal mechanism would require. Selective transfer confined to related tasks, with near-chance accuracy elsewhere, would instead indicate compositional structure.

Cross-Task ablation Using Iterative Null-space Projection (INLP) (Ravfogel et al., 2020) on the best linear probes, we identify the rowspace P^B_{row} and nullspace P^B_{null} of a source-task-B probe; we then project target-task-A activations onto P^B_{null}, and measure the **normalized accuracy drop**:

$$\text{NormDrop} = \frac{\text{Acc}_{\text{base}} - \text{Acc}_{\text{ablated}}}{\text{Acc}_{\text{base}} - 0.5} \quad (1)$$

High NormDrop would causally link the two tasks, implying their underlying representations are shared. A negligible drop would suggest they are encoded independently.

3.3 Specificity (RQ1)

To determine if the signal is exclusive to constraint-following, we compare constrained-task activations to a *null* baseline of open-ended generation without chat templates (e.g. "What a beautiful"). Using the rowspace P_{row} extracted via INLP, we compute the **signal intensity**:

$$\text{Intensity}(X) = \left\| X \cdot (P_{\text{row}})^T \right\|_2 \quad (2)$$

Lower *null* task intensity suggests constraint-specific signals; comparable intensities indicate reliance on general language modeling features.

We choose open-ended generation as the baseline because it elicits no instruction-following behavior, avoiding engagement of instruction-following circuitry. A constraint-free version of the same instruction would not serve this purpose: since the task is defined as the full prompt (§3.1), the constraint is inseparable from the instruction and its removal does not constitute a well-defined condition.

3.4 Temporal dynamics analysis (RQ2)

We distinguish three token positions when training probes and extracting rowspaces: (1) *connectors*, special tokens between the user prompt and the model's response; (2) *body*, tokens forming the generated response; (3) *EOS*, the final end-of-sequence token. For each task and position, we

train linear and non-linear probes on activations from two components: after attention and after the MLP block.

4 Experimental setup

All experiments were conducted on one NVIDIA H100 NVL GPU with 94GB of memory.

Data Each task draws on a mix of LLM-generated responses and existing corpora, with negatives produced by cross-option swapping to enforce a distinction between constraint adherence and linguistic well-formedness. Data sources, options, and verification logic are detailed in Appendix A (Table 3).

Models We experiment on three instruction-tuned models of varying sizes and architectures: Llama 3.1 8B Instruct (Grattafiori et al., 2024), Gemma 2 2B IT (Gemma Team et al., 2024), and Qwen2.5-0.5B-Instruct (Qwen et al., 2025), with 32, 26, and 24 layers and hidden dimensions of 4096, 2304, and 896 respectively.

Probing and INLP We trained five probe architectures (logistic regression, SGD, random forests, k-NN, and MLP) across three random seeds, selecting the best linear and non-linear models: logistic regression and a single-hidden-layer MLP with 128 neurons and ReLU activation. For INLP, at each step we train a linear classifier, record its weight vector (w_i), and project activations into its nullspace, halting when test accuracy falls below 0.55. This yields P_{null} and the complementary rowspace $P_{\text{row}} = I - P_{\text{null}}$.

5 Results

5.1 (RQ1) Universality: no evidence for a general constraint satisfaction mechanism

***General* probes show selective, not uniform, deficits.** Table 2 reports mean accuracy aggregated across all probes, layers, and scopes. The *general* probe trails *specialists* overall, but what is diagnostic is the *structure* of this gap: rather than the uniform margin a pure optimization-asymmetry account predicts, the deficit is selective—the *general* probe matches some specialists on their own tasks (e.g., topic, sentiment) while trailing others—indicating shared structure for some task families but not all. Confusion matrices show approximately balanced error rates across compliant and non-compliant classes (Appendix C), ruling out a degenerate strategy exploiting class imbalance.

Task	Gemma	Llama	Qwen
Character Count	0.84 ± 0.21	0.82 ± 0.24	0.84 ± 0.20
Term Exclusion	0.83 ± 0.20	0.82 ± 0.23	0.83 ± 0.20
JSON Format	0.82 ± 0.20	0.81 ± 0.22	0.83 ± 0.20
Word Count	0.73 ± 0.15	0.76 ± 0.19	0.72 ± 0.14
Register	0.75 ± 0.20	0.78 ± 0.23	0.67 ± 0.18
Term Inclusion	0.73 ± 0.17	0.75 ± 0.20	0.68 ± 0.15
Toxicity	0.70 ± 0.17	0.74 ± 0.19	0.62 ± 0.14
Sentiment	0.69 ± 0.16	0.71 ± 0.18	0.63 ± 0.14
Topic	0.66 ± 0.13	0.69 ± 0.16	0.60 ± 0.09
General	0.68 ± 0.15	0.72 ± 0.18	0.63 ± 0.12

Table 2: Task-specific model performance. Results show mean accuracy over all probes, layers and scopes with standard deviation.

Tasks stratify by complexity across layers. Figure 1 reveals distinct emergence patterns across network depth. Tasks cluster by when constraint information becomes detectable: *early-emerging* tasks (character count, term exclusion, JSON format, word count) reach >0.9 accuracy within the first layers, while *late-emerging* tasks (word inclusion, register, sentiment, topic, toxicity) peak later. This stratification suggests that different tasks rely on information encoded at different levels of abstraction: early layers capture fundamental structural and lexical skills, while later layers encode semantic and stylistic competencies.

Cross-task transfer reveals skill composition. Figure 2a shows cross-task transfer: cell (i, j) reports probe j's accuracy on task i's data. Against the broad, approximately uniform transfer a universal mechanism predicts, the *general* probe (first column) fails to consistently outperform *specialists* on held-out tasks and transfer is selective: some *specialist* probes transfer well to related tasks (e.g., Llama's topic probe reaches 0.78–0.87 on sentiment and term exclusion), while others stay near-chance on unrelated tasks—pointing to constraint satisfaction encoded as a composition of intermediate-level skills shared across subsets of related tasks.

Causal ablation shows sparse, asymmetric task dependencies. Figure 2b shows normalized accuracy drops (Equation 1): cell (i, j) represents task i's probe degradation after removing task j's information via nullspace projection $(X^i \cdot (P^j_{\text{null}})^T) \cdot P^j_{\text{null}}$. Across all three models, we observe sparse,

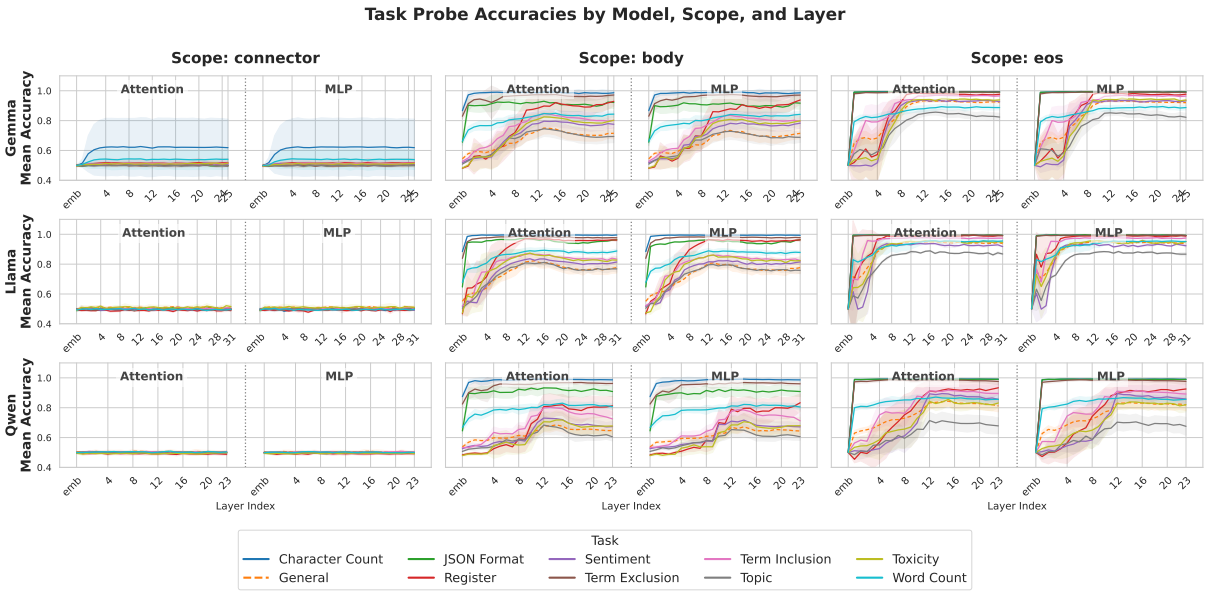
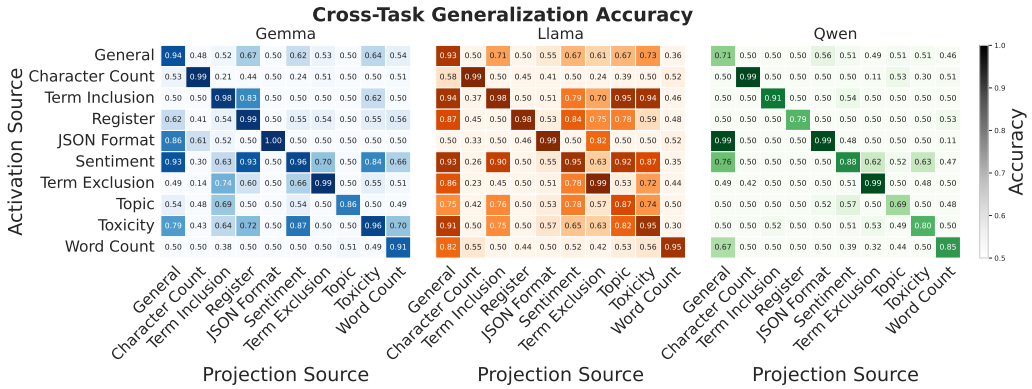
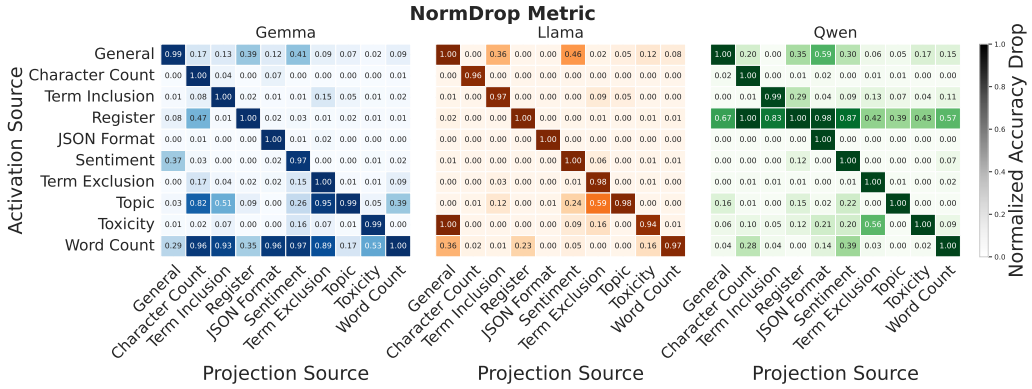


Figure 1: Probe accuracy across network layers for attention (left) and MLP (right) streams, separated by scope. Accuracy values are averaged over the best-performing linear and nonlinear probe for each condition. Colored lines: specialist probes; dashed line: general probe.



(a)



(b)

Figure 2: (a) Cross-task generalization: cell (i, j) shows probe j accuracy on task i data. (b) Normalized accuracy drop: cell (i, j) shows probe i performance drop after removing probe j information via nullspace projection. Darker colors indicate stronger dependencies. Best linear probes used throughout.

asymmetric dependency patterns rather than dense connectivity through a general mechanism. For Gemma, topic and word count probes rely on information from multiple other tasks. For Llama,

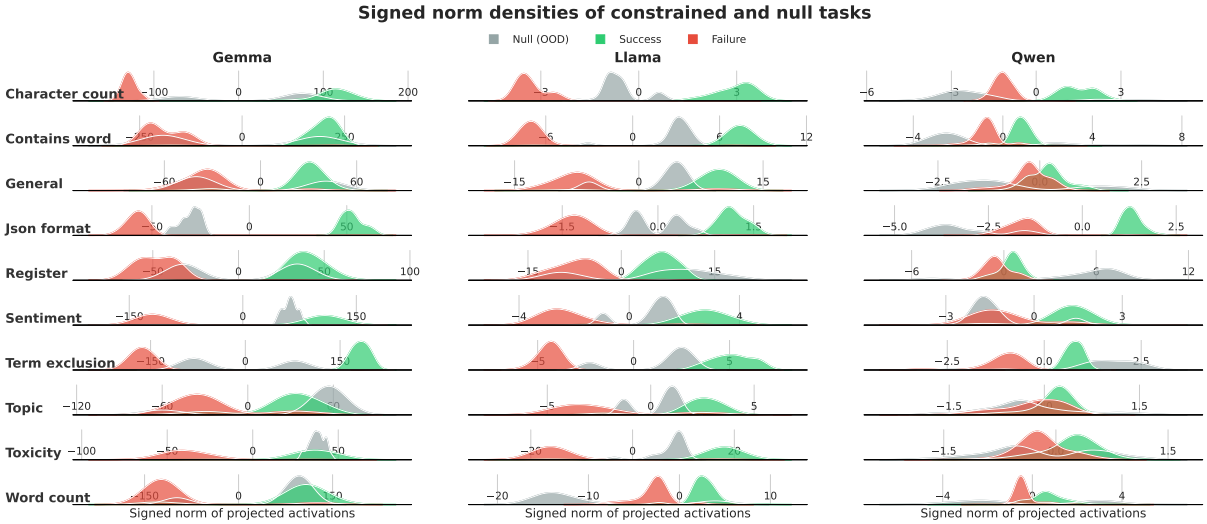


Figure 3: Density distributions of activation projections onto task-specific rowspaces. Green: successful constraint satisfaction; red: constraint violations; gray: null task baseline (open-ended generation without chat templates). Each subplot represents projections onto a different task's rowspace extracted via INLP from the best-performing linear probe.

topic depends on sentiment and term exclusion. For Qwen, register draws from diverse sources. Critically, the *general* probe's column shows minimal impact on most *specialist* probes, confirming that the information it captures is not necessary for individual task performance. Tasks exhibiting high interdependence may require compositional skills rather than atomic capabilities. Because probe quality varies across tasks, larger drops could in principle reflect stronger source probes rather than stronger dependence. However, the probes used for INLP extraction achieve accuracies between 0.89 and 1.00 (Table 2), which under our normalization bounds this confound at approximately ± 0.14 in NormDrop—well below the variation in Figure 2b. As a further control, a rank-matched random-subspace baseline produces diffuse, non-specific accuracy drops, in contrast to the sparse, diagonal-dominated pattern of the real INLP ablation (Appendix D, Figure 6). This dissociation confirms that the observed dependencies are specific to the task-relevant directions extracted by INLP rather than an artifact of removing an equal number of arbitrary directions.

5.2 (RQ1) Specificity: constraint signals are model-dependent and often entangled with general features

Figure 3 displays projection intensity distributions (Equation 2)—the ℓ_2 norms of activations X projected onto task-specific rowspaces P_{row} extracted via INLP. Each subplot compares activations from three sources: successful constraint adherence (green), constraint violations (red), and the null task baseline of open-ended generation without chat templates (gray). If constraint satisfaction signals were highly specific, we would expect null task distributions centered near zero with well-separated success/failure distributions displaced from zero. Instead, substantial overlap would indicate probes capture general language modeling features rather than constraint-specific information.

Results show substantial model-dependent variation. Llama exhibits the expected pattern: null task intensities remain low for nearly all tasks except topic and toxicity, while success/failure distributions are well-separated and, indicating constraint-specific encoding. In contrast, Qwen frequently shows null task intensities *exceeding* constrained ones (e.g., character count, term inclusion, JSON format), indicating that the learned rowspaces capture general language modeling features rather than constraint-specific signals. Gemma exhibits an intermediate pattern with comparable intensities across null and constrained tasks for several conditions.

This specificity tracks universality: Llama, with the most constraint-specific signals, also achieves the highest *general* probe performance (Figure 2a), suggesting that abstract, task-invariant encoding improves separability from general features and cross-task transfer.

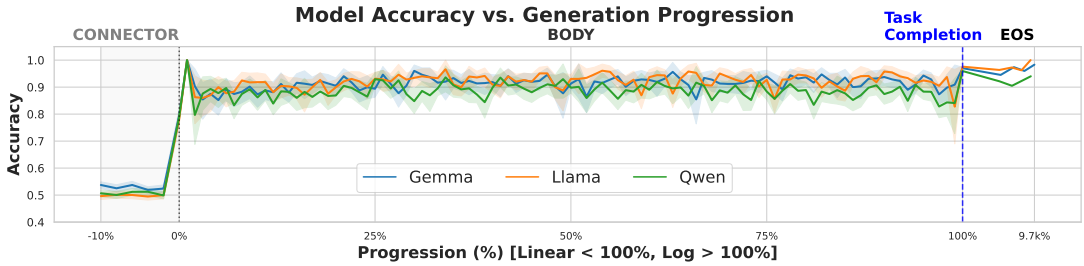


Figure 4: Best linear probe accuracy across generation progression for three models. The x-axis shows progression percentage with a hybrid scale: linear for $\leq 100\%$, logarithmic for $> 100\%$. Shaded regions denote connector positions (left of 0%), generation body (0-100%), and EOS token (spike beyond 100%). Error bands show 95% confidence intervals.

5.3 (RQ2) Temporal dynamics reveal monitoring without pre-generation planning

Figure 4 tracks probe accuracy across generation progression. Across all three models, accuracy remains near baseline (0.5) throughout connector positions, rising sharply only after generation begins (progression $> 0\%$). This indicates minimal pre-generation "planning" signal—constraint satisfaction information emerges dynamically during generation rather than being computed in advance. Accuracy stabilizes at high levels (0.85-0.95) throughout the body generation phase (0-100%), suggesting continuous monitoring of constraint adherence. A notable accuracy peak occurs at the EOS token (visible in the logarithmic tail), indicating a potential final verification phase where the model assesses whether the completed output satisfies the constraint.

5.4 Cross-task similarity with PWCCA

To characterize the compositional structure of constraint satisfaction, we measure cross-task information similarity using Projection Weighted Canonical Correlation Analysis (PWCCA) (Morcos et al., 2018) on task-specific rowspaces extracted via INLP from the best linear probes.

We construct a shared activation pool X_{universe} by concatenating the test set activations from the layer and scope used by each task's best-performing linear probe. This ensures each task contributes data from its optimal representation layer.

For each task pair (i, j), we project X_{universe} onto their respective rowspaces and reconstruct:

$$\text{View}_i = (X_{\text{universe}} \cdot (P^i_{\text{row}})^T) \cdot P^i_{\text{row}} \quad (3)$$

$$\text{View}_j = (X_{\text{universe}} \cdot (P^j_{\text{row}})^T) \cdot P^j_{\text{row}} \quad (4)$$

This isolates the information each task's subspace captures from the shared activations. PWCCA then computes variance-weighted canonical correlations between these views, yielding similarity scores $sim_{\text{PWCCA}} \in [0, 1]$ where 0 indicates orthogonal subspaces (distinct information) and 1 indicates identical subspaces (shared information).

We then apply hierarchical clustering using Ward linkage on the resulting distance matrix ($d = 1 - sim_{\text{PWCCA}}$) to reveal task groupings based on shared representational structure.

Figure 5 reveals substantial diversity in task representations: no tasks exhibit very low distances, indicating that each constraint relies on largely distinct subspaces. Moreover, the organizational structure varies considerably across models, showing that different architectures encode constraint satisfaction information through different compositional strategies. One might ask whether these groupings merely reflect tasks being represented at similar network depths, since each contributes activations from its best-performing layer. The layer of peak encoding is itself a meaningful task property rather than a nuisance variable: related tasks rely on similar mechanisms that the model consolidates at similar depths. Moreover, several tasks grouped together in Figure 5 draw their representations from *different* layers, showing the analysis does not trivially cluster by depth.

For Gemma, tasks form two primary groups. Lexical control (term inclusion, term exclusion) groups with semantic tasks (topic, toxicity), while structural formatting tasks (character count, JSON format, word count) form a separate group alongside register and sentiment. For Llama, character count stands isolated with a unique representational signature; the remaining tasks divide into two groups: one containing register, JSON for-

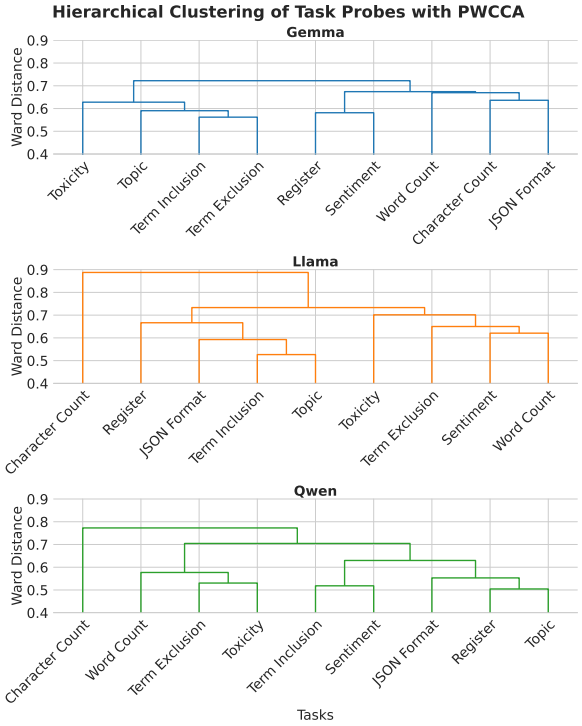


Figure 5: Hierarchical clustering dendrograms of tasks based on PWCCA similarity computed from rowspaces of best linear probes. Each dendrogram shows one model with tasks clustered using Ward linkage.

mat, term inclusion, and topic (lexical patterns and structure), and another grouping toxicity, term exclusion, sentiment, and word count (content evaluation and filtering). For Qwen, term exclusion pairs with toxicity (negative filtering), register with topic (thematic coherence), and term inclusion with sentiment (positive constraints and affect), while structural tasks again form their own group.

Despite model-specific variation, certain task pairings recur (e.g. term exclusion with toxicity, and structural formatting tasks together) suggesting fundamental skill categories that models consistently encode. However, the divergent organizational patterns across models indicate that while the skill types are consistent, their composition and reuse strategies are architecture-dependent, supporting our conclusion that constraint satisfaction emerges from flexible skill deployment rather than a universal mechanism.

5.5 Summary of results

Our analysis provides converging evidence against a universal, task-agnostic constraint satisfaction mechanism in instruction-tuned LLMs. Across four complementary experiments, constraint satisfaction consistently operates through compositional skill sharing rather than a dedicated mechanism: general probes trail specialists by selective rather than uniform margins; cross-task transfer is limited and task-clustered; ablation reveals sparse, asymmetric dependencies; and tasks stratify by complexity, with structural tasks emerging early and semantic tasks late. The specificity analysis adds model-dependent variation: Llama exhibits constraint-specific signals while Qwen relies heavily on general language modeling features, and this specificity tracks cross-task generalization ability. PWCCA-based dendrograms show largely distinct task representations organized along model-specific lines rather than universal skill sharing.

Temporal dynamics reveal that constraint satisfaction emerges dynamically during generation rather than through pre-generation planning. Probe accuracy remains at baseline during prompt processing, rises sharply at generation onset, stabilizes during body generation (continuous monitoring), and peaks at EOS (verification). This temporal profile, combined with the compositional structure of task dependencies and model-specific encoding strategies, suggests that instruction-following in LLMs is better characterized as skillful coordination of diverse linguistic capabilities rather than deployment of a single abstract constraint-checking process.

6 Conclusions

This work challenges the assumption that instruction-tuned LLMs implement constraint satisfaction through a unified, task-agnostic mechanism, pointing instead toward a compositional architecture where diverse linguistic skills are coordinated depending on task requirements and model-specific strategies.

The hierarchical emergence of constraints, sparse task dependencies, and dynamic temporal profiles indicate that instruction-following arises from learned coordination patterns rather than dedicated constraint-checking modules, with different architectures developing distinct solutions to the same challenges.

A critical direction for future work is to investigate the geometric organization of the linguistic primitives underlying this coordination—how they are arranged in representational space, what determines their composability, and how their geometric relationships enable or constrain flexible skill deployment.

Limitations

While our study provides converging evidence for the compositional nature of instruction-following, we acknowledge some limitations and present possible extensions for future work.

6.1 Cross-layer representational alignment

Our diagnostic framework compares information across different layers and generation stages. While the residual stream in Transformer architectures is largely additive—allowing representations to share a common geometric space—successive layers can apply non-linear transformations and rotations to the data manifold (Deora et al., 2024). We acknowledge that projecting activations from one layer onto a rowspace extracted from another may introduce alignment noise. However, our use of INLP focuses on the primary directions of variance (the rowspace), which are generally more stable than finer-grained features. Empirically, the fact that our PWCCA-based dendrograms (Figure 5) successfully cluster related tasks across distant layers suggests that these linguistic skills maintain sufficient geometric alignment for our cross-layer analysis to remain valid. Importantly, this cross-layer persistence does not imply that the same neurons remain active throughout the network: task-relevant information is preserved at the level of representational subspaces, even as its encoding is redistributed or transformed across layers.

6.2 Scaling and skill organization

Our experiments across models of varying sizes (0.5B to 8B) suggest that model scale significantly influences representational clarity. However, we do not interpret this as the emergence of a more "unified" or "abstract" constraint-satisfaction signal in larger models. Instead, our results suggest that as models scale, they develop a more comprehensive and sophisticated organization of the diverse linguistic skills required for adherence. In larger models like Llama-3.1-8B, these task-specific components appear to be more effectively disentangled from general language modeling noise (Figure 3). This implies that scale improves the *coordination* and *precision* of skill deployment rather than shifting the model toward a distinct, task-invariant cognitive mechanism.

6.3 Linguistic coverage and multi-constraint tasks

We evaluated nine tasks spanning four linguistic dimensions. While these represent a range of complexities, they do not exhaust the full taxonomy of human-AI interaction. Future work should investigate multi-constraint prompts (e.g., "Write a formal email [style] under 50 words [structural] without using the word 'meeting' [lexical]"). Such tasks would allow for a deeper investigation into how the compositional coordination we identified handles competing objectives and potential representational bottlenecks.

6.4 From diagnosis to rowspace steering

Finally, while our causal ablation and temporal analysis provide a look into the "when" and "where" of instruction following, they remain primarily diagnostic. Recent work (He et al., 2025; Stolfo et al., 2025) has successfully demonstrated that activation steering—intervening on internal representations—can significantly recover model performance at inference time. While these approaches often rely on contrastive pairs or sparse autoencoders, our findings suggest that the information-rich rowspaces extracted through our framework could provide a robust alternative for computing steering vectors. Exploring how rowspace-based interventions can guide a model's "skill coordination" in real-time is a promising direction for future research.

References

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Guillaume Alain and Yoshua Bengio. 2017. Understanding intermediate layers using linear classifier probes. In *The Fifth International Conference on Learning Representations*.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. *CoRR*, abs/1903.04561.

Puneesh Deora, Rouzbeh Ghaderi, Hossein Taheri, and Christos Thrampoulidis. 2024. On the optimization and generalization of multi-head attention. *Transactions on Machine Learning Research*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

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

Zirui He, Haiyan Zhao, Yiran Qiao, Fan Yang, Ali Payani, Jing Ma, and Mengnan Du. 2025. Saif: A sparse autoencoder framework for interpreting and steering instruction following of language models. *arXiv preprint arXiv:2502.11356*.

Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley You Ren, Andrew Miller, Udhyakumar Nallasamy, and Jaya Narain. 2025. Do LLMs "know" internally when they follow instructions? In *The Thirteenth International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2023. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations*.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. Followbench: A multi-level fine-grained constraints following benchmark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, and 17 others. 2022. Language Models (Mostly) Know What They Know. *arXiv e-prints*, arXiv:2207.05221.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. *Proceedings of the 7th ACM conference on Recommender systems*.

Julian Minder, Clément Dumas, Stewart Slocum, and Neel Nanda. 2025. Narrow finetuning leaves clearly readable traces in the activation differences. In *Mechanistic Interpretability Workshop at NeurIPS 2025*.

Ari S. Morcos, Maithra Raghu, and Samy Bengio. 2018. Insights on representational similarity in neural networks with canonical correlation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 5732–5741, Red Hook, NY, USA. Curran Associates Inc.

Yanzhao Qin, Tao Zhang, Yanjun Shen, Wenjing Luo, Haoze Sun, Yan Zhang, Yujing Qiao, Weipeng Chen, Zenan Zhou, Wentao Zhang, and 1 others. 2024a. Sysbench: Can large language models follow system messages? *arXiv preprint arXiv:2408.10943*.

Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024b. Infobench: Evaluating instruction following ability in large language models. *arXiv preprint arXiv:2401.03601*.

Yujia Qin, Cheng Qian, Jing Yi, Weize Chen, Yankai Lin, Xu Han, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2022. Exploring mode connectivity for pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6726–6746, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang. 2023. Coedit: Text editing by task-specific instruction tuning.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. 2020. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7237–7256. Association for Computational Linguistics.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. Improving instruction-following in language models through activation steering. In *The Thirteenth International Conference on Learning Representations*.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Xuansheng Wu, Wenlin Yao, Jianshu Chen, Xiaoman Pan, Xiaoyang Wang, Ninghao Liu, and Dong Yu. 2024. From language modeling to instruction following: Understanding the behavior shift in llms after instruction tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2341–2369.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Yichu Zhou and Vivek Srikumar. 2022. A closer look at how fine-tuning changes BERT. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1046–1061, Dublin, Ireland. Association for Computational Linguistics.

A Task Datasets

In this section, we provide implementation details for the datasets used in our experiments.

A.1 General Methodology

Data Collection For each task, we designed five prompt templates systematically combined with task-specific options. We generated responses using four LLMs (Llama-3.1-8B, Gemma-2-2b, Qwen1.5-7B, and Mistral-7B), supplemented by natural sentences from various corpora. Since models are autoregressive, we gathered a static set of responses and extracted activations via a single forward pass. All the data served the training process of our probes using a standard train–validation–test split (70/15/15). The specific provenance of each task's data — whether synthetically generated, drawn from an existing corpus, or a combination of both — is detailed per task in Table 3., where we cite the originating publication for each external dataset.

Labeling and Verification Responses were labeled as correct (label 1) or incorrect (label 0). Verification was performed using ground-truth metadata from source datasets or task-specific heuristics (e.g., Python `len()` or regex patterns). Detailed task parameters and verification logic are summarized in Table 3.

Negative Sampling To ensure a 50/50 balance and high-quality negatives, we avoided using ungrammatical noise. Instead, we employed a *cross-option swapping* strategy: the negative samples for a specific prompt were drawn from correct responses to *different* options within the same task. For example, a correct 20-word sentence serves as a label 0 example for a 5-word prompt. This ensures that negative pairs remain fluent, forcing the classifier to distinguish constraint compliance rather than identifying broken text.

B Framework architecture

This section details the design principles and configuration schema of the framework used for the probing experiments. To ensure reproducibility and scalability, the framework is built on a "separation of concerns" principle, decoupling model-specific formatting, task logic, and experimental orchestration.

B.1 Configuration-driven approach

The system architecture is entirely configuration-driven, allowing researchers to scale experiments across dozens of models and tasks without modifying the core codebase. This is achieved through three abstraction layers:

B.1.1 Model configuration

To integrate a new Large Language Model (LLM), a JSON or YAML configuration file must be provided. This metadata ensures the model is probed in its intended instruction-following state and allows the system to precisely map internal activations:

- **`name`**: The HuggingFace identifier used to load the model and tokenizer (e.g., `meta-llama/Llama-3.1-8B-Instruct`).

Task	Data Source	All Requested Options	Verification
Char Count	LLM / **C4** (Raffel et al., 2019) https://huggingface.co/datasets/allenai/c4	30, 50, 100, 140, 200, 280	`len(s.strip())`
Word Count	LLM / **C4** (Raffel et al., 2019)	2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21	`re.findall()`
Term Inclusion	LLM / **C4** (Raffel et al., 2019)	dog, cat, computer, the, and, is, time, people, game, company, teaching, compete, ability, recipe, smoker, function	`word in s`
Term Exclusion	LLM / **C4** (Raffel et al., 2019)	dog, cat, computer, the, and, is, time, people, game, company, teaching, compete, ability, recipe, smoker, function	`word not in s`
JSON format	LLM	an animal, a vehicle, a fruit, a country, a profession, a musical instrument, a building, a sport, a technology, a historical event	`json.loads()`
Topic	**AG News**	world, sports, business, technology	Dataset labels
Sentiment	**IMDB** (Maas et al., 2011) / **Amazon** (McAuley and Leskovec, 2013) https://huggingface.co/datasets/mteb/amazon_polarity	negative, positive	Dataset labels
Register	**CoEdIT** (Raheja et al., 2023) https://huggingface.co/datasets/grammarly/coedit	formal, informal	Dataset labels
Toxicity	**Civil Comments** (Borkan et al., 2019) https://huggingface.co/datasets/google/civil_comments	toxic, non-toxic	Dataset labels

Table 3: Detailed Task Parameters: Exhaustive list of target options and data sources.

- **`prompt_template`**: The exact string structure required by the model's chat interface, utilizing a `PROMPT` placeholder to maintain consistency with the model's pre-training/fine-tuning format.

- **`response_connector`**: The specific string that transitions the prompt into the model's generation (e.g., `<start_of_turn>model\n`). This is critical for locating the precise boundary between input tokens and output activations.

- **`end_of_turn_token_id/token`**: The markers for the model's EOT (End of Turn), used to mask or target the final state of a generation for probing.

B.1.2 Task configuration and logic classes

Tasks are defined by their data requirements and labeling logic. While most parameters are data-driven, specific tasks utilize a **Task Logic Class**—a "plugin" architecture that allows for dynamic code execution to verify model outputs. A task configuration includes:

- **`logic_class`**: A reference to a Python class implementing standardized interfaces. This class defines the "success" criteria (e.g., verifying a valid JSON object or detecting a keyword) and calculates the "completion index" (the specific token where an instruction is fulfilled).

- **`data_sources`**: A multi-modal source list including parameters for synthetic **LLM Generation** (templates, temperature, etc.) or references to **External Datasets** (e.g., C4) used as natural language anchors.

- **`prompts`**: A collection of instructional variations and the specific categories (`requested_options`) the task covers to ensure linguistic diversity.

B.1.3 Experimental orchestration

The experiment configuration ties models and tasks together into a unified execution suite. This layer defines the scope of the run:

- **`type`**: Determines whether the run is a `single_task` execution or an `all_tasks` suite for cross-task generalizability analysis.
- **`model_path`** and **`task_path`**: Pointers to the specific metadata files described above.
- **`output_dir`**: A standardized directory structure for storing probe weights, accuracy statistics, and activation visualizations.

B.2 Extensibility and robustness

By utilizing dynamic importing for Logic Classes, the framework remains highly extensible. Adding a new task does not require modifying existing scripts, which mitigates regression risks. This modularity ensures that the probing pipeline remains agnostic to both the underlying architecture of the LLM and the semantic nature of the task being evaluated.

C Confusion matrices

The probe accuracies reported in Table 2 could in principle be inflated by a degenerate strategy that exploits class imbalance. To rule this out, we report the per-class recall of the best probe for each task and model: the **True Positive Rate** (TPR), the fraction of genuinely compliant outputs correctly classified as compliant, and the **True Negative Rate** (TNR), the fraction of genuinely non-compliant outputs correctly classified as non-compliant. A probe that simply predicts the majority class would yield near-zero recall on the minority class, producing a large TPR/TNR asymmetry.

For each task and model, we extract the test-set predictions of the best linear probe at the EOS token, construct the 2×2 confusion matrix, and compute TPR and TNR from its rows. Table 4 reports these values across all nine tasks and three models. TPR and TNR are approximately equal across all conditions, with no systematic tendency toward false positives (over-predicting compliance) or false negatives (over-predicting non-compliance). This confirms that probe accuracy reflects genuine sensitivity to constraint adherence rather than an artifact of label imbalance.

Task	Gemma		Llama		Qwen	
	TPR	TNR	TPR	TNR	TPR	TNR
Character Count	0.99	1.00	0.99	1.00	0.99	1.00
Term Exclusion	1.00	0.99	1.00	1.00	1.00	0.99
JSON Format	1.00	0.99	1.00	1.00	1.00	0.99
Word Count	0.95	0.95	0.98	0.97	0.92	0.94
Register	0.99	0.99	0.99	0.99	1.00	1.00
Term Inclusion	0.99	0.98	0.99	0.98	0.94	0.96
Toxicity	0.97	0.97	0.98	0.98	0.89	0.87
Sentiment	0.98	0.97	0.98	0.99	0.97	0.94
Topic	0.93	0.84	0.92	0.89	0.82	0.78

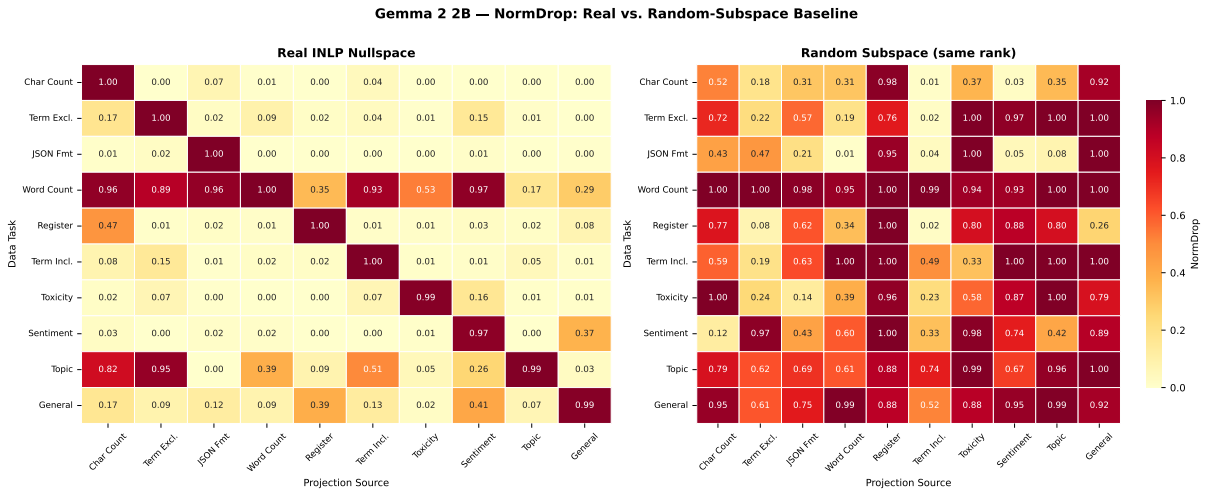
Table 4: Per-class recall for the best probe of each task and model. **TPR**: recall for the compliant class (label 1). **TNR**: recall for the non-compliant class (label 0). Values near 1.0 on both columns indicate balanced probes with no systematic bias toward false positives or false negatives.

D NormDrop robustness: random-subspace baseline

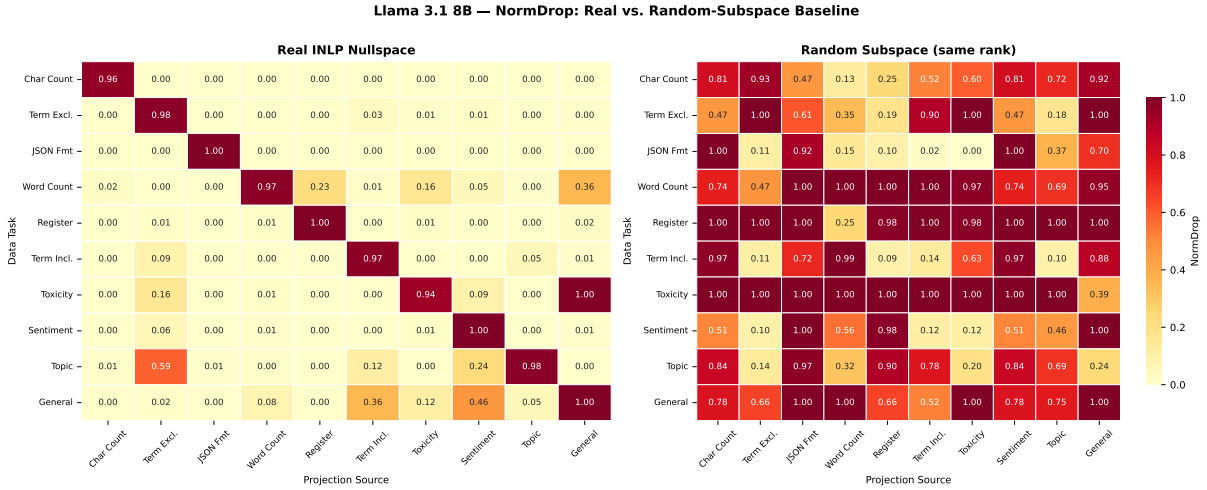
A potential confound in the cross-task causal ablation (§3.2, Figure 2b) is that removing *any* set of directions from the activation space—not only task-relevant ones—could degrade probe accuracy simply because fewer dimensions remain available to the classifier. Under this account, the Normalized Accuracy Drop (NormDrop, Equation 1) would be an artifact of the dimensionality reduction induced by nullspace projection rather than evidence of shared representational structure between tasks.

To rule out this confound, we compare against a random-subspace baseline of matched rank. For each source task j, we replace its INLP nullspace projection P_{null}^{j} with a random orthonormal projection of identical rank: given the rank r_j of the real nullspace, we sample a matrix $R \in \mathbb{R}^{d \times r_j}$ with i.i.d. Gaussian entries, take the orthonormal basis Q from its QR decomposition, and form the random nullspace projection $P_{\text{null}}^{\text{rand}} = I - QQ^{\top}$. We then recompute NormDrop for all task pairs, keeping the data, the target probe, and the normalization identical to the main analysis. Crucially, this control isolates *what* is removed from *how much* is removed: because the random subspace has the same rank as the real nullspace, any difference between the two conditions cannot be attributed to dimensionality loss alone. If NormDrop merely reflected the projection operation, the two conditions would yield comparable, similarly distributed drops; a divergence in their structure would instead confirm that the real drops are specific to the task-relevant directions extracted by INLP.

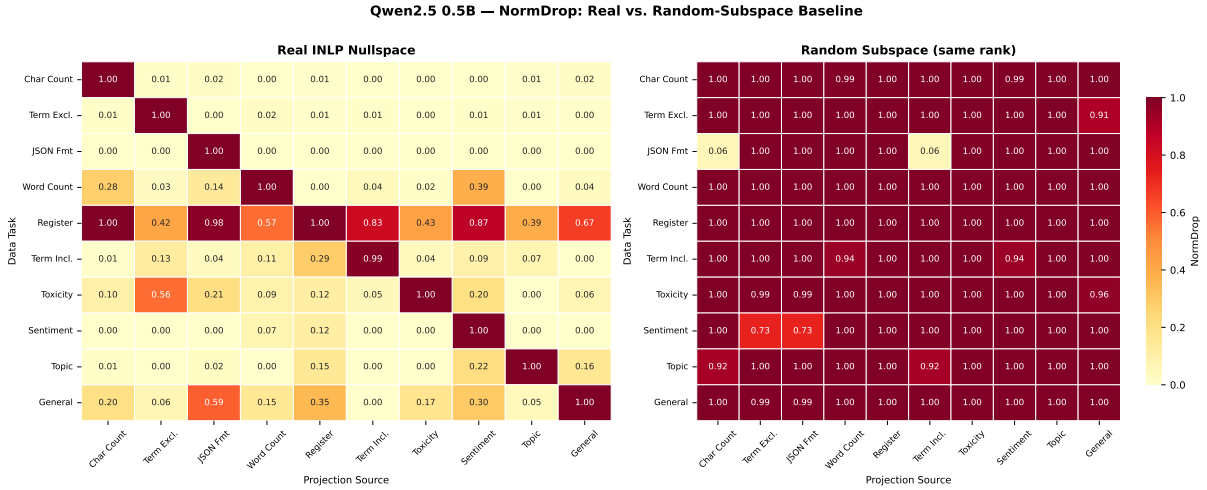
Figure 6 reports the real NormDrop matrices alongside the random-subspace baseline for all three models at the EOS scope. The two conditions produce categorically different signatures. The real INLP ablation is sparse and diagonal-dominated: removing a task's own subspace collapses its probe to chance (diagonal ≈ 1.00), while off-diagonal drops are mostly low and concentrated on a small number of interpretable cross-task dependencies. The random baseline, in contrast, produces diffuse, non-specific degradation that is elevated across the matrix. This dissociation is the relevant evidence: a generic effect of removing r_j arbitrary directions would degrade probes indiscriminately (as the random baseline does), whereas the real procedure degrades a target probe only when the removed subspace carries information that probe relies on. The contrast is sharpest for Qwen and Llama, where random removal is broadly destructive but real removal is highly localized; for Gemma the dissociation is weaker but the same qualitative pattern holds. We note that Word Count is sensitive to subspace removal under both conditions across models, indicating that this particular probe is more fragile to dimensionality loss in general; the conclusion therefore rests on the aggregate sparsity of the off-diagonal structure rather than on every individual cell.



(a) Gemma



(b) Llama



(c) Qwen

Figure 6: Normalized accuracy drop (NormDrop) for the real INLP nullspace ablation vs. random-projection baseline across all three models (EOS scope). Each cell (i, j) shows the NormDrop of probe i after removing probe j's information via nullspace projection or random projection. The real ablation (left panels) is sparse and diagonal-dominated; the rank-matched random baseline (right panels) produces diffuse, non-specific drops, confirming that the real dependencies are specific to the task-relevant directions extracted by INLP rather than an artifact of removing an equal number of arbitrary directions.