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 consistently underperform task-specific specialists, indicating limited representational sharing. 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 fundamental architectural question: Is instruction following a monolithic process where the skill of adhering to a constraint is learned anew for each task? Or do models develop a more general, abstract representation of "rule-following" that exists independently of the task at hand?

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 structure of cross-task similarity.

The remainder of this paper proceeds as follows. Section 2 reviews related work and our contributions. Section 3 describes our diagnostic probing framework and analysis methods. Section 4 details our experimental setup. Section 5 presents our findings. Section 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—word counts, keyword inclusion, formatting rules—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 analyze the impact of instruction tuning on LLMs, researchers increasingly probe models' internal states (Alain and Bengio, 2017; Belinkov, 2022), evolving from mapping surface syntax (Tenney et al., 2019) to detecting 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.

2.1 Contributions

Our research extends and refines the frameworks by Heo et al. (2025) and Stolfo et al. (2025). While these studies utilize the IFEval benchmark to identify binary success/failure states, we address two critical limitations in their design. First, whereas IFEval's artificial split between base queries and instructions often results in labels that ignore total prompt fidelity, we evaluate responses based on the entire input as a single task. Second, while prior work relies on the relatively simple heuristics of IFEval, we test tasks covering a broader range of linguistic complexity levels.

Finally, while Stolfo et al. (2025) suggest that first-token representations are sufficient for predicting adherence, we investigate the longitudinal dynamics throughout the generation process. Our work makes the following contributions:

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

The core of our approach is diagnostic probing (Alain and Bengio, 2017), where we train simple supervised linear and non-linear classifiers to predict a specific property from a model's internal activations. For our study, we train probes to solve a binary classification: predicting whether a model's response successfully adheres to a given prompt's task (*Success*) or fails to do so (*Failure*). In the following, we provide a description of the evaluated tasks and the methods employed to investigate our research questions.

3.1 Tasks

We designed nine tasks to probe linguistic complexities across four dimensions: structural (e.g., word counts), lexical (word inclusion), semantic (topic/sentiment), and stylistic (formality). As

[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.

shown in Table 1, our datasets use fluent but "incorrect" responses to force a distinction between constraint adherence and mere linguistic well-formedness. Each task dataset was constructed by pairing multiple prompt templates with task-specific options, utilizing a balanced mix of LLM-generated and existing datasets. See Appendix A for comprehensive details on the labeling and data gathering process.

3.2 Universality (RQ1)

We investigate the extent to which a shared, task-agnostic representation for constraint satisfaction exists using three distinct methods.

***General* vs. *specialist* probes** We compare the performance of task-specific *specialist* probes against a single *general* probe trained on data aggregated from all tasks. A *general* probe that performs comparably to *specialist* probes across all tasks would be initial evidence of a shared, task-agnostic representation.

Out-of-Distribution (OOD) generalization We directly test how well a representation learned for one task transfers to others. For each *specialist* probe, we evaluate its accuracy on the datasets from all other, unseen tasks. If the *general* probe performs well across all tasks and this accuracy exceeds the OOD accuracy of individual *specialist* probes, it would provide strong evidence for a universal representation. Low cross-task accuracy would indicate that the representations are task-specific.

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} that a probe for a source task B relies on. We then project activations from a target task A onto P^B_{null} and measure the impact on the target probe trained on A using the **normalized accuracy drop**:

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

A significant accuracy drop (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 activations from constrained tasks (formatted with chat templates) to a *null* task baseline defined as open-ended generation without chat templates (e.g. "What a beautiful"). Using the rowspace P_{row} extracted via INLP from our best linear probes, we calculate 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.

3.4 Temporal dynamics analysis (RQ2)

We differentiate our analysis based on the position of the token from which we extract activations. Specifically, when training probes and extracting rowspaces, we distinguish between three distinct positions in the generation sequence: (1) *connectors*, special tokens appearing after the end of the user prompt but before the start of the model's response; (2) *body*, tokens that form the main content of the generated response; (3) *EOS*, the final end-of-sequence token indicating the completion of the assistant's turn. For each task and scope (connector, body, EOS), we train both linear and non-linear probes on activations extracted from two different model components: after the attention mechanism and after the MLP block.

4 Experimental setup

We detail the specific models, data, and implementation choices used to execute the methodology described. All experiments were conducted on one NVIDIA H100 NVL GPU with 94GB of memory.

Models We conduct experiments 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). The models have 32, 26, and 24 layers with hidden dimensions of 4096, 2304, and 896 respectively.

Probing and INLP implementation We trained various probe architectures (logistic regression, stochastic gradient descent, random forests, k-nearest neighbors, and multilayer perceptrons) across three runs with different random seeds, and then we selected the best-performing linear and non-linear models: logistic regression and a single-hidden-layer MLP with 128 neurons and ReLU activation. For INLP, we apply an iterative procedure to the best-performing linear probes: at each step, we train a linear classifier, record its weight vector (w_i), and project the activation dataset into the nullspace of this vector. Iterations halt when test accuracy falls below 0.55. This yields a projection matrix P_{null} mapping to the nullspace intersection, with the complementary matrix $P_{row} = I - P_{null}$ projecting onto the information-rich rowspace.

5 Results

We present our findings organized around two research questions: the scope of constraint satisfaction representations (RQ1) and their temporal dynamics (RQ2), followed by an analysis of cross-task information similarity.

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

General* probes underperform *specialists Table 2 reports mean accuracy for each model-task combination, aggregated across all probes, layers, and scopes. The *general* probe achieves comparable but lower performance than *specialists*. This suggests that a single, task-agnostic representation may not be sufficient to capture constraint satisfaction.

Tasks stratify by complexity across layers Figure 1 reveals distinct emergence patterns across network depth for attention (left) and MLP (right) streams, with accuracy averaged over best linear and nonlinear probes. 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 in later layers. 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. Critically, the *general* probe fails to reach the peak accuracy of *specialist* probes across both attention and MLP streams for all three models, further undermining the hypothesis of a universal constraint satisfaction mechanism.

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.

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. The *general* probe (first column) fails to consistently outperform *specialists* on held-out tasks. Instead, we observe task-specific composition: some *specialist* probes transfer well to related tasks (e.g., Llama's topic probe achieves 0.78-0.87 accuracy on sentiment and term exclusion), while others remain highly specialized. This pattern suggests that rather than a universal mechanism, models encode constraint satisfaction through a composition of intermediate-level skills shared across subsets of related tasks.

Causal ablation shows sparse, asymmetric task dependencies To identify causal information flow between tasks, we measure how much a target probe i's accuracy drops when evaluated on

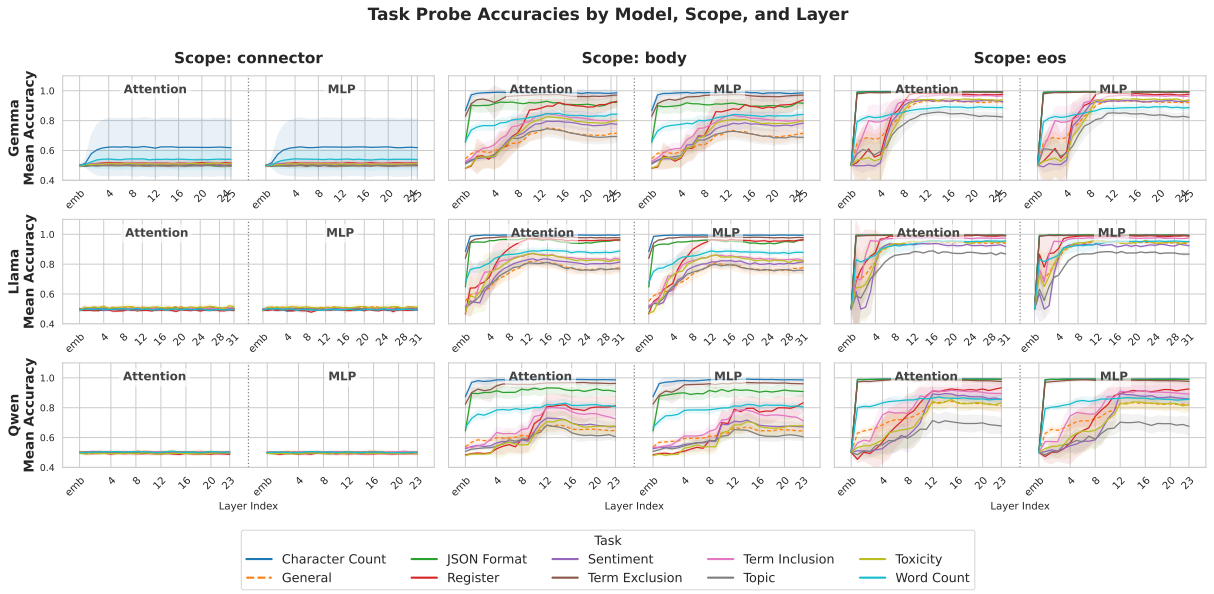


Figure 1: Probe accuracy across network layers for attention (left) and MLP (right) streams, separated by scope. Accuracy values are averaged over the best-performing linear and nonlinear probe for each condition. Colored lines: specialist probes; dashed line: general probe.

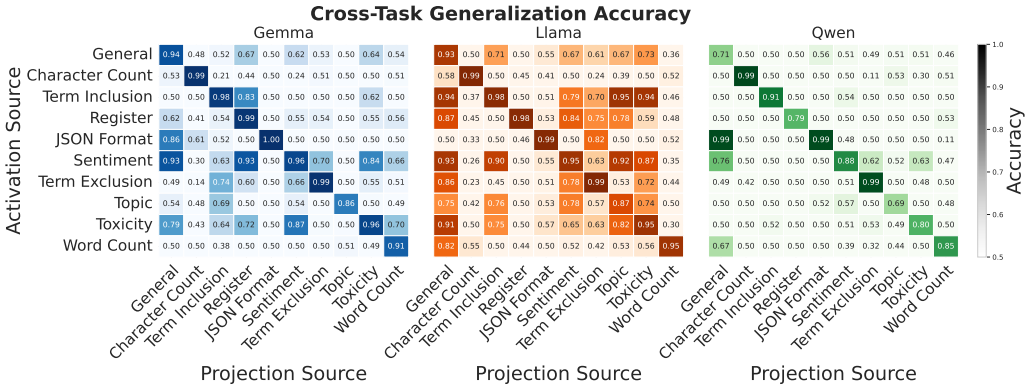
activations X^i with source probe j's information removed via nullspace projection: $(X^i \cdot (P^j_{\text{null}})^T) \cdot P^j_{\text{null}}$. Figure 2b shows these normalized accuracy drops computed using Equation 1: each cell (i, j) represents the performance degradation of task i's probe (rows) after removing task j's information (columns). High values (darker colors) indicate strong causal dependence—removing the source task's information significantly impairs the target probe. Across all three models, we observe sparse, 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, 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.

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

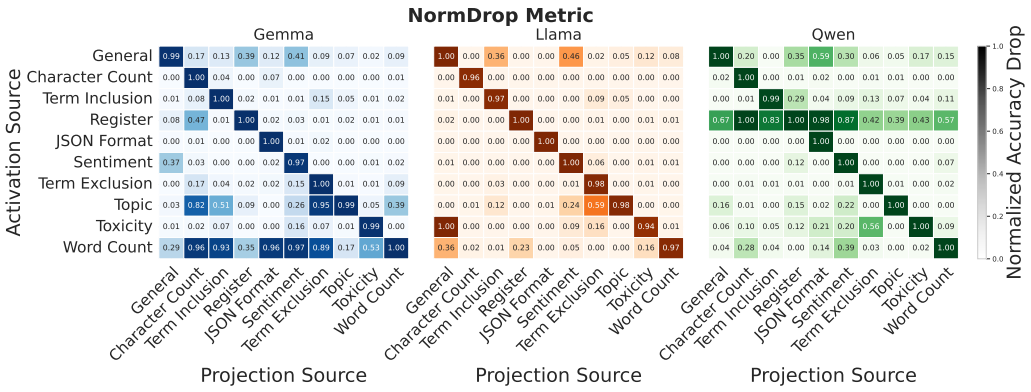
Figure 3 displays projection intensity distributions (Equation 2)—the ℓ_2 norms of activations X projected onto task-specific rowspaces P_{row} extracted via INLP. Each subplot compares activations from three sources: successful constraint adherence (green), constraint violations (red), and the null 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.

The results reveal substantial model-dependent variation. For Llama, most tasks show the expected pattern: null task intensities remain low (concentrated near zero) for nearly all tasks except topic and toxicity, while success/failure distributions are well-separated and show higher magnitudes. This suggests Llama encodes relatively constraint-specific information. In contrast, Qwen frequently shows null task intensities *exceeding* constrained task intensities (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.

The correlation between specificity and universality is notable: Llama, which shows the most constraint-specific signals (low null task norms), also achieves the highest *general* probe perfor-



(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.

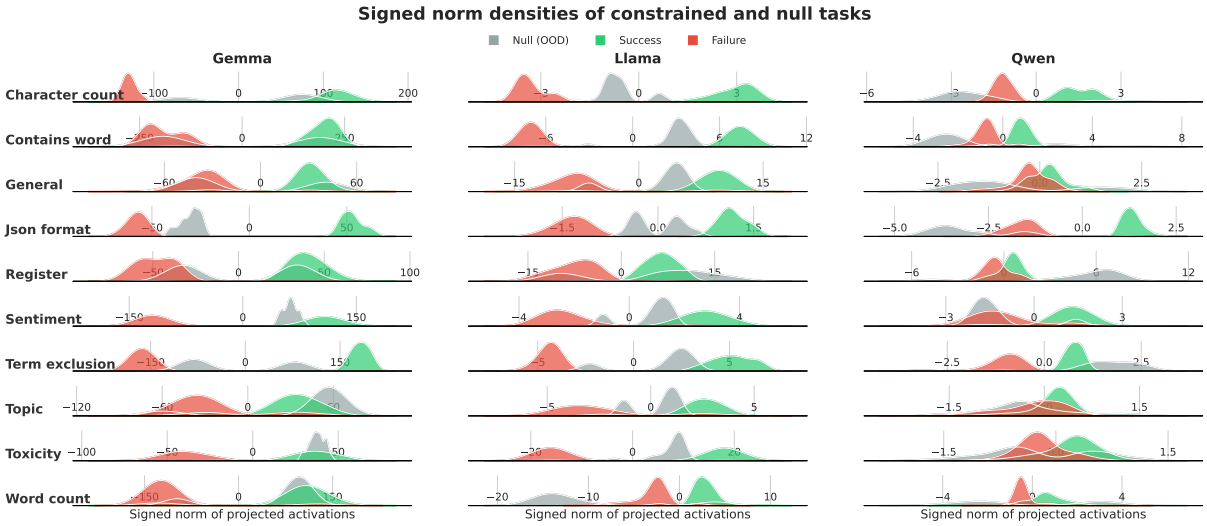


Figure 3: Density distributions of activation projections onto task-specific rowspaces. Green: successful constraint satisfaction; red: constraint violations; gray: null task baseline (open-ended generation without chat templates). Each subplot represents projections onto a different task's rowspace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

mance across tasks (see Figure 2a). This suggests that when constraint satisfaction information is encoded in a more abstract, task-invariant manner, it becomes more separable from general language modeling features. Conversely, when probes heavily rely on low-level linguistic features (high null

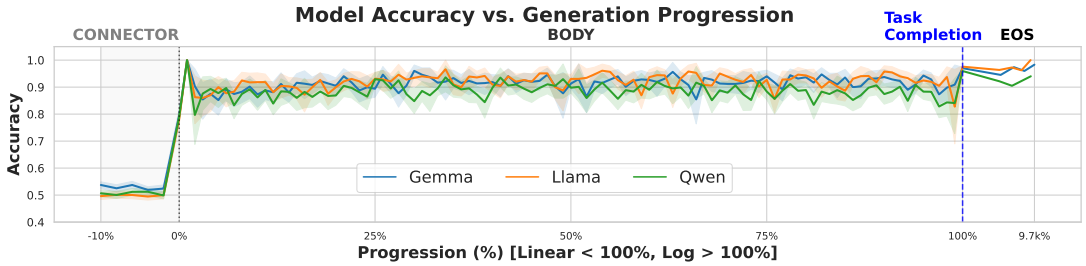


Figure 4: Best linear probe accuracy across generation progression for three models. The x-axis shows progression percentage with a hybrid scale: linear for $\leq 100\%$, logarithmic for $> 100\%$. Shaded regions denote connector positions (left of 0%), generation body (0-100%), and EOS token (spike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

task norms), cross-task transfer suffers.

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

Figure 4 tracks probe accuracy as a function of generation progression, measured as the percentage of tokens generated relative to the final response length. 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.

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 format, 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

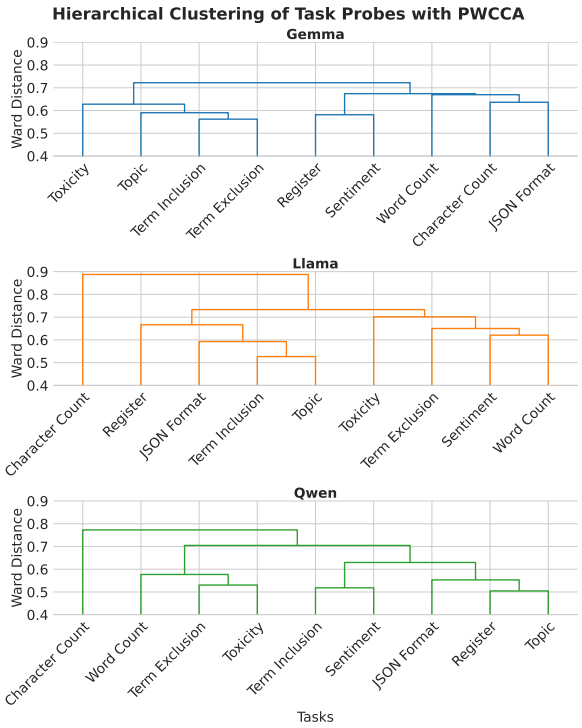


Figure 5: Hierarchical clustering dendrograms of tasks based on PWCCA similarity computed from rowspaces of best linear probes. Each dendrogram shows one model with tasks clustered using Ward linkage.

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 comprehensive analysis provides converging evidence against a universal, task-agnostic constraint satisfaction mechanism in instruction-tuned LLMs. Across four complementary experiments, we consistently find that constraint satisfaction operates through compositional skill sharing rather than a dedicated mechanism. General probes underperform specialists; cross-task transfer is limited and task-clustered; ablation reveals sparse, asymmetric dependencies; and tasks stratify by complexity with early-emerging structural tasks and late-emerging semantic tasks. The specificity analysis shows model-dependent variation: Llama exhibits constraint-specific signals while Qwen relies heavily on general language modeling features, with this specificity correlating with cross-task generalization ability. PWCCA-based dendrograms show largely distinct task representations with model-specific organizational patterns 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. Instead, our evidence points toward a fundamentally compositional architecture where diverse linguistic skills are coordinated depending on task requirements and model-specific organizational strategies.

The hierarchical emergence of constraints (structural in early layers, semantic in later layers) combined with sparse task dependencies and dynamic temporal profiles, indicates that instruction-following arises from learned coordination patterns rather than dedicated constraint-checking modules. Substantial variation across models in how constraints are encoded and composed shows that different architectures develop distinct solutions to the same instruction-following challenges.

These findings open important directions for future research. A critical next step is to investigate the geometric organization of the fundamental linguistic primitives underlying constraint satisfaction—understanding how these skills are arranged in representational space, what determines their composability, and how their geometric relationships enable or constrain flexible coordination.

Limitations

While our study provides converging evidence for the compositional nature of instruction-following, we acknowledge several limitations regarding our methodology and experimental scope.

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.

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.

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.

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

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.

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*.

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*.

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*.

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.

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*.

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).

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`).
- **`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.

Task	Data Source	All Requested Options	Verification
Char Count	LLM / **C4** https://huggingface.co/datasets/allenai/c4	30, 50, 100, 140, 200, 280	`len(s.strip())`
Word Count	LLM / **C4**	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**	dog, cat, computer, the, and, is, time, people, game, company, teaching, compete, ability, recipe, smoker, function	`word in s`
Term Exclusion	LLM / **C4**	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 / Amazon** https://huggingface.co/datasets/amazon_polarity	negative, positive	Dataset labels
Register	**CoEdIT** https://huggingface.co/datasets/grammarly/coedit	formal, informal	Dataset labels
Toxicity	**Civil Comments** 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.