Who is In Charge? Dissecting Role Conflicts in LLM Instruction Following

LLMs are expected to respect hierarchical instructions: system prompts should take precedence over user instructions. Yet [1] shows the opposite: models often ignore system–user priority while obeying social cues such as authority, expertise, or consensus. This mismatch creates safety risks: prompt injections framed as 'authoritative' or even seemingly harmless user requests can bypass system safeguards and override critical constraints. Our paper asks: ***How do models internally represent and resolve system-user conflicts for different hierarchical cues? Can internal interventions restore respect for system authority without retraining?***

Together, our results extend [1] by moving from behavioral observations to mechanistic evidence. Key takeaways include: 1) **Conflict signals exist internally:** The model reliably encodes conflicts early in processing, even if it does not respect the hierarchy in output. 2) **Stronger detection for system–user, but resolution favors social cues:** In system–user cases, conflict detection is more evident but resolution remains inconsistent. By contrast, in social cue conflicts, detection is weaker, yet resolution is consistent and always favors the socially dominant role. 3) **Steering boosts compliance indiscriminately:** Our steering-vector interventions cannot distinguish whether an instruction comes from the system or user, but they surprisingly amplify instruction-following.

Our findings show why system–user obedience is weak: the model notices these conflicts but lacks a stable rule to prefer the system. In contrast, social cues act as strong biases that override conflicts and force compliance, explaining why authority prompts work and system rules break easily. This gap highlights why costly additional training like [2], may be required to reliably enforce hierarchy.

To answer our research questions, first, we trained linear probes to detect whether the model would follow the system (primary), the user (secondary), or neither under conflict. Probes show that decision signals are encoded early with high separability. System–user conflicts formed distinct subspaces from social conflicts, confirming [1]'s finding that obedience to system–user hierarchy is much lower than to social cues. This shows the model internally represents system–user conflicts but treats them differently from socially framed ones.

Furthermore, we decomposed next-token logits into contributions from constraint tokens. In system–user conflicts, 17% of cases showed direct opposition between system and user tokens, closely matching [1]'s conflict acknowledgment rate from full output text. In social-consensus conflicts, detection was weaker than in system–user cases, even as obedience was nearly guaranteed, showing bias toward social cues. Our analysis extends [1] by splitting behavioral benchmarking into two steps—conflict detection in activation space and obedience in outputs—and adds new insight not reported in prior works: conflict identification is suppressed under social cues like majority consensus.

Finally, we tested if steering vectors based on consensus vs. system–user activations' difference could strengthen system–user hierarchy obedience. We observed steering reliably shifted outputs across tasks but in a role-agnostic way: the vector successfully boosted whichever instruction the model potentially internally preferred, regardless of whether it came from the system or the user. Furthermore, even random-direction steering boosted compliance too. Our observation aligns with [3], which also builds vectors from instruction tokens to boost attention. The surprising result is that steering strengthened compliance broadly rather than restoring system authority; we discussed potential fixes in future work.

[1] Geng, Yilin, et al. "Control illusion: The failure of instruction hierarchies in large language models." arXiv preprint arXiv:2502.15851 (2025).

[2] Wallace, Eric, et al. "The instruction hierarchy: Training llms to prioritize privileged instructions." arXiv preprint arXiv:2404.13208 (2024).

[3] Guardieiro, Vitoria, et al. "Instruction Following by Boosting Attention of Large Language Models." arXiv preprint arXiv:2506.13734 (2025).