Mapping and Verifying Multi-Dimensional Compliance in AI Constitutions through Constraint Satisfaction

July 22, 2026

Mentors: Dhairya Dalal (MATS Research) and Marco Valentino (University of Sheffield)

Summary

Frontier labs focused on AI safety and alignment are adopting AI constitutions as a means to ensure alignment with codified principles, values, and behavioral specifications. Anthropic's Claude Constitution (Askell et al., 2026) and OpenAI's Model Spec (OpenAI, 2026) serve as technical governance resources in post-training alignment (Bai et al., 2022; Guan et al., 2024). AI constitutions are generally long, complex documents that combine behavioral requirements, guiding principles, authority structures, priorities, and exceptions. Jakkli et al. (2026) found that, despite improvements across model generations, violations persist when models must resolve competing requirements and sources of authority, particularly in multi-turn and agentic task settings. This project aims to build upon that line of research to more fundamentally examine existing AI constitution documents, better understand the multi-dimensional requirements they present, and formally classify common failure modes. Specifically, this project aims to (1) create a classification framework to analyze failures in multi-requirement compliance settings, (2) create a benchmark of use cases derived from the framework to evaluate AI compliance, and (3) explore constraint-satisfaction methods for verifying compliance in such settings.

Project Description

The project broadly has two goals. The first is to better understand the nature of multi-dimensional requirements presented in AI constitutions, and the second is to develop a formal representation and technical solution that considers compliance as a constraint-satisfaction problem.

AI constitutions are complex governance artifacts that consist of heterogeneous requirement objects. For instance, they contain objectives that are broad but directional alongside hard constraints that are highly specific, authority structures that define whose instructions to follow, and exemptions that carve out cases in which otherwise applicable requirements can be ignored (Askell et al., 2026; OpenAI, 2026). Verifying compliance therefore requires navigating ambiguous cases in which multiple satisfaction conditions can apply, creating opportunities for misalignment and unpredictable behavior (Jakkli et al., 2026; Jiang et al., 2024; Wen et al., 2024). The project therefore aims to formally classify requirement objects and specify the norms governing their interactions. Ideally, the framework will draw on established legal theory, formal specification, and deontic logic to identify a pragmatic and auditable classification scheme, which will then be applied to existing AI constitutions to identify recurring structural ambiguities and patterns of conflict.

Secondarily, the project aims to explore whether requirement objects can be translated into formal constraints and if external solvers can verify compliance, identify violations, and surface conflicts among applicable requirements. Constitutional AI relies on supervised and reinforcement-learning phases where AI-generated preference comparisons are employed to produce reward signals (Bai et al., 2022). The guiding preference model is learned from sampled comparisons, and reward models can perform less reliably on unseen prompts and responses when the evaluation distribution shifts (Yang et al., 2024). Multi-constraint benchmarks further demonstrate significant deficiencies when models must satisfy accumulated or composed requirements (Jiang et al., 2024; Wen et al., 2024), while LLM-based evaluators can produce judgments that vary with irrelevant features of the evaluated response (Chen et al., 2024).

In contrast, the project seeks to leverage constraint solvers, which can provide deterministic results for formally specified requirements and support independent auditing. Such an approach would enable an external verifier whose judgments are not dependent on the model being evaluated (i.e., LLMs as judges to evaluate LLMs). Formalizing these requirements as constraints could make distinctions between hard constraints and soft objectives explicit, encode conditions, priorities, and exceptions, and enable solver-based checks for inconsistency or joint unsatisfiability, alongside analysis of whether relevant cases remain under-specified (Bistarelli et al., 1997; Antoniou, 2004; de Moura and Bjørner, 2008). We envision applying this representation to existing AI constitutions to identify combinations of requirements that are jointly unsatisfiable or under-specified. The resulting verification signals could in turn supply per-constraint rewards for constraint-aware reinforcement learning (Qi et al., 2026), support stage-level audits of plans, tool calls, and state transitions (Chen et al., 2025), and enforce formally checkable requirements before consequential actions are executed (Wang et al., 2026). The project will therefore build on related work in symbolic reasoning, constraint-based instruction verification, and runtime policy enforcement (Pan et al., 2023; Su et al., 2026; Wang et al., 2026) to determine which constitutional requirements can be formally represented and which remain dependent on contextual judgment.

In summary, the project aims to do the following:

- **Audit existing AI constitutions.** Examine Anthropic's Claude Constitution and OpenAI's Model Spec for cases in which hard constraints, soft objectives, conditions, priorities, and exceptions produce conflicting, jointly unsatisfiable, or under-specified requirements. *Ideal outcome:* a classification framework that delineates requirement objects and specifies ideal interaction expectations.
- **Evaluate multi-requirement compliance.** Turn the identified cases into test scenarios that require models to determine which requirements apply, resolve priorities, and satisfy the resulting set of conditions. *Ideal outcome:* an initial dataset and benchmark to evaluate model compliance in constructed multi-dimensional requirement scenarios with a focus on AI safety and alignment.
- **Test constraint-based verification.** Translate a bounded subset of requirements and their interactions into formal constraints and compare solver outputs with human judgments. *Ideal outcome:* an engineered proof of concept integrating a constraint solver into a verification pipeline to evaluate AI agents.

Theory of Change

Commercial frontier labs are increasingly both the developers and distributors of the most capable AI models. In contrast to the previous environment where AI models were developed in academic

settings, the leading capable models are no longer open-source, open-weight, nor fully auditable, as training regimes and datasets are occluded as trade secrets and IP. Proprietary models, post-training reinforcement learning, and internal evaluations limit the ability of external actors to assess whether models comply with their stated requirements. Moreover, compliance is often assessed by the same organizations and increasingly by the same class of models whose behavior is being evaluated. Further, sufficiently capable systems may also learn to evade or manipulate these evaluators. External verification is therefore necessary to provide credible safety assurances and serve as early-warning signals for misalignment. Constraint solvers offer an external and deterministic mechanism for identifying contradictions in alignment targets, supporting reproducible external audits, and enforcing formally specified requirements before consequential actions are taken. This could support better-defined constitutions and independently verifiable deployment safeguards while reducing reliance on corporate self-regulation and AI-mediated compliance.

Ideal Candidate

The project is open to applicants from any academic or professional background. Candidates should be self-motivated, communicate clearly, and be able to plan and run experiments independently. An ideal candidate will have experience with Python and independent research, familiarity with standard data analysis methods, and at least one research output that has been shared with a wider audience, such as a published paper, workshop submission, shared-task contribution, LessWrong post, or course paper. However, these are not strict requirements, and we will consider highly motivated and agentic candidates who can demonstrate the ability to conduct high-quality research. We expect to support one or two mentees.

Work Test

1. Describe one project you are proud of, your role in the project, and what surprised you most while working on it. *Maximum 300 words.*

2. Share a link to a GitHub project, published paper, workshop submission, public post, or course paper that you are most proud of. If possible, try to host documents on Google Drive or in a git repo.

References

Grigoris Antoniou. Defeasible logic with dynamic priorities. *International Journal of Intelligent Systems*, 19(5):463–472, 2004. doi: 10.1002/int.20008. URL https://doi.org/10.1002/int.20008.

Amanda Askell, Joe Carlsmith, Chris Olah, Jared Kaplan, Holden Karnofsky, et al. Claude's constitution. Anthropic, January 2026. URL https://www.anthropic.com/constitution. Published January 21, 2026; Amanda Askell is the primary author.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint*

arXiv:2212.08073, 2022. doi: 10.48550/arXiv.2212.08073. URL https://arxiv.org/abs/2212.08073.

Stefano Bistarelli, Ugo Montanari, and Francesca Rossi. Semiring-based constraint satisfaction and optimization. *Journal of the ACM*, 44(2):201–236, 1997. doi: 10.1145/256303.256306. URL https://dl.acm.org/doi/10.1145/256303.256306.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or LLMs as the judge? a study on judgement bias. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8301–8327, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.474. URL https://aclanthology.org/2024.emnlp-main.474/.

Zhaorun Chen, Mintong Kang, and Bo Li. ShieldAgent: Shielding agents via verifiable safety policy reasoning. *arXiv preprint arXiv:2503.22738*, 2025. URL https://arxiv.org/abs/2503.22738.

Leonardo de Moura and Nikolaj Bjørner. Z3: An efficient SMT solver. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008. doi: 10.1007/978-3-540-78800-3_24. URL https://doi.org/10.1007/978-3-540-78800-3_24.

Melody Y. Guan, Manas Joglekar, Eric Wallace, Saachi Jain, Boaz Barak, Alec Helyar, Rachel Dias, Andrea Vallone, Hongyu Ren, Jason Wei, Hyung Won Chung, Sam Toyer, Johannes Heidecke, Alex Beutel, and Amelia Glaese. Deliberative alignment: Reasoning enables safer language models. *arXiv preprint arXiv:2412.16339*, 2024. doi: 10.48550/arXiv.2412.16339. URL https://arxiv.org/abs/2412.16339.

Arya Jakkli, Senthooran Rajamanoharan, and Neel Nanda. How well do models follow their constitutions? *arXiv preprint arXiv:2605.24229*, 2026. doi: 10.48550/arXiv.2605.24229. URL https://arxiv.org/abs/2605.24229.

Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 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, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.257. URL https://aclanthology.org/2024.acl-long.257/.

OpenAI. Model spec. Versioned public specification, 2026. URL https://github.com/openai/model_spec/blob/19789a7350fda589cd74bc7491f348f68359b0ab/model_spec.md. Repository revision 19789a7350fda589cd74bc7491f348f68359b0ab, committed March 25, 2026; accessed July 22, 2026.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.248. URL https://aclanthology.org/2023.findings-emnlp.248/.

Qiuyi Qi, Jinjian Zhang, Mutian Bao, Tian Liang, Guocong Li, Dongnan Liu, Wei Zhou, Jie Liu, Ming Kong, Linjian Mo, Feng Zhang, and Qiang Zhu. CARL: Constraint-aware reinforcement

learning for planning with LLMs. *arXiv preprint arXiv:2607.04854*, 2026. URL https://arxiv.org/abs/2607.04854.

Yiming Su, Kunzhao Xu, Yanjie Gao, Fan Yang, Cheng Li, Mao Yang, and Tianyin Xu. Neuro-symbolic verification on instruction following of LLMs. *arXiv preprint arXiv:2601.17789*, 2026. URL https://arxiv.org/abs/2601.17789.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering (ICSE 2026)*, Rio de Janeiro, Brazil, 2026. URL https://arxiv.org/abs/2503.18666.

Bosi Wen, Pei Ke, Xiaotao Gu, Lindong Wu, Hao Huang, Jinfeng Zhou, Wenchuang Li, Binxin Hu, Wendy Gao, Jiaxin Xu, Yiming Liu, Jie Tang, Hongning Wang, and Minlie Huang. Benchmarking complex instruction-following with multiple constraints composition. *arXiv preprint arXiv:2407.03978*, 2024. doi: 10.48550/arXiv.2407.03978. URL https://arxiv.org/abs/2407.03978.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/71f7154547c748c8041505521ca433ab-Abstract-Conference.html.