Wisent: A General Framework for Reliable Representation Identification and Representation Steering

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Representation engineering is a powerful method for identifying and modifying high-level concepts within the internal layers of large language models. Despite its potential, real-life deployments of activation steering remain difficult. We present Wisent-Guard, a flexible, open-source framework for monitoring and steering internal activations of large language models. Practical applications of the framework show XXX percent hallucination reduction, XXX percent improvement in coding ability and deep personalization capabilities.

1 Introduction

Large language models, with billions of parameters and Internet-scale training dataset, have displayed significant capabilities across a wide range of tasks, such as writing, coding or reasoning.

However, their internal mechanisms of generating the next token cannot be precisely explained, with interactions between layers and parameters increasing in complexity as the size of these models increases.

Experiments with representation engineering (also known as steering or activation steering) have shown activation modification to be a powerful method of identifying and influencing high-level concepts (representations) within the layers of an LLM. Despite strong empirical performance on selected truthfulness, safety or personalization tasks, representation engineering methods lack a universal formulation and a unifying framework for understanding the underlying phenomenon, comparing methods and applying them to new problems.

We propose Wisent, a modular framework for analyzing the internal mechanisms within a large language model and influencing them to improve performance and individual alignment. Wisent-Guard surpasses state of the art performance in identifying particular behaviors

2 Representation Engineering Problem

We formulate the **Representation Engineering Problem** as the following:

For a given model M and a Representation

Basic primitives and definitions of key terms are outlined in Appendix A.

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

3 Representation Reading

3.1 Classifier

3.2 Detection Handling Method

4 Representation Control

4.1 Classifier

References

[1] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[2] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, 2013. Association for Computational Linguistics.

[3] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019.

[4] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[5] Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha, Michael Greenberg, and Abhinav Jangda. Multipl-e: A scalable and extensible approach to benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*, 2022.

[6] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. 2021.

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, John Schulman, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, and Jerry Tworek. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] CodeParrot. Instructhumaneval, 2023.

[11] Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung 23*, volume 2, pages 107–124, Bellaterra (Cerdanyola del Vallès), 2019. Universitat Autònoma de Barcelona.

[12] Mingzhe Du, Anh Tuan Luu, Bin Ji, Liu Qian, and See-Kiong Ng. Mercury: A code efficiency benchmark for code llms. *arXiv preprint arXiv:2402.07844*, 2024.

[13] Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, and et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*, 2025.

[14] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs. *arXiv preprint arXiv:1903.00161*, 2019.

[15] Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, and Jacob Steinhardt. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

[17] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[18] HuggingFaceH4. Math-500, 2024.

[19] Srinivasan Iyer, Ioannis Konstas, Alvin Cheung, and Luke Zettlemoyer. Mapping language to code in programmatic context. *arXiv preprint arXiv:1808.09588*, 2018.

[20] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

[21] Maxwell Jia. Aime problem set 2024, 2024.

[22] Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada, 2017. Association for Computational Linguistics.

[23] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019.

[24] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*, 2017.

[25] Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Scott Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. Ds-1000: A natural and reliable benchmark for data science code generation. *arXiv preprint arXiv:2211.11501*, 2022.

[26] Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

[27] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *NeurIPS 2023 Datasets and Benchmarks Track*, 2023.

[28] Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. Are your llms capable of stable reasoning? *arXiv preprint arXiv:2412.13147*, 2024.

[29] Shuai Lu, Daya Guo, Shuo Ren, Junjie Huang, Alexey Svyatkovskiy, Ambrosio Blanco, Colin Clement, Dawn Drain, Daxin Jiang, Duyu Tang, Ge Li, Lidong Zhou, Linjun Shou, Long Zhou, Michele Tufano, Ming Gong, Ming Zhou, Nan Duan, Neel Sundaresan, Shao Kun Deng, Shengyu Fu, and Shujie Liu. Codexglue: A machine learning benchmark dataset for code understanding and generation. *arXiv preprint arXiv:2102.04664*, 2021.

[30] Math-AI. Aime problem set 2025, 2025.

[31] MathArena. Hmmt february 2025, 2025.

[32] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[33] Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*, 2021.

[34] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2018.

[35] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, and et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[36] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*, 2018.

[37] Siva Reddy, Danqi Chen, and Christopher D. Manning. Coqa: A conversational question answering challenge. *arXiv preprint arXiv:1808.07042*, 2019.

[38] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[39] Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, Stanford, CA, 2011.

[40] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *arXiv preprint arXiv:1907.10641*, 2019.

[41] Shiqi Wang, Zheng Li, Haifeng Qian, Chenghao Yang, Zijian Wang, Mingyue Shang, Varun Kumar, Samson Tan, Baishakhi Ray, Parminder Bhatia, Ramesh Nallapati, Murali Krishna Ramanathan, Dan Roth, and Bing Xiang. Recode: Robustness evaluation of code generation models. *arXiv preprint arXiv:2212.10264*, 2022.

[42] Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, Qiqian Cang, Yichang Zhang, Fei Huang, Junyang Lin, Fei Huang, and Jingren Zhou. Polymath: Evaluating mathematical reasoning in multilingual contexts. *arXiv preprint arXiv:2504.18428*, 2025.

[43] Pengcheng Yin, Bowen Deng, Edgar Chen, Bogdan Vasilescu, and Graham Neubig. Learning to mine aligned code and natural language pairs from stack overflow. In *Proceedings of the 15th IEEE/ACM International Conference on Mining Software Repositories (MSR)*, pages 476–486. IEEE, 2018.

[44] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, 2018. Association for Computational Linguistics.

[45] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[46] Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*, 2018.

A Wisent Primitives

A.1 Model

A.2 Contrastive Pair

A.3 Activations

A.4 Activation Collection Method

A.5 Additional Utilities

B Representation Reading Functionalities

B.1 Classifier

B.2 Detection Handling Method

C Representation Control Functionalities

D Ablation

A All supported benchmarks

This section enumerates all benchmarks used in our study, the task traits, the evaluation protocol, and the contrastive pair generation method applied to produce minimally perturbed negative targets. We first merged the *coding* and *mathematics* benchmark lists you provided and then appended them to the original master list.

Contrastive pair generation methods (definitions)

Reading Comprehension Abstention Swap [RC-Abstain] For extractive/open-domain RC: positive is the gold span; negative is an abstention (e.g., "Not provided in the text."). If gold is *No answer*, the negative is a confident but wrong claim.

Conversational Reading Comprehension Abstention [ConvRC-Abstain] As RC-Abstain, but with dialogue context (CoQA). Negatives are generic abstentions; yes/no items are flipped when applicable.

Language Modeling Corrupted Continuation [LM-CorruptCont] Language modeling: positive is the true continuation; negative is a corrupted continuation (local shuffles/randomization) to break coherence.

Two-Choice Flip [2C-Flip] Two-option tasks (PIQA, COPA, WinoGrande, CB): negative is simply the other option.

Multichoice First Distractor [MC-FirstDistr] Multi-choice tasks: negative = the first incorrect option in the provided order (deterministic).

Multichoice Random Distractor [MC-RandDistr] Multi-choice tasks: negative = a randomly chosen incorrect option from the same set (used for GPQA).

Multichoice Letter Swap [MC-LetterSwap] Multi-choice tasks scored over option letters (TruthfulQA MC1/MC2): negative = the first incorrect letter.

Boolean Flip [Bool-Flip] Binary tasks (BoolQ): negative is the opposite boolean label.

Exact Match Partial Mask [EM-PartialMask] Exact-match free-form answers (HLE-EM): negative is the gold text with partial token masking (approximately 1/3 words, or partial masking for single-word answers).

Keyword-Preserving Token Deletion [KP-Del] Coding tasks: negative program created by deleting non-keyword tokens while preserving syntax-critical keywords; aims to remain plausible but fail unit tests.

Summary Content-Word Drop [Summ-WordDrop] Code-to-text summarization: negative description formed by dropping content words (nouns/verbs) while keeping scaffolding words to preserve superficial form.

Numeric Offset (+1) Perturbation [Num+1] Math QA: negative is the correct numeric answer offset by a small integer (typically +1); for non-integer answers, apply the minimal unit offset.

Evaluation types (definitions)

Log-likelihood option scoring [LL] The model scores each provided option/target by conditional log-probability given the prompt. Metrics typically compute accuracy over the highest-likelihood choice (MC tasks) or compare likelihoods of gold vs. negative targets.

Text generation string matching [TG] The model generates free-form text (or a number), which is then judged by task-specific metrics (e.g., exact match on numerical value for GSM8K/MATH; span/string matching for RC tasks; structured checks for DROP). Used also for CoT/generative GPQA variants and HLE-Exact-Match.

Perplexity (language modeling) [PPL] The model's next-token distribution is evaluated over a reference text to compute Perplexity (lower is better). Used for language-modeling corpora like WikiText.

Code execution against unit tests [CE] The model generates code, which is executed in a sandbox against unit tests provided by a dataset (e.g., pass@1). Applies to HumanEval/MBPP/APPS, MultiPL-E, DS-1000, LiveCodeBench, etc.

Table 1: Benchmarks (short names), evaluation abbreviations, contrastive method (short), and traits. Versions merged where applicable.

Benchmark	Eval	Method [CM]	Traits
DROP [14]	[TG]	**RC-Abstain**	reading comprehension
ReCoRD [46]	[TG]	**RC-Abstain**	reading comprehension
SQuAD2 [36]	[TG]	**RC-Abstain**	reading comprehension
WebQuestions [2]	[TG]	**RC-Abstain**	factual QA
Natural Questions [23]	[TG]	**RC-Abstain**	factual QA
TriviaQA [22]	[TG]	**RC-Abstain**	factual QA
CoQA [37]	[TG]	**ConvRC-Abstain**	conversational RC
BoolQ [7]	[LL]	**Bool-Flip**	boolean RC
WinoGrande [40]	[LL]	**2C-Flip**	commonsense
PIQA [3]	[LL]	**2C-Flip**	commonsense
COPA [39]	[LL]	**2C-Flip**	causal reasoning
HellaSwag [45]	[LL]	**MC-FirstDistr**	commonsense
SWAG [44]	[LL]	**MC-FirstDistr**	commonsense
OpenBookQA [34]	[LL]	**MC-FirstDistr**	science MCQ
ARC [8]	[LL]	**MC-FirstDistr**	science MCQ
RACE [24]	[LL]	**MC-FirstDistr**	RC (MC)
MMLU [16]	[LL]	**MC-FirstDistr**	multi-subject exams
GPQA [38]	[LL]/[TG]	**MC-RandDistr**	expert STEM exams
SuperGPQA [13]	[LL]	**MC-FirstDistr**	expert STEM exams
HLE [35]	[TG]/[LL]	**EM-PartialMask; MC-FirstDistr**	expert exams

Benchmark	Eval	Method [CM]	Traits
GSM8K [9]	[TG]	**Num+1**	mathematics
ASDiv [33]	[TG]	**Num+1**	mathematics
Arithmetic [4]	[TG]	**Num+1**	mathematics
MATH [17]	[TG]	**Num+1**	mathematics (contest)
MATH–500 [18]	[TG]	**Num+1**	mathematics (contest)
AIME [30][21]	[TG]	**Num+1**	mathematics (contest)
HMMT [31]	[TG]	**Num+1**	mathematics (contest)
PolyMath [42]	[TG]	**Num+1**	mathematics (multiling.)
LiveMathBench [28]	[TG]	**Num+1**	mathematics (EN/ZH)
MBPP [1]	[CE]	**KP-Del**	coding (Python)
HumanEval [6]	[CE]	**KP-Del**	coding (Python)
CoNaLa [43]	[CE]	**KP-Del**	coding (Python)
CONCODE [19]	[CE]	**KP-Del**	coding (Java)
Mercury [12]	[CE]	**KP-Del**	coding (multi-language)
HumanEval+ [27]	[CE]	**KP-Del**	coding (Python)
InstructHumanEval [10]	[CE]	**KP-Del**	coding (Python)
MBPP+ [27]	[CE]	**KP-Del**	coding (Python)
APPS [15]	[CE]	**KP-Del**	coding (Python)
DS–1000 [25]	[CE]	**KP-Del**	coding (Python)
MultiPL–E [5]	[CE]	**KP-Del**	coding (multi-language)
CodeXGLUE [29]	[TG]	**Summ-WordDrop**	coding (code-to-text)
ReCode [41]	[CE]	**KP-Del**	coding (Python)
LiveCodeBench [20]	[CE]	**KP-Del**	coding (Python)
TruthfulQA [26]	[LL]	**MC-LetterSwap**	truthfulness
CB [11]	[LL]	**2C-Flip**	NLI
WikiText (2/103) [32]	[PPL]	**LM-CorruptCont**	language modeling

Category legend

- RC/ODQA
- Multi-choice Reasoning
- Exams & Knowledge Tests
- Mathematics
- Coding
- Other (Truthfulness/NLI/LM)

Abbreviation legend

`[LL]`	Log-likelihood option scoring
`[TG]`	Text generation (string match)
`[PPL]`	Perplexity (LM)
`[CE]`	Code execution vs. unit tests

Method [CM] codes

RC-Abstain	RC abstention swap
ConvRC-Abstain	Conversational RC abstention
LM-CorruptCont	LM corrupted continuation
2C-Flip	Two-choice flip
MC-FirstDistr	First distractor (MC)
MC-RandDistr	Random distractor (MC)
MC-LetterSwap	Letter swap (MC)
Bool-Flip	Boolean flip
EM-PartialMask	Exact-match partial mask
KP-Del	Keyword-preserving deletion
Summ-WordDrop	Summary word drop
Num+1	Numeric offset (+1)

B Per-Task Results

C Detailed Classification Results

D Benchmark-Aided Steering Results

E Optimal Sample Size Calculations

F Fully Synthethic Generation

G Agentic Capabilities