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

Anonymous Author(s)

Affiliation

Address

email

Abstract

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

8 1 Introduction

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

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

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

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

23 2 Representation Engineering Problem

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

25 For a given model M and a Representation

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

27 3 Representation Reading

28 3.1 Classifier

29 3.2 Detection Handling Method

30 4 Representation Control

31 4.1 Classifier

32 References

- 33 Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan,
34 Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large
35 language models. *arXiv preprint arXiv:2108.07732*, 2021.
- 36 Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on freebase
37 from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in*
38 *Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA, 2013. Association for
39 Computational Linguistics.
- 40 Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about
41 physical commonsense in natural language. *arXiv preprint arXiv:1911.11641*, 2019.
- 42 Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal,
43 Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are
44 few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- 45 Federico Cassano, John Gouwar, Daniel Nguyen, Sydney Nguyen, Luna Phipps-Costin, Donald
46 Pinckney, Ming-Ho Yee, Yangtian Zi, Carolyn Jane Anderson, Molly Q. Feldman, Arjun Guha,
47 Michael Greenberg, and Abhinav Jangda. Multipl-e: A scalable and extensible approach to
48 benchmarking neural code generation. *arXiv preprint arXiv:2208.08227*, 2022.
- 49 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared
50 Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri,
51 Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan,
52 Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian,
53 Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios
54 Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino,
55 Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders,
56 Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa,
57 Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob
58 McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating
59 large language models trained on code. 2021.
- 60 Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina
61 Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint*
62 *arXiv:1905.10044*, 2019.
- 63 Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and
64 Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge.
65 *arXiv preprint arXiv:1803.05457*, 2018.
- 66 Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher
67 Hesse, John Schulman, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, and Jerry
68 Tworek. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- 69 CodeParrot. Instructhumaneval, 2023.
- 70 Marie-Catherine de Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: In-
71 vestigating projection in naturally occurring discourse. In *Proceedings of Sinn und Bedeutung*
72 *23*, volume 2, pages 107–124, Bellaterra (Cerdanyola del Vallès), 2019. Universitat Autònoma de
73 Barcelona.

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

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

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

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

85 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Ja-
86 cob Steinhardt. Measuring massive multitask language understanding. In *Proceedings of the*
87 *International Conference on Learning Representations (ICLR)*, 2021.

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

91 HuggingFaceH4. Math-500, 2024.

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

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

97 Maxwell Jia. Aime problem set 2024, 2024.

98 Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly
99 supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual*
100 *Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–
101 1611, Vancouver, Canada, 2017. Association for Computational Linguistics.

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

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

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

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

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

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

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

125 Math-AI. Aime problem set 2025, 2025.

126 MathArena. Hmmt february 2025, 2025.

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

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

131 Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct
132 electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference*
133 *on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics,
134 2018.

135 Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, and et al. Humanity’s last exam. *arXiv preprint*
136 *arXiv:2501.14249*, 2025.

137 Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don’t know: Unanswerable questions
138 for squad. *arXiv preprint arXiv:1806.03822*, 2018.

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

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

144 Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. Choice of plausible alterna-
145 tives: An evaluation of commonsense causal reasoning. In *AAAI Spring Symposium on Logical*
146 *Formalizations of Commonsense Reasoning*, Stanford, CA, 2011.

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

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

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

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

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

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

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

170 **A Wisent Primitives**

171 **A.1 Model**

172 **A.2 Contrastive Pair**

173 **A.3 Activations**

174 **A.4 Activation Collection Method**

175 **A.5 Additional Utilities**

176 **B Representation Reading Functionalities**

177 **B.1 Classifier**

178 **B.2 Detection Handling Method**

179 **C Representation Control Functionalities**

180 **D Ablation**

181 **A All supported benchmarks**

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

186 **Contrastive pair generation methods (definitions)**

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

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

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

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

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

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

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

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

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

208	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.
209		
210		
211	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.
212		
213		
214	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.
215		
216		
217	Evaluation types (definitions)	
218	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.
219		
220		
221	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.
222		
223		
224		
225	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.
226		
227		
228	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.
229		
230		







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

Benchmark	Eval	Method [CM]	Traits
DROP ?	[TG]	RC-Abstain	reading comprehension
ReCoRD ?	[TG]	RC-Abstain	reading comprehension
SQuAD2 ?	[TG]	RC-Abstain	reading comprehension
WebQuestions ?	[TG]	RC-Abstain	factual QA
Natural Questions ?	[TG]	RC-Abstain	factual QA
TriviaQA ?	[TG]	RC-Abstain	factual QA
CoQA ?	[TG]	ConvRC-Abstain	conversational RC
BoolQ ?	[LL]	Bool-Flip	boolean RC
WinoGrande ?	[LL]	2C-Flip	commonsense
PIQA ?	[LL]	2C-Flip	commonsense
COPA ?	[LL]	2C-Flip	causal reasoning
HellaSwag ?	[LL]	MC-FirstDistr	commonsense
SWAG ?	[LL]	MC-FirstDistr	commonsense
OpenBookQA ?	[LL]	MC-FirstDistr	science MCQ
ARC ?	[LL]	MC-FirstDistr	science MCQ
RACE ?	[LL]	MC-FirstDistr	RC (MC)
MMLU ?	[LL]	MC-FirstDistr	multi-subject exams
GPQA ?	[LL]/[TG]	MC-RandDistr	expert STEM exams
SuperGPQA ?	[LL]	MC-FirstDistr	expert STEM exams
HLE ?	[TG]/[LL]	EM-PartialMask; MC-FirstDistr	expert exams
GSM8K ?	[TG]	Num+1	mathematics
ASDiv ?	[TG]	Num+1	mathematics
Arithmetic ?	[TG]	Num+1	mathematics

Benchmark	Eval	Method [CM]	Traits
MATH ?	[TG]	Num+1	mathematics (contest)
MATH-500 ?	[TG]	Num+1	mathematics (contest)
AIME ??	[TG]	Num+1	mathematics (contest)
HMMT ?	[TG]	Num+1	mathematics (contest)
PolyMath ?	[TG]	Num+1	mathematics (multiling.)
LiveMathBench ?	[TG]	Num+1	mathematics (EN/ZH)
MBPP ?	[CE]	KP-Del	coding (Python)
HumanEval ?	[CE]	KP-Del	coding (Python)
CoNaLa ?	[CE]	KP-Del	coding (Python)
CONCODE ?	[CE]	KP-Del	coding (Java)
Mercury ?	[CE]	KP-Del	coding (multi-language)
HumanEval+ ?	[CE]	KP-Del	coding (Python)
InstructHumanEval ?	[CE]	KP-Del	coding (Python)
MBPP+ ?	[CE]	KP-Del	coding (Python)
APPS ?	[CE]	KP-Del	coding (Python)
DS-1000 ?	[CE]	KP-Del	coding (Python)
MultiPL-E ?	[CE]	KP-Del	coding (multi-language)
CodeXGLUE ?	[TG]	Summ-WordDrop	coding (code-to-text)
ReCode ?	[CE]	KP-Del	coding (Python)
LiveCodeBench ?	[CE]	KP-Del	coding (Python)
TruthfulQA ?	[LL]	MC-LetterSwap	truthfulness
CB ?	[LL]	2C-Flip	NLI
WikiText (2/103) ?	[PPL]	LM-CorruptCont	language modeling

231

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)

232	B	Per-Task Results
233	C	Detailed Classification Results
234	D	Benchmark-Aided Steering Results
235	E	Optimal Sample Size Calculations
236	F	Fully Synthetic Generation
237	G	Agentic Capabilities