The Journal of Open Source Software

CCS-Lib: A Python package to elicit latent knowledge from LLMs

Walter Laurito[3*¶], **Nora Belrose**[1*], **Alex Mallen**[1,4], **Kay Kozaronek**[2], **Fabien Roger**[4], **Christy Koh**[5], **James Chua**[1], **Jonathan NG**[2], **Alexander Wan**[5], **Reagan Lee**[5], **Ben W.**[1], **Kyle O'Brien**[1,6], **Augustas Macijauskas**[7], **Eric Mungai Kinuthia**[1], **Marius PL**[2], **Waree Sethapun**[8], **and Kaarel Hänni**[2]

1 EleutherAI **2** Independent **3** FZI Research Center for Information Technology **4** Redwood Research **5** UC Berkeley **6** Microsoft **7** CAML Lab, University of Cambridge **8** Princeton University ¶ Corresponding author * These authors contributed equally.

DOI: 10.21105/joss.06511

Software

- Review
- Repository
- Archive

Editor: Beatriz Costa Gomes

Reviewers:

- @praneethd7
- @isdanni

Submitted: 09 December 2023
Published: 21 October 2025

Summary

CCS-Lib is a library designed to elicit latent knowledge (elk (Christiano et al., December 2021)) from language models. It includes implementations of both the original and an enhanced version of the Contrast-Consistent Search (CCS) method and an approach based on Contrastive Representation Clustering-Top Principal Component (CRC-TPC) (Burns et al., 2022), called VINC. Designed for researchers, the CCS-Lib offers features like multi-GPU support, integration with Hugging Face and the training of supervised probes for comparisons.

Statement of need

The widespread adoption of language models in real-world applications presents significant challenges, particularly the potential generation of unreliable or inaccurate content (Evans et al., 2021; Hendrycks et al., 2021; Park et al., 2024; Weidinger et al., 2021). A notable concern is that models fine-tuned on human preferences may exacerbate existing biases or lead to convincing yet misleading outputs (Perez et al., 2022).

Recent studies indicate that it's possible to extract simulated internal beliefs or 'knowledge' from language model activations (Azaria & Mitchell, 2023; Bubeck et al., 2023; Gurnee & Tegmark, 2023; Li et al., 2022). While supervised probing techniques can be used for this purpose (Alain & Bengio, 2016; Marks & Tegmark, 2023), they rely on labels that may be compromised by human biases or limitations in human knowledge. In some cases, it's crucial to avoid human labels altogether to allow distinguishing between a model's true knowledge and its representation of human beliefs.

These considerations have led to the development of unsupervised probing methods, such as Contrast-Consistent Search (CCS) (Burns et al., 2022). These techniques aim to extract knowledge embedded in language models without relying on ground truth labels (Burns et al., 2022; Zou et al., 2023). Such approaches offer a promising direction for uncovering the latent knowledge within language models while mitigating the influence of human biases and limitations.

Nonetheless, current unsupervised probing methods still face challenges (Farquhar et al., 2023; Laurito et al., 2024; Levinstein & Herrmann, 2024). These issues underscore the need for tools that enable researchers to easily train, investigate, and compare probes while analyzing the internal representations of language models. In this context, one aim of our CCS-Lib is to provide a testbed that allows researchers to experiment with existing unsupervised probing methods — and compare them with their supervised counterparts — to elicit latent knowledge from within the activations of a language model.

Refer to the *Example Usage* section for a demonstration of how to use the library.

Implementation

The CCS-Lib is developed to provide both the original and an enhanced version of the Contrast-Consistent Search (CCS) method described in Burns et al. (2022).

Our enhanced version of CCS uses the Limited-memory BFGS (LBFGS) optimizer instead of Adam, which speeds up the training process. Furthermore, it uses learnable Platt scaling parameters to avoid the problem of sign ambiguity from the original implementation.

In addition, we have implemented an approach called VINC (Variance, Invariance, Negative Covariance). VINC is an enhanced method for eliciting latent knowledge from language models. It builds upon the Contrastive Representation Clustering—Top Principal Component (CRC-TPC) (Burns et al., 2022) approach and incorporates additional principles. VINC aims to find a direction in activation space that maximizes variance while encouraging negative correlation between statement pairs and paraphrase invariance. The method uses eigendecomposition to optimize a quadratic objective that balances these criteria. VINC can be seen as an alternative to CCS, which takes less time to train. Additional changes and more recent results on VINC and its successor can be found here.

Finally, we provide a method to train supervised probes using logistic regression, allowing a comparison with unsupervised methods.

CCS-Lib serves as a tool for researchers to investigate the truthfulness of model outputs and explore the underlying beliefs embedded within the model. The library offers:

- A clean implementation of the enhanced and original version of CCS
- Multi-GPU Support: Efficient extraction, training, and evaluation through parallel processing
- Integration with Hugging Face: Easy utilization of models and datasets from a popular source
- VINC, an alternative to CCS
- Training supervised probes with logistic regression for comparisons

For collaboration, discussion, and support, the Eleuther AI Discord's elk channel provides a platform for engaging with others interested in the library or related research projects.

Example usage - Comparing unsupervised and supervised probes

As mentioned above, one aim of this library is to provide a testbed for experimentation with unsupervised probing methods and to compare them with their supervised counterparts. We provide a simple example of how to use the library to compare the performance of unsupervised and supervised probes.

First install the package with `pip install -e .` in the root directory. This should install all the necessary dependencies.

To fit reporters for the Hugging Face model `model` and dataset `dataset`, run:

```
ccs elicit microsoft/deberta-v2-xxlarge-mnli imdb
```

This will automatically download the model and dataset, run the model and extract the relevant representations if they aren't cached on disk, fit reporters on them, and save the reporter checkpoints to the `ccs-reporters` folder in your home directory. It will also evaluate the reporter classification performance on a held out test set and save it to a CSV file in the same folder. By default the VINC reporter is used.

In addition, as an upper-bound is provided by training a supervised reporter using logistic regression (LR) models.

Once the run is complete, the following files are generated for analysis:

- `results/reporters/`: Folder containing the trained reporters (CCS or VINC probes) for each layer
- `results/cfg.yaml`: Configuration used for the run
- `results/eval.csv`: Evaluation results for the reporters
- `results/train_eval.csv`: Evaluation results for the reporter on the training set
- `results/lr_eval.csv`: Evaluation results for the logistic regression models
- `results/sweeps/`: Folder containing the sweeps for the run
- `results/plots/`: Folder containing the plots for the run
- `results/fingerprints.yaml`: Metadata files that store unique identifiers (fingerprints) for different dataset splits

Now, if you want to run a sweep to compare the performance of different models and datasets, you can use the following command:

```
ccs sweep --models gpt2-{medium,large,xl} --datasets imdb amazon_polarity --add_pooled
```

Additional details are available in the library's README.

State of the field

Most available code is often tailored to demonstrate a paper's specific methods and results rather than being user-friendly for researchers (Burns et al., 2022; Farquhar et al., 2023; Marks & Tegmark, 2023). In contrast, our work is explicitly engineered to simplify the testing, comparison, and enhancement of unsupervised methods. The key features of the library are described in Section *Implementation*.

Acknowledgements

We would like to thank EleutherAI, SERI MATS, and Long-Term Future Fund (LTFF) for supporting our work.

References

Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv Preprint arXiv:1610.01644*. https://doi.org/10.48550/arXiv.1610.01644

Azaria, A., & Mitchell, T. (2023). The internal state of an LLM knows when it's lying. *Findings of the Association for Computational Linguistics: EMNLP 2023*. https://doi.org/10.18653/v1/2023.findings-emnlp.68

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., & others. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv Preprint arXiv:2303.12712*. https://doi.org/10.48550/arXiv.2303.12712

Burns, C., Ye, H., Klein, D., & Steinhardt, J. (2022). Discovering latent knowledge in language models without supervision. *arXiv Preprint arXiv:2212.03827*. https://doi.org/10.48550/arXiv.2212.03827

Christiano, P., Cotra, A., & Xu, M. (December 2021). *Eliciting latent knowledge (ELK)*. https://docs.google.com/document/d/1WwsnJQstPq91_Yh-Ch2XRL8H_EpsnjrC1dwZXR37PC8/.

Evans, O., Cotton-Barratt, O., Finnveden, L., Bales, A., Balwit, A., Wills, P., Righetti, L., & Saunders, W. (2021). Truthful AI: Developing and governing AI that does not lie. *arXiv Preprint arXiv:2110.06674*. https://doi.org/10.48550/arXiv.2110.06674

Farquhar, S., Varma, V., Kenton, Z., Gasteiger, J., Mikulik, V., & Shah, R. (2023). Challenges with unsupervised LLM knowledge discovery. *arXiv Preprint arXiv:2312.10029*. https://doi.org/10.48550/arXiv.2312.10029

Gurnee, W., & Tegmark, M. (2023). Language models represent space and time. *arXiv Preprint arXiv:2310.02207*. https://doi.org/10.48550/arXiv.2310.02207

Hendrycks, D., Carlini, N., Schulman, J., & Steinhardt, J. (2021). Unsolved problems in ML safety. *arXiv Preprint arXiv:2109.13916*. https://doi.org/10.48550/arXiv.2109.13916

Laurito, W., Maiya, S., Dhimoïla, G., Hänni, K., & others. (2024). Cluster-norm for unsupervised probing of knowledge. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2024.emnlp-main.780

Levinstein, B. A., & Herrmann, D. A. (2024). Still no lie detector for language models: Probing empirical and conceptual roadblocks. *Philosophical Studies*, 1–27. https://doi.org/10.1007/s11098-023-02094-3

Li, K., Hopkins, A. K., Bau, D., Viégas, F., Pfister, H., & Wattenberg, M. (2022). Emergent world representations: Exploring a sequence model trained on a synthetic task. *arXiv Preprint arXiv:2210.13382*. https://doi.org/10.48550/arXiv.2210.13382

Marks, S., & Tegmark, M. (2023). *The geometry of truth: Emergent linear structure in large language model representations of true/false datasets*. https://doi.org/10.48550/arXiv.2310.06824

Park, P. S., Goldstein, S., O'Gara, A., Chen, M., & Hendrycks, D. (2024). AI deception: A survey of examples, risks, and potential solutions. *Patterns*. https://doi.org/10.1016/j.patter.2024.100988

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., & Irving, G. (2022). Red teaming language models with language models. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2022.emnlp-main.225

Weidinger, L., Mellor, J., Rauh, M., Griffin, C., Uesato, J., Huang, P.-S., Cheng, M., Glaese, M., Balle, B., Kasirzadeh, A., & Kenton, Z. (2021). Ethical and social risks of harm from language models. *arXiv Preprint arXiv:2112.04359*. https://doi.org/10.48550/arXiv.2112.04359

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., & others. (2023). Representation engineering: A top-down approach to AI transparency. *arXiv Preprint arXiv:2310.01405*. https://doi.org/10.48550/arXiv.2310.01405