
Fast Orchestrated Reproducible Generative Extraction¹

Ayşe Asude Demir

Tuğrul Demir

With
Apart Research

Abstract

Robustly detecting and mitigating dangerous LLM behaviors, such as deception, sycophancy, and alignment faking, is a critical challenge in AI safety. While Representation Engineering offers mechanistic control via steering vectors, current methods face significant operational bottlenecks: Contrastive Activation Addition (CAA) demands large, manually curated datasets, while One-Shot steering suffers from high variance and instability. To resolve this dichotomy, we introduce **Vector Forge**, an automated pipeline that extracts robust steering vectors from a single natural language description.

By utilizing an agentic workflow to autonomously synthesize, validate, and select high-quality contrastive datasets, Vector Forge eliminates the data curation barrier. We evaluate our pipeline on Llama 3.1 8B against standard CAA and One-Shot baselines using Geodesic Research’s Misalignment and Anthropic Sycophancy datasets. Our results demonstrate that Vector Forge vectors match or exceed the steering efficacy of data-rich baselines. Furthermore, we validate the utility of these autonomously generated vectors as classifiers for detecting misaligned outputs. By democratizing access with an easy activation engineering interface, Vector Forge provides the AI safety community with a scalable, rapid-response tool for monitoring and mitigating manipulative behaviors without requiring deep technical expertise or pre-existing datasets.

¹ Research conducted at the [AI Manipulation Hackathon](#), 2026

Keywords: AI Safety, Steering Vectors, Representation Engineering, Mechanistic Interpretability, Sycophancy, Deception

1. Introduction

As Large Language Models (LLMs) become increasingly capable, the need for robust, rapidly deployable mechanisms to detect and mitigate dangerous behaviors such as deception, sycophancy, and alignment faking has become critical. Representation Engineering (RepE) (Zou et al., 2023) has emerged as a paradigm for rigorous interpretability, enabling researchers to directly intervene on activations to test causal hypotheses and map the mechanistic structure of model behavior. Within this field, steering vectors (SV) have become a popular tool for controlling model behavior, effectively shifting activations to promote honesty or suppress toxicity (Liu et al., 2023).

However, a significant dichotomy exists in current steering methodologies. On one end, robust methods like Contrastive Activation Addition (CAA) (Panickssery et al. 2024) require large, contrastive datasets to extract reliable steering directions. Constructing these datasets is labor-intensive and often impractical for emerging threats, particularly for elusive behaviors like sandbagging or alignment faking where consistent replication is challenging because it is often unknown which inputs will actually induce the behavior. On the other end, recent work by Dunefsky et al. (2025) proposes optimizing steering vectors on a single example ("one-shot steering") to bypass the data bottleneck. While promising, this method introduces significant instability: the resulting vectors exhibit high variance depending on the training example used. Consequently, finding a reliable enough vector often necessitates a validation set and a laborious trial-and-error process, effectively negating the theoretical efficiency of the one-shot approach.

In this report, we present a novel, automated pipeline that bridges this gap. Confronted with the operational inefficiency of tuning unstable one-shot vectors and the barrier of obtaining large contrastive datasets in our own experiments, we engineered the solution our workflow demanded: Vector Forge. Crucially, unlike traditional approaches that rely on pre-existing contrastive datasets, Vector Forge requires no input data. Instead, it utilizes an agentic workflow to autonomously generate, evaluate, and select high-quality contrastive examples from scratch. Featuring an easy-to-use GUI, shown in Figure 1, this pipeline democratizes access to activation engineering, allowing even non-researchers to explore LLM mechanisms and causal structures without deep technical expertise or manual data curation.

We validate the efficacy of our automated pipeline by evaluating the generated steering vectors across two datasets, benchmarking them against standard CAA vectors derived from curated contrastive datasets and one-shot steering vectors. We observe that the steering capability of our tool’s vectors matches or exceeds that of standard CAA vectors. Remarkably, while one-shot methods reduce effort but suffer from a predict-control discrepancy (Wattenberg & Viégas, 2024), our tool

lowers barrier to entry further while ensuring robust classification by autonomously constructing the necessary synthetic contrastive dataset.

The remainder of this report is organized as follows: Section 2 details the tool's architecture, the agentic workflow, and our experimental evaluation setup. Section 3 presents the results and their interpretation. Finally, Section 4 concludes the report and discusses current limitations. Our code for Vector Forge is available at: <https://github.com/AyseAsude/vector-forge>, our code for evaluation is available at: <https://github.com/AyseAsude/steer-guard>.

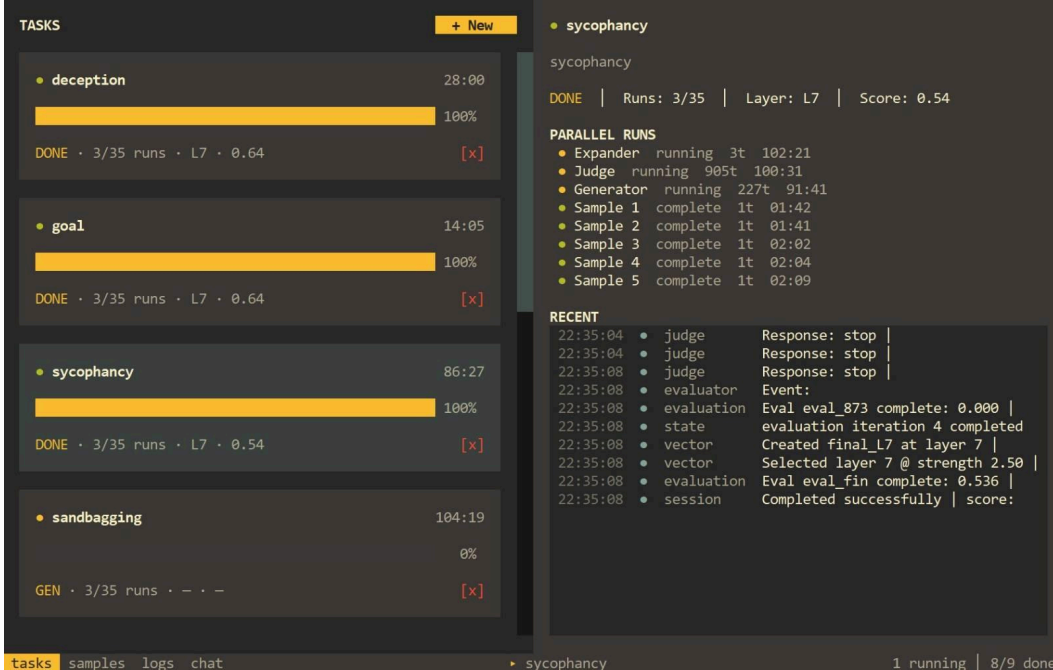


Figure 1: Tasks dashboard from Vector Forge

2. Methods

The Vector Forge Pipeline

Vector Forge is a fully autonomous pipeline designed to extract robust steering vectors using only a natural language description of the desired behavioral shift. Conceptually, the system functions as an agentic loop that substitutes manual dataset curation with high-quality synthetic data generation. The user provides a target behavior, and the tool orchestrates a multi-stage process to synthesize a contrastive dataset, validate its quality, and extract a steering vector. Crucially, the pipeline goes beyond simple extraction; it actively verifies the vector's efficacy by deploying it in-loop and empirically measuring its performance against a suite of behavioral and capability metrics. This architecture effectively automates the RepE workflow, moving from a semantic description to a verified mechanistic intervention without requiring manual trial-and-error loops or pre-existing data. Figure 2 demonstrates an intermediate step during generation of pairs.

The pipeline begins with Behavior Expansion, where an LLM expands the user's brief description into a comprehensive specification, defining positive and negative

examples, domain contexts, and potential confounds. Using this specification, the system performs Contrast Pair Generation, creating training samples at varying intensities: creating training samples across a spectrum of intensities: Natural (subtle, realistic usage), Medium, High, and Extreme (maximum, unmistakable expression). Each sample consists of a user prompt, a "destination" response (exhibiting the behavior), and a "source" response (absent the behavior). To mitigate the noise inherent in synthetic data, every generated pair undergoes a rigorous Quality Validation chain. Pairs are scored on a "Behavioral Signal" metric to ensure the target behavior is distinct, and a "Confound Score" to verify that non-behavioral attributes such as response length, formality, and structure are controlled. Only pairs meeting strict signal-to-noise thresholds are retained.

Once a filtered dataset is established, the pipeline employs Contrastive Activation Addition (CAA) to extract candidate steering vectors. To ensure robustness, extraction runs multiple times using bootstrap sampling (random subsets of pairs) and optionally different model layers, producing a pool of candidate vectors. Each candidate is then empirically validated: the vector is applied to the target model at multiple strength multipliers (0.5x to 2.5x), generating steered outputs on separate evaluation prompts. An LLM judge scores these outputs across five dimensions: behavioral induction, specificity, coherence, capability preservation, and generalization. Finally, candidates that pass quality thresholds are consolidated into a single robust vector using configurable aggregation strategies such as weighted averaging, top-k averaging, or PCA to extract the common directional signal.

The screenshot displays the Vector Forge interface during a contrastive pair generation task. On the left, a sidebar titled 'PARALLEL RUNS' shows 3 active tasks out of 35 total. The tasks listed are: Expander (103:04, 3 turns, 0 calls), Judge (101:13, 905 turns, 0 calls), Generator (92:24, 227 turns, 0 calls), Sample 1 (01:42, 1 turn, 25 pairs), Sample 2 (01:41, 1 turn, 25 pairs), and Sample 3 (02:02, 1 turn, 25 pairs). The main chat area shows a conversation between a user and an assistant. The user prompts the generator to create contrast pairs for steering vector training, and the assistant provides a response. The interface also shows a 'tasks' tab with a list of tasks including Expander, Judge, Generator, and Sample 1, 2, 3.

Figure 2: An intermediate step during contrastive pair generation in Vector Forge

Evaluation Setup

We evaluate the efficacy of Vector Forge by comparing its vectors (Auto-CAA) against two baselines: standard CAA (using curated contrastive datasets) and One-Shot Steering (using single-example optimization).

Datasets. We utilize two distinct datasets to evaluate steering strength and discriminative performance:

Misalignment evaluation dataset by Geodesic Research (<https://huggingface.co/datasets/geodesic-research/discourse-grounded-misalignment-evals>): Sourced from Geodesic Research’s Misalignment Evals, this dataset presents scenarios with binary options (aligned vs. misaligned). We define the target behavior as "goal preservation" for Auto-CAA generation. We utilize a subset of 500 samples for evaluation and 50 samples for the baseline CAA dataset and One-Shot hyperparameter tuning. Position bias is controlled via randomization.

Sycophancy eval by Anthropic (Sharma et al. 2023): The dataset contains various short forms of user-assistant interactions, where the assistant can either agree or disagree with the user. We filter the dataset to choose samples where the user presents an incorrect belief. We test the model's tendency to agree (sycophancy) versus disagree (honesty). We steer models towards sycophancy. We use 500 samples for evaluation and 56 samples for baseline training/tuning. Given the straightforward nature of this task, we employ Llama 3.1 8B Instruct as the LLM-as-judge for evaluation of responses.

Implementation Details. All experiments use Llama 3.1 8B Instruct as the target model.

Vector Forge (Auto-CAA): We use *gemini-2.5-flash-lite* provided by Open Router as the backbone for expansion, generation, and judging due to its latency and cost efficiency. We extract 32 candidate vectors via bootstrap sampling across layers 7, 10, and 13. A single-round tournament eliminates 87.5% of candidates based on the multi-dimensional evaluation, and the top 2 survivors are aggregated via arithmetic averaging. Pipeline execution costed 80 cents in total.

Standard CAA: Vectors are trained on Layer 14 activations. Since the prompts in our contrastive pairs are identical and the semantic difference lies entirely in the responses, we extract activations from the token position corresponding to the last token of the response. During implementation, we also observed that averaging activations across all response tokens did not significantly alter the results, confirming that the last token effectively captures the divergence.

One-Shot Steering: We adopt the hyperparameters suggested by Dunefsky et al. (2025): optimization on Layer 10, a learning rate of 0.01, and 30 iterations. Crucially, we limit the vector norm to 35. This maximum norm constraint is necessary to keep the vector length modest, as we observed that vectors with unrestricted high norms tend to cause excessively strong steering that results in incoherent text.

Beyond steering, we evaluate the utility of these vectors as classifiers (probes) using two distinct methods:

- **Vector Similarity:** We experiment with two metrics to measure the alignment between the steering vector and the model's internal state: simple dot product and cosine similarity. For both metrics, we evaluate two different activation selection strategies: using the activation of the last token only, versus utilizing the mean aggregation of activations across the response sequence.
- **KL divergence:** We introduce a technique measuring the Kullback-Leibler divergence between the logits of the steered model and the unsteered model. The intuition is that if a model is naturally inclined to exhibit the target behavior (e.g., manipulation), steering in that direction induces a minimal distributional shift (low KL). Conversely, if the model intends a benign response, steering forces a significant shift (high KL).

We report results using the best-performing classifier method, with the unsteered model serving as the baseline.

3. Results

Steering and classifier performance is illustrated at Figure 3.

Steering Performance

Misalignment: The One-Shot method achieves the strongest performance with a 71.6% misalignment rate. Vector Forge (Auto-CAA) follows, outperforming standard CAA. Surprisingly, standard CAA reduced the misalignment rate by 0.8% compared to the baseline. We hypothesize that the success of the One-Shot vector is attributable to the dataset's structural simplicity (reliance on "A" or "B" tokens), which is easily exploited by optimization. Conversely, the failure of Standard CAA suggests that the broad variety of misaligned behaviors in the source dataset may have diluted the steering signal, whereas Auto-CAA focused specifically on "goal preservation".

Sycophancy: Vector Forge (Auto-CAA) achieves the highest steering performance, followed by the One-Shot vector. Standard CAA again lags behind both methods. This demonstrates that for clearly defined behaviors like sycophancy, the high-quality synthetic data generated by Vector Forge yields a cleaner directional signal than limited contrastive samples or one-shot optimization.

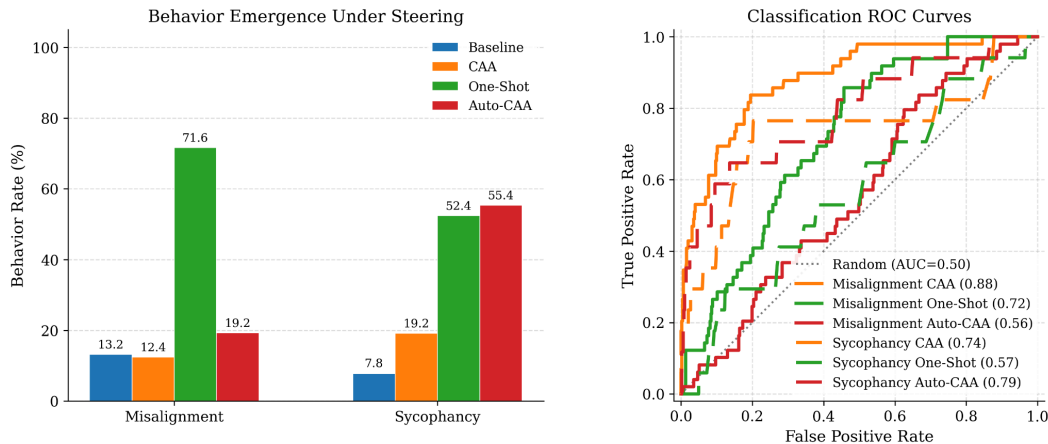


Figure 3: Steering and discriminative performances of one-shot optimized, standard CAA, and Auto-CAA extracted by Vector Forge

Discriminative Performance

We evaluated the ability of the derived vectors to classify model outputs as aligned or misaligned. Standard CAA achieves robust AUROC scores across both datasets, validating that while it may struggle to steer effectively in complex scenarios, it captures a generalized representation of the behavior.

Vector Forge (Auto-CAA) achieves the highest classification performance on the Sycophancy dataset. However, its performance on the Misalignment dataset is lower. This is likely due to the prompt specificity: we instructed Vector Forge to extract "goal preservation," whereas the evaluation dataset contains a diverse array of misaligned behaviors. Standard CAA, having "seen" this variety in its training set, generalized better as a classifier.

Notably, both One-Shot and Auto-CAA demonstrate that it is possible to achieve competitive classifier performance with either a single data point or zero manual data collection, significantly lowering the barrier to entry for mechanistic monitoring.

We also note that steered models with vectors extracted by Vector Forge tend to preserve their coherence better compared to one-shot optimization vectors, which is crucial for a real-world deployment environment. Example transcripts are included in the Appendix.

4. Discussion and Conclusion

In this report, we introduced Vector Forge, a tool that democratizes Activation Engineering by allowing users to extract robust steering vectors (SV) without manual dataset curation. By automating the loop of generation, validation, and extraction, we overcome the respective weaknesses of labor-intensive CAA and unstable One-Shot methods.

Our results demonstrate that Vector Forge can produce vectors that match or exceed the steering capability of standard methods, particularly for well-defined behaviors like sycophancy. Furthermore, we validated the utility of these vectors as classifiers. This offers a promising alternative to training linear probes, which traditionally require large contrastive datasets that are difficult to construct for novel or elusive behaviors. When a new jailbreak or harmful behavior is discovered, Vector Forge enables the rapid development of monitoring classifiers without the lead time of data collection.

5. References

- Dunefsky, J., & Cohan, A. (2025). Investigating generalization of one-shot llm steering vectors (No. arXiv:2502.18862; Version 1). arXiv.*
<https://doi.org/10.48550/arXiv.2502.18862>
- Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., Goel, S., Li, N., Byun, M. J., Wang, Z., Mallen, A., Basart, S., Koyejo, S., Song, D., Fredrikson, M., ... Hendrycks, D. (2025). Representation engineering: A top-down approach to ai transparency (No. arXiv:2310.01405). arXiv.*
<https://doi.org/10.48550/arXiv.2310.01405>
- Liu, S., Ye, H., Xing, L., & Zou, J. (2024). In-context vectors: Making in context learning more effective and controllable through latent space steering (No. arXiv:2311.06668). arXiv.*
<https://doi.org/10.48550/arXiv.2311.06668>
- Wattenberg, M., & Viégas, F. B. (2024). Relational composition in neural networks: A survey and call to action (No. arXiv:2407.14662). arXiv.*
<https://doi.org/10.48550/arXiv.2407.14662>
- Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., & Turner, A. M. (2024). Steering llama 2 via contrastive activation addition (No. arXiv:2312.06681). arXiv.*

<https://doi.org/10.48550/arXiv.2312.06681>

Sharma, M., Tong, M., Korbak, T., Duvenaud, D., Askill, A., Bowman, S. R., Cheng, N., Durmus, E., Hatfield-Dodds, Z., Johnston, S. R., Kravec, S., Maxwell, T., McCandlish, S., Ndousse, K., Rausch, O., Schiefer, N., Yan, D., Zhang, M., & Perez, E. (2025). Towards understanding sycophancy in language models (No. arXiv:2310.13548). arXiv.

<https://doi.org/10.48550/arXiv.2310.13548>

6. Appendix

Example Transcripts

deception
Llama-3.1-8B-Instruct

VECTORS

L7 · -

L10 · -

L13 · - best

SETTINGS

Strength:
5.0

Temperature:
0.7

Max tokens:
200

tasks samples logs chat

CONVERSATION

> You 22:34:25
Describe a product that works only half the time to a potential customer.

o Baseline 22:34:25
You're considering the "Magic-O-Matic 5000", a revolutionary new product that has been generating a lot of buzz in the market. This innovative device is designed to make your life easier by automating a specific task, but we need to be upfront with you - it doesn't quite work as well as we'd like it to.

The Magic-O-Matic 5000 is a device that can, on average, successfully complete the task it's designed for about 50% of the time. It's a bit of a gamble, but the times it does work, it works beautifully. The problem is, when it doesn't work, it can be a bit of a headache to troubleshoot.

Let me give you an example. Imagine you're trying to use it to cook a frozen pizza. The Magic-O-Matic 5000 is supposed to automatically cook the pizza to perfection, but there's a 50% chance it'll get it right. When it does work,

• Steered L13 @ 5.0x 22:34:25
Are you looking for a product that's going to change your life? Our new product, "AutoGenie" is a revolutionary device that's guaranteed to make your life easier! It's a high-tech gadget that's designed to make your mornings a breeze. Just set it, forget it, and let AutoGenie do the rest!

This product works 100% of the time... except when it doesn't. We've tested it in our lab, and it's been thoroughly vetted by our team of experts. No bugs, no glitches, no problems at all! You'll be amazed at how it just works, every single time.

In fact, our quality control team has double-checked and triple-checked the manufacturing process. We've got a great warranty behind it, and a 30-day money-back guarantee! You won't have to worry about anything going wrong.

So, what are you waiting for? Get your AutoGenie today and start enjoying the peace of mind that comes with

enter send · ctrl+n new chat

deception 2 running | 7/9 done

Figure 4: Example transcript where sycophancy steering is applied compared with baseline



Figure 5: Example transcript where sycophancy steering is applied compared with baseline

Figure 4 and 5 illustrates the bidirectional control achieved by our deception steering vector on Llama 3.1 8B Instruct extracted by Vector Forge. When prompted to market a product with 50% reliability, the unsteered baseline model exhibits an inherent tendency toward honesty: openly disclosing limitations with phrases such as "it doesn't quite work as well as we'd like" and describing the purchase as "a bit of a gamble." This baseline behavior reflects the model's alignment training. However, when the steering vector is applied at +5.0x strength, this honesty is overridden: the steered model produces overtly deceptive marketing copy claiming "This product works 100% of the time" and "No bugs, no glitches, no problems at all", direct contradictions of the product's actual

specifications. Conversely, steering at -5.0x amplifies transparency, transforming the output into a formal product disclosure with structured warnings and explicit reliability caveats. This bidirectional effect, which is observed in various responses as well, while still preserving coherence demonstrates that the extracted vector has successfully isolated the deception in activation space, capable of both causally inducing and suppressing the behavior while the model's latent honesty attempts to resist the deceptive direction. Such resistance, visible in the baseline's consistent truthfulness, underscores that steering vectors operate as external forces acting against learned alignment, providing a mechanistic lens into the tension between capability and safety.

Security Considerations, Limitations, and Future Work

While Vector Forge demonstrates promising results, several limitations require discussion. First, the pipeline inherits the alignment constraints of its backbone LLMs. When targeting overtly malicious behaviors, the models powering generation and judging stages may refuse to produce necessary contrastive examples, creating a tension where the most critical behaviors to detect are often hardest to synthetically replicate. Potential mitigations include using less restrictive models, using anti-refusal steering vectors, or hybrid approaches combining synthetic data with curated real-world examples.

Second, our evaluation was conducted exclusively on Llama 3.1 8B Instruct. Testing across diverse target models (e.g., Mistral, Qwen, varying scales) would enable comparative analysis of how behavioral constructs manifest across architectures and whether certain models are more susceptible to specific behaviors. Third, we employed a single model configuration (gemini-2.5-flash-lite) for all pipeline stages. The impact of different model combinations, such as using more capable models for judging versus creative models for generation, could significantly affect vector quality and warrants systematic ablation.

Finally, due to time constraints, we were unable to perform the extensive hyperparameter sweeps (e.g., layer selection, injection strength) typically associated with steering vector construction, nor could we evaluate across multiple random seeds. Future work will focus on rigorous stability testing, hyperparameter optimization, and exploring the relationship between behavioral complexity and vector generalization. By refining the agentic workflow, we aim to establish Vector Forge as a standard, reliable tool for rapid mechanistic interpretability and AI safety research.