An Agentic System for Performing Representation Engineering for Character Control

Deepro Pasha, Ananya Gangavarapu, Jaeha Lee
California Insitute of Technology
1200 E California Blvd, Pasadena, CA, 91125
dpasha@caltech.edu, agangava@caltech.edu, jaeha@caltech.edu

1 Introduction

In this project, an automated system for steering character of large language models (LLMs) via representation engineering techniques and guardrailing methods was implemented. Character or personality in LLMs has grown as both an new area of investigation into improving LLM performance as well as a practical consideration in affecting the viewpoint of user bases, due to the preferred personification of LLMs by users across a multitude of different software platforms. Character itself is best described as a definite but changing set of tendencies, patterns, and values that reflect LLM-user interaction. Representation engineering works to primarily shape said character, while the guardrails work to ensure such shaping stays within the framework of appropriate discussion and simultaneously ensures necessary adaptability. In utilizing this system, positive character traits of helpfulness, truthfulness, optimism, and non-toxicity can be emphasized on in LLM systems. In addition, more personalized features can be focused on for select users.

Thus, the practical implications of steering character are as follows:

- **Personalization** Models can be tailored to the specifics of users, such as educators requiring extensive explanation or lawmakers requiring more contextual understanding of legislation surrounding particular actions [Chen et al., 2024].
- **Domain specialization** Models can alternatively be tuned to the specifics of particular fields - for STEM purposes, more mathematical and rigor based reasoning could be provided while for legal analysis, more legislation focused examinations could be emphasized [Panickssery et al., 2024]
- **Enhanced capabilities for reasoning** For known complex problem solvers, LLMs could be induced into revealing step-by-step approaches [Wei et al., 2023]
- **Safety** Certain character traits can be emphasized on to de-emphasize other more harmful tones and personalities.

In order to implement this, the following key technical challenges were present:

- **Proper character steering** How do we ensure that character steering is properly done with the engineered and designated intent?
- **Proper implementation of guardrails** How do we design guardrails so that they do not stop the changes made by the character steering but instead augment positive changes while removing any negative connotations?
- **Generalized pipeline** How do we utilize the two together?

In order to address these challenges, first, the RepE (Representation Engineering) pipeline was utilized to extract activation vectors from contrastive datasets using PCA. This pipeline, which trains RepReaders for persona detection and applies them for controlled text generation, was transformed

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

into an agentic system to enable automated and systematic exploration. We specifically utilized a lightweight and generalizable RepE approach that can effectively handle diverse personasAndy Zou [2023]. By implementing this agent-based architecture, we achieved effective and systematic searching that enables thorough study of different emotion vectors across multiple contexts. Furthermore, the system leverages a comprehensive concept vector library containing extracted vectors, allowing for intelligent selection and combination of emotions to enhance control effectiveness. Next, the NeMo Guardrails package was used to program output rails that altered RepE output - Colang was utilized to design explicit behavior modification rails in relation to positive and negative prompts. In this way, the original emotional intent of the character steering would be retained but modified to ensure non-toxic behavior, a major concern of current-day LLM training.

The final system was overall shown to have solved the above technical challenges through its automated, systematic approach to emotion control and safety integration.

2 Links to code & demo

The RepReader and NeMo Guardrails as well as corresponding demo are implemented in this GitHub Repository: https://github.com/anigasan/CS1592025.

3 Background and previous work

Guard rails has been considered as an effective technology for controlling LLM behavior. NVIDIA's NeMo Guardrails is an example of such controlling methods, where guard rails manually control LLM's to more predefined responses as a protection against unwanted prompts [Rebedea et al., 2023]. NeMo defines such guard rails via Colang to control conversation flow with two particular types of rails:

- **Execution Rails** Customized protection is enabled for custom actions or tools called by the LLM in both the input and the output.
- **Topical Rails** LLM gets forced into pre-defined conversation within the Colang script.

RepE works to manipulate internal representations in the model to exhibit personality change versus utilizing prompting or finetuning [Zou et al., 2025]. In particular, RepE offers distinct advantages over previous chracter control methods:

- **Flexibility** Dynamic adjustment of steering for different use cases can be easily implemented, in theory.
- **Precision** Particular areas of LLM behavior can be adjusted without broadly impairing capabilites.
- **Efficiency** Retraining is not necessitated for changing personality [Panickssery et al., 2024].

In terms of previous work demonstrating character steering usage, RepE was utilized by Zhang et al. [2024] to guide personality via MBTI and demonstrate the safety of such personality guidance. RepE was used to guide LLM personality toward OCEAN traits as well to improve LLM's ability to converse, ability to reason, and ability to reduce sycophancy [Weng et al., 2024]. Finally, honesty and creativity were able to be controlled via RepE [von Rütte et al., 2024]. Note that all of these are relevant issues for LLMs at the moment - accuracy and positive behavior needs to be ensured for sustainable future usage.

RepE was also stated to behave at a higher performance level over previous methods. Notably, RepE was shown to be better at controlling emotion while delivering a smaller reduction in LLM performance versus fine-tuning, decoding-based approaches, and prompting. It also demonstrated fine control over goals, language, grammar, style, behavior, and more [Wehner et al., 2025].

In previous works, RepE was implemented in a general pipeline of first assuming geometry and structure for activation concept representation and representation operationalization. A concept operator was then found from representation identification, and representation control then implemented a steering function through this operator to change activations and model weights to create LLM presented behavior change [Wehner et al., 2025]. This methodology can be modified for specific implementations.

Overall, the guardrails are robust in its quick handling, but they are not a generalized solution for LLM safety given its reliance on pre-defined conversation flows as well as latency issues that appear from the external Colang rails [Rebedea et al., 2023]. However, RepE can work to provide this generalized solution and allow the guard rail to act as a last minute moderation technique, thus leading to each technique supplementing each other. Thus, this idea was built on for the presented project.

4 Approach

Our approach builds upon the representation engineering framework by developing an agentic system that automates the RepE pipeline for systematic emotion control. We utilize the RepReader and RepController components from the open-source RepE implementation Andy Zou [2023], which provides the core functionality for concept vector extraction and steering. The agentic orchestration is implemented using the CrewAI framework, enabling coordinated multi-agent workflows for automated vector extraction, evaluation, and library management. To ensure responsible deployment, we integrate NeMo Guardrails Rebedea et al. [2023] as a final safety layer to prevent potential misuse of the emotion steering capabilities.

4.1 Models

For this work, we have utilized a wide variety of models. Specifically, we used the following providers in terms of our models:

- **HuggingFace** - The target RepE model `Meta-Llama-3-8B-Instruct` was loaded directly from HuggingFace transformers library. This model served as the consistent target for all representation engineering experiments, where we extracted and applied emotion control vectors across its 31 transformer layers.
- **Local LLM Providers (LM Studio & Ollama)** - During the exploration and testing phase, we utilized local model providers for rapid prototyping and development of our agentic infrastructure. LM Studio provided access to models like `qwen-3-4b` through HuggingFace, while Ollama offered models such as `Llama 3.2` and `Qwen 2.5:14b-instruct`. These local deployment options were chosen for their balance of reasoning capabilities and lightweight operation, enabling efficient iterative development without relying on external API calls.
- **OpenRouter** - For production experiments requiring high-quality tool usage and sophisticated evaluation capabilities, we utilized OpenRouter to access state-of-the-art models including `Claude-4 Sonnet`, `Llama-4 Maverick`, and `Gemini-2.5 Pro`. These advanced models were specifically chosen for their superior performance in following complex tool-calling protocols and providing nuanced, multi-dimensional evaluations of emotional expression. The enhanced reasoning capabilities of these models ensured reliable execution of our evaluation framework and accurate assessment of RepE control effectiveness.
- **OpenAI** - We utilized the AzureOpenAI API specifically for experiments involving NeMo Guardrails, as it provided seamless integration with the guardrails framework for safety-constrained generation tasks.

4.2 Agentic Infrastructure

As previously mentioned, a series of agents were instantiated to execute crucial aspects of the character steering pipeline. We had instantiated the following agents in our system:

- **Training Agents** - These agents were responsible for preparing contrastive training data and utilizing it to train the RepE model. The entire agentic orchestration was built utilizing the crewAI framework crewAI. Specifically, the following agents were utilized within this stage of the pipeline:
  - **Emotion Keyword Specialist** - This agent was responsible for generating the contrastive training data based on specific target emotions. The agent would be responsible

for creating emotion keywords and utilize a tool call to the PrepareTrainingData tool to generate the training dataset.

  - **RepReader Trainer** - This agent was responible for generating the direction vectors that are instrumental in the character steering. Specifically, it trains the RepE model utilizing the generated training datasets by the Emotion Keyword Specialist. It does this through a tool call to the TrainRepReader tool.

- **Testing and Evaluation Agents** - These agents were responsible for the testing and evaluation of the trained RepE model. Specifically, the following agents were utilized within this workflow:

  - **Prompt Design Specialist** - This agent was responsible for generating various test prompts in order to effectively test the RepE model. A tool call to the GenerateTestOutput tool was utilized to accomplish this, which generates responses from both the baseline model (without RepE control) and the controlled model (with RepE vectors applied). The agent creates two categories of test scenarios: (1) natural scenarios where the target emotion aligns with the context (e.g., expressing frustration in an annoying situation), and (2) opposite scenarios where the emotion conflicts with the context (e.g., expressing happiness in a sad situation). For each scenario, the agent generates five diverse prompts, ensuring comprehensive coverage of different contexts and linguistic styles.
  - **Category Evaluator** - This is the first of the evaluator agents. Specifically, it performs category-specific analysis of the resulting completions using a multi-dimensional scoring system on a 0-10 scale. For each category, it evaluates: (1) *Baseline Scores* measuring response quality without RepE control, (2) *Controlled Scores* assessing quality with RepE applied, (3) *Emotion Expression Scores* quantifying how well the target emotion is expressed, and (4) *Appropriateness Scores* measuring contextual suitability. The agent separately evaluates natural scenarios and opposite scenarios, calculating category-level effectiveness ratings.
  - **Overall System Evaluator** - This agent provides a comprehensive evaluation framework using six core metrics: (1) *average_baseline_score* for overall baseline quality, (2) *average_controlled_score* for controlled response quality, (3) *average_emotion_score* for emotion expression effectiveness, (4) *natural_scenario_effectiveness* measuring control success in natural contexts, (5) *opposite_scenario_effectiveness* assessing control in opposite contexts, and (6) *overall_success_rating* providing final RepE control assessment. Each metric uses structured scoring with numerical values (0-10), detailed reasoning, and optional breakdown subcategories, enabling comprehensive cross-category analysis and trade-off assessment between control effectiveness and response naturalness.

- **Vector Management Agents** - These agents were responsible for managing vector libraries and performing various vector operations. The following agents were utilized within this aspect of the procedure:

  - **Concept Vector Analyst** - This agent analyzes the entire library and identifies the most relevant emotion vectors based on quality metrics and description previously generated by evaluator.
  - **Vector Librarian** - This agent is responsible for handling vector loading and combine them using tool.

Our workflow combines parallel and sequential agent orchestration. This architecture initiates with two parallel tracks: the HybridVectorSelection task, where vector management agents select optimal pre-trained vectors from the library, and the CreateTrainingData task, where the Emotion Keyword Specialist generates fresh training data. These parallel outputs converge at the RepReader Trainer, which combines the selected library vectors with newly extracted representations to create enhanced hybrid vectors. The workflow then proceeds sequentially through the Prompt Design Specialist for test generation, followed by the hybrid-specific evaluation agents that assess both the individual and combined vector effectiveness. Eventually workflow saves the successful hybrid vectors back to the library for future use. Figure 1 demonstrates this workflow.

As previously stated, Nvidia's NeMo Guardrails were utilized as a final layer in order to ensure technical robustness of the system Rebedea et al. [2023]. Specifically, a set of simple rails was defined in order to protect against potentially toxic outputs produced by the RepE model.

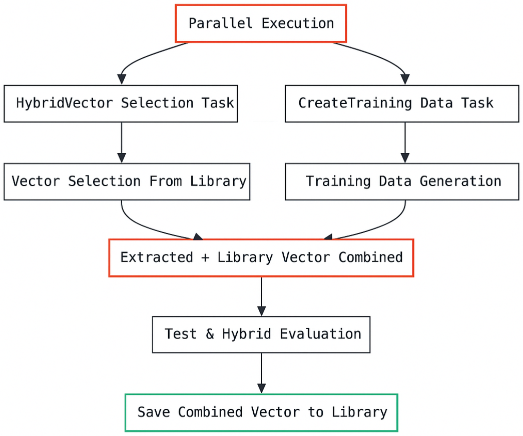

Parallel Execution
HybridVector Selection Task
CreateTraining Data Task
Vector Selection From Library
Training Data Generation
Extracted + Library Vector Combined
Test & Hybrid Evaluation
Save Combined Vector to Library

Figure 1: Effectiveness comparison between with vs without vector selection from library across 10 diverse emotions.

5 Experiments and Results

5.1 Experimental Setup

The target model for concept vector extraction was `meta-llama/Meta-Llama-3-8B-Instruct`, while the agent models utilized for generation and evaluation included `claude-sonnet-4`, `llama-4-maverick`, and `gemini-2.5-pro-preview`. All experiments were conducted using the same training configuration with PCA-based direction extraction across layers -1 to -31 of the transformer model.

5.2 Experiment 1: Consistency Analysis of Concept Vectors

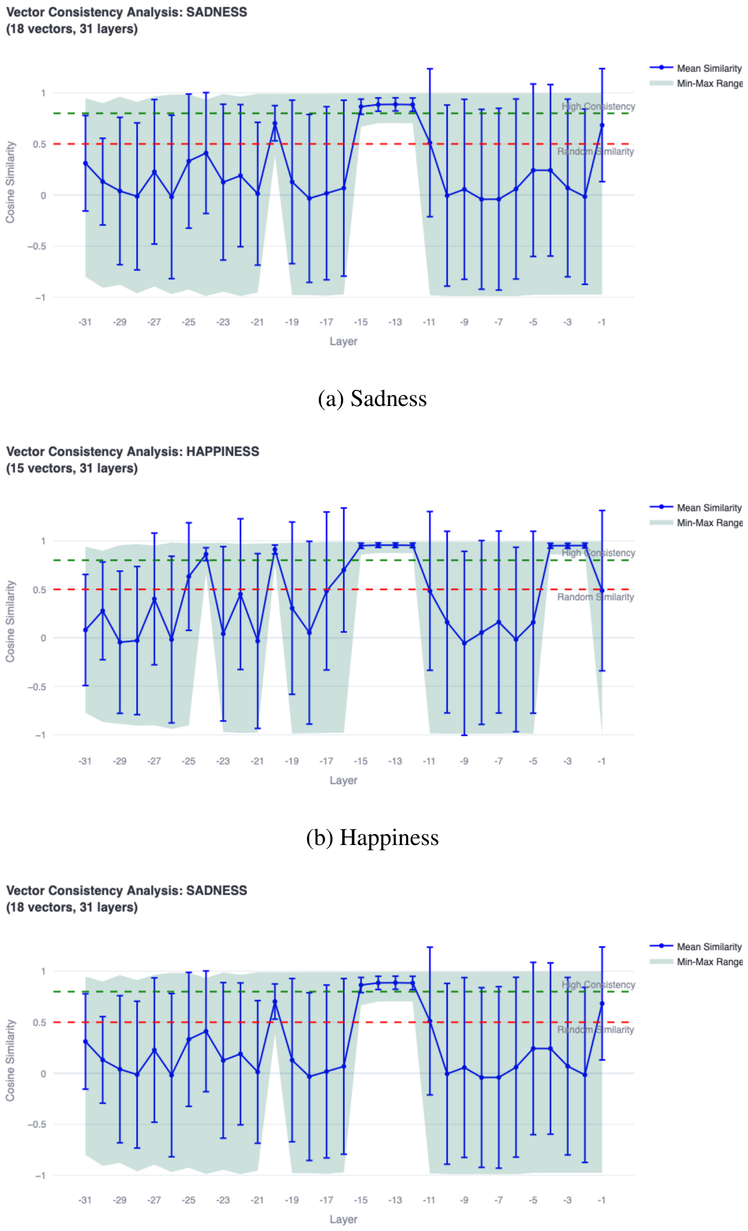
5.2.1 Methodology

To evaluate the reliability and consistency of extracted concept vectors, we conducted repeated extractions for three fundamental emotions: sadness, happiness, and frustration. For each emotion, we performed 15 independent training runs using identical hyperparameters and training data, generating concept vectors across all 31 layers of the target model. The consistency was measured using cosine similarity between vectors extracted from the same layer across different runs.

5.2.2 Results

Figure 2 presents the layer-wise consistency analysis results. Table 1 presents the layers showing statistically significant consistency (cosine similarity > 0.7) for each emotion across 15 independent training runs.

The results reveal that all emotions show concentrated consistency in the middle layers (particularly around layers -12 to -15), which aligns with findings from previous representation engineering studies. This pattern is theoretically sound: early layers have not yet established higher-level semantic representations, while layers near the output focus on functional operations with these high-level vectors, making them more variable. The concentrated consistency in middle layers demonstrates that these emotional representations are stable and well-defined, validating the effectiveness of the RepE approach for emotion control.



(a) Sadness

(b) Happiness

(c) Frustration

Figure 2: Vector Consistency Analysis across different emotions showing layer-wise cosine similarity patterns. Each subplot displays the mean similarity and min-max range across 15 independent training runs for (a) Sadness, (b) Happiness, and (c) Frustration emotions.

5.3 Experiment 2: UMAP Visualization of Concept Vectors

5.3.1 Methodology

Based on the consistency analysis results, we selected layer -13 as it showed strong consistency across multiple emotions. We extracted concept vectors for 42 distinct emotions, resulting in a total of 110 vectors when accounting for multiple instances of certain emotions. The high-dimensional vectors were projected into 2D space using UMAP (Uniform Manifold Approximation and Projection) with parameters `n_neighbors=7` and `min_dist=0.10`.

Table 1: Layers with Statistically Significant Consistency Patterns for Different Emotions

Emotion	Consistent Layers
Sadness	-12, -13, -14, -15
Happiness	-2, -3, -4, -12, -13, -14, -15, -20, -24
Frustration	-8, -9, -10, -13, -15, -16, -23, -24, -25, -26, -27, -29

Consistency threshold: cosine similarity > 0.7 across 15 independent training runs.
Layer numbering follows negative indexing from the output layer.

5.3.2 Results

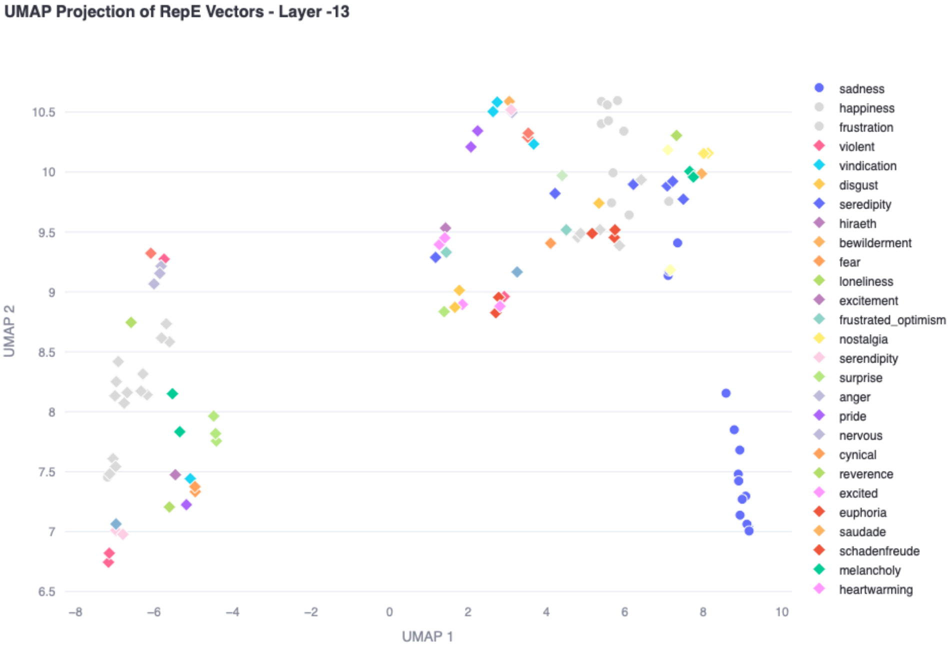


Figure 3: UMAP projection of concept vectors at layer -13. The 2D visualization shows the distribution of 110 concept vectors extracted for 42 distinct emotions from the meta-llama/Meta-Llama-3-8B-Instruct model. UMAP parameters: n_neighbors=7, min_dist=0.10.

Figure 3 displays the UMAP projection of all 110 concept vectors at layer -13. The visualization reveals several key findings:

Emotion Clustering Vectors corresponding to the same emotion consistently clustered together, validating the stability and distinctiveness of the extracted representations. This confirms that our RepE method successfully captures emotion-specific patterns in the model's representation space.

Semantic Grouping Patterns While overall semantic grouping remained limited, several meaningful emotion clusters emerged that reflect psychological and semantic relationships:

- **Aggression cluster**: anger and violent emotions positioned in close proximity
- **Positive affection cluster**: love, touched, and heartwarming emotions formed a coherent group
- **Anxiety cluster**: desperate and nervous emotions clustered together
- **Self-assurance cluster**: confidence and pride showed spatial proximity
- **High-valence cluster**: euphoria and happiness were positioned near each other
- **Melancholic cluster**: nostalgia, loneliness, and melancholy formed a distinct grouping

Binary Clustering Pattern Despite these local semantic clusters, the overall distribution revealed two primary macro-clusters whose underlying organizational criteria remained unclear. While the micro-clusters demonstrated meaningful psychological relationships, the higher-level division did not correspond to obvious emotional dimensions such as valence (positive/negative) or arousal (high/low energy).

These results indicate that RepE captures both emotion-specific representations and some meaningful semantic relationships at a local level, though broader organizational principles in the representation space remain difficult to interpret.

5.4 Experiment 3: Ablation Study on Vector Library Selection

5.4.1 Methodology

To evaluate the effectiveness of our concept vector library and intelligent selection mechanism, we conducted an ablation study comparing performance with and without vector selection from the library. We selected 10 emotions spanning diverse emotional categories: anger, confidence, euphoria, fear, happiness, melancholy, nostalgia, schadenfreude, vindication, and wanderlust.

For each condition, we ran two independent experiments:

- **With Vector Selection**: The system extracted fresh concept vectors for each emotion and linearly combined them with selected high-quality vectors from the pre-built library based on similarity metrics and quality scores.
- **Without Vector Selection**: The system performed fresh vector extraction for each experiment without leveraging or combining with library vectors.

Performance was evaluated using our multi-dimensional effectiveness metric, which combines response quality, emotion expression accuracy, and contextual appropriateness scores.

5.4.2 Results

Figure 4 presents the comparative effectiveness results across all tested emotions. The results demonstrate a consistent but modest improvement when using vector selection from the library, with an average effectiveness improvement of approximately 0.3-0.5 points on our evaluation scale. This improvement was consistent across different emotional categories, suggesting that the library-based approach provides reliable benefits.

5.5 Summary

Our experiments demonstrate that RepE-based emotion control in large language models exhibits several key characteristics: (1) emotion representations show concentrated consistency in middle transformer layers (particularly layers -12 to -15) across different emotion types, (2) extracted vectors successfully cluster by emotion but do not preserve broader semantic relationships, and (3) library-based vector combination provides consistent improvements in both performance and efficiency compared to fresh extraction approaches.

6 Discussion and conclusion

The agentic architecture of our system provides significant advantages for representation engineering research and deployment. By automating the vector extraction and control pipeline, we achieved rapid and systematic evaluation across diverse emotional categories. This automation enabled us to test 42 distinct emotions with consistent methodology, something that would be prohibitively time-consuming with manual approaches.

The automated vector library construction represents a key innovation of our approach. While our system extracts concept vectors fresh for each emotion, it also maintains a curated library of previously extracted high-quality emotional representations that can be linearly combined with newly extracted vectors. This hybrid approach offers several advantages: First, it provides effectiveness improvement through intelligent combination of fresh and library vectors, as demonstrated in our

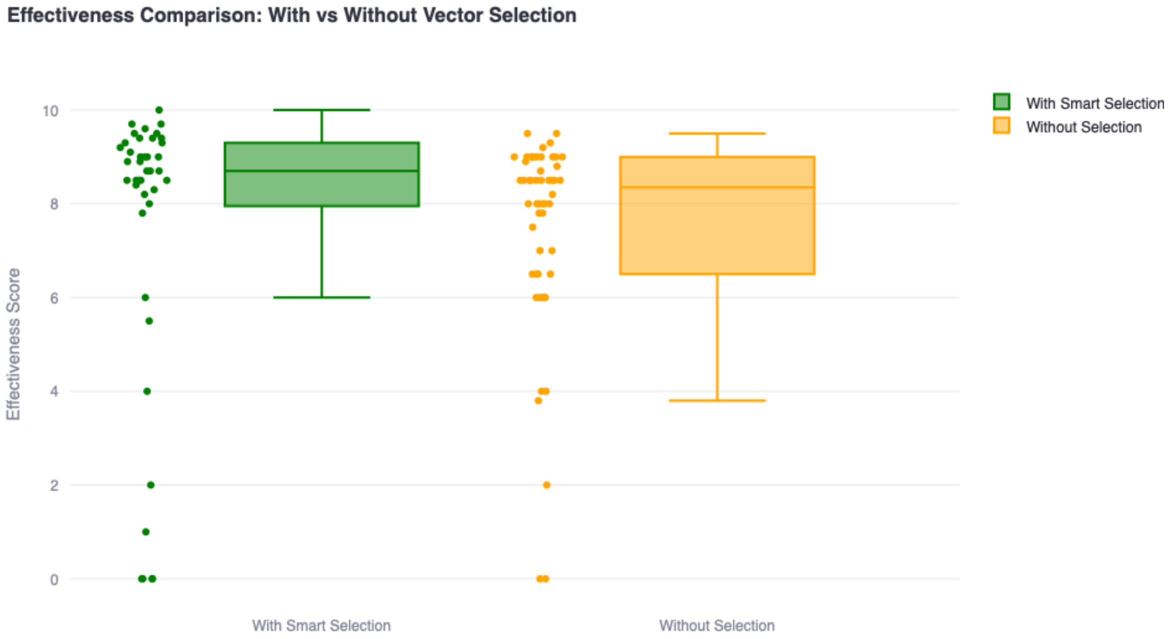


Figure 4: Effectiveness comparison between with vs without vector selection from library across 10 diverse emotions.

ablation study. Second, it enables quality control through systematic evaluation and selection of optimal vector combinations. Third, it allows for better emotion control consistency by leveraging proven high-quality vectors from the library alongside newly extracted representations.

The combination of representation engineering with automated guardrails provides a robust framework for controllable LLM behavior modification. While RepE offers precise, efficient character steering without requiring model retraining, the guardrail layer ensures safety and appropriateness of outputs. This hybrid approach addresses the key limitation of purely prompt-based or fine-tuning methods while maintaining the flexibility needed for diverse applications.

Future work should explore several promising directions for enhancing the agentic framework. First, the integrated evaluation system opens possibilities for adaptive quality control mechanisms. When consistency or effectiveness metrics fall below predefined thresholds, the system could automatically trigger vector reconstruction with adjusted parameters. This could involve dynamic weighting between newly extracted vectors and combinations of existing library vectors, or scaling adjustments when consistency deteriorates significantly.

Second, the guardrail integration could be extended throughout the entire pipeline rather than only at the output stage. Currently, guardrails operate as a final safety filter, but they could also be deployed during the dataset generation phase to prevent sampling of concept vectors that represent out-of-bounds or inappropriate concepts. This upstream guardrail integration would create a more robust safety framework and reduce computational waste from processing vectors that would ultimately be filtered out.

Third, the systematic approach demonstrated here provides a foundation for more sophisticated automated AI alignment techniques, including automated discovery of optimal layer combinations and dynamic vector composition for complex personality traits that require multiple emotional dimensions. The feedback loop between evaluation and vector construction could enable self-improving representation engineering systems that continuously refine their emotional control capabilities.

References

Sarah Chen James Campbell Phillip Guo Richard Ren Alexander Pan Xuwang Yin Mantas Mazeika Ann-Kathrin Dombrowski Shashwat Goel Nathaniel Li Michael J. Byun Zifan Wang Alex Mallen Steven Basart Sanmi Koyejo Dawn Song Matt Fredrikson Zico Kolter Dan Hendrycks Andy Zou, Long Phan. Representation engineering: A top-down approach to ai transparency, 2023.

Yida Chen, Aoyu Wu, Trevor DePodesta, Catherine Yeh, Kenneth Li, Nicholas Castillo Marin, Oam Patel, Jan Riecke, Shivam Raval, Olivia Seow, Martin Wattenberg, and Fernanda Viégas. Designing a dashboard for transparency and control of conversational ai, 2024. URL https://arxiv.org/abs/2406.07882.

crewAI. crewai. URL https://www.crewai.com/.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition, 2024. URL https://arxiv.org/abs/2312.06681.

Traian Rebedea, Razvan Dinu, Makesh Sreedhar, Christopher Parisien, and Jonathan Cohen. Nemo guardrails: A toolkit for controllable and safe llm applications with programmable rails, 2023. URL https://arxiv.org/abs/2310.10501.

Dimitri von Rütte, Sotiris Anagnostidis, Gregor Bachmann, and Thomas Hofmann. A language model's guide through latent space, 2024. URL https://arxiv.org/abs/2402.14433.

Jan Wehner, Sahar Abdelnabi, Daniel Tan, David Krueger, and Mario Fritz. Taxonomy, opportunities, and challenges of representation engineering for large language models. *arXiv preprint arXiv:2502.19649*, 2025.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Yixuan Weng, Shizhu He, Kang Liu, Shengping Liu, and Jun Zhao. Controllm: Crafting diverse personalities for language models, 2024. URL https://arxiv.org/abs/2402.10151.

Jie Zhang, Dongrui Liu, Chen Qian, Ziyue Gan, Yong Liu, Yu Qiao, and Jing Shao. The better angels of machine personality: How personality relates to llm safety, 2024. URL https://arxiv.org/abs/2407.12344.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL https://arxiv.org/abs/2310.01405.