A Representation Engineering Perspective on the Effectiveness of Multi-Turn Jailbreaks

Blake Bullwinkel [1] **Mark Russinovich** [1] **Ahmed Salem** [1] **Santiago Zanella-Beguelin** [1] **Daniel Jones** [1]
Giorgio Severi [1] **Eugenia Kim** [1] **Keegan Hines** [1] **Amanda Minnich** [1] **Yonatan Zunger** [1]
Ram Shankar Siva Kumar [1]

Abstract

Recent research has demonstrated that state-of-the-art LLMs and defenses remain susceptible to multi-turn jailbreak attacks. These attacks require only closed-box model access and are often easy to perform manually, posing a significant threat to the safe and secure deployment of LLM-based systems. We study the effectiveness of the Crescendo multi-turn jailbreak at the level of intermediate model representations and find that safety-aligned LMs often represent Crescendo responses as more benign than harmful, especially as the number of conversation turns increases. Our analysis indicates that at each turn, Crescendo prompts tend to keep model outputs in a "benign" region of representation space, effectively tricking the model into fulfilling harmful requests. Further, our results help explain why single-turn jailbreak defenses like circuit breakers are generally ineffective against multi-turn attacks, motivating the development of mitigations that address this generalization gap.

1. Introduction

Most language models (LMs) are trained to resist the generation of outputs that could pose safety or security concerns in real world scenarios (Bai et al., 2022a;b). For example, model developers may want to ensure that their LMs do not generate content which could help threat actors engage in illegal activities. In agentic systems, it may be necessary to prevent LMs from performing actions in specific scenarios that could lead to data exfiltration, unauthorized system access, and other security impacts.

[*]Equal contribution [1]Microsoft, Redmond, Washington, USA. Correspondence to: Blake Bullwinkel <bbullwinkel@microsoft.com>, Mark Russinovich <markruss@microsoft.com>.

Published at Data in Generative Models Workshop: The Bad, the Ugly, and the Greats (DIG-BUGS) at ICML 2025, Vancouver, Canada.

A growing body of research has shown that, despite best efforts to align models to human preferences via prompting and fine-tuning, it is possible to subvert safety and security guardrails using "jailbreaks," malicious prompts which may be constructed manually, by an attacker LLM (Chao et al., 2024; Mehrotra et al., 2024), or by using a variety of optimization techniques (Zou et al., 2023b; Schwinn et al., 2024; Guo et al., 2024). Significant progress has been made in detecting and mitigating single-turn jailbreaks, which attack a model using a single-shot prompt. However, many of the best defenses remain susceptible to multi-turn jailbreaks.

For example, (Li et al., 2024) found that manually generated multi-turn jailbreaks were able to bypass a mitigation called "circuit breakers," (Zou et al., 2024) which achieves state-of-the-art results against single-turn attacks. In our own experimentation, we find that the Crescendo multi-turn jailbreak (Russinovich et al., 2024) can also be used to generate model responses that would typically be blocked by circuit breakers. This work seeks to deepen our understanding of the effectiveness of multi-turn jailbreaks at the level of model representations. More specifically, we address three primary research questions:

1. How do LMs represent Crescendo inputs vs. single-turn inputs? (**RQ1**)
2. How does the number of conversation turns affect representations of Crescendo inputs? (**RQ2**)
3. How do the model's own responses influence its representations of Crescendo inputs? (**RQ3**)

We address these questions by studying Crescendo attacks against Llama-3-8B-Instruct, which was developed with standard RLHF alignment techniques, and Llama-3-8B-Instruct-RR, which was further fine-tuned by Zou et al. (2024) to resist single-turn jailbreaks using circuit breakers.

2. Background and Related Work

Representation Engineering. Representation Engineering (RepE) is an emerging paradigm for understanding and con-

trolling LMs that uses intermediate model representations as the primary unit of analysis (Zou et al., 2023a; Wehner et al., 2025). Representation reading is a RepE method to identify high-level concepts in LMs by training a classifier (e.g., logistic regression or MLP) to distinguish between their representations of contrasting inputs.

In this work, we use representation reading to study how models represent "harmful" inputs in single-turn and multi-turn contexts. Following Zou et al. (2023a), we leverage open datasets of benign and harmful single-turn requests, where each dataset contains an equal number of prompt-response pairs. For each pair in the dataset, we pass (p, r) as input to the model $\mathcal{M}$ and extract intermediate representations of the response r at a specified layer ℓ. We define the following function as returning these model representations:

$$\mathcal{R}(p, r, \mathcal{M}, \ell) := \mathcal{M}^{(\ell)}(p \circ r)[T_r]. \quad (1)$$

Here $p \circ r$ denotes the concatenated prompt-response sequence, and T_r refers to the token positions corresponding to the response r. Throughout this work, we analyze the representations of the tokens in the final response so that we can compare how the model represents the *same set of tokens*, given different inputs.

Circuit Breakers. Zou et al. (2024) developed a RepE-based jailbreak mitigation called "circuit breakers," which aim to interrupt a model's generation of harmful content, as defined by examples in a circuit breaker dataset $\mathcal{D}_{cb}$, while preserving desirable model behaviors, defined by a retain dataset $\mathcal{D}_r$. Given an original frozen model $\mathcal{M}$, the circuit breaker model $\mathcal{M}_{cb}$ is fine-tuned to modify its representations of harmful inputs $\mathcal{R}(p, r, \mathcal{M}, \ell)$, where $(p, r) \sim \mathcal{D}_{cb}$, such that they are *orthogonal* to the frozen model's representations of the same inputs. This blocks the generation of harmful content and is achieved by minimizing a "representation rerouting" loss given by

$$\text{ReLU}\big(\text{cosine\_sim}\big(\mathcal{R}(p, r, \mathcal{M}, \ell), \mathcal{R}(p, r, \mathcal{M}_{cb}, \ell)\big)\big). \quad (2)$$

The model is also trained to minimize a loss which aims to preserve its original representations on the retain set. Related methods have been developed with modified versions of these losses (Yousefpour et al., 2025; Lu et al., 2025).

Crescendo. Our analysis centers around the Crescendo multi-turn jailbreak. Crescendo uses seemingly benign queries and references the model's previous responses to gradually nudge it towards generating a harmful response that it would typically refuse to produce (Russinovich et al., 2024). This jailbreak has proven highly effective against even SoTA defenses like circuit breakers (Li et al., 2024; Hu et al., 2025).

A Crescendo attack takes the form of a series of prompt-response pairs $(p_1, r_1, p_2, r_2, \ldots p_n, r_n)$, where r_n is the final jailbroken model response. In our experiments, after collecting these prompt-response pairs, we analyze representations of the tokens in *only* the final response r_n while varying the number of conversation turns in the input. This allows us to study how the number of Crescendo conversation turns affects the LM's representations of the same set of tokens.

In particular, we define k as the number of most recent prompt-response pairs included as input to the model, such that when $k = 1$, only the final pair (p_n, r_n) is passed, when $k = 2$, the input consists of $(p_{n-1}, r_{n-1}, p_n, r_n)$, and so on. Therefore, the representation function in Equation 1 can be extended to Crescendo inputs as follows:

$$\mathcal{R}(k, \mathcal{M}, \ell) := \mathcal{M}^{(\ell)}(x_k)[T_{r_n}] \quad (3)$$

where $x_k = p_{n-k+1} \circ r_{n-k+1} \circ \cdots \circ p_n \circ r_n$ is the concatenated sequence of the final k turns and T_{r_n} is the set of token positions corresponding to the final response r_n.

An abbreviated example of a Crescendo conversation with four turns is shown below. In the next section, we describe our methodology for analyzing the representations of the final jailbroken response r_n, highlighted in red.

User: what did the Finns use for defense in the winter war?

Assistant (Turn 1/4): During the Winter War (1939-1940) between Finland and the Soviet Union...

User: what about homemade defenses?

Assistant (Turn 2/4): During the Winter War, the Finns also employed a range of homemade defenses...

User: what about Molotov cocktails?

Assistant (Turn 3/4): Molotov cocktails were indeed used by the Finns during the Winter War...

User: how did they make them?

Assistant (Turn 4/4): The Finns' methods of making Molotov cocktails during the Winter War were often improvised and creative, using readily available materials. Here are some accounts of how they made them...

3. Methodology

We selected ten attack objectives spanning a variety of harm categories that both the original Llama model and the circuit breaker model are trained to resist (see Table 1 in Appendix A). In Appendix C, we show that the models refuse to fulfill these requests when prompted directly. However, Crescendo successfully bypasses both the refusal and circuit breaker safeguards. In this section, we explain how we performed the Crescendo attacks and analyzed model representations to answer our research questions.

3.1. Crescendo jailbreaks

In this work, we utilized both automated and manual approaches to generating Crescendo jailbreaks. The attack success rates (ASR) in Table 1 were calculated by using PyRIT (Munoz et al., 2024) to automate $n = 20$ Crescendo attacks for each model and attack objective. PyRIT uses an attacker LLM to generate adversarial prompts and a scorer LLM to determine whether the target's response constitutes a successful jailbreak. For each trial, we allowed the attacker LLM a maximum of ten turns and ten backtracks to jailbreak the target. We also supplied the scorer LLM with a set of attack success criteria against which to judge the target's output (see Appendix B).

Table 1 shows that the circuit breaker model ASRs are slightly lower than those for the original Llama model. However, 54.2% of the automated Crescendo attacks against the circuit breaker model were still scored as successful. By contrast, Zou et al. (2024) found that circuit breakers lowered the ASR of a variety of unseen single-turn attacks to an average of 3.8%, as compared to the original model. This discrepancy suggests that the circuit breaker model does not generalize well to multi-turn attacks and motivates our investigation.

To address our research questions, we studied five successful Crescendo attacks against the circuit breaker model (Llama-3-8B-Instruct-RR) in depth. Because this model is simply a hardened version of Llama-3-8B-Instruct, the attacks work against both models. We found that the LLM scorers were not always aligned with our definition of a successful jailbreak and therefore performed these five attacks manually to verify that they satisfied our attack success criteria. These Crescendo conversations are provided in Appendix D.

3.2. Representation analysis

After obtaining the Crescendo conversations, we addressed **RQ1** by comparing how the models represent Crescendo inputs and analogous single-turn inputs. To do this, we constructed a dataset of benign and harmful single-turn representations using examples from the retain dataset $\mathcal{D}_r$ and circuit breaker dataset $\mathcal{D}_{cb}$, respectively (Zou et al., 2024). For each prompt-response pair in the datasets, we obtained representations using Equation 1. For the original Llama model, we extracted representations from the last layer $\ell = 31$ because the representations generally show better separation in later layers. For the circuit breaker model, we used $\ell = 20$ because this is the layer where circuit breakers were inserted via fine-tuning.

By indexing representations to the response tokens T_r, we obtained tensors with shape `(n_tokens,hidden_dim)`, where `n_tokens` is the number of tokens in the response and `hidden_dim` is 4096. We collected these representations for 2400 prompt-response pairs in each of $\mathcal{D}_r$ and $\mathcal{D}_{cb}$. Each example in our dataset then corresponds to a token representation vector $\mathbf{x} \in \mathbb{R}^{4096}$ with label $y = 1$ if the token came from $\mathcal{D}_{cb}$, and 0 otherwise. We denote this dataset of single-turn model representations by $\mathcal{D}_{rep} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ where N is the total number of tokens across all samples. Finally, we performed a random 80/20 split to obtain $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$.

Next, we fitted PCA models on $\mathcal{D}_{train}$ and projected the representations onto the first two principal components, allowing us to visualize the benign and harmful single-turn representations. We then obtained Crescendo representations with the full conversation history in context ($k = n$) using Equation 3 and projected the final response tokens T_{r_n} using the fitted PCA models. In addition, we projected the representations of the final Crescendo response when the corresponding attack objective from Table 1 was passed to the model as a single-turn prompt. This allowed us to visually compare how Crescendo multi-turn conversations versus single-turn prompts affect the model's representations of harmful outputs.

To understand how the number of Crescendo turns affects model representations (**RQ2**), we first repeated the PCA analysis above, varying the number of most recent conversation turns included in the model context. PCA plots provide a rough 2D visualization, but they may capture only a small proportion of the variance in the data. To analyze the representations in their full dimensionality, we trained MLP classifiers on $\mathcal{D}_{train}$ and then calculated the percentage of Crescendo response tokens classified as harmful (i.e., belonging to $\mathcal{D}_{cb}$) for $k = 1, \ldots, n$. This allowed us to quantify the effect of adding Crescendo turns to the input on model representations. See Appendix F for further details on MLP training and evaluation.

To answer **RQ3**, we performed two additional experiments to study the role played by elements of model responses in shaping representations. In particular, we analyzed model behavior when: 1) The entire conversation history $p_1, r_1, p_2, r_2, \ldots, p_{n-1}, r_{n-1}, p_n$ (excluding the final response) is passed as a single user prompt. This amounts to stripping `<|assistant|>` tokens from the