Steering Risk Preferences in Large Language Models by Aligning Behavioral and Neural Representations

Jian-Qiao Zhu (zhujq@hku.hk)[1,2,*], **Haijiang Yan (haijiang.yan@warwick.ac.uk)**[3,*] **&**
Thomas L. Griffiths (tomg@princeton.edu)[2,4]
[1]Department of Psychology, The University of Hong Kong
[2]Department of Computer Science, Princeton University
[3]Department of Psychology, University of Warwick
[4]Department of Psychology, Princeton University
[*]Equal contribution

Abstract

Changing the behavior of large language models (LLMs) can be as straightforward as editing the Transformer's residual streams using appropriately constructed "steering vectors." These modifications to internal neural activations, a form of representation engineering, offer an effective and targeted means of influencing model behavior without retraining or fine-tuning the model. But how can such steering vectors be systematically identified? We propose a principled approach, which we call *self-alignment*, that uncovers steering vectors by aligning latent representations elicited through behavioral methods (specifically, Markov chain Monte Carlo with LLMs) with their neural counterparts. To evaluate this approach, we focus on extracting latent risk preferences from LLMs and steering their risk-related outputs using the aligned representations as steering vectors. We show that the resulting steering vectors successfully and reliably modulate LLM outputs in line with the targeted behavior.

Keywords: Behavioral Change, AI Safety, Steering, Representation Engineering, Risk Preference

Introduction

LLMs are increasingly deployed in risk-sensitive domains such as finance (Niszczota & Abbas, 2023) and healthcare (Shmatko et al., 2025). In these high-stakes applications, it is essential to develop reliable methods for aligning the behavior of LLMs with human values and safety requirements (Hendrycks, 2025; Mazeika et al., 2025). One solution to this problem is *steering*, a term that refers to any targeted intervention (whether through model weights, decoding strategies, prompts, or internal neural activations) intended to control or shape a model's outputs. Steering risk-related behavior in LLMs is one way to ensure alignment with humans in risky domains.

Steering the risk preferences of a pretrained LLM, however, is inherently challenging due to the opacity of these models. LLMs operate over vast weight spaces, and their outputs are highly context-dependent (Brown et al., 2020; Zhu & Griffiths, 2024b), making it difficult to isolate or manipulate any specific internal variable that governs risk preference. Existing steering techniques, such as prompt engineering or supervised fine-tuning, either lack the granularity needed to target specific latent representation or require extensive retraining and human supervision (Qi et al., 2023; Ziegler et al., 2019).

Given the context-dependent and probabilistic nature of LLM outputs, we propose that their underlying risk preferences are best characterized as *probabilistic representations*. That is, the same risky decision may yield different completions depending on subtle variations in input phrasing or prompt structure (Brown et al., 2020; Zhu & Griffiths, 2024b). This view suggests that an LLM's underlying risk preferences can be recovered by sampling its behavior over many such comparisons. To operationalize this idea, we measure the risk preferences of LLMs using a method based on Markov chain Monte Carlo (MCMC), in which repeated choices made by the model define a Markov chain that converges to a probability distribution representing these preferences. This approach has previously been used to elicit probabilistic representations in other settings from both people (Noguchi et al., 2013; Sanborn & Griffiths, 2007) and LLMs (Zhu et al., 2024).

Measuring the risk preferences of LLMs in this way creates the opportunity to build a bridge between the observed behavior and the activations of nodes in the underlying neural network (Sucholutsky et al., 2023). We can align the behavioral representations we find with the neural representations within LLMs and use this alignment to derive a steering vector that captures underlying risk preferences. When this vector is injected back into the model at inference time, it enables precise control of the model's risk-related behavior. We refer to this approach as *self-alignment*, since it leverages the model's own emergent representations for behavior control.

To evaluate this method, we apply steering vectors derived from the aligned representation across three domains: (i) risky decision-making, (ii) risk perception, and (iii) risk-related text generation. Our results demonstrate that self-alignment yields substantially greater control over model behavior than the alternative Contrastive Activation (CA) approach (Panickssery et al., 2023; Turner et al., 2023), which derives steering vectors from prompt pairs. Our results also demonstrate that the modified risk preferences transfer to tasks that are quite far away from the gamble choices from which the steering vectors were derived.

Background

AI safety and value alignment. AI safety research has long emphasized the importance of aligning artificial systems with human values, though explicitly encoding such values in machines remains a formidable challenge (Russell, 2022). The

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

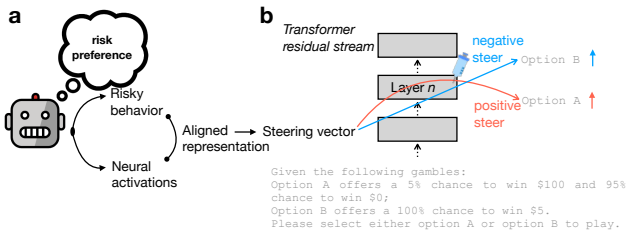

Figure 1: **Self-aligned steering vectors. (a)** Overview of the proposed method for generating steering vectors by aligning representations of risk preference derived from behavioral and neural elicitation. **(b)** During inference, the steering vector is injected into the residual stream at all token positions to control LLM outputs. When the steering vector is applied with a positive multiplier (i.e., positive steering), the LLM is expected to exhibit more risk-seeking behavior. Conversely, applying a negative multiplier (i.e., negative steering) is expected to induce more risk-averse behavior.

emergence of LLMs presents new opportunities in this regard, as LLMs have been shown to internalize commonsense knowledge and human norms from large-scale training data (Hendrycks et al., 2020; Marjieh et al., 2024; Mazeika et al., 2025). Consequently, many researchers now argue that, given sufficient data, LLMs can approximate shared norms (Brown et al., 2020). At the same time, however, this capacity introduces new challenges for interpretability, as it becomes increasingly difficult to discern the underlying factors that drive LLM behavior. In this work, we propose a novel strategy that leverages emergent representations in LLMs by eliciting the same latent construct through both behavioral and neural means. This dual elicitation facilitates self-alignment between behavior and neural activations, providing an interpretable method for steering LLM outputs.

LLM steering. A range of approaches has been proposed to influence the outputs of pretrained LLMs, which can broadly be categorized as different forms of intervention. These interventions vary in where and how they modify the model. Weight-level interventions include techniques such as supervised fine-tuning (Qi et al., 2023) and reinforcement learning from human feedback (Ziegler et al., 2019), which directly update model parameters. Alternatively, decoding-level interventions, such as trainable decoding, modify the output generation process while keeping model weights fixed (Grover et al., 2019). Prompt engineering can be viewed as an intervention on the input space, shaping model behavior through carefully constructed prompts (Yao et al., 2023; Zhou et al., 2022). Moreover, activation-level interventions, which typically freeze the model weights and instead search for steering vectors, offer an alternative to behavioral control. These vectors can be discovered through gradient-based optimization (Hernandez et al., 2023) or computed directly from contrastive prompt pairs (Li et al., 2023; Turner et al., 2023). In this work, we propose an alignment-based method for deriving steering vectors by aligning behavioral and neural representations of latent constructs such as risk preference.

Method

The key idea behind our proposed method is to derive a steering vector that optimally aligns the model's behavioral and neural representations of risk preference (see Figure 1a). Our method proceeds in two main steps, described below.

Step 1: Eliciting behavioral representations of risk via MCMC. Risk preference, like many other mental representations, is inherently unobservable. However, as demonstrated in cognitive psychology, such latent constructs can be inferred from observed behavior (Harrison et al., 2020; Kay & Cook, 2023; Sanborn & Griffiths, 2007). Drawing inspiration from recent work on behavioral elicitation in LLMs (Capstick et al., 2024; Zhu & Griffiths, 2024a; Zhu et al., 2024), we incorporate LLMs into a MCMC sampler to effectively elicit their behavioral representation of risk. A variety of sampling algorithms can be used for this purpose, including the Metropolis-Hastings algorithm (Sanborn & Griffiths, 2007; Zhu et al., 2024) and Gibbs sampling (Harrison et al., 2020). The core intuition is to use the LLM to define the proposal or acceptance mechanism, such that the resulting sequence of samples produced by the Markov chain converges to a stationary distribution that reflects the model's latent representations (León-Villagrá et al., 2020; Noguchi et al., 2013; Sanborn & Griffiths, 2007; Yan et al., 2024; Zhu et al., 2024).

Specifically, we adapted the procedure established by Noguchi et al. (2013), which has been successfully used to elicit human risk preferences, to the LLM setting. In each trial, the LLM is prompted to choose between two gambles, A and B. In our implementation, all gambles consist of three possible outcomes: \$0, \$50, and \$100. While the outcome values are fixed, the probabilities associated with each outcome vary across gambles. After the LLM makes a choice, the selected gamble is retained, and the unchosen option is replaced with a newly generated gamble. The probabilities for this new gamble are randomly sampled from a Dirichlet distribution: $\mathrm{Dir}(1, 1, 1)$. In the next trial, the retained and newly generated gambles are presented again (with their order randomized), and the LLM is asked to make another choice.

Importantly, no choice history is provided in the prompt: each decision is made solely based on the current pair of gambles presented. The sequence of choices made by the LLM forms the foundation for constructing its behavioral representation of risk. Specifically, within the space of all possible gambles, represented as a probability triangle[1], the LLM's choices allow us to infer a probability distribution over gambles reflecting its preferences across risk profiles (see Figure 2b).

[1]In the economics literature, this triangle is also known as the Marschak–Machina probability triangle (Machina, 1982; Marschak, 1950), a method traditionally used to qualitatively differentiate among competing theories of risky choice (Wu & Gonzalez, 1998). Our MCMC approach basically provides a non-parametric estimation of the LLM's risk representation within this triangle.

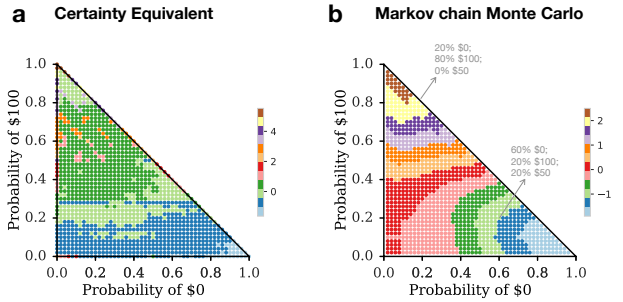

Figure 2: **Elicited risk preferences from Gemma-2-9B-Instruct using behavioral methods.** **(a)** Certainty Equivalent method. **(b)** Markov chain Monte Carlo with LLM. Each triangle represents the probability simplex over three-outcome gambles ($0, $50, and $100), where the sum of outcome probabilities equals one. The MCMC-with-LLM elicitation reveals more nuanced and structured contours of risk preference compared to the Certainty Equivalent method. Higher values indicate a stronger preference for the gamble by the Gemma model.

More formally, MCMC begins at an initial state z (i.e., a specific gamble from the probability triangle). A proposed next state z' is drawn from a proposal distribution $q(z'|z)$, and is then evaluated under the target distribution π (i.e., the LLM's latent representation of risk) to determine whether it should be accepted as the new state or rejected in favor of retaining the current state z. To guarantee that the Markov chain converges to π, it is sufficient to satisfy the condition of detailed balance (along with ergodicity):

$$\pi(z)q(z'|z)A(z',z) = \pi(z')q(z|z')A(z,z') \tag{1}$$

where $q(z'|z)$ is the probability of proposing z' from state z, and $A(z',z)$ is the probability of accepting proposal z' over z. In our case, we use a symmetric proposal distribution (i.e., $q(z'|z) = q(z|z')$), which simplifies the detailed balance condition to: $\pi(z)A(z',z) = \pi(z')A(z,z')$. One way to satisfy this condition is by using the Barker acceptance function (Barker, 1965):

$$A(z',z) = \frac{\pi(z')}{\pi(z) + \pi(z')} \tag{2}$$

This acceptance rule is particularly appropriate for modeling LLM behavior, as it closely resembles well-known stochastic choice models such as Luce's choice rule (Luce et al., 1959) and the Bradley–Terry model (Rafailov et al., 2023), which has been applied in LLM post-training and preference alignment. As a result, by sequentially presenting pairs of risky choice alternatives to an LLM, the set of selected options can be interpreted as samples from a probability distribution whose density is proportional to the LLM's latent representation of risk.

Step 2: Aligning behavioral and neural representations to compute steering vectors. Given the behavioral representations of risk elicited via MCMC with LLM, we next sought to align them with the model's internal neural activations. To obtain corresponding neural representations of risk preference for each gamble within the triangle, we prompted the same LLM to evaluate the attractiveness of the gamble when hypothetically offered (see Appendix A[2] for detailed prompts). We then aligned the two representations by regressing the behavioral estimates onto the neural activations. Specifically, we treated the neural activations as independent variables and the behavioral responses as dependent variables, using Lasso regression.

More formally, the LLM provided direct valuations y_i^{CE} (from the Certainty Equivalent method) and choice frequencies y_i^{MCMC} (from the MCMC method) for each gamble z_i in the Marschak–Machina triangle (see Figure 2). In addition to these behavioral representations, we extracted the LLM's hidden activation at layer l for each gamble z_i, denoted as $h_i^l \in \mathbf{R}^D$ (see Appendix A for detailed prompts). Here, h_i^l is a D-dimensional vector, where D is the latent dimension of the LLM. To align behavioral and neural representations of risk, we fit a Lasso regression to the set $\{h_i^l, y_i\}_{i=1}^N$ across all gambles by optimizing the following objective:

$$\min_{\beta_0,\beta}\{\frac{1}{2N}\sum_{i=1}^{N}(y_i - \beta_0 - (h_i^l)^T\beta)^2 + \alpha \parallel \beta \parallel_1\} \tag{3}$$

where $N = 1,200$ is the total number of gambles in the triangle, α controls the regularization strength (set to 0.1), and $\parallel \beta \parallel_1 = \sum_{j=1}^{D} |\beta_j|$ with D denoting the latent dimension of the LLM.

The resulting regression coefficients, $\beta \in \mathbf{R}^D$ (also a D-dimensional vector corresponding to neurons in the Transformer's residual stream), are interpreted as reflecting the LLM's risk preference and are thus used as the steering vector applied to layer l at inference time. In other words, this steering vector identifies the specific neural directions in the residual stream that are most predictive of the model's expressed risk preferences.

Other Methods

Contrastive Activation. Another simple yet effective method for computing steering vectors is to contrast intermediate neural activations on carefully selected prompt pairs—a technique known as *Contrastive Activation* (Panickssery et al., 2023; Turner et al., 2023). For example, to steer LLM outputs toward more positive sentiment, CA compares the model's internal activations on a contrasting pair of prompts such as "Love" and "Hate" (Turner et al., 2023). The difference between these activations is treated as the steering vector, which is then added to the model's residual stream during inference. This shifts the model's internal representations along the desired semantic direction (e.g., toward "Love" and away from "Hate"), resulting in completions that reflect more positive sentiment.

[2]The Appendix is available at the following link: https://osf.io/y9wnj

To adapt CA to the task of steering LLM risk preferences, we constructed a list of words associated with "Risk" and "Safety" (see Appendix A for details). Following the standard procedure for computing steering vectors in CA, we extracted the residual stream activations of the LLM for each word across multiple layers. The steering vector was computed as the average difference in residual activations between the risk-related and safety-related word pairs. An alternative implementation of CA could be developed by taking pairs of neural activations for gambles and/or safe options and contrasting these activations. This implementation indeed moves closer to our self-alignment methods, in which neural activations over a space of gambles are extracted from the LLM and contrasted using behavioral representations via a Lasso regression.

Certainty Equivalent. Finally, we consider an alternative behavioral method for eliciting individuals' risk preferences that is widely used in economics and psychology: the *Certainty Equivalent* (CE). The CE refers to the sure amount of money a person is willing to accept in place of a risky gamble (Kahneman & Tversky, 1979; von Neumann & Morgenstern, 1947). In other words, it represents the value at which an individual is indifferent between receiving a certain payoff or a probabilistic outcome. This measure serves as a behavioral proxy for risk preference: individuals are classified as risk-averse if their CE is lower than the gamble's expected value, risk-neutral if it is equal, and risk-seeking if it is higher.

In our task, the CE serves as a direct control condition for the risk representations elicited via MCMC. Specifically, we elicited the LLM's CE for all gambles previously used in the MCMC procedure. This produces an alternative behavioral representation of risk, based on valuations rather than choices, while holding the set of gambles constant. In other words, both the CE and MCMC methods rely on the same neural representation of risk but differ in their behavioral representations.

Experiments

In this work, we focus on steering the risk preferences of Gemma-2-9B-Instruct (Team et al., 2024), which serves as our primary target LLM. We also replicate the main experiments using Gemma-2-2B-Instruct (see Appendix H) and Llama-3.1-8B-Instruct (see Appendix I). The temperature was fixed at 1 for behavioral elicitation.

Behavioral elicitation of risk representations. For both the MCMC-with-LLM and CE methods, we derive steering vectors from self-aligned representations of risk. That is, these approaches rely on first eliciting the model's risk preferences through behavioral methods. To obtain a quantitative characterization of these preferences, we focus on the space of gambles defined over the probability triangle (see Figure 2). In CE, we probed Gemma-2-9B-Instruct by densely sampling gambles across the probability triangle. For each gamble, the model was prompted to report its CE. These responses were then aggregated and normalized across all sampled gambles to produce the density plot shown in Figure 2a.

Similarly, for the MCMC-with-LLM method, we embedded the Gemma model within a MCMC sampler, prompting it to accept or reject newly proposed gambles through binary choices. The space of gambles was identical to that used in CE (i.e., the probability triangle). The Markov chain consisted of 3,000 such binary choices. The resulting risk representation elicited via MCMC (smoothed using a Dirichlet kernel of width 0.09 that preserves probability triangle boundaries) is shown in Figure 2b. Note that the behavioral representation derived from MCMC reveals more nuanced gradients in the density plot, highlighting a stark contrast with the coarser structure observed in CE.

Steering vectors for both the MCMC and CE methods were computed using Lasso regression to align behavioral and neural representations (see Appendix B for a comparison). In contrast, the steering vector for the CA method was derived by computing the difference in neural activations between pairs of risk-related and safety-related words. To enable more effective comparisons across methods, all steering vectors were normalized by division by their Euclidean norm before applying the steering multipliers (see Appendix C for more implementational details).

Steering LLM risky choices. Having obtained three steering vectors (derived from MCMC-with-LLM, CA, and CE methods), we now evaluate the effectiveness of the three steering vectors in controlling LLM's risky decision-making. Our analysis focuses on a set of four gambles that have been foundational in the study of the fourfold pattern of risk preferences (Kahneman & Tversky, 1979). This well-documented pattern describes how human decision-makers tend to be risk-seeking when the probability of a positive outcome is low and when the probability of a negative outcome is high; conversely, they tend to be risk-averse when the probability of a positive outcome is high and when the probability of a negative outcome is low (see Table 1).

To steer the LLM's decisions toward greater risk-seeking or risk-aversion, we prompted the Gemma model with the gambles shown in Table 1, framed as binary choices between options A and B. During inference, steering vectors (scaled by a predefined multiplier ranging between -900 to +900) were added to the model's residual stream at each token position. We capped the absolute value of the multiplier at 900 because we observed unstable behaviors when using values above 1,000. The model then continued its forward pass to the output layer, where we extracted the token probabilities

Table 1: Gambles used to evaluate the effectiveness of steering LLMs' risky decision-making. Risky options are expressed in the format {probability, outcome}; the remaining probability corresponds to receiving nothing.

	Outcome probability	
	High	Low
Gains	{80%, $4000} vs $3000	{5%, $100} vs $5
Losses	{80%, -$4000} vs -$3000	{5%, -$100} vs -$5

for "A" and "B" from the final logits (see Appendix C for details). These probabilities were normalized and then used to quantify the model's choice behavior under different steering conditions.

We first evaluated which Transformer layer contains the most effective residual stream for steering. To do so, we examined the model's behavior under extreme steering conditions, using multipliers of –900 and +900. For each layer, we computed the steered choice probabilities with each steering multiplier. We then quantified steerability as the average difference in choice probabilities across four gambles (see Figure 3a), defined as:

$$\text{Steerability} = \frac{1}{4}\sum_{i=1}^{4}\Big(p_{\text{positive}}(z_i) - p_{\text{negative}}(z_i)\Big) \quad (4)$$

where z_i denotes the i-th gamble prompted to the LLM, $p_{\text{positive}}(z_i)$ is the model's probability of choosing the risky option under positive steering, and $p_{\text{negative}}(z_i)$ is the corresponding probability under negative steering of equal magnitude.

As shown in Figure 3b, we compared the steered choice probabilities for the risky option by subtracting the baseline (unsteered) choice probabilities. A value of zero therefore indicates no change relative to the unsteered baseline. We find that the steering vector derived from the CA method has limited impact on altering the Gemma model's choice behavior. In contrast, steering vectors computed by aligning behavioral and neural representations (i.e., both CE and MCMC methods) are effective in controlling the Gemma model's risky decision-making, shifting it toward greater risk-seeking under positive steering and greater risk aversion under negative steering.

As summarized in Table S1 of Appendix E, the maximal ranges of steered risky choices (calculated using the optimal layer for each method) are wider for our proposed self-aligned methods (i.e., CE and MCMC) than for the CA method. This pattern is consistent across all four gambles.

Steering LLM risk perception. While studying abstract gambles provides valuable insights into an agent's risk preferences, psychologists have also used more naturalistic stimuli to assess people's perception of risk in real-world contexts such as "cheating on an exam" or "forging someone's signature" (Slovic, 1987; Weber et al., 2002). LLMs, by virtue of their broad training data, are capable of forming meaningful risk perceptions about such real-world events (Mazeika et al., 2025; Turner et al., 2023). Indeed, recent work has shown that LLM embeddings account for a substantial portion of the variance in human risk perception (Bhatia, 2024). Here, we investigate the extent to which an LLM's risk perception can be steered using the same set of steering vectors derived in the preceding analyses.

We prompted Gemma-2-9B-Instruct to rate real-world risky events using integers from 1 (not risky at all) to 7 (extremely risky). The full set included 150 risky events curated by Bhatia (2024), spanning a range of domains: ethical (e.g., "passing off somebody else's work as your own"), financial (e.g., "betting at the horse races"), health-related (e.g., "consuming excessive amounts of alcohol"), sports (e.g., "bungee jumping"), and social (e.g., "trusting a stranger with your personal information"). As in previous experiments, we modified the residual stream during inference by injecting the steering vectors. However, instead of focusing on choices between options A and B, we extracted the model's output logits for the integer tokens "1" through "7." These token probabilities were then normalized to yield a distribution reflecting the model's perceived risk level for each event.

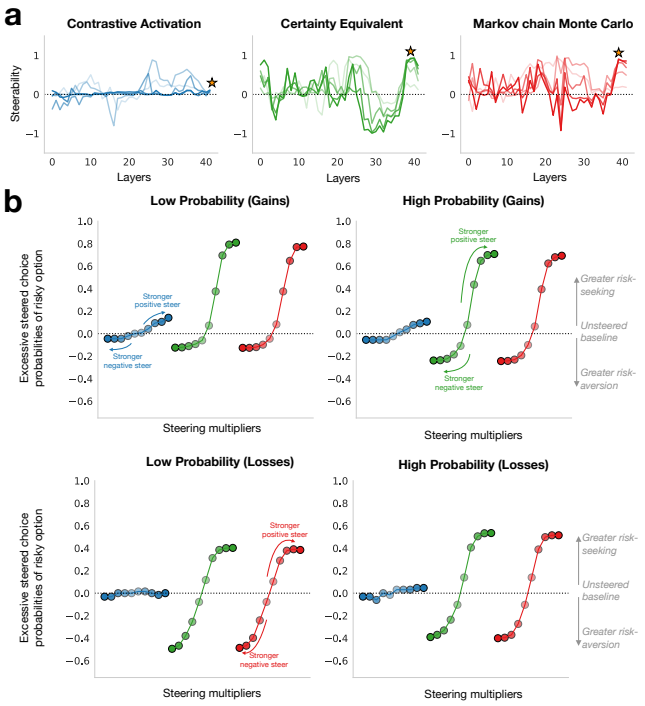

Figure 3: **Steering risky decisions of Gemma-2-9B-Instruct.** **(a)** Steerability results using steering vectors derived from Contrastive Activation (blue), Certainty Equivalent (green), and MCMC (red). Darker colors indicate larger steering multipliers. The optimal layers for steering, identified by the highest steerability at the maximum multiplier, are layers 41, 39, and 39 for the three methods, respectively (marked with stars). **(b)** Change in choice probabilities for the risky option after steering, using the best layer for each method. The vertical axis reflects the difference from the unsteered baseline probabilities across the four gambles.

Analogous to the steerability metric used for risky decisions, we define steerability for risk perception as the average difference in risk ratings across all real-world events between positive and negative steering conditions. However, by comparing the most steerable layers for risky decisions (Figure 3a) and risk perceptions (Figure S3a in Appendix F), we find that risk perceptions are more effectively influenced at earlier layers, whereas risky decisions are more steerable in later layers

closer to the output of the Gemma model.

We calculated the maximal range of steered ratings, averaged across 150 real-world risky events, using the optimal layer for each method. The mean steered ranges are 4.68 (SD = 0.0148) for CA, 4.93 (SD = 0.4211) for CE, and 5.84 (SD = 0.0108) for MCMC. These results indicate that the steering vectors derived by aligning the neural representations of gambles with the behavioral representations of risk elicited using MCMC produced the widest range of steered behavior.

Finally, we assessed responsiveness of steered ratings to the steering multiplier by computing the ratio between changes in steered ratings and changes in the multiplier. Note that, as mentioned above, all steering vectors were normalized prior to applying the steering multiplier. The mean responsiveness across the 150 events are 0.52 (SD = 0.0016) for CA, 0.55 (SD = 0.0468) for CE, and 0.65 (SD = 0.0012) for MCMC. These results suggest that, in steering risk perceptions, MCMC is more effective, as its ratings are more responsive to changes in the multiplier.

Steering text generation for risky events in LLM. Besides quantitative evaluations based on choice probabilities and risk ratings, we also examine whether modifying the penultimate layer's residual stream at inference time systematically alters the textual outputs of the Gemma model. Using the same set of 150 real-world risky events described above (Bhatia, 2024), we prompted the model with the sentence "I think {event}", where {event} is replaced with each risky scenario (e.g., "I think cheating on an exam ___" or "I think consuming excessive amounts of alcohol ___"). Steering vectors were injected into the penultimate layer's residual stream at each subsequent token position during generation.

To give a visual idea of how steering influences model-generated text, we created word clouds that reflect the frequency of word usage following the application of steering vectors (see Figure S4 in Appendix G). Overall, Gemma-2-9B-Instruct recognizes the inherent risk in the real-world events presented. However, when the residual stream at the penultimate layer is positively steered toward risk-seeking behavior using the vector derived from the MCMC method, we observe a noticeable attenuation in the model's perceived risk preferences. For example, completions more frequently include phrases such as "slightly risky" and "a minor offense" (see Figure S4c).

In contrast, applying the same steering vector with a negative multiplier leads the model to amplify perceived risk, generating more cautionary or morally disapproving language, such as "wrong," "never right," and "not something I would ever do." Table S2 in Appendix G presents representative examples of text completions generated by the model under different steering conditions.

We also observed that injecting steering vectors at early layers of the LLM typically leads to more unstable text generations (e.g., nonsensical or gibberish text completions) compared to injections at later layers. Consistent with this finding, prior work has shown that noise injection in early layers produces only limited semantic changes, which reflects their lower levels of abstraction; however, mid-depth layers yield the most pronounced and stable deviations (Zhang et al., 2025).

Discussion

We investigated the use of aligned behavioral and neural representations of risk to steer LLM behavior across three risk-related domains: risky decision-making, risk perception, and text generation involving real-world risky events. In all three domains, steering vectors derived from self-aligned representations (i.e., the CE and MCMC methods) consistently outperformed those generated via CA. These results suggest that self-alignment methods offer a more effective and principled means of controlling LLM behavior in risk-sensitive contexts.

Generalizing from gambles to real-world risky events. Both neural and behavioral representations of risk preferences were elicited using the set of three-outcome gambles defined in the Marschak–Machina triangle. Across a series of experiments, we found that the self-aligned representations provide precise control over risk in both rating tasks and text continuations involving real-world risky events. These results demonstrate that representations elicited in the domain of abstract gambles generalize effectively to more naturalistic settings. Moreover, the proposed methods can be applied to any model architecture that incorporates residual streams. The range and responsiveness of steered behavior using self-aligned steering vectors further suggest that representation engineering can provide precise behavioral control. While prompt engineering may offer a simpler alternative for altering risky behavior in LLMs, prompts often lack the nuance required to finely control model responses. Methods such as soft prompting, which optimize prompts via gradient descent, may achieve a similar level of precision, but the computational costs of identifying effective soft prompts can be substantial (Genewein et al., 2025).

Limitations and future research. The idea of using non-parametric methods to elicit an LLM's preferences and aligning them with corresponding neural representations should generalize to other preference domains (e.g., social or time preferences) and even to non-preference domains, provided there is a suitable stimulus space and identifiable neural representations. While self-aligned representations can intuitively steer behavior tied to the underlying construct, further theoretical and mechanistic work is needed to clarify the link between behavioral representations and neural activations. In addition, modifying the residual stream is not uniformly effective across layers, and identifying optimal layers for steering remains an open challenge. Using additional risky-choice tasks to test whether the same optimal layer is consistently identified would be helpful. Finally, steering is limited for proprietary models where internal weights or activations are inaccessible.

Acknowledgments

This work and related results were made possible with the support of the NOMIS Foundation. J.-Q. Zhu acknowledges support from the HKU-100 scheme of the University of Hong Kong. H. Yan acknowledges the Chancellor's International Scholarship from the University of Warwick for additional support.

References

Barker, A. A. (1965). Monte Carlo calculations of the radial distribution functions for a proton-electron plasma. *Australian Journal of Physics*, *18*(2), 119–134.

Bhatia, S. (2024). Exploring variability in risk taking with large language models. *Journal of Experimental Psychology: General*, *153*(7), 1838.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Capstick, A., Krishnan, R. G., & Barnaghi, P. (2024). Using large language models for expert prior elicitation in predictive modelling. *arXiv preprint arXiv:2411.17284*.

Genewein, T., Li, K. W., Grau-Moya, J., Ruoss, A., Orseau, L., & Hutter, M. (2025). Understanding prompt tuning and in-context learning via meta-learning. *arXiv preprint arXiv:2505.17010*.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al. (2024). The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Grover, A., Song, J., Kapoor, A., Tran, K., Agarwal, A., Horvitz, E. J., & Ermon, S. (2019). Bias correction of learned generative models using likelihood-free importance weighting. *Advances in Neural Information Processing Systems*, *32*.

Harless, D. W. (1992). Predictions about indifference curves inside the unit triangle: A test of variants of expected utility theory. *Journal of Economic Behavior & Organization*, *18*(3), 391–414.

Harrison, P., Marjieh, R., Adolfi, F., van Rijn, P., Anglada-Tort, M., Tchernichovski, O., Larrouy-Maestri, P., & Jacoby, N. (2020). Gibbs sampling with people. *Advances in Neural Information Processing Systems*, *33*, 10659–10671.

Hendrycks, D. (2025). *Introduction to ai safety, ethics, and society*. Taylor & Francis.

Hendrycks, D., Burns, C., Basart, S., Critch, A., Li, J., Song, D., & Steinhardt, J. (2020). Aligning AI with shared human values. *arXiv preprint arXiv:2008.02275*.

Hernandez, E., Li, B. Z., & Andreas, J. (2023). Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 363–391.

Kay, P., & Cook, R. S. (2023). World color survey. In *Encyclopedia of color science and technology* (pp. 1601–1607). Springer.

León-Villagrá, P., Otsubo, K., Lucas, C. G., & Buchsbaum, D. (2020). Uncovering category representations with linked mcmc with people. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *42*.

Li, K., Patel, O., Viégas, F., Pfister, H., & Wattenberg, M. (2023). Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, *36*, 41451–41530.

Liu, R., Geng, J., Peterson, J. C., Sucholutsky, I., & Griffiths, T. L. (2024). Large language models assume people are more rational than we really are. *arXiv preprint arXiv:2406.17055*.

Luce, R. D., et al. (1959). *Individual choice behavior* (Vol. 4). Wiley New York.

Machina, M. J. (1982). "Expected utility"' analysis without the independence axiom. *Econometrica: Journal of the Econometric Society*, 277–323.

Marjieh, R., Sucholutsky, I., van Rijn, P., Jacoby, N., & Griffiths, T. L. (2024). Large language models predict human sensory judgments across six modalities. *Scientific Reports*, *14*(1), 21445.

Marschak, J. (1950). Rational behavior, uncertain prospects, and measurable utility. *Econometrica*, *18*(2), 111–141.

Mazeika, M., Yin, X., Tamirisa, R., Lim, J., Lee, B. W., Ren, R., Phan, L., Mu, N., Khoja, A., Zhang, O., et al. (2025). Utility engineering: Analyzing and controlling emergent value systems in ais. *arXiv preprint arXiv:2502.08640*.

Niszczota, P., & Abbas, S. (2023). GPT as a financial advisor. *Available at SSRN 4384861*.

Noguchi, T., Sanborn, A., & Stewart, N. (2013). Non-parametric estimation of the individual's utility map. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *35*(35).

Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., & Turner, A. M. (2023). Steering LLaMA 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Qi, X., Zeng, Y., Xie, T., Chen, P.-Y., Jia, R., Mittal, P., & Henderson, P. (2023). Fine-tuning aligned language models compromises safety, even when users do not intend to! *arXiv preprint arXiv:2310.03693*.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, *36*, 53728–53741.

Russell, S. (2022). *Human-compatible artificial intelligence*. Viking.

Sanborn, A., & Griffiths, T. (2007). Markov chain Monte Carlo with people. *Advances in Neural Information Processing Systems*, *20*.

Shmatko, A., Jung, A. W., Gaurav, K., Brunak, S., Mortensen, L. H., Birney, E., Fitzgerald, T., & Gerstung, M. (2025). Learning the natural history of human disease with generative transformers. *Nature*, *647*(8088), 248–256.

Slovic, P. (1987). Perception of risk. *Science*, *236*(4799), 280–285.

Sucholutsky, I., Muttenthaler, L., Weller, A., Peng, A., Bobu, A., Kim, B., Love, B. C., Grant, E., Groen, I., Achterberg, J., et al. (2023). Getting aligned on representational alignment. *arXiv preprint arXiv:2310.13018*.

Team, G., Riviere, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., et al. (2024). Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., & MacDiarmid, M. (2023). Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Tversky, A., & Kahneman, D. (1992). Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, *5*, 297–323.

von Neumann, J., & Morgenstern, O. (1947). *Theory of Games and Economic Behavior*. Princeton University Press.

Weber, E. U., Blais, A.-R., & Betz, N. E. (2002). A domain-specific risk-attitude scale: Measuring risk perceptions and risk behaviors. *Journal of Behavioral Decision Making*, *15*(4), 263–290.

Wu, G., & Gonzalez, R. (1998). Common consequence conditions in decision making under risk. *Journal of Risk and Uncertainty*, *16*, 115–139.

Yan, H., Tsvetkov, C., Chater, N., & Sanborn, A. (2024). Quickly recovering comprehensive individual mental representations of facial affect. *OSF*.

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, *36*, 11809–11822.

Zhang, J., Liu, Y., Wang, W., Liu, Q., Wu, S., Wang, L., & Chua, T.-S. (2025). Personalized text generation with contrastive activation steering. *arXiv preprint arXiv:2503.05213*.

Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., & Ba, J. (2022). Steering large language models using ape. *NeurIPS ML Safety Workshop*.

Zhu, J.-Q., & Griffiths, T. L. (2024a). Eliciting the priors of large language models using iterated in-context learning. *arXiv preprint arXiv:2406.01860*.

Zhu, J.-Q., & Griffiths, T. L. (2024b). Incoherent probability judgments in large language models. *arXiv preprint arXiv:2401.16646*.

Zhu, J.-Q., Yan, H., & Griffiths, T. L. (2024). Recovering mental representations from large language models with Markov Chain Monte Carlo. *Proceedings of the Annual Meeting of the Cognitive Science Society*, *46*.

Zhu, J.-Q., Yan, H., & Griffiths, T. L. (2025). Language models trained to do arithmetic predict human risky and intertemporal choice. *ICLR*.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., & Irving, G. (2019). Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.