Linearly Controlled Language Generation with Performative Guarantees

Anonymous authors
Paper under double-blind review

Abstract

The increasing prevalence of Large Language Models (LMs) in critical applications highlights the need for controlled language generation methods that are not only computationally efficient but that also enjoy performance guarantees. To achieve this, we use a common model of concept semantics as linearly represented in an LM's latent space. In particular, we take the view that natural language generation traces a trajectory in this continuous semantic space, realized by the language model's hidden activations. This view permits a control-theoretic treatment of text generation in latent space, in which we propose a lightweight, gradient-free intervention that dynamically steers trajectories away from regions corresponding to undesired meanings. In particular, we propose to directly intervene the activations of the token that is being generated in embedding space in an online fashion. Crucially, we do not simply steer activations towards a desirable region. Instead, our method relies on classical techniques from control theory to precisely *control* activations in a context-dependent way, and guarantees that they are brought into a specific pre-defined region of embedding space that corresponds to allowed semantics. Our intervention is computed in closed-form according to an optimal controller formulation, minimally impacting generation time. This control of the activations in embedding space allows for fine-grained steering of attributes of the generated sequence. We demonstrate the effectiveness of our approach on different objectives—toxicity avoidance and sentiment control—while maintaining text quality.

1 Introduction

Language Models (LMs) have become widespread in critical applications such as content moderation and real-time information dissemination (Zeng et al., 2024). Despite their transformative impact, these models require updates to remain accurate post-deployment. Moreover, as demand for more nuanced text generation rises, strategies that enforce constraints during text generation are increasingly needed. To address these challenges, controllable text generation has emerged as a pivotal research area. In many applications of LMs, it is desirable to set certain attributes of the model's output text, like tone or toxicity, to a certain range. In practice, this range is often quantified via numerical scores; for example, text toxicity rated on a Likert scale, or the likelihood of having a positive sentiment.

Several approaches have been proposed towards controllable text generation (Kumar et al., 2021; Lu et al., 2021; Li et al., 2022; Qin et al., 2022). Of them, a popular approach is prompt engineering (Luo et al., 2023; Bhargava et al., 2023; Cai et al., 2023), where natural language prompts are carefully chosen at input time to steer generation. Other approaches modify LM weights to achieve the desired outputs (Yao et al., 2023; Li et al., 2023b). Lastly, some methods engineer LM *activations*, or input representations, to steer them into the representations of desired outputs (Dathathri et al., 2019; Hernandez et al., 2023; Konen et al., 2024; Li et al., 2024a; Rodriguez et al., 2024; Wu et al., 2024).

Despite current efforts, ensuring the controllability of these models remains a challenge due to their limited interpretability. In particular, existing methods offer steering capabilities, rather than true *control*: their interventions on the representations nudge a target attribute in a direction. Such direction is deemed

through learning to best capture the nature of desired outputs, but current methods lack guarantees on the effectiveness of the steering, compliance, or accuracy with which the control goal will be achieved. For example, some approaches like knowledge editing (Hernandez et al., 2023) provide an efficient alternative to exhaustive retraining, it also poses risks akin to the butterfly effect: minor adjustments can lead to unintended consequences on the knowledge graph. Similarly, works like ReFT (Wu et al., 2024) suffer from the same problem since they operate solely on the prompt representations and do not intervene in the model's activations as generation unfolds. Thus, achieving robust and verifiable controllability remains a critical goal for the safe and reliable deployment of language models. We clarify this distinction by defining two terms that are often conflated in the literature: *steering* and *control*.

Steering vs. Control

Steering refers to interventions that bias the model's internal representations or outputs toward a desired outcome—such as reduced toxicity or positive sentiment—without enforcing guarantees on success. Most steering methods operate by learning a direction in latent space that correlates with desired outputs and nudging the model towards it. In contrast, **control** implies an explicit mechanism that enforces constraints on model representations with *formal guarantees*. A controlled generation system *ensures* that representations or outputs lie within a well-defined, often numerically specified, set of acceptable values.

In this work, we focus on control at the *activation* level, and provide theoretical tools that show the attainment of target outcomes. To this end, we propose to use control theory to tackle controlled language generation. Specifically, optimal control theory (Kirk, 2004) offers principled methods to steer trajectories in latent space that enjoy theoretical guarantees on the performance of the intervention. In the framework of optimal control theory, our intervention method, which we call Linear Semantic Control (LiSeCo), derives from a theoretical formulation of controlled text generation. Our contributions are both theoretical and empirical: (1) we formally pose LM control during generation in activation space as a constrained optimization problem and provide its closed-form solution with guarantees for where the resulting activations lie in embedding space; (2) we study how control in the activation space can, in theory, translate to controllable token generation during decoding, and (3) we empirically demonstrate our method on text attribute steering for toxicity and sentiment. We confirm, with experiment corroborating theory, that LiSeCo dynamically controls the activation's trajectory during generation to avoid disallowed concepts while maintaining text quality and minimal impact on inference time latency. Experimentally, we show that principled *control in activation space* that lends itself to reliable *steering of the output attributes.*

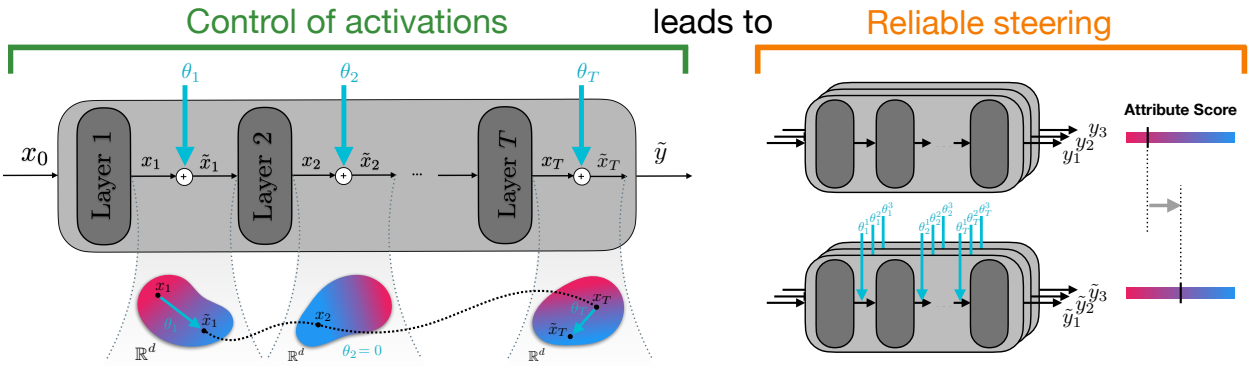

Figure 1: The LiSeCo intervention is computed as the solution to an **optimal control problem**, whose value is dependent on the current activation ($x_t \in \mathbb{R}^d$). When activations naturally fall inside some pre-defined bounds, the value of the intervention is zero. However, when activations fall outside of some pre-defined bounds, the intervention *controls* the activation to *guarantee* that the updated state $x_t + \theta_t \in \mathbb{R}^d$ lies in the desired location of the space. This precise control in the activation space yields to fine-grained steering of the output sequence in token space according to the attribute of interest.

Contributions of LiSeCo

The accepted use of steering vectors for text generation in the literature empirically grounds the promise of this approach. However, none of the approaches found in the literature provide *formal* guarantees on the effectiveness of the steering, compliance, or accuracy with which the control goal is achieved, nor at the activation level neither in the resulting token generations. Here, we provide an intervention that controls the activations during generation and is *theoretically guaranteed* to steer them into the allowed region. Our work differs from existing literature by the following novel contributions:

1. **Optimal Control Formalization with Theoretical Guarantees.** We frame online language model intervention as an optimal control problem, which we leverage to provide an efficient closed-form solution that ensures outputs fall within a target attribute range. Unlike steering methods, which nudge activations toward desired outcomes without guarantees, our method enforces attribute bounds as hard constraints—delivering true control over activations. This is enabled by the use of an optimal control framework to cast the problem in a domain that has been mostly empirical. Providing guarantees on the output text attribute requires further assumptions; we provide sufficient conditions for activation control to translate to output text control in Appendix C. Furthermore, the optimal formulation ensures that the intervention is minimal, i.e., it has the smallest magnitude required to transport the activation into the desired region of representation space. This avoid over- and under-steering and preserves the naturalness of the resulting outputs, unlike other interventions in the literature (see Section 6).
2. **Score-Space Control and Interpretability.** Our approach enables precise targeting of continuous-valued attributes (e.g., toxicity, sentiment) using interpretable numerical scores. Rather than implicitly influencing the model with an intervention that nudges representations on an ad-hoc direction, we directly specify a target value or range and guarantee compliance in embedding space. This affords control over generation with a greater level of granularity as compared to existing approaches. For instance, our method allows for bidirectional steering, meaning that it can be used to both lower or increase an attribute in a given range. Moreover, the level of guarantee compliance in *token space*, given compliance in activation space, serves as an interpretability tool that tests the functional representation of concepts and their causal relevance in generation.
3. **Closed-form, Low-latency Online Intervention.** The intervention is computed analytically in closed form, avoiding the need for backpropagation or iterative optimization at inference time. This yields minimal computational overhead compared to popular steering methods such as FUDGE (Yang & Klein, 2021) or PPLM (Dathathri et al., 2019), while achieving stronger and more reliable control outcomes. Using control-theoretic and optimization tools, we derive an intervention that is lightweight, introduces minimal computation overhead, and is adaptive, i.e., only intervenes when activations are outside of the allowed region. This allows for our intervention to be applied to every layer to achieve dynamic control with guarantees on the activations, and fine-grained steering at the token-level.

We extend the vision of Dalrymple et al. (2024) by demonstrating a concrete instantiation of guaranteed safe AI principles in a real-world language modeling task. Though Soatto et al. (2023) apply theoretical tools from control to LM text generation, to the best of our knowledge, our method is the first to propose a control-theoretic intervention whose theoretical guarantees are validated in practice. We analyze other state-of-the-art activation-based methods, such as ReFT or AcT, under a control theoretic lens in Appendix F.

2 Related Work

Contemporary language models are deep neural networks pre-trained on trillions of tokens of Internet-scale text. In part due to their vast scale and limited interpretability, methods to control them in a fine-grained way remain elusive. A number of approaches have already been proposed towards this end, spanning the whole spectrum of permanent (Meng et al., 2022b; Belrose et al., 2023) to online intervention strategies (Liu et al., 2021; Dathathri et al., 2019; Rodriguez et al., 2024). Here, we review post-hoc intervention methods and situate LiSeCo with respect to the current landscape.

Post-hoc intervention methods can intervene on various components of the LM: for instance, weights via finetuning (Hu et al., 2023); decoding, like FUDGE and GeDI (Yang & Klein, 2021; Krause et al., 2021); or activations, like ActAdd and AcT (Turner et al., 2023; Rodriguez et al., 2024). LiSeCo falls in the later category. All such methods aim to modify some attribute, such as toxicity, while maintaining text fluency. Ultimately, all methods work towards this goal by modifying the LM's final probability distribution, either directly or indirectly. We can situate where different method classes intervene, viewing an LM as a series of T function compositions corresponding to the T layers, where s is a sequence of tokens:

$$\mathbb{P}_{LM}(s_i|s_{<i}) = f_T \circ f_{T-1} \cdots \circ f_1(s_{<i}) := LM(s_{<i}).$$

Decoding-based methods fix the function $LM := f_T \circ f_{T-1} \circ \cdots f_1$ and directly edit its output probability distribution $\mathbb{P}_{LM}(s_i|s_{<i})$ (Yang & Klein, 2021; Liu et al., 2021; Krause et al., 2021). These methods require access to an external evaluator whose feedback is used to calibrate token probabilities, which can result in high inference latency.

Prompt engineering is a technique that controls the LM's output by varying the input context $s_{<i}$, keeping the function $LM := f_T \circ f_{T-1} \circ \cdots f_1$ fixed (Luo et al., 2023; Bhargava et al., 2023; Cai et al., 2023; Wei et al., 2022; Li & Liang, 2021). Prompts are often highly task-specific, requiring either manually crafting or ad-hoc computationally-taxing techniques, and success can be brittle to prompt choice (Weber et al., 2023). While the space of natural language prompts is discrete, LM weights and activations live in continuous high-dimensional space, which is more expressive; then, rather than search over discrete prompts, other approaches that exploit this expressivity directly intervene in the internals of the model.

Of them, **weight-based methods** modify the functions f_i themselves, which permanently constrains the space of final probability distributions $\mathbb{P}_{LM}$. These methods comprise, e.g., reinforcement learning from human feedback (Ouyang et al., 2022), instruction-tuning, parameter-efficient adaptation (Hu et al., 2022), or targeted weight-editing (Meng et al., 2022b; Belrose et al., 2023). In such approaches, weights are modified according to the goal of the controlled generation by, for instance, learning the necessary update (De Cao et al., 2021; Mitchell et al., 2021), or localizing and editing target parameters encoding specific knowledge (Dai et al., 2022; Meng et al., 2022a;c; Li et al., 2024b). Pitfalls range from potential inconsistencies and distortions, to the fact that weight-based methods can only correct errors in the LM's parametric knowledge, but not in-context (Li et al., 2023b).

Activation-based methods, such as LiSeCo, fix $LM := f_T \circ f_{T-1} \circ \cdots f_1$, but intervene at the domain of each f_i, where introducing a steering vector transforms the input to f_i (Li et al., 2023a; Turner et al., 2023). These interventions can be seen as restricting the domain of each f_i, eventually constraining the space of probability distributions $\mathbb{P}_{LM}$ when composed up through the layers. A key advantage of activation steering is rapid adaptation that can be made *context-dependent*. Often, this is achieved with *linear interventions*, i.e. interventions θ_t such that layer $t+1$ receives as input $x_t + \theta_t$, where x_t is the output of layer t. An initial work in this domain was Plug and Play (Dathathri et al., 2019), where a linear intervention is computed at every layer. The control goal is encoded as the objective function in an optimization that is then solved via back-propagation, adding significant computational overhead at inference time. Subsequent approaches also compute linear modifications to the latent state, but reduce computational overhead, act on only a few layers (Subramani et al., 2022; Konen et al., 2024), pre-compute steering vectors to avoid back-propagation (Turner et al., 2023), or address the issue of computational efficiency at the expense of optimality (the intervention is not formulated as an optimizer) (Li et al., 2024a). Recent approaches like REMEDI (Hernandez et al., 2023) or ReFT Wu et al. (2024) find optimal interventions to achieve different target outputs, but these are only used to edit representations in the prompt since they require to first compute all original representations in order to then compute the appropriate intervention. Lastly, AcT (Rodriguez et al., 2024) learns an optimal transport map between two distributions of outputs (toxic and nontoxic), and applies this lightweight map online to the representation being generated. Steering is done in-distribution and, although it can be tuned with a strength parameter, gives coarse control over how much to shift. All in all, none of the existing methods provide a principled *control* strategy—defined here as one that guarantees the activations meet a precise target specification—rather than merely *steering* them in a general direction with the hope of reaching a desired region, often disregarding intermediate regions. In contrast, our method offers control by provably characterizing the distributional structure of the activation space, enabling precise control of the activations.

This, in turn, enables a more fine-grained steering of the token generation, including bidirectional steering along the full spectrum of the attribute to be controlled.

3 Problem Statement

In this section we present the problem studied in this paper, as well as the assumptions and approach. In particular, we approach the problem of controlled language generation as a standard optimal control problem in the field of control theory (Kirk, 2004).

3.1 Problem Formulation

Given a language model, controlled language generation aims to steer the model's output into a desired one. We study the problem of setting attributes of the model's output text, like tone or toxicity, to a certain range. In practice, this range is often quantified via numerical scores, for example, constraining text perplexity to a subset of $\mathbb{R}_+$, text toxicity to a subset of the Likert scale from 1 to 5, or likelihood of having a positive sentiment greater than a value in $[0, 1]$. Formally, an *attribute* is a function $a : \Sigma^* \to \mathcal{A}$ from a language model's string output to a numerical score or categorical label in $\mathcal{A}$.[1] In this work, we consider how to steer the output generation of an *already trained model* towards such a user-defined desired range $\mathcal{A}^* \subset \mathcal{A}$. Specifically, the requirements for the generated output sequence are two fold: its latent trajectory (a) is *guaranteed* to lie in the allowed region, and (b) stays as close as possible to that of the original output sequence, so that text quality is not compromised. In doing this, two questions need to be answered:

1. Given desired attribute scores $\mathcal{A}^* \subset \mathcal{A}$, how can the allowed region be defined for a given language model in latent space?
2. How can an intervention be designed to *guarantee* that the output stays within the allowed region, as defined by scores, while retaining maximal similarity with the original model?

In what follows, we answer the above questions and show that the proposed approach adds minimal computational overhead to language generation without modifying model weights.

3.2 Approach

We design an online method that, by acting linearly on the activations, precisely steers each token generation so that a specific attribute a of the sequence remains within $\mathcal{A}^*$. To do this, we employ a control-theoretic approach at the level of activations. Given the sequential, feedforward nature of LM layers, we consider each new token generation to realize a trajectory through the layers' activation spaces. In particular,

$$x_0 = E(s), \quad x_{t+1} = \ell_{t+1}(x_t), \quad y = U(x_T), \qquad \text{with } t = 0, \dots, T-1 \tag{1}$$

where E and U are the embedding and unembedding maps respectively, ℓ_t is the tth LM layer, T is the number of layers in the LM, $s \in \Sigma^*$ is the prompt sequence, and $x_t \in \mathbb{R}^d$ is the latent representation of string s after layer l_t.

Our strategy is to find a region $\mathcal{X}_t$ of layer t's latent space corresponding to the desired output range $\mathcal{A}^*$. Concretely, we provide a control mechanism by altering each layer's vector embedding, such that latent trajectories are guaranteed to lie in $\mathcal{X}_t$, i.e.,

$$x_0 = E(s), \quad \tilde{x}_t = x_t + \theta_t(x_t), \quad x_{t+1} = \ell_{t+1}(\tilde{x}_t), \quad y = U(x_T), \quad \text{with } t = 0, \dots, T-1, \tag{2}$$

where $\theta_t \in \mathbb{R}^d$ is a control input to the activation after layer t. For each token generation pass, this control mechanism is to be applied to a number of layers: as others have shown that semantic steering performs best when done in intermediate layers (Rimsky et al., 2024), the layers to be controlled is a design parameter, $\mathcal{T} \subset [0, \dots, T-1]$, that we explore experimentally in Section 6. Using a control-theoretic optic, we argue that intervening on several layers across the generation pass allows for robustification to the effects of the intervention, since unintended downstream deviations can be corrected by subsequent interventions in later

[1] Notation: Σ^* is the set of input strings, formally, the Kleene closure over the alphabet Σ.

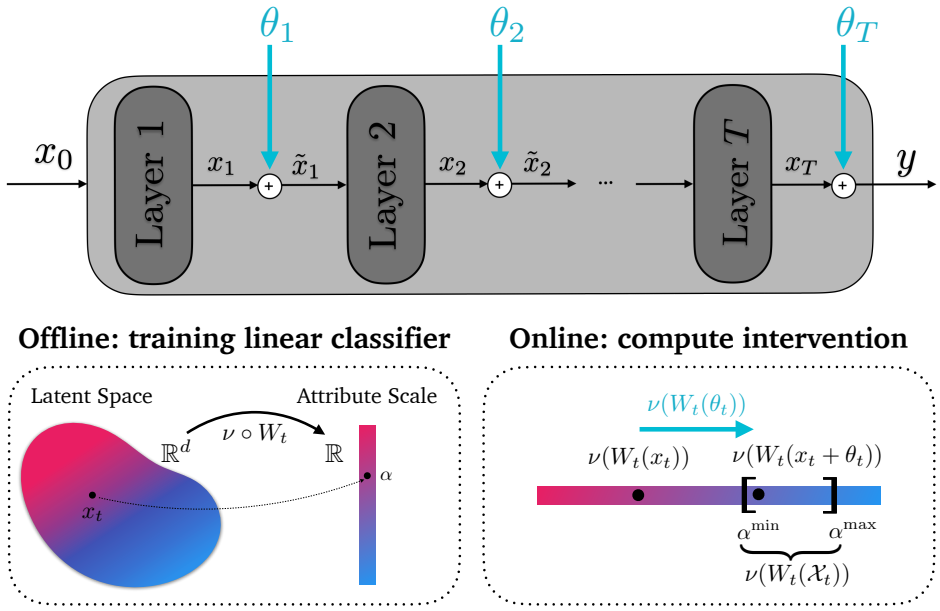

Figure 2: LiSeCo is based on applying a control intervention to the activations after each layer. The intervention is the result of applying a probing classifier $f_t = \nu \circ W_t$ mapping from the latent space $\mathbb{R}^d$ to the attribute space $\mathbb{R}$. Given an input sequence, the probe is trained to map the activation of each layer, x_t for every layer t, to its corresponding attribute score for the input sequence, α. The classifier is then used at inference time to characterize the allowable region ($\mathcal{X}_t$) to which each latent state $\tilde{x}_t$ is constrained. Keeping trajectories (sequences of $\{x_t\}_{t=1}^T$) out of the disallowed region in latent space is equivalent to keeping their image out of the disallowed region in attribute space. At inference time, the state in latent space ($x_t \in \mathbb{R}^d$) is mapped via the learned classifier. If it falls outside of the bounds (forbidden region), an intervention ($\theta_t \in \mathbb{R}^d$) is computed via **optimal control** as to guarantee the updated state $x_t + \theta_t \in \mathbb{R}^d$ lies in the allowable region.

layers. We note that this control intervention acts directly on the representation of the token to be generated, is designed online, and it depends on all previous tokens in the sequence.

The goal of this paper is to design the control input $\theta_t : \mathbb{R}^d \to \mathbb{R}^d; x_t \mapsto \theta_t(x_t)$ such that $\tilde{x}_t := x_t + \theta_t(x_t) \in \mathcal{X}_t$ is guaranteed for each intervened layer $t \in \mathcal{T}$. In what follows, we provide an overview of the approach, which we illustrate in Fig. 2 and present in mathematical detail in Section 4.

3.2.1 At post-training time (offline): Semantic Probe.

We want to nudge the output attribute a towards our desired range $\mathcal{A}^*$ by intervening in latent space. Doing so requires access to the function $a : \Sigma^* \to \mathcal{A}$, which could be given by, e.g., an off-the-shelf toxicity classifier. In our case, we are interested in functions $a : \Sigma^* \to \mathbb{R}$ that assign strings to continuous ratings. Similar to Park et al. (2023), we take the view that, for each layer t, "safe" language occupies a region $\mathcal{X}_t$ of activation space. Formally, a sequence $s \in \Sigma^*$ falls within the desired score range $\mathcal{A}^*$ if and only if its corresponding representations $x_t \in \mathcal{X}_t$. Identifying the allowed region $\mathcal{X}_t$ depends on how the attribute a is encoded in latent space: let layer t's activations encode a as $f_t : \mathbb{R}^d \to \mathcal{A}$. Then, the region $\mathcal{X}_t$ in latent space can be identified as $\mathcal{A}^*$'s pre-image under f_t (see Lemma C.1 for formal statement and proof). The key insight is that the desired outcome $a \in \mathcal{A}^*$ is proxied by enforcing the activation $x_t \in \mathcal{X}_t$.

We rely on the *Linear Representation Hypothesis* Park et al. (2023), by which concepts such as toxicity or sentiment can be identified by means of a linear probe in embedding space. To that end, at each layer t we learn, via regression, a lightweight linear probe f_t that maps the latent state x_t to its output score $a(s)$. We define $f_t : \mathbb{R}^d \to \mathbb{R}; x_t \mapsto a(s)$, such that

$$f_t(x_t) = \nu(W_t^\intercal x_t), \tag{3}$$

where $W_t \in \mathbb{R}^{1\times d}$ is a vector and ν a strictly monotonic nonlinear map of choice, e.g., sigmoid, for the given application. We provide mathematical details for these maps in Section 4. Intuitively, the allowed region $\mathcal{X}_t$ of layer t is the pre-image of an *allowed* classification under f_t. That is, if $\mathcal{A}^* = [\alpha^{\min}, \alpha^{\max}]$ is the range of allowed scores, where $\alpha^{\min}, \alpha^{\max}$ are user-defined bounds based on desired performance on a certain attribute. Then

$$\mathcal{X}_t := \{x \in \mathbb{R}^d |\ \alpha^{\min} \leq f_t(x) \leq \alpha^{\max}\}. \tag{4}$$

We note that the admissible region is defined using representations of naturally occurring text, via linear probes trained on meaningful examples. Consequently, the constraint set lies within regions of activation space already visited during standard generation. When a constraint violation occurs, our controller applies a minimum-norm correction that projects the activation onto the admissible region. Among all feasible interventions, this projection yields the smallest possible deviation from the model's original trajectory, thereby preserving proximity to the manifold of semantically coherent activations. If the admissible region intersects this manifold, the corrected activation remains on (or arbitrarily close to) it.

We remark that LiSeCo's framework permits interpretation of $[\alpha^{\min}, \alpha^{\max}]$ in any continuous space, including, e.g., human rating space. Importantly, this setup allows us to **directly *control* activations** instead of merely steer them into a particular direction, since we restrict attributes to specifically chosen and interpretable scores. So far, this control ability has not been introduced in the literature.

3.2.2 At inference time (online): Optimal Control in Semantic Space

In this step, we apply the control intervention to the representation at each layer t to guide generations towards a desirable region. To this end, we make use of the probe f_t trained offline. We work under the assumption that the feature to be controlled is separable by a linear classifier, hence linearly controllable (Park et al., 2023). This means, the representation x_t can be controlled by means of an additive intervention, i.e. $\tilde{x}_t = x_t + \theta_t$, where θ_t is computed as an optimal control input at each layer t sequentially. In particular, θ_t is designed as

$$\theta_t = \theta_t(x_t; W_t, \nu, \alpha^{\min}, \alpha^{\max}) \quad \text{such that} \quad \tilde{x}_t = x_t + \theta_t \in \mathcal{X}_t. \tag{5}$$

Mathematically, at layer t we solve an optimization problem over θ_t where the pre-computed probe W_t enters as a hard constraint in the formulation. This control strategy guarantees that the latent state x_t remains in the allowed region and retains maximal similarity with the original model. We emphasize that θ_t is **computed online** and **is gradient-free at inference time**. The specific expression and derivation for θ_t is provided in Section 4.

4 Optimal Controller for Language Generation

In this section, we describe the theoretical contribution of this work. First, we provide the offline and online algorithms for the approach described in the previous section. Then, we provide the mathematical details, expressions, and derivations that ground all of this. In particular, we show the training procedure for the probing classifier that allows for linear (additive) control interventions. Then, using the probing classifier, we design a controller to restrict text generation to the safe region. The optimal intervention is derived in closed form, thus computationally efficient at inference-time. Lastly, we compare existing methods with LiSeCo and prove a control theoretic interpretation for their proposed approaches as well.

4.1 Identification of Allowed Region

In order to keep the generation within the allowable score range $[\alpha^{\min}, \alpha^{\max}]$, we learn the scoring function from data: string-score pairs, $(s, \alpha) \in \Sigma^* \times \mathcal{A}$. In particular, at each layer t we learn a lightweight linear probe $f_t : \mathbb{R}^d \to \mathcal{A}; x_t \mapsto \alpha$ that maps the encoded latent state $x_t \in \mathbb{R}^d$ representation of string s to its score $\alpha \in \mathcal{A}$.[2]

[2]Note that the learning task is regression-like, not classification-like– we want the probes to be calibrated to the scoring function, not just the binary labels.

While LiSeCo supports any invertible nonlinearity ν, we will focus on the case where ν is the sigmoid. In this case, scores α are the *likelihood a sentence has a certain attribute*. Then, for each layer t, we minimize with respect to W_t the following loss over the dataset $\{s^{(i)}, \alpha^{(i)}\}_{i=1}^{N}$ of (string, score) pairs. The loss is the cross entropy between the scores α and the probes f_t:

$$\min_{W_t} \mathcal{L}_t(s, \alpha) = \min_{W_t} -\sum_{i=1}^{N} \left(\alpha^{(i)} \log \overbrace{\nu(W_t^\top x_t^{(i)})}^{f_t(x_t)=\hat{p}(\text{toxic})} + (1-\alpha^{(i)}) \log(\overbrace{1-\nu(W_t^\top x_t^{(i)})}^{\hat{p}(\text{nontoxic})}) \right), \tag{6}$$

where $x_t^{(i)}$ is the string $s^{(i)}$'s representation at layer t. The cross entropy loss is minimized when $f_t(x_t^{(i)}) = \alpha^{(i)}$ for all datapoints i.

Algorithm 1 summarizes the post-training computations to be carried out offline.

Algorithm 1 Post-training computation (offline)

1: **Input:** Labeled dataset $\{(s^{(i)}, \alpha^{(i)})\}$
2: **Output:** Classifier W_t
3: **for** $t \in \mathcal{T}$ **do**
4: Extract activations from Eq. 1: $x_t^{(i)} \leftarrow \ell_t(\ldots(E(s^{(i)}))$
5: Train probe using Eq. 6 on $\{(x_t^{(i)}, \alpha^{(i)})\}$ to obtain W_t
6: **end for**

4.2 Optimal Controller Design in Linear Feature Space

The goal is to design an intervention at each layer t such that the output activation x_t is modified to guarantee that it lies within the allowed region $\mathcal{X}_t$. Mathematically, this is to compute θ_t for $\tilde{x}_t := x_t + \theta_t(x_t)$ such that the goal is satisfied. In what follows, we show how θ_t can be seen as the solution to a constrained optimal control problem. We first pose the problem mathematically, and then introduce a relaxation that allows for an efficient online computation of the intervention θ_t. We remark that, as shown in Lemma C.1, control in activation space leads to reliable steering for output sequences.

Optimal Control Setup

The optimal controller aims to keep latent trajectories out of the unsafe region without compromising output quality. That is, we perform constrained optimization where latent trajectories maximally approximate the original ones (proxying text quality) while avoiding the unsafe region as defined by the probe. This gives rise to the following optimization problem:

$$\min_{\{\theta_t\}_{t\in\mathcal{T}}} \quad \sum_{t\in\mathcal{T}} \|\theta_t\|_2^2 \tag{7a}$$

$$s.t. \quad \alpha^{\min} \leq \nu(W_t^\top(x_t + \theta_t)) \leq \alpha^{\max}, \quad \forall t \in \mathcal{T} \tag{7b}$$

$$x_{t+1} = \ell_t(x_t + \theta_t), \quad \forall t = 1, \ldots, T \tag{7c}$$

$$x_0 = E(s), \tag{7d}$$

where s is the prompt sequence string. Optimization problem 7 aims to find the minimum l_2-norm intervention[3] θ_t for $t \in \mathcal{T}$ (Eq. 7a) that satisfies the following constraints: Eq. 7b requires the modified activation $x_t + \theta_t$ be classified as disallowed by the probe $\nu \circ W_t^\intercal$; Eq. 7c captures LM dynamics, i.e., layer t maps the modified activation x_t, θ_t to the next latent state x_{t+1}; Eq. 7d states that the LM's input embeds the input context, so

[3]The choice of L_2 norm is standard in classical optimal control problems. The L_2 norm is usually interpreted as energy, or effort, of the control input to the system. In this context, it can be seeing as trying to minimize the "effort" of the intervention. Moreover, the fact that L_2 is also used to measure distances in an Euclidean space, like the embedding space that we consider in this work, makes it an appropriate choice to measure the "similarity" between the intervened representation and the original one.

that interventions are *context-dependent*. The intervention that solves optimization problem 7 is *guaranteed by construction* to keep intervened activations $\tilde{x}_t$ $\forall t \in \mathcal{T}$ in the allowed region.

Whether attribute control is expressed as a cost or a constraint depends on the use case. Other approaches, in contrast to ours, encode attribute control in the optimization objective, but not via hard constraints (Dathathri et al., 2019; Hernandez et al., 2023). LiSeCo's constrained optimization framework also permits this interpretation by relaxation of constraints; though we leave its testing to future work, we state its equivalent problem and prove its *closed-form optimal solution*, which has only been empirically approximated by hyperparameter search in the literature (Li et al., 2023a), in Appendix E.

Optimal Controller Computation

Optimization problem 7 is a standard problem in the optimal control literature (Kirk, 2004). By Bellman's Optimality Principle, the standard approach to solving problem 7 is dynamic programming (DP) (Kirk, 2004): the optimal solution is computed for the last layer T, then via backward induction for $T-1, \ldots, 1$. But, layer dynamics 7c are highly non-convex, and solutions incomputable in closed form, hence their optimality is not guaranteed. Further, DP requires gradient backpropagation at each LM forward pass, adding significant inference latency.

To overcome these limitations, we relax problem 7. No longer searching for a globally optimal solution across layers, we now search for locally optimal solutions at each layer. Now, Eqs. 7c and 7d cease to play a role, as each layer is optimized for separately. Then, problem 7 is relaxed such that at each controlled layer $t \in \mathcal{T}$, the intervention θ_t is defined as the solution to the following optimization problem:

$$\theta_t^* = \underset{\theta_t \in \mathbb{R}^d}{\operatorname{argmin}} \quad \|\theta_t\|_2^2 \tag{8a}$$

$$s.t. \quad \alpha^{\min} \leq \nu(W_t^\top (x_t + \theta_t)) \leq \alpha^{\max} \tag{8b}$$

The sequence of θ_t that solve problem 8 may not optimize the original formulation 7. However, one is not anyway guaranteed to find global optima anyway due to the high nonconvexity of layer computations. Furthermore, optimality is not essential as the cost aims only to preserve similarity with the original model. Meanwhile, the guarantee to avoid unsafe region $\mathcal{X}_t$ is still enforced via Eq. 8b.

A key advantage of relaxed formulation 8 is that it is solvable in closed-form, per-layer, with minimal computational overhead. The following theorem states the analytical solution for optimal θ_t.

Theorem 4.1 (Optimal θ)**.** *The optimal solution $\theta_t^* \in \mathbb{R}^d$ to the optimization problem 8 is given by Table 1:*

Condition	$\nu(W_t^\top x_t) > \alpha^{\max}$	$\nu(W_t^\top x_t) < \alpha^{\min}$	otherwise
$\boldsymbol{\theta_t^*}$	$\dfrac{\nu^{-1}(\alpha^{\max}) - W_t^\top x_t}{\|W_t\|_2^2} W_t$	$\dfrac{\nu^{-1}(\alpha^{\min}) - W_t^\top x_t}{\|W_t\|_2^2} W_t$	0

Table 1: Optimal value of intervention θ_t^* at layer t.

Proof. Proof relies on leveraging the KKT conditions. See Appendix D for details. □

Geometrically, the optimal solution is the vector from x_t to the closest point in $\mathcal{X}_t$. When $x_t \notin \mathcal{X}_t^C$, which is the set-complement of $\mathcal{X}_t$, no update is needed; hence $\theta_t^* = 0$. Otherwise, the update is a factor of W_t. We note that the value of the *control* intervention θ_t^* depends on the current latent state x_t, and it magnitude and direction are explicitly dependent on x_t. This is in contrast to many *steering* methods, where the activations are often over- and under-steered towards a constant direction with a constant magnitude. Moreover, since θ_t^* exists in closed-form, computing an intervention incurs negligible computational cost. Crucially, it is guaranteed to keep the latent state outside the disallowed region.

Although control occurs *locally* at each layer, the local control steps result in a globally allowed distribution over the next token. To see this, consider a single token generation. Each sequential control action at layer t

guarantees that latent state $\tilde{x}_t \in \mathcal{X}_t$ is classified as "allowed", or equivalently, eliminates the set of disallowed trajectories. By the time we reach the last layer T, the latent trajectory is guaranteed to have been rated as "allowed" at every preceding intervened layer. Then, the last layer T's activation is transformed via the unembedding matrix (linear map) and softmax to the distribution over the vocabulary, then sampled to produce the next token $\tau \in \Sigma$. As a result of the control in the activations, the LM's output is guided towards scoring in the allowable range $\mathcal{A}^*$.

Algorithm 2 summarizes online generation of intervened representations. The problem presented in optimization problem equation 8 and Algorithm 2 addresses the most general case of linear semantic classification. A special case of this one is setting an attribute below (or above) a threshold p, as in the example of toxicity avoidance. The optimization problem is 8 is relaxed into

$$\min_{\theta_t} \quad \|\theta_t\|_2^2 \tag{9a}$$

$$s.t. \quad \sigma(W_t^\top(x_t + \theta_t)) - p \leq 0, \tag{9b}$$

for each layer $t = 1 \cdots T$. Optimal θ_t is given by the following corollary:

Algorithm 2 Inference time computation (online)

Input: Prompt s, control layers $\mathcal{T}$, parameters W_t, ν, $\alpha^{\min}$, $\alpha^{\max}$
Output: Generated token τ
$x_0 \leftarrow E(s)$
for $t \in [1, \ldots, T]$ **do**
  Compute activation from Eq. 1: $x_t \leftarrow \ell_t(x_{t-1})$
  if $t \in \mathcal{T}$ **then**
    Compute score from Eq. 3: $p \leftarrow \nu(W_t^\intercal x_t)$
    Solve for θ_t (from Table 1) using W_t, $\alpha^{\min}$, $\alpha^{\max}$:
    if $p > \alpha^{\max}$ **then**
      $\theta_t \leftarrow \frac{\nu^{-1}(\alpha^{\max}) - W_t^\top z_t}{\|W_t\|_2^2} W_t$
    else if $p < \alpha^{\min}$ **then**
      $\theta_t \leftarrow \frac{\nu^{-1}(\alpha^{\min}) - W_t^\top z_t}{\|W_t\|_2^2} W_t$
    else
      $\theta_t \leftarrow 0$
    end if
    Compute modified representation: $x_t \leftarrow x_t + \theta_t$
  end if
end for
$\tau \leftarrow U(x_T)$

Corollary 4.2 (Optimal θ, threshold)**.** *The optimal solution $\theta_t^* \in \mathbb{R}^d$ to the optimization problem 9 is given by*

$$\theta_t^* = \frac{\nu^{-1}(p) - W_t^\top x_t}{\|W_t\|_2^2} W_t \tag{10}$$

if $\nu(W_t^\top x_t) > p$, and $\theta_t^* = 0$ otherwise.

While the emphasis here is on tuning of an attribute over a continuous range of scores, LiSeCo also permits a binary attribute classification, e.g., toxic vs non-toxic. To that end, the value of p could be interpreted as the probability that a given generation is, for instance, toxic.

Remark 4.3. In the degenerate case where $\alpha^{\min} = \alpha^{\max} =: p$, the intervention is always as in equation 10 for all values of $\nu(W_t^\top x_t)$. This case corresponds to the application of setting an attribute to a specific value. Although, in theory, it is possible to do this, in practice it is impossible to guarantee that the scores of the generations will be equal to p do to the uncertainty introduced by the classifier, as well as numerical errors. Further robustness analysis to ensure that score is within a ball around p is left for future work.

5 Experimental Methods

In this section we provide a description of the LiSeCo pipeline. First, there is an initial probe training phase to find the unsafe regions and probes per layer, see Algorithm 1. Then, probes are integrated into the model at inference-time and the optimal intervention dynamically applied, see Algorithm 2. In this paper, LiSeCo was tested on three separate steering tasks: **toxicity**, **sentiment**, and **output language (English → Spanish)**.

Models We test on three state-of-the-art causal language models: Llama-3-8B (Meta, 2024), Gemma-2-2b (Team et al., 2024), and Mistral-7B (Jiang et al., 2023). While the architectural details of a layer (attention + MLP) differ slightly between models, our intervention treats layers as black boxes and operates at the level of the *residual stream* (Elhage et al., 2021). This permits our intervention to be applied as a lightweight layer wrapper, in an architecture-agnostic way.

Attribute scoring functions Recall that LiSeCo is trained with respect to a scoring function $a : \Sigma^* \to \mathbb{R}$ that the practitioner has access to. Therefore, in evaluating whether guarantees hold, we use a to not only label the training points in the constraint set, but also evaluate the generations at test time. While one can use *any* scoring function $a : \Sigma^* \to \mathbb{R}$, we use off-the-shelf neural classifiers from Huggingface. In particular, to score **toxicity**, we choose a to be the RoBERTa-based toxicity scorer (Logacheva et al., 2022) that maps sentences to a likelihood of being toxic in $[0, 1]$. Logacheva et al. (2022)'s classifier is trained on binary classification on Kaggle's Jigsaw dataset (Adams et al., 2017). To score **sentiment**, we similarly choose a to be a RoBERTa-based sentiment classifier (Camacho-collados et al., 2022), trained on annotated Twitter data (Barbieri et al., 2020), which assigns sentences to the likelihood of being negative in $[0, 1]$. To score the **language** generation task, we used the RoBERTa-based language classifier of Conneau et al. (2020), which maps text to the likelihood of being one of twenty languages (including our targets English and Spanish).

5.1 Offline step: Probe calibration

We explain the offline (calibration) step of the LiSeCo pipeline, i.e., dataset preprocessing and probe training.

5.1.1 Probe-training dataset

We test our method on fine-grained steering of text toxicity and negativity. Borrowing terminology from Ashok & Poczos (2024), we first learn probing classifiers f using a labelled *constraint dataset*. Then, we evaluate text generation on a *task dataset*.

For **toxicity**, we use Kaggle's Jigsaw dataset (Adams et al., 2017) as the constraint dataset. The dataset contains 30k label-balanced natural language comments. Then, we use our toxicity scorer to label all sentences in the constraint set to produce a probe training set of (sentence, score) pairs.

As **sentiment** datasets tend to be domain-specific (e.g., movie reviews), we combine several datasets to form the constraint dataset (N =30k). This consists of +/- label-balanced samples of 7500 datapoints each from IMDb film reviews (Maas et al., 2011), Tweets (Barbieri et al., 2020), Yelp reviews (Zhang et al., 2015), and Amazon reviews (Hou et al., 2024). For preprocessing details, see Appendix G. We score all texts using Camacho-collados et al. (2022) to produce the probe training set of (sentence, score) pairs.

For **language generation (English→Spanish)**, we used the FLORES-Plus dataset (NLLB Team et al., 2024). This contains $N = 2010$ parallel English and Spanish sentences collected from Wikinews, Wikivoyage, and Wikibooks. Because sentences contain only one language, we do not create labels by scoring the texts ourselves. Instead, we simply take the gold binary label (Spanish or English).

5.1.2 Probing classifiers

Our theoretical guarantees rely on a key assumption: that at each layer t, there indeed exists a $\mathcal{R}_t$ separable by linear f_t which together capture a semantics of the text being generated. We first verify, using a linear probe, that it is possible to learn the text attribute score from each layer of the LM. Towards this aim, we

split each of the constraint datasets into an 80% training set and 20% validation set. Then, for each model, dataset, and layer, we extract the last token hidden representations $x_t \in \mathbb{R}^d$ for each training sequence; we choose the last token embedding to represent the entire sequence, as in causal LMs, it is the only to attend to the entire input sequence. We then train one binary classifier f_t per-layer to minimize the cross-entropy loss between the probe prediction and ground-truth scorer in $[0, 1]$. See Appendix H for implementation details.

5.2 Online step: Text generation

For each LM, we insert trained probes f_t at each layer to evaluate layer-wise toxicity likelihood at each forward pass. If layer t's representation x_t is evaluated toxic, then the control input θ_t is dynamically applied. We fix text generation for all methods to max 100 new tokens with top-$p = 0.3$ sampling, a temperature of 1.0 and repetition penalty of 1.2, the same as in published baselines (Rodriguez et al., 2024; Li et al., 2023a).

5.2.1 Baselines

To the best of our knowledge, there are no baselines in the literature offering native guarantees. Therefore, we report only on LiSeCo for fine-grained *activation control*, but we compare LiSeCo against existing methods for *attribute reduction* or *induction* on the text generation. For **toxicity and negativity reduction and English → Spanish**, we test several baselines: no-control and prompting with instruction-tuned models, as well as two activation steering methods Activation Addition (ActAdd) (Turner et al., 2023) and Linear AcT (Rodriguez et al., 2024).

Instruction-tuned models All tested models have instruction-tuned variants. During evaluation, we prompt the instruction-tuned model using a template whose instructions are slightly modified from Mistral's system prompt provided in Jiang et al. (2023) (see Appendix I for details).

ActAdd Like LiSeCo, ActAdd steers text generation in activation space (Turner et al., 2023). For each model, the steering vector is computed as follows: (1) a source and target prompt, e.g., ("hate"→"love"), are each fed through the model and activations collected; (2) for each layer, the steering variable is computed as the difference from source to target activation; (3) at inference time, the steering variable is added to the intermediate representations of the input data. Like LiSeCo, ActAdd is gradient-free at inference-time. But, there are key differences: since steers derive from natural language prompts, ActAdd does not require a supervised learning phase on annotated data as in LiSeCo. For the same reason, the method lacks guarantees. For implementation details, see Appendix J.

AcT Similar to LiSeCo, AcT also steers text generation in activation space (Rodriguez et al., 2024). Using an optimal transport framework, an optimal transport map between two distributions of outputs (toxic and nontoxic) is learned offline at post-training. At inference time, this lightweight map is applied online to the activations being generated. Similar to LiSeCo, it is gradient-free at inference-time. However, there are some fundamental differences. In AcT, steering is done in-distribution and, although it can be tuned with a strength parameter, gives coarse control over how much to shift. Moreover, steering is only one-direction (from toxic to non-toxic) and is not used in a bi-directional manner. Moreover, it lacks guarantees on the effect of the interventions on the controllability of the method.

5.2.2 Evaluation

We evaluate LM generations on toxicity and sentiment steering. At the same time, we want our intervention to minimally compromise language modeling performance. To do so, we score generations' toxicity and sentiment, as well as proxy their naturalness using sequence perplexities.

Test set For the inference-time test set, we repurpose the datasets in Section 5.1.1. To make the test dataset for each task, we sample $N = 1000$ sentences from the respective dataset and truncate each to the first 10 words. We collect the (intervened) models' continuations, capped at maximum 100 new tokens, for evaluation.

Semantic control We rate text generation toxicity, sentiment, or language (English or Spanish) using the previously described attribute scoring functions. We convert the scorer's ratings into labels, where sequences are labeled toxic (negative, English) if the classifier returns a likelihood higher than 0.5, and non-toxic (positive, Spanish) otherwise.

The trained linear probes also provide toxicity likelihoods for the generated text, which we use to post-hoc validate LiSeCo, but not to evaluate generation toxicity/negativity/language per-se. The probe score returns the likelihood that a sequence is toxic/negative/English as determined by the probes' learned semantics, and is used to evaluate control in activation space.

Text naturalness The applied intervention ideally should not compromise language modeling performance. We quantify performance using the average perplexity (PPL) of generations under a different LM, Qwen-2.5-3B (Bai et al., 2023). We used a different model family to score PPL, given evidence that LMs are biased towards their own generations (Long et al., 2024).

6 Experimental Results

We first observe that toxicity and sentiment are approximately linearly represented in latent space (Park et al., 2024). We then demonstrate that LiSeCo predictably reduces the controlled attribute as a function of p while maintaining text naturalness. Second, we demonstrate that LiSeCo achieves precise control of activations, such that specifying the desired range $[\alpha^{\text{min}}, \alpha^{\text{max}}]$ indeed controls the activations' probe scores to that range. Finally, we show that LiSeCo performs competitively with existing baselines for attribute reduction while achieving the best naturalness, without extensive finetuning nor online inference latency.

6.1 Attributes are approximately linearly represented in latent space

Figure 3 shows, for all models, linear probe validation accuracies per-layer, averaged across 5 random seeds. Probes attain high accuracies of ∼90% for toxicity (Figure 3 left) and ∼ 80% for sentiment (Figure 3 right), confirming the disallowed toxic (negative) regions $\mathcal{R}_t$ are approximately linearly decodable, as predicted by the *Linear Representation Hypothesis* Park et al. (2023).

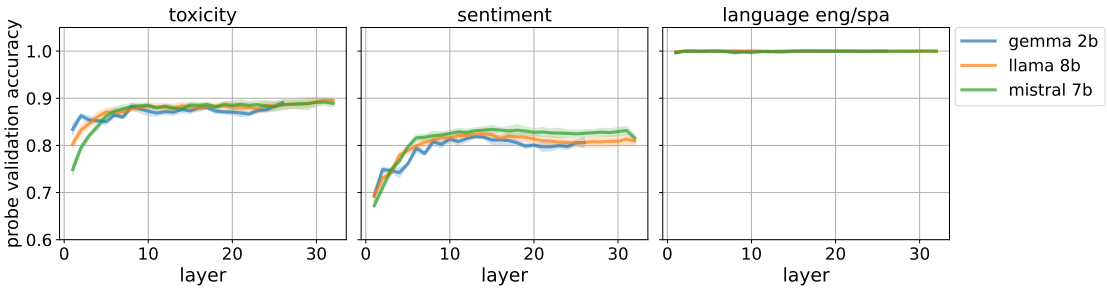

Figure 3: **Linear probe validation accuracy for toxicity (left), sentiment (middle), and language (right) detection.** All curves are shown ± 1 SD across 5 random seeds. Tasks converge to reasonable accuracies of ≥ 75% for most layers of all models, with mid-layers attaining ≈ 90% for toxicity, above 80% for sentiment, and nearly 100% for language detection.

While we use 80% of the constraint set to train the probes (N=24k for toxicity and sentiment, N=1.6k for language), we demonstrate in Appendix H that toxicity and sentiment probes can be learned with much fewer samples. Moving forward, toxicity and sentiment results are shown on the original $N = 24$k training set, as training each layer only took 2 minutes on an A30 GPU.

Finally, Figure 3, in line with prior work (Rimsky et al., 2024; Cheng et al., 2025), suggest to control activations starting from *intermediate layers*, as this is where high-level semantic attributes like sentiment

are most linearly decodable. We therefore apply LiSeCo on all layers after layer 8, where probing validation accuracy appears to plateau in Figure 3.[4]

6.2 LiSeCo achieves control with guarantees in activation space

LiSeCo controls activations to the correct safe set. To demonstrate this, we ran LiSeCo for various ranges $[\alpha^{\min}, \alpha^{\max}]$ from 0.01 ± 0.01 to 0.99 ± 0.01. If LiSeCo truly achieves control in activation space, then we expect the trained probes to score the layer activations, post-intervention, to between $[\alpha^{\min}, \alpha^{\max}]$.

Figure 4 demonstrates this in practice. Figure 4 shows the distribution of the intervened activations' attribute (toxicity, sentiment) scores, scored by the trained probes. Each point is the trained probe's score of a single layer; the bottom row of each plot depicts the activations' score distribution before LiSeCo (brown points), and other rows depict the distribution after LiSeCo. The desired regions of activation space, corresponding to attribute scores $[\alpha^{\min}, \alpha^{\max}]$ computed by the trained linear probes, are shown in green. No matter the LM or task, LiSeCo systematically controls *activations* (colored points) to the desired range.

6.3 Control in activation space translates to reliable steering in output space

Here, we study how control in activation space leads to reliable steering in output space. Specifically, we show that LiSeCo outperforms existing baselines on attribute steering and text naturalness. We show that controlling activations reliably steers the output attribute, where the dependence between LiSeCo α and the output attribute is empirically monotonic, but not identity. Finally, we discuss a path forward for guarantees in activation space to translate to guarantees on the output.

6.3.1 LiSeCo is competitive with baselines for steering and text quality

Empirically, all models without control produced toxic (negative) content on $N \approx 300$ of the original 1000 prompts, see Figure 5. To understand how baselines reduce toxicity and negativity, we first consider these would-be toxic (negative/English) generations. That is, for each LM, we scored the toxicity (negativity/English) of all 1000 no-control continuations using the neural classifiers in Section 5. Then, we filtered for prompts whose no-control continuations were toxic (negative). Finally, on the other baselines, we evaluated attribute reduction for this would-be toxic (negative) prompt set.

Figure 5 shows, for sentiment (top) and toxicity (bottom), the *safety-naturalness plane*, where the output text attribute is plotted against its perplexity. Each baseline's (safety, naturalness) distribution is shown as an ellipse centered at the mean, shown with one standard deviation. Only the best hyperparameter settings for each baseline are shown (LiSeCo $[\alpha^{\min}, \alpha^{\max}] = [0, 0.005]$. Moreover, LiSeCo reduces the desired attribute without sacrificing text naturalness, seen by blue ellipses (LiSeCo) being vertically aligned with the green ones (baseline). For English to Spanish steering (bottom), LiSeCo outperforms baselines by achieving near-perfect accuracy, while maintaining the lowest perplexity. In the case of negativity reduction (top row), LiSeCo outperforms all baselines for both attribute steering and text naturalness. LiSeCo's minimal effect on text naturalness is baked into its design, as it introduces the *minimum norm* intervention when the activation falls into the unsafe region, and does not intervene if the activation is already classified safe. The design choice of minimal intervention, by contrast, is not a part of prompting, ActAdd, or Act, where the intervention is always applied (Rodriguez et al., 2024; Turner et al., 2023). To that end, we show in Figure K.1 how while LiSeCo *abstains* from intervening if the generation is already classified safe, AcT, ActAdd, and prompting with Instruct are always applied. In practice, LiSeCo well-preserves the original safety and naturalness distribution as desired, while other methods may negatively impact either factor (see Appendix K).

6.3.2 Control in activation space permits reliable steering in output space

We have shown that LiSeCo achieves control in activation space. But, do guarantees in activation space translate to guarantees in generation space? Recall that Lemma C.1 states that activation guarantees will

[4] In preliminary experiments, we noted that applying LiSeCo on multiple layers worked better than a single layer, but that controlling early layers led to performance degradation. This may be because early layers still need to build a semantic representation of the text (Cheng et al., 2025; Lad et al., 2025).

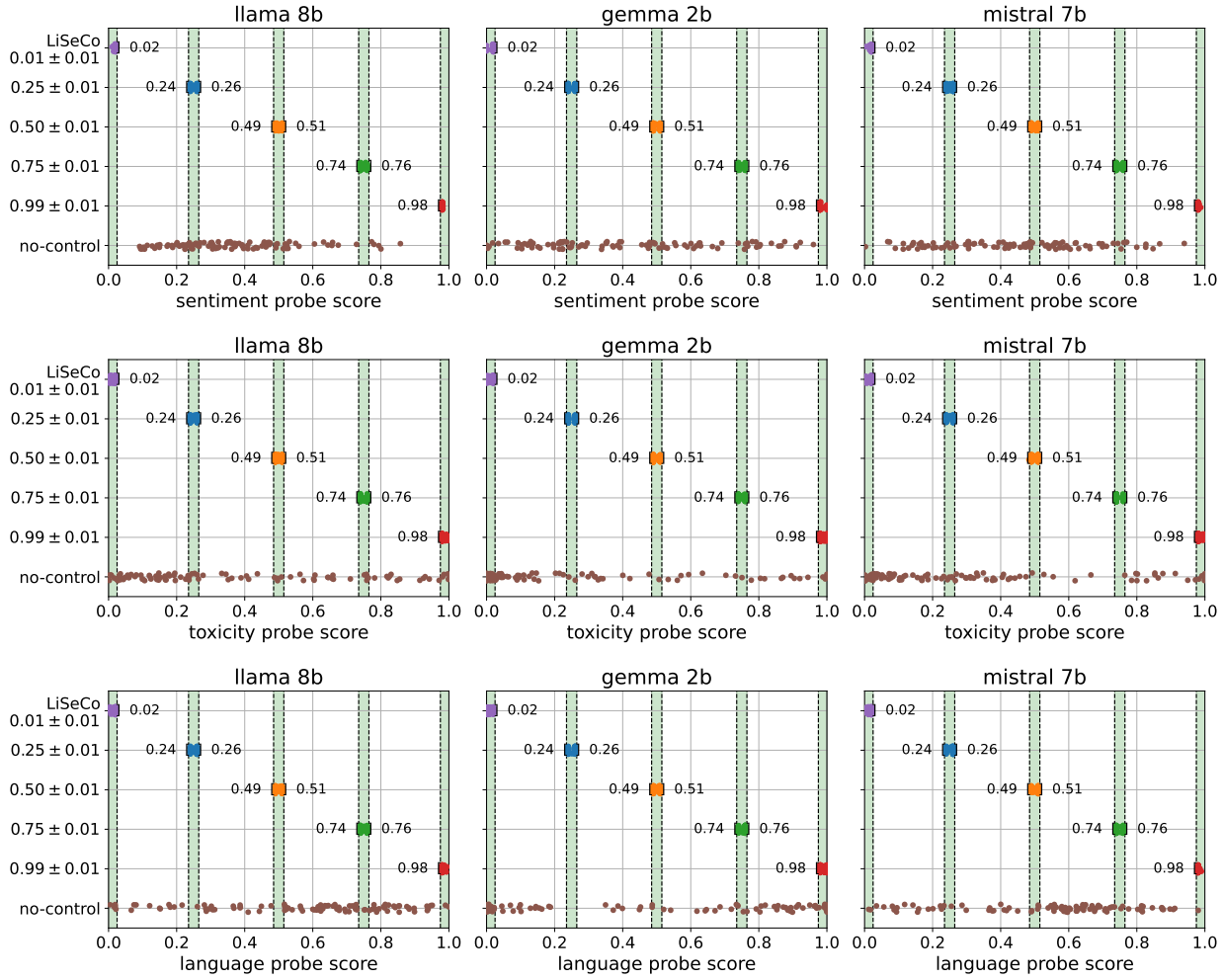

Figure 4: **LiSeCo controls attributes in activation space.** Attribute probe scores for LiSeCo are shown for sentiment (top), toxicity (middle), and language (bottom) on the models Llama, Gemma, and Mistral (left to right). The y-axis of each plot shows the desired range $[\alpha^{\text{min}}, \alpha^{\text{max}}]$, also shaded in green on the plot. The colored points are the actual distribution of the attribute as measured by the trained probes, after applying LiSeCo. A no-control baseline is shown on the bottom row, indicating the distribution if LiSeCo is not applied. In all cases, LiSeCo successfully controls activations to the desired range, seen by the colored dots falling within the green intervals.

transfer to outputs if the guarantee is calibrated for *every reachable point* in $\mathbb{R}^d$. This is a strong assumption that we hypothesize rarely holds in practice. Thankfully, empirically, we find that control in activation space translates to reliable *steering* in output space.

Figure 6 shows a clear monotonic trend between LiSeCo α and the number of unsafe generations. For all models and tasks, smaller LiSeCo α (x-axis) predictably decreases the proportion of toxic/negative generations (y-axis). This allows LiSeCo α to act as an interpretable knob that one can adjust at inference-time to obtain the desired effect. Interestingly, while LiSeCo effectively increases or decreases the target attribute, even extreme values of α do not result in 100% attribute reduction, a known pitfall of existing activation steering methods (Rodriguez et al., 2024; Turner et al., 2023). This suggests a shortcoming of the Linear Representation Hypothesis (Park et al., 2023), where linear decodability does not necessarily translate to exact linear controllability.

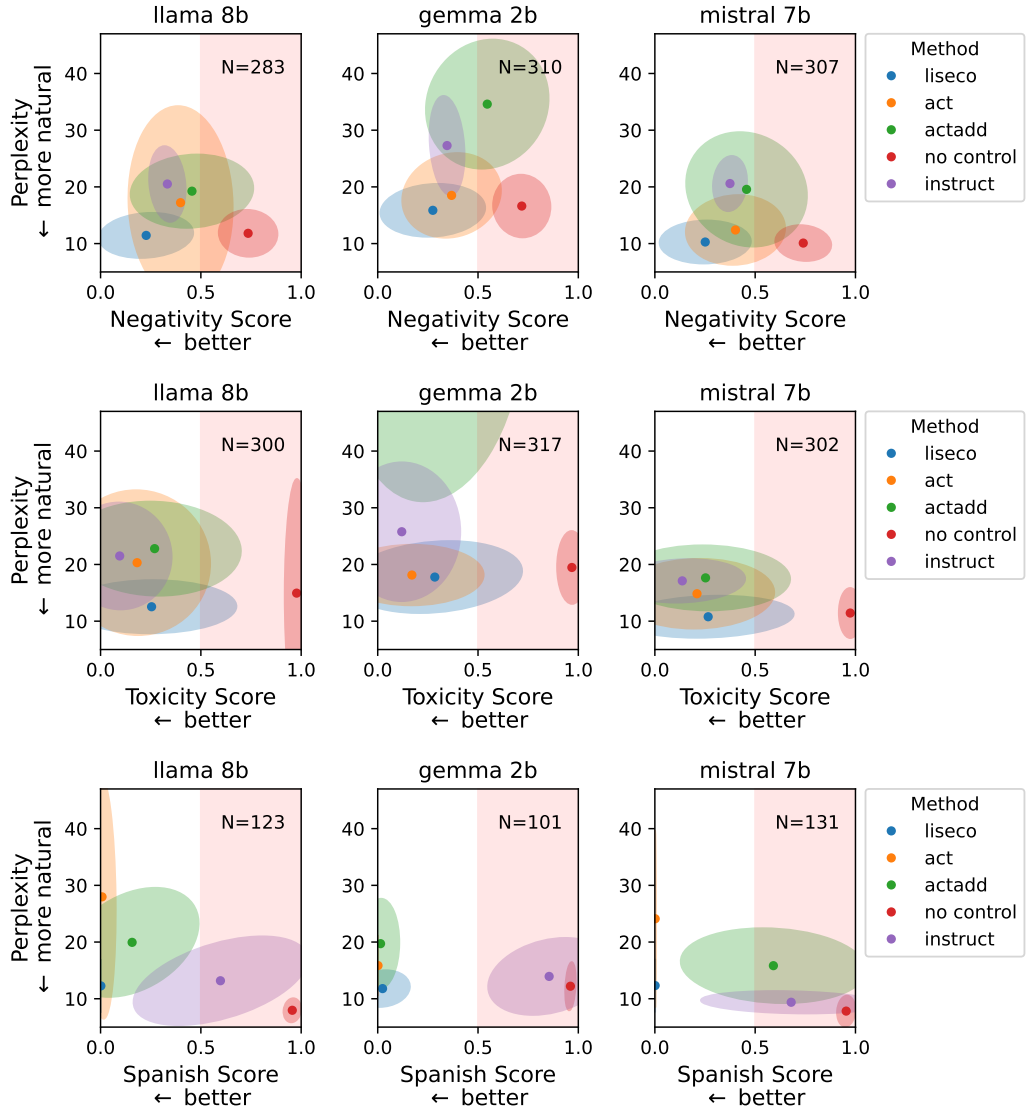

Figure 5: Generations are plotted on the *safety-naturalness plane* (bottom left is best). Each ellipse represents the mean $(\cdot)$ with one standard deviation of a single method; only the best hyperparameter setting is shown for each method. The shaded red region represents the region of the safety-naturalness plane that would be classified negative or toxic by the neural scorer. In general across models and tasks, **LiSeCo performs competitively to baselines while consistently demonstrating the highest naturalness**. In the negativity reduction task (top row), LiSeCo also demonstrates the best negativity reduction.

LiSeCo is able to steer output language to a wider range than sentiment and toxicity (Figure 6 top vs. bottom), achieving higher separation of outcomes with different intermediate settings of α. We hypothesize that this is because the sentiment training labels have better *coverage* over the entire score range, compared to the toxicity training labels (Figure 6 right); this permits a better calibration of the linear probe for intermediate ranges of α. Crucially, calibrating the probe on intermediate scores brings the setting closer to that described in Lemma C.1, which predicts probes calibrated *everywhere* to best steer output generations. This lends itself to a practical recommendation for the probe constraint set, where we expect worse performance in intermediate ranges for binary labels (toxicity and language, Figure 6 bottom-right), and better performance for continuous labels (sentiment, Figure 6 top-right).

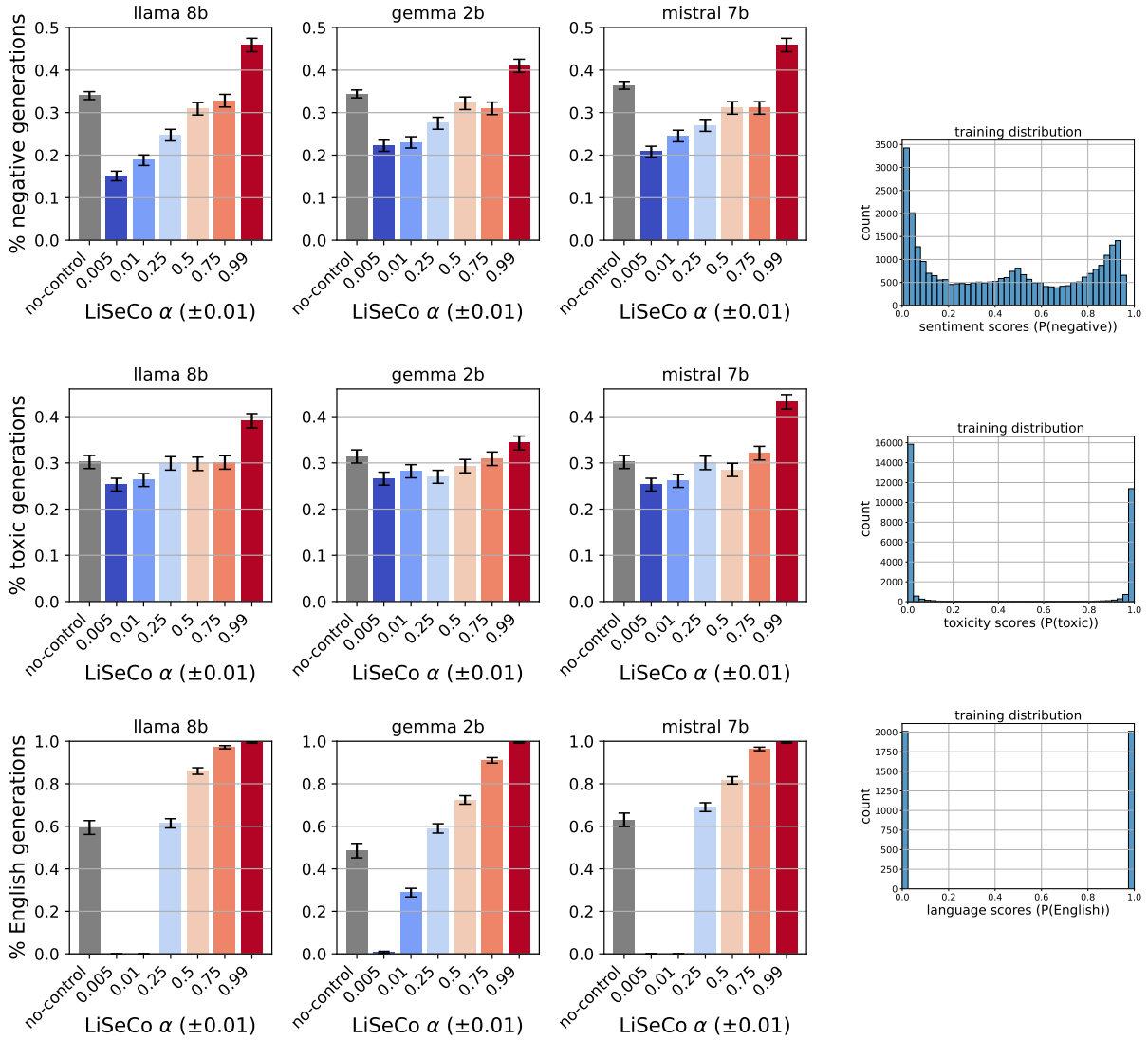

Figure 6: **(Left)** LiSeCo α steers output sentiment (top), toxicity (middle), and language (bottom). We show, for a range of $[\alpha^{\min}, \alpha^{\max}] = \alpha \pm 0.01$ (x-axis), the true proportion of toxic/negative generations (y-axis, $N = 1000$) with one standard error. For all settings, there is a **clear monotonic trend where smaller LiSeCo α translates to fewer unsafe generations**. For reference, the no-control baseline is plotted in gray for each setting. Intervention with LiSeCo $\alpha \approx 0.01$, $\alpha \approx 0.99$ significantly changes the distribution of unsafe generations, seen by non-overlapping error bars between gray and dark blue, dark red, respectively. **(Right)** The training set y-label distributions for sentiment scores (top), toxicity scores (middle), and language labels (bottom). Sentiment (top) achieved steerability to a wider range than toxicity (middle), though English→Spanish achieved the entire range of values between 0 and 1.

6.4 LiSeCo's out-of-distribution generalization (English→Spanish case study)

Recall that LiSeCo's linear probes are calibrated on a training set of annotated data. A key assumption for LiSeCo to work well at inference time is that the training distribution reasonably matches the inference distribution. Experiments until now have tested LiSeCo's performance *in-distribution*. In this section, we conduct a case study testing LiSeCo (English→Spanish, $\alpha^{\min} = 0, \alpha^{\max} = 0.005$) on three out-of-distribution English datasets: poetry, the EuroParl dataset of European parliamentary proceedings (Koehn, 2005), and

	llama 8b	gemma 2b	mistral 7b
in-distribution	$0.9871_{0.0002}$	$0.9127_{0.0112}$	$0.9877_{0.0002}$
poems	$0.9874_{0.0016}$	$0.8161_{0.0349}$	$0.9887_{0.0003}$
europarl	$0.9855_{0.0007}$	$0.6072_{0.0461}$	$0.9864_{0.0005}$
code	$0.8887_{0.0282}$	$0.2871_{0.0445}$	$0.9309_{0.0220}$

Table 2: **English → Spanish OOD Results, Average Prob(Spanish) ↑.** For each dataset (rows) and model (columns), we show the average rating $\pm SE$of the probability the generation is Spanish by the automatic classifier (Conneau et al., 2020). The three bottom rows, corresponding to a dataset of poems, European parliament proceedings, and code, are out-of-distribution, while the in-distribution performance is shown for reference at the top. How well the probes transfer OOD depends on the model and dataset, with performance being generally retained for Llama and Mistral, but not for Gemma.

code snippets from Stanford's CodeAlpaca (Chaudhary, 2023). For each dataset, we randomly sampled $N = 100$ sequences for inference.

Table 2 shows the OOD performance of each model (columns) on each dataset (rows), with the in-distribution performance for reference at the top. In the Table, each value is the average probability of a generation being Spanish as evaluated by the automatic classifier (Conneau et al., 2020), along with one standard error. How well probes transfer to the OOD datasets of poems, parliament proceedings (EuroParl), and code depends on the model; Llama and Mistral maintain very high English → Spanish steering performance, while Gemma's performance degrades out-of-distribution. For Gemma, a frequent error mode was to produce text in Romance languages that were not Spanish, for instance French or Portuguese. This highlights multi-class steering as an important direction for future work.

7 Discussion

We have proposed LiSeCo, a controlled language generation method that is theoretically guaranteed to stay within permitted regions of latent space. Empirically, the method produces non-toxic and non-negative, but still natural, text. By design, the parameters $\alpha^{\min}$ and $\alpha^{\max}$ were shown to steer the probability of generating unwanted text. LiSeCo is compatible with any layered deep learning architecture (not limited to Transformers), as it is agnostic to the layer computation (dynamics) and involves a negligible inference-time latency. In future work, we are interested in testing metrics beyond perplexity such as benchmark performance (Macocco et al., 2026), applying our approach to different tasks and joint constraints, as well as to alternatives to linear probes as the way to ascertain whether a token falls into the undesirable region. An important enhancement of the proposed method would be to study conditions under which the control of the activations directly translates to control in token space. Some preliminary theoretical conditions are provided in Appendix C, where we demonstrate necessary conditions for activation control to translate to output attribute control: in short, the linear encoding of the output attribute trained on data must apply uniformly to all regions of activation space $\mathbb{R}^d$.

Limitations With the increasing ubiquity of LMs comes a growing need to understand their behavior. LiSeCo helps address this need by providing practical and theoretical tools for LM interpretability and control. That said, using LiSeCo has several caveats: (1) it requires supervised learning of the linear probes on annotated data; (2) the intervention is only as good as the probes, which is only as good as their training data. Thus, when training probes, it is crucial that the training data well-represent the use domain, i.e., are *in-distribution*. In particular, although we showed that LiSeCo transferred well out-of-distribution for English→Spanish steering, LiSeCo may in general be less effective on out-of-distribution data. We emphasize that this bottleneck is inherent to any steering method that learns from data (Rodriguez et al., 2024; Dathathri et al., 2019; Li et al., 2023a). (3) LiSeCo will only work for features that are *linearly represented*. The practitioner needs to verify, via linear probes, that their attribute of interest is indeed linearly decodable with high performance. (4) LiSeCo works for *binary* attributes, e.g., toxic/non-toxic, English/Spanish. A needed extension would be to consider multiple classes, e.g., English/Spanish/Chinese. Finally, (5) LiSeCo

worked less well when steering prompts that *already* showed an undesirable attribute; for instance, when steering English prompts towards Spanish, several token generations were needed for the steering to take effect. This behavior is well-documented (Wolf et al., 2024), but future work should determine when and whether LiSeCo is able to steer out of undesirable regions. As such, LiSeCo functions as a *preemptive*, rather than retroactive, mechanism.

Ethics statement Controlling text generation has a dual-use implication, that is, it can be used for benefit or harm. While we have demonstrated our method on toxicity and negativity avoidance, it can equivalently be applied to increase harmful traits. The choice of feature and constraint set when designing the linear probes must be done carefully to ensure that it accurately reflects the use-case and is compliant with safety standards and free of harmful biases.

Reproducibility statement Code and data will be made public upon acceptance. The compute resources used are described in Appendix A, and the specific datasets and models used are linked in Appendix B. The proof of Theorem 1 is detailed in Appendix D.

References

CJ Adams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, Nithum, and Will Cukierski. Toxic comment classification challenge, 2017. URL https://kaggle.com/competitions/jigsaw-toxic-comment-classification-challenge.

Dhananjay Ashok and Barnabas Poczos. Controllable text generation in the instruction-tuning era. (arXiv:2405.01490), May 2024. doi: 10.48550/arXiv.2405.01490. URL http://arxiv.org/abs/2405.01490. arXiv:2405.01490 [cs].

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*, 2020.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: Perfect linear concept erasure in closed form. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=awIpKpwTwF.

Aman Bhargava, Cameron Witkowski, Manav Shah, and Matt Thomson. What's the magic word? a control theory of llm prompting. *arXiv preprint arXiv:2310.04444*, 2023.

Carrie Cai, Tongshuang Wu, and Michael Andrew Terry. Transparent and controllable human-ai interaction via chaining of machine-learned language models, April 13 2023. US Patent App. 17/957,526.

Jose Camacho-collados, Kiamehr Rezaee, Talayeh Riahi, Asahi Ushio, Daniel Loureiro, Dimosthenis Antypas, Joanne Boisson, Luis Espinosa Anke, Fangyu Liu, Eugenio Martinez Camara, et al. TweetNLP: Cutting-edge natural language processing for social media. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–49, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-demos.5.

Sahil Chaudhary. Code alpaca: An instruction-following llama model for code generation. https://github.com/sahil280114/codealpaca, 2023.

Emily Cheng, Diego Doimo, Corentin Kervadec, Iuri Macocco, Lei Yu, Alessandro Laio, and Marco Baroni. Emergence of a High-Dimensional Abstraction Phase in Language Transformers. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=0fD3iIBhlV.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747/.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8493–8502, 2022.

David Dalrymple, Joar Skalse, Yoshua Bengio, Stuart Russell, Max Tegmark, Sanjit Seshia, Steve Omohundro, Christian Szegedy, Ben Goldhaber, Nora Ammann, et al. Towards guaranteed safe ai: A framework for ensuring robust and reliable ai systems. *CoRR*, 2024.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*, 2019.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, 2021.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*, 2023.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *Proceedings of ICLR*, Online, 2022. Published online: `https://openreview.net/group?id=ICLR.cc/2022/Conference`.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2023.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Donald E Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pp. 79–86, Phuket, Thailand, September 13-15 2005. URL `https://aclanthology.org/2005.mtsummit-papers.11`.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style vectors for steering generative large language model. *arXiv preprint arXiv:2402.01618*, 2024.

Ben Krause, Akhilesh Deepak Gotmare, Bryan McCann, Nitish Shirish Keskar, Shafiq Joty, Richard Socher, and Nazneen Fatema Rajani. GeDi: Generative discriminator guided sequence generation. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4929–4952, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.424. URL `https://aclanthology.org/2021.findings-emnlp.424`.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554, 2021.

Vedang Lad, Jin Hwa Lee, Wes Gurnee, and Max Tegmark. Remarkable robustness of LLMs: Stages of inference? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=Wxh5Xz7NpJ.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=aLLuYpn83y.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024a.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 4582–4597, 2021.

Xiaopeng Li, Shasha Li, Shezheng Song, Jing Yang, Jun Ma, and Jie Yu. Pmet: Precise model editing in a transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18564–18572, 2024b.

Zhoubo Li, Ningyu Zhang, Yunzhi Yao, Mengru Wang, Xi Chen, and Huajun Chen. Unveiling the pitfalls of knowledge editing for large language models. In *The Twelfth International Conference on Learning Representations*, 2023b.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 6691–6706, Online, August 2021. Association for Computational Linguistics.

Varvara Logacheva, Daryna Dementieva, Sergey Ustyantsev, Daniil Moskovskiy, David Dale, Irina Krotova, Nikita Semenov, and Alexander Panchenko. ParaDetox: Detoxification with parallel data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6804–6818, Dublin, Ireland, May 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.acl-long.469.

Do Xuan Long, Hai Nguyen Ngoc, Tiviatis Sim, Hieu Dao, Shafiq Joty, Kenji Kawaguchi, Nancy F Chen, and Min-Yen Kan. Llms are biased towards output formats! systematically evaluating and mitigating output format bias of llms. *arXiv preprint arXiv:2408.08656*, 2024.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, et al. Neurologic a* esque decoding: Constrained text generation with lookahead heuristics. *arXiv preprint arXiv:2112.08726*, 2021.

Yifan Luo, Yiming Tang, Chengfeng Shen, Zhennan Zhou, and Bin Dong. Prompt engineering through the lens of optimal control. *arXiv preprint arXiv:2310.14201*, 2023.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Iuri Macocco, Pau Rodriguez, Arno Blaas, Luca Zappella, Marco Baroni, and Xavier Suau. On the effectiveness-fluency trade-off in LLM conditioning: A systematic study, 2026. URL https://openreview.net/forum?id=JAIymWFRU2.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372, 2022a.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372. Curran Associates, Inc., 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/6f1d43d5a82a37e89b0665b33bf3a182-Paper-Conference.pdf.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*, 2022c.

Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL https://ai.meta.com/blog/meta-llama-3/.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2021.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. Scaling neural machine translation to 200 languages. *Nature*, 630(8018): 841–846, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07335-x. URL https://doi.org/10.1038/s41586-024-07335-x.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Causal Representation Learning Workshop at NeurIPS 2023*, 2023. URL https://openreview.net/forum?id=T0PoOJg8cK.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 39643–39666. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/park24c.html.

Lianhui Qin, Sean Welleck, Daniel Khashabi, and Yejin Choi. Cold decoding: Energy-based constrained text generation with langevin dynamics. *Advances in Neural Information Processing Systems*, 35:9538–9551, 2022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Pau Rodriguez, Arno Blaas, Michal Klein, Luca Zappella, Nicholas Apostoloff, Marco Cuturi, and Xavier Suau. Controlling language and diffusion models by transporting activations. *arXiv preprint arXiv:2410.23054*, 2024.

Stefano Soatto, Paulo Tabuada, Pratik Chaudhari, and Tian Yu Liu. Taming AI bots: Controllability of neural states in large language models. https://arxiv.org/abs/2305.18449, 2023.

Nishant Subramani, Nivedita Suresh, and Matthew E Peters. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 566–581, 2022.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL https://arxiv.org/abs/2408.00118.

Alex Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

Lucas Weber, Elia Bruni, and Dieuwke Hupkes. Mind the instructions: a holistic evaluation of consistency and interactions in prompt-based learning. In Jing Jiang, David Reitter, and Shumin Deng (eds.), *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 294–313, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.conll-1.20. URL https://aclanthology.org/2023.conll-1.20.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yotam Wolf, Noam Wies, Oshri Avnery, Yoav Levine, and Amnon Shashua. Fundamental limitations of alignment in large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International*

Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 53079–53112. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/wolf24a.html.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962, 2024.

Kevin Yang and Dan Klein. FUDGE: Controlled text generation with future discriminators. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3511–3535, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.276. URL https://aclanthology.org/2021.naacl-main.276.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, 2023.

Jingying Zeng, Richard Huang, Waleed Malik, Langxuan Yin, Bojan Babic, Danny Shacham, Xiao Yan, Jaewon Yang, and Qi He. Large language models for social networks: Applications, challenges, and solutions. *arXiv preprint arXiv:2401.02575*, 2024.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

A Computing resources

Experiments were run on a cluster with 12 nodes with 5 NVIDIA A30 GPUs and 48 CPUs each.

Extracting LM representations took a few wall-clock hours per model-dataset computation. Training linear probes took around 2 minutes per layer, so overall 64 wall-clock hours. Running evaluation experiments took a total of 30 wall-clock hours.

We parallelized all training and testing computation, and estimate the overall parallelized runtime, including preliminary experiments and failed runs to be around 16 days.

B Assets

Llama `https://huggingface.co/meta-llama/Meta-Llama-3-8B`; license: llama3

Mistral `https://huggingface.co/mistralai/Mistral-7B-v0.1`; license: apache-2.0

Gemma `https://huggingface.co/google/gemma-2-2b`; license: gemma

Qwen `https://huggingface.co/Qwen/Qwen-2.5-3B`; license: qwen-research

PyTorch `https://scikit-learn.org/`; license: bsd

Toxicity constraint `https://huggingface.co/datasets/google/jigsaw_toxicity_pred`; license: CC0

Sentiment constraint `https://huggingface.co/datasets/stanfordnlp/imdb`; license: unknown.
`https://huggingface.co/datasets/cardiffnlp/tweet_eval`; license: unknown.
`https://huggingface.co/datasets/Yelp/yelp_review_full`; license: yelp-license.
`https://huggingface.co/datasets/McAuley-Lab/Amazon-Reviews-2023`; license: MIT.

Languages constraint `https://huggingface.co/datasets/openlanguagedata/flores_plus`; license: CC-4.0.

Code test set `https://huggingface.co/datasets/sahil2801/CodeAlpaca-20k`; license: CC-4.0.

EuroParl `https://huggingface.co/datasets/Helsinki-NLP/europarl`; license: unknown.

Poems `https://huggingface.co/datasets/suayptalha/Poetry-Foundation-Poems`; license: GNU Affero General Public License v3.0.

Toxicity classifier `https://huggingface.co/s-nlp/roberta_toxicity_classifier`; license: Open-RAIL++.

Sentiment classifier `https://huggingface.co/cardiffnlp/twitter-roberta-base-sentiment-latest`; license: CC-by-4.0.

Language classifier `papluca/xlm-roberta-base-language-detection`; license: MIT.

C Identifying the allowable region in latent space

We provide the following lemma, which gives a sufficient condition for control in activation space to transfer to control in attribute space. In brief, if the probe f_t at layer t is the same function as the attribute scorer $\circ$ the rest of the LM layers on all of $\mathbb{R}^d$, then if we constrain the image of f_t by constraining the layer t activations, then we equivalently constrain the end attribute.

Lemma C.1 (Identification of allowed region in activation space)**.** *Let x_t be the layer t activation. Let the probe $f_t : \mathbb{R}^d \to \mathbb{R}; x_t \mapsto f_t(x_t)$ and attribute $a : \Sigma^* \to \mathbb{R}$ satisfy*

$$f_t(x_t) = a \circ \underbrace{l_T \circ l_{T-1} \circ \cdots \circ l_{t+1}(x_t)}_{LM\ output} \quad \forall x_t \in \mathbb{R}^d. \tag{C.11}$$

Write $a_t = a \circ l_T \circ \cdots \circ l_{t+1}$. Then, for any $\mathcal{A}^* \subset \mathbb{R}$, if preimage$(l_{t+1})$ = preimage$_{f_t}(\mathcal{A}^*)$ for all x_t, then im$(a_t) = \mathcal{A}^*$.

Proof. It is given that, for all $x \in \mathbb{R}^d$, $f_t(x) = a_t(x)$. This means that $\text{preimage}_{f_t}(\mathcal{A}^*) = \text{preimage}_{a_t}(\mathcal{A}^*)$ for all sets $\mathcal{A}^* \subset \mathbb{R}$. Suppose the pre-image of l_{t+1} is modified such that $\text{preimage}(l_{t+1}) = \text{preimage}_{f_t}(\mathcal{A}^*)$. Applying a_t to both sides yields

$$a_t(\text{preimage}(l_{t+1})) = a_t(\text{preimage}_{f_t}(\mathcal{A}^*)) \tag{C.12}$$
$$\text{im}(a_t) = a_t(\text{preimage}_{a_t}(\mathcal{A}^*)) \tag{C.13}$$
$$\text{im}(a_t) = \mathcal{A}^*. \tag{C.14}$$

This completes the proof that setting $\text{preimage}(l_{t+1}) = \text{preimage}_{f_t}(\mathcal{A}^*)$ constrains $\text{im}(a_t) = \mathcal{A}^*$. □

D Proof of Theorem 4.2

Theorem D.1 (Optimal θ)**.** *The optimal solution $\theta_t^* \in \mathbb{R}^d$ to the optimization problem 8 is given by*

$$\theta_t^* = \begin{cases} \frac{\nu(\alpha^{max}) - w_t^\top x_t}{\|w_t\|_2^2} w_t & \text{if } \sigma(W_t^\top x_t) > \alpha^{max}, \\ \frac{\nu(\alpha^{min}) - w_t^\top x_t}{\|w_t\|_2^2} w_t & \text{if } \sigma(W_t^\top x_t) < \alpha^{min}, \\ \mathbf{0} & \text{otherwise}, \end{cases} \tag{D.15}$$

where $w_t := W_t^1 - W_t^2$, the difference of the columns of $W_t =: \begin{bmatrix} W_t^1 & W_t^2 \end{bmatrix}$.

Proof. We start by defining the Lagrangian for the optimization problem in Equation (8) as

$$L(\theta_t, \lambda_1, \lambda_2) = \|\theta_t\|_2^2 + \lambda_1(\alpha^{\min} - \nu(W^\top(x_t + \theta_t))) + \lambda_2(\nu(W^\top(x_t + \theta_t)) - \alpha^{\max}), \tag{D.16}$$

where $\lambda_1, \lambda_2 \in \mathbb{R}$ are the Lagrange multipliers.

We now solve this optimization problem by using KKT conditions, which are first-order necessary conditions for optimality:

1. Stationarity.

$$0 \in \partial(\|\theta_t\|_2^2 + \lambda_1(\alpha^{\min} - \nu(W^\top(x_t + \theta_t))) + \lambda_2(\nu(W^\top(x_t + \theta_t)) - \alpha^{\max})) \tag{D.17}$$

2. Complementary slackness.

$$\lambda_1(\alpha^{\min} - \nu(W^\top(x_t + \theta_t))) = 0 \tag{D.18}$$
$$\lambda_2(\nu(W^\top(x_t + \theta_t)) - \alpha^{\max}) = 0 \tag{D.19}$$

3. Primal feasibility:

$$\alpha^{\min} \leq \nu(W^\top(x_t + \theta_t)) \leq \alpha^{\max} \tag{D.20}$$

4. Dual feasibility:

$$\lambda_1, \lambda_2 \geq 0 \tag{D.21}$$

We now consider three cases:

Case 1: $\nu(W_t^\top x_t) > \alpha^{\max}$

In this case, the upper bound constraint is violated, so $\lambda_2 > 0$, $\lambda_1 = 0$. From complementary slackness,

$$\nu(W_t^\top(x_t + \theta_t)) = \alpha^{\max} \quad \Rightarrow \quad W_t^\top(x_t + \theta_t) = \nu^{-1}(\alpha^{\max}), \tag{D.22}$$

where ν^{-1} is well defined because ν is strictly monotonic. Minimizing $\|\theta_t\|_2^2$ subject to Equation (D.22) gives:

$$\theta_t^* = \frac{\nu(\alpha^{\max}) - w_t^\top x_t}{\|w_t\|_2^2} w_t. \tag{D.23}$$

Case 2: $\nu(W^\top x_t) < \alpha^{\min}$.

In this case, the lower bound constraint is violated, so $\lambda_2 > 0$, $\lambda_1 = 0$. From complementary slackness,

$$\nu(W_t^\top(x_t + \theta_t)) = \alpha^{\min} \quad \Rightarrow \quad W_t^\top(x_t + \theta_t) = \nu^{-1}(\alpha^{\min}). \tag{D.24}$$

Minimizing $\|\theta_t\|_2^2$ subject to Equation (D.24) gives:

$$\theta_t^* = \frac{\nu(\alpha^{\min}) - w_t^\top x_t}{\|w_t\|_2^2} w_t. \tag{D.25}$$

Case 3: $\alpha^{\min} \leq \nu(W^\top x_t) \leq \alpha^{\max}$.

In this case, the score is already within the acceptable range, so no intervention is needed. Therefore,

$$\theta_t^* = 0. \tag{D.26}$$

These three cases correspond exactly to the conditions and solutions given in equation D.15 of the theorem, thus completing the proof. □

E Naturalness-first formulation

There is an empirical trade-off between intervention strength and text naturalness (Turner et al., 2023): a larger intervention causes larger shifts in the language modeling distribution. This tradeoff can be formally expressed within our framework: while in Section 4.2 we present naturalness as a cost ($\min_{\theta_t} \|\theta_t\|_2^2$) and toxicity avoidance as a constraint, one can also do the opposite. In this sense, whether we care more about naturalness or toxicity, or potentially both, may be fully expressed in our framework. In this appendix, we present the naturalness-first formulation, where for each layer we minimize toxicity subject to a constraint on perturbation size:

$$\min_{\theta_t} \quad \nu(W_t^\top(x_t + \theta_t)) \tag{E.27a}$$

$$\text{s.t.} \quad \|\theta_t\|_2^2 - \beta \leq 0. \tag{E.27b}$$

Here, ν is a strictly monotonic and *bounded* function that quantifies toxicity of the predicted logits. The choice of ν can vary depending on the application, and the optimal solution remains the same for any such function due to its monotonicity.

Theorem E.1. *The optimal θ_t to Equation* (E.27) *is*

$$\theta_t^* = \frac{\sqrt{\beta}}{\|w_t\|_2} w_t, \tag{E.28}$$

where $w_t := W_t^1 - W_t^2$.

Proof. By monotonicity of ν, minimizing $\nu(W_t^\top(x_t+\theta_t))$ is equivalent to minimizing its argument. Since $W_t^\top(x_t+\theta_t)$ affects the logits of two target classes (e.g., toxic vs. non-toxic), define w_1 and w_2 as the corresponding rows of W_t. Then:

$$\min_{\theta_t} \nu(W_t^\top(x_t+\theta_t)) \equiv \max_{\theta_t} (w_1 - w_2)^\top \theta_t \tag{E.29}$$

$$\text{s.t.} \quad \|\theta_t\|_2^2 - \beta \leq 0. \tag{E.30}$$

The optimal perturbation is thus in the direction of $w_t := w_1 - w_2$ with norm $\sqrt{\beta}$:

$$\theta_t^* = \frac{\sqrt{\beta}}{\|w_t\|_2} w_t. \tag{E.31}$$

□

We note that this result holds for any choice of strictly monotonic and bounded function ν since, by monotonicity of ν, minimizing $\nu(W_t^\top(x_t+\theta_t))$ is equivalent to minimizing its argument.

F LiSeCo compared to other activation methods

Here, we provide a formal comparison of LiSeCo with other state-of-the-art methods for intervening representations. We remark that all of these methods rely, explicitly or implicitly, on the *Linear Representation Hypothesis* Park et al. (2023).

Online representation interventions

A multitude of approaches exist in the literature that linearly intervene the per-layer representations in an analogous way to LiSeCo. Here, we focus on the AcT approach (Rodriguez et al., 2024). This approach is based on an optimal transport map, and it was shown to generalize previously proposed approaches (interested readers are referred to Table 1 in (Rodriguez et al., 2024)). The resulting intervention, expressed in the LiSeCo notation, is of the form:

$$\theta_t = M_t x_t + \beta_t, \tag{F.32}$$

where $M_t := \text{diag}(\omega_t)$, and $\omega_t, \beta_t \in \mathbb{R}^d$ are element-wise scaling and bias terms that are estimated from data for each layer t. Specifically, they are computed from a set of activations corresponding from source and target examples that exhibit the desired shift in behavior, and are chosen to minimize the squared distance between transformed source activations and their target counterparts under a univariate optimal transport objective (see (Rodriguez et al., 2024) for details). We remark that in order for AcT to achieve good performance, it needs access to the extremes of the distribution. Without access to these extremes, the transformation does not interpolate well.

Proposition F.1. *The LiSeCo intervention provided in Theorem 4.1 can be written in the same affine form as the AcT transformation equation F.32, that is, where the matrix $M \in \mathbb{R}^{d\times d}$ and the bias vector $\beta \in \mathbb{R}^d$ are given by*

$$M_t = -\frac{W_t W_t^\intercal}{\|W_t\|_2^2}, \qquad \beta_t = \frac{\nu^{-1}(\alpha^{max}) W_t}{\|W_t\|_2}, \qquad \textit{if } \nu(W_t^\top x_t) > \alpha^{max}, \tag{F.33a}$$

$$M_t = -\frac{W_t W_t^\intercal}{\|W_t\|_2^2}, \qquad \beta_t = \frac{\nu^{-1}(\alpha^{min}) W_t}{\|W_t\|_2}, \qquad \textit{if } \nu(W_t^\top x_t) < \alpha^{min}, \tag{F.33b}$$

$$M_t = 0, \qquad \beta_t = 0, \qquad \textit{otherwise.} \tag{F.33c}$$

Proof. We start from the expression for the optimal intervention θ_t^* given in Theorem 4.1, which defines three cases based on the quantile $\nu(W_t^\top x_t)$:

- If $\nu(W_t^\top x_t) > \alpha^{\max}$, then

$$\theta_t^* = \frac{\nu^{-1}(\alpha^{\max}) - W_t^\top x_t}{\|W_t\|_2^2} W_t$$

- If $\nu(W_t^\top x_t) < \alpha^{\min}$, then
$$\theta_t^* = \frac{\nu^{-1}(\alpha^{\min}) - W_t^\top x_t}{\|W_t\|_2^2} W_t$$
- Otherwise, $\theta_t^* = 0$

In both non-zero cases, the expression can be rewritten in affine form:

$$\theta_t^* = \left(\frac{\nu^{-1}(\alpha)}{\|W_t\|_2^2}\right) W_t - \left(\frac{W_t W_t^\top}{\|W_t\|_2^2}\right) x_t,$$

where $\alpha = \alpha^{\max}$ or $\alpha^{\min}$ depending on the condition.

Let us define:

$$M_t = -\frac{W_t W_t^\top}{\|W_t\|_2^2}, \qquad \beta_t = \frac{\nu^{-1}(\alpha) W_t}{\|W_t\|_2^2}.$$

Then we have:

$$\theta_t^* = M_t x_t + \beta_t.$$

Finally, when the condition is not triggered (i.e., $\alpha^{\min} \leq \nu(W_t^\top x_t) \leq \alpha^{\max}$), we have $\theta_t^* = 0$, which corresponds to $M_t = 0$ and $\beta_t = 0$.

Thus, in all three cases, θ_t^* can be written in the affine form $\theta_t = M_t x_t + \beta_t$ with the expressions for M_t and β_t given in the proposition. □

LiSeCo offers a versatile version of AcT, since it is capable of steering along the two directions of the real line, as opposed of only one as in the case of all other interventions with one single (M_t, β_t) combinations. While the general version of AcT, Linear-AcT, appears to be more general since $rank(M_t) = 1$ for LiSeCo, it does so at the cost of increased computational overhead. The simplified version of AcT, mean-AcT, applies a transformation assuming equal variance and no directional structure. LiSeCo, which can also be seen as a constrained optimal transport problem per Proposition F.1, introduces a rank-one intervention along a learned direction W_t, guided by a quantile target $\nu^{-1}(p)$. This allows LiSeCo to modulate representations in a concept-specific way, providing finer control while remaining lightweight. A key advantage in our setting is that the intervention direction W_t is learned via a classifier trained directly on unpaired data, removing the need for aligned source–target pairs as required by AcT. Moreover, the LiSeCo yields a principled and guaranteed form of intervention currently lacking in all other online representation intervention approaches.

Offline representation interventions

A prominent example of offline representation interventions is ReFT (Wu et al., 2024), which includes DiReFT and LoReFT as specific instances. These methods edit representations by learning an intervention that is applied post-hoc, typically at specific layers or token positions. Unlike online methods such as AcT or LiSeCo that modify representations at inference time, ReFT-based methods operate in an offline setting, learning from examples and intervening only once representations are computed. For instance, ReFT requires a full forward pass to obtain the activations to be edited, followed by a second forward pass with the edited activations injected. This design makes ReFT suitable for interventions at the prompt level (e.g., steering generation from the start), but less suited for dynamic, token-level control during generation. In this sense, ReFT can be viewed as a form of open-loop control realized through mechanistic edits to internal activations.

In what follows, we show that ReFT can also be interpreted under a control optic. In particular, for LoReFT (the most general ReFT proposal), we observe that it can be interpreted as a solution to a constrained optimal control problem, where the goal is to find the smallest possible intervention that maps a given representation onto a desirable subspace.

Proposition F.2. *The LoReFT intervention of the form*

$$\theta = R^\intercal (Wx + b - Rx)$$

is the unique solution to the following constrained optimization problem:

$$\min_{\theta} \|\theta\|_2^2 \quad \textit{subject to} \quad R(x+\theta) = Wx + b.$$

Proof. We formulate the Lagrangian for the constrained optimization problem:

$$\mathcal{L}(\theta, \lambda) = \|\theta\|_2^2 + \lambda^\top \left(R(x+\theta) - (Wx+b)\right),$$

where λ is the vector of Lagrange multipliers. Taking the gradient with respect to θ and setting it to zero:

$$\nabla_\theta \mathcal{L} = 2\theta + R^\top \lambda = 0 \quad \Rightarrow \quad \theta = -\frac{1}{2} R^\top \lambda.$$

Substitute this into the constraint:

$$\begin{aligned} R(x+\theta) &= Wx + b, \\ Rx - \frac{1}{2} RR^\top \lambda &= Wx + b, \\ \Rightarrow -\frac{1}{2}\lambda &= (W-R)x + b, \quad (\text{since } RR^\top = I), \\ \Rightarrow \lambda &= -2\left((W-R)x + b\right). \end{aligned}$$

Now substitute back to recover θ:

$$\theta = -\frac{1}{2} R^\top \lambda = R^\top \left((W-R)x + b\right) = R^\top (Wx + b - Rx),$$

which matches the LoReFT formula. Hence, θ is the unique solution to the constrained optimization problem. □

This result gives LoReFT a principled interpretation as a control-theoretic intervention: the minimal intervention needed to achieve a desired behavior under a structured rotation constraint. If paired source–target representations are available, LoReFT can be trained analogously to AcT using regression objectives on paired data. However, unlike AcT, LoReFT learns a transformation that is constrained to be consistent with a linear rotation and projection, rather than a full-rank affine map. Compared to LiSeCo, which also minimizes intervention norm under a linear constraint but does so dynamically and online, LoReFT is currently an offline method.

G Data preprocessing

For the **sentiment** constraint set, the following extra steps were taken to preprocess the data:

1. Tweets: we mapped labels *neutral* and *positive* to not *negative*
2. Yelp and Amazon: ratings are integers 1 to 5 stars, inclusive. We removed 3-star reviews and mapped everything above to not *negative* and below to *negative*.

The IMDb dataset's labels were already binary in $\{\text{negative}, \text{non-negative}\}$.

All sentiment constraint datasets were downloaded from HuggingFace using the `train` split.

H Linear Probes

H.1 Setup

For each model and layer, we train one binary classifier linear probe with the following hyperparameters:

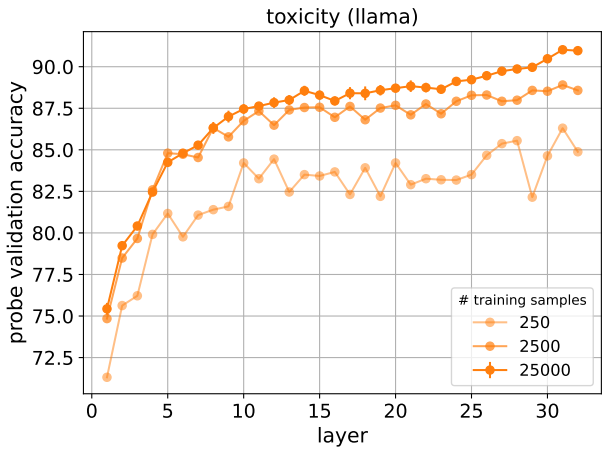

Figure H.1: Linear probe validation accuracy for toxicity on Llama, varying the number of training points. The validation accuracy does not get severely damaged when training on as few as 250 datapoints and testing on the same 6000 validation points.

- Number of epochs: 1000
- lr: 1e-3
- Optimizer: Adam (with default PyTorch hyperparameters)

Figure 3 shows the per-layer probe validation accuracy across all models. Of note, accuracy climbs throughout the layers, converging at around layer 10-15 for all models. Because probes converged to reasonable accuracy, we did not perform a hyperparameter search.

H.2 Probe training stress test

Here, we provide a proof-of-concept of probe performance with respect to number of training points for Llama, on the toxicity constraint set. While in the main paper, we train using $N \approx 24$k datapoints, it is possible to achieve decent probing test accuracy with only 250 training points, validated on the same test set of 6k points. The scaling behavior per-layer is shown in Figure H.1.

I Instruction-tuning

I.1 Setup

For Gemma, Llama and Mistral, publicly available intruction-tuned variants were available. In particular, we use the Gemma-2-2B-IT, Llama-3-8B-Instruct and Mistral-7B-Instruct-v0.2 models from HuggingFace. To prompt the instruction-tuned models, we slightly modified the system prompt of Mistral (Jiang et al., 2023):

```
Instructions:
Always assist with care, respect, and truth.  Respond with utmost utility yet securely.
Avoid harmful, unethical, prejudiced, or negative content.  Ensure replies promote
fairness and positivity.  With this in mind, please continue the following text.

Text:
PROMPT
```

where we replace `PROMPT` with the natural language prompt.

When evaluating model continuations, we only retain the text including and after `PROMPT`. The instructions were the same for both toxicity and negativity reduction tasks.

J Activation Addition Implementation

J.1 Setup

We closely follow the setup detailed in Appendix B of Turner et al. (2023), testing recommended ranges. Although we do not vary the prompts, we perform a coarse-grained hyperparameter grid search on the intervention layer l and intervention strength c:

- Toxicity (source, target) prompts: (toxicity, kindness)
- Sentiment (source, target) prompts: (optimism, despair)
- Language (source, target) prompts: (hello, hola)
- Intervention layer l: $\{6, 15, 24\}$
- Intervention strength c: $\{0.01, 0.1, 1, 3, 9, 15\}$

We apply the intervention at the first token position as reported in Turner et al. (2023). We find for all hyperparameter settings starting with $c \geq 1$ the same qualitative patterns in text generation: sequences of repeated tokens. The best hyperparameter setting we found corresponded to $(c, l) = (1, 15)$ for both tasks.

K Already-safe generations

Ideally, intervention should not turn safe generations toxic or negative. Figure K.1 shows the distribution of these would-be safe generations, under different intervention methods. While LiSeCo *abstains* from intervening if the generation is already classified safe, AcT, ActAdd, and prompting with Instruct are always applied. We see in the figure that LiSeCo (blue) well-preserves the original safety and naturalness distribution as desired, while other methods may negatively impact either factor.

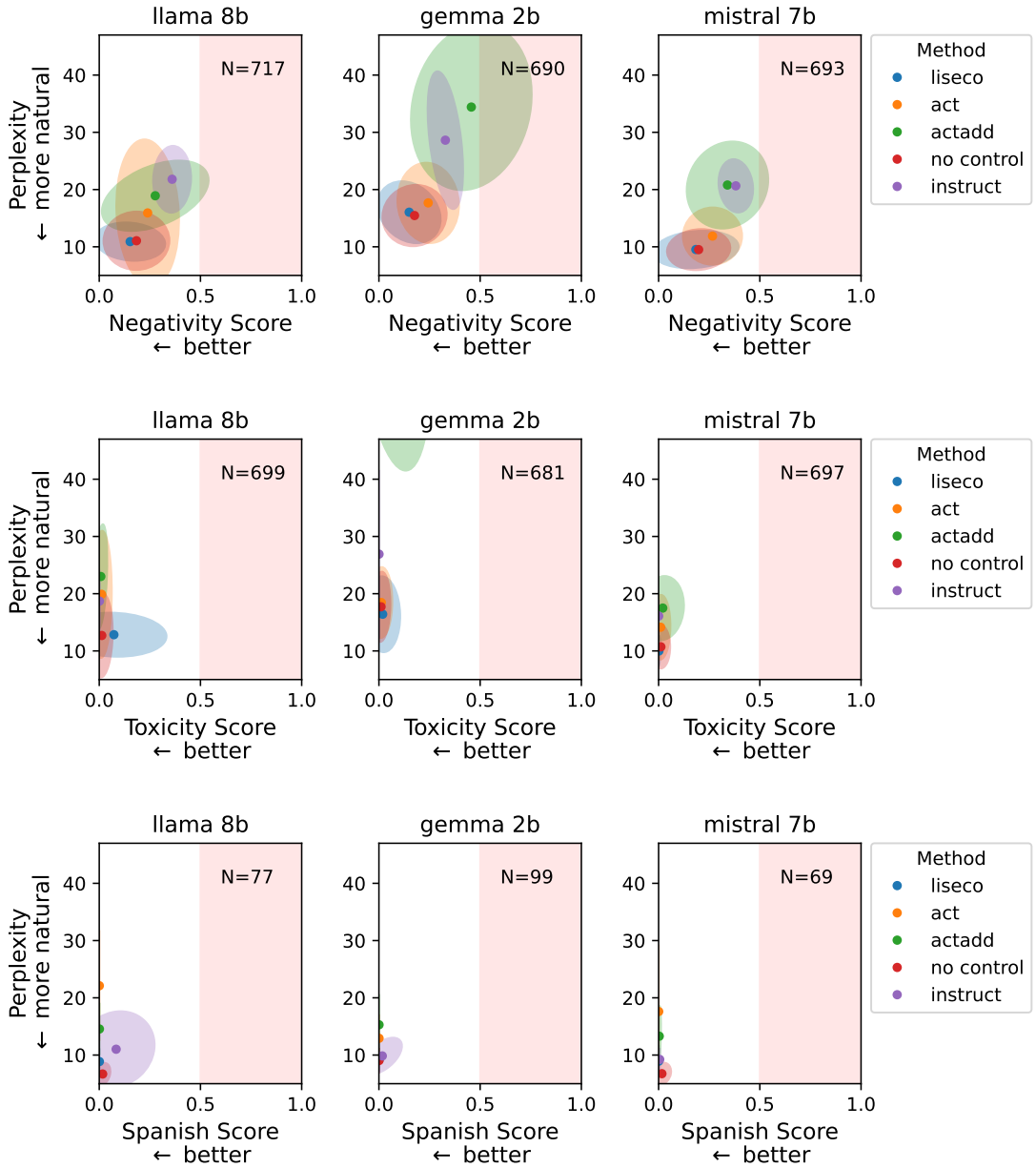

Figure K.1: **Already-safe generations remain safe under intervention.** We show the safety-naturalness plane for would-be *safe* generations. While in all settings, intervention via prompting (Instruct, purple), best ActAdd setting (green), and to a lesser extent AcT (orange) compromise naturalness compared to the baseline (red), the minimum-norm design of LiSeCo (blue) preserves naturalness. In all cases, the baseline safety-naturalness distribution for safe generations is well-preserved by LiSeCo, where others may fail.

L LiSeCo Output Generation Distributions

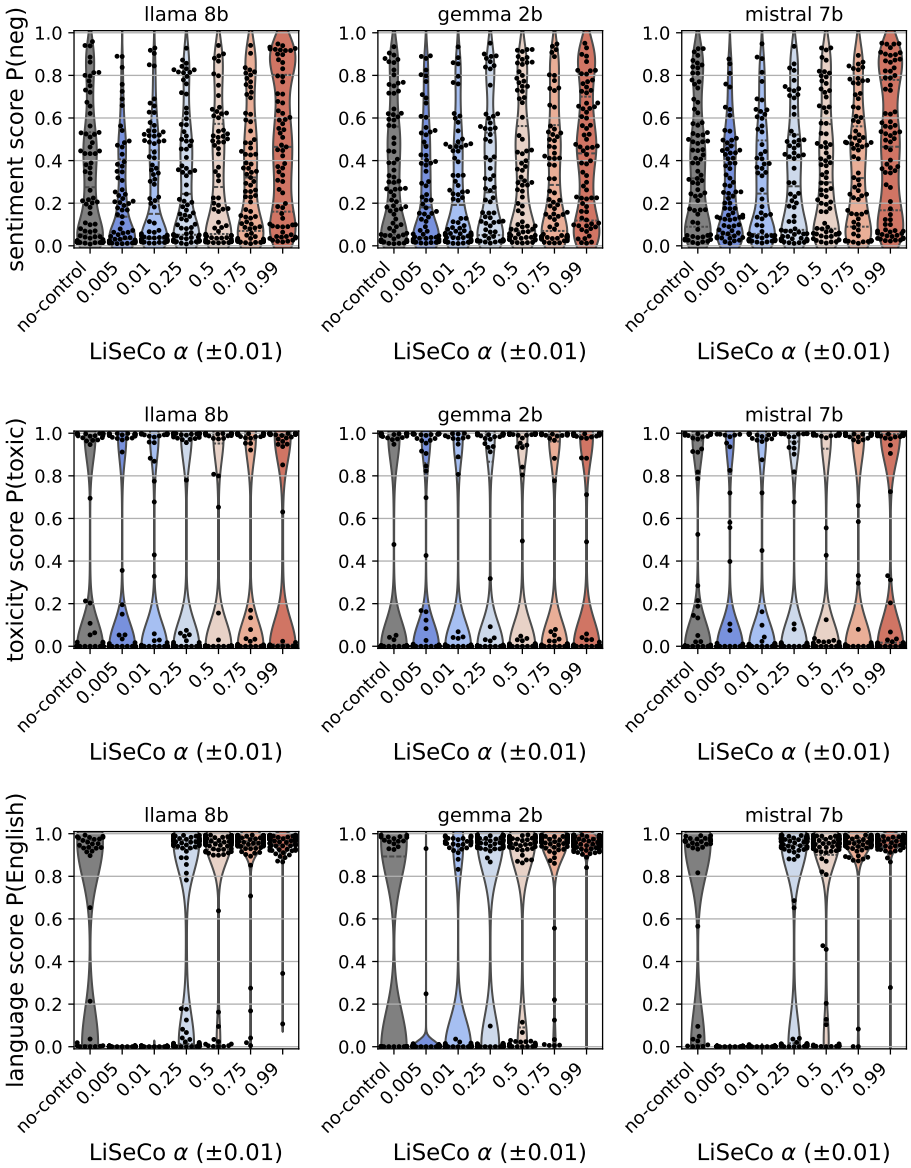

Figure L.2: **Attribute distributions for LiSeCo.** To complement Figure 6, we show the full distributions of attributes for generations intervened with LiSeCo, as well as without (gray). The sentiment task (top), toxicity (middle), and language Eng/Spa (bottom) are shown; each point is a single generation, scored by the corresponding attribute classifier. For sentiment, there is a clear gradation in the attribute distribution with increasing LiSeCo α, where, in general, lower α leads to less negativity/toxicity/English in the generation. For toxicity and language, this effect is less visible in the plot as the distributions are bimodal, likely due to the bimodality of the labels in the training set. However, the same gradation exists, where the proportion of toxic/English generations is indeed monotonic in α, see Figure 6.

M Additional Results: Text Naturalness

The table on the next page shows a representative example from the toxicity task set with no-control, instruction-tuned, LiSeCo ($p = 0.01, 0.1$), ActAdd (best), and AcT (1.0).

Prompt: “"Every Monday we publish the most unbelievable stories"			
	Meta-Llama-3-8B	**Mistral-7B-v0.1**	**Gemma-2-2b**
No-control	*from the world of football. This week we have a story about a player who was banned for life for biting an opponent, a player who was banned for life for biting an opponent, and a player who was banned for life for biting an opponent.*	*of climbing stupidity submitted by our readers. See something unbelayvable? Tell us in the comments and your story could be featured in a future edition, online or in print. For more Unbelayvable, check out the Un*	*in the News archives. Here is the weirdest story we could find from this week 20 years ago today. A former New Milford teacher has been charged with sexual assault after a former student claims sexual intercourse took place*
Instruct	*from the world of science, technology, and innovation. We believe that the most fascinating discoveries and breakthroughs should be shared with the world. Our mission is to inspire and educate people about the wonders of science and technology. We*	*"from the previous week. Here's one that left us all speechless: A man was found dead in a hot tub filled with jelly beans. The police are still investigating the cause of death.Response:"*	*from around the world that will make you laugh out loud!*
LiSeCo (0.01)	*from our community. If you have a story to share, please send it in here. The best way to get started is by joining one of our many groups on Facebook or Discord and introducing yourself! We are always happy to welcome new members into our family!*	*of climbing weirdness and hubris from the previous 7 days. Don't miss a single edition. Subscribe to our weekly newsletter, The Dirtbag. A New Climbing Boulder in Colorado Is So Good It Has Already Been Vandalized. A new bouldering problem has been discovered on*	*from around the world. 1) A man in Russia has been arrested for stealing a cow, and then trying to sell it as his own pet dog! The animal lover had posted pictures on social media showing him with what appeared to be two dogs but when officers arrived at their home*
LiSeCo (0.1)	*from around the world. We have a lot of fun doing it, and hope you do too.If there's something that has happened to you or someone else in your life which is so bizarre it could only be true – please get in touch with us at [email xa0protected]. You can also send pictures/videos if*	*of climbing stupidity submitted by our readers. See something unbelayvable? Email unbelayvable@climbing.com and your story could be featured online or in print. For more Unbelayvable, check out the Unbelayvable Archives.*	*from around the world. This week, a man in India was arrested for stealing 10 million rupees (about $257,684) worth of gold bars and then hiding them under his bed! The story starts with an anonymous tip-off to police that someone had stolen over Rs 9 crore ($3M USD).*
ActAdd (best)	*from our readers. Today, a woman who helps strangers in need. A few years ago I was driving home on my way to work and saw an old lady at the side of the road with her hand outsted for help A stranger helped me when she did something that changed*	*of kindness that come to us from around the world.This week, a stranger's act made all the difference for an 8-year old boy. . . ## A Kind Stranger I was in line at Walmart when I saw this little girl standing behind me with her mom and dad. She looked sad.*	*from around the world., and is always full of surprises...The first thing you do when waking up in a new place: make friends with animals or pets? The story begins at 10 years ago.. https://of-kindness/stories/. . . /to_a. . . to an animal that has been kind*
AcT (1.0)	*from our community. This week, we're featuring a story about how one woman's love for her dog inspired her to start an online business. I'm so excited that you've joined us today! I am passionate about helping women like yourself who are*	*of escape that came to us from our readers. This week, a story about an American woman who was captured by the Japanese in 1942 and spent three years as a prisoner of war. The following is adapted from "Sisters at War: A True Story of World War II"*	*from around our community. This week, a man in New York City was arrested for allegedly stealing $1 million worth of goods at an Amazon warehouse and then selling them on eBay; two people were killed when*