Linearly Controlled Language Generation with Performative Guarantees

Emily Cheng
Univ. Pompeu Fabra
Barcelona, Spain
emilyshana.cheng@upf.edu

Marco Baroni
ICREA & Univ. Pompeu Fabra
Barcelona, Spain
marco.baroni@upf.edu

Carmen Amo Alonso
ETH Zürich
Zürich, Switzerland
camoalonso@ethz.ch

Abstract

With increased use of Large Language Models (LMs) comes a need for controlled text generation strategies with performance guarantees. To achieve this, we use a common model of concept semantics as linearly represented in an LM's latent space. We take the view that each natural language token generation traces a trajectory in this continuous space, realized by the LM's hidden layer activations. This view permits a control-theoretic treatment of text generation in latent space, where we propose a lightweight, gradient-free intervention that is *guaranteed* (in-probability) to steer trajectories away from regions corresponding to undesired meanings. We demonstrate on toxicity and negativity use cases that the intervention steers language away from undesired content while maintaining text quality.

1 Introduction

Large Language Models (LMs) have become widespread in critical applications such as content moderation and real-time information dissemination [27]. Despite their transformative impact, these models require updates to remain accurate post-deployment, as well as strategies to enforce constraints during text generation. As such, controllable text generation has emerged as a pivotal research area.

Of the proposed approaches, prompt engineering [20, 4, 6], consists of carefully choosing natural-language prompts at input-time to steer generation. Others modify LM parameters to achieve the desired outputs [26, 18]. Lastly, most relevant to this work, some approaches directly steer LM *activations* to the desired effect [8, 11, 16, 17]. Despite current efforts, ensuring model controllability remains a challenge due to models' limited interpretability. Moreover, existing methods do not provide controllability and risk guarantees, which are crucial to harness their full potential safely.

We propose to use control theory to address this gap. Optimal control theory [15] offers principled methods to steer trajectories in latent space with theoretical guarantees on the performance of the intervention. In the optimal control framework, 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 as a constrained optimization problem and provide its closed-form solution with guarantees; (2) we empirically demonstrate our method on a toxicity avoidance use case. We confirm, with experiment corroborating theory, that LiSeCo steers LM generation away from undesired content while maintaining text quality.

2 Optimal Language Generation Controller in Latent Space

We frame controlled language generation as a standard optimal control problem [15]. Specifically, we study how to steer a pre-trained LM's output away from a region corresponding to *disallowed semantics*, designing an intervention so the set of possible generated sequences is constrained to

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

an allowed subset. Two requirements are imposed on the generated text: its latent trajectory (1) is *guaranteed* to never lie in the disallowed region, and (2) stays maximally close to that of the original output so text quality is not compromised. How can the disallowed region be defined for a given LM? And, how can an intervention be designed to guarantee (in probability) that the latent trajectory lies in the allowed region while maximally similar to the original model? In what follows, we answer these questions and show the proposed approach adds minimal computational overhead to text generation.

2.1 Approach

We consider language generation to realize a trajectory through the model's layers in activation space. Similar to [23], we take the view that, for every layer t, disallowed language occupies a region $\mathcal{R}_t$ of latent space $\mathbb{R}^d$, where d is the hidden dimension of the LM. Our goal is to provide, by altering the hidden embedding at every layer, a control mechanism that guarantees in-probability that latent trajectories remain out of $\mathcal{R}_t$ for all layers t, for all tokens generated.

Semantic Probe We first identify the disallowed region for the generated token given context. To do so, we feed a set of sequences to the model, and use a lightweight probe to map each latent state $x_t \in \mathbb{R}^d$ to a probability that the sequence is toxic. Specifically, we rely on a *probing classifier function* f_t that maps the latent space $\mathbb{R}^d$ to the decision space $[0, 1]$. For simplicity, we take f_t to be a logistic regression classifier realized by a linear probe [12]. Formally, $f_t : \mathbb{R}^d \to [0, 1]$; $x \mapsto \sigma_2(W_t^\top x)$, where $W_t \in \mathbb{R}^{d\times 2}$ and σ is the softmax. For each layer, we define the disallowed region $\mathcal{R}_t$ to be the pre-image of a *toxic* classification under f_t, using a predefined probability threshold p. That is, $\mathcal{R}_t := \{x|\ \sigma_2(W_t^\top x) \geq p\}$, where $p \in [0, 1]$.

Optimal Control Once the forbidden region $\mathcal{R}_t$ is identified, we design a control strategy that, for all layers t, guarantees the latent state x_t remains in the allowed region and retains maximal similarity with the original model. To do this, we design an optimal controller that generates an input $\theta_t \in \mathbb{R}^d$ at every layer t. Mathematically, we solve an optimization problem over θ_t where the pre-computed classifier enters as a hard constraint in the formulation, i.e., $\sigma_2(W_t^\top(\theta_t + x_t)) \leq p$. This ensures the controlled latent trajectory $\tilde{x}_t = x_t + \theta_t \in \mathbb{R}^d$ lies in the toxic region with probability less than p.

2.2 Derivation of the optimal controller

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.

Optimal Controller Setup The optimal controller aims to keep latent trajectories out of the unsafe region without compromising text 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:

$$\min_{\theta_1,\ldots,\theta_T} \quad \sum_{t=1}^{T} \|\theta_t\|_2^2 \tag{1a}$$

$$s.t. \quad \sigma_2(W_t^\top(x_t + \theta_t)) - p \leq 0, \quad \forall t = 1, \ldots, T \tag{1b}$$

$$x_{t+1} = \texttt{layer}_t(x_t + \theta_t), \tag{1c}$$

$$x_0 = E(\texttt{prompt sequence}), \tag{1d}$$

where E is the embedding map. Optimization problem (1) aims to find the minimum l_2-norm intervention $\theta_1, \ldots, \theta_T$ (Eq. (1a)) that satisfies the following constraints: Eq. (1b) requires the modified latent state $x_t + \theta_t$ be classified non-toxic by the probe f_t; Eq. (1c) captures LM dynamics, i.e., layer t maps the modified latent state $x_t + \theta_t$ to the next latent state x_{t+1}; Eq. (1d) states that the LM's input embeds the input context, so that interventions are *context-dependent*. The intervention that solves optimization problem (1) is *guaranteed by construction* to keep the latent trajectory $\tilde{x}_1, \ldots, \tilde{x}_T$ and output y below the probability threshold from the classifier.

Optimal Controller Design By Bellman's Optimality Principle, the standard approach to solving problem (1) is dynamic programming (DP) [15]: the optimal solution is computed for the last layer T,

then via backward induction for $T-1,\ldots,1$. But, layer dynamics (1c) are highly non-convex, and solutions incomputable in closed form, hence their optimality is not guaranteed. Further, DP requires backpropagating gradients at each text generation's forward pass, adding significant inference latency.

To overcome these limitations, we relax problem (1). No longer searching for a globally optimal solution across layers, we now search for locally optimal solutions at each layer. Now, Eqs. (1c) and (1d) cease to play a role, as each layer is optimized for separately. Then, problem (1) is relaxed into

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

$$s.t. \quad \sigma_2(W_t^\top(x_t+\theta_t)) - p \leq 0, \tag{2b}$$

for each layer $t = 1 \cdots T$. The sequence of θ_t that solve problem (2) may not optimize the original formulation (1). But, optimality is not essential as the cost aims only to preserve similarity with the original model. Meanwhile, the guarantee to avoid unsafe region $\mathcal{R}$ is still enforced via Eq. (2b).

A key advantage of relaxed formulation (2) 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 1 (Optimal θ)**.** *The optimal solution $\theta_t^* \in \mathbb{R}^d$ to the optimization problem* (2) *is given by*

$$\theta_t^* = \begin{cases} \dfrac{\log\left(\frac{1}{p}-1\right) - w_t^\top x_t}{\|w_t\|_2^2} w_t & \text{if } \sigma_2(W_t^\top x_t) > p \qquad (3a) \\ 0 & \text{otherwise,} \qquad (3b) \end{cases}$$

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

Geometrically, the optimal solution is the vector from x_t to the closest point in $\mathcal{R}^C$. When $x_t \notin \mathcal{R}$, i.e., when $p(\text{unsafe}) < p$, no update is needed; hence $\theta_t^* = 0$. Otherwise, the update is a factor of w_t. Since θ_t^* exists in closed-form, computing an intervention incurs negligible computational overhead. Crucially, it is guaranteed with probability p to keep the latent state outside the disallowed region.

3 Experiments

The LiSeCo pipeline is as follows. First, to find the unsafe regions and probes per layer, there is an initial probe training phase. Then, probes are integrated into the model at inference-time, and the optimal intervention dynamically applied. We tested LiSeCo on toxicity and negativity avoidance for three causal LMs, Llama-3-8B [22], Pythia-6.9B [5], and Mistral-8B [14].

Setup We learn probing classifiers f using a human binary-labelled *constraint dataset*. Splitting into 80/20% train and test sets, for each layer we train logistic regression classifier f_t using cross-entropy loss. We use the last token embedding to represent the entire sequence, as it is the only to attend to the entire input context. We use Kaggle's binary-labelled toxicity dataset [1] and a combination of sentiment datasets for the toxicity and negativity avoidance tasks, respectively. See Appendix D for preprocessing details and Appendix E for implementation details.

Text generation is evaluated on a *task dataset* of 300 prompts, sampled respectively from RealToxicityPrompts [9] and its sentiment counterpart [19] for the toxicity and negativity avoidance tasks . We insert trained probes f_t to sequentially evaluate toxicity for each layer t during each forward pass. If layer t's representation x_t is evaluated toxic, then the control vector θ_t is applied. For simplicity, we fix text generation to at most 50 new tokens, greedily decoded. For prompt details, see Appendix D.

Finally, we compare against several baselines: no-control, instruction-tuning where applicable (Llama and Mistral) (Appendix F), FUDGE [25], and Activation Addition (ActAdd) [24] (Appendix G). We automatically rated toxicity and negativity of generated text using RoBERTa-based classifiers trained on Twitter data, which returns $p(\text{toxic})$ in $[0, 1]$ [7, 3]. In addition, we rated text naturalness on a Likert scale from 1 to 5 in a blind setup. For annotation instructions, see Appendix I.

Results Toxicity and negativity are linearly represented in latent space. The table shows, for all models, probe validation accuracies averaged across layers and 5 random seeds. Accuracies reach $\sim$90%, confirming unsafe regions are linearly decodable with high probability (see Figure E.1 for per-layer results).

probe val. acc(%)	**Pythia**	**Llama**	**Mistral**	**Llama-Instr**	**Mistral-Instr**
toxicity	89.6 ± 1.17	87.2 ± 3.46	86.0 ± 5.12	88.1 ± 3.82	86.4 ± 5.01
negativity	87.5 ± 3.91	87.3 ± 5.87	85.9 ± 7.69	87.6 ± 6.08	86.2 ± 7.32

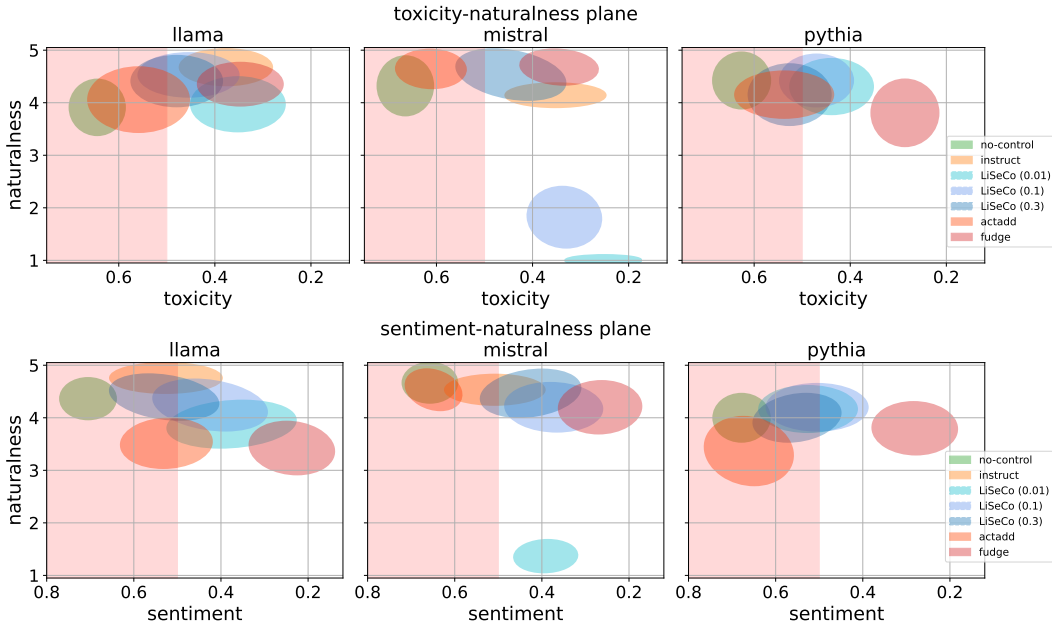

Figure 1: **The toxicity-naturalness plane** (top) and **sentiment-naturalness plane** (bottom) for Llama, Mistral, and Pythia (left to right). The **top-right corner (low toxicity, high naturalness) is best**. Each method's (toxicity/negativity, naturalness) distribution over *would-be toxic* continuations is shown as an ellipse centered at the mean, whose axes reflect ± 1 SD. The red region is that labelled toxic/negative by the external classifier. LiSeCo (blue colors) shifts right, i.e., reduces toxicity/negativity, from no-control (green) and maintains high naturalness, performing on-par with instruction tuning (light orange). ActAdd (orange) least reduces toxicity/negativity. FUDGE (red), which directly optimizes w.r.t. the external classifier, most reduces toxicity/negativity as expected.

We plot the performance of methods in Figure 1 on the toxicity-naturalness and sentiment-naturalness planes for *would-be toxic continuations* (where no-control produced unsafe content). LiSeCo predictably reduces toxicity as p decreases (analysis in Appendix H), while maintaining text naturalness, on-par with instruction tuning without extensive finetuning.[1] Notably, LiSeCo's naturalness correlates to p by construction (Theorem 1); this is visible in Mistral (Figure 1 center). FUDGE, which directly optimizes for the external classifier, is expectedly the "best case" baseline, maintaining naturalness and most reducing toxicity and negativity. In contrast, ActAdd least reduces toxicity and negativity.

4 Discussion

We have proposed LiSeCo, a controlled language generation method that is theoretically guaranteed to stay within safe regions of latent space, and empirically validated for toxicity and negativity avoidance. The method is compatible with all current Transformer-based architectures, and only involves a small inference-time latency. Future work will scale LiSeCo to joint constraints and alternatives to linear probes to ascertain the unsafe region.

[1]Human ratings did not correlate to perplexity, a commonly-used metric in NLP to evaluate text naturalness. This was due to low-perplexity, degenerate outputs (Appendix H.1), so we do not attempt an in-depth analysis of the latter. We leave automated text evaluation to future work.

References

[1] 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.

[2] 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].

[3] 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.

[4] 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.

[5] Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. (arXiv:2304.01373), Apr 2023. URL http://arxiv.org/abs/2304.01373. arXiv:2304.01373 [cs].

[6] 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.

[7] 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*, pages 38–49, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-demos.5.

[8] 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.

[9] Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.301. URL https://aclanthology.org/2020.findings-emnlp.301.

[10] Aaron Gokaslan and Vanya Cohen. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

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

[12] John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1419. URL https://aclanthology.org/N19-1419.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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, 2024.

[18] 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*, 2023.

[19] 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 Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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)*, pages 6691–6706, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.522. URL https://aclanthology.org/2021.acl-long.522.

[20] 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.

[21] 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*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

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

[23] 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.

[24] 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.

[25] 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, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 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.

[26] 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*, pages 10222–10240, 2023.

[27] 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.

[28] 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, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/250cf8b51c773f3f8dc8b4be867a9a02-Paper.pdf.

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 15 minutes per layer, so overall 32 wall-clock hours. Running evaluation experiments took a total of 10 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 8 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

Pythia `https://huggingface.co/EleutherAI/pythia-6.9b`; license: apache-2.0

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.

Toxicity task `https://huggingface.co/datasets/allenai/real-toxicity-prompts`; license: apache-2.0

Sentiment task `https://github.com/alisawuffles/DExperts`; license: unknown

C Proof of Theorem 1

We solve eq. (2) using KKT conditions:

1. Stationarity.
$$0 \in \partial(\|\theta_t\|_2^2 + \lambda(\sigma_2(W^T(x_t + \theta_t)) - p)) \tag{C.4}$$
2. Complementary slackness.
$$\lambda(\sigma_2(W^\top(x_t + \theta_t) - p) = 0 \tag{C.5}$$
3. Primal feasibility:
$$\sigma_2(W^\top(x_t + \theta_t)) - p \leq 0 \tag{C.6}$$
4. Dual feasibility.
$$\lambda \geq 0. \tag{C.7}$$

First, consider when $\lambda = 0$. We apply the stationarity condition (C.4) to obtain $\theta_t = \mathbf{0}$. Plugging θ_t back into the primal constraint, we have that $\sigma_2(W^\top x_t) \leq p$ and recover the second line of (3a). That is, when $\lambda = 0$, we are already in the non-toxic region and do not need to apply an intervention θ_t.

Now, consider $\lambda > 0$. When this is the case, $\sigma_2(W^\top x_t) > p$ and an intervention is needed. Here, it is possible again to solve for θ_t in closed form. By complementary slackness (C.5),

$$p = \sigma_2(W^\top(x_t + \theta_t)) = \frac{\exp(w_2^\top(x_t + \theta_t))}{\exp(w_1^\top(x_t + \theta_t)) + \exp(w_2^\top(x_t + \theta_t))}. \tag{C.8}$$

Hence,

$$w^\top \theta_t + w^\top x_t - \log\left(\frac{1}{p} - 1\right) = 0. \tag{C.9}$$

Now, when $\lambda > 0$, or when $\sigma_2(W^\top x_t) > p$, (2) is equivalent to minimizing $\|\theta_t\|_2^2$ subject to (C.9). The Lagrangian with respect to this new formulation is

$$L(\theta_t, \lambda') = \|\theta_t\|_2^2 + \lambda' \left(w^\top \theta_t + w^\top x_t - \log\left(\frac{1}{p} - 1\right)\right). \tag{C.10}$$

Taking the partial derivative with respect to θ_t, we have

$$0 = \frac{\partial L(\theta_t, \lambda')}{\partial \theta_t} = 2\theta_t + \lambda' w. \tag{C.11}$$

Hence,

$$\theta_t = -\frac{\lambda' w}{2}. \tag{C.12}$$

Now, we plug θ_t back into (C.9) to obtain

$$\lambda' = \frac{2\left(w^\top x - \log\left(\frac{1}{p} - 1\right)\right)}{\|w\|_2^2}. \tag{C.13}$$

Finally, plugging λ' back into (C.12), we have

$$\theta_t = \frac{\log\left(\frac{1}{p} - 1\right) - w_t^\top x_t}{\|w_t\|_2^2} w_t. \tag{C.14}$$

This completes line 1 of (3a).

D Data details and preprocessing

D.1 Toxicity data

We first learn probing classifiers f using a labelled *constraint dataset*, then, we evaluate text generation on a *task dataset* [2]. For the constraint dataset, we use Kaggle's human-labeled toxicity dataset [1]. The dataset contains 30k label-balanced natural language comments and human-annotated binary toxicity labels. For the task dataset, we use RealToxicityPrompts [9], which contains a collection of prompts derived from Internet text, their continuations, and toxicity scores in $[0, 1]$ for both [10]. To form our task dataset, we sample 150 prompts for which there is a toxic continuation and 150 for which there is a non-toxic continuation in the original dataset.

D.2 Sentiment data

For the **sentiment task**, because sentiment datasets tend to be highly domain-specific (for instance, movie reviews), we combine several diverse datasets to form our constraint dataset of 30k datapoints. This consists of +/- label-balanced samples of 7500 datapoints each from IMDb film reviews [21], Tweets [3], Yelp reviews [28], and Amazon reviews [13]. For more preprocessing details, see appendix D. For the task dataset, we sample 300 neutral sentiment prompts from [19], created from OpenWebText as a sentiment counterpart to RealToxicityPrompts. Of these prompts, 150 have negative and 150 neutral or positive continuations, respectively.

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.

E Linear Probes

E.1 Setup

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

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

Figure E.1 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.

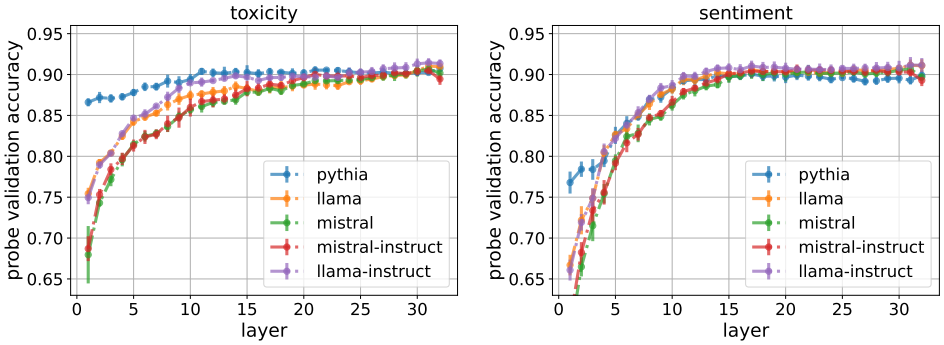

Figure E.1: Linear probe validation accuracy for toxicity (left) and sentiment (right) detection. All curves are shown with $\pm$ one standard deviation across 5 random seeds. The tasks converge to reasonable accuracies of $> 60\%$ for all models and layers, with mid-layers attaining around 90%.

F Instruction-tuning

Instruction-tuning, which relies on extensive LM finetuning, is the gold-standard baseline. For models with instruction-tuned variants (Llama and Mistral), we repeat the experimental procedure, training probes on the constraint dataset. Then, during evaluation, we prompt the instruction-tuned model using a template whose instructions are slightly modified from Mistral's system prompt provided in [14].

F.1 Setup

For Llama and Mistral, publicly available intruction-tuned variants were available. In particular, we use the 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 [14]:

```
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`. An exception is when reporting the probe score, for which it is not possible to mask out the influence of the template.

G Activation Addition Implementation

Like our method, ActAdd steers text generation in activation space [24]. 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. 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 G.

G.1 Setup

We closely follow the setup detailed in Appendix B of [24], 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)
- Intervention layer l: $\{6, 15, 24\}$
- Intervention strength c: $\{0.1, 1, 3, 9, 15\}$

As the text generation is often longer than the source and target prompts, we apply the intervention at the first token position, as reported in [24]. The ActAdd forward generation process is completely deterministic.

We find for all hyperparameter settings in $c \geq 3$, the same qualitative patterns in text generation: sequences of repeated tokens. When $c < 3$, we found text generation to remain natural but for there to be minimal effect on toxicity and negativity reduction.

H Additional Results

H.1 Toxicity

Semantic control Figure H.1 (upper) shows the probe score distribution of would-be toxic continuations ($S > 0.5$), $N = 25$, 37, 37 for Llama, Mistral, and Pythia, respectively. Then, the toxicity probe score reduction brought on by interventions is visible in the plots as a leftward shift. Notably, our intervention with constraint p works as expected, restricting probe scores to be less than p. While ActAdd also decreases the toxicity likelihood, the extent of reduction is sensitive to the hyperparameter setting and model, as shown by the different ordering of scatterplots in Figure H.3. For both Instruct models, the toxicity probe score decreases from the no-control baseline, though not as much as other interventions. Taken together, toxicity probe results show how theoretical guarantees aid interpretability: while toxicity reduction in instruction-tuning and ActAdd remain opaque, that of LiSeCo interpetably depends on toxicity threshold p.

Figure H.1 (bottom) shows the distribution of external toxicity scores for would-be toxic continuations. In particular, all baselines decrease toxicity, although we have seen the ActAdd baseline to compromise text naturalness. Of-note, when LiSeCo is used with a threshold of $p = 0.01$, it performs on-par with instruction-tuning for Llama.

Smaller LiSeCo p, fewer toxic generations We have demonstrated that our intervention reduces the likelihood of toxicity as defined by the linear probes. But, how well does this definition correspond to the true labels? Figure H.1 (bottom) shows that our method predictably decreases the toxicity likelihood as scored by the external classifier: as LiSeCo p decreases (row 5→3 of the plots), so does the percentage of toxic-labeled generations (right-hand side). Note, however, that, besides Mistral, the value of p does not upper-bound the percentage of toxic generations as it theoretically should: this indicates that, in practice, our probes only approximately learned the semantics of toxicity and do not perfectly generalize outside of training data. For Pythia in particular, the probe score results are incongruous with the external toxic label percentage.

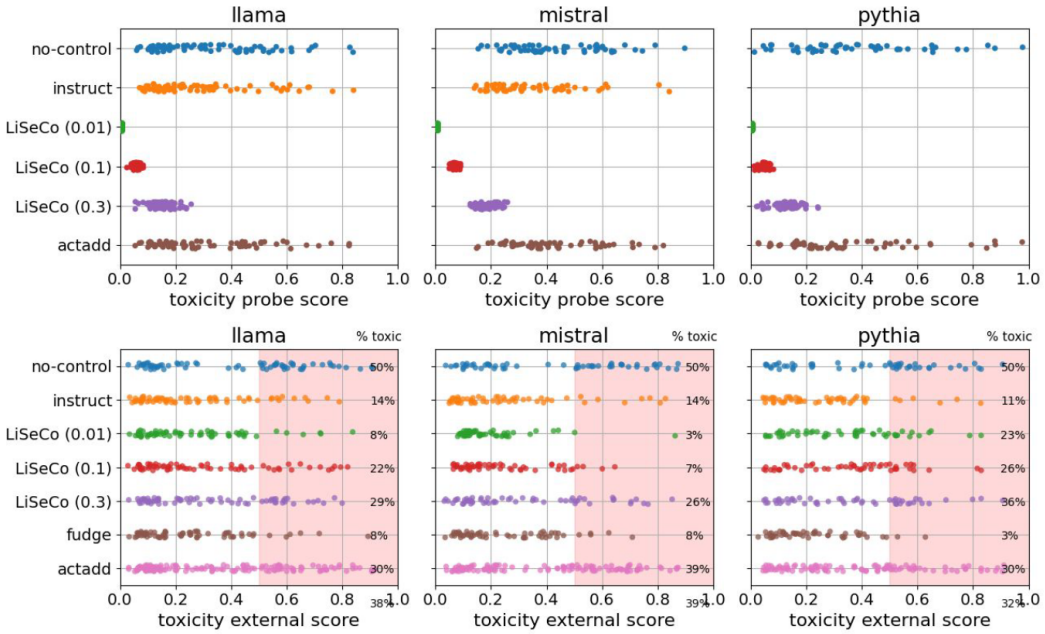

Figure H.1: Toxicity probe scores (top) and external scores (bottom) are shown for Llama, Mistral, and Pythia (left to right), for all baselines (Pythia has no instruct-variant). (Bottom) Probability for toxicity greater than 0.5 is shaded in red, with the toxic-labeled % displayed on the right.

To test our hypothesis that probe-to-external classifier alignment determines success in practice, we computed the Spearman correlations between the probe scores and the external scores for each model, across the no-control and LiSeCo runs. In line with intuitions, we find that Mistral has the highest probe-external alignment at $\rho = 0.38^{***}$, followed by Llama at $\rho = 0.20^{***}$, and finally Pythia at $\rho = 0.06$ (not significant).[2]

Perplexity Figure H.2 shows perplexity distributions for a set of tested methods. We find that perplexity does not correlate with human ratings, where the correlation is taken across a $N = 500$ sample from all continuations. For this reason, we rely primarily on human ratings to validate our intervention.

This low correlation results from ActAdd continuations. ActAdd affected outputs in ways that were not obviously negative from its low perplexity (see Figure H.2): we find, however, that ActAdd's low perplexity was attributed to degenerate outputs of repeated tokens.

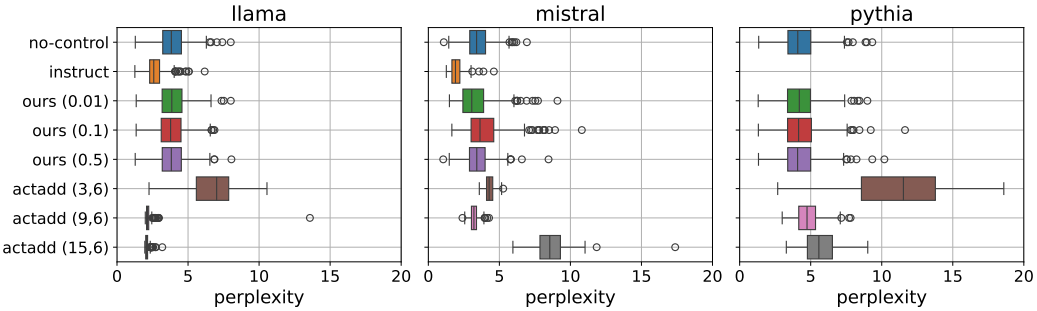

Figure H.2: Generated text perplexity for Llama, Mistral, and Pythia (left to right). Right outliers for ActAdd are not shown.

[2] $(^{***})$ means significant at $\alpha = 1e-3$

H.2 Sentiment

Here, we reproduce the main toxicity mitigation results on text sentiment, specifically negativity reduction.

Semantic control The sentiment score distributions for would-be toxic continuations are shown in Figure H.3, respectively. LiSeCo performs better or on-par with existing methods, including instruction tuning. Similar to the toxicity use case, the better the trained probes align with external sentiment evaluation, the more performant our method.

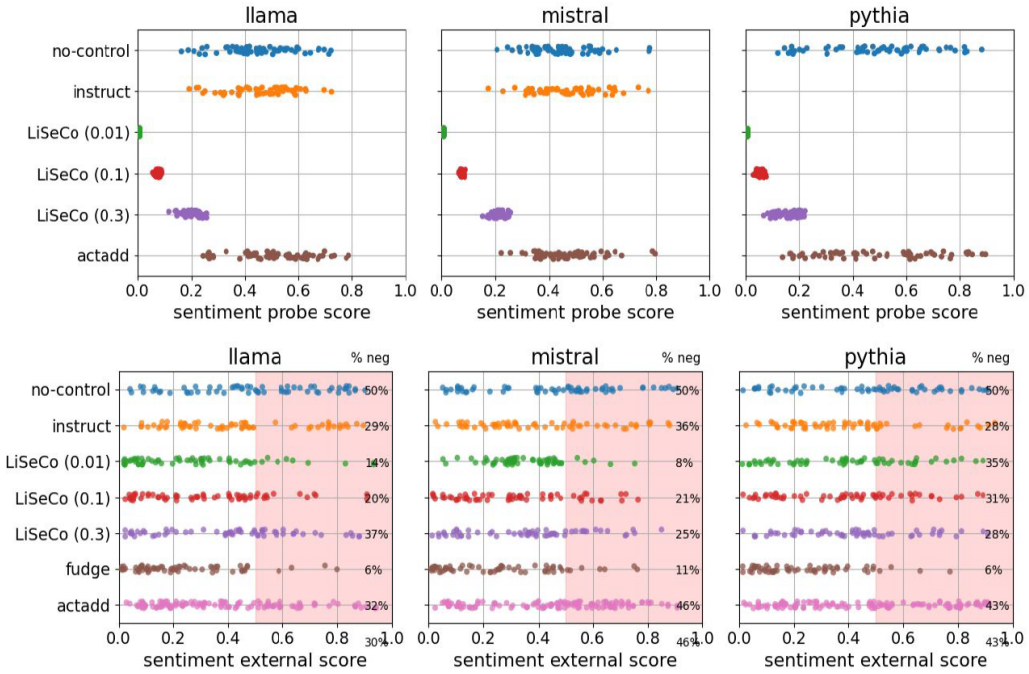

Figure H.3: Top: Would-be toxic continuations distribution of probe scores for sentiment task set ($N = 21, 15, 30$ for Llama, Mistral, and Pythia, respectively), shown for all baselines and the best external score layer for ActAdd, layer 15. Bottom: Probe score distributions for sentiment. Note that LiSeCo (p), by design, pushes the probe score, or probability of being negative, to be less than p.

Smaller LiSeCo p, fewer negative generations Figure H.3 shows the probe and external scores for Llama, Mistral, and Pythia for the would-be toxic continuations ($N = 21$, 15, and 30), respectively.

First, looking at the rows in Figure H.3 top corresponding to LiSeCo, we see that LiSeCo works as expected, where decreasing p thresholds the sentiment probe score to $< p$.

Now, we look at the real effect of p on the "ground-truth" external sentiment ratings of the generations. The intermediate rows in Figure H.3 bottom show that, as we decrease LiSeCo p, the number of negative generations, as given by the external score, decreases for all models.

Better probes, better performance For the sentiment task, LiSeCo performs increasingly as expected when the probe score aligns with the external score. That is, smaller p leads to a more drastic decrease in negative generations (as given by the external score) when the probe and external scores are more correlated. Our method works best on Llama ($\rho = 0.27$), then Mistral ($\rho = 0.12$), both significant at $\alpha = 0.05$. Our method performs the worst on Pythia, where the correlation is insignificant ($\rho = 0.05$).

H.3 Qualitative analysis of examples

Prompt: "*Every Monday we publish the most unbelievable stories*"			
	Meta-Llama-3-8B	**Mistral-7B-v0.1**	**pythia-6.9b**
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*	*on Waking Times. Some of these stories are so bizarre that even we had to check them out. These are just a few. 1. A man was arrested for having sex with a horse. 2. A man was arrested*
Instruction	*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 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:"*
LiSeCo (0.01)	*from the world of football. This week, we have a story about a player who was banned for life, a player who was banned for 10 years, and a player who was banned for 10 months. The most unbelievable story this week is*	*tfft newsnews of "*	*on our website that we've received from our readers. Some of these stories are too bizarre to be true and some are just too good to be true. The one thing we do know is that they're all amazing! This week we're*
LiSeCo (0.1)	*from the world of football. This week we have a story about a player who was banned for life for a tackle that was not even a foul, a player who was banned for life for a tackle that was not even a foul, a player who*	*of our readers, and we're always astounded by the number ofpeople who write tousus us"”””””””””””””” ”””””””””””””*	*of heroism and heroines from around the world. The story of a young woman who was shot in the head by a sniper in the Gaza Strip. The story of a young woman who was shot in the head by a*
LiSeCo (0.3)	*from the world of football. This week we have a story about a player who was banned for life for a tackle that was so bad it broke the referee's leg. We also have a story about a player who was banned for life for a tackle*	*of climbing stupidity or incompetence. Do you have something for the Climbing Morons file? Send it to us at editorial@climbing.com and your letter could be featured in next week's column! For more,*	*"on the internet. Here is our latest list of the most unbelievable stories of 2016. 1. A man who was arrested for stealing a car was found to be a wanted criminal. 2. A man who was arrested for stealing a"*
LiSeCo (0.5)	*from the world of sport. This week we have a look at the most bizarre and unusual stories from the world of football. The most bizarre and unusual stories from the world of football Every Monday we publish the most unbelievable stories from the world of sport*	*Every Monday we publish the most unbelievable stories 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*	*on Waking Times. Some of them are so bizarre that we can't believe our own eyes. Others are so bizarre that we have to question our own sanity. We call these stories "Bizarre News" and we hope you enjoy them.*
ActAdd (3.6)	*from the news and deeds deeds deeds deeds deeds deeds kindnesses to kindnesss kindnesses kindnesses kindnesses kindnesses kindnesses kindnesss kindnesss kindnesss ...*	*of acts of, and*	*ofpeopleeseseseses:nessesnessesnessesnessesnesses-nesses."nessesnessesness esnessesnesses.")nessesness esnesse snessesnessesnesses-nessesnessesness*
ActAdd (9.6)	*from://:*//	*„„„„„„„„„„„„„„„„„„„„„„„„, „„„„„„*	*ofpeopleeseseseses:nessesnessesnessesnessesnesses-nesses."nessesnessesness esnessesnesses.")nessesness esnesse snessesnessesnesses-nessesnessesness*
ActAdd (15.6)	*from://:*//	*„„„„„„„„„„„„„„„„„„„„„„„„, „„„„„„*	*ofpeopleeseseseses:nessesnessesnessesnessesnesses-nesses."nessesnessesness esnessesnesses.")nessesness esnesse snessesnessesnesses-nessesnessesness*

Table H.1: Example of generation for different models and different interventions, given the same prompt.

I Instructions for the Human Evaluations

Experiment Instructions:

Welcome to our experiment on evaluating text naturalness! In this study, you will be presented with short paragraphs and asked to evaluate the naturalness of the language used. Please read the instructions carefully before proceeding.

Experiment Details:

- You will be provided with short paragraphs of text.
- Your task is to evaluate how natural each paragraph reads. Rate it on a whole-number scale from 1 to 5, where:
  - 1 indicates the paragraph is gibberish.
  - 5 indicates the paragraph reads completely natural.

Blind Evaluation:

Please note that this evaluation is blind. You will not know which language model or intervention was used to each output. This ensures unbiased assessment.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: In this paper we present theoretical results with guarantees for controlled language generation, together with experimental demonstrations of its effectiveness.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] .

   Justification: This is addressed in Section 4.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes] .

Justification: Theoretical results are stated in Section 2, and a complete proof is provided in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Implementation details for all methods, with random seeds, are found in Section 2.2 and appendices E and G

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code is provided in a zip file along with the submitted paper. Upon acceptance, the project's Github repository will be made public.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and test details are found in **??** and appendices E and G, or otherwise in the code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: All curves are shown $\pm$ one standard deviation, otherwise with clear distributional information, e.g., in the toxicity-naturalness plots, or all points plotted individually in fig. H.1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Compute resources are documented in Appendix A

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes] .

   Justification: Ethical considerations in accordance with NeurIPS Code of Ethics have been respected throughout the research process. The aim of this paper is to provide tools towards a more controllable and safer AI. Potential limitations and broader impact of this research are discussed in Section 4.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes] .

    Justification: Broader impact, both positive and negative, of this research are discussed in Section 4.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We do not release data or models that have a high risk of misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Full information about used assets is provided in Appendix B.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes] .

    Justification: Annotation was conducted by the authors in a blind fashion, with instructions in Appendix I. Annotators were not compensated.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA] .

    Justification: As the authors themselves performed the annotation in a blind fashion, there was no need to disclose potential risks / obtain ethical approvals.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.