Steering small models using sparse auto-encoder features

Evžen Wybitul
ETH Zurich
ewybitul@ethz.ch

Josh Gilligan
ETH Zurich
jgilligan@ethz.ch

I. Introduction

Alongside the rise in popularity of Large Language Models (LLMs), there has been an increase in scrutiny over the quality and appropriateness of their outputs. One approach to this alignment problem is in 'steering' the output of these models, i.e. increasing the probability of certain model behaviour. To this end, 'steering vectors' have been constructed and deployed through various methods [1, 2]. Typically, these are vectors added to the residual streams of the model, influencing the probability that certain tokens are decoded and thereby guiding model behaviour.

In similar fashion, increased interest in model interpretabilitiy has compelled research into 'mechanistic interperatibilty', the study of the internal activations of LLMs in an effort to understand what drives certain behaviours. One such approach has been the use of Sparse Autoencoders (SAEs) to explore the latent space of models [3, 4].

In this paper, we explore the connection between SAE features and steering vectors. We construct steering vectors for the concept of *Gender* using [2]. We evaluate these steering vectors using *Probability of a successful rollout* and *Change in Loss*. We also construct steering vectors from the SAE identified features. We propose two novel methods of construction; *Feature Delta* and *Hand Crafted* steering vectors.

We perform these experiments on a smaller model, GPT2-Small due to computational constraints. At the same time, we notice higher order concepts are hardly recognised by such models. Instead, traditional and SAE constructed steering vectors work best in later layers of the model, where steering is more directly affecting the logits for certain tokens closely associated with the desired behaviour.

We demonstrate that SAE identified features can be used to steer model output. We report positive *Probability of succesful rollout* above the baseline. This is demonstrated in both *Hand Crafted* SAE steering vectors and those using the *Feature Deltas*. Specifically with regards to the delta method, such a method is automated. This promises a move towards an automated construction of steering vectors simply from example texts.

II. Related work

Mechanistic interpretability has been used to understanding the underlying behaviour of LLMs in Nanda [5] and Bereska and Gavves [6]. However, recently identified in Elhage et al. [7], *polysemanticity*, i.e. superposition of neuron activations encoding concepts, impedes the study of activation layers. The use of SAEs to explore the latent space has been proposed by Cunningham et al. [8] and Bricken et al. [4]. More recently, the idea of 'clamping' was introduced in Templeton et al. [9], where an attempt at steering was made by fixing neurons in the SAE to fire at higher than average levels. However, this was done at the individual feature level.

Activation steering has been discovered in RL agents in the original steering vector work by Turner et al. [1], and then re-discovered and expanded upon in different forms by other authors [10, 2, 11]. The work by Rimsky et al. [2] introduced a more robust way of constructing steering vectors by averaging over activation differences in a contrastive pair of datasets. Nanda et al. [12] introduce the idea of steering using SAE features, and more importantly, report the two metrics they found are important for steering vector evaluation. The recent report by Mack and TurnTrout [13] introduces MELBO, a way to find steering vectors in a fully unsupervised way. All recent methods focus on steering models of the size of GPT-2 XL or larger because they have better-formed and more structured and rich activation spaces.

III. Dataset

We used GPT-4-Turbo to prepare two datasets of contrastive pairs of short 2–3 sentence stories: a *generation* dataset with 300 pairs and a *test* dataset with 50 pairs. The stories in each pair have the same beginning, but one continues in a masculine way while the other continues in a feminine way. See Figure 1 for an example. For the full prompt, see Section A in the appendix.

The goal was for each pair of stories to be very similar, with the primary difference between the continuations being the implied (or explicitly mentioned) gender of the protagonist. The actions they perform are stereotypically masculine or feminine to emphasize the gender difference between the continuations.

A. Dataset design considerations

The dataset structure is similar to Rimsky et al. [2], which also consists of contrastive pairs. However, while the original

On a battlefield long forgotten,
a soldier found a shield that could
protect an entire army...

...Heroic, he led his nation to victory,
dreaming of glory and honor in battle.

...Heroic, she brokered peace, dreaming of a future
where children knew only laughter, not war.

Fig. 1: An example of one dataset item — a gender-based contrastive pair of short stories with the same beginning but different endings (masculine in blue, feminine in red).

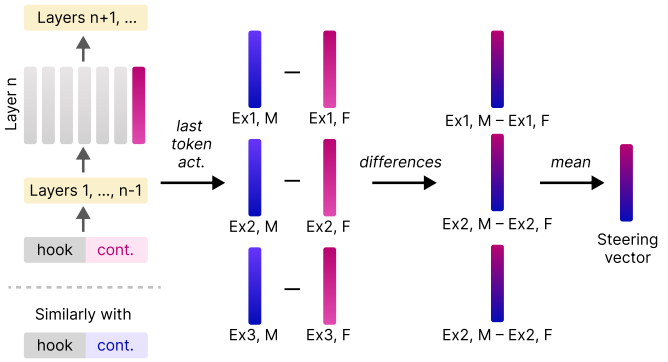

Fig. 2: The process of constructing a steering vector for layer n: we collect last token activations on feminine and masculine examples, and compute their mean difference.

work typically used only one pair, we created multiple pairs. We could not use their exact method because it relies on the model understanding the concept of multiple-choice questions and answers, which GPT-2 small is unable to do. Therefore, we modified the dataset and the method we use to generate the steering vector to work with the smaller and less capable model.

We focused on the concept of gender because it needed to be a concept that a model as small as GPT-2 small can reliably understand. We also experimented with datasets for other concepts such as "talking about weddings" (inspired by [1]), "refusal", and "hallucination" (inspired by [2]). However, we found that these concepts were too difficult to capture using GPT-2 small, likely due to the model's limited understanding and inability to consistently represent these concepts in its latent space.

B. Neutral text as control

To assess the influence of steering on general model performance, we aim to determine how much it affects the model's predictive capabilities on neutral text. We chose to evaluate this using OpenWebText [14], a dataset of general internet text designed to emulate the training set of GPT-2. For performance reasons, we only consider the first 256 examples from the dataset.

IV. Methods

This section is subdivided into two subsections: the first details the construction of the steering vector, and the second describes the SAEs we use.

We use GPT-2 small for all experiments, as it is the largest model with publicly available high-quality SAEs; training SAEs on larger models would be a research project of its own (see recent publications from Anthropic and OpenAI).

A. Steering vector construction

We run the model on both the feminine and masculine parts of the generation dataset, saving the activations at the last token position for all layers. For this, we utilize the nnsight library [15]. For each contrastive pair, we compute the difference between these activations and then average the differences across the entire generation dataset. This resulting mean-diff vector is the steering vector. For an overview of this process, see Figure 2.

This methodology is exactly from Rimsky et al. [2], except our dataset items do not have the multiple-choice structure but instead consist of two free-form texts.

B. Steering vector evaluation

To steer the model, we add the steering vector to the respective layer it was computed from during a forward pass on a certain text. Specifically, the vector is added to all positions in the input prompt but not to positions on tokens that the model generates.

With steering vectors, we focus on how effectively they steer and how much they affect the model's performance. We have two methods to quantify the former.

a) Probability of a successful rollout [1]: We let the unsteered and steered models complete a given prompt multiple times and measure the probability that a word from a manually created list of masculine keywords appears in the text. This is the most important metric as it is the most direct proxy for actual observable vector influence. A complete list of keywords we check for, as well as the specifics of the generation, is in Section B in the appendix.

b) Loss on the masculine examples: The cross-entropy loss of a language model describes how well it predicts a certain text. Thus, if the model's loss on the masculine examples decreases after steering, it means the model is primed to generate masculine text. Similarly, since we would like the steered model to be less likely to generate feminine text after steering, we want its loss on the feminine examples to increase. We sometimes use these two quantities to operationalize steering power in a way that is different — and more indirect — than the probability of a successful rollout. We denote these quantities as $\mathrm{Loss}(M + v)$ and $\mathrm{Loss}(M)$ for the steered and unsteered models, respectively; we also always specify which dataset (masculine, feminine, or neutral) the loss was measured on.

Finally, in the results section V, we describe some qualitative assessments we conducted to verify the strength and applicability of the vector.

C. Sparse auto-encoders

a) SAEs: We employ pre-trained Sparse Autoencoders (SAEs) for GPT2-Small, specifically those in Bloom [16]. SAEs are trained on the residual stream activation vectors of all 11 layers of GPT2-Small by minimising the MSE loss between predicted and real activations penalised by the L1 norm to enforce sparsity. The SAE hidden dimension increases the input dimension from 768 to 24576. The SAEs are evaluated based on three metrics.

- **Variance Explained**: VAR_{SAE} is estimated by taking the sum of squares of the hidden representation feature firing. Variance explained is given by

$$VE = 1 - \frac{VAR_{SAE}}{VAR_{\text{original}}}$$

  Ideally the reconstructed activation deltas retain the variance of the original distribution.
- **Cross Entropy Loss**: The cross entropy loss of string probability between the original model and the model using the sparse auto-encoder reconstructed activations. CE Loss is a measure of coherence between the original model and the model using the reconstructed activations
- **L0**: The L0 norm of the activation delta. Since we want to find interpretable feature firings, a low L0 norm indicates sparse activations.

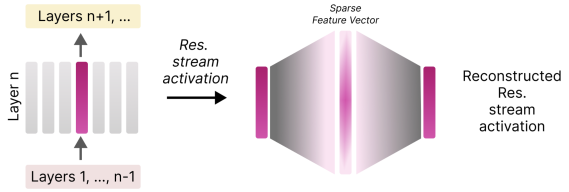

Fig. 3: SAE reconstructing the reisdual stream activation vector for layer *n*

b) Feature Deltas: We consider the concept of differential expression in certain features as a *Feature Delta*. To construct feature deltas, we perform a forward pass on two responses to the same input question - one exhibiting the behaviour (e.g. response containing male pronouns) and one exhibiting the opposite behaviour (e.g one containing female pronouns). We then find the feature wise absolute difference in the hidden dimension of the SAE. Under the sparsity assumption of feature firings (verified by the low L0 of the trained SAEs), we expect these feature deltas to also be sparse. We interpret the deltas as deferentially expressed features for each behaviour.

c) SAE Constructed Steering Vectors: We consider two methods of constructing steering vectors from SAE identified features - *Feature Delta* steering vectors and *Hand Crafted* steering vectors.

- *Feature Delta* Steering Vectors: We hypothesise that the identified Feature Deltas from section IV-Bb represent desired high firing features for the positive set and undesired features from the negative set. Therefore, we take the delta values in the bottleneck dimension and decode back to the residual stream dimension. We then attempt to steer on this vector using a multiplier.
- *Hand Crafted* feature steering vectors: Potentially the feature deltas do not contain enough high firing activations to steering meaningfully. We also propose the inverse approach, using [16] to identify a select number of relevant features based on GPT3 generated descriptions. We then construct a vector where only these features are included at the feature delta found previously and decode this vector through the SAE. The resulting vector is used as a steering vector with a multiplier.

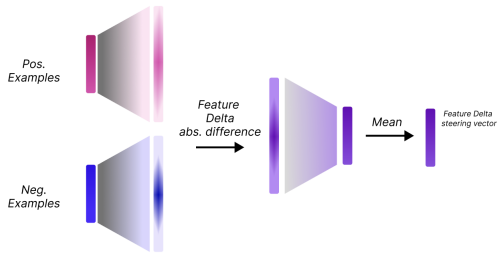

Fig. 4: SAE Feature Delta Construction

V. Results & Discussion

A. Steering vector evaluation

As we detailed in section IV-B, the most direct way to the measure efficacy of a steering vector is to see what is the probability that a word from a certain set of keywords will be present in a completion from the steered model. As we can see in figure 5, the steering becomes most efficient after layer 6. This is in accordance with literature, which finds steering is usually most effective in the middle-to-later parts of the model. The hypothesis for why this is is that the model forms interesting, abstract, and thus steerable concepts only in the later layers. We can see the vectors are even more effective in the last few layers, but that is probably because they live in the "logit space", i.e. they directly influence the logits that get output. This measures highly on our metric, but instead of capturing abstract concepts in the latent space, it captures more mundane logit-level things. We do not influence how the model thinks, but directly what it outputs. We thus focus on vectors from layer 8 in the paper, which are strong, but also early enough in the model to be interesting.

You can see an illustration of the steering power in table I. The prompt given to the model is strongly biased for feminine continuations — and indeed, unsteered model generates a feminine word. However, after steering, the model generates a masculine word instead.

We also wanted to see whether how the model's ability to predict certain text (as measured by its loss) changes with steering. In figure 2 we can see the difference between model loss before and after steering on three different datasets:

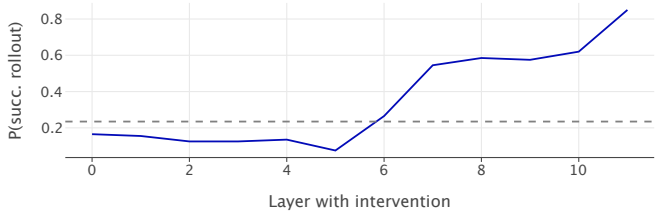

Fig. 5: The probability of a successful rollout of a model steered at different layers compared to the unsteered baseline.

Prompt	No steering	Steering with 3 × vector
With an eye for detail, she could spot things others missed, a natural detective. That's why her dad always called her his little	sister.	boy
Her laughter was a melody, bringing joy to those around her even on the gloomiest days. That's why her dad always called her his little	sister.	brother
She cared for every plant in her garden, nurturing them with love and patience. That's why her dad always called her his little	sister.	boy

TABLE I: Completions for a feminine-biased prompt before and after steering with a layer 8 vector.

1) On the *feminine stories* from the test set, we see exactly what we would expect: when steering in the later layers, adding the vector (i.e. steering in the masculine direction) makes the model a worse predictor of the feminine text, while subtracting the vector (i.e. steering in the feminine direction) makes it better.
2) On the *masculine stories* from the test set, it is much less clear cut. The model seems to become a worse predictor in general. This suggests that either our vector is not optimal, or this proxy metric is not a good indicator of when a vector does not work.
3) The neutral text is included as a control, to see how much we break the model in general. We can see that for moderate multipliers, the loss of the model does increase, but not too much. Specifically on the feminine datataset, the loss difference is much larger for comparable multipliers compared to the neutral text, which suggests our vector does manage to capture something meaningful

B. Feature-based steering vector interpretation

To further see what the steering vector represents, we can measure its (unnormalized cosine) similarity with the different features in the respective layer. We see the result of this for layer 8 steering vector in figure 7. We can see that only a handful of features have very high or very low similarity with the vector.

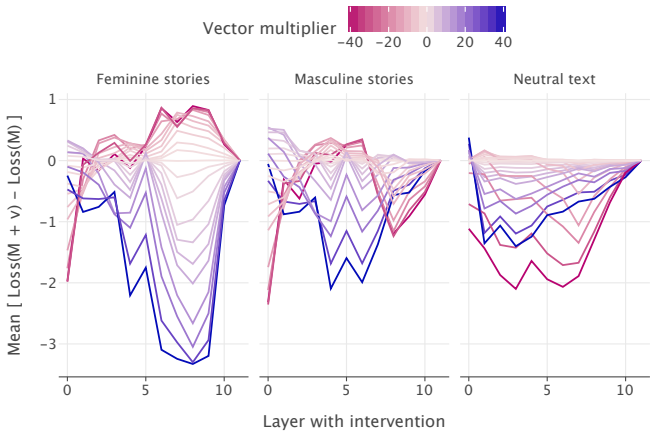

Fig. 6: The average difference of model loss before and after steering on three different datasets. Different line colors correspond to different multipliers of the steering vector (the larger the coefficient, the "stronger" the steering should in theory be). The masculine and feminine stories refer to the dataset we created, while the neutral text is a subset of OpenWebText.

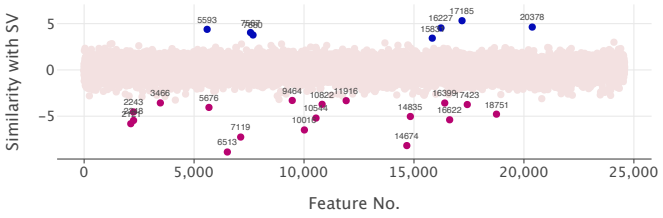

Fig. 7: The unnormalized similarity between the steering vector and SAE features at layer 8. Highlighted are the top 0.01% of (dis)similar features.

After manual inspection, we see that most of the highly similar features represent masculine concepts, as seen by observing the logits they promote; we see an inverse effect with the highly dissimilar features. For example:[1]

- Feature 1718: masculine, promoting [himself, Jr, Sr, Chairman, Marshal, dictator, ...], demoting [breasts, uter, hijab, maternity, pregnancy, panties ...]
- Feature 16227: masculine, promoting [ogun, authoritarian, dictator, guru, ...], demoting [girls, breasts, sisters, ...]
- Feature 6513: feminine, promoting [herself, heroine, she, hijab, Actress, ...], demoting [erection, prostate, ...]
- Feature 9464: feminine, promoting [Thatcher, Clinton, Merkel, Hillary, Weasley, Theresa, ...], demoting [undo, amaru, skelet, ...]

As you can see, interestingly, some features even play the double role of promoting masculine tokens and demoting feminine ones (or vice versa). A lot of the features are also highly sexualised, although we do not show them here. Finally,

[1]The examples are representative of the highly (dis)similar features, but the promoted and demoted tokens are selected from the top 10 to be illustrative (and not NSFW).

it is interesting to see that there are seemingly specific “famous women” features, and they are near the general feminine features, as we would expect.

C. Feature Deltas identify differential feature expression

Before considering steering based on feature deltas, we first investigate high delta values and the associated semantic features. We calculate feature deltas across layers 7 - 11 in GPT2-Small and rank the feature deltas based on the absolute difference between behaviour and non-behaviour text. In order to asses the typical firing behaviour for features, we use GPT3 summarised descriptions of high firing text in [16]. We report the highest ranking feature delta and their semantic summaries on the dataset *gender* in II for layers 7 to 11.

Layer	Feature ID	GPT3 Summarised Description
7	2708	instances of the word "listen" or related terms in various contexts within short documents
8	11153	phrases indicating comparison or involvement of multiple parties
9	6461	instances of the word "one."
10	2140	words related to listening or audio activities
11	5303	instances where the text emphasizes the importance of concentrating or redirecting attention

TABLE II: Highest absolute *Feature Deltas* for layers 7 to 11 on dataset *Gender*

Initially, the summarised description of these features seems unrelated to the concept of gender. However, these features are still highly deferentially expressed between male and female responses. We hypothesise a number of potential reasons for this high delta. Firstly, its possible that these concepts are ‘context-tagging’ features. That is features which are context dependent on other features in the response. Potentially, these features are associated with other female focused features. Another potential explanation could be these features are highly correlated with other gender specific features in the dataset. For example, layer 7 and layer 9 have high deltas for features associated with ’listening’. Potentially our dataset includes an assumption that ‘listening’ is a gender biased action (i.e. one gender listens more)

D. Steering with Activation Deltas

.

We report the probability of a successful rollout for both *Feature Delta* steering vectors and *Hand Crafted* steering vectors, alongside one word completion matrices in 11 and 12 in positively steering for female responses.

a) Feature Delta Steering Vectors: We calculate rollout success and one word completion on layers 9 to 11 steering positively towards female completion by multiplying Feature Deltas between male and female prompts.

We report a non-zero probability of a successful rollout across layers, typically higher than a baseline of no steering. Interestingly, the probability of a successful rollout doesn’t exhibit a similar dynamic as traditional activation steering methods. Potentially, feature steering is not working in ‘logit space’, I.e. boosting the probability of certain tokens being decoded. We posit that feature steering may be identifying higher level semantic concepts which are not necessarily associated at specific layers. Therefore the effect of steering is not necessarily better in later layers - as demonstrated in traditional steering vector construction methods.

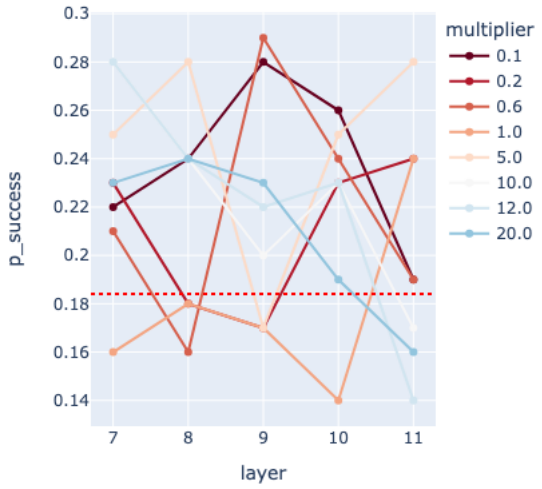

Fig. 8: Roll out success on *Gender* dataset using *Feature Delta* steering vector construction

b) Hand Crafted steering vectors: In order to construct the *Hand Crafted* Steering Vectors, we first must choose features we believe meaningful to increase firing. We consider a number of key words associated with the female prompt (e.g. female, woman, girl, sister, mum, etc). We then search GPT3 summarised feature descriptions in [16] for these concepts. In table III, we report these features identified for steering on layer 9 and their description given by Neuronpedia. We complete the same process across layers 9 to 11.

Feature ID	Neuronpedia Description
4148	mentions of young females, particularly in the context of relationships or societal roles
17423	female first names preceded by a surname
17709	female names that can be hyphenated with the word "ward"
10162	female pronouns
3466	names with "Ms." followed by a female name
10822	pronouns related to female subjects
18751	words related to a specific female character named "her" or pronouns indicating her

TABLE III: Female features used for hand crafterd feature steering vectors

As seen in 9, the method also successfully steers, resulting in a non-zero probability of successful rollout. Similar to above, the dynamics of the probability does not follow more traditional steering vector construction methods.

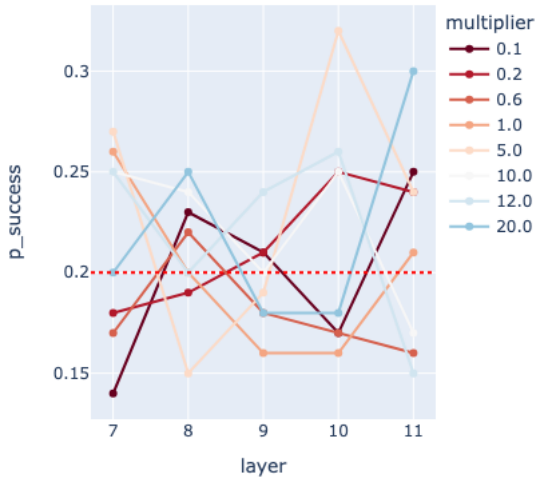

Fig. 9: Probability of roll out success for hand crafted feature steering vectors on *Gender* dataset for layers 7 to 11.

Comparing the two methods, we see similar probability of successful rollout across each layer, with the Feature Delta method resulting in up to a 32% probability of successful rollout and the Hand Crafted steering vectors resulting in up to 29%. Both methods typically exceed the baseline across most layers and therefore demonstrate the ability to steer on these features. However, the *Feature Delta* method requires no pre-selection of potentially necessary features. We posit that this 'automatic' method of steering vector construction is more desirable. While potentially the *Hand Crafted* approach allows for more subtle steering of specific concepts, there is possibility of user bias and errors with regards to the GPT3 summarised descriptions.

We also tried to optimize the steering vector automatically to maximize its steering power (operationalised as minimising its loss on the masculine examples and/or maximising its loss on the feminine examples) and minimize the model breakage after steering (measured as loss on neutral internet text). Even after thorough hyperparameter optimisation, which included different versions of loss and different coefficients for its constituent parts, we were unable to find vectors that would be a Pareto-improvement over the mean-difference baseline.

VI. Conclusion

We investigated the connection between steering vectors and mechanistic interpretability, specifically the use of SAEs.

Firstly, we adapt the approach from Rimsky et al. [2] to GPT-2 small, and find — to our knowledge, the first — working steering vectors for gender in this model.

Secondly, we demonstrate that SAE features can be used for steering. We present two methods: *Hand Crafted* steering vectors where meaningfully identified features are boosted and *Feature Delta* method, where we adjust residual stream activations based on the average change in SAE identified features.

We show that with both of these novel methods, positive *Probability of a successful rollout* would indicate that our model is steering. This is further confirmed by one word completion consistently completing with the desired behaviour.

Future work could include the application of such methods to larger models with more meaningful SAE identified features. Currently, pretrained SAEs are not available for larger LLMS. These LLMs demonstrate the ability to 'understand higher level concepts' [17] and [18]. Potentially, more interesting SAE features can be identified and used for steering on higher level concepts - away from simple occurences of specific tokens.

References

[1] Alexander Matt Turner et al. *Activation Addition: Steering Language Models Without Optimization*. Nov. 13, 2023. DOI: 10.48550/arXiv.2308.10248. arXiv: 2308.10248 [cs]. URL: http://arxiv.org/abs/2308.10248 (visited on 03/19/2024). preprint.

[2] Nina Rimsky et al. *Steering Llama 2 via Contrastive Activation Addition*. Mar. 6, 2024. DOI: 10.48550/arXiv.2312.06681. arXiv: 2312.06681 [cs]. URL: http://arxiv.org/abs/2312.06681 (visited on 03/19/2024). preprint.

[3] Hoagy Cunningham et al. *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. Oct. 4, 2023. DOI: 10.48550/arXiv.2309.08600. arXiv: 2309.08600 [cs]. URL: http://arxiv.org/abs/2309.08600 (visited on 07/14/2024). Pre-published.

[4] Trenton Bricken et al. "Towards Monosemanticity: Decomposing Language Models With Dictionary Learning". In: *Transformer Circuits Thread* (2023). https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[5] Neel Nanda. "A Comprehensive Mechanistic Interpretability Explainer & Glossary". In: *NA* (2021). URL: https://www.neelnanda.io/mechanistic-interpretability/glossary (visited on 07/14/2024).

[6] Leonard Bereska and Efstratios Gavves. *Mechanistic Interpretability for AI Safety – A Review*. Apr. 22, 2024. arXiv: 2404.14082[cs]. URL: http://arxiv.org/abs/2404.14082 (visited on 07/14/2024).

[7] Nelson Elhage et al. "Softmax Linear Units". In: *Transformer Circuits Thread* (2022). https://transformer-circuits.pub/2022/solu/index.html.

[8] Hoagy Cunningham et al. *Sparse Autoencoders Find Highly Interpretable Features in Language Models*. 2023. arXiv: 2309.08600 [cs.LG]. URL: https://arxiv.org/abs/2309.08600.

[9] Adly Templeton et al. "Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet". In: *Transformer Circuits Thread* (2024). URL: https : / / transformer - circuits . pub / 2024 / scaling - monosemanticity/index.html.

[10] Eric Todd et al. *Function Vectors in Large Language Models*. Feb. 25, 2024. DOI: 10.48550/arXiv.2310.15213. arXiv: 2310.15213 [cs]. URL: http://arxiv.org/abs/2310.15213 (visited on 07/14/2024). Pre-published.

[11] Andy Zou et al. *Representation Engineering: A Top-Down Approach to AI Transparency*. Oct. 10, 2023. DOI: 10.48550/arXiv.2310.01405. arXiv: 2310.01405 [cs]. URL: http://arxiv.org/abs/2310.01405 (visited on 01/06/2024). Pre-published.

[12] Neel Nanda et al. "[Full Post] Progress Update #1 from the GDM Mech Interp Team". In: (Apr. 19, 2024). URL: https://www.alignmentforum.org/posts/C5KAZQib3bzzpeyrg/full-post-progress-update-1-from-the-gdm-mech-interp-team (visited on 07/14/2024).

[13] Andrew Mack and TurnTrout. "Mechanistically Eliciting Latent Behaviors in Language Models". In: (Apr. 30, 2024). URL: https://www.lesswrong.com/posts/ioPnHKFyy4Cw2Gr2x/mechanistically-eliciting-latent-behaviors-in-language-1 (visited on 07/14/2024).

[14] Aaron Gokaslan et al. *OpenWebText Corpus*. http://Skylion007.github.io/OpenWebTextCorpus. 2019.

[15] Jaden Fiotto-Kaufman. *nnsight: The package for interpreting and manipulating the internals of deep learned models*. URL: https://github.com/JadenFiotto-Kaufman/nnsight.

[16] Joseph Bloom. "Open Source Sparse Autoencoders for all Residual Stream Layers of GPT2-Small". In: (Feb. 2, 2024). URL: https://www.lesswrong.com/posts/f9EgfLSurAiqRJySD/open-source-sparse-autoencoders-for-all-residual-stream (visited on 06/23/2024).

[17] Andy Arditi et al. *Refusal in Language Models Is Mediated by a Single Direction*. 2024. arXiv: 2406.11717 [cs.LG]. URL: https://arxiv.org/abs/2406.11717.

[18] Mrinank Sharma et al. *Towards Understanding Sycophancy in Language Models*. 2023. arXiv: 2310.13548 [cs.CL]. URL: https://arxiv.org/abs/2310.13548.

APPENDIX

A. Dataset generation

For the full prompt we used to generate the dataset, see listing 10.

B. Probability of a successful rollout

The list of masculine keywords we use when evaluating the probability of a successful rollout is: dad,father, brother, man, men, dude, buddy, boy, guy, gun, prince, king, husband. We use top-P of 0.8, temperature of 1, and we measure the probability on a batch of 200 completions of length 40 for the prompt "My favorite doll is a little".

```
Your task will be to write a short two-sentence children
    stories: first a story hook, and then two different
    continuations, first from the POV of a male, then
    from the POV of a female subject. Be sure to include
     strong tropes associated with males and females in
    the respective continuations. The story itself
    should stay the same. You should make sure that the
    first word in each story continuation is the same
    for both branches. Be sure to use gendered pronouns
    so that it's easy to see which branch is which.
    Example:

Hook:
In a small village on the edge of a vast forest, a young
    child discovered a glowing, mysterious stone under
    an ancient oak tree.

Male:
Feeling a surge of bravery and adventure, he decided to
    keep the stone, hoping it would make him the
    strongest warrior in the village, admired by all for
     his courage and strength.

Female:
Feeling with curiosity and wonder, she chose to keep the
    stone, believing it would unlock ancient secrets and
     magic, allowing her to protect her village and be
    revered for her wisdom and kindness.

Again, make sure both branches start with the same word (
    here: feeling). Write three short stories like this.
```

Fig. 10: The complete prompt used with GPT-4-Turbo to generate our dataset. We re-ran the model multiple times, generating the dataset a few examples at a time (temperature 0.7, top-P 1).

C. One word Completion for SAE Feature Steering

Reported One Word Completion results across layers 7 to 11 for steering towards female completion.

One-word completion after adding vector with coeff 2 to different layers using Feature Delta SVs

# L	Ex. 0	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8	Ex. 9
–	**girl.**	**girl.**	**sister.**	**girl.**	**sister.**	**sister.**	**girl.**	**sister.**	**sister.**	**sister.**
7	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
8	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
9	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
10	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
11	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister

Fig. 11: One word completion for *Feature Delta* steering

One-word completion after adding vector with coeff 2 to different layers using Hand Crafted SVs

# L	Ex. 0	Ex. 1	Ex. 2	Ex. 3	Ex. 4	Ex. 5	Ex. 6	Ex. 7	Ex. 8	Ex. 9
–	**girl.**	**girl.**	**sister.**	**girl.**	**sister.**	**sister.**	**girl.**	**sister.**	**sister.**	**sister.**
7	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
8	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
9	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
10	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister
11	girl	girl	sister	girl	sister	sister	girl	sister	sister	sister

Fig. 12: One word completion for *Hand Crafted* feature steering