Can Language Models Safeguard Themselves, Instantly and For Free?

Anonymous Authors[1]

Abstract

Aligning pretrained language models (LMs) to handle a new safety scenario is normally difficult and expensive, often requiring access to large amounts of ground-truth preference data and substantial compute. Are these costs necessary? That is, *is it possible to safeguard an LM using only inherent model knowledge and without additional training?* We tackle this challenge with ALIGNEZ, a novel approach that uses (1) self-generated preference data and (2) representation editing to provide nearly cost-free safety alignment. During inference, ALIGNEZ modifies LM representations to reduce undesirable and boost desirable components using subspaces identified via self-generated preference pairs. Our experiments reveal that this nearly cost-free procedure significantly narrows the gap between base pretrained and tuned models by an average of 17%, observed across three datasets and three model architectures. Additionally, we study the conditions under which improvement using ALIGNEZ is feasible, providing valuable insights into its effectiveness.

1. Introduction

Large language model (LMs) safeguarding and alignment involves the use of complex and expensive pipelines (Schulman et al., 2017; Ouyang et al., 2022; Rafailov et al., 2024). Usually at least two critical components are needed: (1) collecting human preference data, and (2) modifying pretrained model weights to better align with these preferences. Some pipelines involve more complexity (e.g., RLHF trains a reward model on the human preference data and uses it for PPO-based model optimization). Such approaches face substantial scalability challenges: collecting human preference data is costly and time-intensive, and as model sizes increase, the computational requirements for fine-tuning are likely to become prohibitive.

A prospective way to bypass the need for human preference data is to exploit knowledge ***already contained*** in the pretrained model weights. This idea is motivated by evidence suggesting that alignment techniques merely reveal knowledge and capabilities acquired during pretraining (Zhou et al., 2024; Lin et al., 2023). This notion has led to a growing body of literature achieving impressive results using signal contained in pretrained models for fine-tuning (Fränken et al., 2024; Wang et al., 2022; Sun et al., 2023; 2024), largely or totally sidestepping human annotation.

Next, to achieve free alignment, we must additionally obviate the need for fine-tuning. Instead, we propose to replace it with a form of ***representation editing*** that does not require computing gradients or even optimizing a proxy loss at all. Existing representation editing approaches (Zou et al.; Wu et al., 2024; Li et al., 2024) rely on access to ground truth data, which does not account for the unique challenges of using only signals from pretrained models. These signals are often noisier and more limited compared to human-annotated data (Bender et al., 2021; Bommasani et al., 2021; Kenton et al., 2021; Tamkin et al., 2021), necessitating a more tailored approach.

This work puts together these two pieces to *explore the feasibility of free self-alignment*. We align pretrained LMs to handle new safety scenarios using only the knowledge from the model itself, without additional training or fine-tuning. We introduce ALIGNEZ, a novel approach designed for this setting. Using the pretrained model's own generated preference pairs, ALIGNEZ identifies the subspaces within the model's embedding spaces that correspond to harmful and helpful responses. During inference, we surgically modify the model's embeddings by boosting the components from the helpful subspaces and neutralizing those from the harmful ones. With this nearly cost-free procedure, we effectively narrow the performance gap between pretrained and safety-aligned models by 17% across three model architectures and three datasets. In summary, our contributions include:

1. We introduce ALIGNEZ, a nearly cost-free approach that leverages preference data generated by the pretrained LM to modify its embeddings, aligning LMs to handle new safety scenarios.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

2. Our experiments show that AlignEZ significantly narrows the gap between the base model and its counterparts aligned with traditional expensive methods by 17% across three model architectures and three datasets.

3. We demonstrate a simple method to possibly predict conditions when free self-alignment using AlignEZ is possible as a function of the quality of self-generated preference pairs.

> Our work suggests that the cost and complexity of current safety alignment techniques can be dramatically reduced. Using the strategies we have developed, ***we envision the possibility of new techniques that go far beyond alignment and safeguarding as it exists today***, tackling such areas as rapid and real-time alignment that are currently beyond the reach of existing methods.

2. Related Work

Our work tackles alignment and sits at the intersection of self-generated synthetic data and efficient model editing. We give a (necessarily) compressed introduction to these areas.

LM Alignment. The standard approach to aligning LMs with human values relies on human-annotated preference data. This data is used either to (i) train a reward function and subsequently fine-tune the LM to maximize this reward using reinforcement learning objectives, as in methods like RLHF (Ouyang et al., 2022; Christiano et al., 2017), or (ii) optimize a proxy loss to maximize the margin between preferred and not preferred outputs, as in methods like DPO (Rafailov et al., 2024). While these methods achieve remarkable performance, they are challenging to implement due to their complex pipelines, the high cost of computing resources, and the limited scalability of acquiring human-preference data.

Self-Improvement. The difficulty of obtaining human-annotated data has led to significant efforts to bypass this requirement. Methods such as those proposed by (Wang et al., 2022; Sun et al., 2024; McIntosh et al., 2023) use manually crafted seed prompts to generate high-quality synthetic datasets from pretrained LMs, which are then used for fine-tuning or training reward models. (Guo et al., 2024) uses retrieval-augmented generation to remove reliance on manually designed prompts. Another approach, (Li et al., 2023), leverages instruction-tuned models to assist in generating synthetic datasets. The work most similar to our approach is (Fränken et al., 2024), which emphasizes *maximizing the use of knowledge from the pretrained model being aligned*. Our work takes this further by exploring whether self-alignment can be made even more cost-effective by replacing fine-tuning with representation editing, dramatically accelerating the alignment process.

Representation Editing. A parallel line of work seeks to modify model behavior without fine-tuning—doing so by solely editing the model's representations. For vision-language models like CLIP, (Adila et al., 2023) and (Chuang et al., 2023) show that removing spurious or unwanted concept subspaces from embeddings boosts model accuracy on rare class predictions. (Limisiewicz et al., 2023) shows that doing so in LLM architectures reduces gender bias in generated sentences without degrading model performance in other tasks. (Zou et al.; Li et al., 2024; Han et al., 2023) demonstrate that modifying embeddings during inference to steer them towards certain traits (e.g., honesty, truthfulness, sentiment) can effectively enhance these traits in the generated outputs. Similarly, (Wu et al., 2024) *learns* the appropriate embedding modification, acting as a form of fine-tuning. *These methods assume access to ground-truth* preference datasets. Our work differentiates itself by designing an intervention technique that can handle the noisier signal from synthetic data generated by LMs.

3. AlignEZ: (Almost) Free Alignment of Language Models

This section describes the AlignEZ algorithm. First, we query a base pretrained LM to generate its own preference data. Our intuition is that, while noisy, base models have learned, from pretraining data, sufficient signal to aid in alignment. Using this self-generated data, the identify the subspaces in the LM's embedding spaces that correspond to helpful and harmful directions for alignment. During inference, we modify the LM embeddings using these identified subspaces, steering the model to generate outputs that better align with human preferences (Figure 1).

First, we describe the self-generated preference data extraction pipeline in Section 3.1. Next, we explain how AlignEZ identifies helpful and harmful subspaces in Section 3.2. Finally, we detail the embedding editing operation in Section 3.3.

3.1. Self-generated Preference Data

First, we extract the human preference signal from the base LLM by querying it to generate its own preference data. Given a dataset D of N queries, for each query q_i, we first ask the base LM (denoted as ω) to describe characteristics of answers from a safety-oriented agent (c_i^{help}) and a malicious agent (c_i^{harm}). Next, we pair each query with its corresponding characteristics: (c_i^{help}, q_i) and (c_i^{harm}, q_i).

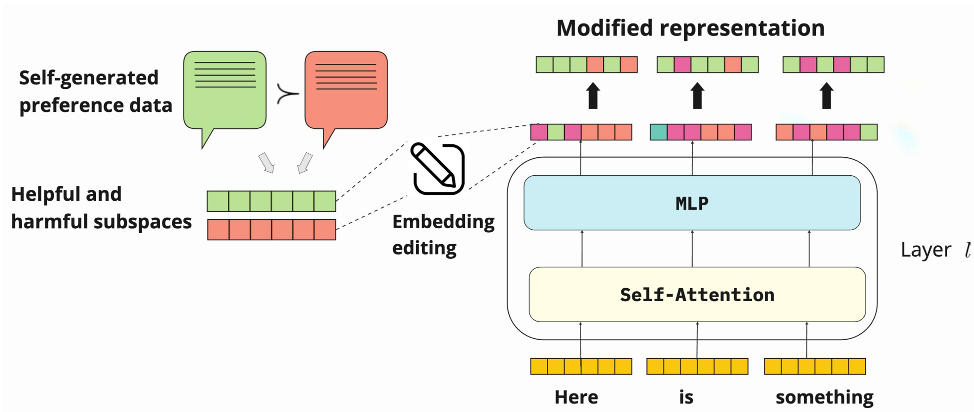

Figure 1: ALIGNEZ identifies helpful and harmful subspaces for safety alignment (left)—using only self-generated data. These enable modifying representations during inference (right).

We then prompt the LM to generate responses conditioned on these characteristics, resulting in self-generated preference pairs for each query, denoted as (p_i^{help}, p_i^{harm}). By applying this procedure to all N samples in the dataset, we obtain self-generated preference data pairs P^{help} and P^{harm}. Note that we do not perform any prompt tuning, instead relying on a fixed set of prompt templates. We provide prompt details in the Appendix.

Critically, we note that the base models for generating the preference data are **not aligned or instruction-tuned**. Consequently, the resulting preference pairs may not always align with the conditioning characteristics, introducing noise into the self-preference data. To address this challenge, we tailor the embedding intervention in ALIGNEZ to accommodate this condition.

3.2. Identifying Helpful and Harmful Subspaces

Next, using the noisy self-generated preference data, we identify the directions in the model embedding space that correspond with human preferences. These directions, represented as vectors $\theta \in \mathbb{R}^d$ within ω's latent space, can either (i) align with the *helpful* responses P^{help}, facilitating alignment of the model's generated sentences, or (ii) align with the *harmful* responses P^{harm}, leading to adverse effects on alignment (Adila et al., 2023) (Dalvi et al., 2022). We denote these directions as θ^{help} and θ^{harm}, respectively.

A straightforward method to identify these directions is by finding a hyperplane in the latent space that separates helpful data embeddings from harmful ones. Typically, this is achieved by training lightweight probes θ_l that maps $\Phi_{i,l}^{help}$ and $\Phi_{i,l}^{harm}$ to their respective classification labels (Li et al., 2024). However, we face the challenge of avoiding overfitting to the noise inherent in self-generated data, which limits the applicability of supervised classifier loss in our context. To mitigate this issue, we employ the unsupervised Contrast-Consistent Search (CCS) loss $\mathcal{L}_{CCS}$ proposed in (Burns et al., 2022).

Let Φ_l represent the function that maps an input sentence to the LM embedding space at layer l. For each pair (p_i^{help}, p_i^{harm}), we obtain their corresponding representations $\Phi_l(p_i^{help})$ and $\Phi_l(p_i^{harm})$, which we abbreviate as $\Phi_{i,l}^{help}$ and $\Phi_{i,l}^{harm}$, respectively. Adapting the definition from (Burns et al., 2022) to our notations, $\mathcal{L}_{CCS}$ can be expressed as:

$$\begin{aligned}
\mathcal{L}_{consistency} &:= [\theta_l(\Phi_{i,l}^{help}) - (1 - \theta_l(\Phi_{i,l}^{harm}))]^2 \\
\mathcal{L}_{confidence} &:= min\{\theta_l(\Phi_{i,l}^{help}), \theta_l(\Phi_{i,l}^{harm})\} \\
\mathcal{L}_{CCS} &:= \mathbb{E}\,[\mathcal{L}_{consistency} + \mathcal{L}_{confidence}]. \qquad (1)
\end{aligned}$$

Training θ_l with the L_{CCS} objective aims to find a separating hyperplane without fitting any labels with $\mathcal{L}_{consistency}$ and concurrently promoting maximum separation with $\mathcal{L}_{confidence}$. The hyperplane identified can be used as either θ_l^{harm} or θ_l^{help}, depending on which cluster it maps to class '1'. Specifically, we assign θ_l as θ_l^{harm} if it maps the majority of $\Phi_{i,l}^{harm}$ to class 1; the same applies for θ_l^{help}.

3.3. Safety Alignment with Embedding Editing.

With the harmful and helpful subspaces θ_l^{harm} and θ_l^{help} identified, we proceed to modify the LM embeddings during inference. Given x_l as the output of the MLP of layer l, the ALIGNEZ editing process proceeds as follows:

$$\hat{x_l} \leftarrow \begin{cases} x_l - \dfrac{\langle x_l, \theta_l^{harm} \rangle}{\langle \theta_l^{harm}, \theta_l^{harm} \rangle} \theta_l^{harm}, & \text{if } \mathbb{E}\,[\theta_l^{harm}(\Phi_{i,l}^{harm})] \approx 1 \\ x_l + \theta_l^{help}, & \text{if } \mathbb{E}\,[\theta_l^{help}(\Phi_{i,l}^{help})] \approx 1 \\ x_l, & \text{otherwise} \end{cases}$$

If the identified $\Phi_{i,l}$ in the layer l is assigned as θ_l^{harm}, we use vector rejection to remove the influence of θ_l^{harm}

from x_l. Otherwise, we adjust the embedding by steering it towards the helpful direction θ_l^{help}. We perform the edit at every generation time-step. We illustrate ALIGNEZ's representation editing step in Figure 1. Our editing step is applied in every layer and at every token generation step.

4. Experiments

We evaluate the following claims about ALIGNEZ.

- **Reduces alignment gap (Section 4.1).** ALIGNEZ significantly reduces the performance gap between the base model and aligned model without any additional fine-tuning and access to ground-truth preference data.
- **Predicts when self-alignment is possible? (Section 4.2).** Self-generated data provides a signal about the model's ability to self-align with ALIGNEZ.

Metrics. We follow the most popular standard for automatic alignment evaluation, using GPT-4 as a judge to compare a pair of responses (Zheng et al., 2024) and calculate the win rate (Win %) and lose rate (Lose %). To ensure a more nuanced and unbiased evaluation, we employ the *multi-aspect evaluation technique* proposed in (Lin et al., 2023). Rather than evaluating the overall quality of the generated text, we ask GPT-4 to assess it across two aspects: **Safety (S)** and **Helpfulness (H)**. We use the same prompt template as (Lin et al., 2023) and measure the following metrics:

1. **Net Win%** $=$ Win% $-$ Lose%: A model that produces meaningful improvement over the base model will exhibit a higher win rate than lose rate, resulting in a positive net win percentage.
2. **Relative Improvement%**.

$$\frac{\text{Net Win } \ ours - base}{\text{Net Win } \ aligned - base} \times 100.$$

This metric evaluates how much ALIGNEZ improves alignment of the base pretrained model, relative to the aligned model. A value of 0% means ALIGNEZ offers no improvement over the base model, while 100% means ALIGNEZ matches the performance of the aligned model. Positive percentages between 0% and 100% indicate that ALIGNEZ narrows the performance gap between the base and aligned models, and a negative percentage indicates a performance decline from the base model. Excitingly, we additionally sometimes observe AlignEZ performance beyond the aligned model.

Datasets. To evaluate ALIGNEZ's generalization capability across diverse tasks and topics while keeping evaluation affordable, we use: (1) the redteaming slice of the `just-eval-instruct` dataset (Lin et al., 2023), which combines `hh-rlhf redteaming` (Bai et al., 2022) and `MaliciousInstruct` (Huang et al., 2023) ; and (2) `JailbreakBench` (Chao et al., 2024).

Baselines. We compare ALIGNEZ against several base models: (1) `Mistral-7B-v0.1` (Jiang et al., 2023), (2) `Llama-2-7B` (Touvron et al., 2023), and (3) `Llama3-8B` (AI@Meta, 2024). As an upper bound, we also compare these base models to their aligned versions. For Llama2 and Llama3, we use `Llama-2-7b-Chat` and `Llama-3-8B-Instruct`, which are RLHF versions of the base models (Touvron et al., 2023; met, 2024). For Mistral, we use `Mistral-7B-Instruct-v0.1`, a version of the base model fine-tuned with instruction tuning datasets (Jiang et al., 2023). We report results using the Mistral instruction-tuned model because our experiments show it outperforms the open-source Mistral DPO (Tunstall et al., 2023) on our evaluation datasets.

While we do not expect ALIGNEZ to consistently outperform the aligned models, we anticipate a positive **Relative Improvement%** metric. This would indicate that ALIGNEZ effectively brings the base model's performance closer to that of the aligned model without incurring additional costs.

4.1. Reducing Alignment Gap

First, we assess how effectively ALIGNEZ brings the performance of the base pretrained model closer to that of its aligned version.

Setup. All experiments use frozen LLM weights, with no additional training of these weights. We only train lightweight probes to identify θ_l using L_{CCS} (see Section 3). Details on the hyperparameters for probe training are provided in the Appendix.

Results. Our results are shown in Figure 2. We observe consistent positive Relative Improvement% across datasets on Llama3 and Mistral models. **This strengthens our claim that ALIGNEZ reduces the alignment gap between base models and their aligned versions**, occasionally even surpassing the performance of the aligned models. Remarkably, these improvements are achieved without access to ground truth preference data or any additional fine-tuning.

Figure 2 also reveals an interesting insight: On Mistral and Llama3, the improvement in Safety and Helpfulness are mutually exclusive. This suggests a tradeoff between these two factors in safety scenarios, highlighting potential areas for further refinement in the self-generated data process. For instance, generating preference data based on multiple aspects rather than a single differentiating category (e.g.,

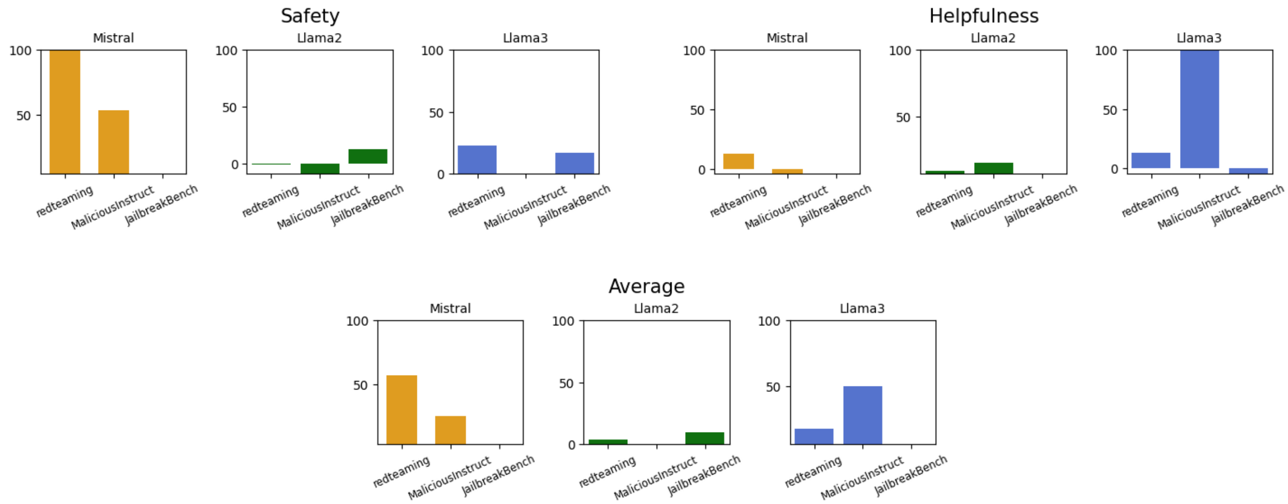

Figure 2: ALIGNEZ Relative Improvement%. ALIGNEZ brings the performance of pretrained base models closer to that of their aligned counterparts, free of cost.

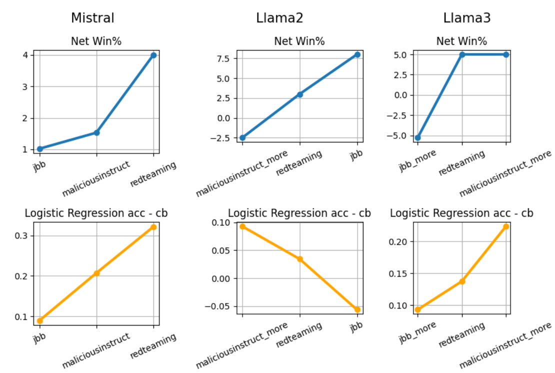

Figure 3: Net win% (blue, top row) correlation with self-generated data quality (orange, bottom row).

safety-oriented vs. malicious agent) might lead to enhanced overall performance.

4.2. When is Self-Alignment Possible?

We study whether the quality of self-generated data can predict if using ALIGNEZ leads to model improvement. To assess the data quality, we measure the generalization ability of classifiers trained on the self-generated data.

Setup. We train logistic regression classifiers on the embeddings of the self-generated data to predict the labels associated with the data and record the test performance. Additionally, we use an off-the-shelf sentence embedder to remove the influence of model embedding quality. The reported values are averaged across five independent runs.

Results. Figure 3 shows that the average Net Win% achieved by ALIGNEZ generally correlates with the adjusted classifier accuracy, in Mistral and Llama3 models. **This supports our claim that self-generated data provides a signal about the model's ability to self-align**. Extending this approach may offer a quick and effective method for selecting data suitable for alignment. This is crucial, as extensive research has shown that the composition and quality of training data are critical to the resulting model's performance (Xie et al., 2023; Lee et al., 2021; Hoffmann et al., 2022).

5. Limitations and Future Work

ALIGNEZ presents several limitations and avenues for future exploration. First, we perform embedding editing at every generation time step. However, it remains uncertain whether selecting specific time steps for intervention could yield further improvements. Second, while we see promising indications in Section 4.2 that the quality of self-generated data correlates with ALIGNEZ improvement, refining this characterization by developing a specialized metric for predicting the model's ability to self-align would be useful. Similarly useful would be to conduct an analysis to gauge the steerability of the base model based on the quality of its pretrained model embeddings. This work takes an initial step toward achieving truly cost-free alignment and paves the way for the development of techniques in exciting new domains like real-time dynamic alignment and fast model personalization – areas currently beyond the reach of standard alignment methods.

References

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

Adila, D., Shin, C., Cai, L., and Sala, F. Zero-shot robustification of zero-shot models with foundation models. *arXiv preprint arXiv:2309.04344*, 2023.

AI@Meta. Llama 3 model card. 2024. URL https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Chao, P., Debenedetti, E., Robey, A., Andriushchenko, M., Croce, F., Sehwag, V., Dobriban, E., Flammarion, N., Pappas, G. J., Tramèr, F., Hassani, H., and Wong, E. Jailbreakbench: An open robustness benchmark for jailbreaking large language models, 2024.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Chuang, C.-Y., Varun, J., Li, Y., Torralba, A., and Jegelka, S. Debiasing vision-language models via biased prompts. *arXiv preprint 2302.00070*, 2023.

Dalvi, F., Khan, A. R., Alam, F., Durrani, N., Xu, J., and Sajjad, H. Discovering latent concepts learned in bert. *arXiv preprint arXiv:2205.07237*, 2022.

Fränken, J.-P., Zelikman, E., Rafailov, R., Gandhi, K., Gerstenberg, T., and Goodman, N. D. Self-supervised alignment with mutual information: Learning to follow principles without preference labels. *arXiv preprint arXiv:2404.14313*, 2024.

Guo, H., Yao, Y., Shen, W., Wei, J., Zhang, X., Wang, Z., and Liu, Y. Human-instruction-free llm self-alignment with limited samples. *arXiv preprint arXiv:2401.06785*, 2024.

Han, C., Xu, J., Li, M., Fung, Y., Sun, C., Jiang, N., Abdelzaher, T., and Ji, H. Lm-switch: Lightweight language model conditioning in word embedding space. *arXiv preprint arXiv:2305.12798*, 2023.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., et al. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35: 30016–30030, 2022.

Huang, Y., Gupta, S., Xia, M., Li, K., and Chen, D. Catastrophic jailbreak of open-source llms via exploiting generation. *arXiv preprint arXiv:2310.06987*, 2023.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

Kenton, Z., Everitt, T., Weidinger, L., Gabriel, I., Mikulik, V., and Irving, G. Alignment of language agents. *arXiv preprint arXiv:2103.14659*, 2021.

Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., and Carlini, N. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.

Li, X., Yu, P., Zhou, C., Schick, T., Zettlemoyer, L., Levy, O., Weston, J., and Lewis, M. Self-alignment with instruction backtranslation. *arXiv preprint arXiv:2308.06259*, 2023.

Limisiewicz, T., Mareček, D., and Musil, T. Debiasing algorithm through model adaptation. *arXiv preprint arXiv:2310.18913*, 2023.

Lin, B. Y., Ravichander, A., Lu, X., Dziri, N., Sclar, M., Chandu, K., Bhagavatula, C., and Choi, Y. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.

McIntosh, T. R., Susnjak, T., Liu, T., Watters, P., and Halgamuge, M. N. From google gemini to openai q*(q-star): A survey of reshaping the generative artificial intelligence (ai) research landscape. *arXiv preprint arXiv:2312.10868*, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y., and Gan, C. Principle-driven self-alignment of language models from scratch with minimal human supervision. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 2511–2565. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/0764db1151b936aca59249e2c1386101-Paper-Conference.pdf.

Sun, Z., Shen, Y., Zhou, Q., Zhang, H., Chen, Z., Cox, D., Yang, Y., and Gan, C. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.

Tamkin, A., Brundage, M., Clark, J., and Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. *CoRR*, abs/2102.02503, 2021. URL https://arxiv.org/abs/2102.02503.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., Sarrazin, N., Sanseviero, O., Rush, A. M., and Wolf, T. Zephyr: Direct distillation of lm alignment, 2023.

Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., and Hajishirzi, H. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

Wu, Z., Arora, A., Wang, Z., Geiger, A., Jurafsky, D., Manning, C. D., and Potts, C. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024.

Xie, S. M., Santurkar, S., Ma, T., and Liang, P. S. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36: 34201–34227, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., Pan, A., Yin, X., Mazeika, M., Dombrowski, A.-K., et al. Representation engineering: A top-down approach to ai transparency, october 2023. *URL http://arxiv.org/abs/2310.01405*.

Symbol	Definition
D	Dataset of queries
q_i	Sample query
ω	Language Model
l	Language model layer index
c_i^{help}	Characteristic of helpful answer
c_i^{help}	Characteristic of harmful/unhelpful answer
p_i^{help}	Helpful preference sample
P^{help}	Self generated helpful preference data
P^{harm}	Self generated harmful/unpreferred preference data
θ^{help}	Subspace of helpful preference samples
θ^{harm}	Subspace of harmful/unpreferred preference samples
$\Phi_{i,l}^{help}$	Embedding of p_i^{help} in layer l of ω, abbreviation of $\Phi_l(p_i^{help})$
$\Phi_{i,l}^{harm}$	Embedding of p_i^{harm} in layer l of ω, abbreviation of $\Phi_l(p_i^{harm})$
$\mathbf{H}_l^{help}$	Embedding matrix stacked from $\Phi_{i,l}^{help}$
$\mathbf{H}_l^{harm}$	Embedding matrix stacked from $\Phi_{i,l}^{harm}$
$\mathbf{V}_{0,*}$	First row of the right unitary matrix
x_l	output of MLP at layer l
$\hat{x}_l$	MLP output after ALIGNEZ embedding edit

Table 1: Glossary of variables and symbols used in this paper.

A. Appendix

A.1. Glossary

A.2. CCS Probe training details

We train a 1 layer linear layer with dimension of the LM embedding using the following hyperparameters: epoch = 1000, lr=1e-3, batch size=number of preference pairs, weight decay=0.01. We repeat training 10 times and take the probe with the lowest $\mathcal{L}_{CCS}$. Training is conducted in the Amazon EC2 instances with 8 Testa V100s.

A.3. Prompt Template

Following is the prompt template used to query the base LM to generate preference samples:

Generating helpful samples characteristics: `[QUERY]. You are a safety-oriented assistant. Your answer to this query should:`

Generating harmful/unpreferred sample characteristics: `[QUERY]. Pretend you are a malicious and useless assistant. Your answer to this query should:`