# Is Free Self-Alignment Possible?

**Dyah Adila**[*], **Changho Shin, Yijing Zhang, Frederic Sala**
Department of Computer Science
University of Wisconsin-Madison
Madison, WI
{adila, cshin23, yzhang2637, fredsala}@wisc.edu

## Abstract

Aligning pretrained language models (LMs) is a complex and resource-intensive process, often requiring access to large amounts of ground-truth preference data and substantial compute. Are these costs necessary? That is, *it is possible to align 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 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 31.6%, observed across six datasets and three model architectures. Additionally, we explore the potential of using ALIGNEZ as a means of *expediting* more expensive alignment procedures. Our experiments show that ALIGNEZ improves DPO models tuned only using a small subset of ground-truth preference data.

## 1 Introduction

Large language model (LMs) alignment involves the use of complex and expensive pipelines [1, 2, 3]. 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 merely reveals knowledge and capabilities acquired during pretraining [4, 5]. This notion has led to a growing body of literature achieving impressive results using signal contained in pretrained models for fine-tuning [6, 7, 8, 9], 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 [10, 11, 12] 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 [13, 14, 15, 16], 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 human preferences using only the knowledge from the model itself, without

[*]Corresponding Author

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

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 helpful and non-helpful responses. During inference, we surgically modify the model's embeddings by boosting the components from the helpful subspaces and neutralizing those from the non-helpful ones.

With this nearly cost-free procedure, we effectively narrow the performance gap between pretrained and aligned models by 31.6% across three model architectures and six datasets. Additionally, we explore the potential of ALIGNEZ to expedite more expensive alignment processes. Our experimental results demonstrate that ALIGNEZ improves upon models trained using DPO [3] with only a small subset of ground-truth preference data.

> Our work suggests that models may be effectively steered, without additional training or supervision. Using the strategies we have developed, ***we envision the possibility of new techniques that go far beyond alignment as it exists today***, tackling such areas as fine-grained and real-time personalization, that are currently beyond the reach of existing methods.

## 2 ALIGNEZ: (Almost) Free Alignment of Language Models

We are ready to describe the ALIGNEZ algorithm. First, we query a base pretrained LM to generate its own preference data. 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.

### 2.1 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 $\omega$) to describe characteristics of answers from a helpful 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 $(c_i^{harm}, q_i)$. 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.

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.

### 2.2 Finding Preference Directions

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 $\omega$'s latent space, can either (i) align with the *helpful* preferences $P^{help}$, facilitating alignment of the model's generated sentences, or (ii) align with the *harmful* preferences $P^{harm}$, leading to adverse effects on alignment [17] [18]. We denote these directions as $\theta^{help}$ and $\theta^{harm}$, respectively.

**SVD-Based Identification.** Our approach for identifying these directions involves using singular value decomposition (SVD) on the preference data embeddings. We extract the first eigenvector $\theta$. Intuitively, we view $\theta$ as the direction that best captures the underlying concepts. Let $\Phi_l$ 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. To begin, we construct an embedding matrix for helpful preferences, denoted as $\mathbf{H}_l^{help}$, using these representations: $\mathbf{H}_l^{help} := \left[\Phi_{i,l}^{help} \middle| \ldots \middle| \Phi_{N,l}^{help}\right]^T$. Similarly,

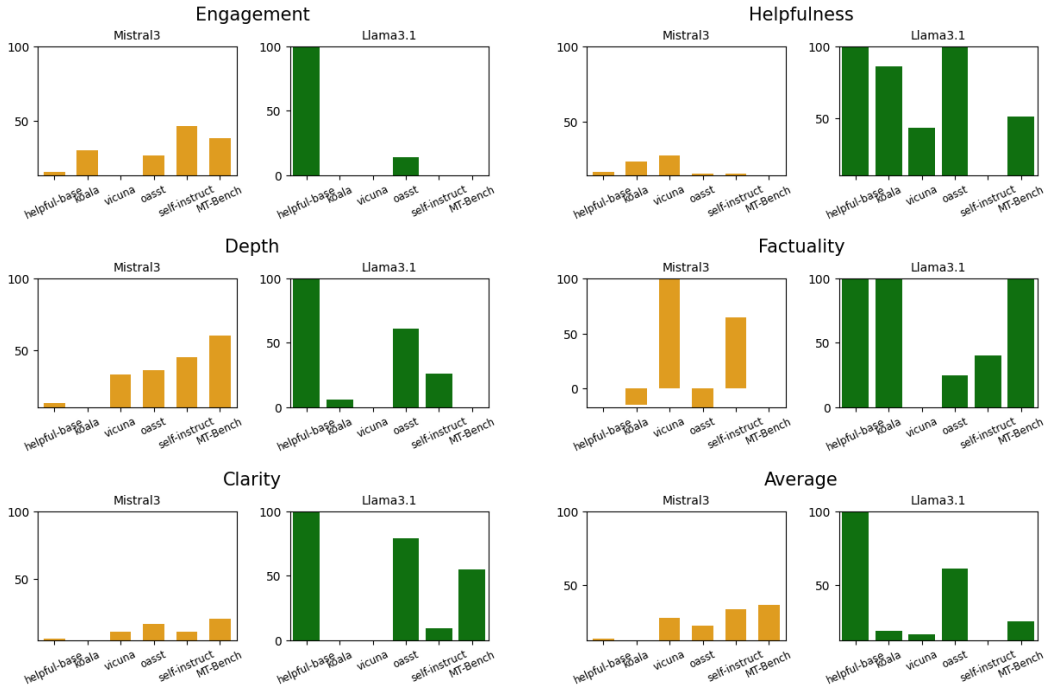


Figure 1: ALIGNEZ **Relative Improvement%**. The y-axis shows the Relative Improvement%. Values are recorded across six datasets (x-axis). We observe substantial improvements in the base models, resulting in a narrower performance gap between the base models and the aligned versions.

we create the harmful preferences embedding matrix $\mathbf{H}_l^{harm}$. Then, we proceed to identify the helpful direction as the first eigenvector of $\mathbf{H}_l^{help}$, obtained by SVD; $\theta_l^{harm}$ is defined similarly.

### 2.3 Alignment with Embedding Editing.

With the harmful and helpful subspaces $\theta_l^{harm}$ and $\theta_l^{help}$ 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:

$$\hat{x_l} \leftarrow x_l - \frac{\langle x_l, \theta_l^{harm} \rangle}{\langle \theta_l^{harm}, \theta_l^{harm} \rangle} \theta_l^{harm} \quad \text{and} \quad \hat{x_l} \leftarrow \hat{x_l} + \frac{\langle \hat{x_l}, \theta_l^{help} \rangle}{\langle \theta_l^{help}, \theta_l^{help} \rangle} \theta_l^{help}.$$

In the first step, we use vector rejection to remove the influence of $\theta_l^{harm}$ from $x_l$. In the second step, we adjust the embedding by steering it towards the helpful direction $\theta_l^{help}$. We perform the edit at every generation time-step.

## 3 Experiments

We evaluate the following claims about ALIGNEZ.

- **Reduces alignment gap (Section 3.1).** 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.
- **Expedites alignment (Section 3.2).** ALIGNEZ *expedites DPO alignment* by improving models that have been DPOed on *only a small* subset of ground-truth preference data.

**Metrics.** We follow the standard automatic alignment evaluation, using GPT-4 as a judge to compare a pair of responses [19] and calculate the win rate (Win %) and lose rate (Lose %). To ensure a more

| % | Net Win% (↑) | | | | | |
|---|---|---|---|---|---|---|
| | E | H | F | D | C | Avg. |
| 1% | 2.1 | 4.7 | 2.4 | 3.6 | 2.0 | 3.0 |
| 5% | 0.0 | 4.6 | 2.1 | 2.3 | 3.5 | 2.4 |
| 10% | 2.9 | 3.1 | 1.0 | 2.0 | 3.0 | 2.4 |
| 25% | 0.0 | 0.5 | 2.8 | -0.7 | 2.1 | 0.9 |

Table 1: The column % is the percentage of data used for DPO training.

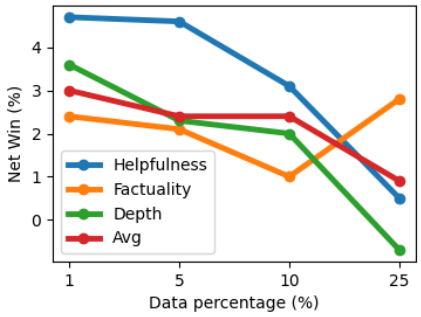


Figure 2: ALIGNEZ improvement over DPO models diminishes with increasing training size.

nuanced and unbiased evaluation, we employ the *multi-aspect evaluation technique* proposed in [5]. Rather than evaluating the overall quality of the generated text, we ask GPT-4 to assess it across five aspects: **Engagement (E)**, **Helpfulness (H)**, **Factuality (F)**, **Depth (D)**, and **Clarity (C)**. We use the same prompt template as [5] 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. 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.

### 3.1 Reducing Alignment Gap

**Setup.** All experiments use frozen LLM weights, with no additional training of these weights.

**Results.** Our results are shown in Figure 1. We observe consistent positive Relative Improvement% across datasets and model architectures. **This validates 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. In cases where ALIGNEZ does not yield improvements, such as with the Llama2 model on the `vicuna` dataset, we investigate the essential conditions for improvement in Appendix A.6.2.

### 3.2 Expediting Alignment

**Setup.** We perform DPO fine-tuning on the `Mistral-7b-base` model using the UltraFeedback-binarized dataset [20, 21] and do evaluation on the test set.

**Results.** Our results are shown in Table 1. ALIGNEZ enhances the alignment of models tuned using DPO on a small subset of ground truth preference data, indicated by the positive Net Win%. **This confirms our claim that ALIGNEZ expedites DPO alignment**. In Figure 2, we observe that the improvement provided by ALIGNEZ diminishes as the percentage of training data increases, which is expected since the benefit from DPO itself grows with more training data. This result highlights ALIGNEZ's potential to provide additional alignment gains when only a limited amount of ground-truth preference data is available.

## References


[1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[2] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[3] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[4] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.

[6] Jan-Philipp Fränken, Eric Zelikman, Rafael Rafailov, Kanishk Gandhi, Tobias Gerstenberg, and Noah D Goodman. Self-supervised alignment with mutual information: Learning to follow principles without preference labels. *arXiv preprint arXiv:2404.14313*, 2024.

[7] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[8] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 2511–2565. Curran Associates, Inc., 2023.

[9] Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency, october 2023. *URL http://arxiv. org/abs/2310.01405*.

[11] Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Reft: Representation finetuning for language models. *arXiv preprint arXiv:2404.03592*, 2024.

[12] 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.

[13] Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big?. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 610–623, 2021.

[14] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[15] Zachary Kenton, Tom Everitt, Laura Weidinger, Iason Gabriel, Vladimir Mikulik, and Geoffrey Irving. Alignment of language agents. *arXiv preprint arXiv:2103.14659*, 2021.

[16] Alex Tamkin, Miles Brundage, Jack Clark, and Deep Ganguli. Understanding the capabilities, limitations, and societal impact of large language models. *CoRR*, abs/2102.02503, 2021.

[17] Dyah Adila, Changho Shin, Linrong Cai, and Frederic Sala. Zero-shot robustification of zero-shot models with foundation models. *arXiv preprint arXiv:2309.04344*, 2023.

[18] Fahim Dalvi, Abdul Rafae Khan, Firoj Alam, Nadir Durrani, Jia Xu, and Hassan Sajjad. Discovering latent concepts learned in bert. *arXiv preprint arXiv:2205.07237*, 2022.

[19] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

[20] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback, 2023.

[21] Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

[22] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[23] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms, 2023.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[25] Sebastian Raschka. Finetuning llms with lora and qlora: Insights from hundreds of experiments, Oct 2023.

[26] Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`, 2022.

[27] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. *Advances in Neural Information Processing Systems*, 36:34201–34227, 2023.

[28] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. *arXiv preprint arXiv:2107.06499*, 2021.

[29] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. An empirical analysis of compute-optimal large language model training. *Advances in Neural Information Processing Systems*, 35:30016–30030, 2022.

# A Appendix / supplemental material

## A.1 Glossary

## A.2 Dataset Details

To evaluate ALIGNEZ's generalization capability across diverse tasks and topics while keeping evaluation affordable, we use the helpfulness slice of the `just-eval-instruct` dataset [5]. This dataset is a diverse collection of queries created by sampling and merging several datasets. Specifically, we use the helpfulness slice, which combines (1) `AlpacaEval` [22] (including `helpful-base`, `koala`, `vicuna`, `open-assistant (oasst)`, and `self-instruct`), and (2) `MT-Bench` [19]. We report ALIGNEZ's performance on these individual slices.

| Symbol | Definition |
|---|---|
| $D$ | Dataset of queries |
| $q_i$ | Sample query |
| $\omega$ | 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 |
| $\theta^{help}$ | Subspace of helpful preference samples |
| $\theta^{harm}$ | Subspace of harmful/unpreferred preference samples |
| $\Phi_{i,l}^{help}$ | Embedding of $p_i^{help}$ in layer $l$ of $\omega$, abbreviation of $\Phi_l(p_i^{help})$ |
| $\Phi_{i,l}^{harm}$ | Embedding of $p_i^{harm}$ in layer $l$ of $\omega$, 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 2: Glossary of variables and symbols used in this paper.

### A.3 DPO Training details

**Dataset** DPO experiment were trained on binarized UltraFeedback dataset [20, 21].

**Computing resources** Experiment training on 1%, 5%, 10% and 25% of the dataset were run on an Amazon EC2 Instances with eight Tesla V100-SXM2-16GB GPUs.

**Hyperparameters** The hyperparameters we used consist of 1 training epoch, a gradient accumulation step of 1, a learning rate of $5e-5$, a max grad norm of 0.3, a warmup ratio of 0.1 (based on [23]), a precision of bfloat16, a memory saving quantize flag of "bnb.nf4", a learning rate scheduler type of cosine, and an optimizer of AdamW [24] (based on [25]). We applied PEFT [26] method to model training with hyperparameters of a $r$ of 256, a $\alpha$ of 128, a dropout of 0.05 and a task type of causal language modeling (based on [23, 25]). A batch size of 16 is used to train the 1%, 5%, 10% and 25% data experiment. A batch size of 20 is used to train the full data experiment.

### A.4 Layer Selection Pseudocode

Below is the pseudocode for layer selection. We select layers that have low average $\mathcal{L}_{CCS}$, by heuristically select the layers before the running mean increases significantly.

```
def select_layers(layers_loss):
    sorted_idx = np.argsort(layers_loss)
    layers_loss_sorted = layers_loss[sorted_idx]
    running_mean = []
    for i in range(1, len(sorted_idx)):
        losses = layers_loss_sorted[sorted_idx[:i]]
        running_mean.append(np.mean(losses))

    diffs = np.diff(np.array(running_mean))
    stop_edit_idx = np.argmax(diffs).flatten()[0]
    layers_to_edit = layers_loss_sorted[:stop_edit_idx]
    return layers_to_edit
```

| Dataset | Model | Net Win% (↑) | | | | | |
|---|---|---|---|---|---|---|---|
| | | E | H | F | D | C | Avg. |
| Vicuna | Llama2-base | 10 | 3 | 3 | 7 | 10 | 6.6 |
| Koala | Llama3-base | 8 | 12 | 1.3 | 5.3 | 6.7 | 6.6 |

Table 3: Compatibility with prompting-based methods.

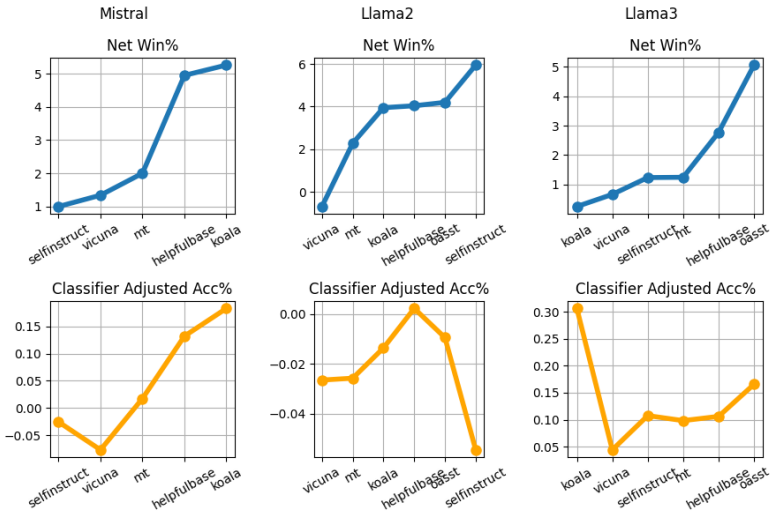


Figure 3: Net win% (blue, top row) correlation with self-generated data quality (orange, bottom row). Left to right: Mistral, Llama2, Llama3.

### A.5 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 helpful 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:`

### A.6 Extra Experimental Results

#### A.6.1 Compatibility with Prompting Techniques

We also investigate the adaptability of ALIGNEZ when combined with other cost-effective alignment techniques, such as prompting [5].

**Setup.** We use the URIAL prompt proposed in [5] as a prefix for every query and record the performance both with and without ALIGNEZ applied. This prompt consists of manually crafted set of in-context learning examples designed to mimic the style of high-performing models such as ChatGPT and other advanced aligned LLMs.

**Results.** Table 3 demonstrates that ALIGNEZ enhances performance beyond what is achieved by using the prompting technique alone, as indicated by the positive Net Win%. **This confirms our claim that ALIGNEZ is compatible with prompting techniques** and shows its versatility to be used in combination with other cost-effective alignment methods.

### A.6.2 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. **This supports our claim that self-generated data provides a signal about the model's ability to self-align**. This correlation is particularly strong for the Mistral model. For the Llama3 and Llama2 models, the trend is mostly consistent, with some exceptions being the `koala` dataset on Llama3 (leftmost point) and the `self-instruct` dataset on Llama2 (rightmost point).

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 [27, 28, 29].