Toward Preference-aligned Large Language Models via Residual-based Model Steering

Lucio La Cava , Andrea Tagarelli
DIMES Dept., University of Calabria, Italy
{lucio.lacava, tagarelli}@dimes.unical.it

Abstract

Preference alignment is a critical step in making Large Language Models (LLMs) useful and aligned with (human) preferences. Existing approaches such as Reinforcement Learning from Human Feedback or Direct Preference Optimization typically require curated data and expensive optimization over billions of parameters, and eventually lead to persistent task-specific models. In this work, we introduce Preference alignment of Large Language Models via Residual Steering (PALRS), a training-free method that exploits preference signals encoded in the residual streams of LLMs. From as few as one hundred preference pairs, PALRS extracts lightweight, plug-and-play steering vectors that can be applied at inference time to push models toward preferred behaviors. We evaluate PALRS on various small-to-medium-scale open-source LLMs, showing that PALRS-aligned models achieve consistent gains on mathematical reasoning and code generation benchmarks while preserving baseline general-purpose performance. Moreover, when compared to models aligned with DPO and SimPO, they perform better with great time-savings. Our findings highlight that PALRS offers an effective, much more efficient and flexible alternative to standard preference optimization pipelines, offering a training-free, plug-and-play mechanism for alignment with minimal data.

Extended version with Suppl. Mat. is available at https://doi.org/10.48550/arXiv.2509.23982

1 Introduction

Large Language Models (LLMs) have rapidly advanced the state-of-the-art performance across various domains, including dialogue, programming, and mathematical tasks [Li *et al.*, 2025]. While most capabilities in such systems are due to rich and wide pretraining [Chen *et al.*, 2024; Kirchenbauer *et al.*, 2024; Shaib *et al.*, 2024; Wang *et al.*, 2025c], a key determinant in their usability is how close their outputs align with *human preferences* [Wang *et al.*, 2023; Shen *et al.*, 2023]. Indeed, preference alignment has emerged in recent years as a focal stage in the LLM deployment pipeline: approaches such as reinforcement learning from human feedback [Ouyang *et al.*, 2022; Bai *et al.*, 2022], direct preference optimization (DPO) [Rafailov *et al.*, 2023], or simple preference optimization (SimPO) [Meng *et al.*, 2024] have become standard practices for eliciting better capabilities from LLMs.

Despite their tangible effects, preference-optimization alignment methods remain costly and inflexible. First, current approaches rely on large volumes of curated preference datasets [Köpf *et al.*, 2023], thus being highly *annotation intensive*. Second, despite parameter-efficient methods such as LoRA adapters, aligning a model remains *computationally intensive* because it requires repeated forward and backward passes through large models, often consuming several GPU-hours [Stiennon *et al.*, 2020]. Third, alignment is typically considered *persistent*: once an LLM has been fine-tuned toward a particular preference setting, adapting it to a different set of preferences generally requires starting again from the base model to produce new checkpoints. Maintaining multiple preference-specific checkpoints can quickly become resource-intensive. These challenges underscore the need to scale alignment methods across three dimensions: efficiency (reducing computational cost), effectiveness (maximizing alignment quality), and flexibility (enabling rapid adaptation to new preference specifications).

Recently, an emerging line of research has unveiled that *residual stream activations* of LLMs encode contextually rich and linearizable features that can be used to manipulate the model behavior surgically, yet without altering their weights or requiring any additional training [Zou *et al.*, 2023; Zhang and Nanda, 2024]. Residual-based interventions have been shown effective to mitigate refusal behaviors [Arditi *et al.*, 2024; Wang *et al.*, 2025b], erase concepts [Belrose *et al.*, 2023], induce desired persona-like behaviors [Chen *et al.*, 2025], or improve factfulness [Li *et al.*, 2023]. These results suggest a promising direction, with residuals functioning as "control knobs," providing a relatively inexpensive yet effective mechanism for steering model behavior at inference time. However, prior work has focused on optimizing narrow behaviors (e.g., refusal mitigation), leaving the broader challenge of using residual interventions to align models with a range of preferences underexplored.

Our Hypothesis. Consider the task of improving a model's mathematical reasoning or coding capabilities. Standard practices, based on preference optimization alignment, would

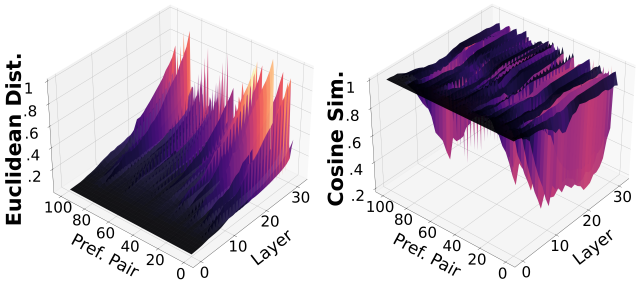

Figure 1: Normalized Euclidean distance and cosine similarity between residual stream activations (from Llama 3.1 8B Instruct) of 100 randomly-sampled chosen-rejected response pairs to math questions, for a fixed token position (cf. Sect. 3.2) and by varying the model layer ids—higher ids are closer to the output layer. Lighter colors denote higher Euclidean distance and lower cosine similarity.

require curating thousands of preference pairs, optimizing the model's weights, and deploying task-specific checkpoints, resulting in a resource-intensive and inflexible process.

In this work, we adopt a different approach, as we hypothesize that the difference between the residual stream activations of chosen and rejected responses—to questions grounded in a particular domain, e.g., math or coding—can be meaningful. If so, we could distill these differences into steering directions that can be used to induce *preferred behaviors* in a model via lightweight inference-time interventions.

To support our hypothesis, consider Fig. 1, which shows Euclidean distance and cosine similarity between residual stream activations of preference pairs (i.e., chosen-rejected responses) to math questions. It is worth noticing that the Euclidean distances and cosine similarities can substantially vary across the pairs. In particular, there is a significant portion of pairs for which the Euclidean distance resp. cosine similarity increases resp. decreases with mid-high layer ids. This implies the possibility of defining a vector, or steering direction, that moves the activation from the rejected toward the chosen response: if the difference were tiny, a steering vector would have little effect; by contrast, large distances suggest the difference is substantial enough to guide model behavior adjustment. In addition, the evidence of cases with low cosine similarity indicates that the differences between those pairs' activations are mostly consistent in direction, thus allowing a generalizable aggregated steering vector to be distilled, rather than needing a separate vector for each example.

Preference alignment of Large Language Models via Residual Steering (PALRS). Building on the above remarks, we introduce a novel approach to preference alignment of LLMs through steering with residual stream activations, dubbed PALRS. Instead of updating model weights via gradient optimization, PALRS computes preference directions based on differences in residual activations extracted from a small set of preference pairs (on the order of 100). These directions are then applied at inference time, enabling lightweight, plug-and-play preferred-behavior steering.

To the best of our knowledge, this is the first study to leverage residual stream directions for preference alignment. Our contributions are threefold:

- We bring the model steering framework based on residual stream activations to the setting of preference alignment of LLMs, showing that residual stream activations encode preference information in a linearly accessible form.
- We introduce a simple difference-in-means approach for estimating candidate preference directions from a small set of chosen–rejected response pairs, without requiring any LLM post-training. We further propose a principled criterion for selecting the steering direction that best aligns an LLM's behavior with target preferences.
- We demonstrate the effectiveness of PALRS through different small-to-medium scale open-source LLMs, with testing on widely used benchmarks. Particularly, as a concrete case study, we show that steering directions derived from preference-alignment datasets conceived for *math reasoning* (GSM8K) and *code generation* (HumanEval), under certain conditions of the steering intensity, lead an LLM to improve performance on corresponding benchmarks, without degrading results on other-domain benchmarks. Additionally, we highlight that PALRS-aligned models take substantial advantage w.r.t. DPO- and SimPO-aligned models on both GSM8K and HumanEval, achieving superior effectiveness while requiring much less computational overhead.

Impact of our research. A key advantage of our method lies in the efficiency afforded by its training-free strategy. PALRS requires only a single forward pass over few (e.g., 100) data samples to create the steering vector, which accounts for its substantial computational speedups. In contrast, standard preference-optimization alignment methods rely on full fine-tuning procedures, where data is a critical bottleneck and (tens of) thousands of paired samples are typically required. As a result, PALRS is markedly data-efficient, making it particularly attractive for low-resource scenarios, including sensitive domains such as medical applications.

The flexibility of PALRS is another notable strength. The approach is not restricted to specific domains such as mathematical reasoning or code generation. Because PALRS operates via task-specific steering vectors rather than permanent model modifications, multiple vectors (e.g., for math, coding, or other tasks) can coexist and be applied as needed. This plug-and-play generality highlights the potential of PALRS as a broadly applicable framework for preference alignment.

2 Related Work

Model Steering via Feature Directions. Recent studies suggest that linear directions in the activation space of LLMs capture richer and more generalizable features than individual neurons [Li *et al.*, 2021; Elhage *et al.*, 2022]. This shift toward subspace-level aspects has motivated research that exploits linear representations to probe model internals and provide better interpretation and steering of their behaviors [Hernandez and Andreas, 2021; Nanda *et al.*, 2023; Park *et al.*, 2024; Bayat *et al.*, 2025]. A key challenge is to reliably identify such feature directions. In this regard, unsupervised approaches based on Sparse Auto-Encoders (SAE) have been used to uncover latent, interpretable features [Huben *et al.*, 2024; Lan *et al.*, 2024; Shu *et al.*, 2025]. A complementary trend involves exploiting contrastive pairs of texts

differing across a specific axis to extract directions that isolate targeted behaviors [Burns *et al.*, 2023]. Once extracted, these features can be used to intervene on model activations, particularly in the residual stream, where editing is known to effectively steer the models' behavior [Zou *et al.*, 2023; Zhang and Nanda, 2024; Wang *et al.*, 2025a]. Such interventions have been shown promising in shifting sentiment and detoxification [Turner *et al.*, 2023], enhancing truthfulness [Li *et al.*, 2023], erasing concepts [Belrose *et al.*, 2023], targeting refusal behaviors [Arditi *et al.*, 2024; Wang *et al.*, 2025b], and controlling character traits [Chen *et al.*, 2025].

Preference Optimization in LLMs. A large body of work has focused on aligning LLMs with human preferences, aiming to render such tools more usable [Wang *et al.*, 2023; Shen *et al.*, 2023]. This has been largely driven by Reinforcement Learning from Human Feedback (RLHF) approaches [Ouyang *et al.*, 2022; Bai *et al.*, 2022], and more efficient methods like Direct Preference Optimization (DPO) [Rafailov *et al.*, 2023]. Nonetheless, despite recent efforts to further improve efficiency [Hong *et al.*, 2024; Meng *et al.*, 2024], the promising and more efficient alternative of steering models toward preferences via residual streams remained almost unexplored.

In a related effort, recent studies [Liu *et al.*, 2024; Rimsky *et al.*, 2024; Cao *et al.*, 2024] explore preference alignment by identifying disparities in activation patterns elicited by preferred vs. dispreferred stimuli. While these represent an important step toward bridging preference data with internal model representations, their approaches differ fundamentally from ours. [Liu *et al.*, 2024], resp. [Cao *et al.*, 2024] requires training with contrastive stimuli to extract the relevant signals, resp. optimizing the bi-directional objective, whereas our method is entirely *training-free*. Also, [Liu *et al.*, 2024] introduces a low-rank adaptation module to perform steering, in contrast to our *inference-time intervention*. In addition, [Rimsky *et al.*, 2024] mostly focuses on contrastive prompt pairs with multiple-choice questions. Overall, the above design choices incur limitations in practice, with more limited steering effects, less generalizable preferences, and less scalable and lightweight improvements compared to PALRS.

3 Methodology

3.1 Preliminaries

Notation. Throughout this paper, we will use capital letters to denote data objects, lowercase letters to denote scalars, and bold lowercase letters to denote vectors.

We are given a collection of text triplets $\langle Q, A^{(+)}, A^{(-)} \rangle$ where Q is a question, and As are two possible (human-provided) answers to Q. Based upon this collection, we define the dataset $\mathcal{D} = \{\langle Q, A^{(+)}, A^{(-)} \rangle \mid A^{(+)} \succ_Q A^{(-)}\}$, where $A^{(+)} \succ_Q A^{(-)}$ denotes that, for question Q, $A^{(+)}$ is preferred over $A^{(-)}$. Let also $\mathbf{t}^{(+)}$, $\mathbf{t}^{(-)}$, and $\mathbf{t}^{(Q)}$ denote the input sequences of tokens from an answer $A^{(+)}$, $A^{(-)}$ and question Q, respectively.

Residual stream activations. Following previous studies [Zhang and Nanda, 2024], we resort to the concept of *residual stream activation*, specifically in the context of the Transformer decoder architecture, with L layers and hidden size d. Given a sequence $\mathbf{t}$, the state of knowledge a model has about a token in position i at the start of layer ℓ (i.e., before layer ℓ processes it) can be expressed by the token's residual stream activation, given the input tokens up to position i and all contributions computed from the layers preceding ℓ. We will denote it as a real-valued d-dimensional vector $\mathbf{x}_{i,\ell}(\mathbf{t})$, or simply with $\mathbf{x}_{i,\ell}$ if $\mathbf{t}$ is clear from the context.

In other words, the token's residual stream activation is the accumulated hidden representation of the token as it flows through the model, thus encoding all contextual information that the model has built up so far—which includes the initial token and positional embeddings, plus the contributions from the multi-head causal self-attention and feed-forward MLP components (sublayers) of all previous layers. This also means that the residual stream is linearly interpretable, since sublayer outputs are added to the residual stream, and is tied to the model's prediction distribution, since at the final layer the model projects the last residual stream through the embedding-to-token matrix to get the next-token logits.

Model steering. The residual stream activation can be used for model steering because it is the central state vector that encodes all of the model's knowledge about a token at a given point.

One effective way to reliably alter the model's outputs and behavior is *activation addition*, i.e., adding an identified direction $\mathbf{r} \in \mathbb{R}^d$ in the residual space that corresponds to some feature, e.g., tokens that are more representative of the chosen responses than the rejected ones. Therefore, shifting the residual stream along that direction will change the model's predictions accordingly.

An opposite approach is *directional ablation*, which consists in subtracting from the residual stream its orthogonal projection onto the direction $\mathbf{r}$. However, as noted in [Arditi *et al.*, 2024], the two approaches have a different impact as the directional ablation applies to all layers and token positions; by contrast, the activation addition involves only a desired layer (and applies across all token positions). While ablation has been proven effective in refusal-removal setups [Arditi *et al.*, 2024; Wang *et al.*, 2025b], it is not well-suited to our setting: indeed, chosen and rejected responses typically differ only by a few key tokens, thus ablating their direction is very likely to disrupt the model's behavior.

3.2 Extracting Preference Directions

To extract the candidate *preference directions* from the model's residual stream activations, we resort to the *difference-in-means* approach, which is proven to be worst-case optimal [Belrose, 2023], i.e., no linear method can achieve lower worst-case error in recovering a linearly encoded concept direction, and has shown to be effective in previous work [Arditi *et al.*, 2024; Tigges *et al.*, 2024; Wang *et al.*, 2025b].

Given the dataset $\mathcal{D} = \{\langle Q, A^{(+)}, A^{(-)} \rangle \mid A^{(+)} \succ_Q A^{(-)}\}$ of triplets of questions and chosen-rejected responses, we first compute two quantities for any choice of token position i and layer ℓ, which correspond to the average of the residual

Model	Template
Llama	<|begin_of_text|> <|start_header_id|>user <|end_header_id|> *instruction* <|eot_id|><|start_header_id|> assistant<|end_header_id|>
Mistral	<s>[INST] *instruction* [/INST]</s>
OLMo	<|endoftext|><|user|> *instruction* <|assistant|>

Table 1: Single-turn chat templates for the model families considered in this study. Blue-colored text indicates *post-instruction tokens*. Note that a single special string may be decomposed into multiple token IDs by the model's tokenizer, and for the sake of readability non-visible tokens (e.g., newlines) have been omitted.

stream activations produced by the model when it receives in input the chosen, resp. rejected, responses:

$$\boldsymbol{\mu}_{i,\ell}^{(+)} = \frac{1}{|\mathcal{D}|} \sum_{A^{(+)} \in \mathcal{D}} \mathbf{x}_{i,\ell}(\mathbf{t}^{(+)})$$
$$\boldsymbol{\mu}_{i,\ell}^{(-)} = \frac{1}{|\mathcal{D}|} \sum_{A^{(-)} \in \mathcal{D}} \mathbf{x}_{i,\ell}(\mathbf{t}^{(-)}). \quad (1)$$

We define the *candidate preference direction* for a given token position i and layer ℓ as the difference between the two means as follows:

$$\mathbf{r}_{i,\ell} = \boldsymbol{\mu}_{i,\ell}^{(+)} - \boldsymbol{\mu}_{i,\ell}^{(-)}. \quad (2)$$

It is worth noticing that, to prevent diluting the residual stream signals, we focus on the chosen and rejected responses while discarding the instruction Q from the computation of directions. Indeed, we are interested in discerning the signals that differentiate the chosen tokens from the rejected ones, regardless of a particular prompt.

Position and layer selection strategies. Following [Arditi *et al.*, 2024], we consider only token positions corresponding to *post-instruction tokens*, i.e., the template tokens following the instruction, ensuring the model processed the given text and starts producing its output (Table 1).

Regarding the selection of layers, we emphasize the importance of focusing on mid-to-late layers for model steering. The rationale is that early layers primarily shape broad syntactic and structural features, well before semantics, knowledge retrieval, and reasoning processes emerge in the mid layers; also, late layers and especially the unembedding layer directly influence the logits, which bias outputs without meaningfully altering intermediate processing steps.

3.3 Selecting the Steering Direction

Our goal is to choose the vector that can effectively steer the model toward the desired preference-aligned behavior, i.e., favoring human-based preferences in generating responses, while avoiding disruption of its general capabilities.

First, we consider the average signal of residual stream activations induced by the chosen responses from $\mathcal{D}$. Specifically, given layer ℓ, we compute the mean residual stream activation at each selected token position, then we take the average across all such positions. We denote the result as $\boldsymbol{\mu}_{\ell}^{(+)}$.

Let us denote with $\mathcal{C} = \{(i, \ell)\}$ the set of pairs (token-position, layer id) that are selected as a result of the previous stage of position and layer selection. We aim to select the steering direction by finding the preference vector $\mathbf{r}_{i,\ell}$ (with $(i, \ell) \in \mathcal{C}$) that is most strictly aligned with $\boldsymbol{\mu}_{\ell}^{(+)}$, that is, *the vector $\mathbf{r}_{i,\ell}$ that preserves the direction of $\boldsymbol{\mu}_{\ell}^{(+)}$ exactly, while boosting its magnitude.*

This can simply be accomplished by finding the vector projection onto $\boldsymbol{\mu}_{\ell}^{(+)}$ of a $\mathbf{r}_{i,\ell}$ with the maximum magnitude *and* along the direction of $\boldsymbol{\mu}_{\ell}^{(+)}$, not backwards. Therefore, by restricting the search to $\mathbf{r}_{i,\ell} \cdot \boldsymbol{\mu}_{\ell}^{(+)} > 0$, we find:

$$\begin{aligned} \mathbf{r}^* &= \arg\max_{(i,\ell)\in\mathcal{C}} \left\| \left(\mathbf{r}_{i,\ell} \cdot \frac{\boldsymbol{\mu}_{\ell}^{(+)}}{\|\boldsymbol{\mu}_{\ell}^{(+)}\|} \right) \frac{\boldsymbol{\mu}_{\ell}^{(+)}}{\|\boldsymbol{\mu}_{\ell}^{(+)}\|} \right\| \\ &= \arg\max_{(i,\ell)\in\mathcal{C}} \mathbf{r}_{i,\ell} \cdot \boldsymbol{\mu}_{\ell}^{(+)}. \end{aligned} \quad (3)$$

Once we have best aligned a preference direction with the direction of the mean activations of chosen responses, we rescale $\mathbf{r}^*$ so that its norm matches $||\boldsymbol{\mu}_{\ell^*}^{(+)}||$, where ℓ^* denotes the layer corresponding to one of the selected $\mathbf{r}^*$:

$$\hat{\mathbf{r}}^* = \frac{||\boldsymbol{\mu}_{\ell^*}^{(+)}||}{||\mathbf{r}^*||} \cdot \mathbf{r}^*. \quad (4)$$

The above transformation makes the two vectors comparable on the same scale, which allows us to better control the effect of the multiplicative factor in the activation addition step, as described next.

3.4 Applying the Steering Direction

The selected preference direction $\mathbf{r}^*$ is eventually used to steer the model toward preference-aligned behavior. As discussed in Sect. 3.1, the model steering is performed through activation addition, which means that the selected preference direction is added to the residual stream activations of any newly generated response by the model. Specifically, given an input token sequence $\mathbf{t}$ and by denoting with $\mathbf{x}_{\ell^*}$ the residual stream activations at any token position and at layer ℓ^*, the steered residuals are defined as follows:

$$\mathbf{x}'_{\ell^*} := \mathbf{x}_{\ell^*}(\mathbf{t}) + \alpha \hat{\mathbf{r}}^*, \quad (5)$$

where $\alpha \in \mathbb{R}^+$ is a coefficient that controls the strength of the steering effect. It should be emphasized that α is strictly positive by design, since the model steering towards preferred behaviors is conceived to be accomplished through activation addition, rather than ablation—*we aim to encourage desired outcomes, not remove undesired outcomes.* Moreover, our empirical evidence indicates that α requires tuning within a narrow range for practical application.

Overview. An illustrative summary of the methodological steps of PALRS is shown in Figure 2.

4 Experimental Setup

Preference datasets. To demonstrate our proposed PALRS approach, we chose to focus on two particularly informative

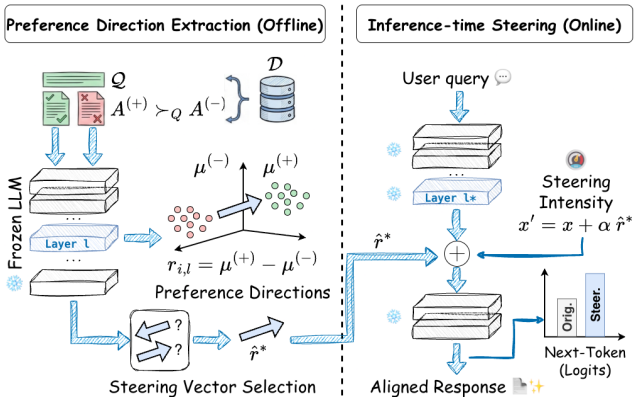

Figure 2: Overview of the main flows and data modules in PALRS.

testbeds for alignment, namely **mathematical reasoning** and **coding**. Compared to broad, general-purpose tasks like commonsense reasoning, math reasoning and coding demand precise logical reasoning and adherence to rules—small alignment errors can directly break correctness. In summary, *our choice for math reasoning and coding is motivated by an intrinsic objectivity in the assessment of the desired outcomes*, as the models' responses can usually be evaluated unambiguously as correct or incorrect. Within this view, we used the argilla/distilabel-math-preference-dpo dataset, which provides chosen/rejected pairs grounded in mathematical correctness, and inclusionAI/Ling-Coder-DPO, whose preferences aim at improving correctness in coding generation.

From each dataset, we randomly sampled 100 triplets to construct the collection $\mathcal{D}$, which we use to compute residual stream activations of chosen and rejected responses. Note that the decision to sample a relatively small number of instances is deliberate, as this is sufficient to capture the difference-in-means signal, in line with findings from related work [Arditi *et al.*, 2024; Wang *et al.*, 2025b]. In addition, unlike prior work using residual vectors for refusal ablation [Arditi *et al.*, 2024; Wang *et al.*, 2025b] or personality steering [Chen *et al.*, 2025], our approach does not rely on external cues for sample selection, such as evaluation scores, targeted refusal tokens, or LLMs-as-judges.

Evaluation goals and benchmarks. We define the following evaluation goals:

• **(E1)** – Assessing the performance of PALRS-aligned models against baseline models (i.e., not steered) using two well-established benchmarks for the target tasks: GSM8K [Cobbe *et al.*, 2021] for mathematical reasoning and HumanEval [Chen *et al.*, 2021] for code generation.

• **(E2)** – Assessing the performance of PALRS-aligned models on tasks outside the target domains, using five widely adopted benchmarks: ARC-Challenge [Bhakthavatsalam *et al.*, 2021] for scientific and commonsense reasoning, HellaSwag [Zellers *et al.*, 2019] for physical and social commonsense inference, MMLU [Hendrycks *et al.*, 2021] for broad knowledge and multi-task understanding, TruthfulQA [Lin *et al.*, 2022] for factuality and truthfulness, and WinoGrande [Sakaguchi *et al.*, 2020] for coreference resolution and pronoun disambiguation. By treating these benchmarks as *guardrails*, this evaluation aims to measure whether, and to what extent, PALRS-aligned models for math or coding steering are able to preserve general-purpose capabilities of baseline models.

Model	Param.	Post-training Strategy	No. Layers
Llama 1B	1B	SFT + RLHF	16
Llama 3B	3B	SFT + RLHF	28
Llama 8B	8B	SFT + RLHF	32
Mistral	7B	SFT	32
OLMo	7B	SFT + DPO + RLVR	32

Table 2: Summary of used LLMs. Post-training strategies are abbreviated as SFT: Supervised Fine-Tuning, RLHF: Reinforcement Learning From Human Feedback, RLVR: Reinforcement Learning with Verifiable Reward, DPO: Direct Preference Optimization.

• **(E3)** – Comparing PALRS-aligned models with DPO-aligned models and SimPO-aligned models on the target tasks, and evaluating their performance on the benchmarks as well as their efficiency. To avoid disadvantaging DPO and SimPO, which are known to be more data-intensive, we do not restrict them to the same 100 preference pairs used by PALRS. In addition to these 100 pairs, we also experiment with 1,000 and 10,000 preference pairs—sampled incrementally and using the same seeds as PALRS—for these methods, ensuring a fairer and more informative comparison across alignment approaches.

• **(E4)** – Assessing the parameter sensitivity of the steering coefficient and its impact on PALRS in the target tasks.

To ensure reliability and reproducibility in the evaluation of models, we used the well-established Language Model Evaluation Harness framework [Gao and et al., 2024] via its TinyBenchmarks tasks [Polo *et al.*, 2024], which are a curated selection of samples from the aforementioned benchmarks that ensure the same evaluation robustness as the full one, at a reduced temporal cost. Note that [Polo *et al.*, 2024] quantify the average estimation error to be up to 2%. Accordingly, any performance gain consistently exceeding this threshold is regarded as a significant model-improvement, rather than biased by sampling variance. We will report performance results that correspond to *exact_match* for GSM8K, *pass@1* for HumanEval, and *accuracy* for the remaining benchmarks.

Models. We experimented with LLMs differing in family, post-training strategy, and parameter size, as summarized in Table 2. These include Llama3 [Dubey *et al.*, 2024] in its 1B, 3B, and 8B variants, Mistral 7B [Jiang *et al.*, 2023], and OLMo2 7B [OLMo *et al.*, 2025]. This variety enables us to assess to some extent the impact of model architectures, post-training approaches, and sizes on the steering behavior induced by PALRS. As mentioned in Sect. 3.2, we inspected mid-to-late layers in each of the models. Specifically, we selected a range $[0.3L..0.9L]$, where L is the number of layers as reported in Table 2.

5 Results

Performance of PALRS on target tasks (E1). Table 3 shows the benchmark results, where PALRS-aligned models correspond to the best settings (cf. *Suppl. Mat. B* for

	Tasks	Targets ↑		Guardrails ↑				
	Variant	GSM8K	HumanE.	ARC-C	HellaS.	MMLU	TruthQA	WinoG.
Llama 1B	Baseline	0.34	0.38	0.44	0.55	0.43	0.43	0.57
	PALRS$_{Math}$	**0.41** (**+20.10%**)	0.41 (+7.89%)	0.42 (-3.68%)	0.54 (-1.03%)	0.44 (+2.39%)	0.42 (-2.89%)	0.58 (+0.90%)
	PALRS$_{Code}$	0.31 (-9.90%)	**0.48** (**+26.32%**)	0.41 (-6.70%)	0.56 (+1.95%)	0.45 (+4.32%)	0.44 (+3.64%)	0.56 (-1.47%)
Llama 3B	Baseline	0.62	0.60	0.57	0.78	0.63	0.48	0.61
	PALRS$_{Math}$	**0.70** (**+13.53%**)	0.56 (-6.67%)	0.54 (-5.54%)	0.73 (-5.60%)	0.62 (-0.85%)	0.45 (-5.61%)	0.62 (+1.61%)
	PALRS$_{Code}$	0.57 (-7.72%)	**0.67** (**+11.67%**)	0.55 (-3.66%)	0.74 (-5.19%)	0.61 (-2.89%)	0.48 (-0.39%)	0.64 (+4.52%)
Llama 8B	Baseline	0.72	0.78	0.65	0.81	0.63	0.54	0.75
	PALRS$_{Math}$	**0.82** (**+13.09%**)	0.77 (-1.28%)	0.63 (-3.32%)	0.81 (+0.32%)	0.63 (-0.12%)	0.55 (+1.62%)	0.73 (-1.71%)
	PALRS$_{Code}$	0.76 (+4.71%)	**0.80** (**+2.56%**)	0.65 (-0.52%)	0.81 (+0.00%)	0.63 (-0.58%)	0.54 (+0.18%)	0.75 (+0.50%)
Mistral	Baseline	0.45	0.15	0.64	0.84	0.64	0.61	0.76
	PALRS$_{Math}$	**0.53** (**+18.42%**)	0.15 (+0.00%)	0.65 (+1.64%)	0.84 (-0.05%)	0.64 (-0.19%)	0.61 (-0.09%)	0.76 (+0.00%)
	PALRS$_{Code}$	0.47 (+3.84%)	**0.23** (**+53.33%**)	0.65 (+0.95%)	0.84 (+0.02%)	0.64 (-0.10%)	0.61 (-1.46%)	0.75 (-1.31%)
OLMo	Baseline	0.75	0.52	0.66	0.83	0.62	0.56	0.76
	PALRS$_{Math}$	**0.77** (**+3.71%**)	0.53 (+1.92%)	0.65 (-1.19%)	0.83 (-0.18%)	0.61 (-1.51%)	0.54 (-2.43%)	0.77 (+0.72%)
	PALRS$_{Code}$	0.78 (+4.88%)	**0.60** (**+15.38%**)	0.66 (+0.26%)	0.83 (+0.17%)	0.62 (+0.00%)	0.55 (-1.71%)	0.75 (-1.28%)

Table 3: Benchmark performance results: Baseline vs. PALRS-aligned models. Bold values correspond to PALRS-aligned model performances on the target tasks. Values in parenthesis indicate percentage change of a PALRS-aligned model w.r.t. the baseline performance on the same task.

details). Looking at the two 'Targets' columns in the table, PALRS$_{Math}$ boosts the math-related task (GSM8K) across all models, with an average improvement over the baseline around +14% (from +3.7% with OLMo up to +20.1% with Llama 1B). Analogously, PALRS$_{Code}$ consistently improves the code-related task (HumanEval) across all models, with an average improvement over the baseline around +22% (from +2.6% with Llama 8B up to +53.3% with Mistral).

Performance of PALRS on guardrail tasks (E2). Valuable insights also arise from performance on the guardrail benchmarks (right-most five columns in Table 3). PALRS-aligned Llama 1B, Llama 8B, and OLMo are fairly stable, with average percentage changes on guardrail benchmarks around -1% or less: for PALRS$_{Math}$ resp. PALRS$_{Code}$, -0.86% resp. +0.35% with Llama 1B, -0.64% resp. -0.08% with Llama 8B, and -0.73% resp. -0.51% with OLMo. PALRS$_{Math}$ with Mistral even slightly improves on average over the guardrails (+0.26%), while keeping average guardrail degradation around -0.38% when using PALRS$_{Code}$. By contrast, Llama 3B is the only to suffer the most guardrail degradation, up to -3.19% with PALRS$_{Math}$.

Regarding the model types, Mistral has the largest boost by PALRS$_{Code}$ (+53.4%) and second-largest by PALRS$_{Math}$ (+18.4%). Using Llama models has shown a scaling effect, since smaller models lead to relatively larger gains but also stronger impact on guardrail tasks. PALRS-aligned OLMo models have the least improvement (+3.7%) over the baseline

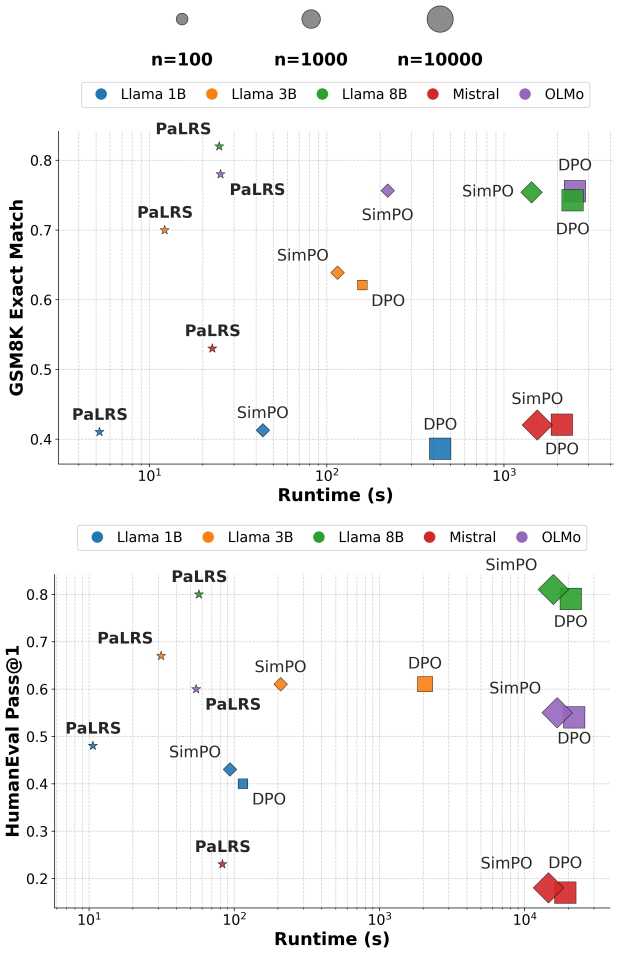

Figure 3: Best-setting benchmark performance and efficiency comparison of PALRS-aligned models vs. DPO-aligned and SimPO-aligned models, for different numbers (n) of preference pairs: GSM8K (top) and HumanEval (bottom). Colors denote models, while stars, resp. circles and squares, denote PALRS-alignment, resp. other alignments. Marker size increases with n.

on the math task, but +15.4% on coding task.

Remarkably, PALRS alignment of the largest models, i.e., OLMo, Mistral and Llama 8B, on one target task reveals little degradation or, more often, improvement over the other target task, suggesting that steering in particular on a math task can have a beneficial effect on a coding task as well.

Comparison of PALRS with DPO and SimPO (E3). Figure 3 compares the best-setting PALRS-aligned models with DPO- and SimPO-aligned models, in terms of effectiveness and efficiency on the GSM8K and HumanEval benchmarks, by varying model size and number n of preference pairs for the alignment. DPO and SimPO based values correspond to the best benchmark scores across various alignment-data sizes n—details are reported in *Suppl. Mat. C*—while for PALRS n is kept fixed to 100 (Sect. 4).

On GSM8K, PALRS$_{Math}$-aligned models consistently surpass the corresponding DPO- and SimPO-aligned models, across model types (the sole exception is Llama 1B, for which PALRS$_{Math}$ ties with SimPO) with performance

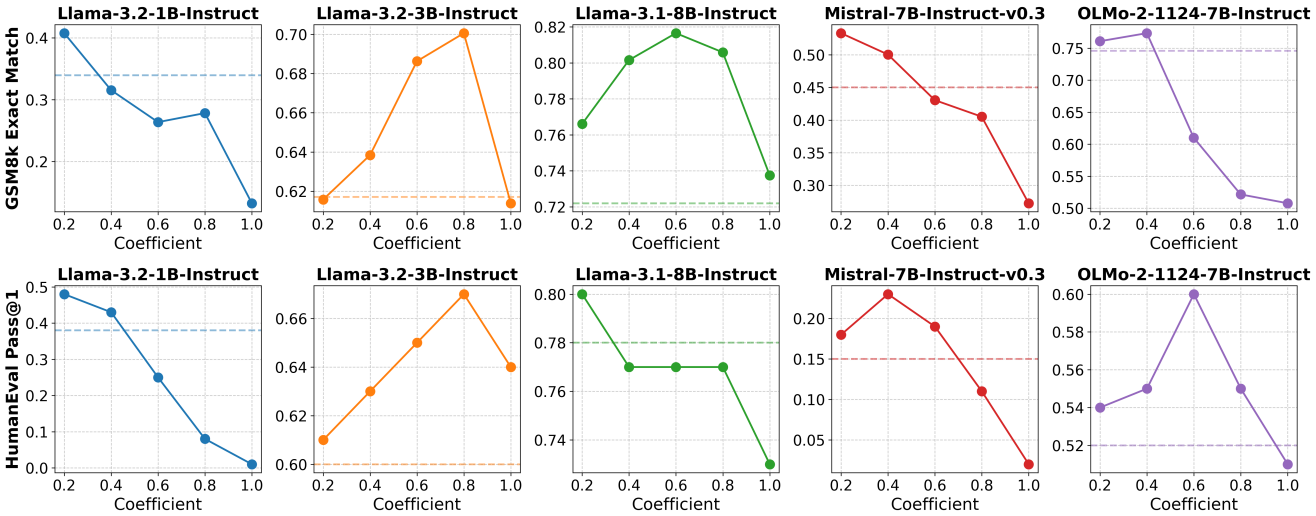

Figure 4: Performance of PALRS-aligned models on the target tasks GSM8K (top) and HumanEval (bottom) by varying the steering coefficient. Dashed horizontal line marks the baseline performance.

gains: +13% (Llama 3B), +10% (Llama 8B), +29% (Mistral), +2% (OLMo). These results couple with the outstanding time efficiency of PALRS$_{Math}$-aligned models, which are 1 or 2 orders of magnitude faster than DPO-aligned models: for example, using Llama 8B, learning the PALRS$_{Math}$-aligned model takes about 25s vs. DPO-aligned one's 2444s, i.e., $\sim$ 98x faster. Even more remarkably, the efficiency advantage taken by PALRS also holds in comparison with SimPO-aligned models, and is particularly evident for larger models (e.g., $\sim$ 57x faster for Llama 8B.)

Analogous remarks can be drawn from the comparison on HumanEval, where performance gains of PALRS vs. SimPO—which in this benchmark behaves consistently better or on par with DPO—are +11.6% (Llama 1B), +9.8% (Llama 3B), +0.0% (Llama 8B), +27.8% (Mistral), +9.1% (OLMo). Compared to GSM8K, PALRS$_{Code}$-aligned models' efficiency is even more marked, up to 3 orders of magnitude faster than both DPO-aligned and SimPO-aligned models, as shown in the figure for Llama 8B, OLMo, and Mistral.

Impact of the steering coefficient (E4). We investigate the sensitivity of the steering coefficient α (cf. Eq. (5)) and its effect on the performance of PALRS-aligned models on the target tasks. Figure 4 provides insights which can be summarized as follows. We notice that the coefficient sensitivity is model- and task-dependent. While in general performance trends drop consistently as the coefficient increases, there is always a regime of the coefficient where the PALRS-aligned models outperform the relative baselines. Low to moderate coefficients (up to 0.8) often yield the best trade-offs, especially for Llama 3B, Llama 8B (GSM8K) and OLMo (HumanEval). Oversteering is most visible at coefficient 1.0 for all models. More specifically, for Llama 3B and OLMo (HumanEval), the beneficial range is around 0.6–0.8 before oversteering occurs; for Mistral and Llama 1B, oversteering happens much earlier (as low as 0.4–0.6), while Llama 8B tolerates higher coefficients better but still shows oversteering when pushed too far. To sum up, steering is actually beneficial, although, to avoid oversteering and performance degradation, α needs tuning, e.g., by leveraging a small held-out split of the preference dataset using the same metrics as the target benchmark.

6 Conclusions

In this work, we presented PALRS, a training-free method for preference alignment that leverages residual stream activations to steer LLM behavior directly at inference time. By distilling steering vectors from as few as a hundred preference pairs, PALRS aligns models with desired behaviors without any parameter updates, costly optimization, or the need to maintain multiple fine-tuned checkpoints. Crucially, our results show that PALRS-aligned models are a safer and consistently superior alternative to their DPO- and SimPO-aligned counterparts: never underperforming, often achieving substantial gains, and doing so with orders-of-magnitude less computation. Our findings render residual-based steering as a powerful paradigm for preference alignment, aiming to make it simpler, more effective yet scalable, and broadly accessible compared to traditional post-training approaches.

Limitations. Our results have highlighted the promise of residual-based preference alignment, through the evaluation of PALRS on a relatively representative sample of model families and post-training modalities. Despite this, generalizability to very-large-scale models needs to be evaluated. Our current method for discovering effective data subsets and steering coefficients relies on heuristic grid search: developing principled, theoretically grounded selection strategies remains a key direction for smoother real-world deployment of our results. Yet, while our results demonstrate that residual interventions effectively steer model behavior and provide empirical support for our hypothesis, more in-depth explainability of how preference information is encoded and disentangled across layers is needed.

Ethical Statement

While highlighting the potential for beneficial applications, we acknowledge that our work might also be misused to steer models toward harmful or malicious behaviors. We strongly discourage any such misuse and do not assume responsibility for applications beyond the scope of this research.

References

[Arditi *et al.*, 2024] Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2024.

[Bai *et al.*, 2022] Yuntao Bai, Andy Jones, Kamal Ndousse, and et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *CoRR*, abs/2204.05862, 2022.

[Bayat *et al.*, 2025] Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. Steering large language model activations in sparse spaces. In *Conference on Language Modeling*, 2025.

[Belrose *et al.*, 2023] Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. LEACE: perfect linear concept erasure in closed form. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2023.

[Belrose, 2023] Nora Belrose. Diff-in-means concept editing is worst-case optimal: Explaining a result by Sam Marks and Max Tegmark, 2023. https://blog.eleuther.ai/diff-in-means/.

[Bhakthavatsalam *et al.*, 2021] Sumithra Bhakthavatsalam, Daniel Khashabi, Tushar Khot, and et al. Think you have solved direct-answer question answering? try arcda, the direct-answer AI2 reasoning challenge. *CoRR*, abs/2102.03315, 2021.

[Burns *et al.*, 2023] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *Proc. Int. Conf. on Learning Representations*, 2023.

[Cao *et al.*, 2024] Yuanpu Cao, Tianrong Zhang, Bochuan Cao, and et al. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2024.

[Chen *et al.*, 2021] Mark Chen, Jerry Tworek, Heewoo Jun, and et al. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021.

[Chen *et al.*, 2024] Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. Parallel structures in pretraining data yield in-context learning. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 8582–8592, 2024.

[Chen *et al.*, 2025] Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. Persona vectors: Monitoring and controlling character traits in language models. *CoRR*, abs/2507.21509, 2025.

[Cobbe *et al.*, 2021] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, and et al. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

[Dubey *et al.*, 2024] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and et al. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783, 2024.

[Elhage *et al.*, 2022] Nelson Elhage, Tristan Hume, Catherine Olsson, and et al. Toy models of superposition. *CoRR*, abs/2209.10652, 2022.

[Gao and et al., 2024] Leo Gao and et al. The language model evaluation harness, 2024. https://zenodo.org/records/12608602.

[Hendrycks *et al.*, 2021] Dan Hendrycks, Collin Burns, Steven Basart, and et al. Measuring massive multitask language understanding. In *Proc. Int. Conf. on Learning Representations*, 2021.

[Hernandez and Andreas, 2021] Evan Hernandez and Jacob Andreas. The low-dimensional linear geometry of contextualized word representations. In *Proc. Conf. on Computational Natural Language Learning*, pages 82–93, 2021.

[Hong *et al.*, 2024] Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, 2024.

[Huben *et al.*, 2024] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *Proc. Int. Conf. on Learning Representations*, 2024.

[Jiang *et al.*, 2023] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, and et al. Mistral 7b. *CoRR*, abs/2310.06825, 2023.

[Kirchenbauer *et al.*, 2024] John Kirchenbauer, Garrett Honke, Gowthami Somepalli, Jonas Geiping, Katherine Lee, Daphne Ippolito, Tom Goldstein, and David Andre. LMD3: Language model data density dependence. In *Conference on Language Modeling*, 2024.

[Köpf *et al.*, 2023] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, and et al. Openassistant conversations - democratizing large language model alignment. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2023.

[Lan *et al.*, 2024] Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. Quantifying feature space universality across large language models via sparse autoencoders. *CoRR*, abs/2410.06981, 2024.

[Li *et al.*, 2021] Belinda Z. Li, Maxwell Nye, and Jacob Andreas. Implicit representations of meaning in neural language models. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 1813–1827, 2021.

[Li *et al.*, 2023] Kenneth Li, Oam Patel, Fernanda B. Viégas, and et al. Inference-time intervention: Eliciting truthful

answers from a language model. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2023.

[Li *et al.*, 2025] Jiawei Li, Yang Gao, Yizhe Yang, and et al. Fundamental capabilities and applications of large language models: A survey. *ACM Comput. Surv.*, 58(2), 2025.

[Lin *et al.*, 2022] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 3214–3252, 2022.

[Liu *et al.*, 2024] Wenhao Liu, Xiaohua Wang, Muling Wu, and et al. Aligning large language models with human preferences through representation engineering. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 10619–10638, 2024.

[Meng *et al.*, 2024] Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2024.

[Nanda *et al.*, 2023] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *Proc. BlackboxNLP Workshop*, pages 16–30, 2023.

[OLMo *et al.*, 2025] Team OLMo, Pete Walsh, Luca Soldaini, and et al. 2 OLMo 2 Furious. *CoRR*, abs/2501.00656, 2025.

[Ouyang *et al.*, 2022] Long Ouyang, Jeffrey Wu, Xu Jiang, and et al. Training language models to follow instructions with human feedback. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2022.

[Park *et al.*, 2024] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Proc. Int. Conf. on Machine Learning*, 2024.

[Polo *et al.*, 2024] Felipe Maia Polo, Lucas Weber, Leshem Choshen, and et al. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proc. Int. Conf. on Machine Learning*, 2024.

[Rafailov *et al.*, 2023] 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. In *Proc. Annual Conf. on Neural Information Processing Systems*, 2023.

[Rimsky *et al.*, 2024] Nina Rimsky, Nick Gabrieli, Julian Schulz, and et al. Steering llama 2 via contrastive activation addition. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 15504–15522, 2024.

[Sakaguchi *et al.*, 2020] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proc. AAAI Conf.*, pages 8732–8740, 2020.

[Shaib *et al.*, 2024] Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. Detection and measurement of syntactic templates in generated text. In *Proc. Conf. on Empirical Methods in Natural Language Processing*, pages 6416–6431, 2024.

[Shen *et al.*, 2023] Tianhao Shen, Renren Jin, Yufei Huang, and et al. Large language model alignment: A survey. *CoRR*, abs/2309.15025, 2023.

[Shu *et al.*, 2025] Dong Shu, Xuansheng Wu, Haiyan Zhao, Daking Rai, Ziyu Yao, Ninghao Liu, and Mengnan Du. A survey on sparse autoencoders: Interpreting the internal mechanisms of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 1690–1712, 2025.

[Stiennon *et al.*, 2020] Nisan Stiennon, Long Ouyang, Jeffrey Wu, and et al. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020.

[Tigges *et al.*, 2024] Curt Tigges, Oskar J. Hollinsworth, Atticus Geiger, and Neel Nanda. Language models linearly represent sentiment. In *Proc. BlackboxNLP Workshop*, pages 58–87, 2024.

[Turner *et al.*, 2023] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, and et al. Steering language models with activation engineering. *CoRR*, abs/2308.10248, 2023.

[Wang *et al.*, 2023] Yufei Wang, Wanjun Zhong, Liangyou Li, and et al. Aligning large language models with human: A survey. *CoRR*, abs/2307.12966, 2023.

[Wang *et al.*, 2025a] Mengru Wang, Ziwen Xu, Shengyu Mao, Shumin Deng, Zhaopeng Tu, Huajun Chen, and Ningyu Zhang. Beyond prompt engineering: Robust behavior control in LLMs via steering target atoms. In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 23381–23399, 2025.

[Wang *et al.*, 2025b] Xinpeng Wang, Chengzhi Hu, Paul Röttger, and Barbara Plank. Surgical, cheap, and flexible: Mitigating false refusal in language models via single vector ablation. In *Proc. Int. Conf. on Learning Representations*, 2025.

[Wang *et al.*, 2025c] Xinyi Wang, Antonis Antoniades, Yanai Elazar, and et al. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *Proc. Int. Conf. on Learning Representations*, 2025.

[Zellers *et al.*, 2019] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proc. Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, July 2019.

[Zhang and Nanda, 2024] Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *Proc. Int. Conf. on Learning Representations*, 2024.

[Zou *et al.*, 2023] Andy Zou, Long Phan, Sarah Li Chen, and et al. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023.