# RankSteer: Can Pointwise LLM Rankers Be Calibrated at the Representation Level?

Anonymous ACL submission

## Abstract

Large language models (LLMs) are strong zero-shot pointwise rankers, but lag behind pairwise and listwise methods. Beyond missing comparative signals, we identify a *calibration gap*: ranking-relevant information encoded in hidden states is not fully captured by the scalar output head. We propose RankSteer, a post-hoc activation-steering framework that calibrates ranking via projection-based interventions along multiple directions at inference time: decision, evidence, and, optionally, role. This is achieved without updating model weights or introducing cross-document comparisons. We instantiate RankSteer on two structurally distinct pointwise variants and observe improvements over their respective baselines on most TREC DL and BEIR datasets across three backbones. This suggests that the calibration gap is a general property of pointwise rankers. Our additional geometric analysis shows that steering improves ranking by concentrating each query's document representations along an existing ranking geometry, offering new insight into how LLMs internally represent and calibrate relevance judgments.

## 1 Introduction

Large language models (LLMs) have emerged as strong zero-shot rankers, with the pointwise paradigm remaining the most efficient and parallelizable formulation (Sun et al., 2025; Liang et al., 2022; Zhuang et al., 2024a). Yet pointwise LLM rankers consistently lag behind on pairwise, setwise and listwise alternatives that explicitly introduce cross-document comparisons (Sun et al., 2023; Pradeep et al., 2023). The standard explanation attributes this gap to the absence of comparative information between documents. We argue that this explanation is incomplete, and a second factor is a calibration gap: *pointwise LLM rankers encode more ranking-relevant signal in their hidden representations than the output head actually reads out at scoring time*.

Several existing observations support this view. First, the relevance score in pointwise ranking is produced by a single one-dimensional projection, such as the logit difference between "Yes" and "No" (Liang et al., 2022). The hidden states feeding this projection, however, are in a space of thousands of dimensions, so information present in the representation but not aligned with the single output projection is discarded at scoring. Second, as shown in Figure 1(a,b), the brittleness of pointwise rankers to prompt phrasing and role framing (Sun et al., 2025; Wang et al., 2026) further suggests that the readout, rather than the representation, is the unstable component. Third, mechanistic analyses show that role and content signals propagate through largely separate channels in LLM rankers' hidden states (Wang et al., 2026), suggesting a structured representation whose components could in principle be addressed independently.

These observations point to a different intervention method than prompt engineering. Existing attempts to calibrate pointwise scores act only at the input or scoring level (Long et al., 2025; Li et al., 2025a; Wang et al., 2026) and leave the hidden representation untouched. Rather than reshaping the input until the output head produces a better score, we investigate whether the representation can be realigned so that more of the ranking-relevant signal it already contains is reflected in the score. This shifts the problem from input search to post-hoc, representation-level calibration of a frozen LLM.

We propose RankSteer (Figure 2), a modular activation-steering framework for calibrating zero-shot pointwise LLM rankers. RankSteer characterizes ranking behavior through up to three steerable directions in the hidden space: (1) a **decision direction**, which is the input-agnostic axis along which the output head produces relevance scores; (2) an **evidence direction**, which captures the relevance signal present in the hidden states but orthogonal

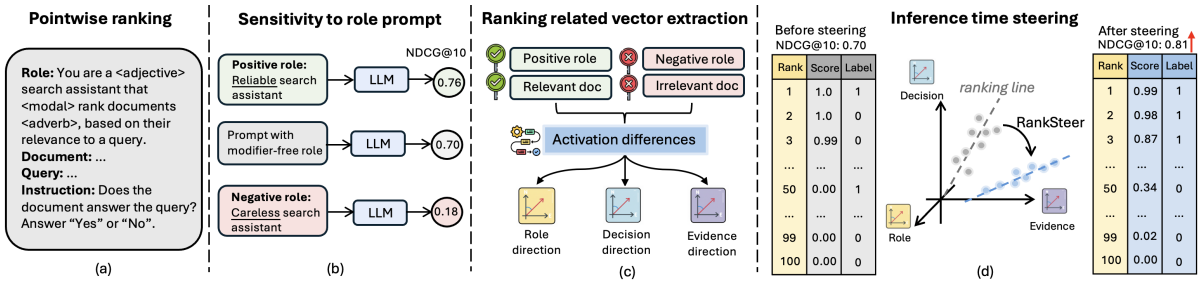


Figure 1: (a)(b) Pointwise LLM ranking prompt and its sensitivity to role phrasings. (c) Decomposing ranking decisions into decision, evidence, and role directions. (d) Performance gain after steering in inference time.

to the decision axis (the calibration gap); and (3) an optional **role direction**, an orthogonal axis that modulates how decisively the decision direction is applied, which is relevant for base rankers with role-style prompting. At inference time, RankSteer applies simple projection-based interventions along these directions to recalibrate ranking representations, **without modifying model weights or introducing cross-document comparisons**.

Because the decision direction is induced directly from the model output head, RankSteer is broadly compatible with all pointwise LLM rankers whose relevance score reduces to a two-token contrast, regardless of differences in prompting or scoring design. We instantiate it on two structurally distinct bases, RankSteer-Role and RankSteer-Ref, where both improve over their respective base on the majority of benchmarks using only a handful of out-of-distribution queries, indicating that the calibration gap generalizes across binary-contrast pointwise settings.

Our contributions are: (1) We propose RankSteer, the first post-hoc activation-steering framework for pointwise LLM reranking, which introduces projection-based interventions along decision, evidence, and role directions to steer relevance scoring at the representation level. (2) We instantiate RankSteer on two distinct pointwise bases, RankSteer-Role and RankSteer-Ref, and show that both improve over their respective base on the majority of ranking benchmarks across three backbones. (3) We provide a geometric analysis showing that calibration operates by concentrating each query's document representations along an existing ranking geometry, and that representation-level calibration recovers more ranking quality than LoRA fine-tuning on the same anchor set. Our code is available at `https://anonymous.4open.science/r/ranksteer-05BE`.

## 2 Related Work

**Zero-Shot Ranking.** Unlike supervised rankers (e.g., MonoT5 (Nogueira et al., 2020), RankT5 (Zhuang et al., 2023a)) that require substantial labeled data, LLM rankers work in a zero-shot setting. Under the traditional pointwise paradigm, a model scores each (query, document) pair independently (Sachan et al., 2022; Zhuang et al., 2023b; Liang et al., 2022; Zhuang et al., 2024a). A line of work aims to improve the reliability of these scores, which we view as *calibrating* the readout. Existing approaches operate at the input or scoring level, including aggregation-level contrastive post-processing such as PAGC (Long et al., 2025), reference-level anchoring such as RefRank (Li et al., 2025a), and prompt-level role framing (Wang et al., 2026). However, these methods do not modify the representation consumed by the output head, leaving the calibration gap insufficiently addressed. RankSteer is the first approach to address calibration at the representation level. Comparative methods, including pairwise (Qin et al., 2024; Luo et al., 2024), setwise (Zhuang et al., 2024b) and listwise (Ma et al., 2023; Pradeep et al., 2023; Sun et al., 2023) prompting, improve ranking through explicit cross-document comparison. However, they incur high inference cost, limited parallelism, and positional bias (Long et al., 2025). RankSteer instead remains within the pointwise paradigm and avoids cross-document comparison.

**Role-Play Prompting.** Role-play prompting assigns an explicit role or persona to a language model, guiding its behavior and output style (Shanahan et al., 2023). Modern LLMs exhibit strong role-playing abilities, enabling them to better perform various tasks (Kong et al., 2024; Wang et al., 2024). For ranking tasks, LLMs have been shown to be highly sensitive to role-

play instructions, heavily affecting their effectiveness (Sun et al., 2025; Wang et al., 2026). Recent causal intervention studies (Meng et al., 2022; Zhang and Nanda, 2023) further show that role-play instructions are processed largely independently from query-document representations in the model's hidden layers (Wang et al., 2026). Building on these findings, our work moves beyond prompt-level manipulation by extracting and steering role-related activation directions, enabling post-hoc and interpretable control over ranking behavior.

**Activation Steering.** Activation steering has emerged as a lightweight and transparent paradigm for controlling LLM behavior without modifying model weights (Zou et al., 2023; Bartoszcze et al., 2025). The core premise is that many semantic concepts, behaviors, and decision patterns are encoded as approximately linear directions in the activation space, which can be identified and manipulated at inference time (Park et al., 2023; Rimsky et al., 2024). Steering directions are typically extracted using contrastive examples, probing classifiers, or sparse auto-encoders, and injected into hidden states to bias model outputs in a targeted manner (Bayat et al., 2025; Wang et al., 2025a; Hong et al., 2025). This paradigm has been successfully applied across a wide range of tasks (Chen et al., 2025; Wang et al., 2025b; Li et al., 2025b; Zhao et al., 2025; Fang et al., 2026). Recent work has shown that role-playing behavior can be steered via internal activations using sparse autoencoders, leading to improved *reasoning* performance and highlighting the potential of activation-level control beyond surface-level prompt engineering (Wang et al., 2025a). The application of activation steering to ranking remains unexplored. We bridge this gap by showing that relevance, decision, and role-play signals in pointwise LLM rankers can be disentangled and calibrated post hoc, enabling effective ranking improvements without modifying model parameters or introducing cross-document comparisons.

# 3 Methodology

We propose a post-hoc activation-steering framework that calibrates zero-shot pointwise LLM rankers at the representation level, illustrated in Figure 2. For brevity, RankSteer refers to RankSteer-Role throughout this section.

## 3.1 Preliminaries

**Pointwise Zero-Shot Reranking.** Following established protocols (Nogueira et al., 2020), we adopt a binary Yes/No formulation (Liang et al., 2022), which lets us analyze relevance judgments directly through the model's internal representations associated with the final token. Details are in Appendix A.1.

**Activation Steering Framework.** Activation steering involves two main steps: Steering Vector Extraction and Inference-Time Intervention. Following prior work (Bartoszcze et al., 2025; Zhao et al., 2025), we adopt these preliminaries and refer the reader to Appendix A.2 for details.

## 3.2 Constructing Steering Vectors

The calibration gap suggests that the hidden state contains more than the output head reads. We decompose the two into a decision direction (what the head reads) and an evidence direction (what it leaves behind), with an optional role direction for role-prompted bases. We construct the steering vectors from a small set of anchor queries from a separate dataset, using them solely for direction estimation without any parameter updates, rather than for fine-tuning.

### 3.2.1 Decision Vector

The decision vector defines a global relevance axis induced by the output head. In pointwise ranking, the model maps a hidden state to the logits of "Yes" and "No" via its output embedding matrix $W \in \mathbb{R}^{V \times d}$. Letting $W_{\text{yes}}$ and $W_{\text{no}}$ denote the corresponding weight rows, we define the decision vector as their normalized difference:

$$v_{\text{dec}} = \frac{W_{\text{yes}} - W_{\text{no}}}{|W_{\text{yes}} - W_{\text{no}}|} \quad (1)$$

Being input-agnostic, $v_{\text{dec}}$ is shared across all queries and documents and fixes the direction along which the hidden state is projected to form a judgment. Steering along this axis directly affects the model's decisiveness without altering the underlying evidence representations.

### 3.2.2 Evidence Vector

To extract a relevance signal that is consistent with the model's existing ranking behavior with respect to a given dataset, we construct contrastive anchor pairs under both label and ranking constraints. Specifically, for a query $q_i$, we select $n$ pairs of documents where the relevant document $d_{i,j}^{+}$ is ranked

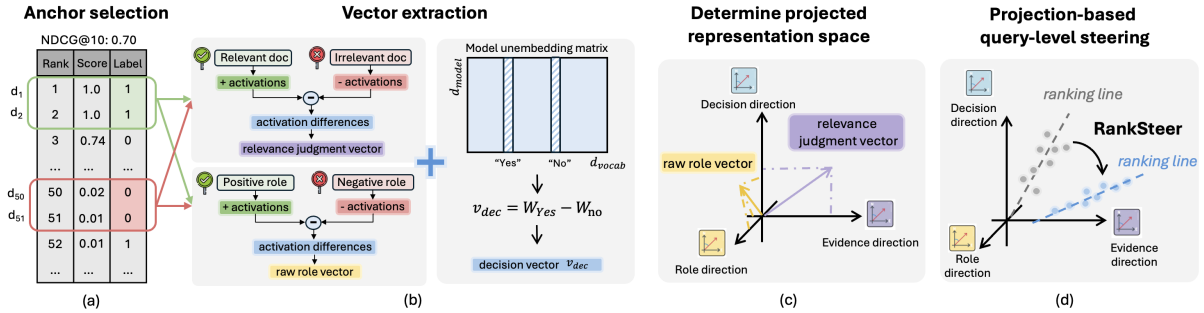


Figure 2: Overview of the RankSteer framework. (a) Selection of query-document anchors for constructing contrastive input pairs. (b) Extraction of ranking-related steering vectors from anchor data. (c) Construction of a projection-based representation space using the extracted vectors. (d) Inference-time, query-level activation steering via projections in the constructed space.

higher and the irrelevant document $d^-_{i,j}$ is ranked lower by the vanilla LLM. This ensures that the resulting contrast captures how the model currently distinguishes relevance, rather than representations associated with model errors or uncertainty. Let $h^{(l)+}_{i,j}$ and $h^{(l)-}_{i,j}$ denote the corresponding last-token hidden states at layer $l$. The raw relevance judgment vector $v^{(l)}_{\text{rel}}$ is computed across $m$ queries as:

$$v^{(l)}_{\text{rel}} = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left( h^{(l)+}_{i,j} - h^{(l)-}_{i,j} \right) \quad (2)$$

**Disentanglement and Calibration.** While $v^{(l)}_{\text{rel}}$ encodes relevance, a significant portion of this signal is already consumed by the decision head ($v_{\text{dec}}$). To isolate the residual relevance evidence that is present in the hidden states but not directly exploited for the final "Yes/No" logit, we project $v^{(l)}_{\text{rel}}$ onto the orthogonal complement of the decision axis and normalize it:

$$v^{(l)}_{\text{evid}} = \text{Norm}(v^{(l)}_{\text{rel}} - \langle v^{(l)}_{\text{rel}}, v_{\text{dec}} \rangle v_{\text{dec}}) \quad (3)$$

While the decision vector determines whether a document is judged relevant, the evidence vector reflects how strongly document content supports relevance in a way that is comparable across documents. As such, evidence serves as a calibration signal: steering along this direction modulates the contribution of under-utilized relevance cues without altering the underlying decision boundary.

#### 3.2.3 Role Vector

Beyond relevance signals derived from the query-document content, LLMs are highly sensitive to high-level task framing and role-play instructions. Such role signals do not introduce new relevance evidence, but instead influence how the model applies its relevance judgment, acting as a form of behavior-level control. We extract a raw role direction from prompt pairs differing only in role instruction (positive vs. negative) with query and document fixed (Appendix B.1).

Let $h^{(l)\text{role}+}_{i,j,k}$ and $h^{(l)\text{role}-}_{i,j,k}$ be the hidden states for $t$ contrastive role pairs. The raw role vector $v^{(l)}_{\text{role_raw}}$ is aggregated over $m$ queries and $n$ documents.

$$v^{(l)}_{\text{role_raw}} = \frac{1}{t} \frac{1}{n} \frac{1}{m} \sum_{k=1}^{t} \sum_{i=1}^{n} \sum_{j=1}^{m} \left( h^{(l)\text{role}+}_{ijk} - h^{(l)\text{role}-}_{ijk} \right) \quad (4)$$

We then project it off both relevance axes and normalize:

$$\begin{aligned} v^{(l)}_{\text{role}} = \text{Norm}\big( & v^{(l)}_{\text{role_raw}} - (v^{(l)}_{\text{role_raw}} \cdot v_{\text{dec}})\, v_{\text{dec}} \\ & - (v^{(l)}_{\text{role_raw}} \cdot v^{(l)}_{\text{evid}})\, v^{(l)}_{\text{evid}} \big) \end{aligned} \quad (5)$$

After normalization, the resulting $v^{(l)}_{\text{role}}$ is orthogonal to both relevance-related axes. We do not attempt to analyze whether role and relevance are naturally disentangled. Instead, we explicitly construct independent directions to isolate the effect of role-play on relevance judgments from relevance evidence itself.

### 3.3 Layerwise Steering at Inference Time

We intervene on the last-token hidden state $h^{(l)}$ at every layer, giving finer control over the model's internal scoring than static prompting.

**Projection Onto Steering Directions.** Given the three unit steering vectors: decision vector $v_{\text{dec}}$, evidence vector $v^{(l)}_{\text{evid}}$, and role vector $v^{(l)}_{\text{role}}$, we first compute scalar projections of the hidden state:

$$p^{(l)}_{\text{dec}} = h^{(l)} \cdot v_{\text{dec}}, \quad p^{(l)}_{\text{evid}} = h^{(l)} \cdot v^{(l)}_{\text{evid}}, \quad p^{(l)}_{\text{role}} = h^{(l)} \cdot v^{(l)}_{\text{role}}. \quad (6)$$

These projections quantify how strongly the current representation aligns with the three directions at layer $l$.

**Decision and Evidence Steering.** We then apply relevance-oriented steering by adjusting the hidden state along the decision and evidence directions:

$$\tilde{h}^{(l)} = h^{(l)} - \alpha \cdot p_{\text{dec}}^{(l)} v_{\text{dec}} - \beta \cdot p_{\text{evid}}^{(l)} v_{\text{evid}}^{(l)} \quad (7)$$

where $\alpha$ and $\beta$ control the scaling of the decision and evidence axes. Steering along $v_{\text{dec}}$ affects the model's global decisiveness, while steering along $v_{\text{evid}}^{(l)}$ modulates under-utilized relevance cues for better cross-document calibration. Together they shift the representation so that more of its relevance content is reflected in the score, closing the calibration gap by reweighting signals already present rather than injecting external information.

**Role-Gated Decision Steering.** For role-prompted bases, the role signal controls how decisively the decision direction is applied rather than adding relevance evidence. We turn the role projection into a gate and use it to modulate decision steering:

$$g^{(l)} = \sigma(p_{\text{role}}^{(l)}) \quad (8)$$

$$h'^{(l)} = \tilde{h}^{(l)} - \gamma \cdot g^{(l)} \cdot p_{\text{dec}}^{(l)} v_{\text{dec}} \quad (9)$$

where $\sigma(\cdot)$ is the sigmoid function and $\gamma$ is the role modulation coefficient. The gate measures the activation strength of role-related features at layer $l$, scaling decision steering accordingly. Role steering adds no relevance information, and only adjusts how decisively the ranker applies its judgment.

**Final Scoring.** After steering all layers, the final hidden state is passed to the output head to compute relevance logits, from which pointwise relevance scores are derived as described in Section 3.1. All hyperparameters $\alpha$, $\beta$, and $\gamma$ are fixed at inference time and tuned on an external validation set.

# 4 Experiments

## 4.1 Experimental Setup

**Backbone.** Following prior works on activation steering (Zou et al., 2023; Zhao et al., 2025), we conduct experiments on three representative instruction-tuned decoder-only LLMs as the backbone of our ranker: Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Qwen et al., 2025), and Mistral-7B-Instruct-v0.3 (Jiang et al., 2023; AI, 2024).

**Datasets and Metrics.** We evaluate on two widely used IR benchmarks: TREC Deep Learning (DL) (Craswell et al., 2020, 2021) and BEIR (Thakur et al., 2021). TREC DL provides graded relevance judgments suitable for constructing reliable anchor data; we therefore use TREC DL 2019 to construct anchor inputs and tune the steering hyperparameters. We evaluate in-distribution generalization on TREC DL 2020, which shares the same corpus as DL 2019 but contains disjoint queries, and out-of-distribution generalization on eight BEIR datasets. We report nDCG@10 as the primary metric, and additionally report MRR@10, MAP and binary accuracy (BA) when analyzing individual steering directions. Binary accuracy measures whether the model correctly predicts relevance by comparing the likelihood of "Yes" versus "No" with a 0.5 threshold.

**Baselines.** Following the calibration taxonomy in Section 2, we group pointwise baselines into plain pointwise rankers that read scores directly off the output head (**QG** (Sachan et al., 2022), **RG-YN** (Liang et al., 2022), **RG-S(0–k)** (Zhuang et al., 2024a)) and calibration-oriented rankers that recalibrate at the aggregation level (**PAGC-YG** (Long et al., 2025)), reference level (**RefRank-(k)** (Li et al., 2025a)), or prompt level (**RG-YN_role** (Wang et al., 2026)). RankSteer calibrates at the representation level and is instantiated on the two strong bases among: **RankSteer-Role** on RG-YN_role, applying all three directions and **RankSteer-Ref** on RefRank-(4), applying only the decision and evidence directions. We further compare against comparative methods (**PRP** (Luo et al., 2024), **Setwise** (Zhuang et al., 2024b), **RankGPT** (Sun et al., 2023)) as a different-paradigm reference. Implementation details for all baselines are in Appendix C.

**Implementation Details.** We retrieve top-100 BM25 candidates with Pyserini (Lin et al., 2021). Relevance anchors use $m = 5$ DL 2019 queries, each with $n = 10$ relevant and $n = 10$ irrelevant documents aligned with the base ranker's behavior; role-play anchors use $t = 3$ instruction pairs from Wang et al. (2026) (see Appendix B.1). The remaining 38 DL 2019 queries tune $\alpha$, $\beta$, $\gamma$, which are then fixed across DL 2020 and BEIR. A multi-fold selection procedure mitigates variance from anchor query sampling. Additional implementation details are provided in Appendix B.2.

| | Methods | TREC DL dl20 | BEIR covid | touche | signal | news | scifact | robust04 | dbpedia | nfcorpus | Avg |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | BM25 | 0.4796 | 0.5947 | 0.4422 | 0.3304 | 0.3952 | 0.6789 | 0.4070 | 0.3180 | 0.3218 | 0.4360 |
| Llama-3.1-8B | QG | 0.3498 | 0.6925 | 0.2205 | 0.1015 | 0.4221 | 0.6041 | 0.4508 | 0.2817 | 0.3506 | 0.3905 |
| | RG-YN | 0.6109 | 0.7261 | 0.2264 | 0.2482 | 0.3665 | 0.5677 | 0.5046 | 0.3093 | 0.3563 | 0.4131 |
| | RG-S(0,4) | 0.5190 | 0.7902 | 0.2205 | 0.2637 | 0.4451 | 0.6241 | 0.5039 | 0.3666 | 0.3646 | 0.4473 |
| | PAGC-YG | 0.6193 | 0.7534 | 0.2424 | 0.2586 | 0.4503 | 0.6485 | 0.5199 | 0.3420 | 0.3596 | 0.4468 |
| | RefRank-(4) | 0.5576 | 0.7828 | 0.2418 | 0.2720 | 0.4408 | 0.6443 | 0.5075 | 0.3483 | 0.3487 | 0.4203 |
| | RG-YN_role | 0.6230 | 0.7792 | 0.2227 | 0.2661 | 0.4456 | **0.6606** | **0.5321** | 0.3574 | 0.3715 | 0.4544 |
| | RankSteer-Ref | 0.5936$^{\dagger}$ | **0.7942**$^{\dagger}$ | 0.2443 | **0.2893**$^{\dagger}$ | **0.4660**$^{\dagger}$ | 0.6561$^{\dagger}$ | 0.5126 | 0.3598$^{\dagger}$ | 0.3575$^{\dagger}$ | **0.4600** |
| | RankSteer-Role | **0.6461**$^{\dagger}$ | 0.7863 | **0.2445**$^{\dagger}$ | 0.2779 | 0.4462 | 0.6498 | 0.5207 | **0.3752**$^{\dagger}$ | **0.3732** | 0.4588 |
| Mistral-7B | QG | 0.4287 | 0.6621 | 0.2277 | 0.1010 | 0.3048 | 0.4816 | 0.3506 | 0.3075 | 0.3019 | 0.3422 |
| | RG-YN | 0.4897 | 0.7315 | 0.1976 | 0.2243 | 0.3963 | 0.5143 | 0.4760 | 0.2791 | 0.3335 | 0.3941 |
| | RG-S(0,4) | 0.5491 | 0.7892 | 0.2468 | 0.2679 | **0.4678** | **0.6324** | 0.4856 | 0.3672 | 0.3512 | **0.4510** |
| | PAGC-YG | 0.5522 | 0.7775 | 0.2212 | 0.2586 | 0.3949 | 0.4476 | 0.4894 | 0.3307 | 0.3338 | 0.4067 |
| | RefRank-(4) | 0.5290 | 0.7520 | 0.2885 | 0.2858 | 0.4382 | 0.5987 | 0.4823 | 0.3454 | 0.3383 | 0.4186 |
| | RG-YN_role | 0.5082 | 0.7136 | 0.1998 | 0.2268 | 0.4192 | 0.5304 | 0.4884 | 0.2954 | 0.3401 | 0.4017 |
| | RankSteer-Ref | 0.5407$^{\dagger}$ | 0.7729$^{\dagger}$ | **0.2965** | 0.2849 | 0.4432 | 0.6127$^{\dagger}$ | 0.4892 | 0.3375 | 0.3422 | 0.4474 |
| | RankSteer-Role | **0.6355**$^{\dagger}$ | **0.8019**$^{\dagger}$ | 0.2254$^{\dagger}$ | **0.2877**$^{\dagger}$ | 0.4441$^{\dagger}$ | 0.5664$^{\dagger}$ | **0.5123**$^{\dagger}$ | **0.3965**$^{\dagger}$ | **0.3623**$^{\dagger}$ | 0.4496 |
| Qwen2.5-7B | QG | 0.3982 | 0.6463 | 0.2190 | 0.0922 | 0.2533 | 0.3499 | 0.3617 | 0.2530 | 0.2845 | 0.3075 |
| | RG-YN | 0.6203 | 0.6780 | 0.2025 | 0.2305 | 0.3040 | 0.2395 | 0.3484 | 0.2733 | 0.3106 | 0.3234 |
| | RG-S(0,4) | 0.6059 | 0.7814 | 0.2333 | 0.2661 | 0.4490 | 0.6261 | 0.4756 | 0.3618 | 0.3599 | 0.4441 |
| | PAGC-YG | 0.6500 | 0.7395 | 0.2413 | 0.2761 | 0.3772 | 0.4632 | 0.4488 | 0.3525 | 0.3552 | 0.4067 |
| | RefRank-(4) | 0.6327 | 0.7818 | **0.2816** | **0.3046** | 0.4793 | 0.6737 | 0.5142 | 0.4070 | 0.3736 | 0.4489 |
| | RG-YN_role | 0.6156 | 0.7212 | 0.1985 | 0.2249 | 0.3048 | 0.2398 | 0.3691 | 0.2909 | 0.3183 | 0.3334 |
| | RankSteer-Ref | 0.6475$^{\dagger}$ | **0.7925**$^{\dagger}$ | 0.2746 | 0.3017 | **0.4913**$^{\dagger}$ | **0.6778** | **0.5226**$^{\dagger}$ | **0.4148**$^{\dagger}$ | **0.3759** | **0.4814** |
| | RankSteer-Role | **0.6594**$^{\dagger}$ | 0.7678$^{\dagger}$ | 0.2356$^{\dagger}$ | 0.2793$^{\dagger}$ | 0.3807$^{\dagger}$ | 0.4133$^{\dagger}$ | 0.4338$^{\dagger}$ | 0.3749$^{\dagger}$ | 0.3697$^{\dagger}$ | 0.4069 |

Table 1: Comparison of RankSteer to pointwise ranking baselines in terms of nDCG@10. Within each backbone, methods are ordered as plain pointwise rankers, calibration-oriented rankers, and RankSteer. RankSteer-Role (our main method) applies three-direction activation steering on the RG-YN_role base. RankSteer-Ref (modular extension) applies decision and evidence steering only on RefRank-(4) as base. Best and second-best results are marked with **boldface** and underline respectively. All LLMs are instruction-tuned versions. † indicates a statistically significant improvement over the corresponding base under a two-tailed paired $t$-test with $p < 0.05$.

## 4.2 Experimental Results

### 4.2.1 Comparison with Pointwise Methods

Table 1 summarizes nDCG@10 against pointwise baselines. We organize our findings around three observations. **First, steering improves its base across backbones.** Across all three backbones, each steered variant improves over its base on TREC DL 2020 and the BEIR average. As the two baselines differ in scoring tokens, anchors, and role usage, these consistent gains suggest that the calibration gap is a general property of binary-contrast pointwise rankers. **Second, in-distribution gains are stable.** RankSteer-Role achieves the best TREC DL 2020 result on every backbone using only five TREC DL 2019 anchor queries, indicating that effective directions can be extracted from a small anchor set and transferred to unseen in-distribution queries. **Third, steering remains beneficial on average for out-of-distribution datasets, but its strength is domain-, base-, and backbone-dependent:** RankSteer-Ref is the more robust variant across BEIR datasets, while RankSteer-Role is less competitive when a plain base is already strong. The datasets on which steering underperforms also vary across backbones, suggesting the extracted directions are partly dataset-dependent: anchors drawn from TREC DL 2019 transfer unevenly to heterogeneous BEIR domains. **Overall, no single base is uniformly best**, and RankSteer's modularity enables flexible adaptation, allowing the base method to be chosen according to the backbone model and data distribution. We provide additional comparisons with non-pointwise baselines in Appendix D.

### 4.2.2 Ablation of Steering Directions

For the following analysis, we study steering behavior using a single representative model, Llama-3.1-8B-Instruct, analyzed on the TREC DL 2019

| Vector | nDCG | MRR | MAP | BA |
|---|---|---|---|---|
| - | 0.6800 | 0.8985 | 0.3656 | 0.7665 |
| $v_{dec}$ | 0.6963 | 0.9193 | 0.3673 | 0.7457 |
| $v_{evid}$ | 0.6843 | 0.9020 | 0.3662 | 0.7665 |
| $v_{dec} + v_{evid}$ | 0.7061 | 0.9386 | 0.3701 | 0.7625 |
| $v_{dec} + v_{role}$ | 0.7029 | **0.9456** | 0.3674 | 0.7486 |
| $v_{dec} + v_{role} + v_{evid}$ | **0.7102** | 0.9342 | **0.3712** | 0.7641 |

Table 2: Ablation of steering directions on a TREC DL 2019 validation fold with Llama-3.1-8B-Instruct. We report nDCG and MRR at a cutoff of 10, MAP at 100.

validation set. Table 2 reports ablation results on a random TREC DL 2019 partition. The decision direction improves ranking but lowers binary accuracy, suggesting a rescaling of decision scores, whereas the evidence direction yields smaller ranking gains while leaving binary accuracy unchanged, indicating that it injects under-utilized relevance signal without shifting the decision boundary. Combining the two outperforms either alone, confirming their complementarity. Adding the role direction gives the best overall ranking while keeping binary accuracy close to the unsteered baseline, suggesting that role steering modulates how decisively the decision direction is applied rather than introducing new relevance evidence. Together, the three directions contribute complementary effects and improve ranking without destabilizing the underlying relevance judgments.

#### 4.2.3 Sensitivity to Anchor Query Set

We assess robustness to anchor selection along two axes: which queries are used, and how many.

**Anchor choice.** We construct steering vectors from eight disjoint five-query partitions of TREC DL 2019, tuning hyperparameters within each (Figure 3). Steering improves ranking across all eight partitions, with nDCG@10 and MRR@10 increasing by comparable margins in every case. MAP and binary accuracy fluctuate more, but the latter is expected, as steering rescales decision scores and can shift the Yes/No threshold without degrading ranking. The magnitude of improvement varies with the anchor set while its direction stays stable, indicating that RankSteer does not rely on a carefully chosen anchor set.

**Anchor count.** We fix 33 queries for validation and sample $m \in \{1, 3, 5, 8, 10\}$ anchors from the remaining 10, repeating five times (Figure 4). Ranking quality is largely insensitive to $m$: even

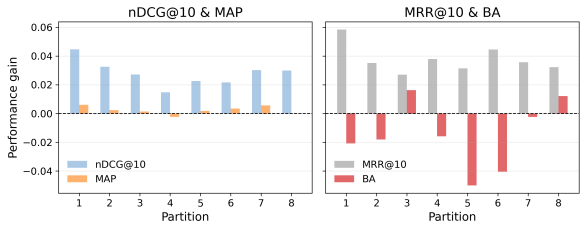


Figure 3: Sensitivity to anchor query set. Performance gain after steering across eight anchor sets (5 queries) of TREC DL 19 using Llama-3.1-8B-Instruct.

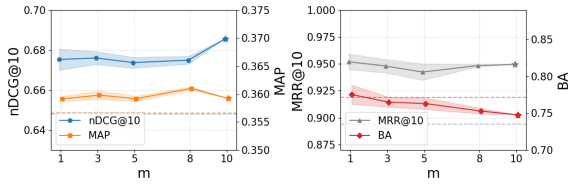


Figure 4: Effect of anchor query counts on TREC DL 2019 with Llama-3.1-8B-Instruct. The horizontal dotted line is the baseline without model steering. For $m = 10$, there is only one set of queries and therefore a single measurement.

a single anchor is competitive on average, though with noticeably higher variance. We therefore recommend $m = 5$ as a balanced setting that delivers robust gains without unnecessary variance.

#### 4.2.4 Interpretability from the Geometric Perspective

Since role-play primarily modulates the decision mechanism rather than document content, we analyze ranking behavior in the two-dimensional space spanned by the decision and evidence directions. Figure 5 visualizes ranking geometry at both the query and dataset levels.

**What is being steered?** For a given query, document representations lie in an approximately linear arrangement on the decision–evidence plane, which we call a *ranking line* (Figure 5(a)); this structure persists both before and after steering. Steering therefore, does not create or remove the ranking line but reshapes its statistics: representations become more concentrated, with reduced spread along the decision direction. At the dataset level (Figures 5(b,c)), for layers where the geometry is well-formed (see full layer geometry in Appendix E), steering consistently reduces both the slope and the spread of fitted ranking lines, where we quantify this spread by their *dispersion*, i.e., the standard error of the fitted line. This means steering lowers the dominance of the decision di-

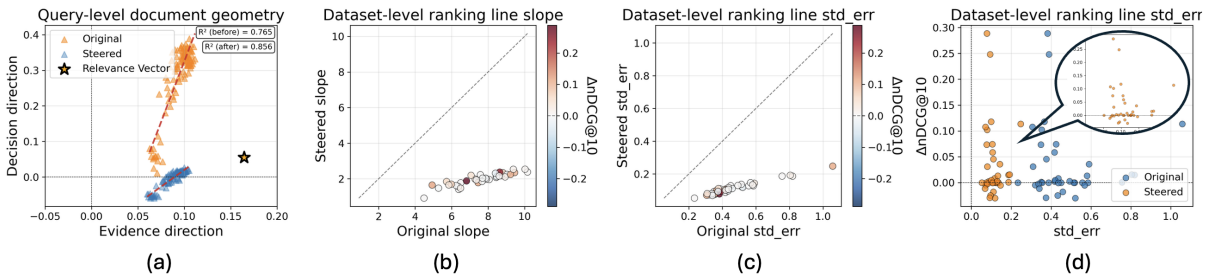


Figure 5: Geometric interpretability of RankSteer-Role at Layer 16 of Llama-3.1-8B-Instruct. (a) Query-level document representations projected onto the decision-evidence plane, illustrating linear ranking geometry. (b) and (c) Dataset-level distributions of the slope and standard error of query-specific ranking lines before and after steering. (d) A complementary view of (c), highlighting how steering redistributes query-level ranking dispersion in relation to changes in nDCG.

| Model | nDCG@10 | MRR@10 | MAP | BA |
|---|---|---|---|---|
| Vanilla | 0.6715 | 0.9079 | 0.3807 | 0.7694 |
| Fine-tuned | 0.6773 | **0.9430** | 0.3783 | 0.7650 |
| Steered | **0.7040** | **0.9430** | **0.3831** | 0.7513 |

Table 3: Comparison of steering and fine-tuned model on the same anchor set using Llama-3.1-8B-Instruct.

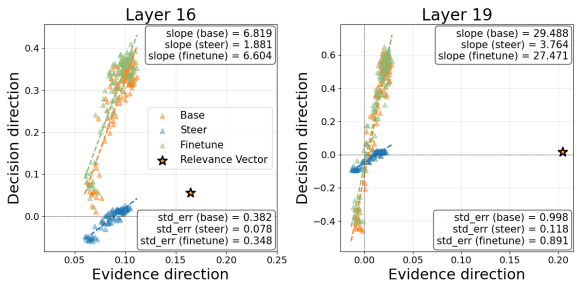


Figure 6: Comparison of query-level document representations under steering and fine-tuning.

rection relative to evidence, letting representations align more closely with evidence-supported relevance and making ranking structure more coherent across queries. Consistently, the dispersion ΔnDCG relationship in Figure 5(d) shows steering shifting queries toward lower dispersion within an unchanged overall pattern, improving ranking by tightening an existing geometry rather than introducing new ranking behavior.

**Comparison with fine-tuning** Using the same five anchor queries, we fine-tune Llama-3.1-8B-Instruct with LoRA (Hu et al., 2022) and evaluate on the same validation set (Table 3). With identical supervision data, fine-tuning improves nDCG@10 only modestly, whereas steering with these anchor queries (without finetuning) achieves a markedly larger gain while matching fine-tuning on MRR@10 and without updating any weights; binary accuracy decreases slightly, consistent with the rescaling of decision scores. Geometrically (Figure 6), fine-tuning largely preserves the base model's ranking lines with only moderate reductions in slope and dispersion, while steering induces a stronger effect, compressing both more substantially and yielding tighter clustering along the decision direction while staying close to the original ranking manifold. These results highlight a sample-efficiency advantage in low-resource settings: with only five anchor queries, directly manipulating activation geometry recovers under-utilized ranking capacity more effectively than parameter updates when supervision is scarce.

## 5 Conclusion

We introduced RankSteer, a post-hoc activation-steering framework that calibrates pointwise LLM rankers at the representation level. Rather than reshaping the input, RankSteer recalibrates the representation at inference time through projection-based interventions along a decision direction, an evidence direction, and an optional role direction, without updating model weights or introducing cross-document comparisons. Across three backbones, we apply RankSteer on two strong base methods and see that it improves over its base on most TREC DL and the BEIR datasets, using steering vectors extracted from only five anchor queries. A geometric analysis shows that it works by pulling each query's document representations into a tighter, more consistent spread along an existing ranking geometry. Our results suggest that the calibration gap is a general property of binary-contrast pointwise rankers, and that representation-level calibration offers a favorable trade-off, recovering under-utilized ranking capacity while preserving the full parallelism and efficiency of standard pointwise ranking.

## Limitations

First, our method applies to pointwise rankers and does not directly extend to other paradigms. This boundary follows from how the decision direction is constructed: it is read off the output head as a two-token margin, so RankSteer applies to any pointwise base whose score reduces to such a contrast, covering both the Yes/No base RG-YN_role and the A/B base RefRank, but not to listwise or setwise methods that lack a single two-token scoring margin. As the first work to bring activation steering to ranking, we leave these paradigms to future work. Second, we limited our experiments to models of 7B and 8B parameters, leaving other model sizes to future work. Third, for the role direction we adopt the role-play templates of prior work (Wang et al., 2026) rather than optimizing them, as template design is not a focus of this paper. That the role direction is already effective under these off-the-shelf templates suggests our gains do not depend on carefully engineered role prompts, though a dedicated search over role templates may yield further improvements.

## Ethics Statement

All models and datasets used in this work are publicly available for research purposes and are used for their intended purpose of evaluating zero-shot ranking. Our use is consistent with their respective licenses and access conditions. We release our method and code for research use only, consistent with the access conditions of the underlying artifacts.

## References


Mistral AI. 2024. Mistral-7b-instruct-v0.3 model card. https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3. Accessed 2025.

Lukasz Bartoszcze, Sarthak Munshi, Bryan Sukidi, Jennifer Yen, Zejia Yang, David Williams-King, Linh Le, Kosi Asuzu, and Carsten Maple. 2025. Representation engineering for large-language models: Survey and research challenges. *arXiv preprint arXiv:2502.17601*.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. *arXiv preprint arXiv:2503.00177*.

Runjin Chen, Andy Arditi, Henry Sleight, Owain Evans, and Jack Lindsey. 2025. Persona vectors: Monitoring and controlling character traits in language models. *arXiv preprint arXiv:2507.21509*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the TREC 2020 deep learning track. In *Text REtrieval Conference (TREC)*.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Rodrigo Nogueira. 2020. Overview of the trec 2019 deep learning track. In *Proceedings of the Twenty-Eighth Text REtrieval Conference, TREC 2019*.

Yi Fang, Wenjie Wang, Mingfeng Xue, Boyi Deng, Fengli Xu, Dayiheng Liu, and Fuli Feng. 2026. Controllable llm reasoning via sparse autoencoder-based steering. *Preprint*, arXiv:2601.03595.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yihuai Hong, Meng Cao, Dian Zhou, Lei Yu, and Zhijing Jin. 2025. The reasoning-memorization interplay in language models is mediated by a single direction. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21565–21585, Vienna, Austria. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Aobo Kong, Shiwan Zhao, Hao Chen, Qicheng Li, Yong Qin, Ruiqi Sun, Xin Zhou, Enzhi Wang, and Xiaohang Dong. 2024. Better zero-shot reasoning with role-play prompting. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4099–4113.

Jieran Li, Xiuyuan Hu, Yang Zhao, Shengyao Zhuang, and Hao Zhang. 2025a. Leveraging reference documents for zero-shot ranking via large language models. *arXiv preprint arXiv:2506.11452*.

Tianlong Li, Zhenghua Wang, Wenhao Liu, Muling Wu, Shihan Dou, Changze Lv, Xiaohua Wang, Xiaoqing Zheng, and Xuanjing Huang. 2025b. Revisiting jailbreaking for large language models: A representation engineering perspective. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 3158–3178, Abu Dhabi, UAE. Association for Computational Linguistics.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

Jimmy Lin, Xueguang Ma, Sheng-Chieh Lin, Jheng-Hong Yang, Ronak Pradeep, and Rodrigo Nogueira. 2021. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362.

Kehan Long, Shasha Li, Chen Xu, Jintao Tang, and Ting Wang. 2025. Precise zero-shot pointwise ranking with llms through post-aggregated global context information. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2384–2394.

Jian Luo, Xuanang Chen, Ben He, and Le Sun. 2024. Prp-graph: Pairwise ranking prompting to llms with graph aggregation for effective text re-ranking. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5766–5776.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In *Findings of the association for computational linguistics: EMNLP 2020*, pages 708–718.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Ronak Pradeep, Sahel Sharifymoghaddam, and Jimmy Lin. 2023. Rankvicuna: Zero-shot listwise document reranking with open-source large language models. *arXiv preprint arXiv:2309.15088*.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, and 1 others. 2024. Large language models are effective text rankers with pairwise ranking prompting. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 26 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522.

Devendra Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3781–3797.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. 2023. Role play with large language models. *Nature*, 623(7987):493–498.

Shuoqi Sun, Shengyao Zhuang, Shuai Wang, and Guido Zuccon. 2025. An investigation of prompt variations for zero-shot llm-based rankers. In *European Conference on Information Retrieval*, pages 185–201. Springer.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is ChatGPT good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937, Singapore. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Anyi Wang, Dong Shu, Yifan Wang, Yunpu Ma, and Mengnan Du. 2025a. Improving LLM reasoning through interpretable role-playing steering. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 731–751, Suzhou, China. Association for Computational Linguistics.

Noah Wang, Zy Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhan Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, and 1 others. 2024. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777.

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. 2025b. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, WWW '25,

page 2562–2578, New York, NY, USA. Association for Computing Machinery.

Yumeng Wang, Jirui Qi, Catherine Chen, Panagiotis Eustratiadis, and Suzan Verberne. 2026. How role-play shapes relevance judgment innbsp;zero-shot llm rankers. In *Advances in Information Retrieval: 48th European Conference on Information Retrieval, ECIR 2026, Delft, The Netherlands, March 29 – April 2, 2026, Proceedings, Part I*, page 228–242, Berlin, Heidelberg. Springer-Verlag.

Fred Zhang and Neel Nanda. 2023. Towards best practices of activation patching in language models: Metrics and methods. *arXiv preprint arXiv:2309.16042*.

Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. 2025. AdaSteer: Your aligned LLM is inherently an adaptive jailbreak defender. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 24559–24577, Suzhou, China. Association for Computational Linguistics.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024a. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. In *Proceedings of the 2024 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies (volume 2: short papers)*, pages 358–370.

Honglei Zhuang, Zhen Qin, Rolf Jagerman, Kai Hui, Ji Ma, Jing Lu, Jianmo Ni, Xuanhui Wang, and Michael Bendersky. 2023a. Rankt5: Fine-tuning t5 for text ranking with ranking losses. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2308–2313.

Shengyao Zhuang, Bing Liu, Bevan Koopman, and Guido Zuccon. 2023b. Open-source large language models are strong zero-shot query likelihood models for document ranking. *arXiv preprint arXiv:2310.13243*.

Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024b. A setwise approach for effective and highly efficient zero-shot ranking with large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 38–47.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, and 1 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

# A Preliminaries

## A.1 Pointwise ranking method

Given a query $q$ and a set of candidate documents $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$, the model is prompted with a binary instruction to judge the relevance of each pair $(q, d)$, such as: "Does the passage answer the query? Answer 'Yes' or 'No'." Let $Z_{\text{yes}}$ and $Z_{\text{no}}$ denote the raw logits assigned to the tokens "Yes" and "No" respectively, at the final generation position. The relevance score $s(q, d)$ is defined as the softmax-normalized probability of the token "Yes":

$$s(q, d) = \frac{\exp(Z_{\text{yes}})}{\exp(Z_{\text{yes}}) + \exp(Z_{\text{no}})} \tag{10}$$

The candidate documents are then ranked in descending order of $s(q, d)$.

## A.2 Activation Steering Framework

**Steering Vector Extraction.** The steering process begins by identifying a steering vector $v^{(l)} \in \mathbb{R}^d$ that captures a desired attribute at layer $l$. A common practice involves using contrastive prompt pairs. For a given concept (e.g., relevance judgment), let $X^+$ be a set of positive examples (query and relevant documents) and $X^-$ be a set of negative examples (query and irrelevant documents). The steering vector is computed as the mean difference between the hidden states $h_i^{(l)}$ of these pairs:

$$v^{(l)} = \frac{1}{n} \sum_{i=1}^{n} \left( h_{i+}^{(l)} - h_{i-}^{(l)} \right) \tag{11}$$

To ensure numerical stability and comparability across different layers, each steering vector is normalized to unit length such that $\|v^{(l)}\|_2 = 1$. In the context of pointwise LLM ranking, we start with a set of vectors $\mathcal{V} = \{v_{relevance}, v_{role}\}$. Here, $v_{\text{rel}}$ represents relevance-judgment-related representation differences, while $v_{\text{role}}$ captures role-induced behavioral biases.

**Inference-Time Intervention.** In this work, we focus on steering the last token hidden state, which typically aggregates the semantic information of the input sequence. For a transformer model with $L$ layers, the intervention is applied at every layer during the forward pass. A basic form of activation steering modifies the hidden state by adding a steering vector scaled by a coefficient $\alpha$:

$$\tilde{h}^{(l)} = h^{(l)} + \alpha \cdot v^{(l)} \tag{12}$$

While the steering vector is identified independently for each layer, its effect is accumulated as the modified signal propagates through the residual stream. A positive $\alpha$ shifts the model's internal state toward the target concept, thereby modulating the output logits toward the desired distribution. More generally, activation steering can be implemented through projection-based interventions, allowing fine-grained control over how hidden states are modulated along specific vectors.

# B Experimental details

## B.1 Role-play prompt

Following Wang et al. (2026)[1], we construct contrastive role-play prompts that share an identical task structure and differ only in three role-defining tokens (highlighted in red): an *adjective* describing the model's capability to perform the task, a *modal verb*, and an *adverb* guiding how the model executes the task. We design two role variants, *positive* and *negative*, which are intended to elicit good and degraded ranking behavior, respectively, providing a controlled behavioral contrast for steering vector extraction.

**Role-Play Prompt Template for Steering Vector Construction**

You are a {adjective} search assistant that {modal verb} rank passages {adverb}, based on their relevance to a query.
Passage: {passage}
Query: {query}
Does the passage answer the query?
Answer 'Yes' or 'No'.
Answer:

To ensure that each backbone exhibits a clear behavioral gap between the two role types, we adopt the model-specific lexical selections released by Wang et al. (2026), which were empirically verified to yield the expected behavioral contrast for each model. Table 4 lists the three positive and three negative role prompts used for each of our backbones (Llama-3.1-8B-Instruct, Qwen2.5-7B-Instruct, and Mistral-7B-Instruct-v0.3).

## B.2 Implementation Details.

For each of the $m = 5$ relevance anchor queries from TREC DL 2019, we take $n = 10$ relevant documents from the top of the base ranker's output and $n = 10$ irrelevant documents starting from rank 50. Choosing relevant documents from the top keeps the extracted vectors aligned with the model's own ranking behavior, while starting irrelevant selection at rank 50 avoids overly diverse documents that would weaken the extracted direction. Because anchor queries are randomly sampled from TREC DL 2019, we mitigate the resulting variance with a multi-fold selection procedure. We split DL 2019 into 9 partitions, where each of the first 8 partitions provides a distinct set of 5 anchor queries. For each partition, we compute the corresponding anchor vector and evaluate it on the validation queries. The vector achieving the highest validation gain is used in all subsequent experiments. The tuned steering coefficients $(\alpha, \beta, \gamma)$ are $(0.60, 0.16, 0.04)$ for Llama, $(0.25, -0.06, 0.08)$ for Qwen, and $(0.40, 0.00, 0.06)$ for Mistral. All experiments are conducted on a workstation equipped with four NVIDIA L40S GPUs (48GB memory each).

| Positive | Negative |
|---|---|
| **Llama-3.1-8B-Instruct** | |
| (brilliant, can, clearly) | (awful, will, terribly) |
| (logical, can, carefully) | (incorrect, will, falsely) |
| (superb, can, swiftly) | (unreliable, will, wrongly) |
| **Qwen2.5-7B-Instruct** | |
| (reliable, can, carefully) | (messy, will, mistakenly) |
| (expert, can, swiftly) | (problematic, will, poorly) |
| (capable, can, clearly) | (awful, will, incorrectly) |
| **Mistral-7B-Instruct-v0.3** | |
| (superb, can, clearly) | (flawed, will, incorrectly) |
| (expert, can, perfectly) | (messy, will, improperly) |
| (capable, can, carefully) | (clumsy, will, sadly) |

Table 4: Model-specific role token triplets (adjective, modal, adverb) used to construct contrastive role-play prompts, following Wang et al. (2026). Each model uses three positive and three negative triplets.

# C Baseline and Method Variant Details

## C.1 Pointwise Baselines

**QG** (Sachan et al., 2022) scores documents based on query generation likelihood. **RG-YN** (Liang et al., 2022) uses the probability of generating "Yes" relative to "No" as the relevance score. **RG-S(0–k)** (Zhuang et al., 2024a) extends RG-YN by prompting the model to rate relevance on a discrete scale; following prior work, we set $k = 4$. **PAGC-**

[1]Available at https://github.com/menauwy/roleplay_zeroshot

**YG** (Long et al., 2025) is a post-aggregation technique combining GCCP (global-consistent pairwise comparisons in a pointwise framework) with RG-YN. **RefRank-(4)** (Li et al., 2025a) reformulates pointwise ranking as an A/B comparison: given a query and a candidate passage, the model compares the candidate against a fixed set of top $K$ ranked reference passages and aggregates the results. We set $k = 4$ following the best practice of the paper. **RG-YN_role** (Wang et al., 2026) augments RG-YN with a neutral role-play description from appendix B.1 but with the three role-defining tokens (adjective, modal, adverb) removed.

### C.2 Comparative Baselines

**PRP** (Luo et al., 2024) produces relative relevance labels for document pairs and aggregates them into a ranking via heapsort, returning the top $k = 10$. **Setwise** (Zhuang et al., 2024b) replaces each pairwise judgment with a setwise comparison over a small group of passages, using generation-based scoring and heapsort with branching factor `num_child=2`, a single permutation, and top $k = 10$. **RankGPT** (Sun et al., 2023) generates a global ordering via sliding-window listwise prompting (window size 4, step size 2) over the full candidate list, repeated for 5 passes with generation-based scoring.

### C.3 RankSteer-Ref: Implementation Details

RankSteer-Ref is built on the reference-based RefRank (Li et al., 2025a) base and follows the same activation-steering recipe as RankSteer but drops the role-play dimension and keeps only two: a decision direction and an orthogonalised evidence direction. Because RefRank scores candidates with a pairwise prompt:

**Prompt Template for RefRank and RankSteer-Ref**

Given a query {query}, which of the following two passages is more relevant to the query?
A: {passage}
B: {reference passage}
Output A or B:

the decision direction is taken as the A-vs-B unembedding margin from the LM head rather than the pointwise yes/no margin, and anchors are built in the same A/B layout so the geometry aligns with inference. The evidence vector is extracted once using the BM25 top-1 reference and is then reused across all references when RefRank runs in multi-reference (top-$k$) mode.

## D Effectiveness and efficiency with non-pointwise methods

Table 5 compares RankSteer with representative pairwise, setwise, and listwise ranking methods, and BM25.

While comparative methods such as pairwise and listwise prompting generally achieve stronger effectiveness by explicitly introducing document comparisons, RankSteer operates under a strictly pointwise setting without access to cross-document information. As a result, it is not expected to consistently outperform listwise methods. The relevant question is how much of the gap can be closed by post-hoc calibration alone. In particular, RankSteer achieves performance comparable to strong comparative methods(listwise) on TREC DL 2020 for all backbones. On BEIR datasets, where the data is out-of-distribution, RankSteer on average stays close to setwise on both Llama (RankSteer-Ref 0.4600 vs. 0.4608) and Qwen (RankSteer-Ref 0.4814 vs. 0.4830). On some BEIR datasets, RankSteer even attains the best score of any method, such as robust04 and nfcorpus on Llama. The two instantiations behave differently, consistent with Section 4.2.1. RankSteer-Ref is the more robust variant against comparative baselines. RankSteer-Role is competitive only when its role-prompted base is strong.

From a computational perspective, RankSteer retains the same lowest time complexity as standard pointwise ranking, incurring only a constant-time ($O(1)$) for activation steering. In contrast, comparative methods require additional document interactions, leading to $O(N)$ complexity for listwise approaches and $O(\log N)$ for heap-based pairwise and setwise methods.[2] Together, these results suggest that a significant portion of the performance gap between pointwise and comparative methods stems from how internal relevance signals are utilized and scaled, rather than from the absence of explicit cross-document comparisons. RankSteer therefore offers a favorable trade-off, recovering much of this gap while preserving the simplicity,

[2] (Long et al., 2025) distinguish between time complexity and number of LLM calls in the analysis of the complexity of ranking methods. Here we focus on time complexity.

| | | TREC DL | BEIR | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **Methods** | **dl20** | **covid** | **touche** | **signal** | **news** | **scifact** | **robust04** | **dbpedia** | **nfcorpus** | **Avg** |
| | BM25 | 0.4796 | 0.5947 | 0.4422 | 0.3304 | 0.3952 | 0.6789 | 0.4070 | 0.3180 | 0.3218 | 0.4360 |
| Llama-3.1-8B | pairwise | 0.6098 | 0.7835 | 0.2578 | 0.2891 | 0.4700 | **0.7032** | 0.5115 | 0.3800 | 0.3658 | 0.4701 |
| | setwise | 0.6112 | 0.7789 | 0.2470 | 0.2946 | 0.4614 | 0.6808 | 0.4965 | 0.3774 | 0.3494 | 0.4608 |
| | listwise | **0.6587** | **0.8112** | **0.2744** | **0.3107** | **0.4833** | 0.6762 | 0.5201 | **0.4189** | 0.3690 | **0.4830** |
| | RankSteer-Ref | 0.5936 | 0.7942 | 0.2443 | 0.2893 | 0.4660 | 0.6561 | 0.5126 | 0.3598 | 0.3575 | 0.4600 |
| | RankSteer-Role | 0.6461 | 0.7863 | 0.2445 | 0.2779 | 0.4462 | 0.6498 | **0.5207** | 0.3752 | **0.3732** | 0.4588 |
| Mistral-7B | pairwise | 0.5559 | 0.7966 | 0.3099 | 0.3036 | 0.4854 | **0.6655** | 0.4981 | 0.3354 | 0.3541 | 0.4696 |
| | setwise | 0.6171 | **0.8160** | 0.2879 | 0.2919 | 0.4520 | 0.6260 | 0.4813 | 0.3953 | 0.3615 | 0.4640 |
| | listwise | **0.6491** | 0.8006 | **0.3521** | **0.3268** | **0.5092** | 0.6584 | **0.5359** | **0.4425** | **0.3713** | **0.4996** |
| | RankSteer-Ref | 0.5407 | 0.7729 | 0.2965 | 0.2849 | 0.4432 | 0.6127 | 0.4892 | 0.3375 | 0.3422 | 0.4474 |
| | RankSteer-Role | 0.6355 | 0.8019 | 0.2254 | 0.2877 | 0.4441 | 0.5664 | 0.5123 | 0.3965 | 0.3623 | 0.4496 |
| Qwen2.5-7B | pairwise | 0.6590 | 0.7989 | 0.3038 | 0.3111 | 0.4968 | 0.6968 | 0.5415 | 0.4134 | 0.3796 | 0.4927 |
| | setwise | 0.6531 | 0.8099 | 0.2883 | 0.2999 | 0.4788 | 0.6800 | 0.5210 | 0.4074 | 0.3786 | 0.4830 |
| | listwise | **0.6785** | **0.8234** | **0.3184** | **0.3363** | **0.4989** | **0.7107** | **0.5438** | **0.4324** | **0.3905** | **0.5068** |
| | RankSteer-Ref | 0.6475 | 0.7925 | 0.2746 | 0.3017 | 0.4913 | 0.6778 | 0.5226 | 0.4148 | 0.3759 | 0.4814 |
| | RankSteer-Role | 0.6594 | 0.7678 | 0.2356 | 0.2793 | 0.3807 | 0.4133 | 0.4338 | 0.3749 | 0.3697 | 0.4069 |

Table 5: Comparison of our RankSteer method to other ranking methods, in terms of nDCG@10. The best result per dataset and LLM is marked with **boldface**; underline indicates the second-best result.

efficiency, and full parallelism of pointwise inference.

## E Query-level document geometry

In Figure 7, we present the layer-wise steering effect on the document representations of a random query from TREC DL 2019, using RankSteer-Role on Llama-3.1-8B-Instruct. Across layers, we observe three consistent patterns. First, the *ranking line*, a single linear axis along which all 100 candidate documents organize in the (Evidence, Decision) plane, is absent in early layers. There, the projections form an isotropic cloud with low slope and low $R^2$. The line emerges around L13 and is stably preserved from roughly L16 onward, indicating that ranking is a mid-to-late-layer geometric phenomenon implemented in the residual stream. Second, after steering, the slope of this line consistently decreases at nearly every well-defined layer. This is the signature of RankSteer: it reduces the ranking's reliance on the raw decision readout, so that the ordering depends less steeply on it while the documents remain on the line. Third, despite the decrease of slope, $R^2$ is largely retained, and is slightly improved in several mid layers. This means steering re-weights documents within the Evidence–Decision subspace without introducing variance orthogonal to the ranking line.

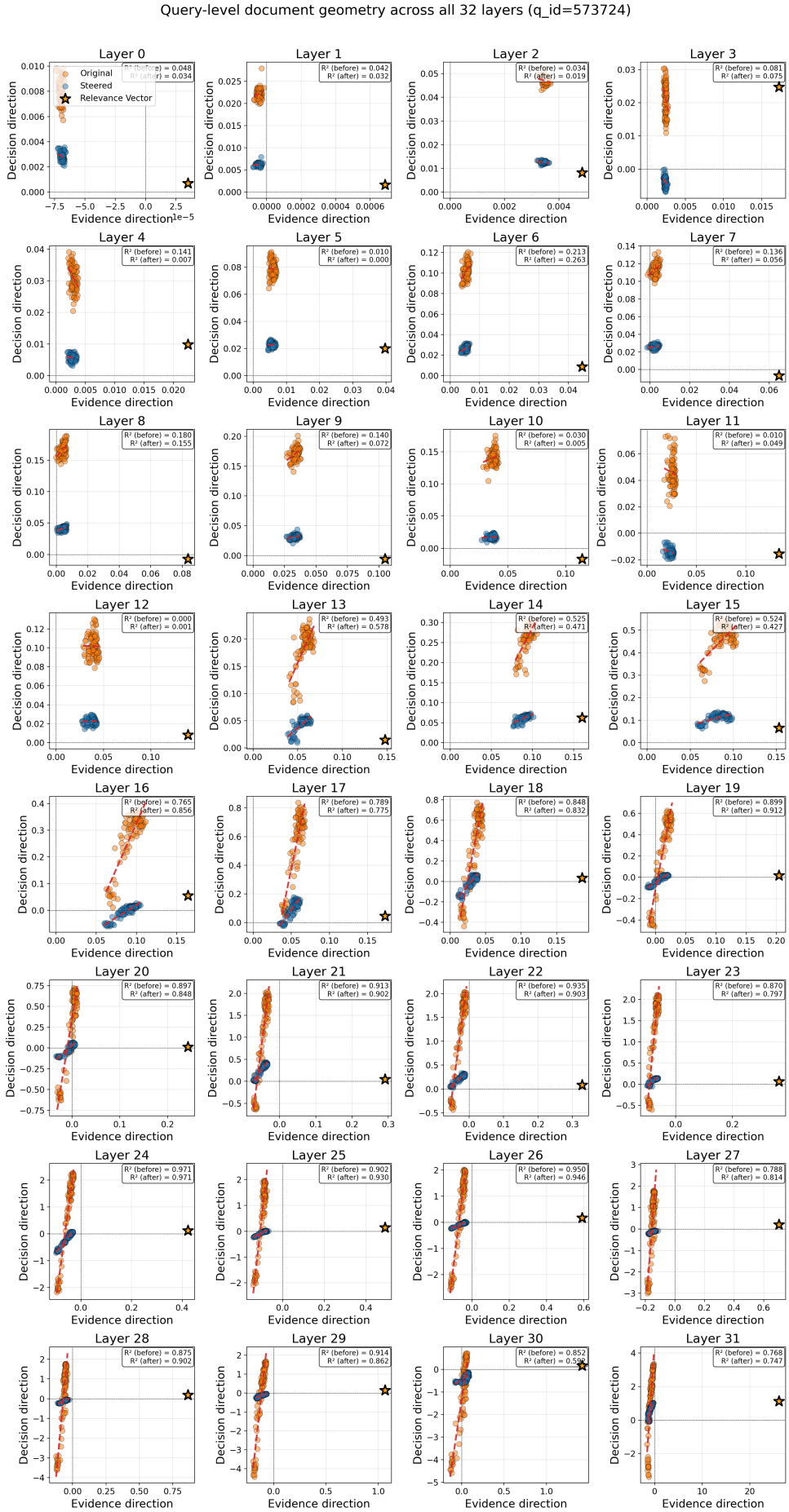


Figure 7: Query-level document geometry across all 32 layers (q = 573724, TREC DL-2019) using RankSteer-Role.