DIRECT HIDDEN-STATE ALIGNMENT: MAPPING AND CONTROLLING PREFERENCE EXPRESSION IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

In many settings, post-training need not create the target behavior from scratch: the base model can already produce it, but not reliably. This shifts part of preference alignment from capability acquisition to behavioral expression. This raises a central question: if target-supporting computation is already present, what prevents the preferred behavior from reliably dominating generation? At each generation step, many Transformer components write to the same residual stream, yet their effects are combined into a single next-token distribution. A target-supporting computation can therefore be present yet be outweighed by other computations. We hypothesize that reliable expression depends on this internal competition during inference. We introduce Residual Competition Maps (RCMs), which map a specified preference onto native residual computation by measuring signed causal effects relative to that preference. Across preference domains, RCMs reveal residual competition whose prevalence varies by task, distinct component roles relative to the target preference, and cases where a single native-component intervention reverses the preference outcome. DPO substantially reorganizes these effects and can weaken opposing effects, which may nevertheless persist. Beyond analysis, RCM provides causal guidance for where to intervene. To directly control preference formation during inference, we propose Direct Hidden-State Alignment (DHSA), which treats inference-time hidden states rather than base-model weights as the direct adaptation space. RCM-guided Causal Activation State Transition (CAST) instantiates this principle through local state interventions at a small number of preference-relevant interfaces while freezing the base model. With only 256–16,384 controller parameters, CAST reaches DPO-competitive operating points across three preference domains, can complement DPO-trained models, and can be enabled or removed at inference time.

1 INTRODUCTION

Preference alignment aims to make desired behavior reliable (Ouyang et al., 2022; Rafailov et al., 2023). Yet being able to generate a target response is different from producing it reliably. LIMA shows that limited instruction tuning can elicit high-quality outputs, suggesting that alignment can draw on knowledge acquired during pretraining (Zhou et al., 2023). Recent work on reinforcement learning with verifiable rewards (RLVR) reports substantial gains in single-sample success (`pass@1`) after post-training, while base models can achieve higher `pass@k` at larger k (Yue et al., 2025). Here, `pass@k` measures whether at least one of k attempts succeeds. These observations distinguish *behavioral reachability* from *reliable expression*, as illustrated in Figure 1(a). **When a target behavior is already reachable, why can it still fail to be expressed reliably, and can this failure be corrected directly?**

To understand this gap, we model generation at the level of the model's native residual computation. Attention heads, feed-forward modules, and other components write additive updates to a shared residual stream (Vaswani et al., 2017; Elhage et al., 2021). At each generation step, these contributions combine into a single next-token distribution. Target-supporting computation can therefore already be present yet fail to dominate the prediction when other native computations outweigh its effect. This motivates our hypothesis of *internal competition during inference*: target-supporting and target-competing native effects can coexist as a prediction is formed, and their relative influence can affect which behavior is expressed.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

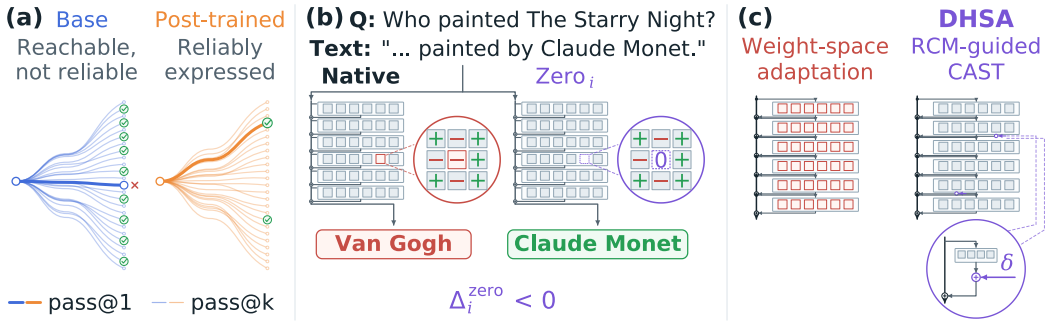

Figure 1: **Overview of RCM-guided direct hidden-state alignment.** (a) A schematic sampling contrast illustrates the gap between behavioral reachability and reliable expression: bold paths denote single draws, while faint trajectories denote candidates exposed under a larger sampling budget; trajectory counts are illustrative rather than empirical `pass@k` measurements. (b) RCM maps an explicit behavioral preference onto signed causal effects of native component computation. Opposing signed effects on the same input form the residual competition studied here. The schematic example shows a generated preference outcome changing after a singleton native-component intervention. (c) Model-wide parameter adaptation updates weights across the model. In contrast, DHSA directly adapts inference-time hidden states. RCM-guided CAST implements this principle at a small number of residual interfaces while keeping the base weights frozen.

We use *residual competition* for this measurable coexistence of opposing native effects. For the same input and preference, some components can causally raise the target preference while others lower it, as illustrated in Figure 1(b). These effects are defined relative to the specified preference; they do not assume that component effects form an additive decomposition of the model output. Under this view, the problem becomes three sequential questions. **First, how can an external behavioral preference be mapped onto native model computation so that component effects become measurable relative to it? Second, what internal structure of preference expression does this mapping reveal? Third, can this internal structure directly guide control of which behavior is expressed?**

The first question requires more than identifying generally important components. A component can strongly affect generation without revealing whether it supports or opposes the preference being aligned. Causal tracing and activation patching have shown how interventions on internal states can connect model components to their outputs (Meng et al., 2022; Zhang & Nanda, 2024). We therefore introduce *Residual Competition Maps* (RCMs), a component-level causal framework that maps an explicit behavioral preference onto native model computation. For each input, RCM uses single-component causal contrasts to measure how each component shifts the target behavior's advantage over alternatives. The resulting signed effect captures both the direction and strength of its preference-relative contribution. Figure 1(b) illustrates this mapping. Because these measurements are causal, RCM can be used not only to analyze preference expression, but also to guide where to intervene.

The resulting measurements reveal residual competition across heterogeneous preference settings, with its prevalence varying across settings. In representative cases, a single native-component intervention reverses the generated preference outcome. RCMs further show that Direct Preference Optimization (DPO) substantially reorganizes these native effects and can weaken opposing effects, without guaranteeing their removal. This extends prior mechanistic evidence that preference post-training reshapes internal computation (Lee et al., 2024): improved preference behavior need not eliminate opposing native effects.

Once this structure is measurable, it also provides a direct route to control. Prior work shows that inference-time state interventions can steer model behavior without updating base weights (Rimsky et al., 2024; Wu et al., 2024). To directly control how a preference is formed during inference, we act on the hidden states that carry this computation while keeping the base-model weights fixed. We call this broader principle *Direct Hidden-State Alignment* (DHSA).

We instantiate DHSA with RCM-guided *Causal Activation State Transition* (CAST). RCM identifies preference-relevant residual interfaces, and CAST learns local state changes at a small number of these sites (Figure 1(c)). Across settings, task-specific scores and controller configurations instantiate the same RCM-guided state-control principle. This gives a low-footprint form of alignment without reparameterizing the base model. Across heterogeneous preference settings and model families, CAST reaches DPO-competitive operating points with only 256–16,384 controller parameters. The controllers remain separate from the base weights, can be enabled or removed at inference time, and can also improve DPO-trained models.

Our main contributions are:

1. **Component-level preference mapping.** We introduce RCMs, which map an external behavioral preference onto signed causal effects of native model computation through single-component contrasts. RCM measures whether and how strongly each component shifts the target behavior's advantage over alternatives, providing both an analysis tool and causal guidance for intervention.
2. **Residual competition and preference-relative component roles.** Across heterogeneous preference settings, RCMs reveal coexisting target-supporting and target-competing native effects and input-dependent component roles. In representative cases, a single native-component intervention reverses the generated preference outcome. DPO substantially reorganizes these effects and can weaken opposition without guaranteeing its removal.
3. **Direct control of preference formation.** We introduce DHSA to directly adapt the hidden states through which preferences are expressed during inference, while keeping base-model weights frozen. RCM-guided CAST provides a simple low-footprint implementation at a small number of preference-relevant interfaces. With 256–16,384 controller parameters, CAST reaches DPO-competitive operating points across the studied settings, can further improve DPO-trained models, and remains detachable at inference time.

2 Related Work

2.1 Preference Alignment and Behavioral Expression

Preference alignment adapts language models using feedback about desired behavior. RLHF optimizes a policy with learned preference rewards (Ouyang et al., 2022), while DPO learns directly from preference pairs (Rafailov et al., 2023). Beyond optimization algorithms, task-complexity analyses examine the additional information needed to access pretrained capabilities (Vergara-Browne et al., 2026). RLVR studies investigate sampling coverage (Yue et al., 2025), while reasoning-aware evaluations assess intermediate-step correctness and report expanded reasoning capabilities (Wen et al., 2026). These perspectives examine different aspects of post-training, from behavioral accessibility to reasoning quality. Our work focuses on behavioral expression: how a specified preference appears in native model computation, how post-training reorganizes that computation, and whether the same internal structure can guide direct control during inference.

2.2 Mechanistic Interpretability and Behavioral Attribution

Causal tracing, activation patching, and circuit analysis connect model behavior to internal components (Meng et al., 2022; Wang et al., 2022). Benchmarks and methodological studies examine the causal efficacy of interpretability methods and the sensitivity of localization to evaluation choices (Arora et al., 2024; Zhang & Nanda, 2024). Studies of refusal identify a low-dimensional control direction (Arditi et al., 2024), while mechanistic analysis of DPO finds that toxicity reduction can bypass pretrained capabilities (Lee et al., 2024). GCM uses contrastive-response likelihoods to select causal mediators for sparse steering (Sankaranarayanan et al., 2026); Preference Heads combines causal head masking with contrastive decoding for personalization (Zhang et al., 2026). Subspace-patching studies distinguish successful behavioral manipulation from faithful attribution (Makelov et al., 2024). These studies connect causal attribution with behavioral control. RCM complements this line by mapping a specified preference onto signed causal effects of native computation, supporting both mechanistic analysis and intervention guidance.

2.3 Representation-Based Control and Alignment

Activation engineering and representation engineering develop internal representations as behavioral control interfaces (Turner et al., 2023; Zou et al., 2023). ITI and CAA demonstrate inference-time steering through localized or contrastively constructed activation interventions (Li et al., 2023; Rimsky et al., 2024). Conditional Activation Steering, also abbreviated CAST, gates steering using input-dependent activations (Lee et al., 2025); our CAST learns state changes at RCM-selected interfaces. ReFT learns interventions on frozen-model representations, with LoReFT using low-rank parameterization; LoFiT learns task-specific attention-head selection followed by additive offsets at selected heads (Wu et al., 2024; Yin et al., 2024). Preference-oriented approaches further extend representation control to alignment: RAHF manipulates preference-related representations (Liu et al., 2024), RE-Control formulates alignment as state-space control (Kong et al., 2024), and BiPO learns steering vectors from pairwise preferences (Cao et al., 2024). PaLRS extracts training-free residual steering vectors from preference pairs and reports performance competitive with or better than DPO and SimPO (Meng et al., 2024) on mathematical reasoning and code generation (La Cava & Tagarelli, 2026). These studies establish that representation-level control can support preference alignment and can reach strong alignment baselines.

We use Direct Hidden-State Alignment (DHSA) to frame preference adaptation at the level of inference-time hidden states of a frozen base model. RCM provides preference-guided localization by identifying preference-relevant native effects, and CAST applies local state changes at a small number of those interfaces. We evaluate this RCM-guided realization of DHSA across heterogeneous preference settings.

3 Method

3.1 How Preference Appears in Token Predictions Formed by Component Writes

In the residual language models studied here, the residual stream carries the shared state used to predict each next token. Attention and MLP components read from this stream and write their outputs back to it. Later components read the updated state. The state at the current prediction position is ultimately mapped to the next-token distribution (Vaswani et al., 2017; Elhage et al., 2021).

Let p be a token position and $r_p \in \mathbb{R}^d$ its residual state before a component write. For component i, let $h_{p,i} \in \mathbb{R}^{d_i}$ be its native (unintervened) output and $P_i \in \mathbb{R}^{d \times d_i}$ its fixed write-back map. The component changes the shared state by

$$z_{p,i} = P_i h_{p,i}, \qquad r_p \leftarrow r_p + z_{p,i}. \tag{1}$$

For an attention head, P_i is its slice of the attention output projection. Through the residual stream, native component writes affect later computation and the tokens the model produces.

Each experiment specifies a preference e: a target behavior relative to a non-target behavior. We quantify its expression using established, widely used forms of task scoring. Write the resulting target advantage for model computation M on input x as $C_e(M; x)$, with larger values favoring the target. Appendix A.1 specifies its implementation for each task in our experiments.

The score connects the external preference to the model's output. Component writes connect native computation to that output. RCM maps the preference onto the causal effects of these writes.

3.2 How RCM Maps Preference to Native Component Effects

RCM uses a single component as its unit of comparison and evaluates the resulting change with the specified preference score.

For component i, let a and b specify two state conditions at the intervention positions. Let M_i^a and M_i^b be the resulting computations, including all downstream changes. For the same input x, RCM measures

$$\Delta_{i,e}^{a \to b}(x) = C_e(M_i^b; x) - C_e(M_i^a; x). \tag{2}$$

Repeating this single-component comparison maps the externally specified preference onto signed causal effects of native component states.

We use two instances of this general component-state contrast in our experiments. RCM-Zero compares a zeroed write with its native write. A positive effect means the native component raises the target advantage; a negative effect means it lowers it. Figure 1(b) illustrates this native-versus-intervened contrast. RCM-Patch compares the receiving state with a specified source state. It measures what that state replacement does to the same advantage. Appendix B.1 specifies the source conditions.

The signs give components a preference-relative meaning. For one input, *residual competition* occurs when scanned components include both target-raising and target-lowering RCM-Zero effects above the stated thresholds. Across inputs, we name components with positive mean effects *target-supporting*, those with negative mean effects *target-competing*, and those with substantial effects of both signs *bidirectional regulatory*. These descriptive roles can overlap. Appendix B.2 gives the role criteria.

3.3 How Native Effects Guide Direct Hidden-State Control

An RCM effect identifies a component state whose change causally moves the measured target advantage. Its location and sign describe how the model's native computation supports or opposes the specified preference. Residual competition further shows where opposing effects coexist. The resulting map therefore provides a causal basis for direct control of preference expression.

Following the DHSA principle introduced in Section 1, we instantiate this control with RCM-guided *Causal Activation State Transition* (CAST). CAST learns local changes to component states at locations selected using RCM. A selection may use target-supporting locations, target-competing locations, or both. In the experiments, we scan coarse components before attention heads and use effect estimates with confidence intervals to rank candidates. Appendix A.2 gives the tested selection rules and budgets.

Let $\mathcal{G}$ be the selected locations. For $i \in \mathcal{G}$, let $h_{p,i}^{\mathrm{ctrl}} \in \mathbb{R}^{d_i}$ be the component state in the current controlled forward pass. The controller $g_i(\,\cdot\,;\theta_i) : \mathbb{R}^{d_i} \to \mathbb{R}^{d_i}$ returns a state change, with trainable parameters θ_i. The mask $m_p \in \{0,1\}$ selects token positions, and $\alpha \geq 0$ sets the intervention strength. CAST applies the resulting state change before the residual write:

$$\widetilde{h}_{p,i} = h_{p,i}^{\mathrm{ctrl}} + \alpha m_p \, g_i(h_{p,i}^{\mathrm{ctrl}};\theta_i), \qquad \widetilde{z}_{p,i} = P_i \widetilde{h}_{p,i}. \tag{3}$$

Here $\widetilde{h}_{p,i}$ is the transformed component state and $\widetilde{z}_{p,i}$ is its residual write. The local state change in Figure 1(c) corresponds to $\alpha m_p \, g_i(h_{p,i}^{\mathrm{ctrl}};\theta_i)$. In our experiments, the controller is either a constant vector $g_i(h;\theta_i) = v_i$, with $v_i \in \mathbb{R}^{d_i}$, or a low-rank map $g_i(h;\theta_i) = B_i A_i h/\sqrt{k_i}$. The latter has rank k_i, $A_i \in \mathbb{R}^{k_i \times d_i}$, and $B_i \in \mathbb{R}^{d_i \times k_i}$. The model then continues its forward computation from the transformed states. When several locations are selected, we evaluate their joint effect on the model's output.

We denote the constant-vector and low-rank variants by CAST-SV and CAST-LR. The `prefill` configuration acts at registered prompt positions; `all` also acts during decoding. Appendix A.4 specifies their training and inference masks.

3.4 How the Control Change Is Learned

Let θ collect the trainable controllers at the selected locations, and let ω denote the fixed base-model weights. CAST specifies how θ changes inference states; the learning rule is separate from the state transformation. In this paper, we evaluate a reference-relative pairwise objective of the DPO form. The same learning form is used across the evaluated tasks; only the task-specific response score and controller configuration differ.

Our experiments use preference pairs (x, y^+, y^-). Training increases the preferred response's score relative to the non-preferred response and to the same model with control disabled. The reference-relative pairwise loss, response scores, and regularization are specified in Appendices A.3 and A.1. Appendix D.1 derives the relation of this experimental loss to DPO's pairwise form (Rafailov et al., 2023).

3.5 Direct Hidden-State Alignment

DHSA describes the broader adaptation principle behind CAST: preference alignment is performed by transforming hidden states during the model's inference computation while keeping the base-model weights fixed.

Let H_t denote a collection of hidden states reached while the model predicts token t, and let ϕ parameterize a state transformation. DHSA acts as

$$\widetilde{H}_t = T_\phi(H_t; x, y_{<t}). \tag{4}$$

The model continues its computation from $\widetilde{H}_t$ to produce the next-token distribution. The transformation can act at different states, positions, and granularities. CAST is the sparse RCM-guided instance studied here: it restricts the transformation to a small number of preference-relevant interfaces while leaving the base weights unchanged. Appendix D.2 analyzes when a local hidden-state change improves a preference objective.

4 Experiments

We first use RCM to characterize preference-relative native effects, then test whether the resulting maps guide localized control, and finally evaluate RCM-guided hidden-state alignment across three preference settings. [1]

4.1 Experimental Setup

Tasks and models. The tasks cover distinct demands on preference control. ConFiQA (Bi et al., 2025) simulates retrieval-augmented factual answering under conflict between supplied evidence and parametric knowledge; PC measures target-answer occurrence, and EM requires the entire normalized output to match a context-supported answer alias. IMDb (Maas et al., 2011) tests the precision of sentiment alignment: reward measures movement toward the desired sentiment, alongside measures of distribution shift. TL;DR tests an open-generation strategy in which the model must select and compress content into a preferred summary (Völske et al., 2017; Stiennon et al., 2020). Appendix C.5 gives the evaluation definitions. Across tasks, preference scores and controller configurations instantiate the same RCM-guided state-alignment principle.

Site experiments use Llama-3-8B and Qwen2.5-14B on ConFiQA, GPT-2-large and Qwen2.5-14B on IMDb, and GPT-J-6B and Qwen2.5-14B on TL;DR. Performance experiments use Llama-3-8B and Qwen2-7B on ConFiQA, GPT-2-large on IMDb, and GPT-J-6B and LION-Llama-3-8B on TL;DR. Starting checkpoints are instruction-tuned or SFT models.

Site test sizes are 2,048 per ConFiQA subset for Llama-3-8B; 500/478/477 for Qwen2.5-14B QA/MR/MC; 2,048 per IMDb model; and 2,048/320 for GPT-J/Qwen TL;DR. Performance uses 2,048 mixed ConFiQA inputs per model, 2,048 IMDb prompts, and 512 TL;DR inputs per model. TL;DR preference is judged against human references and averaged over both presentation orders.

Comparisons and budgets. Released DPO checkpoints provide representative preference-alignment references. DPO serves as an evaluation-matched reference; training data and compute are not matched across methods. BiPO and LoReFT use their released implementations and native objectives. Candidate-training opportunities, active parameter counts, and available runtime measurements are reported separately. Site compares RCM-Zero, RCM-Patch, ITI, and three fixed random selectors under the registered selection and deployment protocols. Validation data select intervention strength where a single operating point is reported; IMDb retains the measured reward–KL curves. Performance reports both `all` and `prefill` control. Full settings and costs appear in Appendices A–C.

4.2 Residual Competition and Component Reorganization

We measure RCM-Zero effects for 32 Llama-3-8B attention-layer components on 300 ConFiQA inputs. Each effect is the native-minus-ablated change in paired target–non-target token-logit margin.

[1] `https://anonymous.4open.science/r/anonymous-research-materials-CEF6/`

An input has bilateral effects at threshold τ when one component exceeds τ and another falls below $-\tau$; τ is an absolute change in margin units.

At $\tau = 0.5$, bilateral effects occur in 283/300 inputs (Figure 2(a)). Target output need not imply uniformly target-supporting component effects: Appendix B.3 gives native target answers with opposing effects, and Appendix B.4 gives actual decoded changes under single-head ablation. Figure 2(b) summarizes signed means; Appendix B.2 distinguishes them from substantial per-input effects that cancel in the mean.

DPO changes layer effects by 1.00 margin units on average in absolute value (95% CI: 0.97–1.04), and 8/32 layers change mean-effect sign (Figure 2(c)). Yet 11 layers retain negative mean effects, 8 with pointwise 95% intervals below zero. DPO therefore substantially reorganizes the measured effects without guaranteeing the removal of target-opposing computation.

Appendix B.2 extends the analysis to IMDb and TL;DR. Bilateral effects occur in 28/300 IMDb inputs at $\tau = 0.01$ and 289/300 TL;DR inputs at $\tau = 0.02$, in each task's own score units. The prevalence of competition therefore differs across tasks.

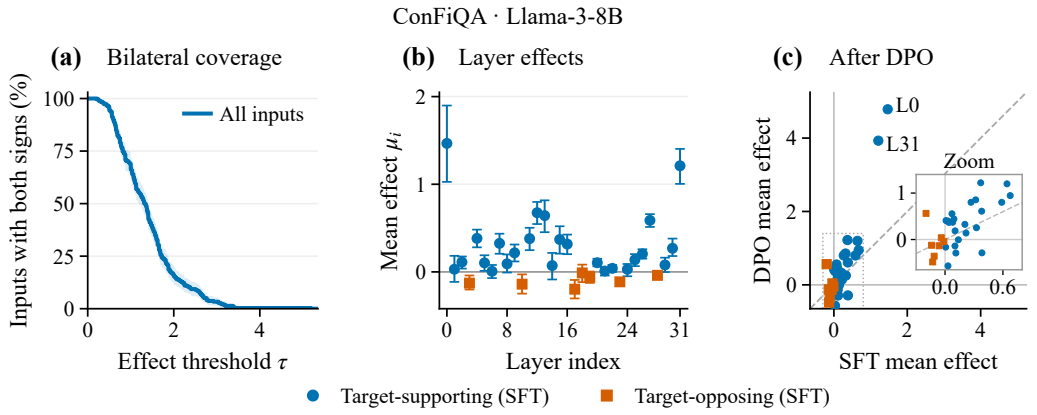

Figure 2: **Residual competition in ConFiQA.** RCM-Zero effects across 32 Llama-3-8B attention layers on 300 inputs. (a) Bilateral coverage against the absolute token-logit-margin threshold. (b) Mean effects with 95% confidence intervals. (c) Paired SFT–DPO mean effects. Colors and markers indicate the SFT mean-effect sign.

4.3 Does RCM Guide Effective Localized Control?

Site experiments test whether preference-relative RCM measurements provide useful locations for subsequent hidden-state control. The unit of comparison is the registered localization-and-control configuration, including each selector's bank organization. Vector configurations use RCM's separate signed banks, whereas ITI and Random use their registered joint banks. For TL;DR, all selectors receive the same single-head training and audit procedure before eight-head composition (Appendix C.3).

Table 1 shows that RCM-Zero exceeds every random selector across both ConFiQA and TL;DR model families. ITI leads on ConFiQA Llama-3-8B, whereas RCM-Zero leads on ConFiQA Qwen2.5-14B and both TL;DR models. The TL;DR advantage is particularly clear; RCM-Patch is less consistent across models. RCM measurements therefore provide actionable localization, with the strongest selector depending on the model and setting.

	ConFiQA PC ↑		TL;DR Win ↑	
Localization	Llama-3-8B	Qwen2.5-14B	GPT-J-6B	Qwen2.5-14B
Random 1	63.74	61.35	45.17	48.59
Random 2	62.48	68.30	50.73	48.59
Random 3	66.08	74.89	46.83	48.28
ITI	**85.64**	75.15	46.83	50.16
RCM-Patch	80.40	72.75	53.88	48.91
RCM-Zero	78.61	**83.53**	**54.64**	**56.09**

Table 1: **Site results** (%). ConFiQA reports macro-average PC across QA, MR, and MC; TL;DR reports order-balanced preference against human references. All three fixed random selectors are shown.

IMDb. Figure 3(a) tests whether RCM-guided localization improves sentiment reward without excessive sequence-level shift. The same signed RCM candidate rule identifies weak competing-side effects in this setting. The main CAST-SV-all efficiency comparison therefore deploys only the four target-supporting heads; the candidate-selection rule itself is unchanged. Appendix C.7 reports the additional Qwen2.5-14B Site result.

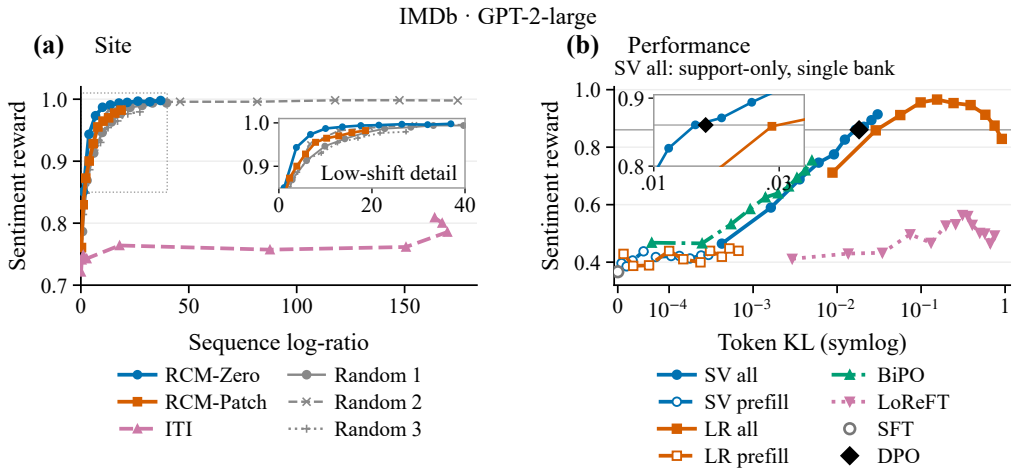

Figure 3: **IMDb localization and control efficiency.** GPT-2-large, 2,048 test prompts per protocol. (a) Site reward–sequence-log-ratio curves. (b) Performance reward–token-KL curves. SV/LR denote CAST's vector/low-rank controllers; LoReFT is the external baseline. SV-all uses the supporting-only bank; the black diamond denotes DPO.

4.4 Can Direct Hidden-State Alignment Reach DPO-Competitive Operating Points?

We next ask whether DHSA can reach the behavioral regime of preference post-training across distinct preference settings. Table 2 compares RCM-guided CAST with DPO, BiPO, and LoReFT. CAST uses the task-appropriate controller form specified in Section 3.3: constant vectors on ConFiQA and low-rank interventions on TL;DR. We also test controllers transferred from SFT to DPO (**T**) and controllers trained directly on DPO (**NT**).

ConFiQA: factual inference under conflict. CAST enters the DPO-competitive range on context-supported answering, but PC and EM expose different outcomes. On Llama-3-8B, prefill control improves exact answers beyond DPO while remaining lower on PC; all-position control raises PC but fails on EM. On Qwen2-7B, CAST-all exceeds DPO on both metrics, while BiPO and LoReFT remain stronger. The contrast also shows that intervention timing changes how the same preference is expressed.

	ConFiQA				TL;DR	
	Llama-3-8B		Qwen2-7B		GPT-J	LION
Method	PC ↑	EM ↑	PC ↑	EM ↑	Win ↑	Win ↑
SFT	59.42	3.71	39.26	0.15	44.43	71.39
DPO	88.92	43.99	79.30	3.86	62.11	84.67
BiPO	81.98	70.80	84.91	75.15	51.86	78.03
LoReFT	82.96	52.05	89.94	77.25	47.07	67.48
CAST (all)	81.93	0.00	82.96	16.31	61.72	79.10
CAST (prefill)	82.52	64.65	66.75	24.95	52.83	82.81
DPO+CAST-T (all)	88.92	47.95	89.31	39.84	68.55	85.55
DPO+CAST-T (prefill)	90.38	39.31	83.50	39.50	65.04	87.01
DPO+CAST-NT (all)	88.33	57.23	83.89	8.59	63.96	82.62
DPO+CAST-NT (prefill)	90.53	55.52	82.28	27.49	61.82	84.67

Table 2: **Preference-alignment performance** (%). CAST uses constant vectors on ConFiQA and low-rank controllers on TL;DR. TL;DR Win is against human references, not directly against DPO. T transfers an SFT-trained controller to DPO and recalibrates α without retraining; NT trains on the DPO checkpoint.

IMDb: precision of sentiment alignment. Figure 3(b) compares sentiment reward at the accompanying token KL. The target-supporting CAST bank reaches the DPO reward with lower measured KL (approximately 0.0166 versus 0.0183 by interpolation). BiPO and LoReFT do not reach that reward within their evaluated grids. In this setting, localized hidden-state adaptation reaches the DPO reward at a smaller measured token KL.

TL;DR: open-generation strategy. CAST approaches DPO on both model families and exceeds the corresponding BiPO and LoReFT results. This extends hidden-state alignment to open-ended generation, where preference depends on content selection and compression across a complete response.

External baselines. Across the three settings, CAST reaches DPO-competitive operating points with task- and metric-dependent tradeoffs; BiPO and LoReFT remain particularly strong on ConFiQA.

Compatibility with DPO. Transferred CAST yields higher observed TL;DR win rates on both models and improves several ConFiQA metrics, while controllers trained directly on DPO yield mixed gains. IMDb shows the same compatibility with a reward–KL tradeoff (Appendix C.8). Thus DHSA can complement parameter-level preference optimization. Paired uncertainty is reported in Appendix C.6.

5 Discussion and Conclusion

RCM maps specified preferences onto signed causal effects of native residual computation, revealing residual competition and preference-relative component roles. Opposing effects can persist under preferred outputs and after substantial reorganization by preference post-training. RCM-guided CAST shows that the resulting causal structure can guide localized control. With frozen base weights, CAST realizes DHSA as a direct adaptation route through inference-time hidden states that can also complement parameter-level preference optimization. This connects a measurable account of preference expression to a concrete intervention space, linking mechanistic analysis with direct state control.

The main limitations are the evaluated preferences, model families, intervention settings, and our empirical focus on reference-relative pairwise training. Future work can study adaptive state control and other learning signals. Overall, when a target behavior is already reachable, its expression can be analyzed and adjusted directly through the inference-time computation that shapes generation.

AI USE STATEMENT

We used generative AI tools for manuscript proofreading, literature retrieval and discovery, and clarification of conceptual and methodological arguments. The authors are responsible for verifying the cited sources, mathematical arguments, and reported results, and take responsibility for the final content and AI-assisted materials.

ETHICS STATEMENT

This work studies how internal model computation supports or opposes specified preferences and how local state interventions can alter model behavior. Such control has potential dual uses: it may support desirable behavior, but could also be used to weaken safety constraints or steer outputs toward harmful objectives. The preference metrics studied here measure task-specific behavior and do not constitute comprehensive safety assessments. Deployment therefore requires evaluation of unintended behavioral changes, appropriate safeguards, and compliance with the licenses and usage conditions of the underlying models and datasets.

REPRODUCIBILITY STATEMENT

Section 3 specifies the RCM measurements and CAST interventions. Appendices A.1, A.2, and A.3 describe the task-specific scores, component-selection procedures, and pairwise training objective used in our experiments. The assumptions and proofs underlying the theoretical results are provided in Appendices D.1, D.2, and D.3. Appendix C reports the experimental settings and evaluation definitions. The anonymous artifact repository is linked in Section 4.

REFERENCES

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 *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/f545448535dfde4f9786555403ab7c49-Abstract-Conference.html.

Aryaman Arora, Dan Jurafsky, and Christopher Potts. CausalGym: Benchmarking causal interpretability methods on linguistic tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14638–14663. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.785. URL https://aclanthology.org/2024.acl-long.785/.

Baolong Bi, Shaohan Huang, Yiwei Wang, Tianchi Yang, Zihan Zhang, Haizhen Huang, Lingrui Mei, Junfeng Fang, Zehao Li, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, and Shenghua Liu. Context-DPO: Aligning language models for context-faithfulness. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10280–10300. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.536. URL https://aclanthology.org/2025.findings-acl.536/.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. In *Advances in Neural Information Processing Systems*, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/58cbe393b4254da8966780a40d023c0b-Abstract-Conference.html.

Nelson Elhage et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. URL https://arxiv.org/abs/2407.21783.

Jochen Hartmann, Mark Heitmann, Christian Siebert, and Christina Schamp. More than a feeling: Accuracy and application of sentiment analysis. *International Journal of Research in Marketing*, 40 (1):75–87, 2023. doi: 10.1016/j.ijresmar.2022.05.005. URL https://www.sciencedirect.com/science/article/pii/S0167811622000477.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. Aligning large language models with representation editing: A control perspective. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/41bba7b0f5c81e789a20bb16a370aeeb-Abstract-Conference.html.

Lucio La Cava and Andrea Tagarelli. Toward preference-aligned large language models via residual-based model steering. In *Proceedings of the Thirty-Fifth International Joint Conference on Artificial Intelligence*, pp. 5766–5774, 2026. doi: 10.24963/ijcai.2026/642. URL https://www.ijcai.org/proceedings/2026/642.

Andrew Lee, Xiaoyan Bai, Itamar Pres, Martin Wattenberg, Jonathan K. Kummerfeld, and Rada Mihalcea. A mechanistic understanding of alignment algorithms: A case study on DPO and toxicity. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 26361–26378. PMLR, 2024. URL https://proceedings.mlr.press/v235/lee24a.html.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2409.05907.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2306.03341.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10619–10638. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.572. URL https://aclanthology.org/2024.acl-long.572/.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.05101.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150. Association for Computational Linguistics, 2011. URL https://aclanthology.org/P11-1015/.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70b8505ac79e3e131756f793cd80eb8d-Abstract-Conference.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems*, 2024. URL https://arxiv.org/abs/2405.14734.

Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, New York, NY, 2 edition, 2006. doi: 10.1007/978-0-387-40065-5. URL https://doi.org/10.1007/978-0-387-40065-5.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2203.02155.

Qwen, An Yang, Baosong Yang, Beichen Zhang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Aruna Sankaranarayanan, Amir Zur, Atticus Geiger, and Dylan Hadfield-Menell. Activation steering via generative causal mediation, 2026. URL https://arxiv.org/abs/2602.16080v2.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, 2020. URL https://arxiv.org/abs/2009.01325.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2023. URL https://arxiv.org/abs/2308.10248.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

Tomás Vergara-Browne, Darshan Patil, Ivan Titov, Siva Reddy, Tiago Pimentel, and Marius Mosbach. Operationalising the superficial alignment hypothesis via task complexity. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.15829.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. TL;DR: Mining Reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pp. 59–63. Association for Computational Linguistics, 2017. doi: 10.18653/v1/W17-4508. URL https://aclanthology.org/W17-4508/.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. https://github.com/kingoflolz/mesh-transformer-jax, 2021.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small, 2022. URL https://arxiv.org/abs/2211.00593.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, Jiang Bian, and Mao Yang. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/517f9b9c227b9dd51dba4560f37165ed-Abstract-Conference.html.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. ReFT: Representation finetuning for language models. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/75008a0fba53bf13b0bb3b7bff986e0e-Abstract-Conference.html.

An Yang, Baosong Yang, Binyuan Hui, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024. URL https://arxiv.org/abs/2407.10671.

Fangcong Yin, Xi Ye, and Greg Durrett. LoFiT: Localized fine-tuning on LLM representations. In *Advances in Neural Information Processing Systems*, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/122ea6470232ee5e79a2649243348005-Abstract-Conference.html.

Xiao Yu, Qingyang Wu, Yu Li, and Zhou Yu. LIONs: An empirically optimized approach to align language models. *arXiv preprint arXiv:2407.06542*, 2024. URL https://arxiv.org/abs/2407.06542.

Yang Yue et al. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/537d5aa768c2d534016a4d06f87bc8fb-Abstract-Conference.html.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2309.16042.

Weixu Zhang, Ye Yuan, Changjiang Han, Yuxing Tian, Zipeng Sun, Linfeng Du, Jikun Kang, Hong Kang, Xue Liu, and Haolun Wu. Preference heads in large language models: A mechanistic framework for interpretable personalization. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 47742–47754. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.2205. URL https://aclanthology.org/2026.acl-long.2205/.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/ac662d74829e4407ce1d126477f4a03a-Abstract-Conference.html.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency, 2023. URL https://arxiv.org/abs/2310.01405.

A Task Scores and CAST Protocols

A.1 Task-Specific Preference Scores

The main text uses $C_e(M; x)$ as a generic preference score and $s_{e,\theta}(x, y)$ as the differentiable response score used for controller training. These quantities share the same preference direction but are not required to be numerically identical.

Let $\ell_{\theta,t} \in \mathbb{R}^{|\mathcal{V}|}$ denote the next-token logits of the controlled model at response position t, where $\mathcal{V}$ is the vocabulary. Let $\mathcal{T}(y)$ denote the token positions of response y. We use two response scores.

Mean token-logit margin. For response y,

$$s_\theta^{\text{margin}}(x, y) = \frac{1}{|\mathcal{T}(y)|} \sum_{t \in \mathcal{T}(y)} \left[\ell_{\theta,t}(y_t) - \max_{v \in \mathcal{V},\ v \neq y_t} \ell_{\theta,t}(v) \right]. \tag{5}$$

The score measures the average logit advantage of the designated response token over its strongest token-level alternative.

Mean response-token log-probability. For response y,

$$s_\theta^{\text{logp}}(x, y) = \frac{1}{|\mathcal{T}(y)|} \sum_{t \in \mathcal{T}(y)} \log \pi_\theta(y_t \mid x, y_{<t}). \tag{6}$$

The response is teacher-forced while this score is evaluated. The normalization by response length is part of the score definition and differs from an unnormalized sequence log-probability.

ConFiQA. For the Performance scan, a registered context-supported response y^+ and competing response y^- define the RCM measurement score

$$C_e(M; x) = s^{\text{margin}}(M; x, y^+) - s^{\text{margin}}(M; x, y^-). \tag{7}$$

The Site scan uses the same per-response token-logit margin but takes the maximum score over the admitted answer aliases separately on each side, in each model state being compared. Performance scores its registered single chosen/rejected endpoint pair. Thus the preference direction and token score agree, while the endpoint aggregation need not give identical numerical effects when multiple aliases are admitted. CAST training uses the corresponding mean token-logit-margin response score in Equation 5. Final evaluation is performed on generated answers rather than on this training surrogate and reports the task-specific generation metrics described in the experimental section.

IMDb. RCM measurement is performed on generated continuations. Let $\widehat{y}_M(x)$ denote a continuation generated by model M under the fixed decoding protocol. The frozen sentiment classifier assigns $R_e(y) = P(\text{POSITIVE} \mid y) - P(\text{NEGATIVE} \mid y) \in [-1, 1]$; the RCM score is

$$C_e(M; x) = R_e(\widehat{y}_M(x)). \tag{8}$$

When multiple generations are used, Equation 8 is replaced by their empirical mean. CAST training does not backpropagate through the sentiment reward; it instead uses $s_\theta^{\text{logp}}(x, y)$ from Equation 6 on fixed preference pairs. Final evaluation again uses generated continuations and reports external sentiment reward together with distribution-shift and generation-health measures.

TL;DR. For a preferred summary y^+ and rejected summary y^-, RCM uses

$$C_e(M; x) = s^{\text{logp}}(M; x, y^+) - s^{\text{logp}}(M; x, y^-), \tag{9}$$

and CAST training uses the same form of mean response-token log-probability. Final evaluation is performed on generated summaries through pairwise preference evaluation rather than through the training score.

Measurement, optimization, and evaluation. The three stages are intentionally distinguished:

$$\text{RCM measurement} \neq \text{CAST training surrogate} \neq \text{final generation metric} \tag{10}$$

in general. What is shared is the target preference e, not necessarily one numerical objective. Intervention positions and response-scoring positions are also distinct: an intervention may modify prompt or response states while the training response score is computed only over its designated response-token positions. RCM effects are singleton differences between complete model computations; they are not assumed to add across components. Native residual writes and CAST state updates are additive at their local application sites, but downstream computation can make a joint score change differ from the sum of singleton RCM effects.

A.2 RCM Candidate Selection

RCM selection proceeds hierarchically from coarse components to attention heads. All selection statistics are computed from per-example singleton RCM effects.

For candidate component i, let

$$\mu_i = \frac{1}{N} \sum_{n=1}^{N} \Delta_{i,e}^{\text{zero}}(x_n), \tag{11}$$

where $\{x_n\}_{n=1}^{N}$ is the RCM discovery set. Let $[L_i, U_i]$ denote the corresponding 95% confidence interval computed over the paired per-example effects.

The high and low sides use the directional confidence scores

$$q_i^{\text{high}} = L_i, \qquad q_i^{\text{low}} = -U_i. \tag{12}$$

Signed selections admit candidates to these sides by the sign of their own mean effect, so the bounds favor large effects stable in the corresponding direction. IMDb Performance instead ranks disjoint high- and low-effect extrema by the same bounds without requiring the low-end mean to be negative. An interval crossing zero does not itself exclude a candidate.

Hierarchical scan. Selection uses the following sequence:

1. scan the registered layer- or component-level candidate set;
2. form the registered signed roles or disjoint high/low extrema;
3. rank the high/positive side by L_i and the low/negative side by $-U_i$;
4. retain the selected parent layers;
5. scan all attention heads contained in the union of those layers;
6. assign and rank every head using the head's own effect.

A head therefore does not inherit the sign of its parent layer. The coarse unit is a pre-output-projection attention layer. Four parent layers are retained on each signed side or effect extreme, and every head in their union is scanned before the final head ranking. Scanning acts on scored decision states. ConFiQA and TL;DR use the teacher-forced prefixes of their fixed answer or summary endpoints; IMDb uses the prefixes of each actually generated continuation. In either case, the final non-padding prompt state predicts the first output token and each subsequent prefix-final state predicts the next token. Padding and post-EOS states are excluded. Native and intervened trajectories use this same position rule, even when their generated tokens differ.

The candidate pool is fixed before actuator training. The registered ConFiQA and IMDb vector Site comparisons select exactly four positive-effect and four negative-effect heads ($K = 8$). Confidence intervals determine the within-role ranking; an interval crossing zero does not exclude a candidate or change the head budget.

IMDb Performance selects four heads at each CI-ranked effect extreme instead; its low-effect group can contain heads with positive mean effects and is not necessarily a competing-side group. Its supporting-only deployment activates the four high-effect heads and leaves the weak low-effect bank inactive.

Low-rank configuration. For the more complex TL;DR open-generation task, a larger signed candidate pool is formed before training. The 16 candidate single-head controllers are trained and audited individually on 40 non-test posts, against the human reference in both presentation orders. The final eight-head control set is assembled from the audit without retraining the selected controllers jointly. Thus the *candidate pool* is determined before training, whereas the final audited combination is determined afterward.

Confidence intervals. The reported RCM mean-effect intervals use paired percentile bootstrap over the 300 discovery inputs (500 resamples, seed 42, pointwise 2.5th and 97.5th percentiles). The interval is for the mean singleton contrast, not for the probability that an individual input has a nonzero effect; it is not corrected for multiplicity across scanned components. No test-set metric is used to alter the RCM candidate pool.

A.3 Pairwise Training Used in Our Experiments

This section specifies the reference-relative pairwise objective and training settings used in our experiments.

For a training triple (x, y^+, y^-), let $s_{e,\theta}(x, y)$ denote the task-specific differentiable response score under CAST, and let $s_{e,0}(x, y)$ denote the score from the same frozen starting model with the controller disabled. Define

$$d_{e,\theta}(x) = s_{e,\theta}(x, y^+) - s_{e,\theta}(x, y^-) \tag{13}$$

and the reference-relative advantage gain

$$A_{e,\theta}(x) = d_{e,\theta}(x) - \mathrm{stopgrad}(d_{e,0}(x)) \,. \tag{14}$$

Here $\mathrm{stopgrad}(\cdot)$ evaluates its argument numerically but blocks gradients through it.

For the experiments reported in this paper, we minimize

$$\mathcal{L}_{\mathrm{CAST}}(\theta) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}_{\mathrm{train}}} \log \sigma(\beta A_{e,\theta}(x)) + \lambda \sum_{W\in\theta} \frac{\|W\|_F^2}{|W|}, \tag{15}$$

where $\mathcal{D}_{\mathrm{train}}$ is the controller-training distribution, σ is the logistic sigmoid, $\beta > 0$ controls the pairwise preference scale, $\lambda \geq 0$ controls explicit controller regularization, W ranges over trainable controller tensors, and $|W|$ is the number of scalar elements of W.

All base-model parameters ω remain frozen. Only controller parameters θ receive gradients. We use

$$\lambda = 10^{-4}, \tag{16}$$

in addition to the decoupled weight decay of AdamW (Loshchilov & Hutter, 2019). The explicit Frobenius penalty is a parameter-space regularizer and is not interpreted as a policy-KL penalty. No additional advantage threshold or explicit policy-KL term is used in this training objective.

The task-specific response score is

$$s_{e,\theta} = \begin{cases} s_\theta^{\mathrm{margin}}, & \text{ConFiQA,} \\ s_\theta^{\mathrm{logp}}, & \text{IMDb,} \\ s_\theta^{\mathrm{logp}}, & \text{TL;DR.} \end{cases} \tag{17}$$

Positive and negative RCM banks are trained toward the same target preference. Their labels record the native RCM effects of the selected sites and do not impose the sign of the learned perturbation.

The constant actuator contains one trainable vector $v_i \in \mathbb{R}^{d_i}$ per controlled site. The rank-k_i actuator contains

$$A_i \in \mathbb{R}^{k_i \times d_i}, \qquad B_i \in \mathbb{R}^{d_i \times k_i}, \tag{18}$$

and therefore contributes $2d_ik_i$ trainable matrix parameters per site, excluding any implementation-specific scalar gates. Reported parameter counts are taken from the instantiated controller rather than inferred from nominal rank alone.

Timing configurations are trained separately unless explicitly described as an inference-only timing audit.

A.4 Intervention Timing

The intervention mask determines which hidden states are modified; it is separate from the response positions used to compute the training score.

`all`. During teacher-forced training, the `all` configuration permits direct intervention at all registered positions in the complete prompt–response sequence. During autoregressive generation, intervention remains enabled in the initial prompt forward pass and the subsequent decoding passes. Each decoding forward pass directly intervenes on the current decision state used to produce the next-token distribution. Earlier controlled states affect later computation through the ordinary autoregressive state and cache.

`prefill`. During training, the `prefill` mask applies only to prompt positions. At inference time, it applies to all registered prompt positions during the initial prompt forward pass and is then disabled for direct decode-time intervention. The modified prompt computation and resulting cached states can nevertheless continue to affect subsequent generation.

`prompt_last`. When used as a diagnostic configuration, `prompt_last` restricts direct intervention to the final prompt position. It is therefore distinct from `prefill`, which can modify multiple prompt positions.

The exact timing mode associated with every reported experiment is stored with the corresponding run configuration. Timing variants are not silently pooled as one condition.

B Additional RCM Analyses

B.1 RCM-Patch

RCM-Patch measures the effect of replacing one component state with a source state or a constructed prototype in the receiving computation; downstream states are recomputed.

Let x^{ref} denote the receiving input and x^{src} the source condition. For structural component i, let h_i^{ref} and h_i^{src} denote the corresponding native component states at the matched registered positions. Let M_i^{ref} denote the unmodified receiving run and $M_{i\leftarrow h_i^{\mathrm{src}}}^{\mathrm{ref}}$ the same run after replacing only the registered state of component i. The Patch effect is

$$\Delta_{i,e}^{\mathrm{patch}}(x^{\mathrm{ref}};x^{\mathrm{src}}) = C_e\Big(M_{i\leftarrow h_i^{\mathrm{src}}}^{\mathrm{ref}};x^{\mathrm{ref}}\Big) - C_e\big(M_i^{\mathrm{ref}};x^{\mathrm{ref}}\big)\,. \tag{19}$$

The receiving prompt, preference definition, scoring rule, intervention positions, and downstream model computation remain fixed. Only the registered component state is replaced. All downstream states are recomputed after the substitution.

A positive Patch effect therefore means that the source state increases the target preference score relative to the reference state at the receiving condition. It does *not* mean that the component's native presence has a positive effect; native presence is measured by RCM-Zero.

Prototype states. When an experiment uses a prototype rather than a single paired source state, let $\mathcal{S}_e$ denote the source set for preference e. The prototype is

$$\bar{h}_{i,e}^{\mathrm{src}} = \frac{1}{|\mathcal{S}_e|}\sum_{x'\in\mathcal{S}_e} h_i(x'), \tag{20}$$

after applying the experiment's fixed position-alignment rule. The prototype replaces h_i^{src} in Equation 19. The reported Site RCM-Patch results use prototypes, not a newly sampled paired source for each receiving input. ConFiQA averages target-answer trajectories from the context-supported condition and patches the receiving RAG prompt; IMDb averages positive-prompt trajectories and patches the neutral receiving prompt. The TL;DR prototype averages decision states within each preferred summary, then across aliases and source inputs. In each case the receiver and scorer are otherwise unchanged. Each available source continuation contributes the mean of its decision states,

rather than one vote per token; where a question has multiple target aliases, their means are averaged before the equal-weight mean across source inputs. Accumulation and storage use float32, followed by a cast to the receiving component's output dtype. The resulting head vector is broadcast to each registered receiving decision state and overwrites that one head. Fixed-response contrasts score both teacher-forced endpoints; IMDb's generated-score contrast follows its receiving generation trajectory. The prototype is neither centered nor normalized. Patch candidates are split by the sign of their own mean paired Patch effect and ranked within each side by the same directional confidence-bound rule as Equation 12. For the vector Site configurations, the four-per-side budget has no confidence-interval admission threshold; ties use ascending layer and head indices.

Patch effects are treated as state-substitution effects and are not interpreted as a decomposition of the component's complete native responsibility.

B.2 Bilateral Effects Across Preference Domains

For the scanned component set $\mathcal{I}$ and an absolute effect threshold τ in the task's own score units, bilateral coverage is

$$\widehat{p}_{\text{comp}}(\tau) = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}\left[\max_{i\in\mathcal{I}} \Delta_{i,e}^{\text{zero}}(x_n) > \tau \ \wedge \ \min_{i\in\mathcal{I}} \Delta_{i,e}^{\text{zero}}(x_n) < -\tau\right]. \tag{21}$$

The two extrema are separate singleton interventions on the same unmodified input, not a jointly ablated trajectory. The rate is conditional on the scanned candidates and their granularity. The thresholds below are descriptive points on the full sensitivity curves, not significance cutoffs or parameters selected by the final test results.

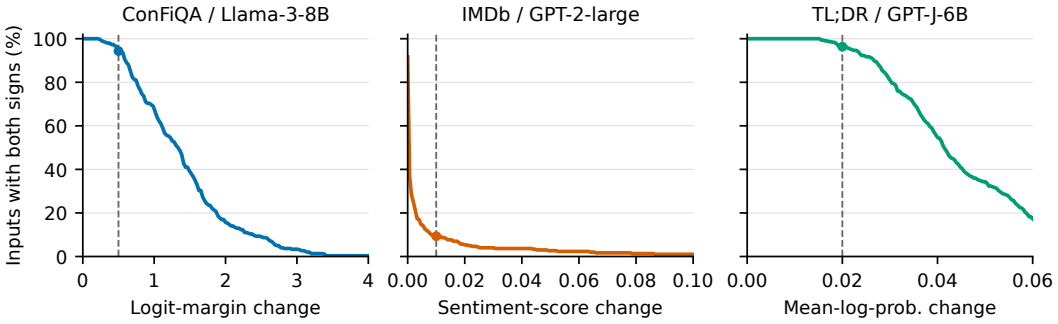

Figure 4: **Cross-domain bilateral RCM-Zero coverage.** All three scans use 300 discovery inputs. ConFiQA scans 32 attention-layer writes and uses paired token-logit-margin changes; IMDb scans 160 attention heads and uses generated sentiment-score changes; TL;DR scans 128 attention heads and uses fixed-summary mean-log-probability changes. Each panel retains its own score units and scanned set. Dashed lines mark the illustrative thresholds in Table 3. The curves quantify signed singleton effects, not simultaneous component ablations or the fraction of a model's output explained by their sum.

Task (scanned set)	τ	Both signs	Mean CI $+/-$
ConFiQA (32 layers)	0.5	283/300	18 / 4
IMDb (160 heads)	0.01	28/300	52 / 0
TL;DR (128 heads)	0.02	289/300	3 / 4

Table 3: **RCM-Zero effects on the frozen discovery inputs.** "Mean CI $+/-$" counts components whose pointwise paired-bootstrap interval for the mean lies entirely above/below zero. These are not input counts on which a component is active. IMDb effects are sparse: bilateral coverage falls from 28/300 at $\tau = 0.01$ to 3/300 at $\tau = 0.1$.

Additional registered scans show the model dependence behind these representative curves. On 300 ConFiQA inputs, the Qwen2-7B SFT scan finds both signs above $\tau = 0.5$ in 244/300 inputs

across 224 scanned heads. For TL;DR at $\tau = 0.02$, the 128-head GPT-J DPO scan finds 299/300, while the 256-head LION-LLaMA-3-8B scans find 300/300 before and after DPO. These are within-model, within-score threshold comparisons; the numbers of scanned heads and score units are not interchangeable across tasks. On the same per-component criterion used below, 1/224 Qwen2-7B, 89/128 GPT-J DPO, and 120/256 and 189/256 LION SFT/DPO heads have effects exceeding the task threshold in both directions on at least 10% of inputs. The paired TL;DR scans therefore show persistent bilateral effects after DPO without requiring the same head to retain the same role.

In IMDb, 52 scanned heads have mean-effect intervals entirely above zero and none has an interval entirely below zero. This weak competing-side signal motivates the supporting-only deployment. The Site and Performance selection rules are specified separately in Appendix A.2.

DPO effect-change aggregation. The 1.00 margin-unit absolute change in Section 4.2 is $\frac{1}{N|\mathcal{I}|}\sum_{n=1}^{N}\sum_{i\in\mathcal{I}}|\Delta_i^{\mathrm{DPO}}(x_n) - \Delta_i^{\mathrm{SFT}}(x_n)|$, with $N = 300$ paired inputs and $|\mathcal{I}| = 32$ layers. Its 95% interval resamples the 300 paired inputs with replacement, retaining the 32-layer vector for each sampled input (500 percentile-bootstrap resamples, seed 42). It is not the absolute change of each layer's sample-mean effect.

Preference-relative component roles. The sign of a component's mean singleton effect describes its average target-supporting or target-competing role under the specified task and intervention. To distinguish a near-zero mean from genuinely small per-input effects, we also report its mean absolute effect ν_i and $\rho_i^+(\tau) = \mathbb{P}_x(\Delta_i(x) > \tau)$ and $\rho_i^-(\tau) = \mathbb{P}_x(\Delta_i(x) < -\tau)$, using the empirical frequencies over the same discovery inputs. For example, ConFiQA layer 14 has mean $+0.072$ but mean absolute effect 1.090 margin units, with effects above $+0.5$ on 40% of inputs and below -0.5 on 29%. IMDb head L12H14 has mean $+0.079$, mean absolute effect 0.453, and respective frequencies 18.3% and 13.7% at $\tau = 0.01$; TL;DR head L10H7 has mean $+0.0015$, mean absolute effect 0.0283, and frequencies 27.3% and 24.7% at $\tau = 0.02$. Across the scanned sets, 23/32, 14/160, and 27/128 components respectively exceed their task's stated threshold in both directions on at least 10% of inputs. These descriptive roles are not a mutually exclusive causal partition and do not imply additive effects.

B.3 Target Outputs with Opposing Singleton Effects

Two frozen ConFiQA QA cases connect the score contrast to the model's actual native answer. In sample 000046, the question is "Who owns Arsenal F.C.?"; the supplied context supports "Kering" rather than the prior answer "Kroenke Sports & Entertainment." The native Llama-3-8B response is "Kering." On this same input, attention-layer L30 has a $+1.758$ and L1 a -6.805 native-minus-zero effect on the target–non-target logit margin. In sample 000086, the supplied context supports "J. K. Rowling" rather than "Tony McNamara" for the creator of *The Great*; the native response is "J. K. Rowling." L0 has a $+4.340$ effect and L18 a -3.406 effect. Each value comes from a separate single-component ablation with the fixed answer endpoints. Thus a generated target answer can coexist with large opposing native effects. These fixed-endpoint score contrasts do not themselves establish that the ablated model would decode a different answer.

B.4 Decoded Answer Changes under Single-Head Ablation

We separately checked whether a single-head ablation changes the decoded answer, rather than only a fixed-answer score. On frozen ConFiQA QA test inputs, we compared native Llama-3-8B-Instruct generation with generation under RCM-Zero of pre-output-projection head `L17.attn.h5`. The head has a negative mean native-minus-zero RCM effect of -0.39176 on the registered target–non-target margin. Both runs use the same prompt and greedy decoding with at most 64 new tokens and EOS enabled. The context-supported answer is the task target, not necessarily the real-world answer.

The check covered 866 test inputs in fixed order. String-based scoring flagged 36 candidates; the following ten were selected after reading both complete outputs and verifying that the ablated answer affirms the context-supported target whereas the native answer does not. Because these are selected cases, neither 36/866 nor 10/866 estimates a semantic flip rate. The outputs below are decoded completions, not teacher-forced answers or margin predictions.

1. **Test 232: Edgar Allan Poe; target: dance music.** Native: "Literary fiction, specifically Gothic fiction and Dark Romanticism." Zero: "According to the passage, the genre associated with Edgar Allan Poe is dance music."
2. **Test 299: Pharrell Williams; target: Michelle Grace.** Native: "Michelle Larcher de Brito (née Williams), a fashion designer and artist." Zero: "Pharrell Williams' spouse is Michelle Grace, a fashion designer and artist."
3. **Test 422: Planet of the Apes (2001); target: Tim Story.** Native: "Tim Story did not direct the 2001 film 'Planet of the Apes'. The actual director of the 2001 film 'Planet of the Apes' is Tim Burton." Zero: "According to the text, the director of the Planet of the Apes (2001 film) is Tim Story."
4. **Test 470: Medici (TV series); target: superhero film.** Native: "The Medici (TV series) belongs to the genre of Historical Drama." Zero: "According to the text, the Medici: Masters of Florence series belongs to the genre of 'superhero film'."
5. **Test 554: Donald Glover; target: action film.** Native: "According to the passage, Donald Glover represents the genres of comedy, acting, and rap." Zero: "According to the passage, Donald Glover represents the genre of action film, as well as music (specifically, rap)."
6. **Test 636: Sea of Thieves; target: action thriller.** Native: "Action-adventure". Zero: "According to the text, the genre of Sea of Thieves is an action thriller."
7. **Test 660: The Conjuring: The Devil Made Me Do It; target: David Leitch.** Native: "Michael Chaves, not David Leitch." Zero: "According to the text, the director of The Conjuring: The Devil Made Me Do It is David Leitch."
8. **Test 697: Axel Witsel; target: Joe Biden.** Native: "Axel Witsel is a Belgian professional footballer, not American. He was born in Liège, Belgium, not the United States of America. Therefore, the head of state of the country where Axel Witsel holds citizenship is the King of Belgium, currently Philippe." Zero: "According to the text, Axel Witsel is a professional footballer who plays for the Turkish club Beşiktaş and the United States of America national team. Therefore, the head of state of the country where Axel Witsel holds citizenship is President Joe Biden, who is the President of the United States of America."
9. **Test 713: Sridevi; target: United States dollar.** Native: "The question is a bit tricky because the text states that Sridevi is a renowned American actress, but it also mentions that she was born in Sivakasi, Tamil Nadu, which is a location in India, not the United States. Therefore, it is not possible to determine the currency of the country where S" Zero: "The answer is USD, which is the United States dollar."
10. **Test 833: Aymeric Laporte; target: North America.** Native: "The answer is Europe. Aymeric Laporte was born in Narbonne, France, not the United States of America. He has represented France at various youth levels and made his senior international debut for France in 2016, not the United States of America." Zero: "The answer is North America."

In the Tim Story and David Leitch examples, the native output mentions the target while explicitly denying it. A lexical answer-occurrence score can therefore mark both native and ablated texts as target hits; the semantic flip in these examples follows from the complete generated statements. The native completion for test 713 ends mid-sentence at the 64-token limit.

C Experimental and Implementation Details

C.1 Optimization Details

All CAST runs freeze the base language-model parameters and update only the controller parameters. AdamW (Loshchilov & Hutter, 2019) is used for controller optimization. The explicit controller penalty uses $\lambda = 10^{-4}$ as defined in Equation 15; any decoupled AdamW weight decay is reported separately and must not be described as policy-KL regularization.

Table 4 reports the locked SFT-starting CAST settings. Within a task, the two reported model families use the same pair counts and optimizer settings; `all` and `prefill` are separately trained.

ConFiQA selects one shared α from $\{0.1, 0.2, \ldots, 1.0\}$ by maximizing the non-test score $r_{\text{context only}} + r_{\text{short exact context}} - r_{\text{prior hit}} - r_{\text{neither}} - \bar{\ell}_{\text{chars}}/400$; exact ties choose the smallest

Setting	ConFiQA	IMDb	TL;DR
Training pairs	240	1,024	8,192
Validation pairs	60	256	512
Epochs	2	1	1
Learning rate	0.05	0.05	0.001
Training batch	4	32	2
β	1.0	1.0	0.5
AdamW weight decay	0.01	0.01	0.01
Random seed	42	42	42

Table 4: SFT-starting CAST optimization settings. ConFiQA and IMDb train the two banks separately; the supporting-only IMDb configuration deploys only the supporting bank. TL;DR trains 16 candidate single-head low-rank controllers before its held-out head audit. Separate timing modes are treated as independent training runs; inference-only timing audits are explicitly identified as such.

Execution choice	Frozen rule
SV initialization	Each head vector v_i starts at zero in float32.
Low-rank initialization	A_i is drawn from a zero-mean normal with standard deviation 0.01 and B_i starts at zero, both in float32. The TL;DR run draws the head-by-rank layout before transposing A_i; the initial intervention is zero.
Training strength and checkpoint	$\alpha_{\text{train}} = 1$ throughout optimization. The payload saved after the configured final epoch is used; validation does not select an intermediate checkpoint.
DPO start	Transfer (T) applies the saved SFT-start payload to the released DPO checkpoint without further controller training. Nontransfer (NT) initializes a new controller and trains it on the DPO checkpoint with the same task-level optimization values in Table 4. Each row retains its registered training and inference timing.
Bank deployment	The retained banks are summed without normalization or joint retraining and multiplied by one shared inference strength α, not independently calibrated strengths.

Table 5: Controller initialization, training, and deployment settings.

strength. Here context-only means a context-answer hit without a prior-answer hit; short-exact-context means an exact context answer in at most 48 decoded characters; prior-hit includes outputs mentioning both answers; neither means no answer hit; and $\bar{\ell}_{\text{chars}}$ is mean decoded length in characters. TL;DR searches $\{0.2, 0.3, \ldots, 0.7\}$ and maximizes the two-order balanced win rate against the human summary on its strength calibration posts. Strengths within 0.003 of the best rate are eligible, with the smallest eligible α chosen. IMDb reports the full $\{0.1, 0.2, \ldots, 1.0\}$ response curve rather than selecting one point by the final test. Transfer recalibrates on the DPO inference checkpoint's non-test data. Test-set outcomes are not used to select α, component sets, ranks, checkpoints, or other operating parameters.

C.2 Model and Hardware Details

The starting/DPO checkpoint pairs are Meta-Llama-3-8B-Instruct / Context-Faithful-LLaMA-3-8B and Qwen2-7B-Instruct / Context-Faithful-Qwen2-7B for ConFiQA; ma921 GPT-2-large SFT / DPO for IMDb; and CarperAI GPT-J TL;DR SFT / quyanh GPT-J DPO and Columbia-NLP LION-LLaMA-3-8B SFT / DPO for TL;DR. These DPO models are released checkpoints, not retrainings on CAST's controller pairs. The anonymous condition index supplies their full public model identifiers, immutable revisions, tokenizer bindings, precision, and per-run software environment. The accompanying anonymous materials are indexed by `EVIDENCE_INDEX.json` at `https://anonymous.4open.science/r/anonymous-research-materials-CEF6/`. The ConFiQA Qwen2-7B SFT+CAST-SV rows and transferred DPO+CAST-SV rows use model revision `f2826a00ceef68f0f2b946d945ecc0477ce4450c`. The Performance comparisons used 80-GB H100 devices with baseline-specific dependency profiles; the Site archive records its own device profile rather than assuming the Performance environment applies to it.

The base model families are GPT-2 (Radford et al., 2019), GPT-J (Wang & Komatsuzaki, 2021), Llama 3 (Grattafiori et al., 2024), Qwen2 (Yang et al., 2024), and Qwen2.5 (Qwen et al., 2024). The Context-Faithful checkpoints originate from Context-DPO (Bi et al., 2025); the LION SFT/DPO checkpoints originate from the LION alignment study (Yu et al., 2024).

Task/model	CAST operator	Rank	Active heads	Learned scalars
ConFiQA / Llama-3-8B	SV	–	8	1,024
ConFiQA / Qwen2-7B	SV	–	8	1,024
IMDb / GPT-2-large	SV, two banks	–	8	512
IMDb / GPT-2-large	SV, support-only	–	4	256
IMDb / GPT-2-large	low-rank, two banks	4	8	4,096
TL;DR / GPT-J-6B	low-rank	4	8	16,384
TL;DR / LION-LLaMA-3-8B	low-rank	4	8	8,192

Table 6: Actual active controller parameter counts from the frozen controller tensors. Separate `all` and `prefill` runs have the same counts. The main IMDb efficiency curve deploys the support-only bank; its four heads are a subset of the eight-head trained candidate. TL;DR trains 16 candidate single-head controllers and retains eight after audit, without joint retraining. Candidate-training cost and final active size are reported separately.

For head-level CAST, interventions are applied to the registered head slice before the attention output projection. For layer- or MLP-level CAST, the corresponding residual-write interface is used. The model implementation, tokenizer, and all non-controller parameters remain unchanged within each comparison. IMDb controller pairs originate from `ma921/imdb-tokenized_noise0` at revision `5ad0e4fbb89a87404918f5f510c9192afb3d350a`; its test comments originate from `stanfordnlp/imdb` at revision `e6281661ce1c48d982bc483cf8a173c1bbeb5d31`. TL;DR pairs and test-post projections derive from `openai/summarize_from_feedback` at revision `b8f7d168b6f4e95b2a92e84768bd6c955bed2f29`. The indexed per-condition configurations identify the exact public source URLs, model revisions, and tokenizer binding; these pair sources must not be inferred from a released DPO checkpoint's name alone.

C.3 BASELINE CONFIGURATIONS

All methods in a given task/model comparison use the same final input set, decoding rule, and evaluator. This is an evaluation-matched comparison; it is not a claim that public DPO checkpoints or the external methods share CAST's training pairs or the same search cost. Search and final active controller size are separate quantities.

DPO. DPO is the main preference-optimization reference. Every DPO row in the main Performance table evaluates a released checkpoint. Its original preference data may differ from CAST's frozen pair set; only the final evaluation is shared. DPO+CAST-T reuses an SFT-trained controller and recalibrates strength on non-test data; DPO+CAST-NT trains a new controller on the DPO checkpoint.

LoReFT. LoReFT retains the published low-rank representation-intervention operator and training objective (Wu et al., 2024). Its four-layer joint tuples and strength grid are selected on non-test data. Mapping published layer depths to a different architecture is our registered adaptation, not an author-provided task default.

BiPO. BiPO retains its bidirectional preference objective and one learned steering vector per candidate layer (Cao et al., 2024). It is not converted to CAST's signed-bank training. Its CAA-based layer preselection and candidate calibration remain separate from RCM. We explicitly apply the registered AdamW weight decay of 0.05; this is a project adaptation because the released training script exposes that value without forwarding it to its DPO configuration.

Task/model	CAST trained	CAST retained	BiPO layers	LoReFT tuples
ConFiQA / Llama-3-8B	8	8	6	8
ConFiQA / Qwen2-7B	8	8	6	8
IMDb / GPT-2-large	8	4–8	10	6
TL;DR / GPT-J-6B	16	8	7	1
TL;DR / LION-LLaMA-3-8B	16	8	14	2

Table 7: Candidate-training opportunity by task/model, not a claim of equal GPU cost. CAST columns count heads. Its TL;DR candidates are individually trained heads; BiPO independently trains each registered layer, whereas each LoReFT tuple is a joint four-layer controller. BiPO uses 20/5/1 epochs on ConFiQA/IMDb/TL;DR; LoReFT uses 24 epochs, while CAST uses the epochs in Table 4. All candidate selection and strength calibration precede final test.

For IMDb, the registered 80-GB H100 pre-test occupancy was approximately 2,448 seconds for SFT CAST-SV-all, 2,343 seconds for the expanded ten-layer BiPO controller search, and 2,410 seconds for the six-tuple LoReFT search. The boundaries are CAST's start-to-pretest pipeline versus external controller jobs; final generation/evaluation is excluded. These durations show comparable observed occupancy for this task, not matched FLOPs, and no strict compute-equality claim is made for the other tasks.

The TL;DR audit tests each trained single-head payload against the human reference at $\alpha \in \{0.2, 0.3, \ldots, 0.7\}$ in both display orders. Its review packet replaces head coordinates with audit IDs and includes balanced win rate, output-health measures, the post, human summary, and generated text. Human review retains a clean gain with healthy summaries; it can also retain a candidate whose high-strength overshoot is recoverable at a lower strength. Persistent factual errors, repetition, unfinished outputs, and weaker or redundant benefit count against retention under the fixed eight-head budget. These are qualitative keep/drop principles, not a scalar sorting rule or numerical tie-break. The decisions are fixed before final test. Retained tensors are united directly without normalization, rescaling, sign rebalance, or an optimizer step. The GPT-J SFT-all freeze contains two target-support and six competitor-support heads, confirming that no four/four final quota is imposed. In the TL;DR Site comparison, RCM-Zero, RCM-Patch, ITI, and each Random seed each receive 16 single-head training opportunities and the same 40-post review, retaining eight heads without joint retraining. TL;DR ITI labels preferred and rejected summaries as positive and negative states, uses the final state of each teacher-forced continuation, and ranks heads by grouped two-fold probe accuracy before this common audit. Random candidates are drawn without replacement from architecture-compatible heads.

Task	RCM search	Final selection
ConFiQA	Four parent layers per signed side; every head in their union	Four heads per side by each head's own directional confidence bound; no post-training head audit.
IMDb Site	Four parent layers per signed side; every head in their union	Four positive- and four negative-effect heads by directional confidence bound.
IMDb Performance	Four parent layers at each high/low effect extreme; every head in their union	Four heads per extreme by CI bound, regardless of the low-end sign; support-only activates the high-effect bank.
TL;DR	Four parent layers per side; every head in their union	Eight signed candidates per side are trained individually; a 40-post non-test audit retains eight without a final sign quota.

Table 8: Task-specific selection and audit budgets. Each condition's ordered head identifiers, bank composition, and final strength are recorded in the position, controller, and strength manifests linked by the anonymous index.

Selector controls. In the ConFiQA and IMDb vector Site comparisons, RCM constructs two signed four-head candidate banks and trains them separately toward the same target preference. The supporting-only IMDb configuration leaves the weak competing bank inactive at deployment without changing the candidate rule. ITI and the three fixed random selectors each use one jointly trained unsigned eight-head bank. These comparisons measure the registered selector-plus-bank configurations, not the site-ranking rule in isolation. TL;DR uses its separate single-head audit and eight-head composition protocol, so the vector two-bank description does not apply there. Final inputs and evaluators are shared within each comparison. Random sites are architecture-compatible and retain their fixed seeds. ITI trains one L2-regularized logistic probe per pre-output attention head on the same paired discovery inputs as RCM. It reads the final non-padding state of each complete labeled continuation and ranks heads by mean held-out accuracy in two grouped folds (train A/test B, then reverse). Both endpoints of a pair stay in the same fold; accuracy ties use ascending layer and head indices. ConFiQA labels identify canonical context-supported versus prior-answer continuations. IMDb labels identify positive-prompt versus neutral-prompt generations with the shared native token sequences; neither raw IMDb sentiment labels nor reward-ranked pairs are substituted. ITI's eight selected heads form one unsigned bank.

C.4 Data Separation and Selection Hygiene

We distinguish RCM selection, controller training, validation, strength/audit calibration where applicable, and final test by their uses. The final test is held out from selection, training, and calibration; other roles are not universally disjoint. In ConFiQA, controller training and validation are partitioned from the same 300 admitted groups used by the selector. In the registered TL;DR procedure, selector inputs may overlap controller-training inputs, while audit, strength-calibration, and final-test posts are separately assigned. The principal Performance rows use respectively 300/240/60/120/2,048 inputs or pairs for ConFiQA (selector/train/validation/strength/test), 300/1,024/256/2,048 for IMDb (selector/train/validation/test), and 300/8,192/512/40/40/512 for TL;DR (selector/train/validation/head audit/strength/test). ConFiQA's final 2,048 inputs mix QA, MR and MC; the TL;DR final set has 512 posts. IMDb's 1,024 training pairs retain the public token-pair source after its frozen cleaning and split rule; its 300 selector prompts and 2,048 official test comments are distinct from the training pool. The TL;DR 16-head audit uses 40 non-test posts and evaluates each candidate against the human reference in both presentation orders; the selected eight-head combination is not jointly retrained. Calibration data are small relative to training, but candidate audit and strength search are included in the method's acquisition cost rather than hidden inside the final eight-head parameter count.

No final-test example selects a component, controller checkpoint, rank, head combination, or strength. The anonymous condition index contains the exact row identifiers, final sites, strengths, checkpoints, scoring traces, public source revisions, and execution environments for each row; the appendix reports the scientifically relevant comparisons rather than duplicating that machine-readable inventory.

C.5 Final Evaluation Definitions

ConFiQA. The scorer case-folds generated text and answer aliases, replaces punctuation by spaces, and collapses whitespace. PC is one if any context-supported answer alias occurs as a whitespace-bounded phrase anywhere in the output. The analogous prior-answer hit is PO. EM is one only if the entire normalized output equals a context-supported alias. Thus PC need not imply an exact or well-formed answer. For example, in frozen QA row 001894 the target alias is "Jane Conrad" and CAST-all outputs "Jane Conrad, a social worker and activist." This has PC $= 1$ and EM $= 0$. Site reports the macro-average PC across QA, MR and MC; Performance reports the combined 2,048-input test result. PC is an occurrence metric, so a negated mention of a target answer can still count as a hit; the decoded examples in Appendix B.4 are also checked for whether the model affirms that answer.

IMDb. The frozen evaluator is SiEBERT (Hartmann et al., 2023), using `siebert/sentiment-roberta-large-english` at revision `74cea614e245`. Its reported sentiment reward is $P(\text{POSITIVE}) - P(\text{NEGATIVE})$, not an unbounded learned reward; positive rate is the fraction with reward greater than zero. Both distribution-shift axes compare the evaluated policy to the registered SFT reference on its generated continuation y:

$$\begin{aligned} K_{\text{seq}}(x, y) &= \sum_t \log \frac{p_M(y_t \mid x, y_{<t})}{p_{\text{SFT}}(y_t \mid x, y_{<t})}, \\ K_{\text{tok}}(x, y) &= \frac{1}{|y|} \sum_t D_{\text{KL}}(p_M(\cdot \mid x, y_{<t}) \,\|\, p_{\text{SFT}}(\cdot \mid x, y_{<t})) \,. \end{aligned} \tag{22}$$

The main GPT-2-large Site plot uses the average K_{seq}, labeled *sequence log-ratio*; the Performance plot uses the average K_{tok}. K_{seq} is a generated-sequence log-probability ratio, not the tokenwise full-vocabulary KL. Under deterministic decoding, it is not an unbiased Monte Carlo estimate of policy sequence KL. The additional Qwen2.5-14B Site plot uses full-vocabulary token KL (Appendix C.7). Generation health is the nonempty-output rate plus mean completion length. At the plotted DPO point these are 99.90% and 230 tokens; for support-only CAST-SV-all at $\alpha = 0.5$ and 0.6 they are 100%/215 tokens and 99.95%/212 tokens, respectively. Reward and KL are always reported together, so a higher reward obtained only with much greater policy shift is not treated as an unqualified improvement. All 2,048 IMDb test inputs remain in the sentiment-reward, KL, and nonempty-rate denominators, including decoded empty strings; the empty string is sent to the same SiEBERT classifier. In the GPT-2 Performance records, a sole generated EOS can decode to empty text but remains one generated token and one token-KL step. For a genuinely zero-token continuation, the Site evaluator records $K_{\text{tok}} = 0$ rather than dividing by zero; this occurs in the reported Qwen2.5-14B Site curve, whose reward and KL averages retain all 2,048 inputs.

TL;DR. The pairwise evaluator is the registered `deepseek-v4-pro` API alias; the provider does not publish an immutable revision for that alias, so the saved request and response traces are the version evidence. The system message is "You are an impartial evaluator. Follow the requested comparison criteria exactly. The summaries are anonymous. Return only a valid JSON object." The user message is:

> Which of the following summaries does a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details? A good summary is both precise and concise.
>
> Post: [post]
> Summary A: [summary A]
> Summary B: [summary B]
>
> First provide a one-sentence comparison of the two summaries, explaining which you prefer and why. Then choose only A or B. Return JSON exactly as: `{"comparison": "one-sentence comparison and explanation", "preferred": "A or B"}`

The first A/B order is deterministic from the sample ID and seed; the second call swaps the order. A row scores $(\mathbf{1}[\text{candidate wins first}] + \mathbf{1}[\text{candidate wins swap}])/2$, so its value is 0, 0.5, or 1; the table reports the mean over rows. There is no tie label or imputed judgment: a malformed response is retried

up to five times and then fails evaluation. Empty model outputs remain in the comparison rather than being silently dropped. All methods use the same human references, prompt and two-order rule. The reported win rates are not head-to-head comparisons between CAST and DPO. The saved API configuration uses `deepseek-v4-pro` with thinking disabled, `reasoning_effort=high`, an 800-token response cap, a 180-second timeout, and at most five retries. Each of the 20 TL;DR Performance conditions has 512 posts with both presentation orders judged. No post exhausted retries, so every reported win rate uses all 512 posts. The provider does not publish an immutable revision for this model alias; the saved requests and responses document the calls.

Task	Final prompt and input handling	Decoding and stopping
ConFiQA	`context, Q: question, A:` prompt, wrapped in the model's registered chat template for instruct checkpoints. No additional input-token cap or truncation is set by the final-generation callback.	Greedy; at most 64 new tokens; registered `Q:` stop string and native EOS; seed 42.
IMDb	A tokenizer-derived 2–8-token review prefix, without a chat template. No final-generation input truncation is set.	One sampled continuation per prompt; at most 256 new tokens; temperature 1, top-k 50, top-p 1, seed 42, native EOS. The same random-number schedule is reused across strengths.
TL;DR	The prepared source post prompt is passed through without an additional chat template or final-generation truncation.	Greedy; at most 100 new tokens; seed 42, native EOS, model KV cache enabled. The registered top-$p = 0.9$ and top-$k = 0$ do not affect greedy selection.

Table 9: Final-generation settings. "No input cap" means no extra truncation in this evaluation path; the model's context capacity still applies. RCM discovery generation has its own registered settings and is not inferred from this table.

C.6 Paired Uncertainty for Main Performance Contrasts

Table 10 reports selected main-table differences using the saved per-input scores. For each comparison, the same test inputs are paired by ID; we resample inputs with replacement 5,000 times using seed 42 and take the 2.5th and 97.5th percentiles of the mean paired difference. TL;DR's first and swapped presentation-order judgments are first averaged within each post, so the 512 posts, not 1,024 judge calls, are the resampling units.

C.7 Additional IMDb Site Result

Figure 5 reports the second IMDb Site model family omitted from the main efficiency panel. On Qwen2.5-14B, RCM-Patch, ITI and one random seed all achieve large sentiment changes as strength increases, whereas RCM-Zero is weaker over the recorded response curve. Thus the GPT-2-large ranking is not assumed to transfer unchanged to Qwen. Every curve contains all ten registered strengths on the same 2,048 test prompts; the three random seeds remain separate. The archived Site score is mean positive-class probability p_+ from a binary classifier, converted to the paper's bounded reward $2p_+ - 1$. The horizontal axis is the full-vocabulary token KL to the unmodified Qwen reference; it is not the sampled sequence log-probability ratio used in the main GPT-2-large Site panel. Nonempty-output checks remain relevant at large strengths: RCM-Zero reaches an empty-output rate of 3.6% at one grid point, while the other plotted selectors remain at zero.

C.8 IMDb DPO-Start Response Curves

Figure 6 completes the DPO+CAST comparison for IMDb without listing every test strength. Transfer reuses the SFT-trained controller; non-transfer trains on the DPO checkpoint. Both constant-vector and low-rank controllers are shown at `all` and `prefill` timings. The complete curves reveal

Task / metric	Contrast	Difference [95% interval], pp
ConFiQA Llama / PC	CAST-all − DPO	-6.98 $[-8.74, -5.27]$
	DPO+CAST-T-all − DPO	0.00 $[-0.88, 0.88]$
ConFiQA Llama / EM	CAST-all − DPO	-43.99 $[-46.19, -41.89]$
	CAST-prefill − DPO	20.65 $[18.50, 22.80]$
	DPO+CAST-T-all − DPO	3.96 $[2.25, 5.66]$
ConFiQA Qwen / PC	CAST-all − DPO	3.66 $[2.05, 5.27]$
	DPO+CAST-T-all − DPO	10.01 $[8.35, 11.62]$
ConFiQA Qwen / EM	CAST-all − DPO	12.45 $[10.84, 14.01]$
	DPO+CAST-T-all − DPO	35.99 $[33.89, 38.13]$
TL;DR GPT-J / Win	CAST-all − DPO	-0.39 $[-3.91, 3.12]$
	DPO+CAST-T-all − DPO	6.45 $[3.52, 9.47]$
TL;DR LION / Win	CAST-all − DPO	-5.57 $[-8.30, -2.93]$
	CAST-prefill − DPO	-1.86 $[-4.69, 1.07]$
	DPO+CAST-T-all − DPO	0.88 $[-1.17, 2.93]$
	DPO+CAST-T-prefill − DPO	2.34 $[-0.10, 4.79]$

Table 10: Paired percentile-bootstrap intervals for the Performance table, in percentage points. ConFiQA uses 2,048 inputs per model; TL;DR uses 512 posts per model. The reproducible calculation is supplied as `reproduction/paired_main_effects.py` with its JSON output in the anonymous materials. These are fixed-checkpoint, fixed-test-set intervals, not intervals over training seeds; an interval spanning zero does not establish equivalence.

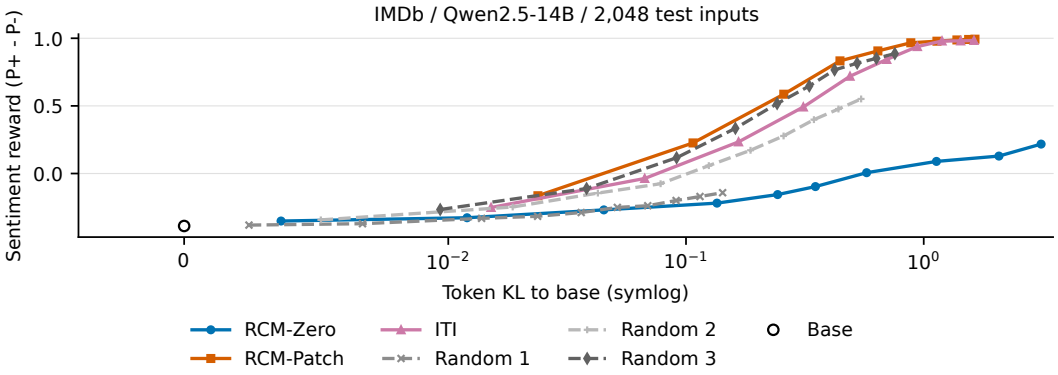

Figure 5: **IMDb Site response curves for Qwen2.5-14B.** Mean bounded sentiment reward against exact token KL to the same model without control. The open circle is the unmodified base model. Each selector is evaluated at ten registered strengths on 2,048 shared test prompts. Random 1–3 are fixed, independently reported seeds; no seed is selected by test score.

the reward–distribution-shift tradeoff: `all` can raise sentiment reward substantially, whereas the `prefill` changes are smaller and stay close to the DPO point. The plotted strength grid is descriptive; no test point was used to choose the deployed strength.

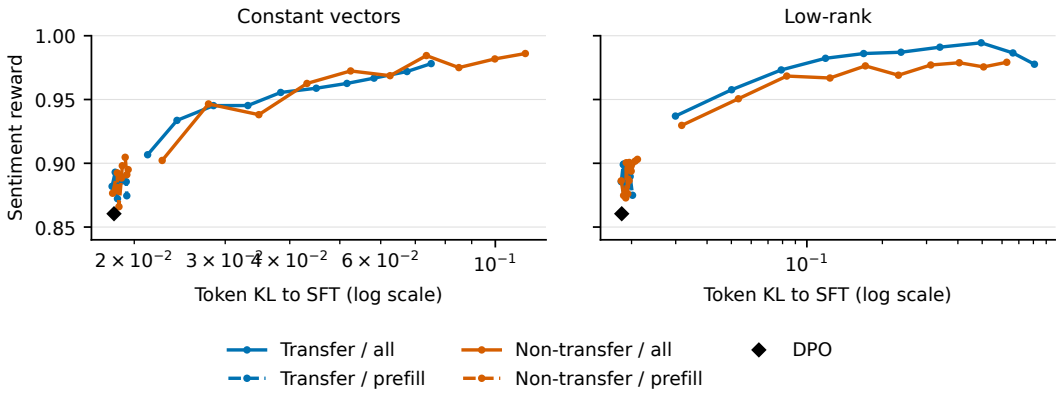

Figure 6: **IMDb DPO+CAST, 2,048 frozen test prompts.** The black diamond is the released DPO checkpoint. Each line contains the ten registered controller strengths and reports the same bounded sentiment reward and tokenwise KL to the SFT reference as the main IMDb plot. Blue/orange denote SFT-to-DPO transfer/new DPO training; solid/dashed denote all-token/prefill application. These are test response curves, not test-selected optima.

DPO-start two-bank ablation. Figure 7 reports the `all` and `prefill` bank comparisons. The ablation replays the four frozen constant-vector DPO+CAST controllers with only their four target-supporting heads active. The original two-bank curves are the matched controls: the DPO checkpoint, 2,048 test prompts, ten strengths, generation settings, and evaluator are unchanged. No head is reselected and no controller is retrained for this ablation. At $\alpha = 0.4$ under `all` timing, transfer yields reward/KL $0.9405/0.0279$ with the supporting bank and $0.9453/0.0332$ with both banks; DPO-start training yields $0.9155/0.0238$ and $0.9626/0.0432$, respectively. The uncontrolled DPO point is $0.8604/0.0183$. Thus the supporting bank alone can raise sentiment reward above DPO, while the full controller can yield additional reward at greater distribution shift. The `prefill` curves stay much closer to DPO. These observations do not establish which bank configuration is superior at matched KL.

D Theoretical Details

D.1 DPO-Form Interpretation of the Pairwise Objective

This section establishes the DPO-form representation of the pairwise objective used in Section 3.4, through a score-induced policy.

Fix an input x. Let $\mathcal{Y}_x$ denote the response space considered by the protocol and let $\pi_0(y \mid x) > 0$ be the frozen reference distribution on that space. Define

$$u_{e,\theta}(x, y) = s_{e,\theta}(x, y) - s_{e,0}(x, y), \tag{23}$$

where $s_{e,\theta}$ and $s_{e,0}$ are the controlled and reference response scores.

Assume the normalizer

$$Z_{e,\theta}(x) = \sum_{y \in \mathcal{Y}_x} \pi_0(y \mid x) \exp(u_{e,\theta}(x, y)) \tag{24}$$

is finite. Define the score-induced policy

$$\bar{\pi}_{e,\theta}(y \mid x) = \frac{\pi_0(y \mid x) \exp(u_{e,\theta}(x, y))}{Z_{e,\theta}(x)}. \tag{25}$$

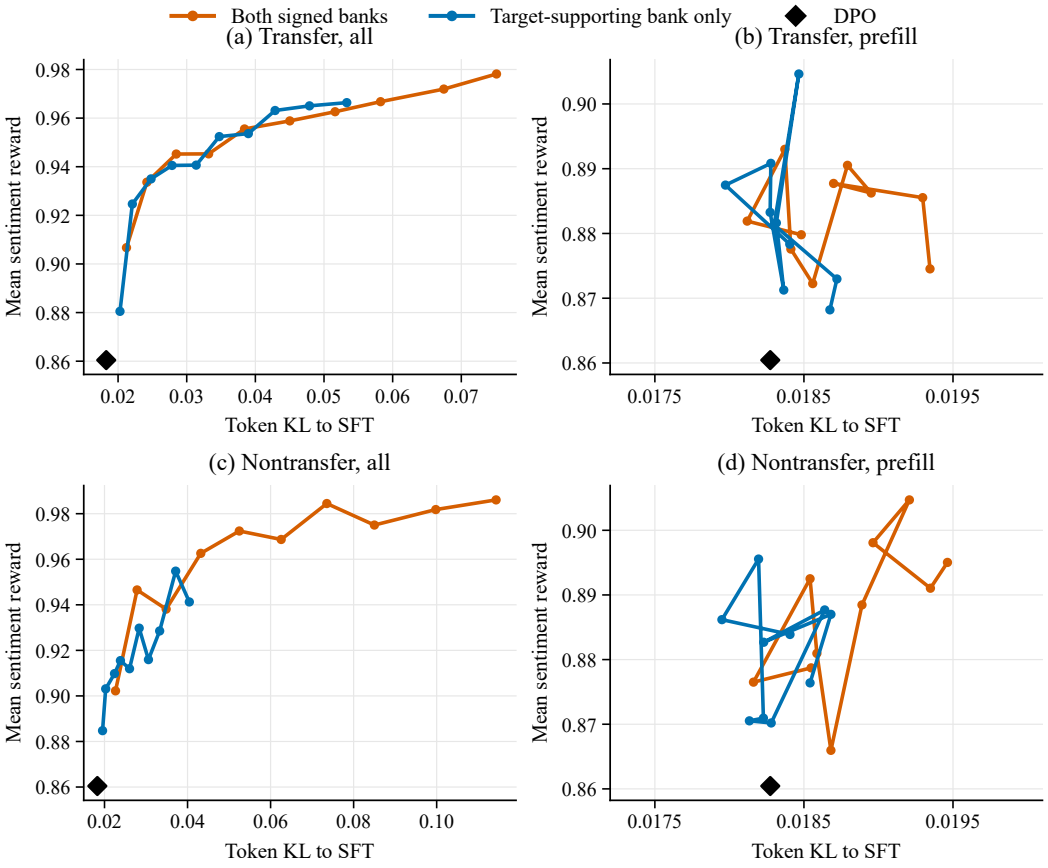

Figure 7: **IMDb DPO+CAST two-bank ablation on H100.** The original two-bank constant-vector controller (orange) is compared with a replay using only its target-supporting four-head bank (blue). Panels separate SFT-to-DPO transfer and DPO-start training, each under `all` and `prefill` timing. Every curve uses the same $\alpha \in \{0.1, 0.2, \ldots, 1.0\}$ grid on 2,048 fixed test prompts; the black diamond is DPO. Vertical axes show bounded sentiment reward; horizontal axes show token KL to the SFT reference. The grid describes response curves and is not a test-set selection rule.

Proposition 1 (DPO-form representation). For every response pair (y^+, y^-),

$$u_{e,\theta}(x, y^+) - u_{e,\theta}(x, y^-) = \log \frac{\bar{\pi}_{e,\theta}(y^+ \mid x)}{\pi_0(y^+ \mid x)} - \log \frac{\bar{\pi}_{e,\theta}(y^- \mid x)}{\pi_0(y^- \mid x)}. \tag{26}$$

Consequently, the pairwise term

$$-\log \sigma\big(\beta[u_{e,\theta}(x, y^+) - u_{e,\theta}(x, y^-)]\big) \tag{27}$$

is exactly the reference-relative logistic form used by DPO (Rafailov et al., 2023) when applied to the induced policy $\bar{\pi}_{e,\theta}$.

Proof. From Equation 25,

$$\log \frac{\bar{\pi}_{e,\theta}(y \mid x)}{\pi_0(y \mid x)} = u_{e,\theta}(x, y) - \log Z_{e,\theta}(x). \tag{28}$$

Subtracting the expressions for y^+ and y^- cancels the common normalization term $\log Z_{e,\theta}(x)$ and yields Equation 26. □

Relation to regularized reward optimization. For fixed θ, for any regularization coefficient $\eta > 0$, define an implicit reward

$$R_{e,\theta}(x, y) = \eta\, u_{e,\theta}(x, y). \tag{29}$$

For an arbitrary response distribution $\pi(\cdot \mid x)$, consider

$$J(\pi) = \mathbb{E}_{y \sim \pi(\cdot|x)}[R_{e,\theta}(x, y)] - \eta D_{\mathrm{KL}}(\pi(\cdot \mid x) \| \pi_0(\cdot \mid x)). \tag{30}$$

Substituting Equation 25 gives

$$J(\pi) = \eta \log Z_{e,\theta}(x) - \eta D_{\mathrm{KL}}(\pi(\cdot \mid x) \| \bar{\pi}_{e,\theta}(\cdot \mid x)), \tag{31}$$

so $\bar{\pi}_{e,\theta}$ is the unique optimizer whenever the usual support conditions hold. Because the implicit reward depends on θ, this characterizes the induced policy for a fixed controller; it is not a fixed-reward optimization theorem for training CAST.

This result concerns the *score-induced policy*. It does not imply that $\bar{\pi}_{e,\theta}$ equals CAST's actual autoregressive policy for every task-specific score.

If π_θ and π_0 are normalized on this same response space under the scoring protocol, and the response scores satisfy:

$$s_{e,\theta}(x, y) = \log \pi_\theta(y \mid x), \qquad s_{e,0}(x, y) = \log \pi_0(y \mid x), \tag{32}$$

then $Z_{e,\theta}(x) = 1$ and $\bar{\pi}_{e,\theta} = \pi_\theta$ exactly.

For the length-normalized score

$$s_{e,\theta}(x, y) = \frac{1}{|y|} \log \pi_\theta(y \mid x), \tag{33}$$

the induced policy remains well defined but generally differs from π_θ. The DPO-form equivalence in Equation 26 remains exact at the objective level.

D.2 Local Hidden-State Controllability

This section proves the single-site local-improvability statement used to motivate DHSA.

Fix a structural component i. Let $\Gamma_i \in \mathbb{R}^{d_i \times d_i}$ parameterize the shared local intervention

$$h_{t,i} \longmapsto h_{t,i} + \Gamma_i h_{t,i}, \tag{34}$$

and let $J_e(\Gamma_i) \in \mathbb{R}$ be a preference objective of the resulting controlled model, with larger values preferred.

Define

$$G_{i,e} = \nabla_{\Gamma_i} J_e(\Gamma_i)|_{\Gamma_i = 0}. \tag{35}$$

Assume J_e is locally L_i-smooth in Frobenius norm around $\Gamma_i = 0$, i.e.,

$$\|\nabla J_e(\Gamma) - \nabla J_e(\Gamma')\|_F \le L_i \|\Gamma - \Gamma'\|_F \tag{36}$$

throughout a Frobenius ball of radius $r_i > 0$ around the origin.

Proposition 2 (single-site rank-one improvement). If $G_{i,e} \neq 0$, then there exists a rank-one matrix $D_i = u_i v_i^\top$ with $\|D_i\|_F = 1$ and an $\epsilon_0 > 0$ such that

$$J_e(\epsilon D_i) > J_e(0) \qquad \text{for all } 0 < \epsilon < \epsilon_0. \tag{37}$$

More specifically, let

$$\gamma_i = \sigma_{\max}(G_{i,e}) > 0, \tag{38}$$

and let u_i, v_i be corresponding unit singular vectors. Then $D_i = u_i v_i^\top$ satisfies

$$\langle G_{i,e}, D_i \rangle_F = \gamma_i. \tag{39}$$

By the standard smoothness lower bound (Nocedal & Wright, 2006),

$$J_e(\epsilon D_i) - J_e(0) \geq \epsilon \gamma_i - \frac{L_i}{2}\epsilon^2. \tag{40}$$

Hence any

$$0 < \epsilon < \min\left\{ r_i, \frac{2\gamma_i}{L_i} \right\}, \tag{41}$$

strictly improves the local preference objective. □

The proposition is an existence statement over the admissible local state-control space. It does not guarantee that every interface satisfies $G_{i,e} \neq 0$, that an RCM-selected interface satisfies the premise, or that a single interface can reproduce an arbitrary globally optimized policy.

Constant-vector special case. For an additive controller $h_{t,i} \mapsto h_{t,i} + v_i$ with trainable $v_i \in \mathbb{R}^{d_i}$, the same argument applies directly in vector space. If

$$\nabla_{v_i} J_e(v_i)|_{v_i=0} \neq 0, \tag{42}$$

then a sufficiently small step in the normalized gradient direction strictly improves J_e.

Thus both the constant and low-rank CAST actuator classes can contain local improvement directions when their corresponding local gradients are nonzero.

D.3 RCM Effects as Local-Sensitivity Proxies

This section formalizes the weaker connection between finite RCM effects and local controllability. It does not claim that RCM identifies an optimal control site.

Let ξ_i^a and ξ_i^b denote the two registered internal quantities compared by an RCM contrast. Depending on the contrast, ξ_i can be a component state or a residual write. Define the interpolation

$$\xi_i(\lambda) = (1-\lambda)\xi_i^a + \lambda \xi_i^b, \qquad \lambda \in [0,1], \tag{43}$$

and

$$\phi_{i,e,x}(\lambda) = C_e\big(M_{i \leftarrow \xi_i(\lambda)}; x\big). \tag{44}$$

Assume $\phi_{i,e,x}$ is absolutely continuous on $[0,1]$ and differentiable almost everywhere. Then the fundamental theorem of calculus gives

$$\Delta_{i,e}^{a \to b}(x) = \phi_{i,e,x}(1) - \phi_{i,e,x}(0) = \int_0^1 \phi'_{i,e,x}(\lambda)\, d\lambda. \tag{45}$$

When C_e is differentiable with respect to ξ_i,

$$\phi'_{i,e,x}(\lambda) = \left\langle \nabla_{\xi_i} C_e,\, \xi_i^b - \xi_i^a \right\rangle, \tag{46}$$

where the gradient is evaluated at the intervened state $\xi_i(\lambda)$. This is the same path-integral identity underlying path-based attribution methods (Sundararajan et al., 2017).

Proposition 3 (nonzero RCM implies path sensitivity). Under the conditions above,

$$\Delta_{i,e}^{a \to b}(x) \neq 0 \quad \Longrightarrow \quad \exists \lambda^\star \in (0,1) \text{ such that } \phi'_{i,e,x}(\lambda^\star) \neq 0. \tag{47}$$

Proof. If $\phi'_{i,e,x}(\lambda) = 0$ almost everywhere on $[0,1]$, then Equation 45 gives $\Delta_{i,e}^{a\to b}(x) = 0$, contradicting the premise. □

If ϕ' is continuous, the mean-value theorem gives the stronger statement

$$\exists \lambda^\star \in (0,1): \qquad \phi'_{i,e,x}(\lambda^\star) = \Delta_{i,e}^{a\to b}(x), \tag{48}$$

because the path parameter has unit length.

Endpoint sensitivity. Suppose additionally that ϕ' is L_{path}-Lipschitz in λ:

$$|\phi'(\lambda) - \phi'(\lambda')| \le L_{\mathrm{path}}|\lambda - \lambda'|. \tag{49}$$

Then

$$\left|\phi'(1) - \Delta_{i,e}^{a\to b}(x)\right| \le \frac{L_{\mathrm{path}}}{2}. \tag{50}$$

Therefore, if

$$\left|\Delta_{i,e}^{a\to b}(x)\right| > \frac{L_{\mathrm{path}}}{2}, \tag{51}$$

the endpoint directional derivative is necessarily nonzero and has the same sign as the finite RCM contrast.

This additional smoothness condition is stronger than that required for Proposition 3 and is not assumed for all experiments.

Zero and Patch paths. For RCM-Zero,

$$\xi_i^a = 0, \qquad \xi_i^b = z_i^{\mathrm{native}}, \tag{52}$$

so the contrast integrates sensitivity along the path from write removal to the native residual write.

For RCM-Patch,

$$\xi_i^a = h_i^{\mathrm{ref}}, \qquad \xi_i^b = h_i^{\mathrm{src}}, \tag{53}$$

so the contrast integrates sensitivity from the receiving state to the specified source state. For a prototype patch, h_i^{src} is replaced by $\bar{h}_{i,e}^{\mathrm{src}}$ from Equation 20. The reported Site experiments use this prototype form.

Stochastic generation protocols. The path-integral statement applies directly to deterministic differentiable scores and, more generally, to an expected score when differentiation may be exchanged with the expectation. A single realized discrete generation is not a differentiable function of the intervention state. For such protocols, RCM remains a well-defined finite intervention contrast, but Equation 45 is not invoked for that individual sampled realization. This distinction is especially relevant when C_e is evaluated through sampled open generation followed by an external reward model.