ROLE-CONSISTENT ATTENTION STEERING FOR HIERARCHICAL INSTRUCTION FOLLOWING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models often violate the instruction hierarchy by following lower-priority instructions that conflict with higher-priority ones. Attention steering can mitigate these violations by emphasizing higher-priority messages, but may also suppress lower-priority context needed for the task. Our analysis shows that this conflict–preservation trade-off depends strongly on head selection, with larger conflict benefits tending to incur larger preservation costs. We introduce **Ro**le-**C**onsistent **A**ttention **St**eering (**RoCAST**), a training-free method that selects a fixed subset of attention heads offline for role-based reweighting. RoCAST scores heads using conflict benefit across role-swapped instructions, role consistency, and preservation without conflict, with calibration requiring only 64 instruction pairs and 250 unlabeled paragraphs. Across five Llama and Qwen models spanning 3B–14B parameters, RoCAST improves IHEval conflict accuracy by 21.3–33.1 percentage points over native models and outperforms the evaluated inference-time baselines. It keeps aligned task execution within 1.3 points of native on every model, improves system-level instruction following, and incurs smaller capability losses than the baselines. Ablations show that role consistency and preservation improve both conflict and aligned performance over conflict-only selection. The same calibrated head sets transfer without recalibration to thinking-enabled inference and hierarchy-trained checkpoints, yielding further conflict gains while largely preserving aligned performance. Together, these results show that resolving instruction conflicts without sacrificing compatible behavior depends critically on which heads are steered, supporting offline head calibration as a lightweight complement to hierarchy-specific training.

1 INTRODUCTION

Large language models (LLMs) commonly receive inputs as messages labeled with roles such as system, user, and tool. The instruction hierarchy specifies that instructions from higher-priority roles should prevail when conflicts arise, with the priority order system $>$ user $>$ tool (Wallace et al., 2024). However, priority does not make lower-priority information irrelevant. User inputs and tool outputs often contain information needed to carry out higher-priority instructions. Reliable hierarchical instruction following therefore requires both resolving conflicts in favor of higher-priority roles and retaining useful lower-priority content when instructions are compatible. Current models often fail to prioritize higher-priority instructions when messages conflict, even when they readily follow compatible instructions (Zhang et al., 2025; Moon et al., 2026; Kariyappa & Suh, 2026).

Training-based approaches improve hierarchy adherence with synthetic demonstrations (Wallace et al., 2024), preference optimization (Mu et al., 2025), or reinforcement learning with verifiable rewards (Huang et al., 2025; Zheng et al., 2026; Guo et al., 2026), but rely on hierarchy-specific data construction and additional optimization. Inference-time approaches instead modify model computations during generation, for example through contrastive decoding (Liu et al., 2026) or cached-value rescaling (Zeng et al., 2026). We focus on attention steering, which reweights attention toward target input spans at inference time (Zhang et al., 2024; Venkateswaran & Contractor, 2026). Because chat templates explicitly mark message roles, attention can be shifted according to role priority without annotating instruction spans or updating model parameters. However, increasing attention to higher-priority instructions can also suppress task-relevant context that should still be incorporated (Guardieiro et al., 2025).

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.

We investigate whether this trade-off depends on *which* attention heads are steered. Existing head-selection methods choose heads based either on how much steering improves task performance on calibration data (Zhang et al., 2024) or on input-specific attribution (Zeng et al., 2026). These criteria do not explicitly consider whether a head's effect is tied to message role or whether steering it preserves behavior when no conflict is present. Our analysis shows that heads with larger conflict benefits tend to incur larger preservation costs, and selecting heads by conflict benefit alone can substantially degrade performance when instructions are compatible.

We introduce **Ro**le-**C**onsistent **A**ttention **St**eering (**RoCAST**), a training-free method that selects a fixed subset of attention heads offline for role-based attention reweighting. RoCAST combines three complementary criteria. **Behavioral Benefit Scoring (BBS)** evaluates heads across role-swapped conflicts, rewarding gains shared across role assignments while penalizing systematic imbalance. **Role-Consistency Filtering (RCF)** compares each head's response to identical content placed in different roles and retains heads whose response is consistent with the aggregate role response of other candidates. **System-Only Preservation Scoring (SPS)** favors heads whose intervention preserves system-side preference when no conflict is present. At inference time, the selected heads are reweighted according to role priority while preserving the total attention mass over message bodies and the relative weights within each role. Calibration requires only 64 instruction pairs and 250 unlabeled paragraphs, after which the head set and steering strength remain fixed across inputs.

Across five Llama and Qwen models spanning 3B–14B parameters, RoCAST improves IHEval conflict accuracy by 21.3–33.1 percentage points over native models and outperforms the evaluated inference-time baselines. Random-head steering with the same operator, strength, and budget gains only 8.5 points on average, compared with 26.3 for RoCAST, highlighting the contribution of head selection. While inference-time baselines can lose more than 40 points on compatible task instructions, RoCAST keeps aligned task execution within 1.3 points of native on every model. It also improves system-level instruction following with smaller capability losses than the baselines. Ablations show that conflict-only selection can perform worse than random heads on aligned inputs, whereas adding role consistency and preservation improves both conflict and aligned performance. The same calibrated head sets transfer without recalibration to thinking-enabled inference and hierarchy-trained checkpoints while largely preserving aligned and capability performance.

Together, these results show that the conflict–preservation trade-off in attention steering is shaped not simply by how strongly or broadly the model is steered, but by *which heads* are selected for intervention. RoCAST addresses this trade-off through lightweight offline calibration, providing a reusable complement to hierarchy-specific training without additional parameter updates.

2 Related Work

Hierarchical Instruction Following. Wallace et al. (2024) formalized the instruction hierarchy to improve LLM safety and controllability by giving precedence to instructions from higher-priority sources when conflicts arise. Benchmarks such as SysBench and IHEval reveal hierarchy violations (Qin et al., 2024; Zhang et al., 2025). Analyses of these violations identify constraint-specific biases (Geng et al., 2026) and language-dependent compliance (Moon et al., 2026). To improve adherence, training-based approaches use synthetic demonstrations (Wallace et al., 2024), realistic system-prompt data and preference optimization (Mu et al., 2025), and reinforcement learning with verifiable rewards (Huang et al., 2025; Zheng et al., 2026; Guo et al., 2026). At the architectural level, Instructional Segment Embedding encodes priority through learned role-specific embeddings (Wu et al., 2025). We use hierarchy data for offline attention-head calibration rather than parameter updates, and evaluate steering as a complement to hierarchy-specific training.

Inference-Time Interventions. At the decoding level, Mu et al. (2025) evaluate classifier-free guidance using predictions with and without system instructions, while IHDec (Liu et al., 2026) uses role-ablated predictions for contrastive decoding. V-Steer (Zeng et al., 2026) rescales cached values, whereas DIRECTER (Kang & Kim, 2026) combines key-cache scaling with plausibility-guided decoding. Split-softmax (Li et al., 2024), PASTA (Zhang et al., 2024), SpotLight (Venkateswaran & Contractor, 2026), and InstABoost (Guardieiro et al., 2025) directly reweight attention toward system prompts or specified spans. RoCAST confines role-based reweighting to prompt message bodies, preserving their combined attention mass and leaving attention to other positions, including generated tokens, unchanged at each edited operation.

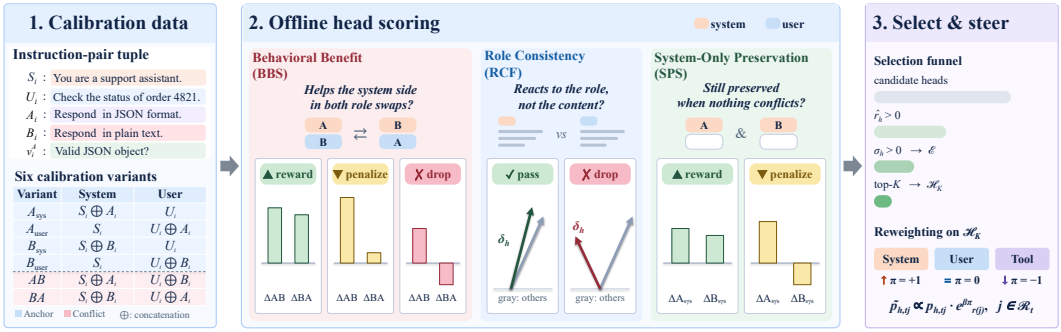

Figure 1: RoCAST selects attention heads through offline calibration and applies role-based attention reweighting at inference time, without parameter updates or input-specific selection.

Head Selection for Targeted Intervention. PASTA (Zhang et al., 2024) selects heads through offline steering-performance profiling, V-Steer (Zeng et al., 2026) uses input-specific direct logit attribution, and FocalLoRA (Shi et al., 2025) targets conflict-sensitive heads for parameter-efficient fine-tuning. Content-controlled role probes further reveal discrepancies between inferred roles and message labels (Ye et al., 2026). RoCAST instead selects a fixed head set offline using intervention benefits across role-swapped conflicts, role consistency, and preservation under uncontested system instructions, directly targeting the conflict–preservation trade-off without input-specific attribution.

3 RoCAST: Role-Consistent Attention Steering

RoCAST combines a fixed role-based attention intervention with a three-stage offline procedure for selecting the attention heads on which to apply it. The three selection stages are **Behavioral Benefit Scoring (BBS)**, **Role-Consistency Filtering (RCF)**, and **System-Only Preservation Scoring (SPS)**. BBS identifies heads with shared conflict benefits, while RCF and SPS refine the selection for role consistency and preservation without conflict. We first define the steering operator and calibration setup shared by these stages.

3.1 Steering and Calibration Setup

Role-Based Attention Reweighting. RoCAST shifts attention toward higher-priority roles while keeping the total attention on message-body tokens unchanged. Within each role, it preserves the relative attention among tokens. For query position t, let $\mathcal{R}_t$ denote the visible key positions belonging to message bodies, and let $r(j)$ denote the role of position j as identified by the chat template. Let $p_{h,tj}$ be the native attention weight of head h from query t to key j, and let $M_{h,t} = \sum_{k\in\mathcal{R}_t} p_{h,tk}$ be the total attention mass over these positions. RoCAST redistributes this mass across roles using a fixed role bias:

$$\widetilde{p}_{h,tj} = \begin{cases} M_{h,t}\dfrac{p_{h,tj}e^{\beta_h \pi_{r(j)}}}{\sum_{k\in\mathcal{R}_t} p_{h,tk}e^{\beta_h \pi_{r(k)}}}, & j \in \mathcal{R}_t, \\ p_{h,tj}, & j \notin \mathcal{R}_t, \end{cases} \qquad \pi_r = \begin{cases} +1, & r = \text{system}, \\ -1, & r = \text{tool}, \\ 0, & \text{otherwise}. \end{cases} \tag{1}$$

The role bias favors system tokens and suppresses tool tokens, with β_h controlling the intervention strength. Positions outside $\mathcal{R}_t$ retain their native attention. After head selection, $\beta_h = \beta$ for selected heads and $\beta_h = 0$ otherwise.

Calibration Data Construction. We construct model-specific calibration sets from the synthetic instruction-pair data of Huang et al. (2025). Each candidate provides a system scaffold S_i, a user task U_i, opposing instructions A_i and B_i, and a verifier v_i^A for A_i. Keeping S_i and U_i fixed, we derive six calibration variants by assigning A_i and B_i to the system or user message. As illustrated in Figure 1, these comprise four single-instruction anchors and two role-swapped conflicts.

For each target model, we greedily generate native responses Y_i^a to the four anchor prompts and retain 64 instruction pairs satisfying

$$v_i^A(Y_i^{A_{\text{sys}}}) = v_i^A(Y_i^{A_{\text{user}}}) = 1, \qquad v_i^A(Y_i^{B_{\text{sys}}}) = v_i^A(Y_i^{B_{\text{user}}}) = 0. \tag{2}$$

3.2 Behavioral Benefit Scoring

Behavioral Benefit Scoring (BBS) identifies heads whose intervention benefits conflict resolution across role assignments, rather than only one particular configuration. We therefore evaluate both AB and BA, swapping the system and user roles of A_i and B_i, and reward gains shared across this role swap while penalizing systematic imbalance.

First-Divergence Token Selection. For each conflict configuration $\kappa \in \{AB, BA\}$, we compare the native response to the system-side anchor with that to the opposing user-side anchor. Let LCP denote the longest common token prefix:

$$\xi_i^{AB} = \mathrm{LCP}(Y_i^{A_{\mathrm{sys}}}, Y_i^{B_{\mathrm{user}}}), \qquad \xi_i^{BA} = \mathrm{LCP}(Y_i^{B_{\mathrm{sys}}}, Y_i^{A_{\mathrm{user}}}). \tag{3}$$

Let $y_{i,\mathrm{sys}}^{\kappa}$ and $y_{i,\mathrm{usr}}^{\kappa}$ be the first diverging tokens after ξ_i^{κ} in the corresponding system-side and user-side anchor responses. We append ξ_i^{κ} to the conflict prompt as an assistant continuation and compare the logits of these two tokens. Thus, the anchors define the competing candidates, while their relative preference is evaluated under the corresponding conflict prompt. The prefix and candidate pair are computed once under the native model and reused across all heads, ensuring that each head is evaluated on the same next-token decision.

Single-Head Intervention Effect. Let x_i^{κ} denote the conflict prompt for configuration κ, and let $\ell_y(x; h, \beta)$ denote the next-token logit of token y when an intervention of strength β is applied to head h. Setting $\beta = 0$ recovers the native model. We define the system-preference margin as

$$M_i^{\kappa}(x; h, \beta) = \ell_{y_{i,\mathrm{sys}}^{\kappa}}(x \oplus \xi_i^{\kappa}; h, \beta) - \ell_{y_{i,\mathrm{usr}}^{\kappa}}(x \oplus \xi_i^{\kappa}; h, \beta)\,. \tag{4}$$

Here, $\oplus$ appends the assistant continuation. A larger $M_i^{\kappa}(x; h, \beta)$ indicates a stronger next-token preference for the system-side candidate. The intervention effect relative to the native model is

$$\Delta_h^{\kappa}(i) = M_i^{\kappa}(x_i^{\kappa}; h, \beta) - M_i^{\kappa}(x_i^{\kappa}; h, 0), \qquad \kappa \in \{AB, BA\}. \tag{5}$$

Thus, $\Delta_h^{\kappa}(i) > 0$ indicates a shift toward the system-side candidate relative to the native model.

Swap-Paired Benefit. A desirable intervention should produce a positive effect shared across the two role-swapped conflict configurations, rather than one driven by a particular placement of A_i and B_i. For each instruction pair, define

$$r_h(i) = \frac{\Delta_h^{AB}(i) + \Delta_h^{BA}(i)}{2}, \qquad c_h(i) = \frac{\Delta_h^{AB}(i) - \Delta_h^{BA}(i)}{2}. \tag{6}$$

Here, $r_h(i)$ captures the effect shared across the two configurations, whereas $c_h(i)$ captures their imbalance; its sign indicates which configuration benefits more. We aggregate these pair-level statistics with medians:

$$\widehat{r}_h = \mathrm{median}_i\, r_h(i), \qquad \widehat{c}_h = \mathrm{median}_i\, c_h(i), \qquad b_h = \widehat{r}_h - |\widehat{c}_h|. \tag{7}$$

Medians limit the influence of a small number of pairs with unusually large effects. We take the absolute value after aggregation so that $|\widehat{c}_h|$ penalizes a systematic preference for one configuration rather than pair-specific asymmetries whose signs vary across examples. We require $\widehat{r}_h > 0$ to retain heads with a positive shared component. For these heads, b_h rewards a large shared effect while penalizing systematic imbalance across configurations.

3.3 Role-Consistency Filtering

BBS identifies heads whose intervention yields a positive behavioral effect across the two role-swapped conflict configurations. However, this alone cannot determine whether the benefit reflects role-sensitive processing or simply increased attention to tokens within the system span. Since the intervention itself explicitly biases attention by message role, behavioral improvement is not sufficient to establish a role-consistent internal response.

We therefore introduce Role-Consistency Filtering (RCF). Using a content-controlled role comparison similar to Ye et al. (2026), we first measure how each head's residual contribution changes when identical neutral content is assigned different message roles. We then retain heads whose role response is directionally aligned with the aggregate response of the other candidate heads.

Content-Controlled Role Response. For each neutral paragraph $c \in \mathcal{C}$, we render the same content under system and user roles. We define

$$w_h(x) = \frac{1}{|I_x|} \sum_{t \in I_x} W_O^{(h)} o_{h,t}(x), \qquad \delta_h = \mathbb{E}_{c \in \mathcal{C}} \left[w_h(x_c^{\text{sys}}) - w_h(x_c^{\text{user}}) \right], \tag{8}$$

where I_x indexes paragraph tokens, $o_{h,t}(x)$ is the native output of head h at position t, and $W_O^{(h)}$ projects it into the residual stream. Here, $w_h(x)$ is the pooled residual write of head h, and δ_h is its *role-response vector*, measuring the average change under role reassignment while holding textual content fixed.

Role-Consistency Criterion. Prior probe analyses show that message role is linearly decodable from residual representations (Ye et al., 2026), motivating the hypothesis that role-conditioned head responses contain a shared directional component. Rather than learning a probe direction, we estimate this component from the other heads in the candidate set $\mathcal{H}_{\text{cand}}$:

$$\bar{\delta}_{-h} = \sum_{g \in \mathcal{H}_{\text{cand}} \setminus \{h\}} \delta_g, \qquad \sigma_h = \operatorname{sign} \left\langle \delta_h, \bar{\delta}_{-h} \right\rangle. \tag{9}$$

A positive σ_h indicates that head h responds to the role change in the same direction as the cross-head aggregate. The leave-one-head-out construction prevents trivial self-agreement. Combining this criterion with the positive shared behavioral effect from BBS defines the eligible set

$$\mathcal{E} = \{ h \in \mathcal{H}_{\text{cand}} : \widehat{r}_h > 0 \ \wedge \ \sigma_h > 0 \}. \tag{10}$$

3.4 System-Only Preservation Scoring

An intervention can increase system-side margins under conflict yet reduce them on non-conflicting inputs. System-Only Preservation Scoring (SPS) therefore measures whether the same intervention preserves system-side preference under uncontested system-anchor conditions. We use A_{sys} and B_{sys}, where the corresponding target instruction appears in the system message while the original task scaffold remains in the user message.

For each system anchor, we reuse the prefix and candidate pair from the corresponding conflict configuration. For the matched pairs $(a, \kappa) \in \{(A_{\text{sys}}, AB), (B_{\text{sys}}, BA)\}$, we define

$$\Delta_h^a(i) = M_i^\kappa(x_i^a; h, \beta) - M_i^\kappa(x_i^a; h, 0). \tag{11}$$

Each margin is oriented toward the instruction present in the system anchor. The system-only preservation score is

$$p_h = \min \left\{ \text{median}_i \, \Delta_h^{A_{\text{sys}}}(i), \ \text{median}_i \, \Delta_h^{B_{\text{sys}}}(i) \right\}. \tag{12}$$

The minimum is taken after computing the median for each anchor condition, so improvement on one instruction cannot compensate for degradation on the other. Larger p_h indicates better preservation, whereas $p_h < 0$ indicates that the intervention reduces the system-side margin under at least one uncontested system-anchor condition.

Final Ranking and Deployment. For heads in $\mathcal{E}$, we combine the imbalance-aware behavioral score b_h with the system-only preservation score p_h after standardizing both over the eligible set:

$$\text{score}_h = z_{\mathcal{E}}(b_h) + \lambda z_{\mathcal{E}}(p_h), \tag{13}$$

where $z_{\mathcal{E}}(\cdot)$ denotes standardization using the mean and standard deviation over heads in $\mathcal{E}$. The coefficient λ controls the relative weight of system-only preservation. For a head budget K, we select

$$\mathcal{H}_K = \text{TopK}_{h \in \mathcal{E}} \, \text{score}_h, \qquad \beta_h = \beta \, \mathbf{1}[h \in \mathcal{H}_K]. \tag{14}$$

After calibration, the selected head set $\mathcal{H}_K$ remains fixed and is reused across test inputs.

Table 1: Think-off results on IHEval and capability benchmarks. IHEval scores (%) are reported by domain (RF: rule following, TE: task execution, Saf.: safety), with the 9-task average and change from native. †IHDec Avg. and Δ use the seven non-tool tasks.

Model	Method	IHEval conflict						IHEval aligned						MMLU	MATH-500	IFEval-sys
		RF	TE	Saf.	Tool	Avg.	Δ	RF	TE	Saf.	Tool	Avg.	Δ			
Llama-3.2-3B	Native	13.0	4.7	37.6	1.3	13.1		60.2	70.1	41.6	2.2	46.5		64.5	47.2	55.5
	IHDec†	51.4	21.6	38.5	–	34.9	+18.4	65.8	30.7	46.6	–	45.3	−13.8	61.7	32.6	58.4
	V-Simple	**65.9**	32.4	**64.0**	1.0	39.9	+26.8	69.8	23.3	**77.2**	5.2	41.6	−4.9	57.3	20.0	57.3
	Random heads	25.9	26.5	47.1	**2.7**	25.7	+12.6	63.1	67.3	42.4	**7.1**	47.5	+1.0	**64.5**	46.0	57.5
	RoCAST (ours)	52.3	**56.8**	41.2	2.0	**40.1**	**+27.0**	**73.5**	**71.4**	35.5	1.8	**48.4**	**+1.9**	**64.5**	**48.6**	**66.4**
Llama-3.1-8B	Native	17.8	5.7	28.4	1.9	12.6		67.7	76.2	55.2	3.0	53.4		70.0	48.2	63.0
	IHDec†	54.5	10.7	43.5	–	32.6	+17.0	72.9	22.2	61.1	–	47.8	−20.0	**71.1**	37.2	67.1
	V-Simple	**71.3**	26.2	**61.3**	4.0	39.1	+26.5	76.0	28.4	**68.0**	0.4	41.5	−11.9	62.3	20.8	67.7
	Random heads	28.3	32.8	27.5	1.7	23.7	+11.1	72.4	72.9	45.1	0.6	50.5	−2.9	70.0	**46.0**	67.8
	RoCAST (ours)	51.0	**60.6**	51.3	**12.5**	**45.7**	**+33.1**	**77.5**	**76.7**	49.6	**1.6**	**54.2**	**+0.8**	70.0	45.4	**70.2**
Qwen3-4B	Native	23.9	31.8	8.2	6.3	19.1		81.0	77.5	72.4	68.1	75.1		70.8	77.4	73.9
	IHDec†	44.5	46.9	16.5	–	37.5	+14.7	82.9	74.2	69.1	–	75.2	−1.9	70.6	**76.8**	75.0
	V-Simple	**72.4**	14.2	**45.5**	6.6	32.4	+13.3	72.6	17.8	78.9	23.3	44.8	−30.3	64.5	32.8	63.4
	Random heads	26.0	43.8	15.6	**21.7**	28.7	+9.6	80.9	73.9	**79.1**	**55.6**	**72.5**	**−2.6**	**70.8**	71.8	75.2
	RoCAST (ours)	60.8	**56.8**	31.8	13.8	**42.6**	**+23.5**	**87.5**	**77.2**	75.1	27.8	68.1	−7.0	**70.8**	68.8	**77.4**
Qwen3-8B	Native	27.1	45.0	14.3	4.7	25.2		86.3	76.9	81.4	66.5	77.6		74.4	77.6	79.7
	IHDec†	45.3	52.1	21.2	–	41.4	+10.3	86.7	70.6	80.4	–	78.0	−2.8	74.1	**74.8**	80.6
	V-Simple	73.1	22.3	**44.8**	2.9	34.3	+9.1	73.8	24.1	**86.2**	37.0	51.8	−25.8	69.0	46.6	63.6
	Random heads	33.4	51.2	19.8	**21.8**	33.7	+8.5	86.5	66.9	81.1	**53.0**	**71.3**	**−6.3**	**74.4**	73.8	78.2
	RoCAST (ours)	**74.7**	**67.9**	35.7	20.8	**51.8**	**+26.6**	**88.7**	**76.3**	80.0	26.8	68.9	−8.7	**74.4**	72.8	**81.7**
Qwen3-14B	Native	30.2	55.5	16.2	2.7	29.4		88.0	83.5	85.1	78.8	83.8		79.7	80.2	79.7
	IHDec†	53.0	60.8	24.6	–	48.2	+11.1	88.1	82.2	70.5	–	80.5	−4.7	79.5	74.6	80.0
	V-Simple	**74.2**	13.3	**64.6**	0.9	35.5	+6.1	75.5	11.9	**90.3**	65.7	55.4	−28.4	77.1	26.0	69.1
	Random heads	35.5	53.5	15.7	3.4	30.0	+0.6	88.1	82.0	80.3	**76.3**	**81.7**	**−2.1**	**79.7**	**80.8**	80.4
	RoCAST (ours)	73.1	**74.7**	26.3	**16.8**	**50.7**	**+21.3**	**90.8**	**83.2**	42.9	53.0	69.2	−14.6	**79.7**	80.4	**84.3**

4 EXPERIMENTS

Experimental setup. Our experiments cover the Llama Instruct models Llama-3.2-3B and Llama-3.1-8B (Grattafiori et al., 2024), together with Qwen3 models at 4B, 8B, and 14B scales (Yang et al., 2025). Instruction-hierarchy performance is evaluated on IHEval (Zhang et al., 2025), while 5-shot MMLU (Hendrycks et al., 2021a), MATH-500 (Hendrycks et al., 2021b; Lightman et al., 2024), and IFEval-sys (Zhou et al., 2023) evaluate general capability and system-level instruction following. We compare RoCAST with the native model, IHDec (Liu et al., 2026), V-Simple (Zeng et al., 2026), and a random-head control using the same intervention strength and head budget. We fix $\beta = 4$ and $\lambda = 1$ across models. The head budget is selected on held-out calibration pairs, yielding $K = 96$ for all five models. The full experimental protocol is provided in Appendix A.

4.1 HIERARCHY COMPLIANCE AND PRESERVATION

We first ask whether stronger instruction-hierarchy compliance can be obtained without broadly degrading behavior when instructions are compatible. Table 1 reports the think-off results across all five models. We consider conflict and aligned performance jointly, since an intervention that resolves conflicts only by suppressing otherwise useful instructions would provide a poor hierarchy–preservation trade-off.

Conflict–preservation trade-off. Averaged across the five models, RoCAST improves IHEval conflict accuracy by 26.3 percentage points over native performance and by 17.8 points over random-head steering. Since random-head steering applies the same intervention, the 17.8 point gap highlights the contribution of RoCAST's calibrated head selection. On Llama, RoCAST outperforms IHDec and V-Simple in conflict resolution while also improving aligned averages over native performance. Both baselines instead degrade aligned performance. RoCAST also achieves the highest conflict performance on all three Qwen3 models, with smaller aligned losses than V-Simple. Ro-

Table 2: Think-on results for Qwen3-4B and Qwen3-8B. Δ is measured relative to the native model.

Model	Method	IHEval conflict						IHEval aligned						MMLU	MATH-500	IFEval-sys
		RF	TE	Saf.	Tool	Avg.	Δ	RF	TE	Saf.	Tool	Avg.	Δ			
Qwen3-4B (think-on)	Native	23.9	39.2	29.1	39.3	33.6		82.3	81.7	98.4	78.2	84.8		77.0	85.2	76.5
	VerIH	48.0	54.3	29.3	**47.0**	45.7	+12.1	88.5	84.0	97.6	**77.7**	**86.6**	**+1.8**	**78.7**	**89.2**	**80.2**
	RoCAST (ours)	45.8	66.9	49.3	42.1	52.8	+19.2	84.6	82.5	97.0	77.6	85.1	+0.3	77.0	83.2	77.8
	VerIH + RoCAST (ours)	**70.0**	**72.6**	**49.4**	39.4	**59.5**	**+25.9**	**88.9**	**84.6**	**97.9**	75.3	86.4	+1.6	**78.7**	87.8	79.7
Qwen3-8B (think-on)	Native	28.8	49.1	28.4	13.6	32.1		87.8	86.1	97.6	81.5	88.0		79.2	82.0	77.8
	VerIH	47.4	65.6	31.7	26.1	45.2	+13.1	90.5	85.6	96.2	**81.0**	88.0	+0.0	**81.8**	**88.6**	**81.5**
	RoCAST (ours)	64.0	70.3	49.5	37.8	57.1	+25.0	88.6	87.6	**97.1**	78.7	88.0	+0.0	79.2	85.8	78.9
	VerIH + RoCAST (ours)	**75.4**	**78.4**	**50.6**	**43.2**	**63.7**	**+31.6**	**91.8**	**88.3**	96.7	76.0	**88.2**	+0.2	**81.8**	**88.6**	81.1

CAST's remaining aligned loss averages 10.1 points relative to native and is concentrated mainly in safety and tool use. The advantage is particularly clear in IHEval's task-execution domain, which requires following system instructions while using information provided in user messages. RoCAST scores highest among the evaluated inference-time methods under both conflicting and aligned instructions on every model, with aligned task-execution averages remaining within 1.3 points of native performance. These results show that RoCAST can prioritize system instructions while effectively using task information from user messages.

Instruction following and capability. RoCAST achieves the highest IFEval-sys accuracy among the compared methods on every model, improving over native by 5.6 points on average. On MMLU, the prompts contain no system or tool messages, so RoCAST leaves attention unchanged and preserves native accuracy. In contrast, IHDec and V-Simple reduce accuracy by 0.5 and 5.8 points on average, respectively. On MATH-500, RoCAST incurs a mean decrease of 2.9 points relative to native performance, compared with 6.9 for IHDec and 36.9 for V-Simple, although effects vary by model. Under the evaluated configurations, RoCAST therefore combines stronger system-level instruction following with smaller average losses on the general-capability benchmarks.

4.2 Transfer across Thinking Modes and Hierarchy Training

For Qwen3-4B and Qwen3-8B, we reuse each native model's think-off head set and intervention strength for thinking-enabled inference on both the native and RLVR-trained VerIH checkpoints (Zheng et al., 2026), without recalibration. Table 2 reports the results.

Transfer across thinking modes. RoCAST's conflict benefit transfers to think-on inference without mode-specific calibration, improving across all four conflict domains on both native models. Importantly, these gains coexist with nearly unchanged aligned averages, in contrast to the aggregate aligned losses under think-off inference in Table 1. Relative to the native baseline in each mode, the MATH-500 penalty also narrows on Qwen3-4B and becomes a gain on Qwen3-8B, while IFEval-sys improves over the think-on native baselines. Thus, enabling thinking does not remove the benefit of the think-off-calibrated intervention, but the associated preservation cost is reduced in the evaluated setting. These results suggest that the selected heads remain useful for priority resolution across modes, while the resulting conflict–preservation trade-off varies with the inference setting.

Complementarity with hierarchy training. RoCAST alone achieves higher conflict averages than VerIH, and their combination improves them further on both models. These gains show that hierarchy-specific training does not make native-calibrated attention steering redundant. The complementarity is clearest in rule following and task execution, where the combination improves both conflict and aligned performance over either method alone. In task execution, these paired improvements support the intended balance between resolving competing instructions and retaining useful information from user messages. The combination also largely retains VerIH's overall aligned performance and its capability and instruction-following scores. The benefit is nevertheless domain-dependent. Conflict safety performance remains close to that of RoCAST alone, while aligned tool-use accuracy decreases relative to VerIH. Taken together, these results show that calibration on the native model can support further improvements after hierarchy training while largely retaining the trained model's performance, without additional parameter updates or head reselection.

Table 3: Comparison with training-based methods on Llama-3.1-8B-Instruct. *Data*: hierarchy fine-tuning data (*64 calibration pairs plus 250 unlabeled neutral paragraphs).

Method	Data	IHEval conflict				IHEval aligned				MMLU
		RF	TE	Saf.	Avg.	RF	TE	Saf.	Avg.	
Native	–	17.8	5.7	28.4	15.6	67.7	76.2	55.2	67.8	70.0
SFT (Mu et al., 2025)	150K	26.1	34.2	72.5	42.8	60.4	48.9	62.7	56.1	60.9
SFT + DPO (Mu et al., 2025)	150K + 30K	64.4	51.4	**85.5**	64.9	88.6	78.5	67.6	**78.3**	65.7
SFT + GRPO (Huang et al., 2025)	150K + 22.9K	54.9	59.4	60.6	58.5	80.5	**79.1**	66.6	75.9	66.7
RoCAST (ours)	0.06K*	51.0	60.6	51.3	55.2	77.5	76.7	49.6	69.2	**70.0**
SFT + DPO + RoCAST (ours)	150K + 30K + 0.06K*	**76.7**	**65.2**	85.4	**74.2**	**89.0**	77.6	**67.6**	78.0	65.7

Selection	Qwen3-8B				Llama-3.1-8B			
	Cf.	Δ	Al.	Δ	Cf.	Δ	Al.	Δ
Native	25.2		77.6		12.6		53.4	
Random heads	33.7	+8.5	**71.3**	−6.3	23.7	+11.1	50.5	−2.9
BBS only	37.9	+4.2	52.9	−18.4	43.6	+19.9	44.6	−5.9
+ RCF	50.1	+12.2	66.3	+13.4	45.7	+2.1	52.6	+8.0
+ SPS (RoCAST)	**51.8**	+1.7	68.9	+2.6	**45.7**	+0.0	**54.2**	+1.6

Table 4: Cumulative ablation at K=96 (IHEval conflict and aligned accuracy, %). Δ: change over the previous row.

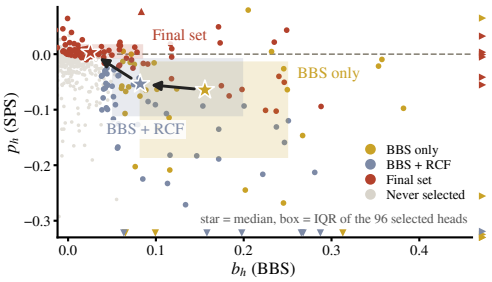

Figure 2: Head sets of Table 4 on Qwen3-8B in the (b_h, p_h) plane, colored by the last stage that selected each head.

5 Analysis

Comparison with hierarchy training. Table 3 compares RoCAST with training-based hierarchy methods on Llama-3.1-8B. RoCAST draws its calibration pairs from the verifier-supervised dataset used for GRPO by Huang et al. (2025), but uses them for head selection rather than parameter updates. With 64 accepted instruction pairs and 250 neutral paragraphs for head-level calibration, RoCAST outperforms SFT (Mu et al., 2025) in both conflict and aligned accuracy. Its conflict accuracy of 55.2 also approaches the reported SFT + GRPO score of 58.5, obtained after a GRPO stage using 22,922 instances. This comparison illustrates that a small subset of the GRPO data can support competitive conflict resolution through calibration without fine-tuning. As in Sec. 4.2, RoCAST also complements SFT + DPO. Applying the original head set increases conflict accuracy by 9.3 percentage points to 74.2, the highest score in the table, with a 0.3-point aligned decrease and unchanged MMLU accuracy.

Contribution of the selection criteria. Table 4 compares the selection stages with the steering operator, strength, and head budget held fixed. BBS improves conflict accuracy over random-head steering, but substantially reduces aligned accuracy. RCF provides the largest additional gains beyond BBS in both conditions on both models. On Qwen3-8B, it improves conflict accuracy by 12.2 points and aligned accuracy by 13.4 points, before any explicit preservation criterion is introduced. On Llama-3.1-8B, the gains are concentrated in aligned performance. SPS further improves aligned accuracy over BBS+RCF without reducing conflict accuracy on either model. Figure 2 helps interpret these gains through the distributions of conflict benefit b_h and system-only preservation p_h on Qwen3-8B. The two scores are weakly negatively correlated within the eligible set, with Spearman $\rho = -0.17$, indicating a tendency for larger conflict benefits to accompany greater preservation costs. The BBS-only set reflects this tension, with a median b_h of 0.16 but a median p_h of -0.064. RCF replaces 38 of its 96 heads and substantially improves both evaluation conditions, even though the median p_h rises only to -0.054. These gains show that role consistency provides selection information not captured by conflict benefit alone. Notably, the median b_h falls to 0.03 in the final set, yet both accuracies exceed those of BBS alone. RCF and SPS thus improve the realized conflict–preservation trade-off rather than simply maximizing isolated conflict benefits. Appendix B provides detailed head-distribution analyses across all five models.

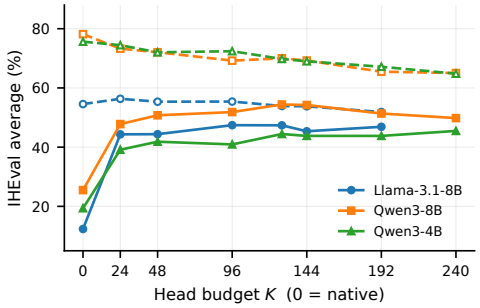

Figure 3: IHEval conflict (solid) and aligned (dashed) averages versus the head budget K.

Method	Latency ↓ (ms/token, B=1)	$B^\star$ ↑	Throughput ↑ (tokens/s, B=$B^\star$)
Native	18.69 (1.00×)	128	1196.8 (1.00×)
IHDec	33.70 (1.80×)	1	32.6 (0.03×)
V-Simple	**18.73 (1.00×)**	16	623.0 (0.52×)
RoCAST (ours)	36.96 (1.98×)	**128**	**881.3 (0.74×)**

Table 5: Inference efficiency on Llama-3.1-8B-Instruct (one H100 80 GB GPU). Latency is generation time per output token at $B = 1$. Throughput is measured at each method's largest feasible batch $B^\star$. Means over three runs. Bold marks the best non-Native result.

Effect of the head budget. Figure 3 varies K on a fixed 760-prompt IHEval subset, with $\beta = 4$ and all other components, including the head ranking, held fixed. At $K = 48$, the three models shown already obtain approximately 86–91% of their largest conflict gains over native on the reported grid. The effects of further expansion differ across models. From $K = 144$ to $K = 240$, Qwen3-4B gains 1.7 conflict points while losing 4.2 aligned points, whereas Qwen3-8B loses 4.4 conflict points alongside the same 4.2-point aligned decrease. Some aligned degradation is therefore avoidable, rather than a necessary cost of better conflict resolution: a smaller head set can retain most of the conflict benefit and sometimes outperform a larger set in both conditions. Limiting the intervention can thus protect not only aligned behavior, but also the conflict gains already obtained. Appendix C reports all five models and domain-level results.

Inference efficiency. Table 5 compares single-request latency and memory-budgeted throughput on a randomly sampled subset of 256 IHEval prompts. IHDec is evaluated at $B = 1$ because its implementation processes conversations individually. At $B = 1$, V-Simple achieves near-Native latency by editing cached values once before decoding, whereas RoCAST's eager attention and per-step reweighting incur roughly twice Native's latency. This single-request disadvantage does not directly determine batched throughput, because feasible batch size is constrained by prefill memory. V-Simple retains all layers' attention maps for input-specific attribution, whereas RoCAST reuses a fixed offline-selected head set and processes maps one layer at a time. At $B = 16$, V-Simple's largest feasible batch, RoCAST achieves lower throughput but uses only 21.5 GB of peak memory, less than half of V-Simple's 48.4 GB. This memory saving lets RoCAST scale to $B = 128$, matching Native and reaching eight times V-Simple's maximum batch size. At each method's own $B^\star$, RoCAST therefore achieves $1.41\times$ V-Simple's throughput and $27.0\times$ IHDec's throughput. Thus, RoCAST's lower memory footprint enables enough parallelism to outweigh its per-step steering cost, yielding higher throughput under the same memory budget despite slower single-request execution. Appendix D provides further profiling and matched-batch results, with RoCAST's throughput advantage over V-Simple widening to $2.47\times$ on the 256 longest IHEval prompts.

6 Conclusion

We presented RoCAST, a training-free approach to hierarchical instruction following through offline-calibrated attention steering. Its key design is to evaluate conflict benefit, content-controlled role consistency, and preservation under uncontested system instructions separately when selecting intervention heads. Experiments demonstrate substantial conflict gains, near-native aligned task execution, and transfer across thinking modes and hierarchy-trained checkpoints without recalibration. Ablations and head-budget analyses further show that effective steering depends on where it is applied, not simply on selecting heads with large conflict gains or intervening on more heads. Preservation nevertheless remains model- and domain-dependent, particularly in safety and tool use, and the current implementation adds decoding latency. These limitations motivate more domain-sensitive preservation criteria and more efficient attention implementations. Overall, RoCAST supports offline head calibration as a complement to hierarchy-specific training, improving priority resolution without updating model parameters.

AI USE STATEMENT

Generative AI tools were used to assist with the implementation and debugging of research code, the preparation and formatting of tables and figures, and editing the manuscript for clarity, readability, and phrasing. The authors reviewed and verified all AI-assisted code and manuscript content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details necessary to reproduce our results throughout the main paper and appendices. The proposed method and experimental setup are described in Sections 3–4, while Appendices A–D provide detailed calibration, inference, evaluation, baseline, and efficiency protocols.

REFERENCES

Yilin Geng, Haonan Li, Honglin Mu, Xudong Han, Timothy Baldwin, Omri Abend, Eduard Hovy, and Lea Frermann. Control illusion: The failure of instruction hierarchies in large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30816–30824, 2026.

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

Vitoria Guardieiro, Avishree Khare, Adam Stein, and Eric Wong. Instruction following by principled boosting attention of large language models. *arXiv preprint arXiv:2506.13734*, 2025.

Chuan Guo, Juan Felipe Ceron Uribe, Sicheng Zhu, Christopher A Choquette-Choo, Steph Lin, Nikhil Kandpal, Milad Nasr, Sam Toyer, Miles Wang, Yaodong Yu, et al. Ih-challenge: A training dataset to improve instruction hierarchy on frontier llms. *arXiv preprint arXiv:2603.10521*, 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021b.

Sian-Yao Huang, Li-Hsien Chang, Che-Yu Lin, and Cheng-Lin Yang. Beyond oracle: Verifier-supervision for instruction hierarchy in reasoning and instruction-tuned LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=IQ513IX1G5.

Minjae Kang and Jaehyung Kim. Enhancing instruction following of llms via activation steering with dynamic rejection. *arXiv preprint arXiv:2603.06745*, 2026.

Sanjay Kariyappa and G Edward Suh. Where instruction hierarchy breaks: Diagnosing and repairing failures in reasoning language models. *arXiv preprint arXiv:2606.07808*, 2026.

Kenneth Li, Tianle Liu, Naomi Bashkansky, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Measuring and controlling instruction (in) stability in language model dialogs. *arXiv preprint arXiv:2402.10962*, 2024.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Nicole Geumheon Liu, Haeun Jang, Yonghyun Jun, and Hwanhee Lee. Ihdec: Divergence-steered contrastive decoding for securing multi-turn instruction hierarchies. *arXiv preprint arXiv:2606.29960*, 2026.

Jiwon Moon, Yerin Hwang, and Kyomin Jung. Language shapes instruction hierarchy compliance in multilingual llms. *arXiv preprint arXiv:2607.23545*, 2026.

Norman Mu, Jonathan Lu, Michael Lavery, and David Wagner. A closer look at system prompt robustness, 2025. URL https://arxiv.org/abs/2502.12197.

Yanzhao Qin, Tao Zhang, Yanjun Shen, Wenjing Luo, Haoze Sun, Yan Zhang, Yujing Qiao, Weipeng Chen, Zenan Zhou, Wentao Zhang, et al. Sysbench: Can large language models follow system messages? *arXiv preprint arXiv:2408.10943*, 2024.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Zi-Tong Shi, Frank Wan, Haixin Wang, Ruoyan Li, Zijie Huang, Wan-Jia Zhao, Yi-Jia Xiao, Xiao Luo, Carl Yang, Yizhou Sun, and Wei Wang. Don't forget the enjoin: Focallora for instruction hierarchical alignment in large language models. *Advances in Neural Information Processing Systems 38*, 2025. URL https://api.semanticscholar.org/CorpusID:282687427.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15725–15788, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.840. URL https://aclanthology.org/2024.acl-long.840/.

Sam Toyer, Olivia Watkins, Ethan Mendes, Justin Svegliato, Luke Bailey, Tiffany Wang, Isaac Ong, Karim Elmaaroufi, Pieter Abbeel, Alan Ritter, et al. Tensor trust: Interpretable prompt injection attacks from an online game. In *International Conference on Learning Representations*, volume 2024, pp. 18714–18746, 2024.

Praveen Venkateswaran and Danish Contractor. Spotlight your instructions: Instruction-following with dynamic attention steering. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3752–3770, 2026.

Eric Wallace, Kai Xiao, Reimar Leike, Lilian Weng, Johannes Heidecke, and Alex Beutel. The instruction hierarchy: Training llms to prioritize privileged instructions. *arXiv preprint arXiv:2404.13208*, 2024.

Tong Wu, Shujian Zhang, Kaiqiang Song, Silei Xu, Sanqiang Zhao, Ravi Agrawal, Sathish Reddy Indurthi, Chong Xiang, Prateek Mittal, and Wenxuan Zhou. Instructional segment embedding: Improving llm safety with instruction hierarchy. In *International Conference on Learning Representations*, volume 2025, pp. 93728–93758, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Charles Ye, Jasmine Cui, and Dylan Hadfield-Menell. Prompt injection as role confusion. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2603.12277.

Siqi Zeng, Sewoong Lee, Han Zhao, and Julia Hockenmaier. Steering instruction hierarchies at inference time. *arXiv preprint arXiv:2607.26228*, 2026.

Qingru Zhang, Chandan Singh, Liyuan Liu, Xiaodong Liu, Bin Yu, Jianfeng Gao, and Tuo Zhao. Tell your model where to attend: Post-hoc attention steering for llms. In *International Conference on Learning Representations*, volume 2024, pp. 42411–42430, 2024.

Zhihan Zhang, Shiyang Li, Zixuan Zhang, Xin Liu, Haoming Jiang, Xianfeng Tang, Yifan Gao, Zheng Li, Haodong Wang, Zhaoxuan Tan, et al. Iheval: Evaluating language models on following the instruction hierarchy. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8374–8398, 2025.

Zishuo Zheng, Vidhisha Balachandran, Chan Young Park, Faeze Brahman, and Sachin Kumar. Reasoning up the instruction ladder for controllable language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39332–39354, 2026.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Experimental Protocol

A.1 Inference and Calibration

Inference settings. We use each model's chat template, BF16 weights, and greedy decoding. Qwen3 runs use non-thinking mode unless otherwise stated. For thinking-enabled runs, only the answer after </think> is scored. RoCAST operates during both prefill and decoding, reweighting attention to prompt message bodies while leaving headers and generated key positions unchanged.

Calibration and layer band. Head calibration uses 64 accepted instruction pairs per model from Beyond Oracle (Huang et al., 2025) and 250 neutral passages from Dolma (Soldaini et al., 2024) and C4 (Raffel et al., 2020), each truncated to 1,024 tokens. The single-head margin scan is performed in FP32 rather than BF16 to reduce rounding noise. For all models, we restrict the head-selection candidate set $\mathcal{H}_{\text{cand}}$ to layers $0, \ldots, \lfloor 0.7L \rfloor$, where L is the total number of layers. This restriction prevents the substantially larger role-response norms of later-layer heads from dominating the cross-head reference used during selection. We fix $\beta = 4$ and $\lambda = 1$. The head-budget selection procedure in Appendix A.2 selects $K = 96$ for every model. The resulting head set and intervention strength are fixed after calibration and reused across all test inputs.

A.2 Head-Budget Selection

Validation protocol. We select K using 800 held-out instruction pairs from the data released by Huang et al. (2025), after excluding the discovery pool used for head calibration. The same validation pairs are used for all models and are disjoint from the evaluation benchmarks. We evaluate the AB configuration (A_i in system, B_i in user) with greedy generation and score responses with verifier v_i^A. The eligible set $\mathcal{E}$, head ranking, $\beta = 4$, and $\lambda = 1$ remain fixed.

Selection rule. For each model, we consider the ordered grid $\{48, 96, 144, 192, 240\}$, excluding budgets with $K > |\mathcal{E}|$. Starting at $K = 48$, we test each successive increase using paired validation outcomes. For a transition $K_{\text{prev}} \to K$, let n_+ and n_- count pairs whose verifier outcomes change from incorrect to correct and from correct to incorrect, respectively. We compute the one-sided exact paired sign-test p-value

$$p_{\text{step}} = \Pr[X \geq n_+], \qquad X \sim \text{Binomial}(n_+ + n_-, 1/2). \tag{15}$$

We advance to K only if $p_{\text{step}} < 0.05$. At the first non-significant transition, we stop and select K_{prev}. If all transitions are significant, we select the largest budget in the grid.

Table 6: Held-out conflict accuracy (%) across candidate head budgets. The second row for each model reports the one-sided paired p_{step} for the transition from the preceding budget. Bold indicates the budget selected by the stopping rule. A dash denotes an infeasible budget ($K > |\mathcal{E}|$).

Model	Native	$K = 48$	$K = 96$	$K = 144$	$K = 192$	$K = 240$
Llama-3.2-3B	6.8	33.6	**36.2**	36.0	–	–
p_{step}			0.015	0.598		
Llama-3.1-8B	7.1	35.4	**39.1**	40.1	47.4	–
p_{step}			0.001	0.240	< 0.001	
Qwen3-4B	3.0	29.5	**32.8**	33.6	34.1	37.1
p_{step}			< 0.001	0.201	0.354	0.009
Qwen3-8B	3.6	28.2	**33.6**	34.9	36.8	37.6
p_{step}			< 0.001	0.110	0.038	0.250
Qwen3-14B	3.4	36.6	**38.6**	39.9	40.4	39.4
p_{step}			0.024	0.061	0.279	0.934

Table 7: IHEval domains and tasks. Task counts determine the weights in the overall task-macro average.

Domain	Tasks	Evaluation focus
Rule following (RF; 2 tasks)	Single-turn; Multi-turn	Follow system-level response constraints across conversation turns.
Task execution (TE; 3 tasks)	Verb extraction; English-to-Spanish translation; Language detection	Execute the system-specified task using user-provided text without following competing user instructions.
Safety (Saf.; 2 tasks)	User-prompt hijacking; System-prompt extraction	Enforce password-based access control and prevent secret extraction using TensorTrust (Toyer et al., 2024) attacks.
Tool use (Tool; 2 tasks)	Intrinsic Instruction; Injected Instruction	Fulfill user requests using tool outputs without obeying conflicting instructions in those outputs.

Outcome. Table 6 summarizes the held-out validation results across candidate budgets. The $48 \to 96$ transition is significant for all five models, while the $96 \to 144$ transition is not significant for any. The stopping rule therefore selects $K = 96$ for every model. Later transitions are not considered after this first non-significant step.

A.3 Benchmarks and Metrics

IHEval. We evaluate all 38 aligned and conflict settings across nine tasks, excluding reference settings (Zhang et al., 2025). Table 7 summarizes the four domains.

We follow IHEval's official evaluation protocol and use its released evaluators. Scores are first averaged across settings within each task and then across tasks, separately for aligned and conflict conditions. With each domain score defined as the mean of its constituent task scores,

$$\mathrm{Avg}_9 = \frac{2\,\mathrm{RF} + 3\,\mathrm{TE} + 2\,\mathrm{Saf} + 2\,\mathrm{Tool}}{9}, \qquad \mathrm{Avg}_7 = \frac{2\,\mathrm{RF} + 3\,\mathrm{TE} + 2\,\mathrm{Saf}}{7}. \tag{16}$$

Thus, domain weights reflect task counts rather than assigning equal weight to each domain. IHDec and the training-based comparison use Avg_7. Changes relative to Native are computed over the same seven-task scope.

MMLU. We evaluate a fixed 1,000-question subset of the MMLU test set covering all 57 subjects (Hendrycks et al., 2021a), constructed deterministically and shared across all models and methods. We use 5-shot prompting, with the five development examples from the same subject presented as prior user–assistant exchanges before the test question and no system message.

IFEval-sys. We evaluate system-level instruction following using a system-message variant of IFEval (Zhou et al., 2023; Mu et al., 2025). Across the 541 prompts, verifiable response constraints are placed in the system message, while the underlying task is provided in the user message. We report prompt-level strict accuracy, which requires all constraints in a prompt to be satisfied.

MATH-500. We evaluate all 500 problems zero-shot (Hendrycks et al., 2021b; Lightman et al., 2024). The user message contains the problem, and the system message is:

> Solve the following math problem. Reason step by step, and put your final answer within `\boxed{}`.

We report accuracy using extracted boxed answers, surface normalization, and symbolic-equivalence checks; missing boxed answers count as incorrect. Unlike MMLU, this layout exercises capability preservation under an active intervention.

A.4 Baselines and Transfer

Native and random heads. Native denotes the original model without attention steering. The random-head control applies the same RoCAST operator, intervention strength, and head budget, but selects its heads uniformly from all attention heads without the calibration-based eligibility or ranking criteria.

IHDec. IHDec (Liu et al., 2026) uses divergence between full and role-ablated predictions to guide contrastive decoding (Li et al., 2023). We use the authors' implementation with initial correction coefficient 1,300, per-token decay 0.97, and batch size 1.

V-Simple. We evaluate V-Simple, the extraction-free variant of V-Steer (Zeng et al., 2026). Unlike the default V-Steer configuration, which assumes identified competing instruction spans, V-Simple operates directly on message boundaries. We use this variant to avoid oracle span annotations or an extraction procedure not used by RoCAST or the other inference-time baselines. We use $\gamma_A = 2.5$ and $\gamma_B = 0.75$ to scale the cached values by $1 + \gamma_A$ and $1 - \gamma_B$.

VerIH. VerIH (Zheng et al., 2026) trains thinking-enabled models to resolve system–user instruction conflicts using verifiable reinforcement learning. We use the released Qwen3-8B checkpoint[1] and train Qwen3-4B following the released recipe. For both models, RoCAST reuses the corresponding native model's think-off head set and intervention strength without recalibration.

Training-based Llama comparison. For Llama-3.1-8B, we reevaluate the released RealGuardrails SFT and SFT+DPO checkpoints (Mu et al., 2025) in our evaluation setting.[2] [3] Beyond Oracle (Huang et al., 2025) initializes from the RealGuardrails SFT checkpoint and applies verifier-supervised GRPO on 22,922 verified instruction-conflict instances. RoCAST is applied to the SFT+DPO checkpoint using the original Llama-3.1-8B head set, without head reselection.

B Distribution of Selected Heads

This appendix characterizes the head sets that RoCAST deploys on the five models of Table 1, using the calibration artifacts and scores of Section 3.

Eligibility and coverage. Within the candidate set $\mathcal{H}_{\text{cand}}$, the criterion $\widehat{r}_h > 0$ retains 53–59% of the heads and $\sigma_h > 0$ retains 54–65%. Their intersection $\mathcal{E}$ contains 146–429 heads, within 7% of the size expected if the two criteria were independent. Behavioral benefit and role consistency therefore capture largely different properties of a head. The budget $K = 96$ covers 22–44% of $\mathcal{E}$

[1] https://huggingface.co/TheToad/Qwen3-8B-VerIH

[2] https://huggingface.co/normster/RealGuardrails-Llama3.1-8B-Instruct-SFT

[3] https://huggingface.co/normster/RealGuardrails-Llama3.1-8B-Instruct-SFT-DPO

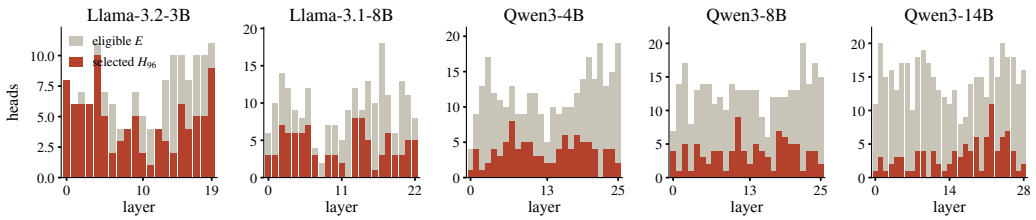

Figure 4: Eligible heads in gray and selected heads in red for each candidate layer.

Table 8: Head sets at each stage of the cumulative ablation with $K = 96$. BBS is the top-96 by b_h among heads with $\widehat{r}_h > 0$, +RCF the top-96 by b_h over $\mathcal{E}$, and +SPS the deployed set. Not in $\mathcal{E}$ counts BBS heads outside the eligible set, all of which +RCF removes. $\sum b_h$ is the total single-head benefit of the set, and $p_h < 0$ the number of heads that lower the system-only margin.

		$\sum b_h$			$p_h < 0$		
Model	Not in $\mathcal{E}$	BBS	+RCF	+SPS	BBS	+RCF	+SPS
Llama-3.2-3B	49	9.5	4.7	3.9	78	69	59
Llama-3.1-8B	46	7.2	4.6	3.5	80	80	64
Qwen3-4B	40	28.5	18.6	11.0	85	83	39
Qwen3-8B	38	24.1	16.8	10.9	81	78	41
Qwen3-14B	47	22.2	15.0	10.2	79	77	34

on four models and 66% on Llama-3.2-3B, whose eligible set is the smallest. As Figure 4 shows, the selected heads occupy nearly every candidate layer with at most 11 heads in any one layer, so selection does not reduce to a depth rule.

What each selection stage changes. Table 8 and Figure 5 show the head sets of the cumulative ablation in Table 4 for all five models. BBS only is the top-96 by b_h among heads with $\widehat{r}_h > 0$, +RCF is the top-96 by b_h over $\mathcal{E}$, and +SPS is the deployed set. The BBS-only set is dominated by heads that fail the other two criteria. Of its 96 heads, 38–49 are not in $\mathcal{E}$, and 78–85 have $p_h < 0$, so steering them lowers the system-side margin even when there is no conflict. Moving to +RCF replaces the 38–49 ineligible heads with the next eligible heads by b_h and keeps every other head. The change between these two rows of Table 4 therefore comes only from this exchange. The heads that RCF removes are not those with the lowest b_h or the lowest p_h. They spread over the whole range of both scores, so RCF adds information that neither score contains. Moving to +SPS replaces 24–54 further heads. The removed heads have mean p_h between -0.19 and -0.03, and the added heads have mean p_h at or above zero. Across the three stages, the total single-head benefit $\sum b_h$ of the set falls by 51–61% on every model, and only 26–37 of the original BBS heads remain. Yet Table 4 shows conflict accuracy rising and aligned accuracy recovering along the same path. Single-head benefits therefore do not add up. Many heads with a large benefit on their own hurt the model when steered together with others, and RCF and SPS remove two different kinds of such heads. The count of heads with $p_h < 0$ tells the two stages apart. It changes little under +RCF, from 78–85 to 69–83, even though aligned accuracy already recovers at this stage in Table 4, and it drops to 34–64 under +SPS. The random-head control draws only 21–29% of its heads from $\mathcal{E}$. Its gap to RoCAST in Table 1 therefore reflects both the eligibility filter and the ranking within $\mathcal{E}$.

Benefit versus preservation. Table 9 summarizes the deployed sets. Most eligible heads have $p_h < 0$: steering 74–81% of them lowers the system-side margin when there is no conflict. The two scores are also weakly negatively correlated, with Spearman ρ between -0.30 and -0.05. Heads with a large conflict benefit therefore tend to have a large preservation cost. SPS pushes the selected set toward heads with p_h near or above zero, the upper right of Figure 5. Among the deployed heads, the share with $p_h < 0$ is 35–43% on Qwen3 and 61–67% on Llama, and the median p_h is positive on Qwen3 and close to zero on Llama. Unselected eligible heads have a negative median p_h on every model. As a result, the deployed sets share only 42–72 of 96 heads with the top-96 by b_h alone, but 62–87 with the top-96 by p_h alone.

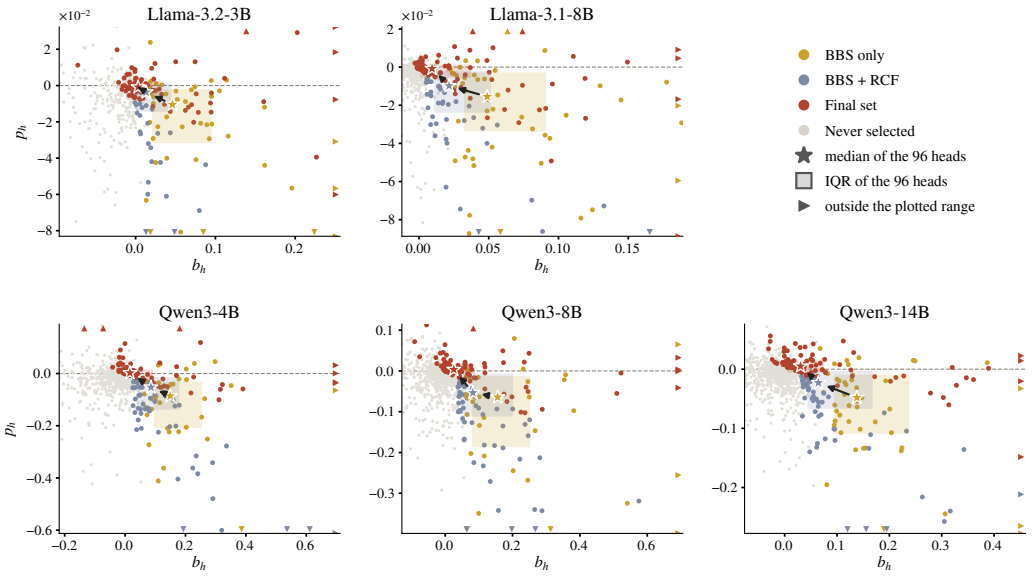

Figure 5: Head sets of the cumulative ablation in Table 4 for all five models, drawn as in Figure 2. Each head is colored by the last stage that selected it, and boxes and stars mark the IQR and median of each stage's 96 heads.

Table 9: Benefit and preservation scores. $p_h < 0$: heads whose intervention lowers the system-only margin, in the eligible set $\mathcal{E}$ and in the deployed set $\mathcal{H}_{96}$. $\rho(b, p)$: Spearman correlation of b_h and p_h over $\mathcal{E}$.

	$p_h < 0$		
Model	in $\mathcal{E}$	in $\mathcal{H}_{96}$	$\rho(b, p)$
Llama-3.2-3B	108/146	59/96	-0.05
Llama-3.1-8B	175/216	64/96	-0.21
Qwen3-4B	240/307	39/96	-0.30
Qwen3-8B	263/329	41/96	-0.17
Qwen3-14B	326/429	34/96	-0.18

C Additional Head-Budget Analysis

Protocol. We use a fixed 760-prompt IHEval subset (20 prompts per setting), shared across models and budgets. With $\beta = 4$, the eligible set $\mathcal{E}$ and the head ranking held fixed, we vary $K \in \{24, 48, 96, 144, 192, 240\}$ subject to $K \leq |\mathcal{E}|$. Figure 6 shows all five models, and Table 12 reports their aggregate and domain scores, with eligible-set sizes in the model headers.

Model-dependent budget effects. The additional models illustrate that limiting the intervention is not equivalent to minimizing K. Llama-3.2-3B improves in both conditions from $K = 24$ to $K = 48$ but declines in both from $K = 96$ to $K = 144$. Qwen3-14B shows a different non-monotonic response: both scores decline from $K = 48$ to $K = 144$ and then rise from $K = 144$ to $K = 240$. Smaller head sets can therefore avoid losses in both conditions, but the benefit of reducing K depends on the model and budget range.

Domain-level effects. Comparing $K = 48$ with each model's largest reported budget, all five models have higher conflict rule-following scores but lower conflict tool-use scores. Qwen3-4B's conflict macro-average increases over this interval despite its tool-use regression. Its aligned tool-use score falls by 32.4 points, whereas the other aligned domains change by at most 2.5 points. Thus, aggregate conflict improvements can conceal domain-specific regressions, while aligned losses can be concentrated rather than uniform across domains.

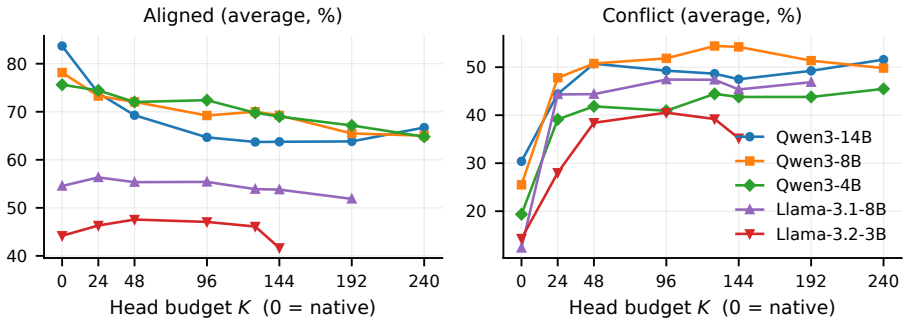

Figure 6: Head-budget sensitivity across all five models with $\beta = 4$. The left plot shows aligned performance and the right plot shows conflict performance. $K = 0$ denotes native inference.

Table 10: Matched-batch inference efficiency on Llama-3.1-8B-Instruct. Batch size is fixed to V-Simple's $B^\star$ on each IHEval subset. Throughput is generated tokens/s and memory is peak CUDA allocated memory in GB. Bold marks the best non-Native result.

	Random 256 ($B = 16$)		Longest 256 ($B = 4$)	
Method	Throughput ↑	Memory ↓	Throughput ↑	Memory ↓
Native	639.9	20.41	174.0	17.88
V-Simple	**623.0**	48.44	**166.8**	49.27
RoCAST (ours)	370.6	**21.52**	100.7	**20.91**

D Inference Efficiency Details

Setup. We profile Llama-3.1-8B-Instruct in BF16 on a single H100 80 GB GPU. Random 256 uniformly samples 256 non-reference IHEval prompts with seed 0, while Longest 256 selects the 256 longest prompts from the same pool. Native uses the default SDPA backend. RoCAST uses eager attention with the same $K = 96$ head set and intervention strength used in the main experiments. V-Simple uses eager attention for its attribution step and SDPA for subsequent decoding. IHDec is evaluated at $B = 1$ using its official implementation, which processes conversations individually.

Measurement. For the batched methods, we use greedy generation with a 128-token budget and report mean latency and throughput over three runs after warm-up. Latency is total generation time per output token, including prefill. Throughput is computed as the number of generated tokens divided by total inference time. We sweep batch sizes $B \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$ until the first out-of-memory failure. We define $B^\star$ as the largest tested batch that completes successfully. Timing excludes model loading and warm-up. Peak memory is measured as peak CUDA allocated memory under the same profiling procedure used to determine batch feasibility.

Matched-batch comparison. Table 10 fixes the batch size to V-Simple's $B^\star$ on each IHEval subset, separating execution speed from the effect of batch capacity. At the same batch size, V-Simple achieves higher throughput, while RoCAST uses less than half of its peak memory on both subsets. This is consistent with the different attribution strategies: V-Simple retains attention maps across layers for input-specific head selection, whereas RoCAST reuses its offline-selected head set and processes attention layer by layer.

Longest 256. Table 11 reports both subsets, of which Table 5 in the main text shows Random 256 only. Longer prompts reduce every method's largest feasible batch, from 128 to 64 for Native, from 16 to 4 for V-Simple, and from 128 to 32 for RoCAST. V-Simple loses the most capacity because it retains the attention maps of all layers during prefill, whose size grows with the square of the prompt length, whereas RoCAST holds only one layer's maps at a time. Single-request latency barely changes with prompt length. At $B = 1$, RoCAST remains about twice as slow as Native, $2.11\times$ on Longest 256 against $1.98\times$ on Random 256, and V-Simple remains close to Native. At

Table 11: Inference efficiency on Llama-3.1-8B-Instruct on both IHEval subsets (one H100 80 GB GPU). Latency is generation time per output token at $B = 1$ and throughput is measured at each method's largest feasible batch $B^\star$. Means over three runs. Bold marks the best non-Native result.

	Random 256			Longest 256		
Method	Latency ↓ (ms/token, $B = 1$)	$B^\star$ ↑	Throughput ↑ (tokens/s, $B = B^\star$)	Latency ↓ (ms/token, $B = 1$)	$B^\star$ ↑	Throughput ↑ (tokens/s, $B = B^\star$)
Native	18.69 (1.00×)	128	1196.8 (1.00×)	18.17 (1.00×)	64	663.0 (1.00×)
IHDec	33.70 (1.80×)	1	32.6 (0.03×)	38.72 (2.13×)	1	26.9 (0.04×)
V-Simple	**18.73 (1.00×)**	16	623.0 (0.52×)	**18.99 (1.05×)**	4	166.8 (0.25×)
RoCAST (ours)	36.96 (1.98×)	**128**	**881.3 (0.74×)**	38.32 (2.11×)	**32**	**412.2 (0.62×)**

each method's own $B^\star$, RoCAST's throughput advantage over V-Simple therefore widens from $1.41\times$ on Random 256 to $2.47\times$ on Longest 256.

Table 12: Aggregate and domain-level head-budget results (%). Model headers give the eligible-set size $|\mathcal{E}|$. RF denotes rule following, TE task execution, Saf. safety, and Tool tool use. Avg. is the nine-task macro-average rather than an unweighted mean of the four domains. $K = 0$ denotes native inference.

	IHEval conflict					IHEval aligned				
K	RF	TE	Saf.	Tool	Avg.	RF	TE	Saf.	Tool	Avg.
Qwen3-14B $\lvert\mathcal{E}\rvert = 429$										
0	37.3	56.4	13.8	1.1	30.4	84.3	85.3	87.5	76.8	83.7
24	58.9	68.3	11.2	27.2	44.4	83.8	85.9	52.5	67.3	73.9
48	73.3	72.3	18.8	27.8	50.7	88.6	85.4	35.0	60.1	69.3
96	74.1	74.9	20.0	15.3	49.3	90.2	86.3	30.0	41.4	64.7
144	72.1	75.2	18.8	10.0	47.5	93.2	87.0	30.0	33.1	63.7
192	76.4	75.1	22.5	10.0	49.2	93.8	86.2	30.0	34.1	63.8
240	74.4	75.1	26.2	18.8	51.6	92.6	86.5	30.0	47.9	66.7
Qwen3-8B $\lvert\mathcal{E}\rvert = 329$										
0	32.3	44.2	11.2	4.9	25.5	88.2	78.6	82.5	63.1	78.2
24	68.9	66.2	16.2	30.6	47.8	90.1	76.8	85.0	39.4	73.3
48	75.9	67.5	27.5	23.8	50.8	88.0	75.8	87.5	35.1	72.1
96	81.8	67.7	30.0	20.0	51.8	86.8	76.0	82.5	28.2	69.2
144	80.2	68.4	35.0	26.2	54.2	85.9	76.9	80.0	30.4	69.2
192	79.8	67.5	36.2	13.8	51.4	83.8	75.4	72.5	25.2	65.5
240	86.4	60.1	37.5	10.0	49.8	80.8	76.2	70.0	27.5	65.0
Qwen3-4B $\lvert\mathcal{E}\rvert = 307$										
0	31.5	30.3	5.0	5.1	19.4	81.9	80.0	72.5	65.8	75.6
24	62.8	52.0	13.8	21.6	39.1	90.2	79.6	72.5	52.9	74.4
48	69.2	54.1	21.2	16.6	41.8	83.7	80.4	80.0	39.9	72.0
96	64.3	55.6	22.5	13.9	40.9	86.4	81.3	82.5	35.0	72.4
144	69.2	60.2	21.2	16.4	43.8	85.0	79.4	80.0	26.4	69.0
192	70.7	60.9	23.8	11.4	43.8	87.7	77.9	80.0	17.6	67.2
240	74.8	63.2	27.5	7.5	45.5	84.5	78.1	82.5	7.5	64.8
Llama-3.1-8B $\lvert\mathcal{E}\rvert = 216$										
0	17.1	6.0	27.5	1.9	12.4	71.8	75.1	57.5	3.5	54.5
24	52.4	52.9	50.0	17.6	44.3	84.6	78.3	50.0	1.4	56.3
48	49.0	58.8	48.8	13.7	44.4	80.2	78.3	50.0	1.4	55.3
96	54.3	61.2	57.5	9.8	47.4	80.8	77.6	50.0	2.0	55.4
144	58.6	59.8	53.8	2.0	45.4	83.1	77.3	42.5	0.5	53.8
192	66.8	55.7	60.0	0.5	46.9	85.3	73.5	37.5	0.4	51.9
Llama-3.2-3B $\lvert\mathcal{E}\rvert = 146$										
0	17.0	4.8	38.8	1.3	14.3	60.1	70.9	30.0	2.3	44.2
24	45.9	26.5	38.8	1.6	28.0	69.9	71.9	27.5	3.1	46.3
48	55.4	40.0	43.8	13.8	38.4	79.0	72.0	25.0	2.0	47.6
96	54.0	55.7	41.2	3.6	40.5	78.0	71.1	22.5	4.6	47.1
144	60.1	44.2	31.2	0.7	35.2	81.7	55.4	22.5	0.1	41.7