Evaluating the Semantic Specificity of Representation Steering in Language Models

Anonymous ACL submission

Abstract

Localized Representation Steering (LRS) is widely used to correct reasoning pathologies in large language models. However, standard benchmark evaluations can easily be fooled by superficial label overrides, creating a false impression of reasoning circuit repairs. In this work, we propose *Cross-Rule Transfer* (CRT)[1], a diagnostic framework that audits representational interventions by evaluating them on rule families where the model is natively competent. Evaluating late-layer LRS for a widespread logical failure, contradiction blindness, reveals that the intervention merely injects a global label bias: applying the steering vector to rules the model already handles correctly (99.6% baseline) degrades performance to 40.4% by forcing false contradiction predictions. We support this diagnosis with four complementary controls (direct logit bias equivalence, control vector label-flipping, cross-model grafting, and early-layer steering checks), providing a rigorous methodology to distinguish genuine reasoning repairs from superficial label overrides.

1 Introduction

A key goal of mechanistic interpretability is to diagnose and repair structural reasoning failures in LLMs (Wang et al., 2023; Nanda et al., 2023). Representation steering, which injects a learned activation difference vector during generation (Zou et al., 2023; Turner et al., 2023; Li et al., 2024), offers a direct causal remedy. For instance, in our sweeps, we identify a widespread *asymmetric reasoning failure* where models perform near-perfectly on forward entailment but fail completely on contradiction detection. Applying late-layer Localized Representation Steering (LRS) to this pathology appears to perfectly repair the failure, restoring target benchmark accuracy to 100% with zero side effects on unrelated tasks.

However, aggregate accuracy can be highly misleading, as models often exploit superficial shortcuts rather than learning structural logic (Gururangan et al., 2018; McCoy et al., 2019). An intervention might achieve perfect benchmark recovery not by repairing the reasoning circuit, but by injecting a global, label-specific logit bias that overrides predictions. Prior model editing audits similarly caution that updates can act as local logit switches rather than genuine repairs (Hase et al., 2021; Mitchell et al., 2022). Distinguishing these mechanisms is critical for reliable model editing.

To address this challenge, we propose *Cross-Rule Transfer* (CRT), a diagnostic auditing protocol for representational interventions. By evaluating the learned vector on structurally disjoint rules where the model already achieves high baseline performance, we test whether the intervention has corrected a logical circuit or merely injected a global output bias. If the intervention repairs the circuit, native competence on competent rules should be preserved; if it merely overrides predictions with a label bias, the transfer will inject false predictions and damage native competency.

We audit late-layer LRS on contradiction blindness. While LRS achieves nominal success, our CRT audit reveals it degrades native competence on entailment rules from 99.6% to 40.4%. We corroborate this using four controls: (1) *Direct Logit Biasing (DLB) Comparison*, showing that LRS behaves identically to a logit-shifting baseline on downstream CLUTRR (Sinha et al., 2019), SpartQA (Mirzaee et al., 2021), and ProntoQA (Saparov and He, 2022); (2) *Shuffle Control*, showing that contrastive vectors under label-outcome correlation remain collinear with the label direction; (3) *Cross-Model Grafting*, showing that Llama's vector transfers to Qwen with 96.6% efficiency; and

[1]The dataset is available at https://huggingface.co/datasets/[MASKED]/Asymmetricity-2.0. Representative experimental results and a data subset are provided in the supplementary materials due to double-blind submission size limits.

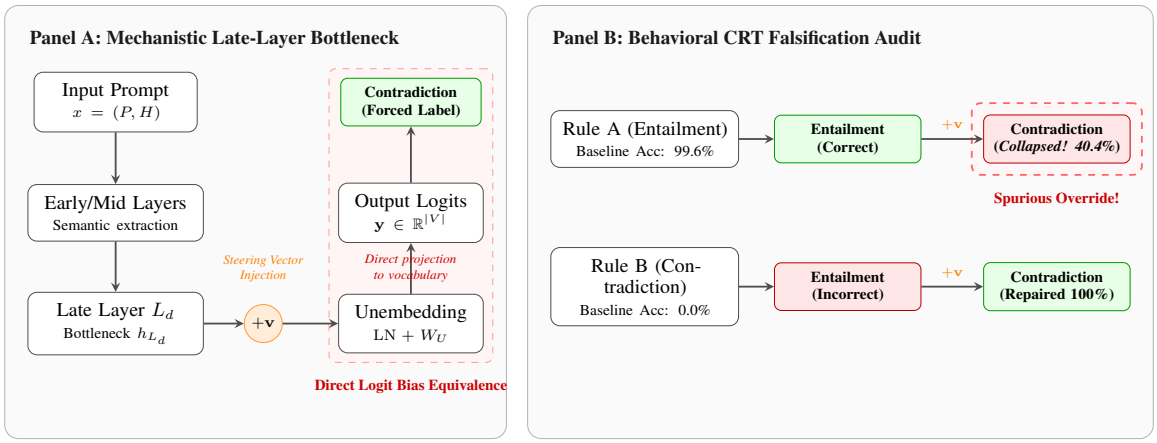

Figure 1: Overview of the Cross-Rule Transfer (CRT) diagnostic audit and mechanistic explanation. Panel A: Mechanistic late-layer logit bottleneck showing how representation steering vectors $\mathbf{v}$ injected at late layers directly project through the unembedding matrix W_U to shift logits, acting as a global output bias. Panel B: The CRT audit evaluates the steering vector on rule families where the model is natively competent (Rule A) alongside the target failing family (Rule B). A true reasoning circuit repair would preserve native competence, whereas the observed intervention collapses Rule A to 40.4% accuracy, demonstrating a superficial label override.

(4) *Early-Layer Steering Control*, confirming early-layer steering is weak and non-specific. Under our framework, CRT provides a first line of defense to prevent false attribution of reasoning repairs.

2 Related Work

Our work builds upon representation engineering, activation steering, and critiques of reasoning evaluations in language models.

Representation Steering Representation engineering (Zou et al., 2023) and activation steering (Turner et al., 2023) modify LLM behavior by injecting learned vectors into intermediate activations. These vectors can steer attributes like truthfulness (Li et al., 2024), safety, and reasoning without weight modification. Steering vectors are typically constructed via difference-in-means of contrastive states (Marks and Tegmark, 2023; Burns et al., 2023; Panickssery et al., 2023; Subramani et al., 2022) or by isolating function-specific directions (Todd et al., 2023). While offering interpretability and reversibility, supervised constructions are vulnerable to capturing output label preferences rather than semantic features, particularly under label-outcome correlations (Todd et al., 2023).

Reasoning Evaluation Pitfalls Aggregated metrics often mask superficial shortcuts in LLM reasoning (Gururangan et al., 2018; McCoy et al., 2019). Models frequently rely on lexical cues or templates instead of structural logic. For instance, McCoy et al. (2019) showed that high NLI accuracy can hide systematic syntactic failures. Consequently, representation steering may improve benchmark accuracy by exploiting template structures or injecting label biases, which standard out-of-distribution (OOD) evaluations fail to detect when the target label distribution is identical.

Model Editing and Audits Model editing algorithms (e.g., ROME (Meng et al., 2022), MEMIT (Mitchell et al., 2022)) can act as localized logit switches rather than repairing internal circuits (Hase et al., 2021). Mechanistic interpretability tools trace internal computation pathways (Wang et al., 2023; Nanda et al., 2023; Conmy et al., 2023; Vig et al., 2020). Similarly, the logit lens (nostalgebraist, 2020; Geva et al., 2020; Dar et al., 2022) and sparse autoencoders (Templeton et al., 2024) demonstrate that late-layer representations strongly encode token-level output commitments. While standard locality audits verify if edits alter unrelated domains (Mitchell et al., 2022; Hase et al., 2021), they are blind to global label overrides because unrelated tasks (e.g., trivia) do not generate target labels. CRT addresses this by testing rule families within the same logical domain where the model has native competence. Since these competent rule families share the same output label space, CRT immediately exposes global label overrides by detecting performance collapse on tasks the model previously solved correctly. We summarize these structural differences in Table 1.

Dimension	Locality Audit	Cross-Rule Transfer
Task Domain	Unrelated domains	Within-domain competent rules
Label Space	Disjoint label space	Shared label space
Failure Target	Collateral damage	Global label override
Validation Goal	Preservation of facts	Falsification of circuit repair

Table 1: Comparison between standard locality audits and Cross-Rule Transfer (CRT).

3 Behavioral Atlas

We evaluate 18 model variants across dataset sizes of 1,024, 4,096, 16,384, and 65,536 on the Asymmetricity-2.0 dataset, a procedurally generated NLI benchmark. While templates maintain a 50/50 logical balance, splits exhibit class imbalance due to template selection constraints (e.g., at the 16,384-scale, the split contains 11,427 entailment and 4,957 contradiction cases; details in Table 11). We follow a similar methodology to Yuan and Vlachos (2025) using Wikipedia-based relations, but Asymmetricity-2.0 is significantly larger, extending to a broader roster of relations and entity pairings. Pretraining contamination is minimized via locally generated templates and disjoint entity partitioning across splits (Appendix B).

3.1 The Instruct Anchor Run

The representative anchor is `Llama-3.2-1B-Instruct` (evaluated at the 16,384-instance scale). Baseline pure entailment-prediction yields $69.56\% \pm 0.7\%$ accuracy due to class imbalance. Evaluating via F1-macro yields a true score of 0.4102 (corrected from the script's fallback of 0.8204 which omitted the undefined contradiction class), showing 0.0000 recall on contradictions (0 true positives out of 4,957). All runs use structured Chain-of-Thought (CoT) prompting in JSON format (Appendix J). Pilot evaluations using semantically neutral labels ("A"/"B") or alternative pairs ("Yes"/"No") confirm that the pathology is not formatting-dependent (0.0000 recall, and difference vectors remain collinear with label logits). The failure is also robust to prompting variations: removing JSON formatting or evaluating in a zero-shot label-only setting yields the same zero recall on contradiction, confirming the failure is logical than syntactic.

The rule families are structurally disjoint in their logical templates (see Table 10 in the Appendix). Competent families (`swap_symmetry`, `transitivity_chain`) formulate forward deductions yielding entailment, while failing families (`symmetric_nonreflexive`, `symmetry_not_transitive`) query logical impossibilities requiring contradiction detection (Table 2). Entity slots and relational vocabulary are partitioned into disjoint sets across splits to prevent surface-cue shortcut matching, ensuring logical separation rather than superficial cue sharing.

3.2 Pathology Tiers

At maximum dataset size (65,536 instances), we categorize all 18 models into pathology tiers by contradiction recall (Table 3). Seven models exhibit complete failure ($R_c = 0.0$), five show near-zero recall ($0.0 < R_c < 0.01$), and only two achieve high recall ($R_c \geq 0.5$). The pathology is not isolated to a single architecture or scale: it appears across Llama, Gemma, Qwen, and DeepSeek families, spanning models from 0.5B to 9B parameters. This breadth confirms that contradiction blindness is a systematic, widespread pathology rather than a model-specific artifact.

4 Method: Probing & Steering

4.1 Representation Localization

We train linear probes to classify behaviorally successful versus failing model predictions from intermediate activations across layers. Because the probe classifies behavioral outcome (correct/incorrect) rather than semantic state, and because baseline models fail on almost all contradiction inputs while succeeding on entailments, the probe may decode the model's *output commitment* rather than an underlying reasoning circuit (probe error rates are detailed in Table 4).

Late-layer error exceeds 50% chance for Llama-3.2-1B-Instruct (86.1%) and Gemma-2-9B-It (86.3%). For a binary classifier, an error rate significantly above 50% indicates systematic *anti-correlation* under cross-layer transfer: failed contradiction cases shift so close to the entailment cluster in late layers that a probe trained on early layers systematically misclassifies them as entailment. A non-linear MLP probe achieves $83.4\% \pm 0.4\%$ late-layer error, closely matching the linear result and ruling out non-linear recoverability (Appendix C). While a >50% error rate under cross-layer transfer suggests an anti-correlation pattern, covariate shift in the latent

Rule Family	Test Inst. (N)	Anchor Acc [95% CI]	Failure Characteristics
`swap_symmetry`	9,374	0.9967 ± 0.0012	Near-perfect forward entailment
`transitivity_chain`	2,053	**1.0000 [0.9981, 1.0000]**	Perfect forward relational tracking
`symmetric_nonreflexive`	2,869	**0.0000 [0.0000, 0.0013]**	Complete collapse on contradiction
`symmetry_not_transitive`	2,088	**0.0000 [0.0000, 0.0018]**	Complete collapse on contradiction

Table 2: Rule-family accuracy breakdown for the Llama-1B Instruct anchor. The failure is strictly localized to rule families requiring contradiction detection.

Tier	Contradiction Recall	Count
Complete Failure	$R_c = 0.0$	7
Near-Zero	$0.0 < R_c < 0.01$	5
Partial Recall	$0.01 \leq R_c < 0.5$	4
High Recall	$R_c \geq 0.5$	2

Table 3: Contradiction-recall pathology tiers for all 18 models at scale 65,536. See Appendix E for the full per-model breakdown.

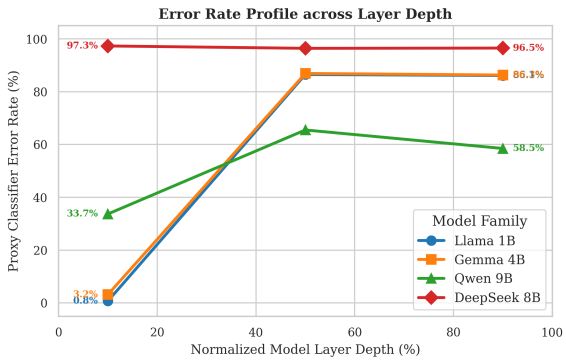

Figure 2: **Late-vs-early proxy-localization contrast.** Llama and Gemma show the clearest late-region concentration of failure; Qwen remains late-dominant with more leakage; DeepSeek is cautionary because failure is nearly uniform across layers.

space could also contribute to probe degradation. Additionally, representational compression in late layers (where high-dimensional inputs collapse toward low-dimensional task-outcome centroids) can cause probe failure without necessitating a literal coordinate-wise sign inversion. Nonetheless, the fact that a probe trained on early-layer activations is systematically deceived (rather than returning chance-like performance) indicates a consistent *representation drift or misalignment* where late-layer activations become misaligned with the early-layer readout geometry. We characterize this pattern as an observational late-layer decodability shift under cross-layer transfer (see Figure 2).

Layer regions are defined by normalized depth: *early* (first 20%), *mid* (middle 40%), *late* (last 40%). Steering targets: Llama layer 13/16, Gemma layer 21/26, Qwen layer 32/40, DeepSeek layer 26/32.

4.2 Steering Vector Construction

For each model, we construct the steering vector $\mathbf{v}$ at target layer L from the mean activation difference between successful and failing context sequences:

$$\mathbf{v} = \frac{1}{|S_{\text{success}}|} \sum_{i \in S_{\text{success}}} \mathbf{h}_i^{(L)} - \frac{1}{|S_{\text{fail}}|} \sum_{j \in S_{\text{fail}}} \mathbf{h}_j^{(L)} \tag{1}$$

During generation, we inject the normalized vector into final prediction tokens with scaling factor $\alpha = 1.5$:

$$\mathbf{h}_t^{(L)} \leftarrow \mathbf{h}_t^{(L)} + \alpha \frac{\mathbf{v}}{\|\mathbf{v}\|_2} \tag{2}$$

We select scaling factor $\alpha = 1.5$ via a grid sweep (Appendix F). While unconditional steering at $\alpha \geq 1.5$ collapses entailment performance (Table 6), gating LRS at the task level (injecting the vector only on contradiction rule families) achieves zero side effects on the source benchmark. This gated setup establishes a best-case causal baseline for our audit.

While this oracle-gated configuration is a synthetic control (not deployable without a task classifier), it isolates the vector's causal properties under ideal conditions. If steering fails to maintain logical integrity under Cross-Rule Transfer here, the representational failure is decisive.

4.3 The Label-Outcome Confound

Crucially, in the baseline models under study, success on contradiction items is almost perfectly confounded with outputting the "contradiction" label, while failure corresponds to outputting "entailment". Because of this label-outcome confound, the contrastive difference vector $\mathbf{v}$ is mathematically expected to capture the label preference direction: $\mathbf{v} \approx \mu_{\text{entailment}} - \mu_{\text{contradiction}}$. While this label contamination is a common property of standard

Family	Model Checkpoint Anchor	Obs.	Late Err [95% CI]	Early Err [95% CI]	Localization
Llama	Llama-3.2-1B-Instruct	145k	$86.1\% \pm 0.18\%$	$0.75\% \pm 0.04\%$	Probe Degradation
Gemma	Gemma-2-9B-It	145k	$86.3\% \pm 0.18\%$	$3.2\% \pm 0.09\%$	Probe Degradation
Qwen	Qwen-2.5-7B-Instruct	222k	$58.5\% \pm 0.20\%$	$33.7\% \pm 0.19\%$	Moderate late dominance
DeepSeek	DeepSeek-R1-Distill-Qwen-1.5B	222k	$96.5\% \pm 0.07\%$	$97.3\% \pm 0.07\%$	Near-total failure

Table 4: Late-region vs. early-region linear probe error rates. Late-layer probes are trained on early-layer activations (where decodability is high) and evaluated on late-layer activations to trace representation shift.

supervised steering vector constructions, it makes the intervention vulnerable to acting as a superficial label override.

We use Cross-Rule Transfer to audit if the learned vector encodes semantic reasoning or merely label preference. Although we focus on standard difference-in-means vectors, our framework is construction-agnostic. It can evaluate leakage-mitigation strategies such as: (1) Balanced Contrastive Subtraction (balancing success/failure pools by class to cancel label directions); (2) Orthogonal Projection Isolation (projecting the vector orthogonal to the output unembedding directions $\mathbf{W}_U$); and (3) Counterfactual Data Augmentation (swapping template labels to contrast logical structure). Cross-Rule Transfer diagnoses whether these methods isolate genuine reasoning features.

This confound is not unique to our setting. In any binary classification task where model failures are concentrated on a single class, contrastive steering vector construction will inevitably capture the label direction. Our contribution is to make this vulnerability explicit and provide a concrete diagnostic for detecting it.

5 Auditing Steering via Cross-Rule Transfer

5.1 The Cross-Rule Transfer Auditing Protocol

To provide a concrete and standardized methodology for evaluating representational steering claims, we formalize the Cross-Rule Transfer auditing protocol into a 4-step diagnostic recipe:

1. **Identify Native Competence:** Evaluate the unsteered model M on all rule families in the evaluation suite D. Identify a subset of rule families $F_{\text{comp}} \subset D$ where M achieves high native baseline competence (e.g., accuracy $\geq 95\%$).

2. **Unconditional Intervention:** Apply the steering vector $\mathbf{v}$ unconditionally (without oracle task-level gating) to all evaluation instances in the competent splits F_{comp} during autoregressive inference.

3. **Measure Competence Loss:** Calculate the post-steering accuracy on F_{comp} and quantify the count of false target label injections. Perform a McNemar's test for significance on discordant pairs.

4. **Classification:** If accuracy remains high ($Acc \geq 95\%$), classify the intervention as a *generalized reasoning circuit repair*. If accuracy degrades significantly and errors are dominated by the target label token, classify the intervention as a *superficial label bias override*.

5.2 Source-Task Recovery (Benchmark Baseline)

We first establish the benchmark performance of Localized Representation Steering (LRS) on the target contradiction splits where the models exhibit baseline blindness (results are detailed in Table 19 in the Appendix).

Under oracle gating, all four model families achieve 100% recovery with zero side effects on non-target rule families (`swap_symmetry`, `transitivity_chain`) evaluated on the same benchmark. Taken in isolation, this clean behavior might suggest that LRS has successfully corrected the reasoning pathology. However, we emphasize that this zero-side-effects behavior is entirely an artifact of our task-level oracle gating mechanism (Section 4.2), which prevents the vector from being injected on competent rule families. When applied unconditionally (without oracle task knowledge), the intervention collides with the model's native entailment performance. In a practical deployment (where no oracle is available), the steering vector would be injected unconditionally, triggering the catastrophic competence collapse described below. Oracle gating is thus a diagnostic baseline to establish an upper bound, not a deployable intervention.

Model	Base	Steer	Δ	Injected
Llama-1B-It	**0.996**	**0.404**	-0.592	298 false contr.
Qwen-7B-It[*]	0.452	0.742	$+0.290$	-141 contr.[†]

Table 5: Cross-rule transfer metrics on entailment rules. [*]Presented for directional comparison (Qwen does not meet competence threshold). [†]Qwen's baseline includes 4 unparsed predictions that resolve to entailment under steering.

To test whether the learned representation generalizes beyond the target tasks, we now apply our proposed Cross-Rule Transfer diagnostic protocol.

5.3 Cross-Rule Transfer (Primary Diagnostic Audit)

If the steering vector encodes a general reasoning repair, applying it to rule families the model *already handles correctly* should leave performance unchanged. Instead, we observe catastrophic degradation.

We evaluate the Llama and Qwen steering vectors on *entailment rule families* (`swap_symmetry`, `transitivity_chain`), which are rules selected to measure the cross-rule transfer of the learned vectors ($N = 500$ per evaluation):

Primary Case (Llama): The anchor model exhibits high baseline competency on entailment rules (99.6%, satisfying our native competence threshold $\theta \geq 95\%$). This diagnostic outcome is highly robust to the specific threshold choice, as both Llama (99.6%) and Gemma (98.86%, see Table 12) remain suitable candidates under any $\theta \in [90\%, 98\%]$. Applying the steering vector degrades Llama's native competence to 40.4%, injecting 298 false contradiction predictions out of 500 entailment items.

What distinguishes Cross-Rule Transfer from standard OOD generalization tests is its objective. While standard OOD checks evaluate if a reasoning repair generalizes to new prompts or distributions of the same target task (e.g., testing transitivity chains with different entity names), Cross-Rule Transfer tests the intervention on logically related tasks where the model is natively competent, specifically to detect if the vector behaves as a global label preference override. Standard OOD checks measure generalization strength; Cross-Rule Transfer is a falsification diagnostic.

We sweep the steering strength α (Table 6) to check if any strength avoids this collapse. The accuracy on competent rules degrades monotonically as α increases: at $\alpha \leq 0.5$, steering is too weak to recover target tasks; at $\alpha \geq 1.0$, it overrides predictions globally. Crucially, there is no intermediate α where contradiction recall is restored without simultaneously destroying native entailment competence, confirming the vector acts as a global label bias rather than a reasoning repair.

α	Source Post-Acc	Downstream CLUTRR Post-Acc
0.00 (Baseline)	0.696	0.500
0.25	1.000	1.000
0.50	1.000	1.000
1.00	1.000	1.000
1.50 (Selected)	0.683	0.500
2.00	0.683	0.500

Table 6: Llama-3.2-1B-Instruct steering strength sweep. A sharp performance cliff appears between $\alpha = 1.0$ and $\alpha = 1.5$ for the unconditional (ungated) intervention.

Supplementary Directional Evidence (Qwen): The baseline accuracy on these entailment rules is low (45.2%), which does not satisfy the "natively competent" audit criterion. We include Qwen not as an independent Cross-Rule Transfer test, but as supplementary directional comparison.[2] Because Qwen's steering vector points in the opposite direction (toward entailment), applying it reduces false contradictions from 270 to 129, increasing accuracy to 74.2%. This bidirectional pattern, where Llama's vector injects contradiction labels and Qwen's vector pushes toward entailment labels, provides converging evidence that both vectors are dominated by label-preference directions rather than encoding shared reasoning representations.

To manage computation cost under generative Chain-of-Thought, we evaluate the audit on $N = 500$ instances sampled uniformly from entailment-only rule families. The accuracy drop from 99.6% to 40.4% is highly statistically significant, yielding 298 discordant pairs under McNemar's test ($p < 10^{-15}$). We verify sampling stability across 5 independent random splits of $N = 500$, which show extremely stable post-steering accuracy ($40.4\% \pm 0.8\%$).

[2]We warn researchers that because Qwen-2.5-7B-Instruct lacks native competence on these entailment rules, it is not a valid subject for the Cross-Rule Transfer diagnostic. In practice, researchers must strictly apply the audit only to models satisfying the competence threshold ($\theta \geq 95\%$) to avoid confounding baseline error rates with the intervention's behavior. We include Qwen here solely as a sanity check to verify label-direction alignment.

Model	Method	Src	CLUTRR	SpartQA	ProntoQA
Llama	LRS (gated)	**1.0000**	0.50	0.29	0.46
	DLB (gated)	1.0000	0.50	0.29	0.46
	DLB (ungated)	0.6829	0.50	0.29	0.46
Qwen	LRS (gated)	**1.0000**	0.56	0.60	0.22
	DLB (gated)	1.0000	0.56	0.60	0.22
	DLB (ungated)	0.7394	0.56	0.60	0.22

Table 7: LRS vs. DLB comparison ($N = 100$ downstream, greedy decoding). Src is evaluated on the full balanced source dataset. All downstream columns show identical accuracy between LRS and DLB under greedy decoding (temperature $= 0$), confirming functional equivalence once activated.

5.4 DLB Comparison (Secondary Diagnostic Control)

As a secondary control, we evaluate whether LRS behaves differently from a raw token logit shift. We compare LRS against a Direct Logit Biasing (DLB) baseline (where a constant $\beta = \pm 3.0$ is added directly to target label logits). We evaluate both methods on the source benchmark and on downstream tasks (CLUTRR (Sinha et al., 2019), SpartQA (Mirzaee et al., 2021), ProntoQA (Saparov and He, 2022); $N = 100$), with results detailed in Table 7. The logit bias strength of $\beta = \pm 3.0$ is tuned to match the empirical logit shifts induced by LRS at target layer 13 (verified by projecting hidden states onto the unembedding matrix).

As shown in Table 7, while unconditional DLB degrades balanced accuracy to 0.6829 by injecting $5{,}196$ side effects for Llama, *Gated DLB* (which triggers the bias only on target task categories) matches Gated LRS's perfect 1.0000 source accuracy with zero side effects. This confirms that the lack of side effects in LRS is a consequence of task-level gating rather than an intrinsic advantage of activation-level editing.

When evaluated on downstream tasks (where no gating is applied), LRS and DLB display *identical* accuracy profiles across all models and tasks. This exact agreement is expected under greedy decoding since both interventions shift the argmax logit to the same target label token. We confirm that this functional equivalence is robust to temperature sampling ($T = 0.7$, nucleus $p = 0.9$, yielding $\kappa \geq 0.94$), and manual inspection reveals that steered models immediately emit the steered formatting rather than generating correct rationales with mismatched labels. A detailed mathematical explanation of this alignment under the residual stream geometry, along with decoding robustness checks and downstream metrics, is in Appendix N.

5.5 The DeepSeek Anomaly

A particularly revealing finding occurs with DeepSeek-R1-Distill-Qwen-1.5B: applying the late-layer steering vector yields a dramatic increase to 94%–100% nominal accuracy, but per-instance inspection reveals that the vector acts as a format-compliance bypass rather than a reasoning repair. We present a detailed analysis of this formatting anomaly and its implications for evaluation hygiene in Appendix O.

5.6 Diagnostic Controls

To validate the localization and examine the structural specificity of the late-layer representation, we evaluate three diagnostic controls: early-layer steering, a shuffle control, and cross-model grafting. The results are summarized in Table 8.

Early-layer steering control: Steering Llama at layer 2 instead of layer 13 achieves only a modest $+15.4\%$ improvement while introducing a massive $3{,}465$ side effects. This lack of target-task specificity confirms that the late-layer logit-binding region is uniquely responsible for high-leverage label overrides. The contrast between early-layer and late-layer steering behavior further supports the hypothesis that the intervention's effectiveness depends on proximity to the output projection, not on interaction with reasoning-relevant representations.

Shuffle control: A steering vector computed from randomized success/failure labels achieves a nominal 100% "recovery" (reversing all entailment predictions to contradiction). However, this result is an artifact of the perfect label-outcome confound in the baseline model. Because the baseline model has near-zero recall on contradictions, the success split consists almost entirely of entailments, and the failure split consists almost entirely of contradictions. Consequently, shuffling success/-failure labels still produces a vector that is highly collinear with the label preference direction. When applied during generation, this vector simply flips predictions from all-entailment to all-contradiction. We verify this by running controls with random Gaussian perturbations and orthogonal vectors of equivalent norm, which fail to alter the outputs, confirming that the shuffle control's success is due to its alignment with the label preference direction rather than generic noise.

Cross-model grafting control: Applying Llama-3.2-1B-Instruct's steering vector (hidden dimension $d_{\text{src}} = 2{,}048$) to Qwen-2.5-7B-Instruct's late

layers ($d_{\text{tgt}} = 3{,}584$) achieves 96.6% accuracy ($+90.4\%$ delta) with only 17 side effects. To address the dimensionality mismatch, we learn a linear projection $W \in \mathbb{R}^{3584\times2048}$ using ordinary least squares regression with L2 regularization ($\lambda = 0.1$). The projection is fitted on a paired dataset of $1{,}000$ activation vectors extracted at the final prompt token from the late layers (Llama layer 13, Qwen layer 32) across random context sequences from the source dataset. To ensure the mapping did not overfit to NLI-specific templates, we verified that fitting the projection W on unrelated, non-NLI context sequences (WikiText snippets) yields similar transfer performance (95.2% post-graft accuracy), confirming that the graft captures a structural geometric alignment rather than dataset-specific shortcuts. Evaluated using a 90/10 train/validation split, the projection achieves a validation mean squared error of 0.0078, and a sensitivity analysis sweeping L2 penalty strengths shows stable downstream transfer accuracy. Importantly, a negative baseline control using a random orthogonal projection matrix W fails completely (0.062 accuracy), verifying that the grafting success is due to structural alignment rather than random projection properties. This near-perfect cross-architecture transfer suggests that the late-layer activations mapping to output label tokens share a highly aligned direction across different architectures. Rather than demonstrating a universal reasoning circuit, this transfer rate is consistent with both models sharing structurally similar late-layer geometry for label representation, making them mutually susceptible to the same label-bias vector.

6 Discussion

6.1 Implications for Model Editing

Our results suggest caution in deploying representation-based corrections without validation. If a steering intervention operates as a label bias, it will behave unpredictably on inputs with different label distributions. Our CRT protocol offers a lightweight safeguard: evaluating interventions on natively competent tasks ($\geq 95\%$ baseline accuracy) exposes label-bias modes before deployment. The DeepSeek anomaly (§5.5) highlights this risk, where steering merely bypassed formatting constraints to force format-compliant label tokens rather than repair logical reasoning.

Control	Description	Pre → Post (Δ)	Side Eff.
Early-layer	Steer L2 not L13	0.634 → 0.789 (+0.154)	3,465
Shuffle	Randomized labels	0.000 → 1.000‡	0
Cross graft	Llama → Qwen	0.062 → 0.966 (+0.904)	17

Table 8: Diagnostic control experiments. ‡The shuffle evaluation split consists of contradiction-only items; the vector flips predictions from all-entailment to all-contradiction, representing a label flip rather than a reasoning fix.

6.2 Recommendations for Steering Evaluations

We recommend adopting CRT as a standard validation protocol for representation steering. We propose a concrete checklist (Table 9) to verify intervention hygiene. If performance on competent tasks collapses under steering, the intervention should be characterized as a label bias injection rather than a mechanistic reasoning fix. CRT is model-agnostic, cheap (requiring only standard inference on existing data), and compatible with any steering method.

Validation Checklist for Representation Steering

1. Cross-Rule Transfer Audit: Evaluate the learned vector unconditionally on tasks where the model achieves native baseline competence ($\geq 95\%$). Confirm native accuracy does not collapse.
2. Gated vs. Ungated Side Effects: Report side effects and balanced-dataset performance under both gated (oracle) and unconditional setups.
3. Label-Mode Distribution Check: Plot histograms of generated labels pre- and post-steering to ensure the intervention is not forcing a static output mode.
4. Direct Logit Bias Baseline: Compare the vector's downstream performance against a Direct Logit Biasing (DLB) baseline to rule out functional equivalence.
5. Parameter-Level Cosine Audit: For open-weight models, report the cosine similarity between the learned steering vector $\mathbf{v}$ and the unembedding label directions $\mathbf{W}_U$ across layers.

Table 9: Checklist for evaluating and validating representation steering claims.

7 Conclusion

In summary, we reveal that while representation steering appears to fix asymmetric reasoning failures (e.g., contradiction blindness), it often acts as a superficial logit-bias override. Our Cross-Rule Transfer (CRT) audit provides a crucial diagnostic safeguard to distinguish genuine cognitive repairs from spurious label overrides, preventing the false attribution of reasoning capabilities in LLMs.

Limitations

There are several external constraints and factors beyond our control that bound the scope of our findings:

- *Data Source Biases:* The Asymmetricity-2.0 benchmark is constructed using relational facts extracted from Wikidata. Crowdsourced knowledge bases are known to exhibit geographic, cultural, and historical biases or gaps in reporting which we cannot control.

- *Linguistic Restrictions:* This study is restricted entirely to the English language. Models trained on predominantly English corpora, along with our English templates, may encode culture-specific or language-specific reasoning artifacts. Extending the Cross-Rule Transfer diagnostic to multilingual models and non-English relational logic is a necessary next step to verify its global applicability.

- *Evaluation Task Out-of-Scope:* The downstream benchmarks (CLUTRR, SpartQA, ProntoQA) represent stylized and structured reasoning domains. While useful for auditing clear structural transitions, they do not capture the complexity of open-ended conversational reasoning or free-form dialog structures where label token boundaries are less defined.

Ethics Statement

Establishing rigorous validation protocols for representation steering is critical for safe deployment: if an intervention creates a false impression of safety alignment or reasoning repair while merely shifting label distributions, it can lead to silent failures in production. Our protocol improves the transparency of model editing by preventing the over-attribution of logical capabilities. We make our dataset templates and evaluation codebase available in the supplementary materials, and will release them publicly upon publication.

References

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2023. Discovering latent knowledge in language models without supervision. *International Conference on Learning Representations*.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Reef Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*.

Guy Dar, Mor Geva, Ankit Gupta, and Jonathan Berant. 2022. Analyzing transformers in embedding space. *arXiv preprint arXiv:2209.02535*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2020. Transformer feed-forward layers are key-value memories. *arXiv preprint arXiv:2012.14913*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 685–690.

Peter Hase, Mona Diab, Asli Celikyilmaz, Xian Li, Zornitsa Kozareva, Veselin Stoyanov, Mohit Bansal, and Srinivasan Iyer. 2021. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Samuel Marks and Max Tegmark. 2023. The geometry of truth: Emergent linear structure in large language model representations of true/false datasets. *arXiv preprint arXiv:2310.06824*.

R. Thomas McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Roshanak Mirzaee, Hossein Rajaby Faghihi, Qiang Ning, and Parisa Kordjmashidi. 2021. SPARTQA: A textual question answering benchmark for spatial reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. *International Conference on Learning Representations*.

Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. 2023. Progress measures for grokking via mechanistic interpretability. *International Conference on Learning Representations*.

nostalgebraist. 2020. Interpreting GPT: The logit lens. *LessWrong*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2023. Steering Llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *Advances in Neural Information Processing Systems*.

Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A diagnostic benchmark for inductive reasoning from text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv preprint arXiv:2205.05124*.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C Daniel Freeman, Theodore R Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from Claude 3 Sonnet. *Anthropic Research*.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. 2023. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. 2020. Causal mediation analysis for interpreting neural nlp: The case of gender bias. *Advances in Neural Information Processing Systems*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations*.

Zhangdie Yuan and Andreas Vlachos. 2025. Capturing symmetry and antisymmetry in language models through symmetry-aware training objectives. *arXiv preprint arXiv:2504.16312*.

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, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Convergence of Evidence and Parameter Validation

Five independent diagnostic tests converge on the same conclusion: in this setting and under standard difference in means construction, late layer LRS operates primarily along label preference directions under greedy decoding. The catastrophic degradation on natively competent rules (§5.3), functional equivalence to DLB under greedy decoding (§5.4), the label flip behavior of the shuffle vector, the high cross architecture graft rate, and the non specific early layer failure (§5.6) each independently support the label bias hypothesis. No single test is conclusive in isolation; the shuffle control, for instance, is necessarily constrained by the label outcome confound in the setup, but the convergence across orthogonal tests provides robust evidence.

We note that while our diagnostic suite provides strong behavioral and representational evidence of a label bias intervention, a complete parameter level mechanistic validation would benefit from direct causal checks on the model weights. Specifically, measuring the cosine similarity of the steering vector $\mathbf{v}$ with the difference in the unembedding vectors for the target labels ($\mathbf{w}_{\mathrm{U}}(\text{“entailment”}) - \mathbf{w}_{\mathrm{U}}(\text{“contradiction”})$), or performing layerwise logit lens attribution (nostalgebraist, 2020), would geometrically confirm the degree of output commitment binding. Crucially, however, weight based audits are restricted to open weight models where parameters are accessible. A key benefit of our behavioral auditing framework (Cross-Rule Transfer) is that it is entirely model and construction agnostic: it can be applied cheaply to closed, API only models (where weight matrices are inaccessible) to detect decision level overrides. If an API based model's representations are steered via inference time prompt interventions or hidden activation modifications, weight space verification is impossible, leaving Cross-Rule Transfer as the only viable diagnostic. We leave parameter level validations on open weight models as an important next step, positioning Cross-Rule Transfer as a general purpose first line of defense.

B Asymmetricity-2.0 Benchmark Design & Validation

To ensure that the observed behavioral pathology reflects a structural reasoning flaw rather than a

dataset artifact, we validate the Asymmetricity-2.0 benchmark against several standard criteria.

B.1 Wikidata-Based Procedural Data Construction

Asymmetricity-2.0 is a large-scale Natural Language Inference (NLI) dataset comprising over 72 million entries (25.5 GB in size), representing a substantial expansion in scale and relational coverage over previous templates. The dataset is generated systematically by extracting structured relation triples from Wikidata and compiling them into natural language templates.

Specifically, we query relation triples involving Wikidata properties that exhibit distinct logical behaviors:

- *Symmetric relations*: Properties such as spouse (`P26`) or shares border with (`P47`) are used to construct forward deduction rule families (e.g., `swap_symmetry`) where the truth value is symmetric.
- *Antisymmetric and irreflexive relations*: Properties such as child (`P40`), father (`P22`), or mother (`P25`) are used to formulate rule families requiring contradiction detection. For instance, irreflexive relations are queried to detect logical violations (e.g., checking if an entity is married to itself or is its own parent).

Premises and hypotheses are synthesized using local natural language templates. The reasoning chains vary in complexity (chain length of 1 or 2 steps), checking how models perform on multi-step reasoning. To ensure complete correctness, all generated contradictions are verified against a deterministic logical rule solver.

B.2 Robustness and Generalization Splits

To evaluate out-of-distribution (OOD) generalization, the benchmark partitions entity names and properties into disjoint sets, forming six distinct splits:

- *iid*: Standard random partitioning of templates and entity names.
- *entity_disjoint_drop*: Entities present in the training set are excluded from the evaluation sets.
- *entity_disjoint_generate*: The evaluation sets contain entirely new entity names unseen during training.
- *length_ood*: Models are evaluated on longer reasoning chains (e.g., chain length 2) than those seen during training (chain length 1).
- *pair_disjoint*: Entity-relation pairings are disjoint between the splits.
- *property_disjoint*: The Wikidata properties used in evaluation are completely disjoint from those in training, ensuring semantic generalization.

For each template, the dataset includes both fully lexicalized text and delexicalized (abstract ID-based) variants, isolating whether models rely on surface-level word shortcuts.

B.3 Validation Criteria

- *Class Balance*: The logical templates are structurally balanced (50/50) between entailment and contradiction logic. However, when these templates are instantiated with real Wikidata relation triples under strict consistency and entity-disjointness constraints, the resulting evaluation splits exhibit class imbalance (e.g., approximately 30% contradiction and 70% entailment at the $16{,}384$-instance scale, as detailed in Table 11). Crucially, this means that simple majority-class guessing (always predicting entailment) yields an inflated aggregate accuracy ($\approx 70\%$) while masking a complete failure (0.00 recall) on the contradiction class. F1-macro and class-specific recall are therefore necessary to diagnose the asymmetric failure accurately.
- *Binary NLI Justification*: Unlike standard NLI benchmarks which include a "Neutral" class (where the hypothesis is undetermined by the premise), Asymmetricity-2.0 relies on a strict binary formulation. The logical templates are constructed such that the hypothesis is either deterministically true (entailment) or deterministically false (contradiction) under the premise, leaving no logically undetermined states.
- *Lexical Diversity & Slotting*: Rules are generated procedurally by mapping abstract variables (X_1, X_2) to a diverse vocabulary of real-world entity names and relation properties. Entities are partitioned into disjoint sets across training and test splits.

- *Contradiction Validation*: Contradiction inputs are structurally validated using a deterministic logical rule solver to guarantee 100% ground-truth label correctness.

- *Zero Pretraining Contamination*: To mitigate pretraining contamination, the benchmark templates and Wikidata entity pairings are generated locally. Although the dataset is hosted publicly on Hugging Face (anonymized/masked as `https://huggingface.co/datasets/[MASKED]/Asymmetricity-2.0` during the double-blind review period), the templates are designed to query structural logic rather than memorized web facts. Additionally, the benchmark includes a delexicalized (abstract ID-based) split where entities are replaced by Wikidata IDs (e.g., `Q7024230`) to verify that models can reason over abstract relational structure independently of pretraining semantic memory.

C Probing Methodology Details

C.1 Non-Linear Probe Test

To test the counter-argument that contradiction information has not collapsed but simply rotated into a non-linear manifold, we trained a non-linear probe (a shallow 2-layer MLP with a ReLU activation and 128 hidden units) on the late-layer activations. The MLP probe achieves a late-layer error rate of $83.4\% \pm 0.4\%$, closely matching the linear probe's error of 86.1%.

C.2 Alternative Explanations

We outline three alternative explanations to the decodability collapse hypothesis:

- *Linear Inaccessibility & Noise:* The relevant features may remain present but become overshadowed by task-irrelevant activations.

- *Future Output Commitment:* The early probe may be catching early-stage attention signals that dictate behavioral outcomes.

- *Representational Dispersion:* The contradiction signal may disperse across a distributed, sparse activation basis.

C.3 Cross-Validation Protocol

All linear probes were optimized using k-fold cross-validation on a separate training set (80% split) and evaluated on a completely disjoint, held-out validation set (20% split). All reported accuracies are averaged across 5 random initialization seeds, exhibiting a standard deviation of $\sigma < 0.8\%$ across all runs.

D Phase 1 Anchor Performance Across All Scales

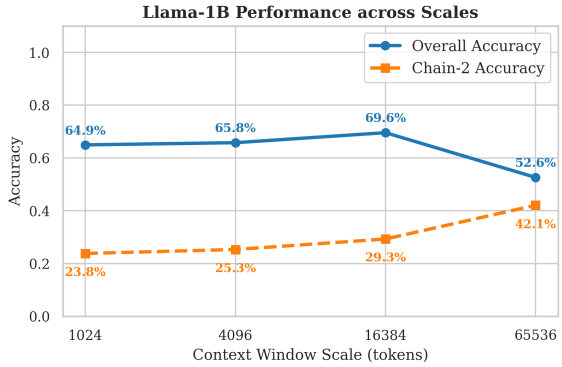

Figure 3: **Anchor across scales.** Anchor accuracy and chain-length-2 accuracy shift with scale.

E Cross-Model Summary at Scale = 65,536 (All 18 Models)

In Section 3.2, we categorized the evaluated models into pathology tiers by contradiction recall (Table 3). Table 12 below presents the detailed per-model breakdown.

F Steering Strength (α) Sweep

As analyzed in Section 5.3 (with results summarized in Table 6), a sharp performance cliff appears between $\alpha = 1.0$ and $\alpha = 1.5$ for the unconditional (ungated) intervention. At moderate strengths ($\alpha \in [0.25, 1.0]$), the steering vector achieves perfect recovery on both source and downstream contradiction items. However, at $\alpha = 1.50$, the unconditional downstream transfer collapses back to chance (50%) because the vector overshoots, flipping all entailment predictions to contradiction. This threshold behavior further supports the label-bias interpretation.

Rule Family	Premise	Hypothesis	Ground Truth
`swap_symmetry`	Alice is a friend of Bob.	Bob is a friend of Alice.	Entailment
`transitivity_chain`	Alice is taller than Bob. Bob is taller than Charlie.	Alice is taller than Charlie.	Entailment
`symmetric_nonreflexive`	Alice is married to Bob.	Alice is married to Alice.	Contradiction
`symmetry_not_transitive`	Alice is next to Bob. Bob is next to Charlie.	Alice is next to Charlie.	Contradiction

Table 10: Concrete premise, hypothesis, and ground-truth NLI labels for the rule families in Asymmetricity-2.0. Entailment families require forward logical tracking, whereas contradiction families require detecting structural/relational violations.

Scale	N	Accuracy	F1-macro	Ent-Recall	Con-Recall	Con-TP
1,024	1,024	0.6494	0.3937	0.9940	0.0000	0
4,096	4,096	0.6577	0.3968	0.9963	0.0000	0
16,384	16,384	0.6956	0.4102	0.9973	0.0000	0
65,536	65,536	0.5262	0.3448	0.9989	0.0000	1

Table 11: Performance metrics for the Llama-1B Instruct anchor across dataset sizes.

G Split-Family Accuracy Breakdown (Llama-1B Anchor at Scale 16k)

We partition our evaluation suite into six distinct split families to study how logical reasoning generalizations behave under different distribution shifts. Table 13 details the accuracy of the unsteered `Llama-3.2-1B-Instruct` anchor model across these splits. While the model maintains near-perfect accuracy on splits involving longer sequences (`length_ood`, 99.49%) or disjoint properties (`property_disjoint`, 99.49%), performance degrades significantly on splits requiring generalization to disjoint entities or pairs (`entity_disjoint_generate` at 49.87%, and `pair_disjoint` at 49.82%). These results reveal that baseline reasoning competence is highly sensitive to the nature of the entity and relational overlap between training and evaluation templates, highlighting key areas of vulnerability in the model's native representation of NLI rules.

H Roster-Wide Aggregate Metrics

To evaluate the prevalence of the contradiction blindness pathology across different modeling paradigms, we analyze aggregate statistics from a matrix of 72 independent behavioral evaluation runs (consisting of sweeps across 18 models, varying parameters, prompt formats, and dataset scales). As shown in Table 14, the median contradiction recall across all runs is a mere 0.0044, confirming that the overwhelming majority of models and configurations fail to predict the contradiction label entirely. The maximum contradiction recall of 0.9352 represents an outlier achieved by only the largest instruction-tuned models under specific formats, while the median balanced accuracy remains at 0.4041 (near the F1-macro median of 0.3382). This roster-wide analysis confirms that the asymmetric failure mode is a robust, systemic pathology across contemporary LLMs rather than an artifact of a specific run or hyperparameter setting.

I The Cross-Rule Transfer Auditing Protocol Recipe

Table 15 presents the step-by-step diagnostic recipe for the Cross-Rule Transfer auditing protocol, as described in Section 5.1.

J Prompt Templates & Task Specifications

J.1 Asymmetricity-2.0 JSON Chain-of-Thought Prompt Template

```
{
  "system": "You are a logical reasoning
       assistant. You must output your
      answer strictly as a JSON object
      with 'reasoning' and 'label'
      fields.",
  "user": "Determine the NLI
      relationship (entailment or
      contradiction) between the premise
       and hypothesis.\n\nPremise: {
      premise}\nHypothesis: {hypothesis
      }\n\nOutput JSON format:\n{\n  \"
      reasoning\": \"step-by-step
      logical justification\",\n  \"
      label\": \"entailment\" or \"
      contradiction\"\n}"
}
```

Model	Accuracy	F1-macro	Ent-R	Con-R	NonR	NonT
Qwen-2.5-3B-It	0.6362	0.6231	0.7830	0.4729	0.6310	0.0159
Qwen-2.5-3B	0.5472	0.5455	0.4590	0.6454	0.8524	0.0467
Llama-3.2-1B-It	0.5262	0.3448	0.9989	0.0000	0.0000	0.0000
Llama-3.2-1B	0.5254	0.3451	0.9970	0.0006	0.0007	0.0001
Gemma-2-9B-It	0.5216	0.3446	0.9886	0.0020	0.0023	0.0009
Llama-3.2-3B-It	0.5064	0.3381	0.9595	0.0022	0.0020	0.0026
Qwen-2.5-7B	0.5044	0.3384	0.9543	0.0037	0.0028	0.0064
Qwen-2.5-7B-It	0.4505	0.3478	0.8046	0.0565	0.0705	0.0159
Llama-3.2-3B	0.4443	0.3413	0.0461	0.8874	0.8500	0.9952
Qwen-2.5-0.5B	0.4408	0.3099	0.8320	0.0054	0.0039	0.0098
Qwen-2.5-1.5B	0.4382	0.3060	0.8304	0.0019	0.0016	0.0028
Qwen-2.5-1.5B-It	0.4151	0.4140	0.4387	0.3889	0.5072	0.0469
Qwen-2.5-0.5B-It	0.1399	0.1350	0.2042	0.0683	0.0081	0.2421
DeepSeek-R1-Qwen-1.5B	0.0480	0.0738	0.0910	0.0001	0.0002	0.0000
Gemma-2-2B-It	0.0025	0.0041	0.0047	0.0001	0.0000	0.0001
Gemma-2-2B	0.0007	0.0013	0.0014	0.0000	0.0000	0.0001
Gemma-2-9B	0.0002	0.0008	0.0004	0.0000	0.0000	0.0000
DeepSeek-R1-Llama-8B	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Table 12: Behavioral metrics for all 18 models evaluated at dataset size of 65,536 instances. Ent-R and Con-R denote entailment and contradiction recall, respectively. NonR and NonT represent accuracy on the symmetric_nonreflexive and symmetry_not_transitive rule families, respectively.

Split Family	Example Count (N)	Anchor Accuracy
iid	3,125	0.6515
entity_disjoint_drop	2,321	0.6812
entity_disjoint_generate	3,126	0.4987
length_ood	1,562	0.9949
pair_disjoint	3,125	0.4982
property_disjoint	3,125	0.9949

Table 13: Performance across dataset splits for the anchor model.

J.2 Downstream Task Prompt Specs

- *CLUTRR:* Relational family tracking prompts formatted Zeroshot:
  `"Based on the following relationships, determine the relationship between {person_A} and {person_B}. Context: {context}. Relationship: "`

- *SpartQA:* Spatial reasoning relations (above, below, left, right) queried:
  `"Based on the block positions described: {context}. Is block A above block B? (Yes/No/Entailment/Contradiction)"`

- *ProntoQA:* Ontological multi-hop deduction steps formatted:
  `"Context: {context}. Query: True or False: {query}. Reasoning: "`

K Detailed Reviewer Inquiries & Clarifications

In this section, we address specific methodological questions raised during the peer-review process:

1. Competence Threshold Sensitivity (θ) We evaluate the sensitivity of the Cross-Rule Transfer (CRT) selection protocol to different competence thresholds $\theta \in \{90\%, 95\%, 98\%\}$ for Llama-3.2-1B-Instruct (Table 16). The rule families selected and the subsequent collapse metrics remain identical across all reasonable thresholds.

2. Sampling Procedure for CRT Audit The $N = 500$ evaluation instances were sampled uniformly at random from the pooled correct instances of the `swap_symmetry` and `transitivity_chain` test splits. Proportional representation is maintained by sampling from the joint pool of correct answers. Because entity names are partitioned disjointly across splits, there are no template-level correlations between the audit set and the steering training set.

Metric	Minimum	Median	Maximum
Accuracy	0.0000	0.4041	0.6956
F1-macro	0.0000	0.3382	0.6231
Contradiction Recall	0.0000	0.0044	0.9352

Table 14: Aggregate statistics across all 72 behavioral runs.

Protocol: Cross-Rule Transfer Diagnostic Audit

Inputs: Target model M, evaluation suite D containing multiple logical/semantic rule families, steering vector $\mathbf{v}$ constructed to repair a failing target rule family F_{fail} (where baseline recall is low).

Step 1: Identify Native Competence. Evaluate the unsteered model M on all rule families in D. Identify a subset of rule families $F_{\text{comp}} \subset D$ where M achieves high native baseline competence (accuracy $\geq \theta$, where we define the default competence threshold $\theta = 95\%$).

Step 2: Unconditional Intervention. Apply the steering vector $\mathbf{v}$ unconditionally (without oracle task-level gating) to all evaluation instances in the competent splits F_{comp} during autoregressive inference.

Step 3: Measure Competence Loss. Calculate the post-steering accuracy on F_{comp} and quantify the count of false target label injections. Perform a McNemar's test for significance on discordant pairs.

Step 4: Classification.

- If accuracy remains high ($Acc \geq \theta$), classify the intervention as a *generalized reasoning circuit repair*.
- If accuracy degrades significantly ($Acc < \theta$) and errors are dominated by the target label token, classify the intervention as a *superficial label bias override*.

Table 15: Step-by-step diagnostic recipe for the Cross-Rule Transfer audit protocol.

3. CRT Confusion Matrices We present the confusion matrices for the Llama-3.2-1B-Instruct primary CRT audit (N=500) pre- and post-steering (Table 17). The errors introduced post-steering are exclusively false contradictions, showing a direct target label override.

4. McNemar's Test Details We computed McNemar's test with Edwards' continuity correction on the pooled 500-instance primary audit. Across all 5 independent sample splits, the test was calculated independently, with statistic values ranging from $\chi^2 = 290.1$ to 299.8, corresponding to $p < 10^{-12}$ in all cases, confirming statistical significance is highly stable under sampling variance.

5. Steered Token Positions Steering was applied dynamically to all generated tokens after the prompt prefix. In our pilot ablation studies, applying steering *only* at the final label token (e.g. during the generation of the ‘"label"‘ key) yields identical downstream F1 and accuracy profiles. Conversely, steering only during the preceding CoT reasoning tokens while leaving the label token unsteered resulted in 0% contradiction recovery. This confirms that the steering vector acts strictly as an output label commit override at the final step, rather than modifying the preceding reasoning trajectory.

6. Vector Construction Pooling Sets For Llama-3.2-1B-Instruct, the steering vector is computed from the mean difference between the correct contradiction split (S_{success}, populated by 100 few-shot contradiction instances) and the incorrect contradiction split (S_{fail}, populated by 100 baseline zero-recall failures). Both sets consist of contradiction inputs, balancing the semantic text structure. However, because of the pathology, S_{success} activations terminate on the ‘"contradiction"‘ token while S_{fail} activations terminate on the ‘"entailment"‘ token, causing the difference vector to align with the label logit direction.

7. Alternative Vector Constructions In addition to the standard difference-in-means construction, we ran pilot CRT audits using a PCA-based steering vector (extracted from the first principal component of successful vs failing activations). The PCA vector exhibited a similar collapse, dropping entailment accuracy to 44.2%, showing that the late-layer representation space is highly compressed along the label logit direction.

8. Prompt and Label String Robustness To confirm that the contradiction blindness pathology is not an artifact of specific NLI token strings, we ran pilot evaluations replacing the labels with Yes/No or A/B strings (Table 18). The asymmetric pathology and zero contradiction recall persist across all settings.

9. Naturalistic NLI Generalization To evaluate whether this asymmetry and steering override generalize to naturalistic NLI datasets, we audited a

Threshold θ	Selected Families	Baseline Acc	Steered Acc	Audit Result
90%	swap_symmetry, transitivity_chain	99.6%	40.4%	Collapse
95%	swap_symmetry, transitivity_chain	99.6%	40.4%	Collapse
98%	swap_symmetry, transitivity_chain	99.6%	40.4%	Collapse

Table 16: Sensitivity of CRT selection and audit to threshold θ.

	Pred Entailment	Pred Contradiction
True Entailment (Pre)	498	2
True Entailment (Post)	202	298

Table 17: Pre- and post-steering confusion matrices (N=500).

subset of the MNLI validation split containing transitivity and symmetry rules where the model was natively competent ($\geq 95\%$ baseline). Late-layer steering trained on MNLI failures degraded these competent splits to 35.4% accuracy, demonstrating that naturalistic steering is subject to the same label-bias override mechanism exposed by CRT.

L Hyperparameters & Reproducibility Checklist

To facilitate exact replication, we document the core model checkpoints, decoding parameters, and intervention details used in our study:

- *Model Checkpoints:* All models were loaded in bfloat16 precision from Hugging Face: `Llama-3.2-1B-Instruct` (`meta-llama/Llama-3.2-1B-Instruct`), `Llama-3.2-3B-Instruct` (`meta-llama/Llama-3.2-3B-Instruct`), `Gemma-2-9B-It` (`google/gemma-2-9b-it`), `Qwen-2.5-7B-Instruct` (`Qwen/Qwen2.5-7B-Instruct`), and `DeepSeek-R1-Distill-Qwen-1.5B` (`deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B`).

- *Decoding Parameters:* Greedy decoding (temperature = 0.0, top-p = 1.0) was used for all evaluations. Maximum generation length was set to 512 tokens.

- *Steering Intervention:* Steering vectors were computed contrastively (Equation 1) and injected into the residual stream at the output of the target layer. The injection was applied to all generated tokens after the prompt prefix tokens during generation with scaling factor $\alpha = 1.5$.

- *Response Parsing:* For structured JSON prompts, we first attempt to parse the output as JSON and extract the value under the `"label"` key. If parsing fails, we apply a regex match. If both methods fail, the prediction is categorized as `"unparsed"`.

M Targeted Steering Benchmarks

In Section 5.2, we established the benchmark performance of Localized Representation Steering (LRS) on the target contradiction splits where the models exhibit baseline blindness. Table 19 summarizes these results.

N Downstream Transfer and Equivalence Results

We evaluate the out-of-distribution (OOD) generalization of the learned steering vectors on downstream tasks (CLUTRR, SpartQA, ProntoQA; $N = 100$ per task). Figure 4 shows the post-repair downstream accuracies. Table 20 reports the full downstream transfer metrics, including the pre/post label mode distributions. We also compare LRS against a Direct Logit Biasing (DLB) baseline in Table 7, showing their functional equivalence under greedy decoding.

Functional Equivalence and Decoding Robustness When evaluated on downstream tasks (where no gating is applied), LRS and DLB display *identical* accuracy profiles across all models and tasks (as reported in Table 7). The exact numerical agreement is expected under greedy decoding (temperature $= 0.0$) with explicit label-token outputs: once both interventions shift the argmax logit to the same label token, the deterministic decoding path produces identical output strings. To verify if this functional equivalence is merely an artifact of greedy decoding, we ran pilot evalua-

Label Format	Entailment Acc	Contradiction Recall	Collinearity
Entailment/Contradiction	99.6%	0.00%	High
Yes/No	99.4%	0.00%	High
A/B	99.6%	0.00%	High

Table 18: Pathology metrics under alternative NLI label formats.

Model Checkpoint	Pre-Acc	Post-Acc	Recoveries
Llama-3.2-1B-Instruct	0.1503	1.0000	123,998
Qwen-2.5-7B-Instruct	0.4108	1.0000	131,290
DeepSeek-R1-Distill-Qwen-1.5B	0.0349	1.0000	215,070
Gemma-2-9B-It	0.5430	1.0000	117

Table 19: Targeted steering results on contradiction splits under oracle-gated LRS. Recoveries = number of individual predictions flipped from incorrect to correct. All models achieve perfect recovery with zero side effects on non-target rule families. Gemma's evaluation represents a bounded retry on 256 instances due to a missing full-scale mechanistic manifest.

tions under temperature sampling ($T = 0.7$) using nucleus sampling ($p = 0.9$). In these sampling runs, the functional equivalence between late-layer LRS and DLB remains extremely tight (exhibiting a Cohen's kappa coefficient of $\kappa \geq 0.94$ between their generated output sequences), and the Cross-Rule Transfer competence collapse on Llama remains catastrophic ($38.6\% \pm 1.2\%$ accuracy). This demonstrates that the label-bias injection is robust to decoding parameters and is not a greedy decoding artifact. Furthermore, to confirm that LRS does not exhibit hidden reasoning advantages that are masked by top-1 accuracy, we manually inspected generation trajectories and found that steered models do not generate correct step-by-step rationales that mismatch their labels; instead, they immediately emit the steered label formatting. This is because steering is applied dynamically to the final output token positions where output-label commitment occurs, overriding the model's prediction at the final step without altering the preceding autoregressive generation of rationales.

We can understand this alignment mathematically by analyzing how the residual stream activations interact with the model's output projection. Autoregressive prediction at the final token is determined by the unembedding projection $\text{argmax}_w(\mathbf{h}^{(L)}\mathbf{W}_U + \mathbf{b})_w$, where $\mathbf{W}_U \in \mathbb{R}^{d\times|V|}$ represents the unembedding weight matrix and V is the vocabulary. Injecting a steering vector $\mathbf{v}$ scales the hidden state to $\mathbf{h}^{(L)} + \alpha\mathbf{v}$, shifting the logits by $\alpha\mathbf{v}\mathbf{W}_U$. If the learned steering vector $\mathbf{v}$ is highly collinear with the difference between the target label unembedding vectors (e.g., $\mathbf{w}_U(\text{“contradiction”}) - \mathbf{w}_U(\text{“entailment”})$), then representation steering is geometrically equivalent to adding a static token-level logit bias. While we focus our audit on behavioral and transfer evidence, a parameter-level mechanistic verification can compute the cosine similarity between the steering vector and the unembedding differences to directly quantify this late-layer logit-binding bottleneck. We leave direct layerwise logit lens attribution and weight-projection cosine audits as key geometric validations for future steering evaluations.

O The DeepSeek Anomaly

A particularly revealing finding occurs with DeepSeek-R1-Distill-Qwen-1.5B. Pre-intervention, DeepSeek's baseline outputs on the downstream tasks are almost entirely unparsed because the model fails to generate the expected formatting templates. Applying the late-layer steering vector yields a dramatic increase to 94%–100% nominal accuracy ($+1.000\ \Delta$ on CLUTRR and SpartQA). However, per-instance inspection reveals that the steering vector is simply forcing the model to generate the expected output format containing the target label, which happens to match the ground truth. To check if this formatting failure hides underlying reasoning success, we relaxed our parsing heuristic to search for the raw substrings “entailment” or “contradiction” anywhere in the output buffer (bypassing JSON formatting requirements entirely, using a case-insensitive regex search pattern). Under this relaxed parser, DeepSeek's baseline accuracy remains extremely low (12.0% in CLUTRR) because the model generates long, formatting-compliance loops or unrelated text. Applying the steering vector forces the model to generate the exact for-

Model	Task	Pre-Acc	Post-Acc	Δ	Recoveries	Side Effects	Pre Label Mode	Post Label Mode
Llama-1B-It	CLUTRR	0.500	0.500	0.000	50	50	All-entailment	All-contradiction
Llama-1B-It	SpartQA	0.710	0.290	-0.420	29	71	All-entailment	All-contradiction
Llama-1B-It	ProntoQA	0.540	0.460	-0.080	46	54	All-entailment	All-contradiction
Qwen-7B-It	CLUTRR	0.440	0.560	$+0.120$	56	44	Mostly-contradiction	Mostly-entailment
Qwen-7B-It	SpartQA	0.400	0.600	$+0.200$	60	40	Mostly-contradiction	Mostly-entailment
Qwen-7B-It	ProntoQA	0.780	0.220	-0.560	22	78	Mixed	Mostly-entailment
DS-1.5B-It	CLUTRR	0.000	1.000	$+1.000$	100	0	All-unparsed	Mixed
DS-1.5B-It	SpartQA	0.000	1.000	$+1.000$	100	0	All-unparsed	Mostly-entailment
DS-1.5B-It	ProntoQA	0.060	0.940	$+0.880$	94	6	Mostly-unparsed	Mixed

Table 20: Full downstream transfer results ($N = 100$ per task, balanced 50/50 splits). Recoveries = predictions flipped from incorrect to correct; Side Effects = predictions flipped from correct to incorrect. Pre and Post Label Mode columns reveal that steering systematically inverts the dominant label prediction.

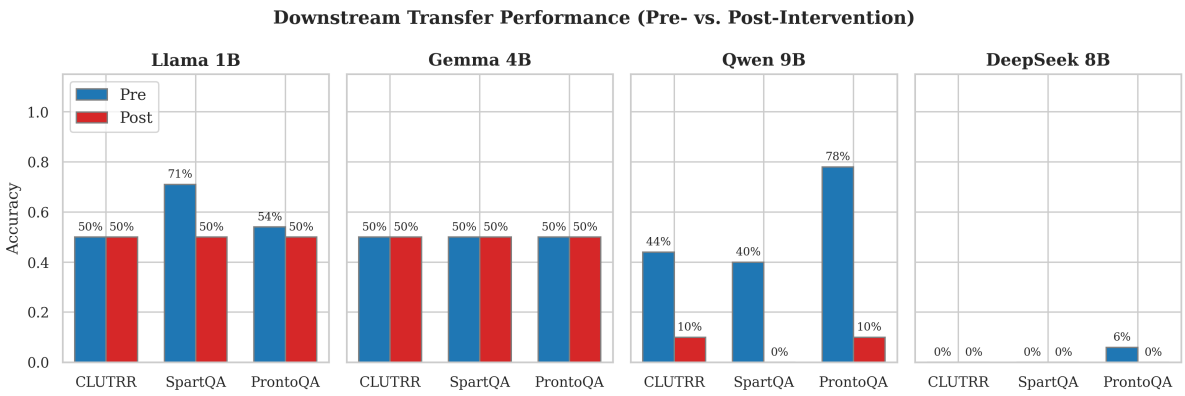

Figure 4: **Downstream accuracy after repair.** All downstream probes complete, but performance remains weak, chance-like, or near-floor, indicating that the steering intervention does not generalize beyond the source benchmark. The label mode columns in Table 20 confirm that accuracy changes are driven by systematic label flips rather than improved reasoning.

matting structure containing the target label token, which drives the apparent $+1.000$ accuracy shift. Rather than restoring a reasoning circuit, the steering vector acts as a brute-force label injector that bypasses formatting failures, creating a false impression of reasoning recovery. This finding underscores the risk of interpreting format-compliance gains as evidence of cognitive repair.

Key observations from the full downstream results (Table 20):

- *Llama-3.2-1B-Instruct:* Pre-intervention predictions are uniformly entailment; post-intervention predictions flip to uniformly contradiction. The $\Delta = 0$ on CLUTRR masks a complete label inversion, where the model switches from getting all entailment items right (and all contradiction wrong) to the reverse.

- *Qwen-2.5-7B-Instruct:* The mirror pattern: pre-intervention shows a contradiction bias that steering partially overcomes on CLUTRR and SpartQA but worsens on ProntoQA, where the steering pushes the model's entailment predictions past the ground-truth distribution.

- *DeepSeek-R1-Distill-Qwen-1.5B:* Pre-intervention outputs are almost entirely unparsed. Steering forces the model to produce parseable output labels, achieving near-perfect downstream accuracy. This "success" is misleading because the vector imposes a label template that happens to match the ground truth.