Where Reasoning Breaks: Logic-Aware Path Selection by Controlling Logical Connectives in LLMs Reasoning Chains

Seunghyun Park
Independent Researcher
qkrpsh97@gmail.com

Yuanyuan Lei
University of Florida
Gainesville, FL, United States
yuanyuan.lei@ufl.edu

Abstract

While LLMs demonstrate impressive reasoning capabilities, they remain fragile in multi-step logic deduction, where a single transition error can propagate through the entire reasoning chain, leading to unstable performance. In this work, we identify *logical connectives* as primary points of this structural fragility. Through empirical analysis, we show that logical connective tokens function as high entropy forking points, at which models frequently struggle to determine the correct logical direction. Motivated by this observation, we hypothesize that intervening in logical connective selection can guide LLMs towards the correct logical direction, thereby improving the overall reasoning chain. To validate this hypothesis, we propose a multi-layered framework that intervenes specifically at these logic-critical junctions in the reasoning process. Specifically, we introduce (1) *Gradient-based Logical Steering* to guide LLMs internal representations towards valid reasoning subspaces, (2) *Localized Branching* to resolve ambiguity via targeted look-ahead search, and (3) *Targeted Transition Preference Optimization*, a surgical reinforcement learning objective that selectively optimizes single-token preferences at logical pivots. Crucially, by concentrating intervention solely on logic-critical transitions, our framework achieves a favorable accuracy–efficiency trade-off compared to global inference time scaling methods like beam search and self-consistency[1].

1 Introduction

Logical reasoning (Zhang et al., 2025b,c) serves as a cornerstone of general intelligence, enabling large language models (LLMs) to tackle rigorous domains such as complex decision-making, program synthesis, or mathematics (Wei et al., 2024; Zhang et al., 2025a; He et al., 2025; Luo et al., 2025). Unlike open-ended generation, logical reasoning requires the model to construct a coherent chain of thought, where valid conclusions are derived from premises through a structured, multi-step deductive process (Wei et al., 2022a; Lightman et al., 2023; Liu et al., 2025; McGinness and Baumgartner, 2024). In this strict framework, the integrity of the sequence is paramount: a single flaw in an intermediate step can propagate into a complete failure of the reasoning chain. Therefore, ensuring the precision of each transition within the reasoning process is critical for reliable model performance.

We observe that *logical connectives* serve as critical points of fragility in reasoning. Logical connectives, such as *therefore*, *however*, and *but*, function as linguistic pivots that explicitly direct the logical links between successive reasoning steps. Throughout this paper, we use the term *logical connectives* to refer to explicit discourse markers that signal logical relations between reasoning steps, following the discourse relation framework of Robaldo (2008). These tokens act as directional signals: a single connective chosen at the wrong step can redirect the reasoning chain and ultimately determine whether the deduction is valid or incorrect. This high-leverage role is illustrated in Figure 1, where replacing only the logical connective is sufficient to flip the final conclusion, revealing that valid reasoning often hinges on these specific transition points.

To further validate this observation, we conduct a diagnostic analysis to uncover the fragility of reasoning process in LLMs (Section 3). Our analysis reveals that logical connectives are pivot points of reasoning fragility and function as critical decision junctions. Firstly, token-level entropy analysis shows that uncertainty is predominantly concentrated at connective positions. Secondly, inspection of the top candidates in the model's distribution indicates that the model's decision primarily focuses on which connective to use, rather than refining the surrounding content. Thirdly, substituting connective tokens alone produces statistically signif-

[1]The code link is: https://github.com/lei-nlp-lab/reasoning_fragility_acl_2026

Findings of the Association for Computational Linguistics: ACL 2026, pages 26309–26324

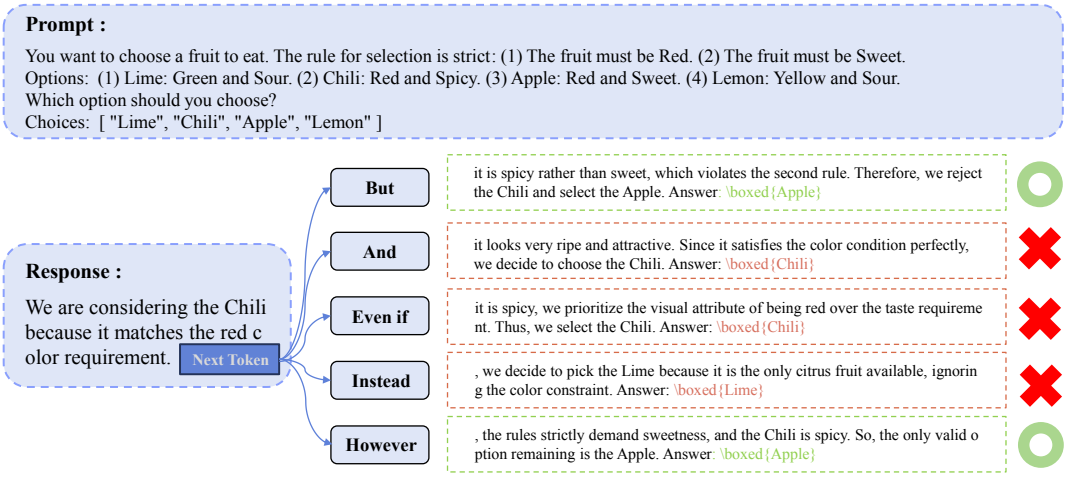


Figure 1: Example showing the fragility of reasoning at logical connective pivots: replacing only the logical connective at a single transition flips the final answer.

icant changes in the final reasoning performance, with such connective perturbations derailing correct chains nearly twice as often as perturbations at other high-entropy positions. Furthermore, analysis at the level of logical relations reveals that successful reasoning repairs occur when connective perturbations drive transitions between different relation types, rather than token substitutions within the same relation category, which confirms that connective choices direct the logical trajectory of the reasoning chain. Taken together, these findings suggest that logical connectives operate as ambiguous forking points where the model struggles to determine the next logical direction, and small mistakes at these junctures can propagate and ultimately degrade the entire reasoning chain.

Motivated by the above observation and analysis, we arrive at the following core hypothesis: *Optimizing logical connectives as the key decision points will guide the model to select a valid logic-aware reasoning path, thereby leading to correct reasoning answer and improving reasoning performance.*

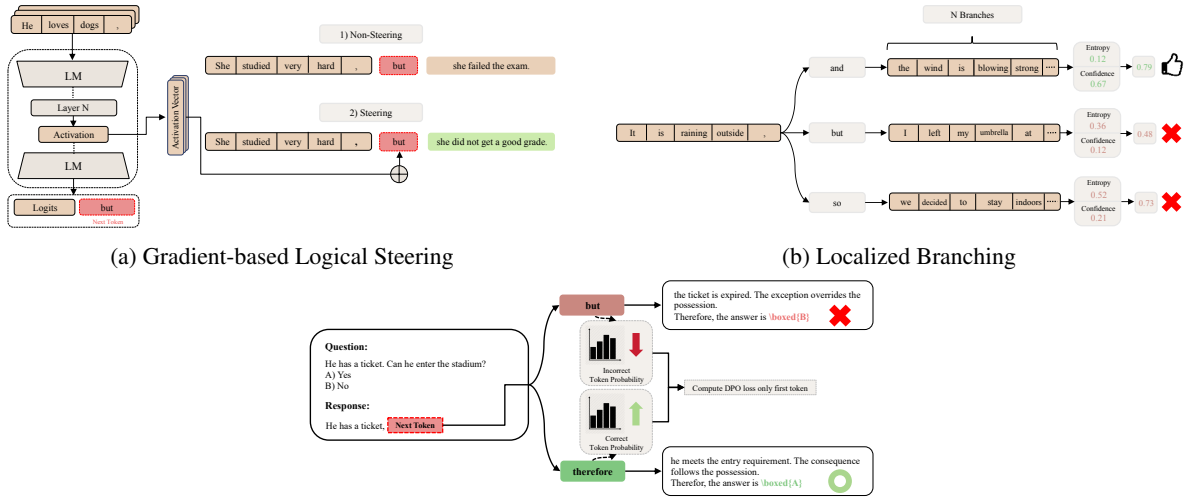
To substantiate this hypothesis, we propose a multi-layered framework that intervenes directly at these logic-critical junctions and provides focused control across three complementary levels. To elaborate: (i) at the *activation level*, we design a gradient-based steering mechanism to steer LLMs internal representations toward valid logic directions, enabling recovery from failing reasoning chains without modifying model weights (Turner et al., 2023; Rimsky et al., 2024), (ii) at the *inference level*, we introduce localized branching, a look-ahead strategy that computes an ambiguity score at connective positions and selectively explores only the most promising logic-guided paths (Zhao et al., 2024; Chen et al., 2023), (iii) at the *training level*, we propose *Targeted Transition Preference Optimization*, a surgical reinforcement learning objective that optimizes single-token preferences at logical pivots to refine the model's distribution. Together, these three strategies guide LLMs towards an optimal logic-aware reasoning path by controlling the logic transition points where reasoning failures most often originate.

The experiments on five logical reasoning benchmarks and four LLMs demonstrate that by shifting the focus from the entire sequence to these high leverage transitions, our approach achieves performance gains while maintaining the efficiency of greedy decoding. Our main contributions are:

- We empirically diagnose logical connectives as the primary points of fragility in reasoning
- We introduce a multi-layered framework to guide LLMs towards a logic-aware reasoning path by intervening directly at logic junctions

2 Related Work

Logical Reasoning Logical reasoning (Zhang et al., 2025c; Yang et al., 2023) in NLP generally refers to the ability to derive valid conclusions from a given set of premises. While LLMs have demonstrated emergent capabilities in deductive reasoning (Creswell et al., 2022; Wei et al., 2022b), they often struggle with *compositionality*, the ability to maintain logical consistency across long, multi-step chains. Previous works (Creswell



(a) Gradient-based Logical Steering

(b) Localized Branching

(c) Targeted Transition Preference Optimization (TTPO)

Figure 2: Connective centric methods across three stages. (a) Steering provides training-free activation intervention at connective junctions. (b) Branching provides compute localized test time search by scoring short continuations for alternative connectives. (c) TTPO provides minimal tuning by optimizing only the single token preference at connective pivots, correcting the underlying distribution that causes brittle transitions.

et al., 2022; Wei et al., 2022b) have attempted to bridge this gap by decomposing reasoning into explicit 'Selection-Inference' steps or by integrating neuro-symbolic modules (Creswell and Shanahan, 2022; Pan et al., 2023) that translate natural language into formal logic representations. However, these approaches often require rigid formalisms or external solvers. In contrast, our work focuses on the *intrinsic linguistic mechanism* of reasoning within the model. We posit that the failures in multi-step deduction often stem from imprecise transitions at the token level, specifically at logical connectives, which serve as the natural language operators for deduction.

Inference Time Intervention Test-time scaling methods improve reasoning by allocating extra compute via global search or sampling (e.g., Self-Consistency, ToT) (Lightman et al., 2023; Chen et al., 2025; Wang et al., 2022; Yao et al., 2023), but they are often compute-inefficient because they scale over entire trajectories. We instead localize intervention to *connective pivots*: (i) activation-level steering nudges hidden states without weight updates (Panickssery et al., 2024; Turner et al., 2023), and (ii) pivot triggered lookahead branching explores only when connective ambiguity is detected (Zhao et al., 2024; Chen et al., 2023). This yields a targeted alternative to global scaling while preserving near greedy decoding behavior.

Critical Tokens and Token-Level Analysis Recent studies show that uncertainty is often dominated by a small set of high-entropy tokens that act as branching points, and that editing such tokens can disproportionately change the final answer (Wang et al., 2025; Zur et al., 2025; Bogdan et al., 2025; Lin et al., 2024). We adopt this token level perspective but focus on a linguistically grounded subset: logical connectives. Using entropy statistics on CoT traces, we find that high entropy events concentrate on connective positions, and we intervene directly at these junctions during generation via both inference time mechanisms and a targeted training objective.

3 Preliminary

In this section, we empirically validate our core hypothesis: *logical connectives act as the primary decision-making pivots in CoT reasoning, yet they represent points of fragility in current LLMs.* We conduct our pilot analysis primarily on ZebraLogic (Lin et al., 2025), and additionally use the deductive subset of BIG-Bench Hard (BBH) (Suzgun et al., 2023) as a controlled testbed for single token replacement experiments.

3.1 High Entropy Rate of Logical Connectives

To quantify the intrinsic uncertainty associated with logical transitions, we measure the **High Entropy Rate** (R_{HE}) specifically for the set of logical connectives $\mathcal{S}_l$ (Appendix A) on the ZebraLogic

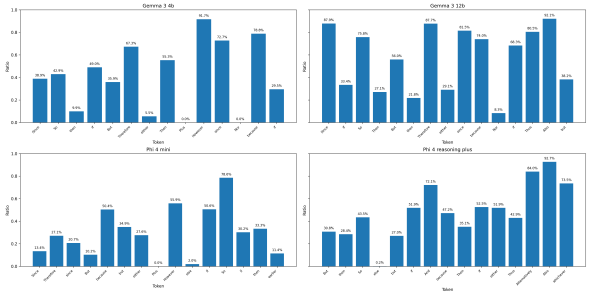


Figure 3: Analysis of token entropy at logical connective positions. The proportion of connectives exceeding the entropy threshold $\tau = 1.0$.

dataset. This metric represents the probability that the model encounters high uncertainty when generating a connective. Following our experimental setup, we define R_{HE} as:

$$R_{HE}(\mathcal{S}_l) = \frac{\sum_t I(w_t \in \mathcal{S}_l \wedge H_t > \tau)}{\sum_t I(w_t \in \mathcal{S}_l)} \quad (1)$$

where $I(\cdot)$ is the indicator function and $\tau = 1.0$ is the entropy threshold.

As illustrated in Figure 3, our empirical analysis reveals that a substantial proportion of logical connectives are generated under high entropy conditions. To ensure this finding is not an artifact of a specific threshold, we conducted two supplementary analyses (detailed in Appendix B). First, a threshold free quantile enrichment analysis shows that connectives are over-represented in every high-entropy tail by a factor of 2.0–5.8×, despite constituting only ∼4–7% of all generated tokens. Second, a comparison across token categories over a broad τ sweep confirms that logical connectives consistently exhibit the highest R_{HE} among all categories at every threshold tested.

3.2 Logical Ambiguity in Top-K Candidates

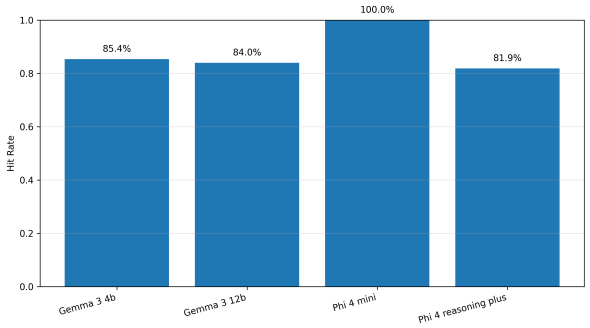


Figure 4: Logical connective presence in the Top-5 candidate set across models.

To determine if high entropy reflects mere stylistic variation or genuine logical ambiguity, we inspect the Top-5 candidate distribution at these junctions, again on ZebraLogic. We find that in the majority of high entropy cases, the Top-5 set contains at least one alternative logical connective.

This indicates that these positions are *forking points*, the model's internal representations are simultaneously entertaining multiple, divergent reasoning trajectories.

3.3 Disproportionate Causal Leverage of Logical Connectives

To evaluate the causal leverage of logical connectives, we perform single-token perturbation experiments on both ZebraLogic and the BBH deductive subset. Specifically, we replace exactly one greedily selected connective with a random alternative from $\mathcal{S}_l$ and observe the impact on the final reasoning outcome.

Original → Modified	Label	Rate (%)
Correct → Correct	C → C	23.4
Correct → Incorrect	C → I	16.3
Incorrect → Incorrect	I → I	50.5
Incorrect → Correct	I → C	9.8

Table 1: Distribution of answer correctness under a single random logical connective replacement on ZebraLogic.

As shown in Table 1, a single random token change at a logical junction causes a significant divergence in the final answer. To properly interpret this causal impact, we measure two conditional metrics. First, we observe a *conditional fragility* of 41.1% among originally correct chains ($\frac{C \to I}{C \to C + C \to I}$), meaning nearly half of the correct reasoning trajectories are completely derailed by changing just one connective.

Conversely, we observe a *conditional repair rate* of 16.2% among originally incorrect chains ($\frac{I \to C}{I \to I + I \to C}$). This rate represents a strict lower bound on repairability, as it reflects the outcome of *random* connective selection; our targeted methods (Section 4), which guide the model to select connectives based on contextual signals, are designed to operate substantially above this baseline. The asymmetry between high fragility (41.1%) and low random repair (16.2%) reveals that the space of trajectory-derailing connectives is much larger than the space of trajectory-correcting ones, directly motivating the need for principled connective selection at these high-leverage pivots.

Single-Token Replacement Type	C → I Rate (%)
Connective → Random connective	22.8
Connective → Same-class connective	21.7
Non-connective → Random token	13.0

Table 2: Controlled single-token replacement on the BBH dataset. Perturbing a connective derails the correct reasoning chain nearly twice as often as perturbing a general high-entropy token.

Crucially, we prove that this disproportionate leverage is unique to logical connectives, rather than a generic property of any uncertain position. Specifically, we conduct a controlled single-token replacement study on the deductive subset of the BBH dataset (Table 2), and compare the impact of perturbing a connective against perturbing a generic non-connective token drawn from the high-entropy tail. We found that perturbing a connective derails a correct chain ($C \rightarrow I$) at a rate of 22.8%, whereas perturbing a non-connective high-entropy token disrupts the chain only 13.0% of the time. This occurs despite logical connectives constituting only a small fraction (∼4–7%) of all generated tokens. Furthermore, replacing a connective with another connective from the *same* logical class (e.g., "but" to "however") still yields a high 21.7% failure rate, comparable to random cross-class replacements. This disproportionate per-token impact confirms that connective positions are structurally brittle pivot points. They do not merely represent unstable decoding steps, but act as high-leverage operators that structurally amplify local perturbations into downstream trajectory divergence.

3.4 Logical Relation Level Analysis of Connective Repairs

The previous subsections establish that connective positions exhibit disproportionate causal leverage over reasoning outcomes. A natural follow-up question is whether the repairs observed at these pivots stem from genuine shifts in logical relations, or merely from stylistic redistribution of near-synonyms (e.g., swapping "but" for "however").

To address this, we analyzed the transition patterns of successful repairs at connective pivots on the BBH deductive subset using Gemma-3-4b-it. Our transition-level data reveals that the actual repairs driving performance improvements are highly concentrated in specific, semantically meaningful *cross-class* relation transitions, rather than stylistic within-class swaps.

Source Class	Target Class	Repair Rate (%)
Causal	Instantiation	38.6
Causal	Analogy	32.9
Causal	Condition	27.2
Causal	Contrast	26.9

Table 3: Top logical relation transitions driving reasoning repair ($I \rightarrow C$) on the BBH deductive subset (Gemma-3-4b-it). Repair Rate denotes the conditional probability of repair given the specific source→target transition ($\frac{I \rightarrow C}{I \rightarrow I + I \rightarrow C}$ within each transition type). The highest-yield repairs involve redirecting a premature causal commitment to different logical operations.

As shown in Table 3, the most effective repairs involve redirecting the model from an incorrect *causal* conclusion (e.g., "therefore", "hence") to fundamentally different logical operations. Specifically, 61.0% of all cross-class $I \rightarrow C$ repairs originate from causal connectives. The highest-yield repairs involve shifting from a causal commitment to an *instantiation* (38.6% conditional repair rate).

These transitions are not stylistic swaps. Replacing "therefore" with "for example" or "unless" fundamentally alters the logical operation from a deductive commitment to an illustration, exception, or qualification. This redirection forces the entire downstream chain into a different trajectory. This transition analysis directly confirms that correcting a reasoning chain at these pivots requires context-dependent shifts in logical relations—validating our core hypothesis that targeted intervention at connective positions fundamentally guides the model toward valid logical paths.

4 Method

4.1 Gradient-Based Logical Steering

To accurately capture the latent representation responsible for logical connectives, we design a gradient based steering method. While prior steering approaches (Rimsky et al., 2024; Turner et al., 2023) often rely on averaging activation states, such methods risk capturing context specific semantic noise rather than the functional role of the connective. In contrast, our approach utilizes the gradient of the target probability, which isolates the causal direction in the activation space that maximizes the likelihood of the correct logical connective.

Vector Extraction Let $\mathcal{D}$ (Appendix C.3.1) be a dataset containing prompts paired with reference logical connectives. For a given input sequence x, let w_c denote the correct logical connective token

at index t. We focus on the hidden state $\mathbf{h}_t^l \in R^d$ from a target layer l immediately preceding w_c. We compute the gradient of the log probability of w_c with respect to $\mathbf{h}_t^l$:

$$\mathbf{g}^{(i)} = \nabla_{\mathbf{h}_t^l} \log P(w_c \mid \mathbf{h}_t^l; x_{<t}) \tag{2}$$

Here, $\mathbf{g}^{(i)}$ represents the direction in the representation space that locally pushes the model to generate the logical token. We aggregate these gradients over N samples from OpenThoughts (Guha et al., 2025) to compute the global steering vector $\mathbf{v}_{steer}$. The vector is obtained by averaging the normalized gradients:

$$\mathbf{v}_{\text{steer}} = \frac{1}{N} \sum_{i=1}^{N} \text{Normalize}\Big(\mathbf{g}^{(i)}\Big) \tag{3}$$

Inference Time Intervention During inference, we inject this extracted feature back into the model to guide reasoning when encountering logical connectives. For a new input at time step t and layer l, the original hidden state $\mathbf{h}_t^l$ is modified by adding the steering vector scaled by a coefficient α:

$$\tilde{\mathbf{h}}_t^l = \mathbf{h}_t^l + \alpha \cdot \mathbf{v}_{steer} \tag{4}$$

This operation effectively shifts the activation state towards the logical reasoning subspace without altering the model's weights. By tuning α, we can control the strength of the logical guidance, ensuring the model remains attentive to transitions even in greedy decoding.

4.2 Logical Connective Branching

While the steering method (Section 4.1) implicitly guides the model within the activation space, our branching strategy intervenes explicitly in the decoding space to resolve ambiguity at critical junctions. As identified in our analysis, logical connectives often serve as high entropy 'forking points'. When multiple plausible connectives appear at a pivot, we expand K candidate continuations and select among the branched candidates using both signals: (i) an entropy based uncertainty estimate and (ii) a confidence score. This idea is inspired by the general principle from (Fu et al., 2025), which ranks the branch with entropy and confidence score. To navigate these bifurcations, we propose a lookahead based selection mechanism that prioritizes the reasoning path with the highest model certainty.

Trigger Mechanism and Branching To efficiently detect intervention points without computational overhead, we define a set of target logical connectives $\mathcal{S}_l$ (Appendix A). During the greedy decoding process at step t, we apply a two-stage trigger criterion: branching is activated only when (i) the top-1 candidate token w_{top1} belongs to $\mathcal{S}_l$, and (ii) at least two candidates from $\mathcal{S}_l$ appear within the top-K predictions. This ensures that branching occurs only at genuine decision points where the model exhibits structural uncertainty, avoiding unnecessary lookahead at low-ambiguity steps.

When triggered, instead of committing to w_{top1}, we filter the top-K candidates for tokens present in $\mathcal{S}_l$, forming a candidate set $\mathcal{C} = \{c^{(1)}, \ldots, c^{(m)}\}$ where $m \leq K$.

Lookahead Evaluation For each candidate connective $c^{(k)}$, we perform a lookahead generation of length L to obtain a continuation trajectory $\tau^{(k)} = (y_1, \ldots, y_L)$. We evaluate the quality of each trajectory using two complementary metrics:

Trajectory Entropy (H) This measures the model's uncertainty regarding the *path* generated after the connective. Lower entropy implies a clearer reasoning chain.

$$H(\tau^{(k)}) = \frac{1}{L} \sum_{j=1}^{L} \mathcal{H}\left(P(\cdot \mid x, c^{(k)}, \tau_{<j}^{(k)})\right) \tag{5}$$

where $\mathcal{H}$ denotes the entropy of the next token distribution.

Sequence Confidence (S) Entropy measures the spread of the distribution, but not the correctness. To ensure the generated path is high probability, we compute the length normalized log probability of the sequence:

$$S(\tau^{(k)}) = \frac{1}{L} \sum_{j=1}^{L} \log P(y_j \mid x, c^{(k)}, \tau_{<j}^{(k)}) \tag{6}$$

Selection Strategy Directly combining $H(\tau^{(k)})$ and $S(\tau^{(k)})$ can be scale sensitive. We therefore normalize each signal *across the candidate set at the current pivot*. Let $\{H_k\}_{k=1}^m$ and $\{S_k\}_{k=1}^m$ denote the entropy and confidence values for candidates in $\mathcal{C}$. We compute

$$\tilde{H}_k = \frac{H_k - \mu_H}{\sigma_H + \epsilon}, \qquad \tilde{S}_k = \frac{S_k - \mu_S}{\sigma_S + \epsilon}, \tag{7}$$

where (μ_H, σ_H) and (μ_S, σ_S) are the mean and standard deviation over $\mathcal{C}$, and ϵ is a small constant. We then select the branch using a joint score:

$$c^* = \text{argmin}_{c^{(k)} \in \mathcal{C}} \left(\tilde{H}_k - \tilde{S}_k \right). \quad (8)$$

This lookahead ensures that the chosen connective leads to a coherent and confident reasoning trajectory, mitigating the error propagation typical of standard greedy decoding.

4.3 Targeted Transition Preference Optimization (TTPO)

While inference time strategies (Sections 4.1 and 4.2) effectively guide the model, they do not rectify the underlying probability distribution that causes logical errors. Traditional alignment methods like RLHF or DPO optimize preferences over entire response sequences. However, training on full sequences to correct specific local transitions is computationally expensive and may introduce gradients that degrade general language modeling capabilities on non-critical tokens.

To address this, we propose **TTPO**. This method adapts the DPO objective to focus exclusively on the single token decision step at logical branching points, serving as a highly efficient, surgical fine-tuning stage.

Objective Formulation Let $\mathcal{D} = \{(x, w_c, w_r)\}$ (Appendix C.3.1) be a dataset of triplets, where x is the context right before a logical transition, w_c is the chosen connective, and w_r is the rejected one. To focus on the transition step, we define a per token log ratio score $s_\theta(x, w) = \log \pi_\theta(w \mid x) - \log \pi_{\text{ref}}(w \mid x)$. Then TTPO maximizes the margin between w_c and w_r only at this step:

$$\Delta_\theta(x, w_c, w_r) = s_\theta(x, w_c) - s_\theta(x, w_r). \quad (9)$$

$$\mathcal{L}_{\text{TTPO}} = -E_{\mathcal{D}} \big[\log \sigma(\beta \, \Delta_\theta(x, w_c, w_r)) \big]. \quad (10)$$

where π_θ is the policy model, π_{ref} is the frozen reference model, and β is a hyperparameter controlling the deviation penalty.

Surgical Gradient Updates The key advantage of TTPO lies in its *gradient sparsity*. During training, we perform a forward pass only up to the logical connective position. The loss is computed solely based on the logits of w_c and w_r, meaning gradients are backpropagated only from this specific timestep.

5 Result

5.1 Experimental Setup

Models All experiments are conducted on two families of instruction tuned model: Gemma 3 series (Team et al., 2025), Gemma-3-4b-it and Gemma-3-12b-it, Phi 4 series (Abdin et al., 2024), Phi-4-mini-instruct and Phi-4-reasoning-plus.

Baselines We report two standard decoding baselines. **(1) Greedy decoding** serves as our primary baseline, representing the common single pass inference setting. Additionally include **(2) Beam search** as a stronger search-based baseline that explores multiple candidate continuations while remaining deterministic at inference time.

Dataset To evaluate deductive logical reasoning across diverse forms of structured inference, we use five benchmark datasets: ZebraLogic (Lin et al., 2025), BIG-Bench Hard (deductive subset) (Suzgun et al., 2023), RuleBERT (Saeed et al., 2021), LogiQA 2.0 (Liu et al., 2023), and PrOntoQA (Saparov and He, 2022). These datasets collectively cover multi-step deduction, rule based inference, and formal/implicit logical transitions, providing a comprehensive testbed for evaluating robustness at logical junctions.

We additionally use OpenThought only for extracting the gradient-based steering vector (Section 4.1). Importantly, OpenThought (Guha et al., 2025) is not used for inference time generation, candidate selection, branching evaluation, or RL training; all reported results are produced solely by the four target models listed above.

Method Configurations Detailed hyperparameters and implementation settings for STEERING, BRANCHING, and TTPO are provided in Appendix C and D.

5.2 Main Results

Table 4 summarizes results across five benchmarks, comparing deterministic decoding baselines against our connective centric methods spanning different stages:

Greedy is fragile, and global search is not uniformly helpful. Beam search improves performance on several datasets, but it can also degrade accuracy on others, notably at ZebraLogic. This reinforces the practical limitation that the best decoding strategy remains task dependent, and stronger global search does not guarantee a better reasoning

	Gemma-3-4b-it					Phi-4-mini-instruct (4B)				
	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA
Greedy (BL)	38.8	75.3	60.3	55.2	90.0	38.8	67.3	50.3	57.7	93.6
Beam Search (BL)	33.4	**77.0**	**67.7**	54.0	**90.6**	31.8	**73.8**	51.7	59.0	93.2
Steering	39.0	75.5	62.0	55.5	90.2	**39.6**	68.8	50.8	58.3	94.2
Branching	**42.0**	74.7	63.4	**56.8**	90.2	**39.6**	69.1	**51.9**	57.0	94.8
TTPO	40.4	75.9	60.8	56.0	90.4	38.6	67.2	50.7	**59.4**	**96.6**
	Gemma-3-12b-it					Phi-4-reasoning-plus (13B)				
	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA
Greedy (BL)	53.2	87.5	58.2	59.2	99.0	**60.8**	70.2	41.6	47.8	**97.8**
Beam Search (BL)	51.8	**89.2**	57.5	**60.1**	**99.2**	57.8	91.3	**47.4**	**49.8**	94.2
Steering	53.2	87.8	**59.0**	59.4	99.0	60.2	88.7	41.4	47.0	**97.8**
Branching	52.4	87.4	57.0	58.5	98.8	60.4	**92.5**	43.2	43.6	96.0
TTPO	**55.6**	86.4	57.3	59.5	**99.2**	58.4	86.4	43.7	45.7	97.6

Table 4: Overall results of steering, branching, and TTPO.

	Gemma-3-4b-it					Phi-4-mini-instruct (4B)				
	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA	ZebraLogic	BBH (Ded.)	RuleBERT	LogiQA 2.0	ProntoQA
Greedy (BL)	38.8	75.3	60.3	55.2	90.0	38.8	67.3	50.3	57.7	93.6
Beam Search (BL)	33.4	77.0	**67.7**	54.0	90.6	31.8	**73.8**	51.7	59.0	93.2
TTPO	40.4	75.9	60.8	**56.0**	90.4	38.6	67.2	50.7	**59.4**	**96.6**
TTPO + Steering	**41.4**	76.5	60.7	54.9	90.0	**40.2**	70.4	51.1	58.0	94.2
TTPO + Branching	39.2	**77.1**	61.3	55.4	**92.2**	38.6	69.1	**52.1**	57.7	94.2

Table 5: Comparing greedy, beam search and TTPO with branching and steering.

trajectory under strict evaluation. We additionally compare with self-consistency ($n = 5$) in Table 11 (Appendix), where our methods achieve comparable average accuracy at substantially lower compute cost (Section 6.4).

Inference time interventions recover greedy failures at logical pivots. On the smaller models, our inference time methods yield consistent gains at connective junctions. For Gemma-3-4b-it, localized branching achieves the best ZebraLogic score (38.8 vs. 42.0) and improves LogiQA (55.2 vs. 56.8), indicating that resolving ambiguity only at connective forking points can recover failures without global search. Steering also improves over greedy across both 4B models (e.g., ZebraLogic 39.0/39.6, BBH 68.8 on Phi), suggesting that a lightweight representation shift can stabilize local transitions even under greedy decoding.

TTPO improves greedy behavior with minimal, localized optimization. TTPO consistently improves or matches greedy on most settings, while remaining strictly focused on the single token decision at logical pivots. Notably, TTPO is particularly strong on Phi-4-mini-instruct for ProntoQA (93.6 vs. 96.6) and LogiQA (57.7 vs. 59.4), demonstrating that refining *connective logits* alone can yield meaningful end task gains. For the larger Gemma-3-12b-it model, TTPO achieves the best ZebraLogic performance (53.2 vs. 55.6), suggesting that token level transition refinement remains beneficial even when the base model is already strong.

Ablation and Method Complementarity. To understand how our three interventions interact, we report additional ablations on the 4B models in Table 5. The results indicate that the proposed methods are not merely redundant variants of test time scaling, but provide complementary control signals at different stages of generation.

TTPO alone improves greedy decoding on multiple benchmarks, confirming that refining connective level decisions translates to end task gains. Combining TTPO with steering further increases robustness where greedy trajectories are brittle at local transitions, while combining TTPO with branching benefits tasks where connective ambiguity is frequent and lookahead is informative.

6 Analysis

6.1 The Relationship between Connective Density and Methodological Gains

Our methods trigger only at explicit connective pivots; thus, gains correlate with connective density (Figure 5). Benchmarks with sparse connectives (e.g., BBH, LogiQA 2.0) offer fewer intervention opportunities, limiting improvements even when overall reasoning remains imperfect.

6.2 Discriminative Performance of Lookahead Evaluation

The Sparsity of Logical Transitions As shown in Figure 5, LogiQA 2.0 and BBH contain fewer

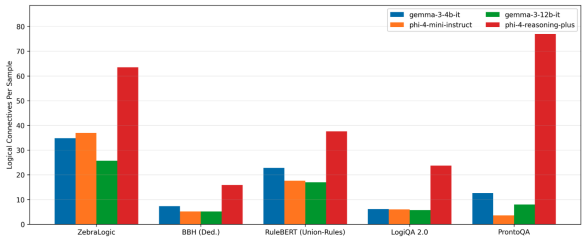


Figure 5: Average count of logical connectives per sample across benchmarks under greedy decoding.

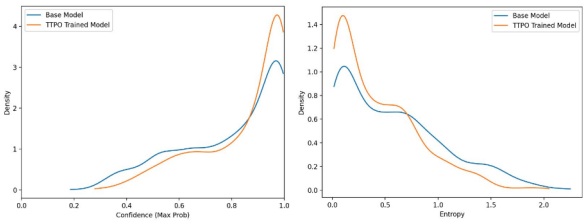


Figure 6: Distribution of prediction confidence *(left)* and entropy *(right)* at logical connective positions. TTPO trained models exhibit a significant shift toward higher certainty and lower ambiguity.

explicit logical connectives per sample than ZebraLogic, limiting intervention opportunities for Steering and Branching. This limits the achievable gains, since failures outside explicit connective transitions are less likely to be corrected by our interventions.

To validate the reliability of our branching metric, we evaluated whether the lookahead score effectively identifies the ground truth logical connective. We ranked a candidate set containing the ground-truth token alongside the model's top-K predictions based on our proposed metric.

Model	Match Rate (%)
Gemma-3-4b-it	73.41
Phi-4-mini-instruct	69.14

Table 6: Match rate of the lookahead selection mechanism against ground-truth logical connectives.

As shown in Table 6, our metric demonstrates strong discriminative capability, aligning with the ground truth in 73.4% and 69.1% of cases for Gemma and Phi models, respectively. This confirms that the lookahead score acts as a robust proxy for logical validity, successfully guiding the model toward the correct reasoning path.

6.3 Distributional Sharpening at Logical Pivots

To investigate how TTPO refines the model's internal decision making, we analyze the probability distribution at the specific positions where logical connectives ($\mathcal{S}_l$) are predicted. Figure 6 illustrates the density of prediction confidence and entropy for Gemma-3-4b-it.

The results demonstrate that TTPO effectively sharpens the probability distribution at logical pivots. While the baseline model exhibits high entropy and low confidence at these forking points, TTPO trained models show a marked shift toward higher certainty and lower entropy. This distributional sharpening indicates that our surgical optimization successfully resolves the structural ambiguity inherent in logical transitions. By increasing the margin between the optimal connective and sub-optimal alternatives, the model commits decisively to a single reasoning path, thereby mitigating error propagation during greedy decoding without the need for exhaustive search.

6.4 Accuracy and Efficiency Trade-off

Table 12 shows that global test time scaling baselines incur $3{\sim}6\times$ higher token cost and up to $2.8{\sim}4.1\times$ higher latency than greedy on 4B models, whereas our connective centric methods remain close to single pass decoding. As shown in Table 11, on Gemma-3-4b-it our best method per task achieves higher average accuracy than self-consistency (n=5) (65.7 vs. 65.2) at $\leq 1.45\times$ greedy cost, compared to $2.90\times$ for self-consistency. On Phi-4-mini-instruct, self-consistency achieves marginally higher accuracy (63.8 vs. 63.3) but at $4.37\times$ token cost—a substantial overhead for a 0.5 point gain. Overall, targeting connective pivots approximates the benefits of global sampling at a fraction of the compute.

7 Conclusion

We identify logical connectives as critical pivots that dictate reasoning trajectories. Our diagnostic analysis reveals that these tokens concentrate model uncertainty and exhibit disproportionate causal leverage over reasoning outcomes. Rather than relying on global optimization, we demonstrate that intervening only at these transitions, via activation-level steering, localized branching, and targeted preference optimization, is sufficient to enhance reasoning performance while preserving near-greedy efficiency, suggesting that linguistically grounded token-level intervention offers a complementary paradigm to global test time scaling.

Limitations

Our approach faces limitations stemming from its reliance on explicit linguistic markers. First, by utilizing a predefined taxonomy of logical connectives, our method cannot address implicit reasoning steps where transitions occur without specific discourse markers. Second, the detection mechanism is sensitive to tokenizer artifacts and polysemy, which may introduce alignment errors across different models. Finally, the effectiveness of our framework is bounded by the density of connectives; tasks with sparse explicit transitions, such as BBH, show limited performance gains compared to linguistically rich datasets like ZebraLogic.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Paul C Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Antonia Creswell, Murray Shanahan, and Irina Higgins. 2022. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, and 1 others. 2025. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*.

Junxian He, Yu Wu, Junlong Li, Daya Guo, Dejian Yang, and Runxin Xu. 2025. Codei/o: Condensing reasoning patterns via code input-output prediction.

Yuanyuan Lei and Ruihong Huang. 2024. Boosting logical fallacy reasoning in llms via logical structure tree. *arXiv preprint arXiv:2410.12048*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. 2025. Zebralogic: On the scaling limits of llms for logical reasoning. *arXiv preprint arXiv:2502.01100*.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. 2024. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. 2025. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. *arXiv preprint arXiv:2506.04592*.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.

Jiebo Luo, Xin Qian, Julian McAuley, Yuwei Cao, Daoan Zhang, Tianyang Liu, Xiaoyi Liu, Antoine Simoulin, Grey Yang, Dayu Yang, and Zhaopu Teng. 2025. Code to think, think to code: A survey on code-enhanced reasoning and reasoning-driven code intelligence in llms.

Lachlan McGinness and Peter Baumgartner. 2024. Automated theorem provers help improve large language model reasoning. *arXiv preprint arXiv:2408.03492*.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Wang. 2023. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3806–3824.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition, 2024. 3.

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

Livio Robaldo. 2008. The penn discourse treebank 2.0. In *Lrec*.

Mohammed Saeed, Naser Ahmadi, Preslav Nakov, and Paolo Papotti. 2021. Rulebert: Teaching soft rules to pre-trained language models. *arXiv preprint arXiv:2109.13006*.

Abulhair Saparov and He He. 2022. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. *arXiv preprint arXiv:2210.01240*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and 1 others. 2023. Challenging big-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Furu Wei, Wenshan Wu, Shaoguang Mao, Yadong Zhang, Yan Xia, Tao Ge, Man Lan, Xun Wang, Ting Song, and Adrian de Wynter. 2024. Llm as a mastermind: A survey of strategic reasoning with large language models.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022a. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zonglin Yang, Xinya Du, Rui Mao, Jinjie Ni, and Erik Cambria. 2023. Logical reasoning over natural language as knowledge representation: A survey. *arXiv preprint arXiv:2303.12023*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models, 2023. 3:1.

Jiaxin Zhang, Zhijiang Guo, Haotian Xu, Yingying Zhang, Jiahua Dong, Yuxuan Yao, Le Song, Cheng-Lin Liu, Junhao Zheng, Fei Yin, Duzhen Zhang, Xiuyi Chen, Zhong-Zhi Li, Ming-Liang Zhang, Zengyan Liu, and Pei-Jie Wang. 2025a. From system 1 to system 2: A survey of reasoning large language models.

Kun Zhang, Zhouchen Lin, Haoxuan Li, Robert van Rooij, Fengxiang Cheng, and Fenrong Liu. 2025b. Empowering llms with logical reasoning: A comprehensive survey.

Yue Zhang, Chaoli Zhang, Hanmeng Liu, Ruoxi Ning, Mengru Ding, Zhizhang Fu, and Xiaozhang Liu. 2025c. Logical reasoning in large language models: A survey.

Yao Zhao, Zhitian Xie, Chen Liang, Chenyi Zhuang, and Jinjie Gu. 2024. Lookahead: An inference acceleration framework for large language model with lossless generation accuracy. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6344–6355.

Amir Zur, Atticus Geiger, Ekdeep Singh Lubana, and Eric Bigelow. 2025. Are language models aware of the road not taken? token-level uncertainty and hidden state dynamics. *arXiv preprint arXiv:2511.04527*.

A Logical Connective Set

To systematically identify the pivotal points in reasoning chains, we define a set of target logical connectives, denoted as $\mathcal{S}_l$. Our taxonomy is grounded in the discourse relation framework proposed by Robaldo (2008) and further refined by recent work in logical fallacy detection (Lei and Huang, 2024).

We categorize logical relations into ten distinct types that are essential for multi step deduction: *Conjunction, Alternative, Restatement, Instantiation, Contrast, Concession, Analogy, Temporal, Condition,* and *Causal* relations. The set $\mathcal{S}_l$ is constructed by aggregating explicit discourse connectives associated with these categories. In this process, we explicitly exclude high frequency ambiguous tokens such as "and" and "or". While they theoretically denote logical relations, they predominantly function as syntactic coordinators in natural language. Pruning these tokens ensures that our framework targets explicit deductive transitions rather than simple grammatical conjunctions.

Multi-token connectives. Importantly, many discourse connectives are realized as multitoken phrases (e.g., "as a result", "in contrast", "on the other hand") rather than a single token under subword tokenization. Therefore, our intervention target $\mathcal{S}_l$ is defined at the string/phrase level, and we implement connective detection using suffix matching over the most recent n generated tokens. Concretely, at each decoding step we detokenize the last n tokens (we use n large enough to cover the longest connective phrase in $\mathcal{S}_l$) and check whether any connective phrase in $\mathcal{S}_l$ matches the resulting suffix. This allows STEERING and BRANCHING to reliably trigger at both single token and multi token connective junctions.

Table 9 presents the categorization and representative examples of the logical connectives used in our experiments. These connectives serve as the trigger points for our STEERING, BRANCHING, and TTPO mechanisms.

B Entropy Analysis

To ensure that our findings regarding the high entropy of logical connectives are robust and not artifacts of a specific threshold (τ), we present detailed supplementary analyses conducted on the ZebraLogic dataset.

B.1 Threshold Free Quantile Enrichment

We first conducted a threshold free quantile enrichment analysis. Let H_q be the q-th quantile of the entropy distribution over all generated tokens. We measure the proportion of logical connectives in the overall generation (Base %) and compare it to their proportion within the high-entropy tail (Tail %).

As shown in Table 7, even though connectives constitute only a small fraction of all tokens (4.05% for Gemma, 6.73% for Phi), they are strongly overrepresented in every high-entropy tail. For instance, in the top 1% most uncertain tokens ($q = 0.99$), connectives are enriched by $5.77\times$ and $2.01\times$ for Gemma and Phi, respectively. This confirms that connectives intrinsically dominate the model's structural uncertainty, independent of any specific threshold.

B.2 Entropy Across Token Categories

To verify that this uncertainty is unique to connectives and not a generic property of all functional tokens, we computed the High Entropy Rate (R_{HE}) for various token categories across a broad sweep of τ thresholds.

As shown in Table 8, logical connectives consistently maintain the highest R_{HE} among all categories across all thresholds. While quantifiers in Phi-4-mini show a relatively high R_{HE} at certain thresholds, they constitute an extremely rare portion of the output ($\sim$0.36% of all tokens) and do not function as inter-step relation operators that direct logical trajectories. These results empirically validate our focus on logical connectives as the primary and most actionable intervention targets for reasoning.

C Experimental Setup

C.1 Logical Steering

For gradient based steering, we utilized the **OpenThought** dataset to extract reasoning vectors. We computed the steering vector from the activations of the **penultimate layer** (the second to last layer) of the model. This layer was chosen to capture high level semantic reasoning features before they are projected into the vocabulary space. During inference, we applied the steering vector with an injection coefficient of $\alpha = 0.5$ across all models.

C.2 Logical Branching

The branching mechanism was configured with a lookahead depth of $L = 20$. To ensure computational efficiency, we trigger branching only at steps where the greedy top 1 token is itself a logical connective, i.e., $w_{\text{top1}} \in \mathcal{S}_l$. Even then, we apply an additional ambiguity filter: branching is activated only when at least two candidates from our predefined connective set $\mathcal{S}_l$ appear within the top 20 token predictions. This two-stage criterion ensures that branching is performed only at genuine decision points, connective pivots where the model exhibits structural uncertainty, while avoiding unnecessary lookahead on non-connective or low ambiguity steps.

C.3 Targeted Transition Preference Optimization (TTPO)

C.3.1 Data Construction

To train the TTPO objective, we constructed a pairwise preference dataset focused on logical transitions. We first split the **ZebraLogic** dataset into training and test sets with an 8:2 ratio. Using the

	Tail Quantile (q)	Base Conn. (%)	Tail Conn. (%)	Enrichment
Gemma-3-4b-it	0.90	4.05	18.42	4.55×
Gemma-3-4b-it	0.95	4.05	22.85	5.64×
Gemma-3-4b-it	0.99	4.05	23.37	5.77×
Phi-4-mini-instruct	0.90	6.73	18.58	2.76×
Phi-4-mini-instruct	0.95	6.73	18.17	2.70×
Phi-4-mini-instruct	0.99	6.73	13.54	2.01×

Table 7: Threshold free quantile enrichment of logical connectives in the high-entropy tails.

Threshold (τ)	Conn.	Non-conn.	Negation	Quantifier	Number	Punct.
0.5	64.5%	17.7%	25.1%	13.7%	5.8%	15.5%
1.0 (Default)	40.9%	7.1%	12.3%	6.9%	1.0%	3.7%
1.5	21.0%	2.7%	3.5%	1.7%	0.1%	0.6%
2.0	6.6%	0.9%	0.9%	0.4%	0.0%	0.1%

Threshold (τ)	Conn.	Non-conn.	Negation	Quantifier	Number	Punct.
0.5	70.1%	33.9%	41.1%	53.5%	32.8%	40.8%
1.0 (Default)	56.8%	23.0%	34.5%	42.8%	20.6%	25.3%
1.5	43.0%	14.6%	26.1%	30.3%	11.3%	12.0%
2.0	27.9%	8.8%	16.3%	22.3%	4.3%	5.1%

Table 8: High Entropy Rate (R_{HE}) comparison across different token categories and τ thresholds. Logical connectives (Conn.) consistently show the highest structural uncertainty.

Algorithm 1 TTPO Dataset Construction

Require: Training set $\mathcal{D}_{\text{train}}$, connective pool $\mathcal{S}_l$, model $\mathcal{M}$
Ensure: Preference pairs $\mathcal{D}_{\text{pair}}$

$\mathcal{D}_{\text{pair}} \leftarrow \emptyset$
for all $(x, y_{\text{gold}}) \in \mathcal{D}_{\text{train}}$ **do**
  $t \leftarrow \text{FINDPIVOT}(\mathcal{M}, x)$
  $w_{\text{greedy}} \leftarrow \text{GREEDYTOKEN}(\mathcal{M}, x_{\leq t})$
  $C \leftarrow \{w_{\text{greedy}}\} \cup \text{SAMPLE}(\mathcal{S}_l, 5)$
  $\mathcal{P}_{\text{pos}} \leftarrow \emptyset$; $\mathcal{P}_{\text{neg}} \leftarrow \emptyset$
  for all $c \in C$ **do**
    $y_{\text{gen}} \leftarrow \text{GREEDYDECODE}(\mathcal{M}, x_{\leq t} \| c)$
    if $\text{CHECKANSWER}(y_{\text{gen}}, y_{\text{gold}})$ **then**
      $\mathcal{P}_{\text{pos}} \leftarrow \mathcal{P}_{\text{pos}} \cup \{c\}$
    else
      $\mathcal{P}_{\text{neg}} \leftarrow \mathcal{P}_{\text{neg}} \cup \{c\}$
    end if
  end for
  if $\mathcal{P}_{\text{pos}} \neq \emptyset$ & $\mathcal{P}_{\text{neg}} \neq \emptyset$ **then**
    $w_c \leftarrow \text{SELECT}(\mathcal{P}_{\text{pos}})$; $w_r \leftarrow \text{SELECT}(\mathcal{P}_{\text{neg}})$
    $\mathcal{D}_{\text{pair}} \leftarrow \mathcal{D}_{\text{pair}} \cup \{(x, w_c, w_r)\}$
  end if
end for

training split, we performed inference to identify logical pivot points.

The data construction process is detailed in Algorithm 1. When the model encounters a logical connective during generation, we create branches using the original greedy token and 5 randomly sampled connectives from $\mathcal{S}_l$. For each branch, we continue generation using greedy decoding until the EOS token. We then verify the correctness of each generated path. If we identify a pair where one transition leads to the correct answer (w_c) and another leads to an incorrect answer (w_r), we form a preference pair (x, w_c, w_r). This dataset is also utilized for the discriminative analysis in Section 6.2.

C.3.2 Training Configuration

We fine tuned the models using the constructed preference pairs for 3 epochs with a batch size of 1. To prevent catastrophic forgetting and ensure stable convergence on the specific transition tokens, we used a low learning rate: 1×10^{-6} for the 4B parameters models and 1×10^{-7} for the 13B models.

C.4 Benchmark

We summarize benchmark statistics in Table 10. For ZebraLogic, the training split is used exclusively for TTPO preference pair data construction (Section 4.3); it is not used for reporting evaluation results.

D Prompt Template

D.1 Rulebert-Union

Rulebert-Union Prompt

```
[System Instruction]
You are an expert in logical reasoning and
reading comprehension. Your task solve
questions.

Follow these steps strictly:
1. Reasoning step by step.
2. Select the answer in the format '/boxed{
ANSWER}'. for example, if the answer is
option A, the output should be '/boxed{A}'
```

Logical Relations	Relation Connectives
Conjunction	as well as, as well, also, separately
Alternative	either, instead, alternatively, else, neither
Restatement	specifically, particularly, in particular, besides, additionally, in addition, moreover, furthermore, plus, not only, indeed, in other words, in fact, in short, in the end, overall, in summary, in details
Instantiation	for example, for instance, such as, including, as an example, an as instance, for one thing
Contrast	but, however, yet, while, unlike, rather, rather than, in comparison, by comparison, on the other hand, on the contrary, contrary to, in contrast, by contrast, whereas, conversely
Concession	although, though, despite, despite of, in spite of, regardless, regardless of, nevertheless, nonetheless, even if, even though, even as, even when, even after, even so, no matter
Analogy	likewise, similarly, as if, as though, just as, just like, namely
Temporal	during, before, after, when, as soon as, then, next, until, till, meanwhile, in turn, meantime, afterwards, simultaneously, at the same time, beforehand, previously, earlier, later, thereafter, finally, ultimately
Condition	if, as long as, unless, otherwise, except, whenever, whichever, once, only if, only when, depend on
Causal	because, cause, as a result, result in, due to, therefore, hence, thus, thereby, since, now that, consequently, in consequence, in order to, so as to, so that, why, for, accordingly, given, turn out

Table 9: Taxonomy of logical relations and examples of logical connectives included in $\mathcal{S}_l$. This set acts as the intervention target.

Dataset	Train	Test	Task
ZebraLogic	2700	500	Multi Class
BIG Bench Hard	–	1000	Multi Class
RuleBERT-Union	–	1000	Binary Class
LogiQA 2.0	–	1000	Binary Class
ProntoQA	–	500	Binary Class

Table 10: Benchmark datasets and evaluation splits used

```
[User Prompt]
# Context:
[context]

# Question:
[question]

# Options:
A. True
B. False

Think step by step.
```

Figure 7: Rulebert-Union prompt template

D.2 Logi QA 2.0

Logi QA 2.0 Prompt

```
[System Instruction]
You are an expert in logical reasoning and
reading comprehension. Your task solve
questions.

Follow these steps strictly:
1. Reasoning step by step.
2. Select the answer in the format '/boxed{
ANSWER}'. for example, if the answer is
option A, the output should be '/boxed{A}'

[User Prompt]
# Hypothesis:
[hypothesis]

# Question:
[question]

# Options:
A. not-entailment
B. entailment

Think step by step.
```

Figure 8: Logi QA 2.0 prompt template

D.3 ProntoQA

ProntoQA Prompt

```
[System Instruction]
You are an expert in logical reasoning and
reading comprehension. Your task solve
questions.

Follow these steps strictly:
1. Reasoning step by step.
2. Output the answer in the format '/boxed
{}' ANSWER is one of /boxed{A}, /boxed{B}
```

```
[User Prompt]
# Context:
[context]

# Question:
[question]

# Options:
A. True
B. False

Think step by step.
```

Figure 9: ProntoQA prompt template

D.4 ZebraLogic

ZebraLogic Prompt

```
[System Instruction]
You are an expert at solving puzzle problems.
 Follow these rules strictly:
1. Solve and think step by step.
2. Do not explain anything else.
3. Give the final answer only inside /boxed
{}.
- Example :
# Puzzle
...

# Question
...

# Choices
[
"Eric",
"Bob",
"Alice",
"Peter",
"Carol",
"Arnold"
]

- Answer example
# Reasoning
...

# Answer
/boxed{Bob}

[User Prompt]
# Puzzle
[puzzle]

# Question
[question]

# Choices
[choices]

Think step by step.
```

Figure 10: ZebraLogic prompt template

D.5 BIG-Bench Hard (deductive subset)

BIG-Bench Hard (deductive subset) Prompt

```
[System Instruction]
You are an expert in logical reasoning and
reading comprehension. Your task solve
questions.

Follow these steps strictly:
1. Reasoning step by step.
2. Select the answer in the format '/boxed{
ANSWER}'. for example, if the answer is
option A, the output should be '/boxed{A}'

[User Prompt]
# Context:
[context]

# Question:
[question]

# Choices:
[choices]

Think step by step.
```

Figure 11: Big Bench Hard (deductive subset) prompt template

E Efficiency Comparison

We report an efficiency comparison between decoding baselines and our connective centric methods. All efficiency numbers are measured on the *ZebraLogic test split* with Gemma-3-4b-it model and are *normalized by greedy decoding* for each model (Greedy $= 1.00$), so values > 1 indicate additional compute/latency relative to greedy.

Token cost ($\times$ Greedy). Token cost counts the total number of next token forward steps required by each method, aggregated over the test set and normalized by greedy. This includes any extra forward passes introduced by the decoding algorithm, e.g., multiple sampled trajectories for *Self-Consistency* ($n = 5$), multiple hypotheses maintained for *Beam Search* (beam size as in Table 4), and additional lookahead generation for *Branching* (triggered at connective pivots with hyperparameters K and lookahead length L). In contrast, Steering and TTPO preserve single trajectory greedy decoding at inference time; their token cost deviations from 1.0 mainly reflect changes in the generated length, not additional search.

Time ($\times$ Greedy). Time is measured as *wall-clock latency* from the start of generation to termination (EOS or max length), also aggregated over

Gemma-3-4b-it					
Dataset	Greedy	Beam	Self-Cons. (n=5)	Ours (best)	Ours method
ZebraLogic	38.8	33.4	39.6	**42.0**	Branching
BBH (Ded.)	75.3	**77.0**	75.5	75.9	TTPO
RuleBERT	60.3	**67.7**	63.9	63.4	Branching
LogiQA 2.0	55.2	54.0	55.3	**56.8**	Branching
ProntoQA	90.0	90.6	**91.8**	90.4	TTPO
Avg.	63.9	64.5	65.2	**65.7**	–
Phi-4-mini-instruct (4B)					
Dataset	Greedy	Beam	Self-Cons. (n=5)	Ours (best)	Ours method
ZebraLogic	38.8	31.8	**41.0**	39.6	Steering
BBH (Ded.)	67.3	**73.8**	73.6	69.1	Branching
RuleBERT	50.3	51.7	48.8	**51.9**	Branching
LogiQA 2.0	57.7	59.0	58.8	**59.4**	TTPO
ProntoQA	93.6	93.2	**96.8**	96.6	TTPO
Avg.	61.5	61.9	**63.8**	63.3	–

Table 11: Accuracy comparison on 4B models. Self-consistency uses n=5 samples. Beam uses the same beam size as in Table 4.

Efficiency (normalized by Greedy)			
Method	Token cost (×)	Time (×)	Main hyperparams
Gemma-3-4b-it (4B)			
Greedy	1.00	1.00	–
Beam Search	3.32	1.88	beam=5
Self-Consistency	2.90	2.80	$n = 5$
Steering	0.93	0.99	$\alpha = 0.5$
TTPO	1.26	1.12	–
Branching	1.18	1.45	K=20, L=20
Phi-4-mini-instruct (4B)			
Greedy	1.00	1.00	–
Beam Search	5.61	1.75	beam=5
Self-Consistency	4.37	4.09	$n = 5$
Steering	0.89	1.00	$\alpha = 0.5$
TTPO	0.89	0.93	–
Branching	1.02	1.53	K=20, L=20

Table 12: Efficiency for 4B models. Token-cost counts the total number of next-token forward steps (including lookahead branches) normalized by greedy. Time is measured wall-clock latency normalized by greedy under the same hardware and decoding setup.

the ZebraLogic test set and normalized by greedy.

Hyperparameters. For reference, we summarize the main decoding hyperparameters in Table 12: beam size for beam search, n for self-consistency, α for steering, and (K, L) for branching. These knobs directly control the amount of additional inference time computation for search based methods, while TTPO modifies model behavior through training and does not introduce inference time search.