STATe-of-Thoughts: Structured Action Templates for Tree-of-Thoughts

Zachary E. Bamberger[1,*] **Till R. Saenger**[2,*] **Gilad Morad**[3]
Ofra Amir[1] **Brandon M. Stewart**[2] **Amir Feder**[4]

[1]Technion [2]Princeton University [3] Independent [4]Hebrew University

[*]Equal contribution

Abstract

Inference-Time-Compute (ITC) methods like Best-of-n and Tree-of-Thoughts are meant to produce output candidates that are both high-quality and diverse, but their use of high-temperature sampling often fails to achieve meaningful output diversity. Moreover, existing ITC methods offer limited control over *how* to perform reasoning, which in turn limits their interpretability. We present **STATe-of-Thoughts** (STATe), an interpretable ITC method that *searches* over high-level reasoning patterns. STATe branches over discrete and interpretable textual interventions rather than over token-level samples: a *controller* selects actions encoding high-level reasoning choices; a *generator* produces reasoning steps conditioned on those choices; and an *evaluator* scores candidates to guide search. This structured approach yields three main advantages. First, action-guided textual interventions reliably influence LLM generations and produce greater response diversity than temperature-based sampling. Second, in a case study on argument generation, STATe's explicit action sequences capture interpretable features that are highly predictive of output quality. Third, estimating the association between performance and action choices allows us to identify promising yet unexplored regions of the action space and steer generation toward them. STATe is most useful when a task admits multiple solutions and when understanding *why* an output succeeds matters beyond *whether* the output succeeds. Together, these results establish STATe as both a practical framework for diverse and controllable text generation, and as a tool for understanding the reasoning patterns that drive performance.

1 Introduction

Many applications of LLMs require more than generating high-quality responses: they need systematic and interpretable control over how text is produced. For example, in subjective tasks like persuasive writing, researchers vary the rhetorical structure and content themes of arguments to study the features that drive belief change (Tan et al., 2016; Saenger et al., 2024; Salvi et al., 2025; Hackenburg et al., 2025a; Costello et al., 2026). Similarly, in creative writing, researchers are concerned with generating diverse yet high-quality outputs that satisfy the preferences of the audience (Doshi & Hauser, 2024; Lee & Chung, 2024; Xu et al., 2025). In both settings, the challenge is to produce text that varies systematically along dimensions of interest while maintaining coherence and quality. Both settings admit multiple high-quality responses, and in both, *why* a response succeeds matters beyond just *whether* it does.

ITC methods address part of this challenge by allocating additional compute for LLM reasoning (Wei et al., 2022; Kojima et al., 2022) and for producing multiple candidate responses (Brown et al., 2020; Stiennon et al., 2020; Wang et al., 2023). Tree-based methods (Beeching et al., 2024; Hao et al., 2024; 2023), like Yao et al. (2023a)'s Tree of Thoughts (ToT), further enhance quality by branching on intermediate thoughts and pruning less-promising reasoning trajectories. However, these methods rely primarily on temperature-

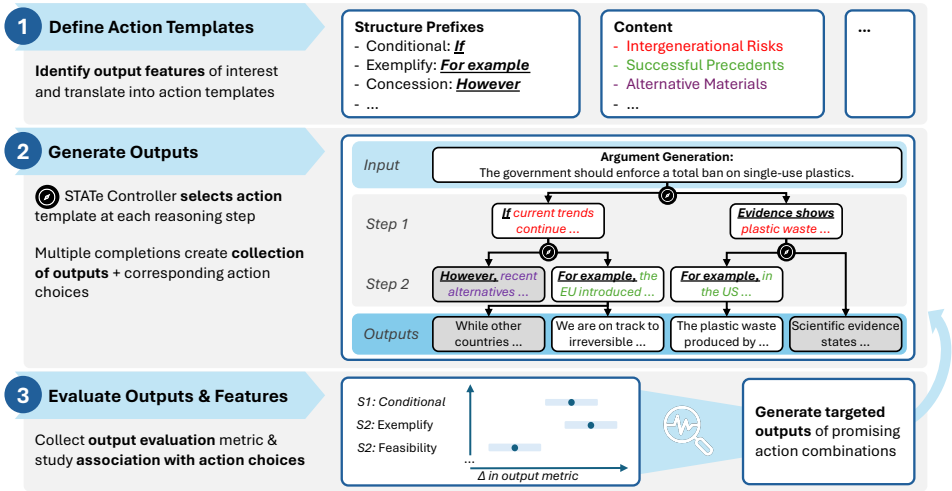

Figure 1: **STATe for argument generation.** Tasked with generating persuasive arguments in favor of banning single-use plastics, STATe's workflow involves the following steps: (1) Define action templates that control output features of interest, such as structural prefixes and content themes. (2) Generate outputs via tree search (Grey nodes indicate pruned branches; the rightmost path illustrates *early stopping* after a single step). (3) Evaluate outputs on a downstream metric, and study associations between action choices and performance.

based sampling for diversity, which yields limited meaningful variation (Zhang et al., 2025c; Jiang et al., 2025). Moreover, since ITC methods sample at the token-level, decisions about what to say and how to say it remain implicit in the decoding process (Holtzman et al., 2020; Xie et al., 2020). As a result, they provide limited control over *which* decisions are explored and limited insight into which decision patterns drive success or failure.

To induce interpretable yet diverse sampling, we prepend prefixes to each LLM completion. Specifically, we define discrete *action templates* that encode high-level reasoning choices (such as which rhetorical structure to employ, which content theme to develop, or which writing operation to perform). We use intervention-based sampling to build **STATe-of-Thoughts** (STATe), an inference-time compute framework that searches over sequences of high-level reasoning actions. STATe's *controller* selects which actions to explore at each reasoning step, and then its *generator* produces reasoning steps conditioned on the selected actions.[1] An *evaluator* scores both intermediate and final states to guide beam search (Beeching et al., 2024; Hao et al., 2024). We illustrate STATe's three-step workflow in Figure 1 through the lens of an argument generation task.

We compare STATe to existing ITC methods in both creative writing and argument generation. On NoveltyBench (Zhang et al., 2025c) (Section 4.1), we find that STATe's branching mechanism produces outputs that are both more diverse and of higher quality than standard ITC branching. In our case study on argument generation (Section 4.2), we find that textual interventions reliably manifest in the generated reasoning steps and responses (Section 4.2.1), that sequential action features are highly predictive of argument quality on held-out data (Section 4.2.2), and that model-guided trajectory selection allows for generating high-quality outputs from promising yet unexplored regions of the action space (Section 4.2.4). Our work is open-sourced (github repo) and provides the following contributions:

1. A controllable ITC framework for action-space search.
2. A diversity mechanism beyond high-temperature sampling.
3. An action-based framework for analyzing the quality of reasoning patterns.

[1]Unlike latent interventions (Anthropic, 2024; Durmus et al., 2024b; Anthropic, 2025b; Feldman et al., 2026), interventions in STATe are explicit text prefixes and thus directly auditable.

2 Background

2.1 Inference-Time Compute

The Input-Output (I/O) approach to using LLMs applies an input sequence (prompt) x to model p_θ and produces output sequence y: $G_{I/O}(x) \rightarrow y$. While effective for many tasks, this approach exhibits limited robustness to common failure modes such as hallucinations (Simhi et al., 2024; Orgad et al., 2025; Simhi et al., 2025), sycophancy (Sharma et al., 2024), and other biases (Itzhak et al., 2024; Orgad & Belinkov, 2023). Building on the intuition that human reasoning benefits from more "time to think" (Kahneman, 2011), ITC methods provide LLMs with additional "reasoning" tokens (Pfau et al., 2024) to scale reasoning depth (Appendix A.2.1), and permit parallel reasoning attempts to scale reasoning breadth (Appendix A.2.2).

Chain-of-Thought (CoT) reasoning (Wei et al., 2022; Kojima et al., 2022) scales *depth* by enabling models to generate intermediate reasoning steps before arriving at a final answer. Formally, we define CoT as $G_{CoT}(x) \rightarrow Z, y$, where Z is the chain of reasoning steps and y the final answer. While CoT improves performance on many reasoning tasks (Sprague et al., 2025; DeepSeek-AI et al., 2025; OpenAI et al., 2024), errors can propagate through the reasoning chain, and there is no principled mechanism to revisit decisions or explore alternative strategies. Instead of scaling depth, Best-of-n methods (Brown et al., 2020; Stiennon et al., 2020) scale *breadth* by generating n independent candidate outputs and selecting the best according to some criterion. This enhances robustness by reducing the impact of individual generation failures. We refer to sampling more than one completion from an LLM as *branching* (Yao et al., 2023a). In Best-of-n methods, branching produces multiple complete reasoning chains along with their associated answers: $G_{CoT}(x; n, \texttt{temp}) \rightarrow \{(Z^1, y^1), \ldots, (Z^n, y^n)\}$, where each Z^j represents a complete reasoning chain and y^j is its corresponding final answer. Best-of-n methods typically branch only at the initial reasoning step, without principled exploration of intermediate reasoning decisions. Moreover, inducing diversity across branches through high-temperature sampling often yields homogeneous outputs or degrades quality (Minh et al., 2025; Jiang et al., 2025; Zhang et al., 2025c;a).

2.2 Tree of Thoughts

Tree of Thoughts (ToT) (Yao et al., 2023a) combines both ITC strategies: improving reasoning quality by scaling depth, and enhancing robustness by scaling breadth. ToT methods (Appendix A.2.3) reframe LLM generation as a search problem over a tree of partial reasoning steps. ToT branches at each reasoning step, evaluates the quality of each branch, and prunes unpromising paths. Each node contains a state $s_i := [x, Z_i]$ that captures a partial solution with the input (x) and reasoning steps so far ($Z_{i-1} := [z_1, \ldots, z_{i-1}]$).[2] A leaf node s_{d+1} represents a complete solution $[x, Z_d, y]$ where y is the final answer and d is the predefined maximum reasoning depth. Formally, at step i, we sample candidate reasoning steps $\{z_i^1, \ldots, z_i^n\} \sim G_{ToT}(z_i)$, where $G_{ToT}(z_i) := p_\theta(z_i \mid x, Z_{i-1}; n, \texttt{temp})$. In practice, ToT often implements both intermediate and final evaluation through LLM-as-a-Judge (Zheng et al., 2023; Li et al., 2023). Process Reward Models (PRMs) (Yao et al., 2023a; Lightman et al., 2024; Wang et al., 2025) score partial trajectories to prune low-value branches and prevent exponential tree growth: $V(Z_i \mid x) \rightarrow [0, 1], i \leq d$. Conversely, Outcome Reward Models (ORMs) (Zheng et al., 2023; Kim et al., 2024a;b) score completed outputs to select the best final answer: $V(y \mid x) \rightarrow [0, 1]$.

Traditional ToT implementations face two primary limitations. First, sampling at high temperatures fails to promote diversity, since branches tend to cluster around similar content (Jiang et al., 2025; Zhang et al., 2025c). Second, ToT implementations perform a predetermined number of reasoning steps, which can lead to "overthinking" (Sprague et al., 2025; Liu et al., 2025a; Muennighoff et al., 2025; Hong et al., 2025) or insufficient reasoning.

[2] For ease of notation, we denote the reasoning steps $z_1, \ldots, z_i$ by Z_i, but treat s_i as a flat vector of inputs (x), reasoning steps ($z_1, \ldots, z_i$), and optionally final outputs (y), not a nested vector.

3 Methods

STATe replaces ToT's stochastic temperature sampling with discrete action templates that diversify branches in tree search. This allows each branch to explore fundamentally different reasoning strategies from its neighbors and enables "early stopping" (producing a final answer before depth d) if the reasoning so far is sufficient. Moreover, STATe tracks actions along a trajectory, enabling researchers to study associations between controllable, concept-level interventions and downstream outcomes (Goyal et al., 2019; Abraham et al., 2022).

3.1 STATe Components

At each layer i, STATe starts with a list of states, each of the form $s_i = [x, Z_i]$. The controller selects n interventions for each state in the frontier. The generator then produces completions that extend each of these interventions. Finally, the evaluator scores the resulting trajectories, and retains the top-k states for the next layer. We present the full process in Algorithm 1 and discuss its computational complexity in Appendix D.

Controller (C): We treat each action as a *tool call*. Selecting an action corresponds to choosing a tool name from a fixed set of action templates (Appendix H) and providing values for the tool's arguments. Given a parent state $s_{i-1} = [x, Z_{i-1}]$ representing the input and reasoning so far, the controller must choose up to n actions from the action space $\mathcal{A}$ to explore in parallel branches (for guidance on how to build an action space, see Appendix H.3). Formally, we define the controller output as: $\{a_i^1, \ldots, a_i^n\} = \mathrm{C}(s_{i-1}, \mathcal{A}, n)$. Implicitly, the controller implements a scoring function $Q(s_{i-1}, a_i)$ that estimates the value of taking action a_i from state s_{i-1} such that $\{a_i^1, \ldots, a_i^n\} = \arg\max_{A \subset \mathcal{A}, |A|=n} \sum_{a_i \in A} Q(s_{i-1}, a_i)$. If the controller determines that reasoning is sufficient, it selects a dedicated `FINISH` action, signifying that the generator should produce a final answer. This mechanism helps prevent "overthinking" where additional steps become degenerate after the model has effectively converged (Liu et al., 2025a; Ringel et al., 2025; Sui et al., 2025; Hong et al., 2025; Muennighoff et al., 2025). We present additional implementation details in Appendix C.1.

```
<thinking>
<step>
...
</step>
...
<step>
## internal_reasoning
I should examine case studies from ...
## claim
For example, California's ban on single-use plastics demonstrates...
```

Figure 2: Task: generate an argument for banning single-use plastics. The controller selects `{"subtopic": "success_of_existing_bans", "structure": "exemplification"}` from the action space in Appendix H.2. Internal reasoning guides the model's next completion, while the prefix forces the model's completion to open with "For example".

Generator (G): For each action $a_i^j \in \{a_i^1, \ldots, a_i^n\}$, we "execute" the corresponding tool to obtain text guidance $a_i^j()$. We append $a_i^j()$ to the state's existing reasoning,[3] and force it to generate text consistent with the chosen action (Figure 2). Formally, given the parent state $s_{i-1} = [x, Z_{i-1}]$, we sample a continuation using the generator:

$$z_i^j \sim G\big(z \mid x, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp}\big)\ [\texttt{:}\ \texttt{</step>}] \tag{1}$$

[3]prefilling (Muennighoff et al., 2025; Bricken et al., 2025) injects text into the assistant message.

for each action a_i^j. We combine each generated thought z_i^j with the current state to create a child state $s_i^j = [s_{i-1}, z_i^j]$. At maximum depth d, or when the controller selects the `FINISH` action, STATe reaches the *synthesis* step, which produces final outputs:

$$y^j \sim G(y \mid x, \texttt{prefill}(Z_{i-1}, \texttt{FINISH}()); \texttt{temp})[\texttt{: \</answer\>}]. \tag{2}$$

We provide additional details on the Generator in Appendix C.2.

Evaluator (V_{PRM} & V_{ORM}) After generating child states from each parent, we evaluate their quality using either score-based LLM-as-a-Judge models (Zheng et al., 2023; Kim et al., 2024a;b; Liu et al., 2026b), or verifiable rewards (Lambert et al., 2025; Gao et al., 2024; Team et al., 2025b). Following the Tree-of-Thoughts framework, we evaluate intermediate states $s_i = [x, Z_i]$ using a PRM, $V_{PRM}(s_i) := V(Z_i|x) \rightarrow [0, 1]$, and complete solution states $s_i = [x, Z_{i-1}, y]$ using an ORM, $V_{ORM}(s_i) := V(y|x) \rightarrow [0, 1]$. Our LLM-based evaluators use custom rubrics that explicitly assess backward compatibility (coherence with prior reasoning steps) and forward compatibility (projected final answer quality) for intermediate reasoning steps, and task-specific criteria such as instruction adherence, coherence, and stylistic appropriateness for final outputs. See additional details in Appendix C.3.

Algorithm 1 STATe-of-Thoughts$(x, G, C, V_{\text{PRM}}, V_{\text{ORM}}, \mathcal{A}, n, k, d, \texttt{temp})$

Require: Input x, generator G, controller C, process evaluator V_{PRM}, outcome evaluator V_{ORM}, action space $\mathcal{A}$, branching factor n, beam width k, depth d, temperature `temp`

Initialize $L_0 \leftarrow \{x\}$ ▷ Initial layer with just the input
Initialize $F \leftarrow \emptyset$ ▷ Collection of final states with answers
for $i = 1$ to $d + 1$ **do**
  $L'_i \leftarrow \emptyset$ ▷ Candidate states for layer i
  for each state $s_{i-1} \in L_{i-1}$ **do**
    $\mathcal{A}_i \leftarrow \{\texttt{FINISH}\}$ if $i = d + 1$, else $C(s_{i-1}, \mathcal{A}, n)$ ▷ Select actions or finish
    for each action $a_i^j \in \mathcal{A}_i$ **do**
      if a_i^j is `FINISH` **then**
        $y^j \sim G(s_{i-1}, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp})[\texttt{: \</answer\>}]$ ▷ Generate response
        $s_i \leftarrow [s_{i-1}, y^j]$ ▷ Create final state
        $F \leftarrow F \cup \{s_i\}$ ▷ Add to collection of final states
      else
        $z_i^j \sim G(s_{i-1}, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp})[\texttt{: \</step\>}]$ ▷ Generate thought
        $s_i \leftarrow [s_{i-1}, z_i^j]$ ▷ Create new intermediate state
        $L'_i \leftarrow L'_i \cup \{s_i\}$ ▷ Add to next layer's candidates
      end if
    end for
  end for
  if $L'_i = \emptyset$ **then break** ▷ All branches terminated via early stopping
  end if
  Score all candidates: $v_{s_i} \leftarrow V_{\text{PRM}}(s_i)$ for all $s_i \in L'_i$
  $L_i \leftarrow \arg\max_{L \subset L'_i, |L| = \min(k, |L'_i|)} \sum_{s_i \in L} v_{s_i}$ ▷ Select top-k states for layer i
end for
Score all final states: $v_s \leftarrow V_{\text{ORM}}(s)$ for all $s \in F$
return $\arg\max_{s \in F} v_s$ ▷ Return highest-scoring final state

3.2 Attributing Outcomes to Controller Actions

A key advantage of STATe-of-Thoughts is its ability to attribute differences in outcomes to specific controller actions, since each branch in the reasoning tree carries a logged action sequence. However, estimating causal effects is complicated by sequential confounding: actions are selected conditional on prior actions in the same sequence. We therefore focus on associational analysis, aiming to identify action patterns that consistently correlate with

better or worse outcomes. Let $\tau = (a_1, a_2, \ldots, a_n)$ denote a complete action sequence. We explore whether the *sequential structure* of actions matters beyond their mere presence.

A **presence-based model** represents actions through binary indicators, $\mathbf{1}_a(\tau) \in \{0,1\}^{|\mathcal{A}|-1}$, to determine whether the action type a appears anywhere in τ, and fits $Y_i = \alpha + \mathbf{1}_a(\tau_i)\boldsymbol{\beta} + \epsilon_i$. Conversely, a **sequential model** extends this with (i) *position features* $\mathbf{1}_{a,k}(\tau)$, indicating whether action a occurs at step k, and (ii) *transition features* $\mathbf{1}_{a \to a'}^{k \to k+1}(\tau) = \mathbf{1}_{a,k}(\tau) \cdot \mathbf{1}_{a',k+1}(\tau)$, capturing consecutive action bigrams. When the action space is multi-dimensional, cross-dimensional interactions at each step can be included as additional features.

4 Experiments

We evaluate STATe in two settings that probe its capacity for diversity, controllability, and interpretability. First, we compare STATe to existing ITC methods on NoveltyBench (Zhang et al., 2025c) to test whether structured interventions improve semantic *diversity* (Section 4.1). Next, in Section 4.2, we use STATe for a case study on argument generation. In Section 4.2.1 we measure the *controllability* of our interventions by the frequency with which they materialize in generated reasoning steps (claims) and final responses (arguments). We then test whether STATe's action sequences improve *predictability* of argument quality scores by LLM-judges (Section 4.2.2), and demonstrate that these scores correlate with human annotations (Section 4.2.3). Finally, we show that learned associations can guide *discovery* of promising regions of the action space (Section 4.2.4).

4.1 Improving Diversity and Quality in Creative Writing

Setup: We evaluate the diversity of STATe's branching mechanism on NoveltyBench (Zhang et al., 2025c), using its curated 100-prompt set for creative writing.[4] Each generation method (I/O, CoT, ToT, and STATe) produces 8 responses per prompt. For STATe, the action space combines two dimensions (Appendix H.1): *personality traits* (following the Big Five model; Goldberg, 1990) and *target audience* (demographic age to appeal to). We report NoveltyBench's *diversity* metric, the number of functional equivalence classes induced by a fine-tuned DeBERTa (He et al., 2021) embedding space across the response set. Since diversity often comes at the cost of *quality*, we also report NoveltyBench's quality score, based on LLM evaluations (Liu et al., 2026b). For ToT and STATe, we isolate and measure the diversity of the branching mechanism by restricting search to shallow trees (d=1).[5] We set n=k=8 and repeat each configuration across 10 random seeds and 3 temperature regimes (low, medium, high). All experiments follow a strict development/evaluation separation to prevent post-hoc selection. We provide additional details and ablation studies in Appendix E.1.

Results: STATe improves both the semantic diversity of responses and their perceived quality across all three temperature regimes (Table 1). Relative to the best non-STATe baseline (CoT with action space), STATe improves diversity by 42% at T=0.5 (4.24 vs. 2.98), 37% at T=0.7 (4.57 vs. 3.33), and 31% at T=1.0 (4.94 vs. 3.76). While diversity often comes at the cost of quality, STATe's intervention-based branching mechanism also outperforms the strongest baseline in quality: 30% gains at T=0.5 (3.36 vs. 2.59), 21% at T=0.7 (3.52 vs. 2.90), and 16% (3.73 vs. 3.23) at T=1.0. With STATe, Qwen-3-30B-A3B-Instruct comes closest to reaching human performance on NoveltyBench (accessible here) in diversity (5.58) and quality (4.37). Neither ToT nor the inclusion of actions in the prompt, in isolation, matches STATe's performance, suggesting that prefix-based interventions provide a meaningful boost. In Appendix E.1.1 we demonstrate that STATe's performance on NoveltyBench generalizes over 7 models from 4 families: Qwen3 (Yang et al., 2025), Gemma-3 (Team et al., 2025a), Nemotron-3 (NVIDIA et al., 2025), and Ministral-3 (Liu et al., 2026a).

[4] We validate our setup on 10 prompts and test on the remaining 90 (see Appendix E.1).

[5] Deeper heuristic search optimizes for evaluator-aligned scores rather than frontier diversity. Moreover, deeper trajectories often share parent states, introducing overlapping reasoning.

Method	T=0.5		T=0.7		T=1.0	
	Diversity	**Quality**	**Diversity**	**Quality**	**Diversity**	**Quality**
I/O	1.68 ± 0.05	1.67 ± 0.05	1.98 ± 0.03	1.9 ± 0.04	2.41 ± 0.05	2.25 ± 0.05
CoT	2.31 ± 0.06	2.13 ± 0.06	2.59 ± 0.09	2.31 ± 0.08	3.0 ± 0.1	2.66 ± 0.11
I/O w/ Actions	1.94 ± 0.05	1.69 ± 0.04	2.26 ± 0.1	1.91 ± 0.1	2.84 ± 0.09	2.37 ± 0.09
CoT w/ Actions	2.98 ± 0.09	2.59 ± 0.08	3.33 ± 0.12	2.9 ± 0.1	3.76 ± 0.1	3.23 ± 0.1
ToT	1.97 ± 0.05	1.72 ± 0.06	2.27 ± 0.05	1.99 ± 0.06	2.78 ± 0.11	2.4 ± 0.08
ToT w/ Actions	2.38 ± 0.06	1.99 ± 0.06	2.76 ± 0.08	2.32 ± 0.06	3.29 ± 0.11	2.7 ± 0.12
STATe of Thoughts	$\mathbf{4.24 \pm 0.11}$	$\mathbf{3.36 \pm 0.09}$	$\mathbf{4.57 \pm 0.13}$	$\mathbf{3.52 \pm 0.08}$	$\mathbf{4.94 \pm 0.1}$	$\mathbf{3.73 \pm 0.09}$

Table 1: NoveltyBench diversity and quality for Qwen3-30B across ITC methods and temperatures (mean$\pm$ std over 10 seeds). Best performance in **bold**, runner-up underlined.

4.2 Analyzing What Makes an Argument Effective

We conduct a case study on argument generation, in which an LLM must produce an argument in favor of a provided topic. For our action space, we instantiate two dimensions suggested by Wachsmuth et al. (2017): content (subtopics to discuss) and structure (discourse relations; Prasad et al., 2008; Webber et al., 2019), detailed in Appendix H.2.

4.2.1 Granular Control of Argumentative Reasoning

Setup: We generate 1,000 arguments with STATe on a fixed topic and use an LLM as a Judge (GPT-5-mini; Singh et al., 2025) to verify whether interventions materialize in individual reasoning steps (claims) and in final responses (arguments). At the *step-level* we verify for each step whether (i) it exhibits its prescribed discourse structure and (ii) it discusses its prescribed subtopic. At the *response-level* we verify whether the argument reflects each step's prescribed structure and subtopic, and whether the prescribed ordering across steps is preserved (see prompt templates in Appendix E.2).

Results: We find that controller interventions reliably manifest in the LLM's generated text (Table 2). Structure adherence at the step level is near-perfect (99.7%), confirming that the prefix mechanism reliably controls the discourse structure of the reasoning step. Subtopic adherence at the step level is strong but lower (87.8%), reflecting that content guidance operates through text-based guidance rather than explicit prefilling. Response-level structure (96.2%) and subtopic (93.5%) pass rates confirm that prescribed properties propagate through response synthesis. Moreover, the order of subtopics and structural decisions is mostly preserved (87.9%). In Appendix E.2 we discuss the impact of the Generator's synthesis prompt (Appendix C.2) on the faithfulness of interventions.

Check Category	N	Pass Rate (%)	95% CI
Step structure	3,000	99.7	[99.5, 99.9]
Step subtopic	3,000	87.8	[86.6, 89.0]
Final structure	3,000	96.2	[95.5, 96.9]
Final subtopic	3,000	93.5	[92.6, 94.4]
Order preservation	1,000	87.9	[85.9, 89.9]
Overall (all 13)	**13,000**	**93.8**	**[93.3, 94.4]**

Table 2: Controllability evaluation for strict synthesis (plastic pollution). Pass rates assessed by LLM judge across 13 boolean checks per argument. 95% bootstrap CIs computed by resampling arguments ($B = 10{,}000$).

4.2.2 Predicting the Quality of Arguments through Action Sequences

Setup: We evaluate the quality of arguments across 5 topics with 3 LLM judges (Singh et al., 2025; Google DeepMind, 2026; Anthropic, 2025a) (Table 10). We quantify the quality of arguments through pairwise comparisons that we aggregate into ranks based on Bradley–

Terry scores (Bradley & Terry, 1952). Using STATe with Qwen3-30B-A3B-Instruct, we generate 5,000 arguments from 20 trees (with $d = 3$, $n = 100$, $k = 250$), each initialized with a different random seed. We then fit attribution models (Section 3.2) that map controller-action trajectories to final argument quality. Our simplest model (M0) only captures argument length.[6] The presence-only models (M1a: structure only, M1b: content only, M1c: both) add binary indicators for which actions appeared in the trajectory. The sequential model (M2) additionally encodes step position, within-step content–structure interactions, and cross-step transitions. See Appendix E.3 for additional details.

Results: We apply the attribution framework of Section 3.2 to reasoning trajectories for argument generation (Section 4.2). Across all topics and judges, the sequential model (M2) substantially outperforms the presence-based baseline (M1a-c) in predicting the effectiveness of arguments out-of-sample (Figure 3). In other words, predictive information comes not only from which actions the controller takes, but also from when it takes them and how they combine within a step. We present additional experimental details and ablations in Appendix E.3.

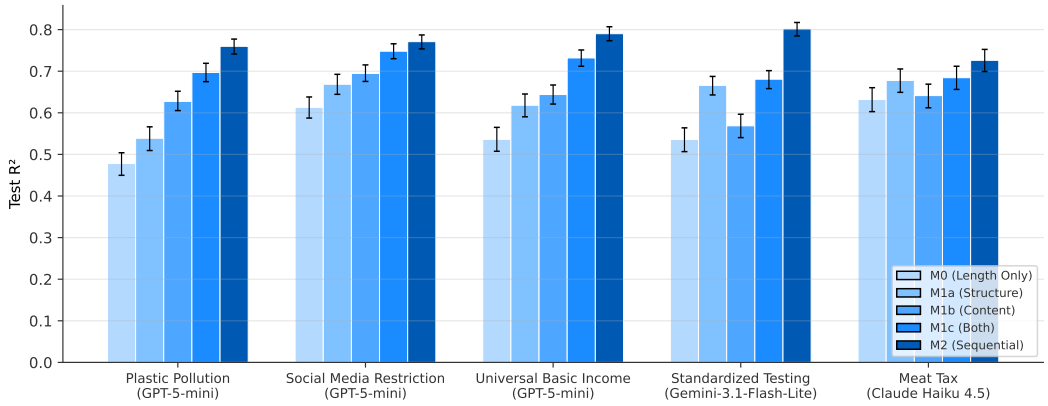

Figure 3: Predictability of argument quality from controller actions across argument topics and LLM judges. Each panel shows the performance (R^2 including 95% bootstrap CIs) on the held-out test set (40% of data).

4.2.3 *Validating LLM-as-a-Judge with Human Preferences*

Setup: The attribution results above demonstrate the mechanistic finding that action *sequences* predict argument quality as measured by an LLM judge. This finding holds for any consistent preference signal since our machinery is agnostic to the source of the outcome variable. To determine whether the LLM judge also captures meaningful human preferences, we conducted a validation study with 288 Prolific participants (Appendix E.4). The study focused on one topic (single-use plastics ban) and one judge (GPT-5-mini), using arguments drawn from the same pool evaluated in our attribution analysis. Each participant evaluated 10 argument pairs in a forced binary choice, drawn from a pool of 50 pairs. For our evaluation, we draw on the ALT test (Calderon et al., 2025), a leave-one-out procedure that compares how well the LLM judge aligns with the annotator pool relative to each individual human annotator (Section E.4.2).

Results: The ALT test's advantage probability reaches 0.841, meaning the LLM judge matches or exceeds an excluded human's alignment with the remaining pool 84.1% of the time. The LLM agrees with the human majority vote 72.0% of the time, and its choices correlate significantly with human preference rates.[7] For context, human pairwise agreement on this task is only 56.0% (Krippendorff's $\alpha = 0.118$), reflecting the inherent subjectivity of

[6] All attribution models include argument length (number of characters) as a baseline feature since LLM-as-a-Judge is biased towards long responses (Dubois et al., 2024; Saenger et al., 2024).

[7] See Appendix E.4 for more details.

persuasiveness judgments (Rescala et al., 2024). Importantly, even if the LLM judge were poorly aligned with human preferences, the attribution methodology we present would remain valid. STATe learns to predict whatever preference signal is provided, and the same pipeline applies with human respondents as the outcome source. We use an LLM judge because it provides cheap, fast, and reproducible evaluations at scale, but the human study confirms that, for this topic and judge, the LLM signal is well aligned with human judgment. Full study details, including design, metrics, and stratified results, are provided in Appendix E.4.

4.2.4 *Discovering Promising Unexplored Action Sequences*

Setup: We test whether M2's learned coefficients generalize beyond observed trajectories and can guide search toward high-quality, previously unseen regions of the action space. Concretely, we score unseen trajectories with M2, generate targeted arguments from top-ranked trajectories, and compare them against random exploration and simpler topic-presence guidance (M1b). To mitigate length confounding, we evaluate all comparisons on length-matched sets.[8]

Baseline	Win (T) (%)	95% CI	Top-10	Top-100
Random	78.7	[73.9, 83.2]	8/10	78/100
M1b (Topic Presence)	63.3	[57.3, 69.2]	6/10	57/100
Original Top 5%	68.0	[60.9, 74.8]	9/10	68/100

Table 3: Targeted trajectory exploration vs. baselines (N=204–354 length-matched arguments, 5,000 pairwise comparisons each). **Win (T)**: share of pairwise wins by targeted arguments, with 95% bootstrap CIs (B=10,000); see Appendix E.3.3. **Top-10/Top-100**: targeted arguments among the top n by Bradley–Terry score.

Results: In Table 3 we show that targeted arguments substantially outperform the random baseline (78.7% win rate), the topic-presence baseline (63.3% win rate), and the original top 5% baseline (68.0% win rate). This confirms that M2's trajectory rankings identify genuinely promising regions of the action space, more so than a simpler presence-based approach, analogous to a topic model. We provide additional details and results in Appendix E.3.3.

5 Discussion

We developed STATe-of-Thoughts (STATe) as a controllable inference-time compute framework that makes step-level decisions explicit and auditable (Section 3). On NoveltyBench (Section 4.1), STATe not only produces substantially higher semantic diversity but also improves output quality, demonstrating that intervention-based branching can produce diverse candidates without the typical quality degradation associated with high-temperature sampling. Furthermore, STATe opens up ITC as a tool for exploring what makes open-ended writing effective or ineffective. In our controllability study, we find that STATe's interventions reliably manifest in both intermediate reasoning and final outputs (Section 4.2.1). When evaluating predictive power, we show that action *sequences* (not just action presence) improve outcome predictions (Section 4.2.2). Crucially, we also show that these learned associations can be operationalized: by scoring and targeting previously unseen trajectories, STATe can systematically explore under-visited regions of the controllable feature space and surface strong candidates, rather than repeatedly sampling near-duplicates (Section 4.2.4). Taken together, these results position STATe as a practical method to (1) generate diverse yet high-quality texts, (2) understand which writing strategies drive quality, and (3) discover and target promising new strategies.

[8]For each targeted argument, we find the closest-length baseline argument within ± 5 characters, using each baseline argument at most once.

6 Limitations

STATe has several practical limitations. First, our method relies on prefilling for interventions, but modern closed-source APIs (e.g., GPT, Claude, Gemini) do not expose this functionality. Second, our action–outcome analysis is associative rather than causal, as the design introduces sequential confounding that our current attribution models do not address.[9] Third, STATe's interventions strictly involve adding a new reasoning step to an existing trajectory, which limits its expressivity. STATe does not support interventions that affect final output generation, nor does it support interventions that *alter* rather than *extend* existing content. Fourth, the synthesis step that converts reasoning traces into final outputs introduces a control–quality trade-off. Concretely, strict synthesis preserves reasoning faithfulness and enables high predictability but can produce stilted prose, whereas flexible synthesis improves fluency at the cost of faithfulness and predictability. Finally, the framework strictly supports single-turn interactions and does not support external tool-calls (e.g., RAG (Lewis et al., 2020), code execution (Karpas et al., 2022), etc.). As a result, in tasks like argument generation, we cannot ground argumentative claims in retrieved context, nor validate whether claims are hallucinated. We provide an expanded limitations discussion in Appendix F.

7 Future Work

STATe's ability to balance diversity and quality (Section 4.1) and the associations we identify between controller actions and output performance (Section 4.2) motivate a shift from association toward explicit causal claims about how reasoning patterns shape downstream outcomes. We can therefore model action trajectories as sequential treatments, and use randomized interventions to identify per-step causal effects (Appendix G.1). We can then use this framework in large-scale studies that measure belief change and induced actions after exposure to generated arguments to study the causal effects of complex rhetorical strategies (Appendix G.2).

An equally important direction is optimizing STATe itself. First, replacing fixed beam search with more sophisticated tree-search methods such as Monte Carlo Tree Search (Kocsis & Szepesvári, 2006; Coulom, 2006; Browne et al., 2012; Hao et al., 2023; Silver et al., 2016; 2018) could adapt exploration toward high-performing regions of the action space under constrained evaluation budgets (Appendix G.3). Second, weight-based optimization via reinforcement learning (e.g., PPO (Schulman et al., 2017), GRPO (Shao et al., 2024)) could train the controller, generator, or evaluator to improve downstream performance (Appendix G.4). Third, prompt-optimization pipelines like GEPA (Agrawal et al., 2026) could refine the instructions and demonstrations used by each module (Appendix G.5).

8 Ethics Statement

Argument generation systems can be misused to manipulate at scale by generating misleading, deceitful, or otherwise harmful messages. Prior work shows that LLM-generated arguments can affect human beliefs and preferences in public policy (Bai et al., 2025; Hackenburg et al., 2025a), support harmful narratives (e.g., conspiratorial content; Costello et al., 2026), coerce LLMs into performing harmful requests (Zeng et al., 2024), and draft convincing phishing or social engineering messages (Qi et al., 2025; Lynch et al., 2025). STATe is particularly capable of such manipulation, since it can search over a diverse yet high-quality collection of arguments, and identify the one most likely to sway behavioral outcomes. By interacting with or simulating an audience (Park et al., 2023; 2024), STATe can uncover the rhetorical patterns that systematically increase target audience susceptibility. In adversarial hands, such micro-targeting (Salvi et al., 2025) can potentially persuade people to vote against their interests, purchase unsuitable products, or adopt harmful beliefs.

[9] Notably, our inability to make causal claims does not stem from issues with the STATe framework, but rather from how we process trajectories in our study (Section 4.2.2).

However, persuasion is not inherently manipulative; it is the mechanism by which individuals and institutions communicate urgency, build trust, and motivate action. In public health, well-intentioned guidance often fails to account for patients' specific fears or cultural context, and more tailored communication could improve adherence and outcomes (Brown et al., 2024; Hou et al., 2025). Improved persuasion can serve prosocial goals, from increasing vaccine uptake to encouraging charitable giving and democratic participation.

Persuasion attempts, both prosocial and adversarial, will only increase as LLMs become more widely available. Researchers can use STATe to uncover argumentative patterns that are emotionally abusive or associated with misuse, and steer LLMs away from employing them. Rigorous and transparent tools for analyzing persuasion are a prerequisite for defending against its misuse.

9 Acknowledgments

Funded by the European Union (ERC, Convey, 101078158). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work was supported in part by the Israel Science Foundation (grant 3123/25). We thank the Princeton Laboratory for Artificial Intelligence for providing computational resources and the Princeton Data-Driven Social Science Initiative for feedback and support. We also thank OpenAI for providing additional resources through the Researcher Access Program. We are grateful to Omar Khattab for his ongoing support of this project and for reviewing our paper at its early stages. We also thank Justin Grimmer, Yamil Velez, Allen Roushe, Devin Gonier, John Hines, Matthew Salganik, Susan Murphy, Dongling Zeng, Inbal Billie Nahum-Shani, and Queenie Luo for providing helpful comments and feedback. Finally, we appreciate the discussions, support, and feedback of our colleagues at Princeton, Technion, and HUJI.

References

Eldar D Abraham, Karel D'Oosterlinck, Amir Feder, Yair Gat, Atticus Geiger, Christopher Potts, Roi Reichart, and Zhengxuan Wu. CEBaB: Estimating the causal effects of real-world concepts on NLP model behavior. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 17582–17596. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/701ec28790b29a5bc33832b7bdc4c3b6-Paper-Conference.pdf.

Sahar Admoni, Ofra Amir, Assaf Hallak, and Yftah Ziser. Towards Large Language Models with self-consistent natural language explanations, 2025. URL https://arxiv.org/abs/2506.07523.

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alex Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. GEPA: Reflective prompt evolution can outperform reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=RQm2KQTM5r.

Anthropic. Golden Gate Claude. Anthropic News, 2024. URL https://www.anthropic.com/news/golden-gate-claude. Research demo of feature activation steering; accessed 2025-10-17.

Anthropic. Introducing Claude Haiku 4.5. Anthropic News, 2025a. URL https://www.anthropic.com/news/claude-haiku-4-5. Released October 2025.

Anthropic. Tracing the thoughts of a large language model. Anthropic Research, 2025b. URL https://www.anthropic.com/research/tracing-thoughts-language-model.

Hui Bai, Jan G. Voelkel, Shane Muldowney, Johannes C. Eichstaedt, and Robb Willer. LLM-generated messages can persuade humans on policy issues. *Nature Communications*, 16:6037, 2025. doi: 10.1038/s41467-025-61345-5. URL https://doi.org/10.1038/s41467-025-61345-5.

Edward Beeching, Lewis Tunstall, and Sasha Rush. Scaling test-time compute with open models, 2024. URL https://huggingface.co/spaces/HuggingFaceH4/blogpost-scaling-test-time-compute.

Yoav Benjamini and Daniel Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001. ISSN 00905364, 21688966. URL http://www.jstor.org/stable/2674075.

David M. Blei. Probabilistic topic models. *Commun. ACM*, 55(4):77–84, April 2012. ISSN 0001-0782. doi: 10.1145/2133806.2133826. URL https://doi.org/10.1145/2133806.2133826.

Esther Boissin, Thomas H Costello, Daniel Spinoza-Martín, David G Rand, and Gordon Pennycook. Dialogues with large language models reduce conspiracy beliefs even when the AI is perceived as human. *PNAS Nexus*, 4(11):pgaf325, 10 2025. ISSN 2752-6542. doi: 10.1093/pnasnexus/pgaf325. URL https://doi.org/10.1093/pnasnexus/pgaf325.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Simon Martin Breum, Daniel Vædele Egdal, Victor Gram Mortensen, Anders Giovanni Møller, and Luca Maria Aiello. The persuasive power of Large Language Models. *Proceedings of the International AAAI Conference on Web and Social Media*, 18(1):152–163, May 2024. doi: 10.1609/icwsm.v18i1.31304. URL https://ojs.aaai.org/index.php/ICWSM/article/view/31304.

Trenton Bricken, Rowan Wang, Sam Bowman, Euan Ong, Johannes Treutlein, Jeff Wu, Evan Hubinger, and Samuel Marks. Building and evaluating alignment auditing agents. https://alignment.anthropic.com/2025/automated-auditing/, July 2025.

Dan Brown, Adelaida Barrera, Lucas Ibañez, Iván Budassi, Bridie Murphy, Pujen Shrestha, Sebastian Salomon-Ballada, Jorge Kriscovich, and Fernando Torrente. A behaviourally informed chatbot increases vaccination rates in Argentina more than a one-way reminder. *Nature Human Behaviour*, 8:2314–2321, 10 2024. doi: 10.1038/s41562-024-01985-7.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Nitay Calderon, Roi Reichart, and Rotem Dror. The alternative annotator test for LLM-as-a-judge: How to statistically justify replacing human annotators with LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16051–16081, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.782. URL https://aclanthology.org/2025.acl-long.782/.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip J.K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=bx24KpJ4Eb. Survey Certification, Featured Certification.

Tuhin Chakrabarty, Christopher Hidey, Smaranda Muresan, Kathy McKeown, and Alyssa Hwang. AMPERSAND: Argument mining for PERSuAsive oNline discussions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2933–2943, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1291. URL https://aclanthology.org/D19-1291.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for Large Language Models. In *ICML 2024 Workshop on In-Context Learning*, 2024. URL https://openreview.net/forum?id=LjsjHF7nAN.

Thomas H. Costello, Gordon Pennycook, and David G. Rand. Durably reducing conspiracy beliefs through dialogues with AI. *Science*, 385(6714):eadq1814, 2024. doi: 10.1126/science.adq1814. URL https://www.science.org/doi/abs/10.1126/science.adq1814.

Thomas H Costello, Gordon Pennycook, and David G Rand. Just the facts: How dialogues with AI reduce conspiracy beliefs, Feb 2025. URL osf.io/preprints/psyarxiv/h7n8u_v1.

Thomas H Costello, Kellin Pelrine, Matthew Kowal, Antonio A Arechar, Jean-François Godbout, Adam Gleave, David Rand, and Gordon Pennycook. Large language models

can effectively convince people to believe conspiracies. *arXiv preprint arXiv:2601.05050*, 2026.

Rémi Coulom. Efficient Selectivity and Backup Operators in Monte-Carlo Tree Search. In Paolo Ciancarini and H. Jaap van den Herik (eds.), *5th International Conference on Computer and Games*, Turin, Italy, May 2006. URL https://inria.hal.science/inria-00116992.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Sebastian Deri, Jeremie Rappaz, Luca Maria Aiello, and Daniele Quercia. Coloring in the links: Capturing social ties as they are perceived. *Proc. ACM Hum.-Comput. Interact.*, 2(CSCW), November 2018. doi: 10.1145/3274312. URL https://doi.org/10.1145/3274312.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive capabilities of LLMs: an exploration in mathematical problem solving. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

Kefan Dong, Arvind V. Mahankali, and Tengyu Ma. Formal theorem proving by rewarding LLMs to decompose proofs hierarchically. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=D83tiHiNfF.

Anil R. Doshi and Oliver P. Hauser. Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science Advances*, 10(28):eadn5290, 2024. doi: 10.1126/sciadv.adn5290. URL https://www.science.org/doi/abs/10.1126/sciadv.adn5290.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. Length-controlled AlpacaEval: A simple debiasing of automatic evaluators. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=CybBmzWBX0.

Esin Durmus, Faisal Ladhak, and Claire Cardie. The role of pragmatic and discourse context in determining argument impact. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5668–5678, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1568. URL https://aclanthology.org/D19-1568.

Esin Durmus, Liane Lovitt, Alex Tamkin, Stuart Ritchie, Jack Clark, and Deep Ganguli. Measuring the persuasiveness of language models, 2024a. URL https://www.anthropic.com/news/measuring-model-persuasiveness.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, et al. Evaluating feature steering: A case study in mitigating social biases. Anthropic Research, 2024b. URL https://www.anthropic.com/research/evaluating-feature-steering.

Naoki Egami, Christian J. Fong, Justin Grimmer, Margaret E. Roberts, and Brandon M. Stewart. How to make causal inferences using texts. *Science Advances*, 8(42):eabg2652, 2022. doi: 10.1126/sciadv.abg2652. URL https://www.science.org/doi/abs/10.1126/sciadv.abg2652.

Roxanne El Baff, Khalid Al Khatib, Milad Alshomary, Kai Konen, Benno Stein, and Henning Wachsmuth. Improving argument effectiveness across ideologies using instruction-tuned Large Language Models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 4604–4622, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.265. URL https://aclanthology.org/2024.findings-emnlp.265/.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082/.

Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, and Diyi Yang. Causal inference in natural language processing: Estimation, prediction, interpretation and beyond. *Transactions of the Association for Computational Linguistics*, 10:1138–1158, 2022. doi: 10.1162/tacl_a_00511. URL https://aclanthology.org/2022.tacl-1.66/.

Omri Feldman, Amar Venugopal, Jann Spiess, and Amir Feder. Causal effect estimation with latent textual treatments, 2026. URL https://arxiv.org/abs/2602.15730.

Christian Fong and Justin Grimmer. Discovery of treatments from text corpora. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1600–1609, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1151. URL https://aclanthology.org/P16-1151/.

Jingchu Gai, Guanning Zeng, Huaqing Zhang, and Aditi Raghunathan. Differential Smoothing mitigates sharpening and improves LLM reasoning, 2025. URL https://arxiv.org/abs/2511.19942.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective RL reward at training time for LLM reasoning, 2024. URL https://arxiv.org/abs/2410.15115.

Lewis R Goldberg. An alternative "description of personality": The big-five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216–1229, 1990.

Google DeepMind. Gemini 3.1 Flash-Lite — model card. Google DeepMind, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/. Published March 2026.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019.

Justin Grimmer, Margaret E Roberts, and Brandon M Stewart. *Text as data: A new framework for machine learning and the social sciences*. Princeton University Press, 2022.

Melody Y. Guan, Miles Wang, Micah Carroll, Zehao Dou, Annie Y. Wei, Marcus Williams, Benjamin Arnav, Joost Huizinga, Ian Kivlichan, Mia Glaese, Jakub Pachocki, and Bowen Baker. Monitoring monitorability, 2025. URL https://arxiv.org/abs/2512.18311.

Kobi Hackenburg, Ben M. Tappin, Luke Hewitt, Ed Saunders, Sid Black, Hause Lin, Catherine Fist, Helen Margetts, David G. Rand, and Christopher Summerfield. The levers of political persuasion with conversational artificial intelligence. *Science*, 390(6777):eaea3884, 2025a. doi: 10.1126/science.aea3884. URL https://www.science.org/doi/abs/10.1126/science.aea3884.

Kobi Hackenburg, Ben M. Tappin, Paul Röttger, Scott A. Hale, Jonathan Bright, and Helen Margetts. Scaling language model size yields diminishing returns for single-message political persuasion. *Proceedings of the National Academy of Sciences*, 122(10):e2413443122, 2025b. doi: 10.1073/pnas.2413443122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2413443122.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.507. URL https://aclanthology.org/2023.emnlp-main.507/.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with Large Language Models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=b0y6fbSUG0.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Miguel Ángel Hernán, Babette Brumback, and James M Robins. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. *Epidemiology*, 11(5):561–570, 2000.

Christopher Hidey, Elena Musi, Alyssa Hwang, Smaranda Muresan, and Kathy McKeown. Analyzing the semantic types of claims and premises in an online persuasive forum. In Ivan Habernal, Iryna Gurevych, Kevin Ashley, Claire Cardie, Nancy Green, Diane Litman, Georgios Petasis, Chris Reed, Noam Slonim, and Vern Walker (eds.), *Proceedings of the 4th Workshop on Argument Mining*, pp. 11–21, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5102. URL https://aclanthology.org/W17-5102/.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Kelly Hong, Anton Troynikov, and Jeff Huber. Context Rot: How increasing input tokens impacts LLM performance. Technical report, Chroma, July 2025. URL https://research.trychroma.com/context-rot.

Zhiyuan Hou, Zhengdong Wu, Zhiqiang Qu, Liubing Gong, Hui Peng, Mark Jit, Heidi Larson, Jianhong Wu, and Leesa Lin. A vaccine chatbot intervention for parents to improve HPV vaccination uptake among middle school girls: a cluster randomized trial. *Nature Medicine*, 31:1855–1862, 04 2025. doi: 10.1038/s41591-025-03618-6.

Itay Itzhak, Gabriel Stanovsky, Nir Rosenfeld, and Yonatan Belinkov. Instructed to bias: Instruction-tuned language models exhibit emergent cognitive bias. *Transactions of the Association for Computational Linguistics*, 12:771–785, 2024. doi: 10.1162/tacl_a_00673. URL https://aclanthology.org/2024.tacl-1.43/.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial hivemind: The open-ended homogeneity of language models (and beyond). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025. URL https://openreview.net/forum?id=saDOrrnNTz.

Daniel Kahneman. *Thinking, Fast and Slow*. Farrar, Straus and Giroux, New York, first paperback edition, 2011. ISBN 0374533555 9780374533557.

Ehud Karpas, Omri Abend, Yonatan Belinkov, Barak Lenz, Opher Lieber, Nir Ratner, Yoav Shoham, Hofit Bata, Yoav Levine, Kevin Leyton-Brown, Dor Muhlgay, Noam Rozen, Erez Schwartz, Gal Shachaf, Shai Shalev-Shwartz, Amnon Shashua, and Moshe Tenenholtz. MRKL systems: A modular, neuro-symbolic architecture that combines large language models, external knowledge sources and discrete reasoning, 2022. URL https://arxiv.org/abs/2205.00445.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=8euJaTveKw.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. Prometheus 2: An open source language model specialized in evaluating other language models. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 4334–4353, Miami, Florida, USA, November 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou (eds.), *Machine Learning: ECML 2006*, pp. 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-46056-5.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are zero-shot reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. SummaC: Revisiting NLI-based models for inconsistency detection in summarization. *Transactions of the Association for Computational Linguistics*, 10:163–177, 2022. doi: 10.1162/tacl_a_00453. URL https://aclanthology.org/2022.tacl-1.10/.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=i1uGbfHHpH.

Byung Cheol Lee and Jaeyeon Jae Chung. An empirical investigation of the impact of chatgpt on creativity. *Nature Human Behaviour*, 8:1906 – 1914, 2024. URL https://api.semanticscholar.org/CorpusID:271857922.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLPtasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 5 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, Alexandre Sablayrolles, Amélie Héliou, Amos You, Andy Ehrenberg, Andy Lo, Anton Eliseev, Antonia Calvi, Avinash Sooriyarachchi, Baptiste Bout, Baptiste Rozière, Baudouin De Monicault, Clémence Lanfranchi, Corentin Barreau, Cyprien Courtot, Daniele Grattarola, Darius Dabert, Diego de las Casas, Elliot Chane-Sane, Faruk Ahmed, Gabrielle Berrada, Gaëtan Ecrepont, Gauthier Guinet, Georgii Novikov, Guillaume Kunsch, Guillaume Lample, Guillaume Martin, Gunshi Gupta, Jan Ludziejewski, Jason Rute, Joachim Studnia, Jonas Amar, Joséphine Delas, Josselin Somerville Roberts, Karmesh Yadav, Khyathi Chandu, Kush Jain, Laurence Aitchison, Laurent Fainsin, Léonard Blier, Lingxiao Zhao, Louis Martin, Lucile Saulnier, Luyu Gao, Maarten Buyl, Margaret Jennings, Marie Pellat, Mark Prins, Mathieu Poirée, Mathilde Guillaumin, Matthieu Dinot, Matthieu Futeral, Maxime Darrin, Maximilian Augustin, Mia Chiquier, Michel Schimpf, Nathan Grinsztajn, Neha Gupta, Nikhil Raghuraman, Olivier Bousquet, Olivier Duchenne, Patricia Wang, Patrick von Platen, Paul Jacob, Paul Wambergue, Paula Kurylowicz, Pavankumar Reddy Muddireddy, Philomène Chagniot, Pierre Stock, Pravesh Agrawal, Quentin Torroba, Romain Sauvestre, Roman Soletskyi, Rupert Menneer, Sagar Vaze, Samuel Barry, Sanchit Gandhi, Siddhant Waghjale, Siddharth Gandhi, Soham Ghosh, Srijan Mishra, Sumukh Aithal, Szymon Antoniak, Teven Le Scao, Théo Cachet, Theo Simon Sorg, Thibaut Lavril, Thiziri Nait Saada, Thomas Chabal, Thomas Foubert, Thomas Robert, Thomas Wang, Tim Lawson, Tom Bewley, Tom Bewley, Tom Edwards, Umar Jamil, Umberto Tomasini, Valeriia Nemychnikova, Van Phung, Vincent Maladière, Virgile Richard, Wassim Bouaziz, Wen-Ding Li, William Marshall, Xinghui Li, Xinyu Yang, Yassine El Ouahidi, Yihan Wang, Yunhao Tang, and Zaccharie Ramzi. Ministral 3, 2026a. URL https://arxiv.org/abs/2601.08584.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, and Yang Liu. Human-AI curation synergy: Scaling preference data curation via human-guided AI feedback. In *The Fourteenth International*

Conference on Learning Representations, 2026b. URL https://openreview.net/forum?id=ofgxkMLqic.

Ryan Liu, Jiayi Geng, Addison J. Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L. Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=J3gzdbYZxS.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. In *2nd AI for Math Workshop @ ICML 2025*, 2025b. URL https://openreview.net/forum?id=jLpC1zavzn.

Bruce T. Lowerre. *The Harpy Speech Recognition System*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1976.

Aengus Lynch, Benjamin Wright, Caleb Larson, Stuart J. Ritchie, Soren Mindermann, Evan Hubinger, Ethan Perez, and Kevin Troy. Agentic misalignment: How LLMs could be insider threats, 2025. URL https://arxiv.org/abs/2510.05179.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=FBkpCyujtS.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Allen Newell and Herbert A. Simon. Computer science as empirical inquiry: symbols and search. *Commun. ACM*, 19(3):113–126, March 1976. ISSN 0001-0782. doi: 10.1145/360018.360022. URL https://doi.org/10.1145/360018.360022.

NVIDIA, :, Aaron Blakeman, Aaron Grattafiori, Aarti Basant, Abhibha Gupta, Abhinav Khattar, Adi Renduchintala, Aditya Vavre, Akanksha Shukla, Akhiad Bercovich, Aleksander Ficek, Aleksandr Shaposhnikov, Alex Kondratenko, Alexander Bukharin, Alexandre Milesi, Ali Taghibakhshi, Alisa Liu, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amir Klein, Amit Zuker, Amnon Geifman, Amy Shen, Anahita Bhiwandiwalla, Andrew Tao, Anjulie Agrusa, Ankur Verma, Ann Guan, Anubhav Mandarwal, Arham Mehta, Ashwath Aithal, Ashwin Poojary, Asif Ahamed, Asit Mishra, Asma Kuriparambil Thekkumpate, Ayush Dattagupta, Banghua Zhu, Bardiya Sadeghi, Barnaby Simkin, Ben Lanir, Benedikt Schifferer, Besmira Nushi, Bilal Kartal, Bita Darvish Rouhani, Boris Ginsburg, Brandon Norick, Brandon Soubasis, Branislav Kisacanin, Brian Yu, Bryan Catanzaro, Carlo del Mundo, Chantal Hwang, Charles Wang, Cheng-Ping Hsieh, Chenghao Zhang, Chenhan Yu, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christopher Parisien, Collin Neale, Cyril Meurillon, Damon Mosk-Aoyama, Dan Su, Dane Corneil, Daniel Afrimi, Daniel Lo, Daniel Rohrer, Daniel Serebrenik, Daria Gitman, Daria Levy, Darko Stosic, David Mosallanezhad, Deepak Narayanan, Dhruv Nathawani, Dima Rekesh, Dina Yared, Divyanshu Kakwani, Dong Ahn, Duncan Riach, Dusan Stosic, Edgar Minasyan, Edward Lin, Eileen Long, Eileen Peters Long, Elad Segal, Elena Lantz, Ellie Evans, Elliott Ning, Eric Chung, Eric Harper, Eric Tramel, Erick Galinkin, Erik Pounds, Evan Briones, Evelina Bakhturina, Evgeny Tsykunov, Faisal Ladhak, Fay Wang, Fei Jia, Felipe Soares, Feng Chen, Ferenc Galko, Frank Sun, Frankie Siino, Gal Hubara Agam, Ganesh Ajjanagadde, Gantavya Bhatt, Gargi Prasad, George Armstrong, Gerald Shen, Gorkem Batmaz, Grigor Nalbandyan, Haifeng Qian, Harsh Sharma, Hayley Ross, Helen Ngo, Herbert Hum, Herman Sahota, Hexin Wang, Himanshu Soni, Hiren Upadhyay, Huizi Mao, Huy C Nguyen, Huy Q Nguyen, Iain Cunningham, Ido Galil, Ido Shahaf, Igor Gitman, Ilya

Loshchilov, Itamar Schen, Itay Levy, Ivan Moshkov, Izik Golan, Izzy Putterman, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jatin Mitra, Jeffrey Glick, Jenny Chen, Jesse Oliver, Jian Zhang, Jiaqi Zeng, Jie Lou, Jimmy Zhang, Jinhang Choi, Jining Huang, Joey Conway, Joey Guman, John Kamalu, Johnny Greco, Jonathan Cohen, Joseph Jennings, Joyjit Daw, Julien Veron Vialard, Junkeun Yi, Jupinder Parmar, Kai Xu, Kan Zhu, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kevin Shih, Kezhi Kong, Khushi Bhardwaj, Kirthi Shankar, Krishna C. Puvvada, Krzysztof Pawelec, Kumar Anik, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Li Ding, Lizzie Wei, Lucas Liebenwein, Luis Vega, Maanu Grover, Maarten Van Segbroeck, Maer Rodrigues de Melo, Mahdi Nazemi, Makesh Narsimhan Sreedhar, Manoj Kilaru, Maor Ashkenazi, Marc Romeijn, Marcin Chochowski, Mark Cai, Markus Kliegl, Maryam Moosaei, Matt Kulka, Matvei Novikov, Mehrzad Samadi, Melissa Corpuz, Mengru Wang, Meredith Price, Michael Andersch, Michael Boone, Michael Evans, Miguel Martinez, Mikail Khona, Mike Chrzanowski, Minseok Lee, Mohammad Dabbah, Mohammad Shoeybi, Mostofa Patwary, Nabin Mulepati, Najeeb Nabwani, Natalie Hereth, Nave Assaf, Negar Habibi, Neta Zmora, Netanel Haber, Nicola Sessions, Nidhi Bhatia, Nikhil Jukar, Nikki Pope, Nikolai Ludwig, Nima Tajbakhsh, Nir Ailon, Nirmal Juluru, Nishant Sharma, Oleksii Hrinchuk, Oleksii Kuchaiev, Olivier Delalleau, Oluwatobi Olabiyi, Omer Ullman Argov, Omri Puny, Oren Tropp, Ouye Xie, Parth Chadha, Pasha Shamis, Paul Gibbons, Pavlo Molchanov, Pawel Morkisz, Peter Dykas, Peter Jin, Pinky Xu, Piotr Januszewski, Pranav Prashant Thombre, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Qing Miao, Qiyu Wan, Rabeeh Karimi Mahabadi, Rachit Garg, Ran El-Yaniv, Ran Zilberstein, Rasoul Shafipour, Rich Harang, Rick Izzo, Rima Shahbazyan, Rishabh Garg, Ritika Borkar, Ritu Gala, Riyad Islam, Robert Hesse, Roger Waleffe, Rohit Watve, Roi Koren, Ruoxi Zhang, Russell Hewett, Russell J. Hewett, Ryan Prenger, Ryan Timbrook, Sadegh Mahdavi, Sahil Modi, Samuel Kriman, Sangkug Lim, Sanjay Kariyappa, Sanjeev Satheesh, Saori Kaji, Satish Pasumarthi, Saurav Muralidharan, Sean Narentharen, Sean Narenthiran, Seonmyeong Bak, Sergey Kashirsky, Seth Poulos, Shahar Mor, Shanmugam Ramasamy, Shantanu Acharya, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shiqing Fan, Shreya Gopal, Shrimai Prabhumoye, Shubham Pachori, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Simeng Sun, Smita Ithape, Somshubra Majumdar, Soumye Singhal, Stas Sergienko, Stefania Alborghetti, Stephen Ge, Sugam Dipak Devare, Sumeet Kumar Barua, Suseella Panguluri, Suyog Gupta, Sweta Priyadarshi, Syeda Nahida Akter, Tan Bui, Teodor-Dumitru Ene, Terry Kong, Thanh Do, Tijmen Blankevoort, Tim Moon, Tom Balough, Tomer Asida, Tomer Bar Natan, Tomer Ronen, Tugrul Konuk, Twinkle Vashishth, Udi Karpas, Ushnish De, Vahid Noorozi, Vahid Noroozi, Venkat Srinivasan, Venmugil Elango, Victor Cui, Vijay Korthikanti, Vinay Rao, Vitaly Kurin, Vitaly Lavrukhin, Vladimir Anisimov, Wanli Jiang, Wasi Uddin Ahmad, Wei Du, Wei Ping, Wenfei Zhou, Will Jennings, William Zhang, Wojciech Prazuch, Xiaowei Ren, Yashaswi Karnati, Yejin Choi, Yev Meyer, Yi-Fu Wu, Yian Zhang, Yigong Qin, Ying Lin, Yonatan Geifman, Yonggan Fu, Yoshi Subara, Yoshi Suhara, Yubo Gao, Zach Moshe, Zhen Dong, Zhongbo Zhu, Zihan Liu, Zijia Chen, and Zijie Yan. NVIDIA Nemotron 3: Efficient and open intelligence, 2025. URL https://arxiv.org/abs/2512.20856.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo,

Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. OpenAI o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Krista Opsahl-Ong, Michael J Ryan, Josh Purtell, David Broman, Christopher Potts, Matei Zaharia, and Omar Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9340–9366, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.

Hadas Orgad and Yonatan Belinkov. BLIND: Bias removal with no demographics. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8801–8821, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.490. URL https://aclanthology.org/2023.acl-long.490.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of llm hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KRnsX5Em3W.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701320. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Joon Sung Park, Carolyn Q. Zou, Aaron Shaw, Benjamin Mako Hill, Carrie Cai, Meredith Ringel Morris, Robb Willer, Percy Liang, and Michael S. Bernstein. Generative agent simulations of 1,000 people, 2024. URL https://arxiv.org/abs/2411.10109.

Jacob Pfau, William Merrill, and Samuel R. Bowman. Let's think dot by dot: Hidden computation in transformer language models. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=NikbrdtYvG.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. The Penn Discourse TreeBank 2.0. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias (eds.), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2008/pdf/754_paper.pdf.

Qinglin Qi, Yun Luo, Yijia Xu, Wenbo Guo, and Yong Fang. SpearBot: Leveraging large language models in a generative-critique framework for spear-phishing email generation. *Inf. Fusion*, 122(C), October 2025. ISSN 1566-2535. doi: 10.1016/j.inffus.2025.103176. URL https://doi.org/10.1016/j.inffus.2025.103176.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Paula Rescala, Manoel Horta Ribeiro, Tiancheng Hu, and Robert West. Can language models recognize convincing arguments? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8826–8837, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.515. URL https://aclanthology.org/2024.findings-emnlp.515/.

Liran Ringel, Elad Tolochinsky, and Yaniv Romano. Learning a continue-thinking token for enhanced test-time scaling. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 3324–3345, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. URL https://aclanthology.org/2025.ijcnlp-long.177/.

Margaret E. Roberts, Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082, 2014. ISSN 00925853, 15405907. URL http://www.jstor.org/stable/24363543.

James M. Robins and Miguel A. Hernan (eds.). *Causal Inference: What If*. Taylor & Francis, Boca Raton, 2024.

Till Raphael Saenger, Musashi Hinck, Justin Grimmer, and Brandon M. Stewart. AutoPersuade: A framework for evaluating and explaining persuasive arguments. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16325–16342, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.913. URL https://aclanthology.org/2024.emnlp-main.913/.

Francesco Salvi, Manoel Horta Ribeiro, Riccardo Gallotti, and Robert West. On the conversational persuasiveness of gpt-4. *Nature Human Behaviour*, 9(8):1645–1653, May 2025. ISSN 2397-3374. doi: 10.1038/s41562-025-02194-6. URL http://dx.doi.org/10.1038/s41562-025-02194-6.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In Marine

Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz (eds.), *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Shai Shalev-Shwartz and Amnon Shashua. From reasoning to super-intelligence: A search-theoretic perspective, 2025. URL https://arxiv.org/abs/2507.15865.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Esin DURMUS, Zac Hatfield-Dodds, Scott R Johnston, Shauna M Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tvhaxkMKAn.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and Gothrough self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL https://www.science.org/doi/abs/10.1126/science.aar6404.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. Distinguishing ignorance from error in LLM hallucinations. *arXiv preprint arXiv:2410.22071*, 2024.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. Trust me, I'm wrong: LLMs hallucinate with certainty despite knowing the answer. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 14665–14688, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.792. URL https://aclanthology.org/2025.findings-emnlp.792/.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, et al. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=w6nlcS8Kkn.

Christian Stab and Iryna Gurevych. Annotating argument components and relations in persuasive essays. In Junichi Tsujii and Jan Hajic (eds.), *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pp. 1501–1510, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1142.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. Stop overthinking: A survey on efficient reasoning for Large Language Models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=HvoG8SxggZ.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. Is ChatGPT good at search? investigating Large Language Models as re-ranking agents. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14918–14937, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.923. URL https://aclanthology.org/2023.emnlp-main.923/.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, WWW '16, pp. 613–624, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. ISBN 9781450341431. doi: 10.1145/2872427.2883081. URL https://doi.org/10.1145/2872427.2883081.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20090–20111, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1030. URL https://aclanthology.org/2025.findings-acl.1030/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp

Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025a. URL https://arxiv.org/abs/2503.19786.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Weixin Xu, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, Zonghan Yang, and Zongyu Lin. Kimi k1.5: Scaling reinforcement learning with LLMs, 2025b. URL https://arxiv.org/abs/2501.12599.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. URL https://openreview.net/forum?id=wCu6T5xFjeJ.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=bzs4uPLXvi.

Tyler VanderWeele. *Explanation in causal inference: methods for mediation and interaction*. Oxford University Press, 2015.

Henning Wachsmuth, Nona Naderi, Yufang Hou, Yonatan Bilu, Vinodkumar Prabhakaran, Tim Alberdingk Thijm, Graeme Hirst, and Benno Stein. Computational argumentation quality assessment in natural language. In Mirella Lapata, Phil Blunsom, and Alexander Koller (eds.), *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 176–187, Valencia, Spain, April 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-1017/.

Henning Wachsmuth, Manfred Stede, Roxanne El Baff, Khalid Al-Khatib, Maria Skeppstedt, and Benno Stein. Argumentation synthesis following rhetorical strategies. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle (eds.), *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3753–3765, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1318/.

Kaiwen Wang, Jin Peng Zhou, Jonathan Daniel Chang, Zhaolin Gao, Nathan Kallus, Kianté Brantley, and Wen Sun. Value-guided search for efficient chain-of-thought reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=jOsuKwiCL0.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Bonnie Webber, Rashmi Prasad, Alan Lee, and Aravind Joshi. The penn discourse treebank 3.0 annotation manual. *Philadelphia, University of Pennsylvania*, 35:108, 2019.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. Chain of Thought prompting elicits reasoning in Large Language Models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=_VjQlMeSB_J.

Yujia Xie, Hanjun Dai, Minshuo Chen, Bo Dai, Tuo Zhao, Hongyuan Zha, Wei Wei, and Tomas Pfister. Differentiable top-k with optimal transport. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20520–20531. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ec24a54d62ce57ba93a531b460fa8d18-Paper.pdf.

Weijia Xu, Nebojsa Jojic, Sudha Rao, Chris Brockett, and Bill Dolan. Echoes in AI: Quantifying lack of plot diversity in LLM outputs. *Proceedings of the National Academy of Sciences*, 122(35):e2504966122, 2025. doi: 10.1073/pnas.2504966122. URL https://www.pnas.org/doi/abs/10.1073/pnas.2504966122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with Large Language Models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 11809–11822. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=WE_vluYUL-X.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative AI by backpropagating language model feedback. *Nature*, 639:609–616, 2025.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How Johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14322–14350, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.773. URL https://aclanthology.org/2024.acl-long.773/.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=8rcFOqEud5.

Jiayi Zhang et al. Verbalized Sampling: How to mitigate mode collapse and unlock LLM diversity, 2025a.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in LLMs. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=2cczgOfMP4.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating creativity and diversity in language models. In *Second Conference on Language Modeling*, 2025c. URL https://openreview.net/forum?id=XZm1ekzERf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91f18a1287b398d378ef22505bf41832-Paper-Datasets_and_Benchmarks.pdf.

A Related Work

A.1 Social Science Experiments with Text

Persuasion is central to human communication, spanning political discourse (Bai et al., 2025; Hackenburg et al., 2025b;a), human-AI interaction (Salvi et al., 2025; Costello et al., 2026; Durmus et al., 2024a), and misinformation correction (Costello et al., 2025; Boissin et al., 2025). Computational social science increasingly formalizes persuasion research by treating text as a treatment variable to study how linguistic features causally affect downstream behaviors (Grimmer et al., 2022; Feder et al., 2022). Traditional approaches focus on identifying content themes across document corpora and assessing how these themes affect outcomes (Fong & Grimmer, 2016; Roberts et al., 2014). For example, Saenger et al. (2024) use topic modeling to discover persuasive themes in argument collections, while Egami et al. (2022) analyze how different framings affect bureaucratic responsiveness. Recently, researchers have examined how conversations with LLMs affect beliefs (Costello et al., 2024; 2026; Salvi et al., 2025), identifying consistent patterns in effective messaging, such as emphasizing facts and evidence (Costello et al., 2025).

However, empirical methods in computational social science face limitations in studying fine-grained textual features. Topic modeling approaches (Blei, 2012; Grimmer et al., 2022; Saenger et al., 2024) naturally capture content themes but struggle with structural and stylistic variation. These methods typically identify latent features *ex-post* from existing corpora, constraining analysis to features already present in the data and making it difficult to systematically explore novel feature combinations. Such text-as-treatment experiments ideally manipulate specific features, rhetorical structure (Stab & Gurevych, 2014; Hidey et al., 2017; Chakrabarty et al., 2019; Wachsmuth et al., 2018) (e.g., whether arguments begin with concessions or lead with strong claims), stylistic choices (Deri et al., 2018; Wachsmuth et al., 2018; Breum et al., 2024; El Baff et al., 2024) (e.g., formality, tone, pragmatic objective), and content themes, while maintaining coherence (Durmus et al., 2019) and logical soundness. However, these features are difficult to control systematically in text generation (Saenger et al., 2024), and are therefore rarely analyzed at scale. Moreover, most prior work examines feature presence (whether a theme appears) rather than sequential ordering (when in a message a feature appears), limiting insights into how narrative structure affects argument quality. STATe offers a framework through which to study the effects of granular decision sequences on downstream outcomes.

A.2 Inference-Time-Compute

Inference-time compute (ITC) methods augment LLM generation by allocating additional computation *after* training, either by extending the reasoning depth of individual trajectories or by generating many candidates and selecting among them. These two axes, depth and breadth, are complementary, and many modern systems combine them. The unifying motivation is the empirical finding that the quality of reasoning often scales with test-time computation even when the model weights are held fixed (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Beeching et al., 2024). STATe belongs to this family of methods and specifically extends Tree-of-Thoughts-style search with an *explicit action space* over reasoning strategies.

A.2.1 Depth-oriented ITC

Chain-of-Thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022) scales reasoning *depth* by eliciting intermediate steps before the final answer. This seemingly simple change yields substantial gains on arithmetic, symbolic reasoning, and commonsense tasks, suggesting that the reasoning process itself carries value beyond the final token (Sprague et al., 2025). The rationale behind CoT can be understood through the lens of hidden computation: additional tokens allow the model to perform iterative refinement that a single forward pass cannot (Pfau et al., 2024).

Despite these benefits, CoT reasoning is not always faithful to the underlying inference process (Admoni et al., 2025; Anthropic, 2025b; Guan et al., 2025). Turpin et al. (2023) demonstrate that CoT explanations are frequently post-hoc rationalizations: when models

are biased toward incorrect answers through prompt manipulation, they generate superficially coherent but misleading rationales, causing model performance to drop. This brittleness raises serious concerns for settings where the chain of thought is meant to serve as an auditable record of model reasoning.

A separate line of work asks how to *train* models to reason more effectively. Zelikman et al. (2022) introduce the Self-Taught Reasoner (STaR), which iteratively fine-tunes a model on its own correct rationales, bootstrapping reasoning capability without requiring large annotated rationale datasets. Reinforcement Learning from Verifiable Rewards (RLVR) takes this further: rather than relying on human-curated signal, the model receives reward based on objective correctness criteria such as code compilation or arithmetic verification. Shao et al. (2024) introduce Group Relative Policy Optimization (GRPO), a memory-efficient variant of PPO, and show that it substantially improves mathematical reasoning. DeepSeek-AI et al. (2025) then demonstrate that pure RL training without supervised warm-start can induce emergent reasoning behaviors such as self-reflection, backtracking, and extended chains of thought—matching OpenAI o1 on competitive mathematics benchmarks. Similarly, OpenAI et al. (2024) and Muennighoff et al. (2025) show that models explicitly optimized for long-horizon reasoning substantially amplify the benefits of depth-oriented ITC.

STATe is complementary to this line of work. Where depth-oriented methods focus on optimizing *how long* a model reasons, STATe focuses on *what* it reasons about at each step. By conditioning generation on explicit action templates, STATe makes high-level decisions in a reasoning trajectory auditable and manipulable in a way that standard CoT, even when faithful, does not support.

A.2.2 Breadth-oriented ITC

Breadth-oriented methods generate multiple candidate responses and select among them according to an external criterion, improving robustness [10] by reducing reliance on a single reasoning chain (Brown et al., 2020; Stiennon et al., 2020). For example, Self-Consistency (Wang et al., 2023; Chen et al., 2024; Taubenfeld et al., 2025) samples multiple candidate reasoning paths and then selects an answer by majority voting. The central challenge is inducing *meaningful diversity* across candidates rather than many near-duplicates of the same response.

The standard approach is high-temperature sampling, which expands the vocabulary distribution over the next token. More principled truncation strategies have been proposed to improve the quality–diversity trade-off. Nucleus (top-p) sampling (Holtzman et al., 2020) truncates the distribution to the smallest set of tokens whose cumulative probability exceeds p, preventing catastrophically low-probability tokens at modest quality cost. Top-k sampling (Fan et al., 2018) truncates to the top-k tokens by probability mass, offering a simpler but less adaptive alternative. More recently, min-p sampling (Minh et al., 2025) introduces a dynamic threshold that scales the cutoff by the top token's probability, effectively widening the candidate set when the model is uncertain and narrowing it when the model is confident. These token-level strategies share a common limitation: they operate on the logit distribution at each decoding step and therefore do not control the *semantic* content or rhetorical strategy of the generated response.

At higher levels of abstraction, prompt-based diversity methods attempt to elicit variation through the input rather than the decoding algorithm. Zhang et al. (2025a) propose Verbalized Sampling (VS), a training-free prompting strategy that asks the model to jointly generate a set of responses and verbalize a probability distribution over them. By surfacing the model's internal uncertainty as explicit text, VS bypasses the typicality bias introduced by post-training alignment and recovers diversity that was suppressed during RLHF. VS achieves 1.6–2.1$\times$ diversity gains over direct prompting on creative writing tasks without sacrificing quality. However, VS is fundamentally bounded by a single LLM generation: to produce n diverse responses, the entire batch must be generated within one context window. This constraint makes VS poorly suited to large n.

[10] For example, the model may fail to create a valid generation due to a refusal, exceeding the context limit, or failing to adhere to a structured output schema.

STATe addresses the diversity bottleneck at a higher level of abstraction. Rather than modifying the decoding algorithm or asking the model to self-sample a distribution, STATe precomputes an explicit set of *action templates*—discrete, interpretable specifications of rhetorical strategy—and uses a reranker controller to select top-n distinct actions for each branching step. This guarantees that each branch explores a semantically distinct region of the reasoning space without requiring high temperature or long-context self-sampling. Diversity is therefore a structural property of the search procedure rather than a statistical side-effect of decoding.

A.2.3 Tree-of-Thoughts

Tree-of-Thoughts (ToT) (Yao et al., 2023a) unifies depth and breadth by recasting LLM inference as search over a tree of partial reasoning states. At each layer, the model branches into multiple candidate thoughts, an evaluator scores them, and low-value branches are pruned, preventing exponential growth and error propagation. Hao et al. (2023) formalize this connection by treating LLM inference as planning in a world model, while Hao et al. (2024), Beeching et al. (2024), and Shalev-Shwartz & Shashua (2025) explore the performance of depth-first versus breadth-first search for complex reasoning tasks.

Several extensions enrich the basic ToT framework with more principled search algorithms. Monte Carlo Tree Search (MCTS) (Kocsis & Szepesvári, 2006; Coulom, 2006; Browne et al., 2012) balances exploration and exploitation via upper-confidence bounds, enabling adaptive allocation of the evaluation budget toward high-value regions of the reasoning tree. Zhang et al. (2024a) integrate MCTS with process reward models to guide search and simultaneously generate high-quality training data for policy and reward model improvement, outperforming both Best-of-n and standard ToT under the same computation budget. Chain of Preference Optimization (Zhang et al., 2024b) uses the preference signal implicit in the ToT search tree—which branches were kept versus pruned—to fine-tune the model with DPO, achieving CoT-level inference cost at ToT-level quality. These RL-flavored formulations are natural: ToT can be read as a form of tree-structured policy search, where each branching action is sampled from a policy π_θ, intermediate states receive process rewards from a value function V, and the goal is to maximize the reward of the final leaf state (Schulman et al., 2017; Shao et al., 2024).

Despite this expressiveness, standard ToT implementations share two important limitations that STATe addresses. First, existing methods rely exclusively on stochastic temperature sampling to differentiate branches. Because sampling operates at the token level, branches in the same tree often converge on semantically similar content (Jiang et al., 2025; Zhang et al., 2025c), undermining the exploration benefit that motivates tree search in the first place. Second, because reasoning decisions are implicit in the decoding process, it is difficult to attribute differences in output quality to specific choices made at specific reasoning steps. This opacity limits the interpretive utility of ToT: researchers can observe that some trajectories outperform others, but not *why*.

STATe replaces token-level stochasticity with an explicit, structured action space, making every branching decision both interpretable and auditable. The controller selects a named action from a fixed vocabulary—specifying, for example, which rhetorical structure and content theme to employ at each step—and the generator prefills that action as a textual intervention before sampling the continuation. This design decouples *what to reason about* (the action) from *how to express it* (the generated token sequence), a separation that standard ToT conflates. As a result, each path through the STATe tree corresponds to a logged, human-readable action sequence that can be subjected to formal attribution analysis.

B DSPy Background

STATe is built on **DSPy** (Khattab et al., 2024), which provides a modular, declarative approach to programming LLMs. DSPy separates *what* a task does (expressed as a **Signature**) from *how* it is executed (determined by a **Module**, **Adapter**, and **Language Model**). This separation of concerns makes components independently configurable and composable, and enables automatic prompt optimization without manual re-engineering of prompts.

B.1 Core DSPy Primitives

Fields. The fundamental building blocks in DSPy are *Fields*, which define the input/output schema of a task through typed annotations and natural-language descriptions. `InputField` objects describe the variables a module expects; `OutputField` objects specify what the module should produce. For example:

```
topic: str = InputField(desc='Debate topic')
stance: Literal['PRO', 'ANTI'] = InputField(desc='Stance to argue')
argument: str = OutputField(desc='Generated argument')
```

Signatures. A *Signature* is a declarative task specification: it bundles a set of Fields together with a task-level docstring instruction, defining *what* the module should do without specifying any prompt template. Signatures are defined as Python classes that subclass `dspy.Signature`:

```
class GenerateArgument(dspy.Signature):
    '''Generate an argument for the given topic and stance.'''
    topic: str = dspy.InputField(desc='Debate topic')
    stance: Literal['PRO', 'ANTI'] = dspy.InputField(desc='Stance to argue')
    argument: str = dspy.OutputField(desc='Generated argument')
```

Modules. A *Module* is a parameterized layer that executes a Signature. The basic DSPy module `dspy.Predict` takes a Signature and, at inference time, calls the configured Language Model to produce predictions. Modules are composable: larger pipelines can be assembled from multiple modules, each handling a distinct subtask (e.g., planning, generation, evaluation).

Adapters. An *Adapter* bridges a Signature and a Language Model by formatting the inputs and the signature's instruction into a concrete prompt string, and by parsing the LLM's raw textual response back into typed, structured field values. DSPy ships with several built-in adapters (e.g., `ChatAdapter`, `JSONAdapter`); STATe uses custom adapters (`LocalVLLMAdapter` for generative models and `LocalVLLMScoringAdapter` for reranker models) that extend these to support tool-call formatting, prefill injection, and query–document scoring. After the Adapter parses the output, it type-checks each `OutputField` value—raising an error if the response does not conform to the declared type (e.g., a `Literal` constraint).

B.2 Instantiation and Forward Pass

Instantiation phase. A Module is created by combining a **Signature** (task definition: *what* to do) with a **Language Model** (executes prompts: *which* LLM) and an **Adapter** (formats prompts, parses outputs).

Forward (inference) phase. When a Module is called with an *Example* (a dictionary of input field values matching the Signature), the pipeline proceeds as follows: (1) the Adapter formats the Signature instruction, field descriptions, and input values into a prompt string; (2) the Language Model generates a textual response; (3) the Adapter extracts and type-checks values for each `OutputField` from the response, returning a **Prediction** object with structured field values.

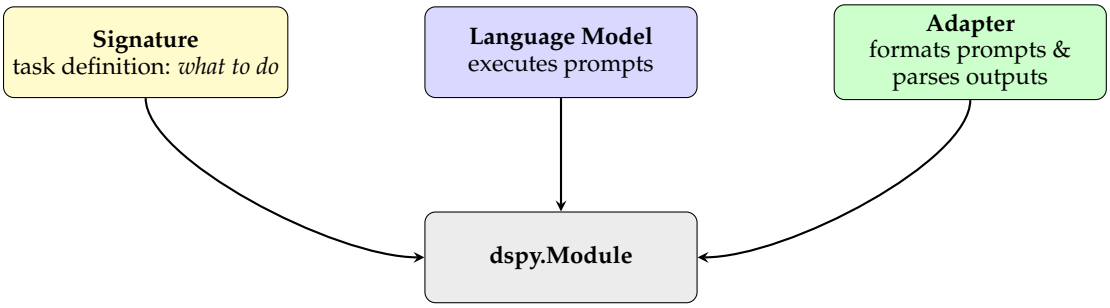

Figure 4: A `dspy.Module` is assembled from a Signature (task specification), a Language Model (inference backend), and an Adapter (prompt formatting and output parsing).

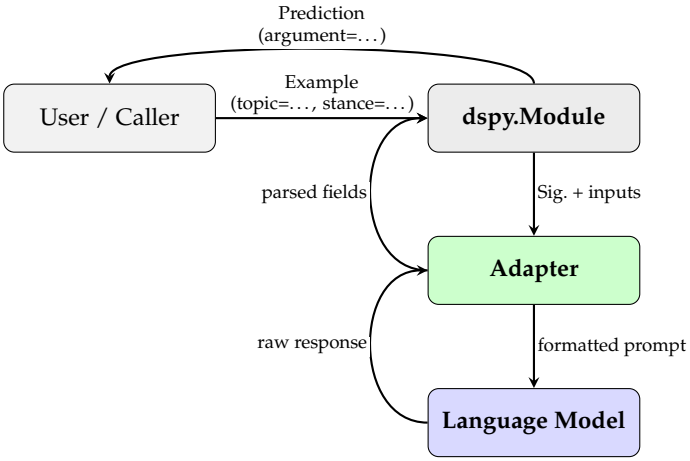

Figure 5: The Module delegates prompt formatting to the Adapter, which calls the Language Model and parses its response into a Prediction.

B.3 Prompt Optimization with DSPy

A key advantage of the DSPy abstraction is that, because all prompt logic is encapsulated inside Signatures and Adapters, the *textual content* of prompts can be treated as learnable parameters and optimized automatically. DSPy optimizers, like the two examples below, search over the space of candidate prompts and select those that maximize a user-defined metric on a development set.

MIPROv2 (Opsahl-Ong et al., 2024). MIPROv2 (Multi-prompt Instruction PRoposal Optimizer v2) is a Bayesian optimization-based prompt optimizer. It jointly optimizes two components of each DSPy module: (i) **Signature instructions**—the free-text preamble in the system message that describes the task to the LLM; and (ii) **few-shot demonstrations** ("demos")—a small set of example (input, output) pairs prepended to the user message to guide in-context learning. MIPROv2 uses a surrogate model over the candidate-prompt space to propose high-quality instruction rewrites and select informative demo subsets, significantly reducing the number of LLM calls required compared to brute-force search.

GEPA (Agrawal et al., 2026). GEPA (**Ge**netic **Pa**reto) frames prompt optimization as a reflective evolution process. An LLM "editor" is given the current instruction, recent failed examples (coupled with their scores and textual feedback), and is asked to propose revised instructions. These proposals are evaluated on a held-out set, and the best-performing candidates are carried forward. GEPA complements MIPROv2 by enabling more targeted, semantically-informed instruction rewrites, and is particularly well-suited to iterative refinement when the optimization landscape is smooth.

C STATe Modules

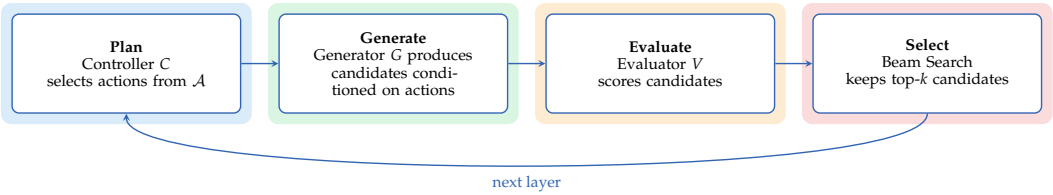

Figure 6: STATe module loop used in tree search: controller planning, candidate generation, evaluation, and beam selection, followed by expansion to the next layer.

STATe's Controller, Generator, and Evaluator (Section 3.1) are implemented with DSPy (Khattab et al., 2024) Modules (Appendix B). Figure 7 illustrates how these components interlock across layers of the search tree. At each layer i, the **controller** observes the current beam of states and, for each state s_{i-1}, selects up to n actions from the action space $\mathcal{A}$. Each selected action is executed to produce a `ReasoningIntervention` (a prefix and context guidance), which the **generator** injects into the LLM's assistant message as a prefill before sampling a continuation. This produces a set of candidate child states; each child is either an intermediate reasoning state $s_i = [s_{i-1}, z_i]$ or a final-answer state $s_i = [s_{i-1}, y]$, depending on whether the controller signalled early stopping. Intermediate children are passed to the **evaluator**, which scores them with a Process Reward Model V_{PRM}; final-answer children are scored with an Outcome Reward Model V_{ORM} and collected for the terminal selection step. **Beam selection** then retains the top-k intermediate states by score, forming the beam for layer $i+1$. This Plan→Generate→Evaluate→Select cycle repeats until the maximum depth d is reached or all active branches have terminated via early stopping, at which point the highest-scoring final-answer state is returned. In practice, LLM calls from all components are parallelized across all nodes in a given layer using vLLM (Kwon et al., 2023), significantly reducing end-to-end latency compared to sequential decoding.

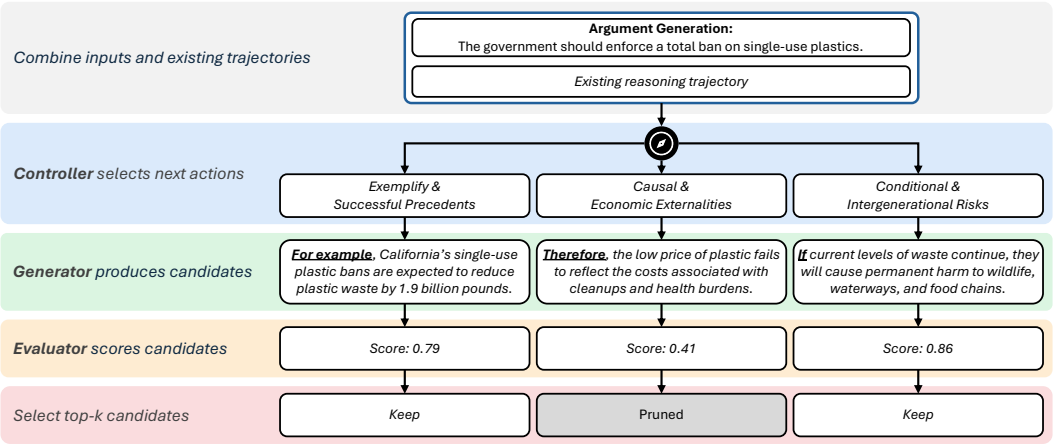

Figure 7: Stylized example of STATe's Plan→Generate→Evaluate→Select loop. The controller plans which actions to explore, the generator expands candidate trajectories, the evaluator scores them, and beam selection retains the top-k states.

First step	Intermediate step	Final step
<thinking> <step> ## internal_reasoning I should evaluate how plastics persist for centuries in the environment, meaning today's convenience imposes long-term burdens on future generations who had no say in their creation. ## claim If current levels of plastic waste continue, they will cause permanent harm to marine ecosystems. . .	<thinking> <step> ## internal_reasoning I should identify risks. . . ## claim If current levels of plastic waste continue. . . </step> . . . <step> ## internal_reasoning I should examine case studies from Rwanda, the EU, Kenya, and various US states showing that bans are enforceable and produce measurable reductions in pollution. ## claim For example, California's single-use plastic bans are expected to reduce plastic waste by 1.9 billion pounds. . .	<thinking> <step> ## internal_reasoning I should identify risks. . . ## claim If current levels of plastic waste continue. . . </step> . . . <step> ## internal_reasoning I should evaluate. . . ## claim For example, California's single-use plastic ban. . . </step> . . . </thinking> <answer>

Table 4: Illustrative interventions for argument generation example in favor of single-use plastics ban. Templates are in black, internal reasoning in teal, prefixes in blue, and the model continuation in orange. Each column shows the generation state at different stages: first step (single claim), intermediate (multiple claims), and final (complete reasoning with answer delimiter). The action space used here is presented in Appendix H.2.

C.1 Controller

We treat each action as a *tool call*. Selecting an action corresponds to choosing a tool name from a fixed set of action templates (Appendix H) and providing values for the tool's arguments. Executing the tool returns a structured intervention, i.e., a prefix and internal reasoning guidance, that is injected into the next generation step. In our argument generation example (Figure 1 and Figure 7), the actions encode *structure* and *content* dimensions, per Wachsmuth et al. (2018). Each argument in a tool has a fixed set of choices, which are dictated by an action-space's "dimensions" (Appendix H.3). This mirrors the iterative tool-use paradigm in ReAct (Yao et al., 2023b), with two key differences: (i) we allow *branching* by selecting multiple tools per input state, and (ii) our tools are lightweight prefix interventions rather than external capabilities such as retrieval or code execution. The controller's action selection process supports *early stopping*: if the controller determines that reasoning is sufficient, it selects a dedicated `FINISH` action, signifying that the generator should produce a final answer.[11] We implement two kinds of controllers: one that uses a generative LLM to produce tool calls (Appendix C.1.1), and another that uses a reranker LLM (Zhang et al., 2025b) to pick among all possible tool calls (Appendix C.1.2).

The performance of a controller matters most when the action space for the task is large, and relatively few actions from the action space would enable a high-quality final response. For example, in a math task, where the action space consists of dozens or hundreds of math operations, but less than a handful are valid or productive, a poor-quality controller would steer generation towards flawed responses. On the other hand, open-ended tasks like argument generation, where many actions from the action space could yield high-quality responses, are less-dependent on the controller's quality. Our experiments in this paper

[11]This is represented in our controller as an additional tool that takes no arguments.

focus on open-ended tasks with relatively small action-spaces. There are relatively few cases in which a sequence of choices by the controller would explicitly steer the response in a poor direction (e.g., starting with an example or specification in argument generation).

C.1.1 Generative controller

The generative controller prompts an LLM to *propose* a tool call (Yao et al., 2023b; Karpas et al., 2022) from action space $\mathcal{A}$ conditioned on the current state: it chooses which tool to use (which action template) and provides values for the tool's arguments (permitted choices are specified through a `Literal` type). In this setup, the same generator model that produces thoughts can also decide what to do next. A key advantage is that the controller can utilize ITC methods like CoT, and produce *natural-language rationales* for why a tool call is appropriate given the chain so far.[12] A key limitation is *low action diversity*: because tool-call generation is itself sampled from an LLM, the controller can collapse to repeatedly predicting the same high-probability action, reducing the benefit of branching. While verbalized sampling (Zhang et al., 2025a) can help if n (the number of actions the controller must select) is small, it is not effective if n is large (Appendix A.2.2).

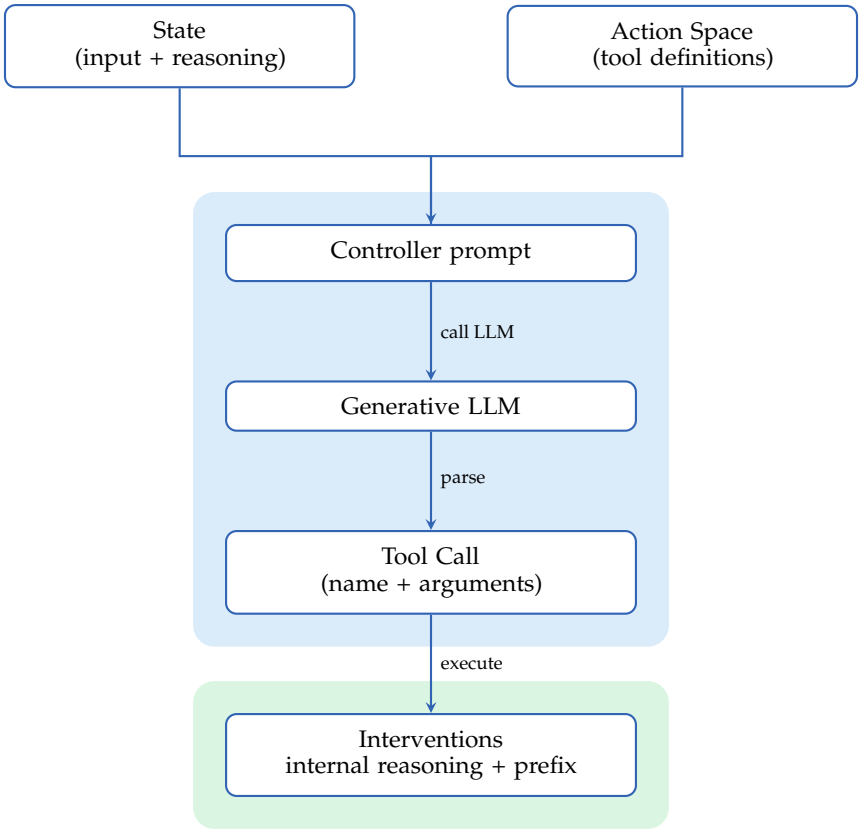

Figure 8: Generative controller pipeline. Blue band = controller, green band = output (feeds generator).

[12]Preceding tool call outputs with reasoning enables more informed decision-making and easier debugging.

System Message

```
Generate an argument which takes the provided stance towards the provided topic.

You are given `topic` and `stance` and your goal is to finish with `argument`.
To accomplish this goal, you will need to reason about the problem step by step
rather than generating `argument` directly.
You have up to `number_of_additional_reasoning_steps` additional steps to reason
about the problem before generating `argument`.
Refer to the existing reasoning steps under the `reasoning` header to inform
your next step. Reasoning steps are ordered sequentially, and each one includes
a `claim` header above the content of the step itself.
Choose a tool to use from the following options:
(1) `intervene_on_next_reasoning_step`:
        * Description:
                Determine the best choice in each dimension to improve the quality
                of the next reasoning step. You must select one choice for each
                of the provided dimensions.
        * Arguments:
                - causal_styles: Literal['figurative_language,
                  statistical_and_data_driven, narrative_and_anecdote,
                  expert_and_authoritative_voice, repetition_and_parallelism,
                  contrast_and_antithesis, measured_and_authoritative_tone,
                  passionate_and_urgent_tone, direct_engagement, scope_and_framing']
                        Forces the next reasoning step to adopt a specific rhetorical
                        style or expressive technique.
                - causal_subtopics: Literal['cost_benefit_and_impact_analysis,
                  rights_and_liberties, justice_and_fairness, ethical_principles,
                  governance_and_accountability, risk_and_unintended_consequences,
                  feasibility_and_implementation, incentives_and_power_dynamics,
                  precedent_and_long_term_effects, stakeholder_responsibility']
                        Forces the next reasoning step to analyze the issue through
                        a specific argumentative lens or topical framework.
(2) `finish`:
        * Description:
                Signals that the reasoning so far is sufficient for producing a
                high-quality response. If selected, the generator will produce
                the final output rather than reasoning further.

Your inputs will be:
1. `topic` (str): The topic to generate an argument about
2. `stance` (Literal['PRO', 'ANTI']): The stance to take on the topic
3. `reasoning` (str): The existing reasoning steps towards producing `argument`.
   Each step's content is under the `claim` header.
4. `number_of_additional_reasoning_steps` (int): The maximum number of additional
   reasoning steps you can take before you must produce `argument`.

Your goal is to produce the following outputs:
1. `action` (str): The selected action (tool) to guide the next reasoning step.
2. `action_arguments` (dict[str, Any]): The input arguments for the selected action.

Please provide your response with each output field under its own header using
the format:
## field_name
Your response for that field here

NOTE: Line breaks, capitalization, and punctuation (i.e., '##' followed by a
space) are important. If you do not follow these guidelines, your response will
be rejected.
```

Prompt 1: Generative Controller Teal: task signature and action-space configuration. Blue: concrete runtime value.

User Message

```
## topic
Should single-use plastics be banned?

## stance
PRO

## reasoning
<thinking>
<step>
## internal_reasoning
Opening with env harm.
## claim
Single-use plastics are a leading contributor to ocean pollution.
</step>
</thinking>

## number_of_additional_reasoning_steps
2

Respond with the corresponding output fields, starting with the field
`## action`, then `## action_arguments` (must be formatted as a valid
Python dict[str, Any])
```

Prompt 2: Generative Controller User Message

C.1.2 *Reranker controller*

To introduce reliable diversity, we instead use a discriminative reranker controller that scores *all* candidate action-argument combinations and selects the top-n. We formulate action selection by measuring the relevance of a query and a document using a cross-encoder (Reimers & Gurevych, 2019; Thakur et al., 2021; Santhanam et al., 2022; Laban et al., 2022; Sun et al., 2023; Zhang et al., 2025b): the *query* contains the input x and the reasoning chain Z_i, while each *document* is a description of the effect of the given tool and parameter (e.g., "introduce a new claim that expands on financial impacts"). A reranker assigns a relevance score to each document, yielding a distribution over actions, and we take the top-n scoring actions for expansion. This design supports diverse branching (by selecting different high-scoring actions) and enables efficient enumeration when $\mathcal{A}$ is finite and structured.

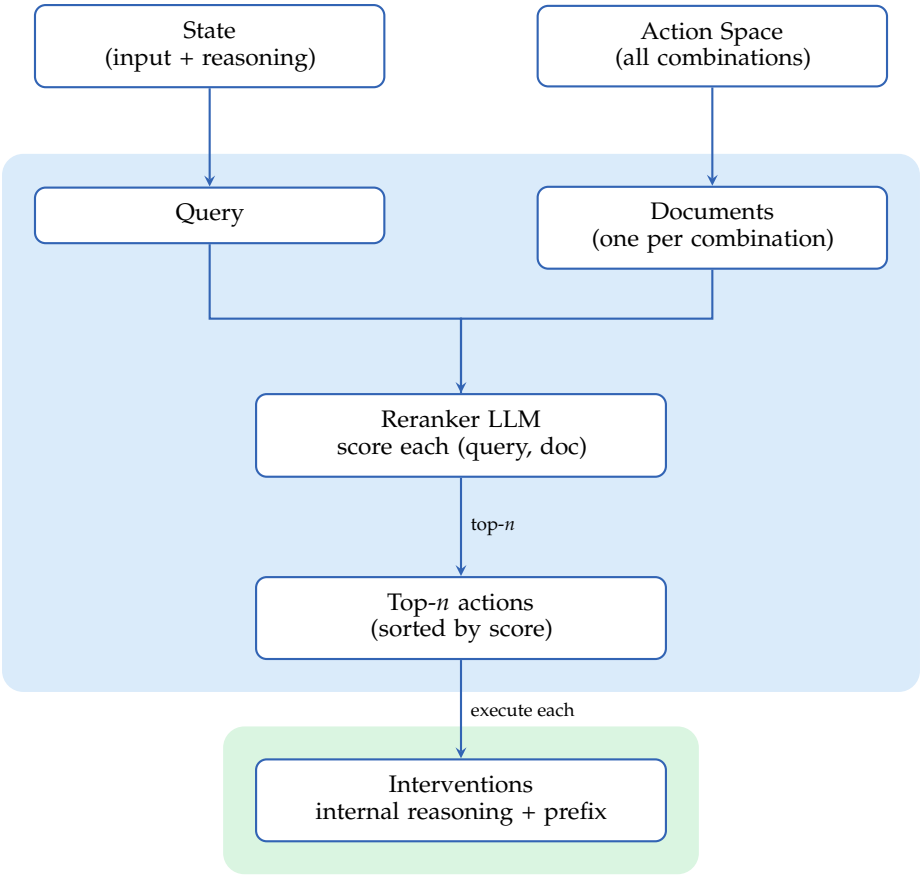

Figure 9: Reranker controller pipeline. Blue band = controller, green band = output (feeds generator).

System Message

```
Judge whether the Document meets the requirements based on the Query and the
Instruct provided. Note that the answer can only be `yes' or `no'.
```

Prompt 3: Reranker Controller Each action is scored as a query–document pair (Zhang et al., 2025b). The system message constrains output to "yes"/"no"; the user message carries the instruction, query (inputs + reasoning), and document (action description). Teal: task signature. Blue: concrete runtime value.

User Message

```
<Instruct>: Your objective is to decide what action to take for the next
reasoning step for a user-assigned task.

The user provided the following task:
"Generate an argument which takes the provided stance towards the provided topic."

This task requires taking a sequence of reasoning steps to reach a solution.
You must determine what action to take next.
You will find the inputs for this task under the # Inputs header in the Query.
You will find the intermediate reasoning trajectory towards solving this problem
under the # Reasoning header in the Query.
You will find the action under consideration under the # Action heading in the
Document.

Judge whether the provided action is likely to be a good next step for
addressing the user's task.
NOTE: You have 2 actions remaining before you must return a final answer.

<Query>: # Inputs
topic: Should single-use plastics be banned?
stance: PRO

# Reasoning
<thinking>\n<step>\n## claim\nSingle-use plastics are a leading contributor...\n</step
    >\n</thinking>

<Document>: # Action
structure: causal_reasoning | subtopic: environmental_impact
Forces the next step to state causes, effects, and consequences.
I should analyze environmental causes and cascading ecological effects.
```

Prompt 4: Reranker Controller User Message

C.2 Generator

The Generator expands the search tree by producing candidate reasoning steps or final answers. Responses are structured using XML tags that create natural stopping points: each intermediate reasoning step is enclosed in `<step>...</step>`, and the final answer in `<answer>...</answer>`. Generation is terminated via *stop tokens*: `</step>` for intermediate steps and `</answer>` for final answers. The choice of stop token depends on whether the controller has signalled to continue reasoning (`continue_reasoning=True`) or to produce a final answer (`continue_reasoning=False`).

The generator injects controller guidance into the LLM through *assistant prefilling* via vLLM's `continue_final_message` mechanism. Given an action a_i^j that returns a `ReasoningIntervention` with fields `internal_reasoning` (context guidance) and `prefix` (text that opens the next step), the adapter builds a prefill string of the form:

```
## internal_reasoning
{guidance}
## {reasoning_field}
{prefix}
```

This is appended to the existing assistant message (i.e., prior reasoning steps if they exist), and the model continues generating from the prefix. The prefill therefore simultaneously injects high-level guidance (via `internal_reasoning`) and steers the surface form of the generation (via `prefix`).

Figure 10 illustrates the full pipeline from state and intervention to the child state.

Intermediate reasoning steps. Once the controller selects actions $\{a_i^1, \ldots, a_i^n\}$ for a given parent state s_{i-1}, each action a_i^j is executed to obtain text guidance $a_i^j()$, which is appended to the LLM's assistant message as a prefill. Given the parent state $s_{i-1} = [x, Z_{i-1}]$, where $Z_{i-1} = [z_1, \ldots, z_{i-1}]$ represents the reasoning steps so far, we sample a continuation for each action as:

$$z_i^j \sim G\big(z \mid x, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp}\big)\ [:\texttt{</step>}] \tag{3}$$

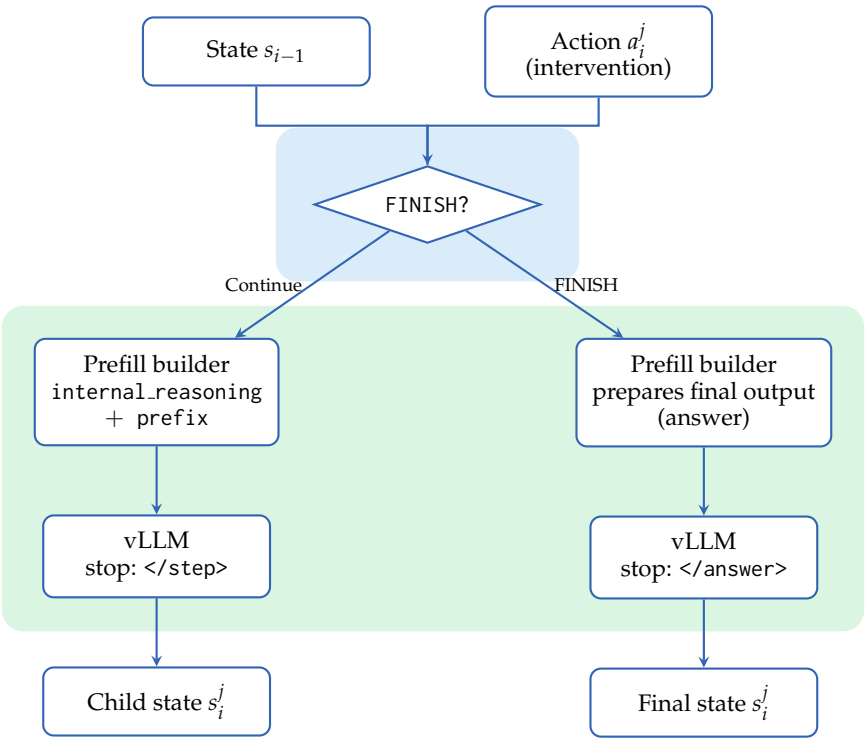

Figure 10: Generator pipeline. The controller decides whether to continue reasoning or finish (blue band). The generator executes accordingly (green band).

Each generated thought z_i^j is combined with the current state to form a child state $s_i^j = [s_{i-1}, z_i^j]$. The continuation is terminated by the stop token `</step>`, per the prompt described in Prompt 5. Table 4 illustrates what these interventions look like for our argument generation task.

Final answer generation. A final answer is always generated by selecting the `FINISH` action: either the controller chooses it explicitly (early stopping) or it is forced when maximum depth d is reached. In both cases, the formalism is the same: we condition on the reasoning trace and the `FINISH` action's prefill:

$$y^j \sim G(y \mid x, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp})\ [\texttt{: </answer>}] \tag{4}$$

where a_i^j is the `FINISH` action, and the stop token is `</answer>`.

System Message

```
# Instructions

Generate an argument which takes the provided stance towards the provided topic.

Your inputs will be:
1. `topic` (str): The topic to generate an argument about
2. `stance` (Literal['PRO', 'ANTI']): The stance to take on the topic

Your goal is to produce the following output:
`argument` (str): The generated argument

When solving this problem, you must break down your solution into a series of
reasoning steps, followed by a final answer.
Each step towards the answer should be encased within <step>...</step> tags,
and contain a `claim` that advances the solution towards producing `argument`.
Your final answer must remain highly faithful to the reasoning steps and their
underlying ideas.

- Preserve the full set of reasoning steps and their original order.
- You may lightly rephrase for clarity and readability, but the meaning must remain
    unchanged.
- Structure and sequence should closely follow the original reasoning steps.
- Do NOT introduce new ideas, arguments, facts, or examples.
- Do NOT remove or significantly alter any existing reasoning.

Your goal is to produce a clear synthesis that respects both the content and
structure of the original reasoning, while allowing minimal refinement.

Your reasoning process should follow the rules below:
- Each `claim` (of type `str`) entails a component of the argument that advocates
  for the given stance towards the topic.
- Before writing a new `claim`, start with some internal reasoning which
  discusses and guides what to do with the next `claim`.

## Response Format

Once a user provides `topic` and `stance`, your response must follow this
exact template:

<thinking>
<step>
## internal_reasoning
Your internal reasoning about the first `claim`
## claim
The first reasoning step towards producing `argument`
</step>
<step>
## internal_reasoning
Your internal reasoning about the second `claim`
## claim
The second reasoning step towards producing `argument`
</step>
...
<step>
## internal_reasoning
Your internal reasoning about the final `claim`
## claim
The final reasoning step towards producing `argument`
</step>
</thinking>
<answer>
## argument
Your response for `argument` here
</answer>
```

Prompt 5: Generator — System Message Instantiated for the argument generation running example (topic: *Should single-use plastics be banned?*, stance: *PRO*). Teal: task signature and hyperparameters. Blue: concrete inference-time value.

User Message

```
Generate an argument which takes the provided stance towards the provided topic.

## topic
Should single-use plastics be banned?

## stance
PRO
```

Prompt 6: Generator — User Message

Assistant Message (prefill)

```
<thinking>
<step>
## internal_reasoning
I'll open with the core environmental harm caused by single-use plastics.
## claim
Single-use plastics are a leading contributor to ocean pollution, harming
marine ecosystems and entering the food chain.
</step>
<step>
## internal_reasoning
I should address the causal chain between plastic production and environmental
degradation, using concrete data.
## claim
Banning single-use plastics would
```

Prompt 7: Generator — Assistant Message (prefill)

C.2.1 Synthesis modes

Once a final answer is triggered, the generator instantiates one of four *synthesis modes* that determine how tightly the output must mirror the intermediate reasoning trace. The modes span a continuum from verbatim preservation to unconstrained generation, trading off action attribution strength against output fluency. The synthesis mode is injected into the Generator's system prompt at inference time, so the same generator can operate in any of the four synthesis modes without retraining. The system-prompt instructions for each mode are shown in Prompts 8–11.

Final Output Instruction — Strict

```
Your final answer must include the full text from all reasoning steps,
copied nearly word-for-word and in sequential order.

- Preserve the exact wording, phrasing, structure, and examples.
- Maintain the original order and logical flow exactly as provided.
- You may add only: A brief introduction/conclusion, short transitions.
- Do NOT rewrite, paraphrase, summarize, or restructure.
- Do NOT add new ideas, arguments, facts, or examples.
```

Prompt 8: Final Output Instruction (Strict) Preserves exact wording from reasoning steps.

Final Output Instruction — Faithful

```
Your final answer must remain highly faithful to the reasoning steps and
their underlying ideas.

- Preserve the full set of reasoning steps and their original order.
- You may lightly rephrase for clarity and readability, but the meaning
  must remain unchanged.
- Structure and sequence should closely follow the original reasoning steps.
- Do NOT introduce new ideas, arguments, facts, or examples.
- Do NOT remove or significantly alter any existing reasoning.

Your goal is to produce a clear synthesis that respects both the content
and structure of the original reasoning, while allowing minimal refinement.
```

Prompt 9: Final Output Instruction (Faithful) Allows light rephrasing while preserving meaning.

Final Output Instruction — Restructured

```
Your final answer should preserve the same core ideas and reasoning from
the steps provided, while improving clarity and coherence.

- Maintain the essential arguments and logical intent.
- You may rephrase, reorganize, and restructure the content for better
  flow and readability.
- The overall set of ideas should remain the same, but the presentation
  may differ.
- Do NOT introduce new ideas or factual content beyond what appears in
  the reasoning steps.

Your goal is to produce a well-structured synthesis that faithfully
reflects the original reasoning while optimizing expression and
organization.
```

Prompt 10: Final Output Instruction (Restructured) Allows free reorganisation while preserving core ideas.

Final Output Instruction — Conclusion

```
Your final answer must be a standalone response to the user's task and
instructions.

- Focus on producing a clear, logically consistent, and high-quality
  final answer.
- You are not required to preserve the structure, wording, or order from
  the reasoning steps (between <thinking>...</thinking> tags).
- Use the reasoning steps only as internal guidance; do NOT mention them
  or refer to them.
- The user will *not* have access to the reasoning steps you wrote, so
  referencing them is confusing and unhelpful.
- Do NOT explain what you are going to do; just produce the final
  deliverable.
- If the task requires strict formatting (math, code, etc.), follow those
  requirements exactly in the final output.

Your goal is to output only the final answer content that satisfies the
user's instructions.
```

Prompt 11: Final Output Instruction (Conclusion) Treats reasoning as internal guidance only, producing a standalone answer.

C.3 Evaluator

C.3.1 Generative Evaluator

The generative evaluator prompts an LLM to act as a judge and score the reasoning (PRM) or output (ORM) at hand. With this setup, the same generator model that produces thoughts and outputs is used to score them. In practice, our evaluators follow a *multi-item rubric*: a list of domain-dependent criteria [13] the judge should check (e.g., constraint satisfaction, correctness, coherence, style). Importantly, we allow rubric items to carry different importance weights, so that violations of high-priority requirements dominate the score. This is implemented via weighted, multi-dimensional scoring (each rubric item has a weight, and the final score is a weighted aggregate).

In the original ToT paper, Yao et al. (2023a) proposed a generative LLM-as-a-Judge system that utilized majority voting over an ensemble of judges. This allowed the evaluator in ToT to calibrate its score given other candidates, rather than rate each one blindly. While we support an ensemble of scoring judges, we do not support voting as a PRM or ORM evaluation mechanism. We deliberately avoid the voting mechanism, since it does not scale well in generative models if the number of candidates to evaluate (in our case, either n in the first layer or $n \times k$ in subsequent layers) is too large. We also built a generative Evaluator that utilizes many pairwise comparisons, and then ranks candidates according to Bradley-Terry (BT) scores. Even without covering all possible candidate pairs, BT allows us to estimate the relative quality of candidates in a non-isolated fashion, and without exceedingly large contexts.

[13] For example, in argument generation tasks, the criteria should be centered on the argument's persuasiveness, logical rigor, and effective structure (Wachsmuth et al., 2017).

System Message (PRM)

```
Judge the quality of reasoning steps for a problem-solving process.
The task requires producing `argument` given `topic` and `stance`.
Reasoning steps towards producing `argument` are provided in
`reasoning_steps`.

Since you are evaluating intermediate reasoning, don't score based on
completeness, and instead score based on the rubric items below:
- persuasiveness: Persuasiveness score (an int between 1 and 7).
- coherence: Coherence score (an int between 1 and 7).
- relevance: Relevance score (an int between 1 and 7).

Your inputs will be:
1. `topic` (str): The topic to generate an argument about
2. `stance` (Literal['PRO', 'ANTI']): The stance to take on the topic
3. `reasoning_steps` (list[str]): List of `claim`s to evaluate toward
producing `argument`

Your goal is to produce the following outputs:
1. `persuasiveness` (int): Persuasiveness score
        Constraints: >= 1 and <= 7
2. `coherence` (int): Coherence score
        Constraints: >= 1 and <= 7
3. `relevance` (int): Relevance score
        Constraints: >= 1 and <= 7

Please provide your response with each output field under its own header using
the format:
## field_name
Your response for that field here

NOTE: Line breaks, capitalization, and punctuation (i.e., '##' followed by a
space) are important. If you do not follow these guidelines, your response will
be rejected.
```

Prompt 12: Generative Evaluator — System Message (PRM) Teal: task signature and rubric as rendered by the adapter. Blue: concrete runtime value.

User Message (PRM)

```
## topic
Should single-use plastics be banned?

## stance
PRO

## reasoning_steps
['Single-use plastics are a leading contributor to ocean pollution, harming
marine ecosystems and entering the food chain.', 'Banning them would
accelerate adoption of reusable and biodegradable alternatives while reducing
long-term cleanup costs.']

Respond with the corresponding output fields, starting with the field
`## persuasiveness` (must be formatted as a valid Python int), then
`## coherence` (must be formatted as a valid Python int), then
`## relevance` (must be formatted as a valid Python int)
```

Prompt 13: Generative Evaluator — User Message (PRM)

System Message (ORM)

```
Judge the quality of a response for the provided task.
The task requires producing `argument` given `topic` and `stance`.

Evaluate the response using the rubric items below and assign numeric scores
to each:
- persuasiveness: Persuasiveness score (an int between 1 and 7).
- coherence: Coherence score (an int between 1 and 7).
- relevance: Relevance score (an int between 1 and 7).

Your inputs will be:
1. `topic` (str): The topic to generate an argument about
2. `stance` (Literal['PRO', 'ANTI']): The stance to take on the topic
3. `argument` (str): The generated argument

Your goal is to produce the following outputs:
1. `persuasiveness` (int): Persuasiveness score
        Constraints: >= 1 and <= 7
2. `coherence` (int): Coherence score
        Constraints: >= 1 and <= 7
3. `relevance` (int): Relevance score
        Constraints: >= 1 and <= 7

Please provide your response with each output field under its own header using
the format:
## field_name
Your response for that field here

NOTE: Line breaks, capitalization, and punctuation (i.e., '##' followed by a
space) are important. If you do not follow these guidelines, your response will
be rejected.
```

Prompt 14: Generative Evaluator — System Message (ORM) Teal: task signature and rubric as rendered by the adapter. Blue: concrete runtime value.

User Message (ORM)

```
## topic
Should single-use plastics be banned?

## stance
PRO

## argument
The proliferation of single-use plastics represents one of the most pressing
environmental crises of our time. Each year, millions of tonnes of plastic waste
enter our oceans, choking marine life and entering the food chain. A
comprehensive ban would catalyse the development of sustainable alternatives
while sending a clear signal to industry and consumers alike.

Respond with the corresponding output fields, starting with the field
`## persuasiveness` (must be formatted as a valid Python int), then
`## coherence` (must be formatted as a valid Python int), then
`## relevance` (must be formatted as a valid Python int)
```

Prompt 15: Generative Evaluator — User Message (ORM)

C.3.2 *Reranker Evaluator*

Analogous to the reranker controller (Appendix C.1.2), we can use a cross-encoder to score candidate states rather than candidate actions. Here, the *query* contains the input x and the evaluation criteria (e.g., coherence, correctness, constraint satisfaction), while each *document* is a candidate reasoning chain Z_i (for intermediate evaluation) or final output y (for outcome evaluation). The reranker assigns a relevance score to each candidate, which we interpret as a quality estimate. This approach is particularly efficient when the number of candidates is large, as cross-encoders can score all candidates in a single batched forward pass without requiring the longer generations that LLM-as-a-Judge evaluators produce. Further, this resolves the sycophancy issues of the generative evaluator, which tends to award perfect scores to all good candidates (Sharma et al., 2024).

Unlike the generative evaluator, which produces explicit integer scores that can be directly weighted and summed, the reranker model returns a single relevance logit. STATe extracts this logit by computing the log-probability of the token `yes` (vs. `no`) from the model's output distribution and applies a softmax to convert logit differences into a scalar in $(0, 1)$. This scalar serves directly as the candidate's quality estimate for beam selection.

System Message

```
Judge whether the Document meets the requirements based on the Query and the
Instruct provided. Note that the answer can only be `yes' or `no'.
```

Prompt 16: Reranker Evaluator — System Message Uses the same Qwen3 query–document format as the reranker controller (Zhang et al., 2025b). The static system prompt is shared across PRM and ORM; the user message carries the instruction, rubric, query (inputs), and document (candidate output or reasoning trajectory). Teal: task signature and rubric. Blue: concrete runtime value.

User Message (PRM)

```
<Instruct>: Your objective is to judge a reasoning trajectory towards solving a
user-assigned task.

The user provided the following task:
"Generate an argument which takes the provided stance towards the provided
topic."

Since this is a reasoning task, we are interested not only in the final
output, but also in the reasoning process that leads to it.
You will find the inputs for this task under the # Inputs header in the
Query.
You will find the intermediate reasoning trajectory towards solving this
problem under the # Reasoning header in the Document.

Judge whether the provided reasoning trajectory is a strong partial solution
for the task given the inputs, using the rubric below.
Rubric:
- persuasiveness: Persuasiveness score
- coherence: Coherence score
- relevance: Relevance score

<Query>: # Inputs
topic: Should single-use plastics be banned?
stance: PRO

<Document>: # Reasoning
<thinking>
<step>
## claim
Single-use plastics are a leading contributor to ocean pollution, harming
marine ecosystems and entering the food chain.
</step>
</thinking>
```

Prompt 17: Reranker Evaluator — User Message (PRM)

User Message (ORM)

```
<Instruct>: Your objective is to judge an output for a user-assigned task.

The user provided the following task:
"Generate an argument which takes the provided stance towards the provided topic."

You will find the inputs for this task under the # Inputs header in the Query.
You will find the output under consideration under the # Output heading in the
Document.

Judge whether the provided output is a strong final output for the task given
the inputs, using the rubric below.
Rubric:
- persuasiveness: Persuasiveness score
- coherence: Coherence score
- relevance: Relevance score

<Query>: # Inputs
topic: Should single-use plastics be banned?
stance: PRO

<Document>: # Output
argument: The proliferation of single-use plastics represents one of the most
pressing environmental crises of our time. Each year, millions of tonnes of
plastic waste enter our oceans, choking marine life and entering the food chain.
A comprehensive ban would catalyse the development of sustainable alternatives
while sending a clear signal to industry and consumers alike.
```

Prompt 18: Reranker Evaluator — User Message (ORM)

C.3.3 Programmatic Evaluator

In some domains, intermediate and final states admit *programmatic* evaluation: a deterministic procedure can verify correctness, constraint satisfaction, or structural validity without invoking an LLM (Lambert et al., 2025; Gao et al., 2024; Team et al., 2025b). This setting crucially assumes an *additive action space*, where each reasoning step produces text that is concatenated to all previous steps, so that a partial trajectory $Z_i = [z_1, \ldots, z_i]$ represents a prefix of a well-formed candidate solution. Many instruction-following and mathematical tasks satisfy this property, as successive steps monotonically construct a single output whose validity can be checked incrementally (e.g., JSON well-formedness, exact string constraints, section counts, or arithmetic consistency). This stands in contrast to *metacognitive* action spaces – such as self-reflection, critique, or targeted editing of an existing draft – where actions do not compose into a single executable artifact and intermediate states cannot be interpreted as partial answers. As a result, programmatic evaluators are inherently task-dependent and cannot be assumed to exist for all domains. Formally, we define a *Programmatic Evaluator* as a deterministic scoring function conditioned on the input x:[14]

$$V^*_{\text{PRM}}(\text{concat}(Z_i) \mid x) \to [0, 1] \tag{5}$$

$$V^*_{\text{ORM}}(y \mid x) \to [0, 1] \tag{6}$$

which evaluates whether the concatenated reasoning ($\text{concat}(Z_i)$) or answer (y) satisfies all constraints induced by the task, x.[15] In tasks where constraint satisfaction is prefix-monotonic, V^* can be used interchangeably as both a Process Reward Model and an Outcome Reward Model, i.e.,

$$V^*_{\text{PRM}} \equiv V^*_{\text{ORM}} \equiv V^*,$$

allowing invalid trajectories to be pruned immediately upon violation. When available, programmatic evaluators eliminate judge variance, avoid sycophancy effects, and provide exact credit assignment over the action space. However, their applicability fundamentally relies on additive action spaces and reliable prefix-level validation; extending programmatic evaluation to non-additive or revision-based action spaces remains an open challenge.

[14] It is also possible to condition the ORM on $\text{concat}(x, Z_i)$ as opposed to just conditioning on x.

[15] In practice, the concatenation operation ignores intermediate or internal reasoning fields (e.g., chain-of-thought or planning annotations), retaining only the externally visible output fields. Including internal reasoning in the concatenation would often invalidate otherwise correct partial outputs and interfere with reliable programmatic grading.

D Beam-Search Complexity

STATe supports both full-tree exploration (no pruning) and heuristic-based tree search (Newell & Simon, 1976) in the form of Beam Search (Lowerre, 1976). This section analyzes the search-space complexity under both regimes and discusses the implementation-level effects of batching with vLLM (Kwon et al., 2023).

Notation. Let d denote the maximum reasoning depth, n the branching factor, and k the beam width. Let L_i denote the frontier after i reasoning steps, with $b_i := |L_i|$ and $b_0 = 1$ (contains root only).

One layer. At each non-terminal layer $i < d$, every state in L_i is expanded into n children,[16] producing a candidate pool of size $b_i \cdot n$. If pruning is disabled, all candidates survive to the next layer. If pruning is enabled, the evaluator scores the candidates and the selector retains the top-k.

We first analyze the unpruned case to establish the baseline exponential growth, and then show how beam-search pruning reduces this scaling.

Full-tree expansion (no pruning). Without pruning, every candidate survives, so the frontier grows as $b_i = n^i$. Each layer requires b_i controller invocations (to select n actions per state) and $b_i \cdot n$ generator completions (one per child), giving a total intermediate generation cost of

$$N_{\text{gen}}^{\text{inter}} = \sum_{i=0}^{d-1} n^{i+1} = \begin{cases} dn, & n = 1, \\ \dfrac{n(n^d - 1)}{n - 1}, & n > 1. \end{cases} \tag{7}$$

The controller cost matches the generator cost for a generative controller (n calls per state), but may be higher for a reranker controller that scores all $|\mathcal{A}_i|$ actions per state (see below). Our implementation does *not* score non-terminal nodes when pruning is disabled; intermediate PRM-style evaluation is skipped, and only nodes containing final outputs are evaluated. The practical implication is that disabling pruning converts the method from beam search into explicit tree expansion with exponential growth in depth.

Beam search (with pruning). With pruning, the frontier is truncated to at most k states after each layer, so $b_i \leq k$ for all $i \geq 1$. Under a generative controller, all three components (generator, evaluator, controller) scale with the same leading term for intermediate layers:

$$N_{\text{gen}}^{\text{inter}},\ N_{\text{eval}}^{\text{inter}},\ N_{\text{ctrl}}^{\text{gen}} \ \leq\ n \sum_{i=0}^{d-1} b_i \leq n\big(1 + (d-1)\,k\big). \tag{8}$$

Under pruning, the logical search cost thus grows *linearly* in depth and beam width, rather than exponentially in depth. This is the standard computational advantage of beam search, now instantiated over reasoning trajectories.

Reranker controller cost. The bounds above assume a generative controller, which produces n action choices per state. In contrast, a reranker controller scores *all* admissible actions individually, so its cost at layer i is $b_i \cdot m_i$ where $m_i \leq |\mathcal{A}_i|$ is the number of admissible actions. Hence, reranker-controller complexity is governed not only by beam width but also by action-space size:

$$N_{\text{ctrl}}^{\text{rank}} = \sum_{i=0}^{d-1} b_i\, m_i, \qquad m_i \leq |\mathcal{A}_i|. \tag{9}$$

Final synthesis step. At maximum depth d, the implementation sets every remaining frontier state to `FINISH` and generates n_{final} outputs per surviving trajectory. The final generation and evaluation costs are

$$N_{\text{gen}}^{\text{final}} = n_{\text{final}}\, b_d, \qquad N_{\text{eval}}^{\text{final}} = n_{\text{final}}\, b_d. \tag{10}$$

Under pruning, $b_d \leq k$, so both costs are at most $n_{\text{final}} \cdot k$. Without pruning, $b_d = n^d$ in the worst case.

[16] It is possible that some generations fail, and that in practice there are fewer than n children for a given parent. For our complexity analysis, which concerns an upper bound, we assume zero failures

Batched implementation calls. The expressions above count *logical* state-level interactions. In practice, the implementation batches all states in a frontier into a single vLLM call per layer. In the worst case (no early stopping), there are at most

$$B_{\text{ctrl}} \leq d, \qquad B_{\text{gen}} \leq d+1, \qquad B_{\text{eval}} \leq d+1 \tag{11}$$

batched forward passes for the controller, generator, and evaluator, respectively. The extra pass comes from the final synthesis step; the controller is not called during that step since `FINISH` is set deterministically. These bounds explain why batch inference remains practical even when the logical number of trajectories is large: the number of batched model calls grows linearly with depth and is independent of beam width at the level of batched calls.

Early stopping and special cases. When early stopping is enabled, the controller may emit `FINISH` before depth d, which can only reduce the realized cost relative to the bounds above. All expressions in this section should therefore be interpreted as worst-case upper bounds.

Several common decoding procedures emerge as boundary cases. With $d = 1$ and no pruning, the procedure reduces to Best-of-n over a single reasoning step. With $d = 1$ and $k < n$, it becomes a one-step beam search. With larger d and pruning, it yields multi-step beam search over partial trajectories.

Overall, STATe does not introduce a new asymptotic search primitive; rather, it instantiates beam search over *structured reasoning actions and partial trajectories*. The resulting complexity is controlled by the same core quantities as classical beam search (depth d, branching factor n, beam width k), together with three STATe-specific factors: the first-layer beam width k_1 (when set independently of k), the action-space size $|\mathcal{A}|$ (for reranker controllers), and the number of final responses n_{final}.

E Complete Experiments

We conduct all runs with vLLM (Kwon et al., 2023) in offline mode to enable efficient layer-wise batching across tree expansions. Baseline methods (including ToT) require 1 GPU to generate outputs, while STATe requires 2 GPUs. When using 2 GPUs for STATe, one is meant for the generative LLM that serves the Generator (Appendix C.2) and the other is meant for the reranker LLM that serves the Controller (Appendix C.1.2) and Reranker (Appendix C.3.2).

E.1 NoveltyBench

We evaluate on the curated NoveltyBench set of 100 prompts spanning four categories: *randomness*, *factual knowledge* with underspecified queries, *creative writing*, and *subjectivity*. Because NoveltyBench provides only a single "test" split, we use the first 10 prompts as a development subset to refine prompts and system settings, and report final results on the remaining 90 prompts.

We compare Best-of-n baselines (I/O and CoT), ToT, and STATe across seven models from four families: Qwen3 (Yang et al., 2025), Gemma-3 (Team et al., 2025a), Nemotron-3 (NVIDIA et al., 2025), and Ministral-3 (Liu et al., 2026a). For STATe, we use Qwen3-8B-Reranker (Zhang et al., 2025b) for both the reranker controller (Appendix C.1.2) and reranker evaluator (Appendix C.3.2). We run depth-1 trees with wide branching ($n = 8, k = 8$), so each candidate includes one reasoning step and one final answer, and repeat each configuration over 10 random seeds. We focus on a single branching operation since deeper heuristic search optimizes for evaluator-aligned scores rather than frontier diversity, and deeper trajectories often share parent states (which introduces overlapping reasoning). Therefore, diversity as a function of tree depth is outside this experiment's scope.

To isolate the impact of controller-guided interventions, we also include baselines that expose the same action spaces directly in prompts (as DSPy input fields, described in Appendix B). This setup evaluates diversity as a function of (1) choice of reasoning template (I/O vs. CoT vs. ToT), (2) including the action space as an explicit prompt input, and (3) enabling controller-guided interventions. The action space combines two dimensions with 5 choices each, yielding 25 action combinations per step (Appendix H.1): *personality traits* (following the Big Five model; Goldberg, 1990) and *target audience* (demographic age to appeal to). Each action provides internal reasoning guidance (Appendix C.2) that steers generation toward the selected persona or audience. In this configuration, STATe's internal evaluator does not affect search or final selection because all outputs are returned. We sweep low-, medium-, and high-temperature regimes per model: $T \in \{0.5, 0.7, 1.0\}$ for most models and $T \in \{0.1, 0.3, 0.5\}$ for Ministral-3 (consistent with provider recommendations).

Metrics. Diversity is measured as Mean Distinct$_k$ (Zhang et al., 2025c), which counts the number of meaningfully distinct responses a model generates in k samples (where two responses are considered distinct if a user would benefit from seeing both). Zhang et al. (2025c) measure diversity through clustering responses with a fine-tuned DeBERTa model (He et al., 2021). NoveltyBench also computes the "quality" of a response through an LLM-as-a-Judge score (Liu et al., 2026b), which is assigned to each individual response. Utility is measured as Mean Utility$_k$ (Zhang et al., 2025c), which formalizes the cumulative benefit to a user who sequentially inspects k generations and only gains value from a response when it is distinct from all previously seen ones. Formally, $\text{Utility}_k = \sum_{i=1}^{k} q_i \cdot \mathbb{1}[\text{response } i \text{ is distinct from responses } 1, \ldots, i-1]$, where q_i is the quality score of the i-th response, and responses are distinct if they are assigned to different clusters (in our case, by DeBERTa embeddings). Utility thus jointly penalizes low quality *and* redundancy: a method can only accumulate high utility by generating responses that are both good and genuinely new. Tables 5, 6, and 7 report all three metrics across all model and temperature configurations.

E.1.1 Generalizability Across Models

STATe achieves the highest diversity across all seven models and all three temperature regimes (Table 5). Relative to Best-of-n with I/O prompting, the simplest and most common ITC method, STATe roughly doubles diversity at medium temperature across most models. STATe's gains range from +49% for Qwen3-8B (4.63 vs. 3.11) to +153% for Gemma-3-27B (4.35 vs. 1.72), with a mean improvement of +95% across the seven models. These gains reflect a systematic limitation of I/O sampling: without structural interventions, repeated high-temperature sampling collapses toward the same few high-probability completions.

Relative to the strongest non-STATe baseline (which varies by model), the margin is more modest but consistent. Gains are largest for models in the mid-capability range—+37% for Qwen3-30B (4.57 vs. 3.33 for CoT w/ Action Space), +35% for Gemma-3-4B (3.77 vs. 2.79), +33% for Gemma-3-27B (4.35 vs. 3.26)—and smaller for models where action-space-augmented baselines already achieve strong

diversity on their own: +7% for Nemotron-3-30B (5.39 vs. 5.05 for ToT w/ Action Space) and +5% for Ministral-3-14B (4.86 vs. 4.64 for CoT w/ Action Space). The consistent pattern across all models is that controller-guided prefix interventions provide a meaningful additional boost beyond simply exposing the action space in the prompt: "CoT w/ Action Space" and "ToT w/ Action Space" always trail STATe despite having access to the same action vocabulary, confirming that it is the *enforcement* of actions via prefilling—not their mere presence in the context—that drives diversity.

Model	Method	Low	Medium	High
Ministral 3 14B	Baseline	1.96 ± 0.05	2.76 ± 0.08	3.52 ± 0.07
	Baseline CoT	3.16 ± 0.09	3.79 ± 0.06	4.28 ± 0.12
	Baseline w/ Action Space	2.56 ± 0.07	3.55 ± 0.1	4.5 ± 0.08
	Baseline CoT w/ Action Space	4.08 ± 0.1	4.64 ± 0.1	4.95 ± 0.11
	Baseline ToT	2.91 ± 0.09	3.73 ± 0.1	4.53 ± 0.09
	Baseline ToT w/ Action Space	3.67 ± 0.1	4.48 ± 0.09	5.2 ± 0.12
	STATe of Thoughts	$\mathbf{4.66 \pm 0.12}$	$\mathbf{4.86 \pm 0.07}$	$\mathbf{5.29 \pm 0.08}$
Qwen3 4B	Baseline	1.89 ± 0.06	2.26 ± 0.08	2.73 ± 0.08
	Baseline CoT	2.68 ± 0.11	2.87 ± 0.09	3.25 ± 0.11
	Baseline w/ Action Space	1.85 ± 0.06	2.16 ± 0.06	2.66 ± 0.08
	Baseline CoT w/ Action Space	3.18 ± 0.1	3.38 ± 0.09	3.76 ± 0.09
	Baseline ToT	2.31 ± 0.06	2.65 ± 0.06	3.11 ± 0.1
	Baseline ToT w/ Action Space	2.54 ± 0.09	2.91 ± 0.11	3.41 ± 0.06
	STATe of Thoughts	$\mathbf{3.73 \pm 0.08}$	$\mathbf{4.1 \pm 0.13}$	$\mathbf{4.67 \pm 0.06}$
Qwen3 8B	Baseline	2.91 ± 0.07	3.11 ± 0.12	3.24 ± 0.11
	Baseline CoT	3.14 ± 0.08	3.27 ± 0.1	3.46 ± 0.06
	Baseline w/ Action Space	3.19 ± 0.07	3.32 ± 0.1	3.58 ± 0.06
	Baseline CoT w/ Action Space	3.19 ± 0.1	3.38 ± 0.09	3.58 ± 0.1
	Baseline ToT	2.47 ± 0.09	2.83 ± 0.06	3.32 ± 0.12
	Baseline ToT w/ Action Space	3.16 ± 0.08	3.6 ± 0.11	4.22 ± 0.1
	STATe of Thoughts	$\mathbf{4.31 \pm 0.1}$	$\mathbf{4.63 \pm 0.07}$	$\mathbf{5.28 \pm 0.07}$
Qwen3 30B	Baseline	1.68 ± 0.05	1.98 ± 0.03	2.41 ± 0.05
	Baseline CoT	2.31 ± 0.06	2.59 ± 0.09	3.0 ± 0.1
	Baseline w/ Action Space	1.94 ± 0.05	2.26 ± 0.1	2.84 ± 0.09
	Baseline CoT w/ Action Space	2.98 ± 0.09	3.33 ± 0.12	3.76 ± 0.1
	Baseline ToT	1.97 ± 0.05	2.27 ± 0.05	2.78 ± 0.11
	Baseline ToT w/ Action Space	2.38 ± 0.06	2.76 ± 0.08	3.29 ± 0.11
	STATe of Thoughts	$\mathbf{4.24 \pm 0.11}$	$\mathbf{4.57 \pm 0.13}$	$\mathbf{4.94 \pm 0.1}$
Gemma 3 4B	Baseline	1.58 ± 0.03	1.78 ± 0.09	2.14 ± 0.05
	Baseline CoT	2.25 ± 0.08	2.47 ± 0.07	2.72 ± 0.08
	Baseline w/ Action Space	1.85 ± 0.06	2.17 ± 0.06	2.7 ± 0.08
	Baseline CoT w/ Action Space	2.53 ± 0.09	2.79 ± 0.08	3.12 ± 0.08
	Baseline ToT	1.86 ± 0.05	2.16 ± 0.06	2.6 ± 0.08
	Baseline ToT w/ Action Space	2.12 ± 0.09	2.5 ± 0.08	2.92 ± 0.09
	STATe of Thoughts	$\mathbf{3.55 \pm 0.07}$	$\mathbf{3.77 \pm 0.05}$	$\mathbf{4.09 \pm 0.08}$
Gemma 3 27B	Baseline	1.54 ± 0.04	1.72 ± 0.05	2.04 ± 0.05
	Baseline CoT	2.35 ± 0.07	2.55 ± 0.11	2.86 ± 0.09
	Baseline w/ Action Space	1.81 ± 0.04	2.05 ± 0.08	2.47 ± 0.1
	Baseline CoT w/ Action Space	3.03 ± 0.09	3.26 ± 0.07	3.59 ± 0.05
	Baseline ToT	2.13 ± 0.07	2.49 ± 0.08	2.88 ± 0.08
	Baseline ToT w/ Action Space	2.69 ± 0.08	3.0 ± 0.09	3.44 ± 0.09
	STATe of Thoughts	$\mathbf{4.21 \pm 0.08}$	$\mathbf{4.35 \pm 0.09}$	$\mathbf{4.66 \pm 0.09}$
Nemotron 3 30B	Baseline	2.74 ± 0.08	3.25 ± 0.07	4.32 ± 0.13
	Baseline CoT	3.28 ± 0.09	3.58 ± 0.1	4.33 ± 0.12
	Baseline w/ Action Space	3.82 ± 0.1	4.73 ± 0.15	5.91 ± 0.12
	Baseline CoT w/ Action Space	4.38 ± 0.09	4.82 ± 0.14	5.71 ± 0.15
	Baseline ToT	3.25 ± 0.1	3.82 ± 0.09	4.78 ± 0.07
	Baseline ToT w/ Action Space	4.49 ± 0.11	5.05 ± 0.15	6.05 ± 0.09
	STATe of Thoughts	$\mathbf{4.81 \pm 0.11}$	$\mathbf{5.39 \pm 0.12}$	$\mathbf{6.28 \pm 0.12}$

Table 5: NoveltyBench Mean Distinct (diversity) for all models (mean±std over seeds). Low, medium, and high temperature correspond to $T = 0.1, 0.3, 0.5$ for Ministral 3 14B, and $T = 0.5, 0.7, 1.0$ for others. Best and second best per model per column are **bolded** and underlined.

E.1.2 Quality and Utility

Quality. STATe achieves the highest raw quality score (Table 6) across all seven models and all temperature settings without exception, demonstrating that diversity gains do not come at the expense of individual response quality. Relative to I/O Best-of-n, quality improvements at medium temperature range from +13% for Qwen3-8B (3.29 vs. 2.90) to +89% for Gemma-3-27B (3.46 vs. 1.83),

with a mean of +60% across models. This is particularly notable because I/O sampling is optimized to maximize the quality of the single most likely response, yet STATe's structurally diversified outputs surpass it on quality as well. When we introduce an inductive bias in the form of our action space, quality tends to improve on both baselines and STATe.

Model	Method	Low	Medium	High
Ministral 3 14B	Baseline	2.09 ± 0.05	2.9 ± 0.09	3.74 ± 0.07
	Baseline CoT	3.12 ± 0.09	3.74 ± 0.05	4.23 ± 0.13
	Baseline w/ Action Space	2.39 ± 0.06	3.17 ± 0.11	3.84 ± 0.08
	Baseline CoT w/ Action Space	3.84 ± 0.11	4.37 ± 0.12	4.66 ± 0.14
	Baseline ToT	2.9 ± 0.07	3.74 ± 0.11	4.53 ± 0.12
	Baseline ToT w/ Action Space	3.05 ± 0.13	3.66 ± 0.07	4.24 ± 0.09
	STATe of Thoughts	$\mathbf{4.81 \pm 0.12}$	$\mathbf{5.01 \pm 0.1}$	$\mathbf{5.4 \pm 0.1}$
Qwen3 4B	Baseline	1.6 ± 0.05	1.89 ± 0.07	2.2 ± 0.06
	Baseline CoT	2.25 ± 0.09	2.4 ± 0.09	2.71 ± 0.08
	Baseline w/ Action Space	1.44 ± 0.05	1.66 ± 0.04	1.98 ± 0.06
	Baseline CoT w/ Action Space	2.58 ± 0.09	2.74 ± 0.1	3.0 ± 0.1
	Baseline ToT	1.96 ± 0.05	2.24 ± 0.08	2.56 ± 0.1
	Baseline ToT w/ Action Space	1.94 ± 0.07	2.21 ± 0.07	2.52 ± 0.07
	STATe of Thoughts	$\mathbf{2.67 \pm 0.09}$	$\mathbf{2.87 \pm 0.11}$	$\mathbf{3.23 \pm 0.06}$
Qwen3 8B	Baseline	2.69 ± 0.09	2.9 ± 0.12	3.01 ± 0.1
	Baseline CoT	2.89 ± 0.12	2.97 ± 0.11	3.14 ± 0.07
	Baseline w/ Action Space	2.6 ± 0.05	2.69 ± 0.09	2.82 ± 0.08
	Baseline CoT w/ Action Space	2.73 ± 0.09	2.9 ± 0.08	3.08 ± 0.13
	Baseline ToT	1.98 ± 0.06	2.25 ± 0.07	2.62 ± 0.08
	Baseline ToT w/ Action Space	1.9 ± 0.09	2.17 ± 0.08	2.52 ± 0.08
	STATe of Thoughts	$\mathbf{3.14 \pm 0.09}$	$\mathbf{3.29 \pm 0.08}$	$\mathbf{3.64 \pm 0.09}$
Qwen3 30B	Baseline	1.67 ± 0.05	1.9 ± 0.04	2.25 ± 0.05
	Baseline CoT	2.13 ± 0.06	2.31 ± 0.08	2.66 ± 0.11
	Baseline w/ Action Space	1.69 ± 0.04	1.91 ± 0.1	2.37 ± 0.09
	Baseline CoT w/ Action Space	2.59 ± 0.08	2.9 ± 0.1	3.23 ± 0.1
	Baseline ToT	1.72 ± 0.06	1.99 ± 0.06	2.4 ± 0.08
	Baseline ToT w/ Action Space	1.99 ± 0.06	2.32 ± 0.06	2.7 ± 0.12
	STATe of Thoughts	$\mathbf{3.36 \pm 0.09}$	$\mathbf{3.52 \pm 0.08}$	$\mathbf{3.73 \pm 0.09}$
Gemma 3 4B	Baseline	1.47 ± 0.04	1.63 ± 0.05	1.96 ± 0.05
	Baseline CoT	1.92 ± 0.08	2.08 ± 0.09	2.28 ± 0.05
	Baseline w/ Action Space	1.58 ± 0.06	1.84 ± 0.04	2.26 ± 0.07
	Baseline CoT w/ Action Space	2.1 ± 0.12	2.34 ± 0.06	2.6 ± 0.07
	Baseline ToT	1.56 ± 0.06	1.78 ± 0.06	2.1 ± 0.08
	Baseline ToT w/ Action Space	1.69 ± 0.09	1.95 ± 0.06	2.24 ± 0.07
	STATe of Thoughts	$\mathbf{2.53 \pm 0.08}$	$\mathbf{2.69 \pm 0.06}$	$\mathbf{2.84 \pm 0.08}$
Gemma 3 27B	Baseline	1.67 ± 0.04	1.83 ± 0.05	2.13 ± 0.05
	Baseline CoT	2.18 ± 0.07	2.35 ± 0.1	2.63 ± 0.08
	Baseline w/ Action Space	1.76 ± 0.03	2.01 ± 0.07	2.35 ± 0.07
	Baseline CoT w/ Action Space	2.77 ± 0.08	3.01 ± 0.07	3.35 ± 0.06
	Baseline ToT	1.94 ± 0.07	2.24 ± 0.05	2.58 ± 0.06
	Baseline ToT w/ Action Space	2.41 ± 0.07	2.67 ± 0.06	2.99 ± 0.07
	STATe of Thoughts	$\mathbf{3.39 \pm 0.1}$	$\mathbf{3.46 \pm 0.1}$	$\mathbf{3.75 \pm 0.08}$
Nemotron 3 30B	Baseline	2.48 ± 0.07	2.87 ± 0.07	3.6 ± 0.09
	Baseline CoT	2.97 ± 0.08	3.2 ± 0.1	3.86 ± 0.09
	Baseline w/ Action Space	2.87 ± 0.1	3.47 ± 0.09	3.74 ± 0.15
	Baseline CoT w/ Action Space	3.39 ± 0.12	3.66 ± 0.13	4.04 ± 0.17
	Baseline ToT	2.66 ± 0.11	3.14 ± 0.07	3.76 ± 0.08
	Baseline ToT w/ Action Space	3.38 ± 0.12	3.79 ± 0.1	4.1 ± 0.1
	STATe of Thoughts	$\mathbf{3.7 \pm 0.11}$	$\mathbf{4.02 \pm 0.12}$	$\mathbf{4.21 \pm 0.11}$

Table 6: NoveltyBench Mean Quality (raw LLM-as-a-Judge score) for all models (mean±std over seeds). Low, medium, and high temperature correspond to $T = 0.1, 0.3, 0.5$ for Ministral 3 14B, and $T = 0.5, 0.7, 1.0$ for others. Best and second best per model per column are **bolded** and underlined.

Utility. Utility results (Table 7) tell a similar story. Relative to I/O Best-of-n at medium temperature, STATe improves utility by +33% for Ministral-3-14B (6.53 vs. 4.90), +33% for Qwen3-30B (5.12 vs. 3.84), +30% for Gemma-3-27B (5.08 vs. 3.90), and +19% for Nemotron-3-30B (5.62 vs. 4.73). STATe ranks first on five of the seven models across most temperatures.

The two exceptions are the smallest models in each family. For Qwen3-4B, CoT w/ Action Space achieves higher utility at low and medium temperature (4.37 and 4.51 vs. STATe's 4.27 and 4.40), and for Gemma-3-4B the margin is negligible at high temperature (4.47 vs. 4.45). Both cases share a likely cause: STATe's reranker controller orders responses by proximity to its top-ranked action preference.

As a result, high-ranked controller outputs tend to cluster around a dominant action choice (e.g., most actions preferring the same target audience). Because utility accumulates sequentially and is discounted for redundant responses, this ordering effect can understate STATe's utility relative to methods that return responses in arbitrary order. Crucially, the raw quality tables show STATe is strongest everywhere, confirming this is an ordering artifact rather than a genuine quality deficit.

Model	Method	Low	Medium	High
Ministral 3 14B	Baseline	4.12 ± 0.04	4.9 ± 0.08	5.67 ± 0.06
	Baseline CoT	5.01 ± 0.09	5.59 ± 0.06	5.99 ± 0.11
	Baseline w/ Action Space	4.11 ± 0.06	4.84 ± 0.11	5.42 ± 0.09
	Baseline CoT w/ Action Space	5.62 ± 0.12	6.03 ± 0.08	6.28 ± 0.14
	Baseline ToT	4.85 ± 0.1	5.62 ± 0.14	6.32 ± 0.14
	Baseline ToT w/ Action Space	4.8 ± 0.14	5.32 ± 0.07	5.78 ± 0.12
	STATe of Thoughts	$\mathbf{6.34 \pm 0.11}$	$\mathbf{6.53 \pm 0.1}$	$\mathbf{6.88 \pm 0.14}$
Qwen3 4B	Baseline	3.52 ± 0.05	3.79 ± 0.08	4.07 ± 0.07
	Baseline CoT	4.12 ± 0.07	4.24 ± 0.1	4.55 ± 0.07
	Baseline w/ Action Space	3.23 ± 0.06	3.47 ± 0.05	3.75 ± 0.07
	Baseline CoT w/ Action Space	$\mathbf{4.37 \pm 0.08}$	$\mathbf{4.51 \pm 0.1}$	4.72 ± 0.09
	Baseline ToT	3.84 ± 0.07	4.12 ± 0.08	4.39 ± 0.1
	Baseline ToT w/ Action Space	3.78 ± 0.08	4.03 ± 0.08	4.3 ± 0.07
	STATe of Thoughts	4.27 ± 0.1	4.4 ± 0.08	$\mathbf{4.75 \pm 0.06}$
Qwen3 8B	Baseline	4.61 ± 0.07	4.78 ± 0.1	4.87 ± 0.09
	Baseline CoT	$\mathbf{4.77 \pm 0.13}$	4.81 ± 0.09	4.99 ± 0.08
	Baseline w/ Action Space	4.34 ± 0.1	4.46 ± 0.1	4.52 ± 0.1
	Baseline CoT w/ Action Space	4.57 ± 0.09	4.7 ± 0.08	4.88 ± 0.13
	Baseline ToT	3.83 ± 0.06	4.1 ± 0.08	4.43 ± 0.09
	Baseline ToT w/ Action Space	3.57 ± 0.1	3.82 ± 0.08	4.14 ± 0.09
	STATe of Thoughts	4.74 ± 0.09	$\mathbf{4.88 \pm 0.09}$	$\mathbf{5.16 \pm 0.09}$
Qwen3 30B	Baseline	3.64 ± 0.06	3.84 ± 0.05	4.17 ± 0.07
	Baseline CoT	4.05 ± 0.05	4.21 ± 0.08	4.53 ± 0.12
	Baseline w/ Action Space	3.56 ± 0.07	3.72 ± 0.08	4.17 ± 0.09
	Baseline CoT w/ Action Space	4.39 ± 0.12	4.68 ± 0.12	4.96 ± 0.11
	Baseline ToT	3.61 ± 0.08	3.88 ± 0.07	4.27 ± 0.07
	Baseline ToT w/ Action Space	3.79 ± 0.08	4.13 ± 0.05	4.47 ± 0.09
	STATe of Thoughts	$\mathbf{4.98 \pm 0.08}$	$\mathbf{5.12 \pm 0.08}$	$\mathbf{5.3 \pm 0.1}$
Gemma 3 4B	Baseline	3.42 ± 0.05	3.57 ± 0.05	3.88 ± 0.05
	Baseline CoT	3.81 ± 0.07	3.95 ± 0.1	4.14 ± 0.03
	Baseline w/ Action Space	3.44 ± 0.08	3.69 ± 0.06	4.1 ± 0.06
	Baseline CoT w/ Action Space	3.98 ± 0.13	4.2 ± 0.08	$\mathbf{4.47 \pm 0.1}$
	Baseline ToT	3.46 ± 0.07	3.71 ± 0.06	3.98 ± 0.09
	Baseline ToT w/ Action Space	3.56 ± 0.08	3.8 ± 0.07	4.06 ± 0.07
	STATe of Thoughts	$\mathbf{4.16 \pm 0.08}$	$\mathbf{4.33 \pm 0.04}$	4.45 ± 0.08
Gemma 3 27B	Baseline	3.73 ± 0.05	3.9 ± 0.05	4.2 ± 0.05
	Baseline CoT	4.15 ± 0.08	4.3 ± 0.13	4.56 ± 0.07
	Baseline w/ Action Space	3.68 ± 0.04	3.93 ± 0.07	4.24 ± 0.09
	Baseline CoT w/ Action Space	4.69 ± 0.08	4.89 ± 0.09	5.21 ± 0.09
	Baseline ToT	3.88 ± 0.05	4.17 ± 0.05	4.51 ± 0.06
	Baseline ToT w/ Action Space	4.26 ± 0.09	4.49 ± 0.08	4.77 ± 0.06
	STATe of Thoughts	$\mathbf{4.97 \pm 0.09}$	$\mathbf{5.08 \pm 0.1}$	$\mathbf{5.33 \pm 0.07}$
Nemotron 3 30B	Baseline	4.36 ± 0.07	4.73 ± 0.08	5.36 ± 0.11
	Baseline CoT	4.85 ± 0.1	5.06 ± 0.08	5.62 ± 0.11
	Baseline w/ Action Space	4.57 ± 0.11	5.08 ± 0.11	5.14 ± 0.17
	Baseline CoT w/ Action Space	5.05 ± 0.13	5.28 ± 0.12	5.51 ± 0.18
	Baseline ToT	4.52 ± 0.11	4.95 ± 0.06	5.46 ± 0.11
	Baseline ToT w/ Action Space	5.08 ± 0.12	5.4 ± 0.09	5.55 ± 0.11
	STATe of Thoughts	$\mathbf{5.33 \pm 0.13}$	$\mathbf{5.62 \pm 0.1}$	$\mathbf{5.77 \pm 0.12}$

Table 7: NoveltyBench Mean Utility for all models (mean$\pm$std over seeds). Low, medium, and high temperature correspond to $T = 0.1, 0.3, 0.5$ for Ministral 3 14B, and $T = 0.5, 0.7, 1.0$ for others. Best and second best per model per column are **bolded** and underlined.

E.1.3 Token Usage

Table 8 reveals that *input-token* usage increases with prompt complexity: I/O uses the simplest prompt, whereas ToT and STATe require substantially more context. Including the action space in the generator prompt further increases input-token usage, particularly when the action space is large. The CoT variants generate the most explicit *reasoning-output* tokens, with CoT + Actions producing substantially more than the search-based methods. Much of the iterative computation performed by ToT and STATe instead appears as reasoning-stage input, as intermediate search states are repeatedly expanded and evaluated. STATe generates only slightly more reasoning-output tokens than ToT and fewer than ToT + Actions. Finally, *output-token* usage shows that STATe's improvements over the action-conditioned baselines do not result from longer answers. STATe produces fewer final-output tokens than both I/O + Actions and CoT + Actions. This is particularly relevant given the known length bias in LLM-as-a-Judge evaluations (Dubois et al., 2024): STATe's quality and utility advantages cannot be attributed to verbosity.

	Problem	Reasoning		Final	
	Input	Input	Output	Input	Output
Method / preset	cached / total (hit rate)	cached / total (hit rate)	tokens	cached / total (hit rate)	tokens
I/O	149.3 / 160.3 (93.2%)	N/A	N/A	N/A	328.4
I/O + Actions	401.1 / 427.3 (93.9%)	N/A	N/A	N/A	988.5
CoT	175.5 / 184.3 (95.2%)	N/A	955.0	N/A	200.3
CoT + Actions	414.4 / 451.3 (91.8%)	N/A	1565.3	N/A	557.5
ToT	462.2 / 479.3 (96.4%)	5951.3 / 6437.7 (92.4%)	311.0	3829.9 / 3935.6 (97.3%)	232.1
ToT + Actions	704.5 / 750.3 (93.9%)	9506.3 / 10806.0 (88.0%)	591.0	6222.8 / 6349.0 (98.0%)	550.3
STATe	568.2 / 646.4 (87.9%)	4018.3 / 4517.6 (88.9%)	378.4	5475.9 / 5609.8 (97.6%)	494.0

Table 8: **Per-example usage on NoveltyBench with Qwen3-30B.** Input cells report cached tokens, total input tokens, and cache-hit rate in parentheses. Reasoning-output and final-output columns report generated tokens. "N/A" denotes a stage that is not instantiated by the corresponding method. All entries are means over the profiling run. "Actions" denotes configurations in which the action space is included directly in the generator prompt.

Figure 11 relates wall-clock runtime to total token volume, defined as the sum of all input tokens and generated reasoning and final-output tokens. Runtime generally increases with token volume, but token volume does not fully explain the latency differences between methods. In particular, STATe and ToT process similar total token volumes, approximately 11.6K and 11.4K tokens per example, respectively, yet their mean runtimes are 13.48 and 3.20 seconds. The additional latency for STATe arises primarily from its reranker controller. ToT and the other baselines do not invoke this separate controller, so STATe incurs controller-inference latency beyond the cost represented by generator-side token volume alone.

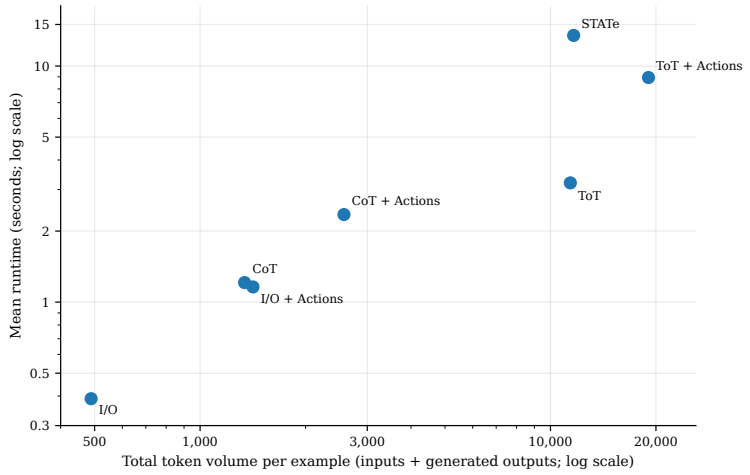

Figure 11: **Runtime versus token volume on NoveltyBench.** Each point denotes an inference method or preset. Total token volume is the sum of all input tokens and generated reasoning and final-output tokens per example. Both axes use logarithmic scales.

Prompt caching nevertheless eliminates most repeated input processing. Approximately 93.4% of STATe's input tokens are cache hits. This is close to ToT's aggregate cache-hit rate of approximately 94.4% and higher than the approximately 91.8% rate obtained by ToT + Actions. At the stage level, STATe obtains cache-hit rates of 87.9% for the problem input, 88.9% for reasoning input, and 97.6% for final input. The near-complete cache reuse during final synthesis indicates that STATe efficiently reuses its long shared prefixes. Consequently, STATe's additional runtime is better explained by its controller calls than by broadly inefficient prompt-cache use.

E.2 Controllability Evaluation

We evaluate controllability on a subset of 1,000 arguments generated on a fixed topic. An LLM judge (GPT-5-mini; Singh et al., 2025) performs up to 13 boolean checks per argument, split across two separate calls to prevent cross-contamination between the reasoning trace and the final output.

E.2.1 Step-Level Evaluation Prompt

The step-level call receives only the three reasoning steps (the final argument is withheld). For each step, it checks two properties: (i) whether the step exhibits its prescribed discourse structure (verified via a prescribed prefix word) and (ii) whether the step discusses its prescribed subtopic (verified via the guidance description). This yields 6 boolean checks in total.

System Message

```
You are an expert evaluator of argumentative text. You will be given text
along with descriptions of prescribed properties. For each check, determine
whether the text exhibits the described property.

Each property may include:
- A 'prefix': the prescribed opening word(s) that the text was instructed
  to start with (e.g., 'However', 'Therefore').
- A 'guidance': the internal instruction given to the model describing what
  angle or lens to reason through. The text should reflect this guidance
  in its content and framing.

Respond with a JSON object containing exactly the keys specified, each with
a boolean value (true or false). Do not include any other text.
```

Prompt 19: Controllability Step-Level Judge — System Message

User Message

```
Topic: {topic}
Stance: {stance}

=== Step 1 ===
{step_1_text}

=== Step 2 ===
{step_2_text}

=== Step 3 ===
{step_3_text}

=== Evaluation Checks ===
For each check below, respond true if the text clearly exhibits the
described property, or false otherwise.

step_1_has_structure: Does Step 1 exhibit the discourse structure
  '{structure_1}' -- defined as: '{structure_1_def}'
  (prescribed prefix: '{structure_1_prefix}')?
step_1_has_subtopic: Does Step 1 discuss the subtopic
  '{subtopic_1}' -- defined as: '{subtopic_1_def}'
  (guidance: '{subtopic_1_guidance}')?
step_2_has_structure: ...
step_2_has_subtopic: ...
step_3_has_structure: ...
step_3_has_subtopic: ...

Respond with a JSON object containing exactly these 6 keys, each
with a boolean value:
step_1_has_structure, step_1_has_subtopic, step_2_has_structure,
step_2_has_subtopic, step_3_has_structure, step_3_has_subtopic
```

Prompt 20: Controllability Step-Level Judge — User Message Teal: action-space configuration (structure definitions, subtopic descriptions, prescribed prefixes). Blue: concrete inference-time values.

E.2.2 Final-Level Evaluation Prompt

The final-level call receives only the finished argument (the reasoning steps are withheld). It checks whether the argument reflects each step's prescribed structure and subtopic, and whether the prescribed step ordering is preserved in the argument. This yields 7 boolean checks: 6 mirroring the step-level checks (applied to the final text) plus one order-preservation check.

User Message

```
Topic: {topic}
Stance: {stance}

=== Argument ===
{argument_text}

=== Prescribed Properties ===
This argument was generated using three sequential reasoning steps. Each
step was prescribed a discourse structure and a subtopic. Evaluate whether
the argument reflects these prescriptions.

Step 1 prescription -- structure: '{structure_1}' ('{structure_1_def}',
  prefix: '{structure_1_prefix}'), subtopic: '{subtopic_1}'
  ('{subtopic_1_def}', guidance: '{subtopic_1_guidance}')
Step 2 prescription -- ...
Step 3 prescription -- ...

=== Evaluation Checks ===
For each check below, respond true if the argument clearly exhibits the
described property, or false otherwise.

final_has_structure_1: Does the argument contain content reflecting the
  structure '{structure_1}'?
final_has_subtopic_1: Does the argument contain content reflecting the
  subtopic '{subtopic_1}'?
final_has_structure_2: ...
final_has_subtopic_2: ...
final_has_structure_3: ...
final_has_subtopic_3: ...
final_preserves_order: Does the argument contain all three steps'
  prescribed content, presented in the correct order (Step 1 material
  appears before Step 2, which appears before Step 3)?

Respond with a JSON object containing exactly these 7 keys, each with a
boolean value:
final_has_structure_1, final_has_subtopic_1, final_has_structure_2,
final_has_subtopic_2, final_has_structure_3, final_has_subtopic_3,
final_preserves_order
```

Prompt 21: Controllability Final-Level Judge — User Message The same system message as Prompt 19 is reused. Teal: action-space configuration. Blue: concrete inference-time values.

E.2.3 Controllability Results Across Synthesis Modes

The three synthesis modes (Appendix C.2.1) serve as ablations along a control-quality continuum. **Strict** synthesis produces near-verbatim concatenations of reasoning steps, yielding the strongest attribution between actions and output text. **Faithful** synthesis permits light rephrasing while preserving the structure and ordering of reasoning steps. **Restructured** synthesis allows free reorganization of the reasoning content, providing the weakest action attribution but the best prose fluency. This gradient reveals how synthesis flexibility affects controllability: as the model is given more freedom to deviate from the reasoning trace, step-level fidelity may be preserved while final-argument fidelity degrades.

Table 9 extends Table 2 with pass rate breakdowns across all three synthesis modes, and Figure 12 provides further detail by action space dimension. All evaluations use plastic pollution as a case study. Strict synthesis achieves the highest pass rates overall, as expected. However, the envisioned gradient from strict to restructured does not fully manifest: faithful synthesis unexpectedly achieves lower pass rates than restructured synthesis, particularly for the subtopic *recycling system failure*. This suggests that strict synthesis operates reliably, but the distinction between faithful and restructured modes requires further investigation and potential fine-tuning.

	Strict	Faithful	Restructured
Overall (all 13 checks)	**93.8** [93.3, 94.4]	**81.4** [80.6, 82.2]	**86.2** [85.6, 86.9]
Step structure avg	99.7 [99.5, 99.9]	99.9 [99.7, 100.0]	99.9 [99.8, 100.0]
Step subtopic avg	87.8 [86.6, 89.0]	74.4 [72.9, 75.8]	86.4 [85.0, 87.7]
Final structure avg	96.2 [95.5, 96.9]	67.1 [65.1, 69.1]	63.4 [61.4, 65.3]
Final subtopic avg	93.5 [92.6, 94.4]	89.7 [88.6, 90.8]	97.3 [96.8, 97.9]
Final preserves order	87.9 [85.9, 89.9]	64.7 [61.7, 67.6]	80.0 [77.5, 82.5]

Table 9: Controllability evaluation results across synthesis modes. An LLM judge (GPT-5-mini) performs 13 boolean checks per argument (6 step-level, 7 final-level) on 1,000 arguments per synthesis type. Values are pass rates (%) with 95% bootstrap CIs (resampling arguments, $B = 10{,}000$).

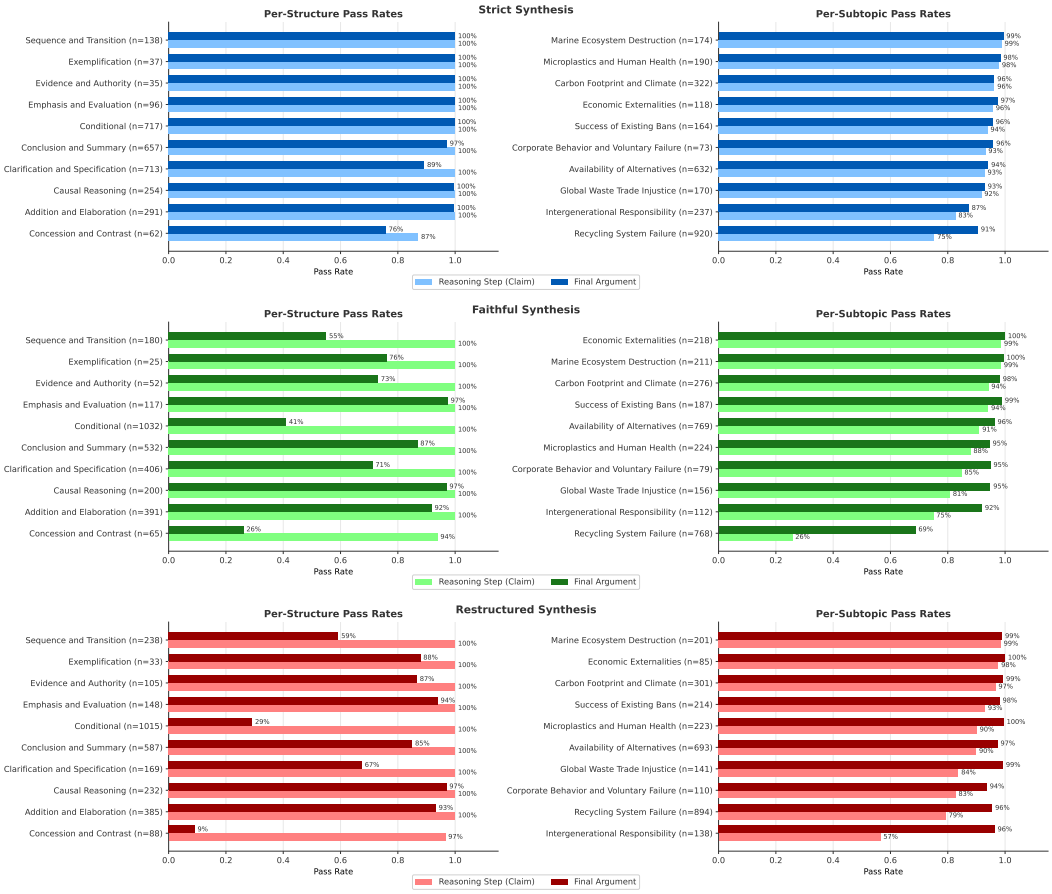

Figure 12: Controllability pass rates across synthesis modes (plastic pollution topic). Each group of bars shows a different evaluation check. Strict synthesis achieves the highest overall pass rate, particularly for final-level structure and order preservation. Restructured synthesis maintains high subtopic adherence but loses structural fidelity at the final-argument level.

E.2.4 Effect of Subtopic Specificity on Controllability

The controllability results above use *topic-specific* subtopics designed for the plastic pollution domain (e.g., *marine ecosystem destruction*, *recycling system failure*; Table 17). To test whether action-space specificity itself affects controllability, we repeat the strict-synthesis evaluation with *generic* subtopics, consisting of ten topic-independent argumentative lenses such as *justice and fairness* or *ethical principles* (Table 22), applied to the same plastic pollution topic. The generic evaluation covers 500 arguments (6,500 total checks), compared with 1,000 arguments (13,000 checks) for the topic-specific evaluation.

Subtopic adherence drops substantially with generic subtopics: from 87.8% to 49.8% at the step level and from 93.5% to 52.4% at the response level (Figure 13). Structure adherence, by contrast, remains comparable (96.9% vs. 99.7% step-level; 92.8% vs. 96.2% response-level), confirming that the prefix mechanism is robust regardless of subtopic design. Per-subtopic variance is also markedly higher with generic categories. *Justice and fairness* achieves only 2.7% step-level adherence, while the lowest-performing specific subtopic (*recycling system failure*) still reaches 75.3%.

We conclude that topic-specific subtopics yield higher controllability because concrete, domain-grounded categories are semantically distinct and easier for the generator to follow and for the judge to detect. Vague categories such as *justice and fairness* overlap semantically with many arguments about plastic pollution, making adherence harder to enforce and verify. This finding reinforces the practitioner guidance in Appendix H.3. When designing content dimensions, practitioners should prefer specific, potentially topic-grounded categories over abstract, generic lenses.

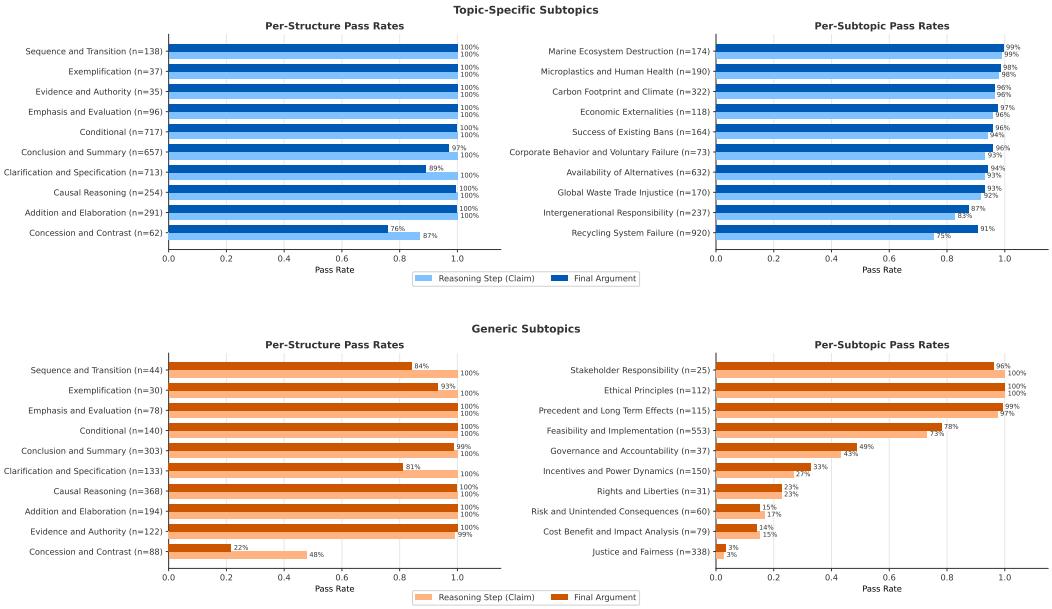

Figure 13: Controllability comparison between topic-specific subtopics (top, blue; n=1,000 arguments) and generic subtopics (bottom, orange; n=500 arguments), both under strict synthesis on the plastic pollution topic. Structure adherence is comparable across both designs, but subtopic adherence drops substantially with generic categories.

E.3 Predictability of Argument Quality from Actions

E.3.1 Argument Generation

The argument generation task is defined by a proposition (topic statement) and stance (pro vs. anti). We evaluate argument quality across five debate topics, each assigned to one of three LLM judges for pairwise comparisons (Table 10).

The main results in Section 4.2 use strict synthesis. Here, we additionally report ablation results for the faithful and restructured synthesis modes introduced in Appendix C.2.1, to assess the robustness of our attribution findings. For each mode, we generate 5,000 arguments using 20 random-seed trees (depth 3, beam width 250), then sample 50,000 random argument pairs for judge comparisons and fit a Bradley–Terry model to obtain quality rankings. We standardize these rankings and use them as our response variable. Further, we randomly split each dataset into training (60%) and test (40%) partitions at the level of individual arguments, so each argument and its trajectory appear in exactly one partition. The split is not grouped at the tree level. As each dataset comprises 20 tree-search runs of 250 arguments each, and sibling arguments from the same tree share partial reasoning trajectories, overlapping trajectories can appear on both sides of the split. Test performance should therefore be read as predictive accuracy for new arguments from the same generation process, rather than generalization to independently grown trees.

Proposition (stance: PRO)	Topic	LLM Judge
The government should enforce a total ban on single-use plastics.	Plastic Pollution	GPT-5-mini
Social media platforms should enforce a minimum age restriction of 16.	Social Media Restriction	GPT-5-mini
The government should implement a universal basic income program.	Universal Basic Income	GPT-5-mini
Standardized testing should be abolished as a primary measure of student performance.	Standardized Testing	Gemini-3.1-Flash-Lite
A special tax should be imposed on meat products to reduce consumption.	Meat Tax	Claude-Haiku-4.5

Table 10: Five debate topics with their assigned LLM judges for pairwise quality evaluation. The controllability evaluation (Section 4.2.1) and targeted trajectory exploration (Section 4.2.4) are conducted on the plastic pollution topic as a detailed case study.

The sequential model (M2) uses two action dimensions, content and structure, with features that encode (i) action identities by position, (ii) within-step content-structure interactions, and (iii) within-dimension transitions of length two across steps. These are compared against a length-only baseline (M0) and presence-based models (M1a–c). All models include argument length in characters as a feature. In this multi-dimensional setting, the feature space is high-dimensional and sparse, which increases overfitting risk and numerical instability. To address this, we use LASSO regression for the M2 models (Tibshirani, 1996), with α selected by grid search and 10-fold cross-validation, so irrelevant coefficients are shrunk toward zero:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \mathbf{x}_i^{\text{sequential}} \boldsymbol{\beta})^2 + \alpha \|\boldsymbol{\beta}\|_1 \right\} \tag{12}$$

where $\mathbf{x}_i^{\text{sequential}}$ is the sequential feature encodings for the M2 models. For M0 and M1a–c, we use ordinary least squares. We report bootstrap 95% confidence intervals (1,000 resamples of individual test arguments) for the test-set R^2. M2's feature space strictly refines the presence-only features, which enables a direct comparison of whether sequential structure improves predictability.

Topic	Synthesis	M0 R^2	M1a R^2	M1b R^2	M1c R^2	M2 R^2
Plastic Pollution	Strict	0.478 ± 0.027	0.539 ± 0.028	0.627 ± 0.023	0.697 ± 0.022	$\mathbf{0.760 \pm 0.018}$
	Faithful	0.345 ± 0.034	0.363 ± 0.035	0.610 ± 0.025	0.615 ± 0.025	$\mathbf{0.626 \pm 0.024}$
	Restructured	0.200 ± 0.032	0.220 ± 0.032	0.611 ± 0.025	0.613 ± 0.024	$\mathbf{0.614 \pm 0.024}$
Social Media Restriction	Strict	0.613 ± 0.025	0.669 ± 0.024	0.695 ± 0.020	0.748 ± 0.018	$\mathbf{0.771 \pm 0.017}$
	Faithful	0.294 ± 0.033	0.304 ± 0.032	0.620 ± 0.025	0.621 ± 0.024	$\mathbf{0.634 \pm 0.024}$
	Restructured	0.321 ± 0.033	0.320 ± 0.034	0.608 ± 0.025	0.608 ± 0.025	$\mathbf{0.609 \pm 0.027}$
Universal Basic Income	Strict	0.536 ± 0.029	0.618 ± 0.027	0.644 ± 0.023	0.732 ± 0.020	$\mathbf{0.790 \pm 0.017}$
	Faithful	0.311 ± 0.032	0.338 ± 0.033	0.704 ± 0.022	0.711 ± 0.021	$\mathbf{0.734 \pm 0.021}$
	Restructured	0.232 ± 0.028	0.276 ± 0.031	0.772 ± 0.016	0.773 ± 0.016	$\mathbf{0.799 \pm 0.016}$
Standardized Testing	Strict	0.536 ± 0.029	0.666 ± 0.022	0.569 ± 0.028	0.681 ± 0.022	$\mathbf{0.802 \pm 0.016}$
	Faithful	0.416 ± 0.027	0.482 ± 0.027	0.532 ± 0.026	0.584 ± 0.025	$\mathbf{0.634 \pm 0.023}$
	Restructured	0.410 ± 0.031	0.432 ± 0.030	0.609 ± 0.025	0.610 ± 0.025	$\mathbf{0.625 \pm 0.024}$
Meat Tax	Strict	0.632 ± 0.029	0.678 ± 0.028	0.641 ± 0.028	0.684 ± 0.028	$\mathbf{0.726 \pm 0.027}$
	Faithful	0.342 ± 0.032	0.351 ± 0.032	0.401 ± 0.031	0.398 ± 0.032	$\mathbf{0.408 \pm 0.035}$
	Restructured	0.406 ± 0.032	0.428 ± 0.032	0.460 ± 0.030	0.462 ± 0.031	$\mathbf{0.484 \pm 0.030}$

Table 11: Model comparison with argument length control across topics. R^2 values on held-out test set (40%) with 95% bootstrap CI ($\pm$ half-width). M0 uses only argument length (characters). All other models additionally include length.

E.3.2 Predictability Results Across Synthesis Modes

Figure 14 presents how well controller actions predict argument quality across all five debate topics, shown separately for each synthesis mode. The consistent pattern across topics is that the sequential model (M2) outperforms presence-based baselines (M1a–c) and the length-only baseline (M0), confirming that action ordering and transitions carry predictive information beyond simple topic

presence. Strict synthesis yields the highest predictability in most topics, while restructured synthesis generally shows the lowest, reflecting the control-fidelity gradient observed in the controllability evaluation. Notably, the relative gain of M2 over presence-based models is largest for strict synthesis, where action sequences are most faithfully preserved.

Cross-topic variation in absolute R^2 values is likely driven by differences in argument length distributions (see Figure 15), judge consistency, and adherence to intended synthesis mode behavior (see controllability results discussed in Appendix E.2.3). Some length distributions are bimodal, which appears to increase predictability across all models under consideration. Figures 16 and 17 show the selected α values and the composition of retained features for M2 across topics and synthesis modes, illustrating how the LASSO regularization selects different feature subsets depending on the setting.

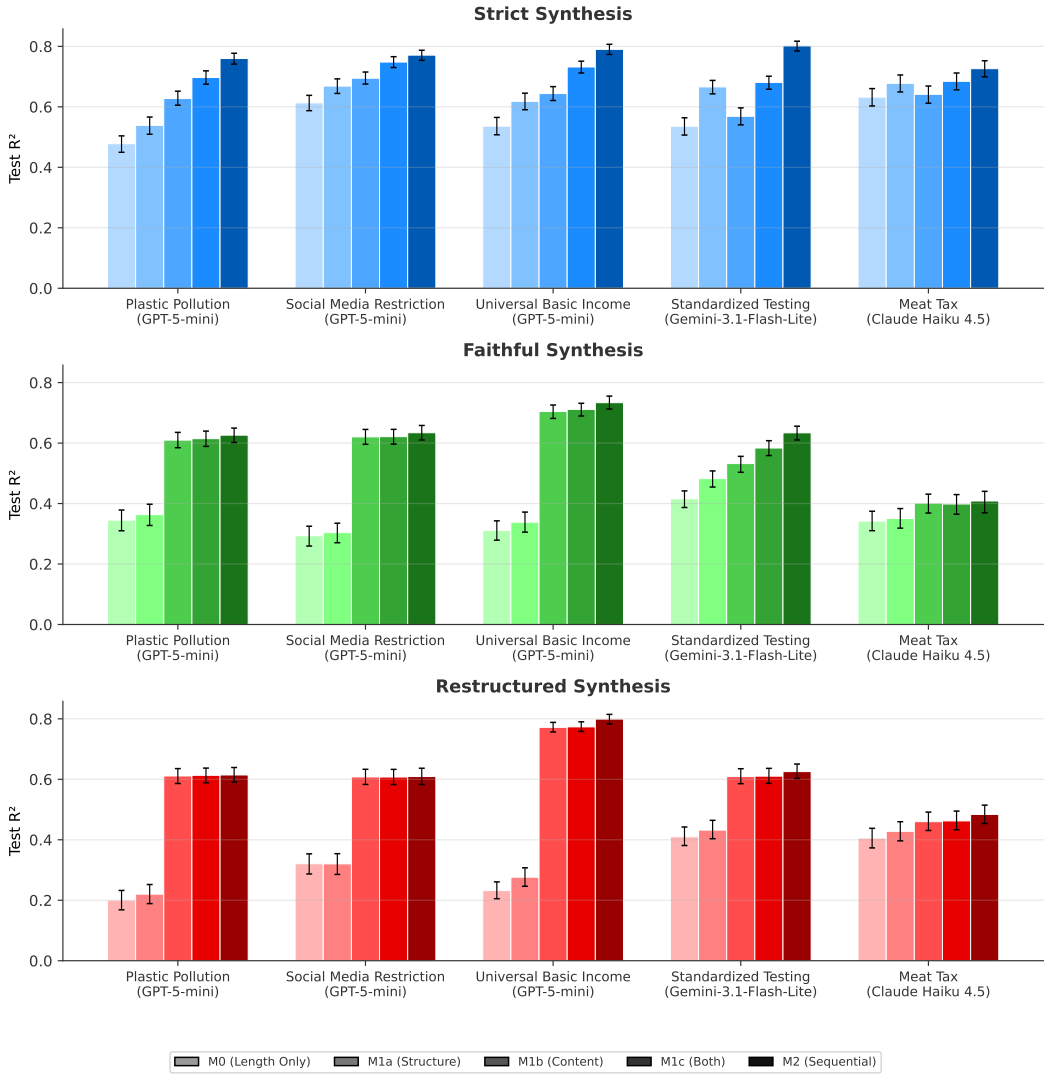

Figure 14: Predictability of argument quality from controller actions across all five debate topics, shown separately for each synthesis mode (with length control). Each row corresponds to one synthesis mode (Strict, Faithful, Restructured); within each row, bars show test-set R^2 (with 95% bootstrap CIs) for M0 (length only), M1a (structure presence), M1b (content presence), M1c (both), and M2 (full sequential model).

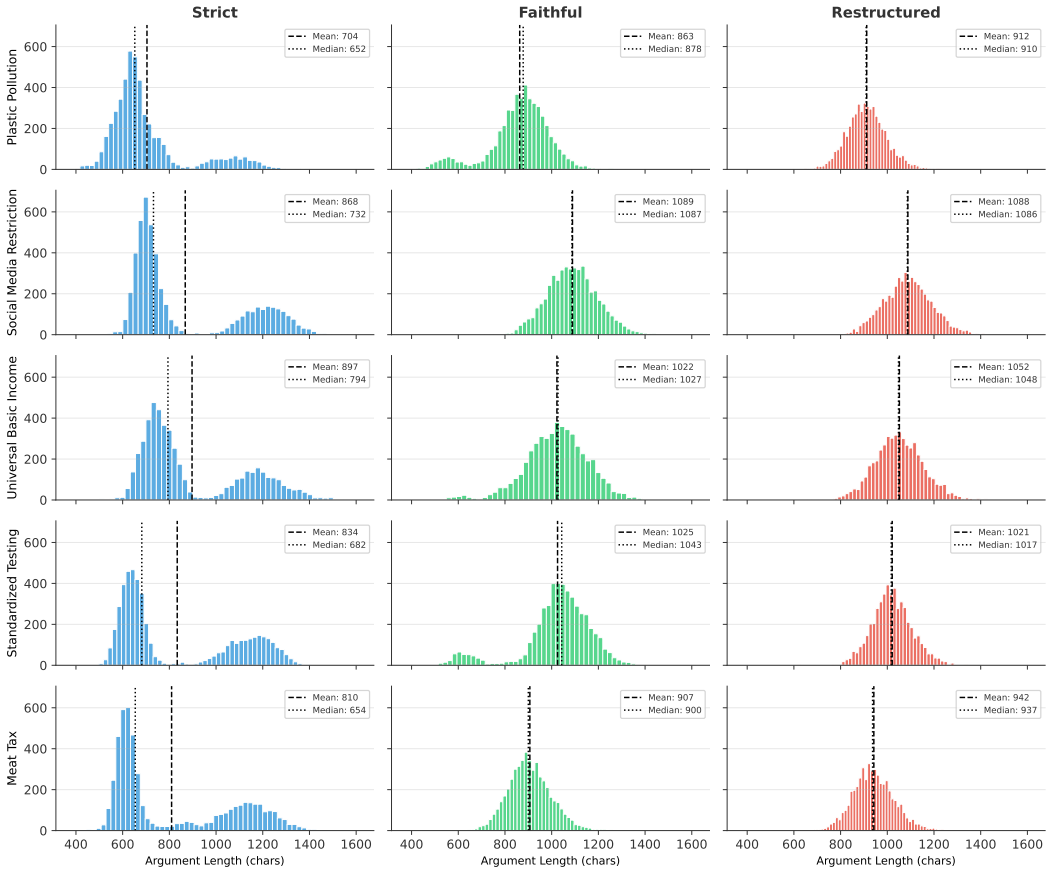

Figure 15: Distribution of argument lengths (in characters) across all five debate topics and three synthesis modes. Columns correspond to synthesis modes (strict, faithful, restructured); rows correspond to debate topics. Dashed and dotted vertical lines indicate the mean and median length, respectively.

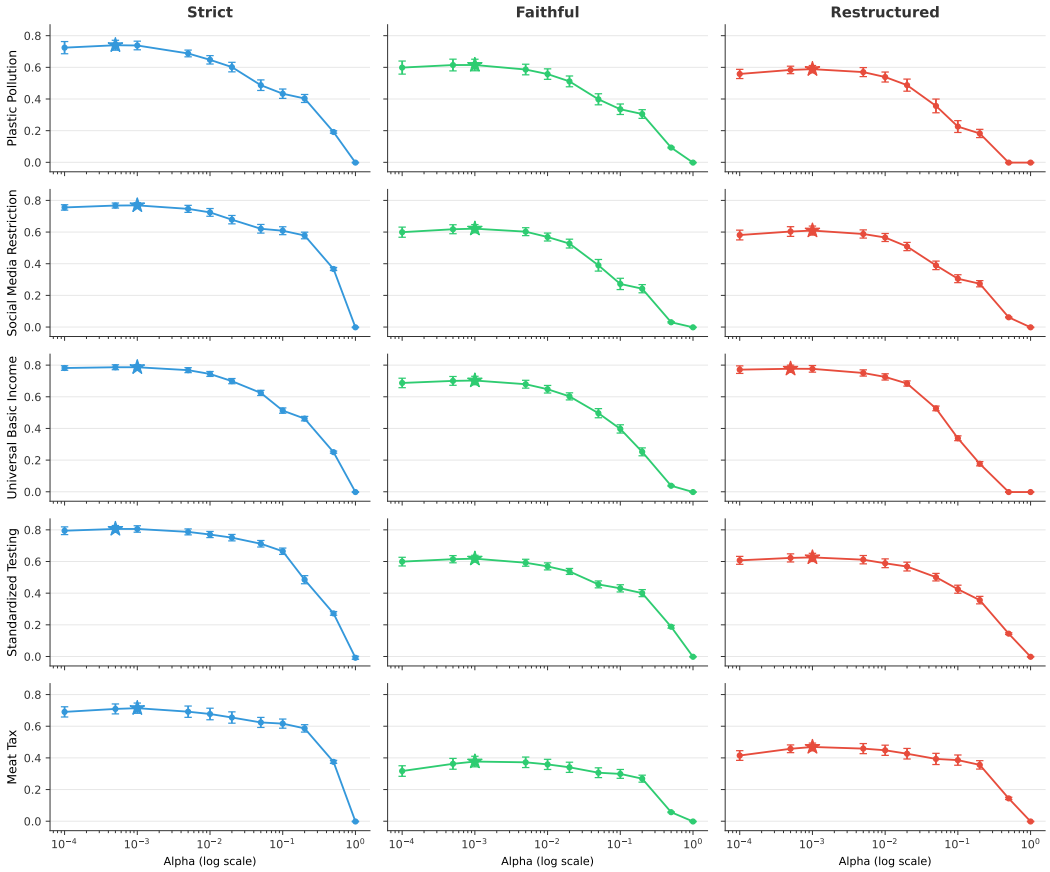

Figure 16: Cross-validation R^2 as a function of LASSO regularization parameter α across all five topics. Columns correspond to synthesis modes (strict, faithful, restructured); rows correspond to debate topics. Stars indicate the selected α for each configuration.

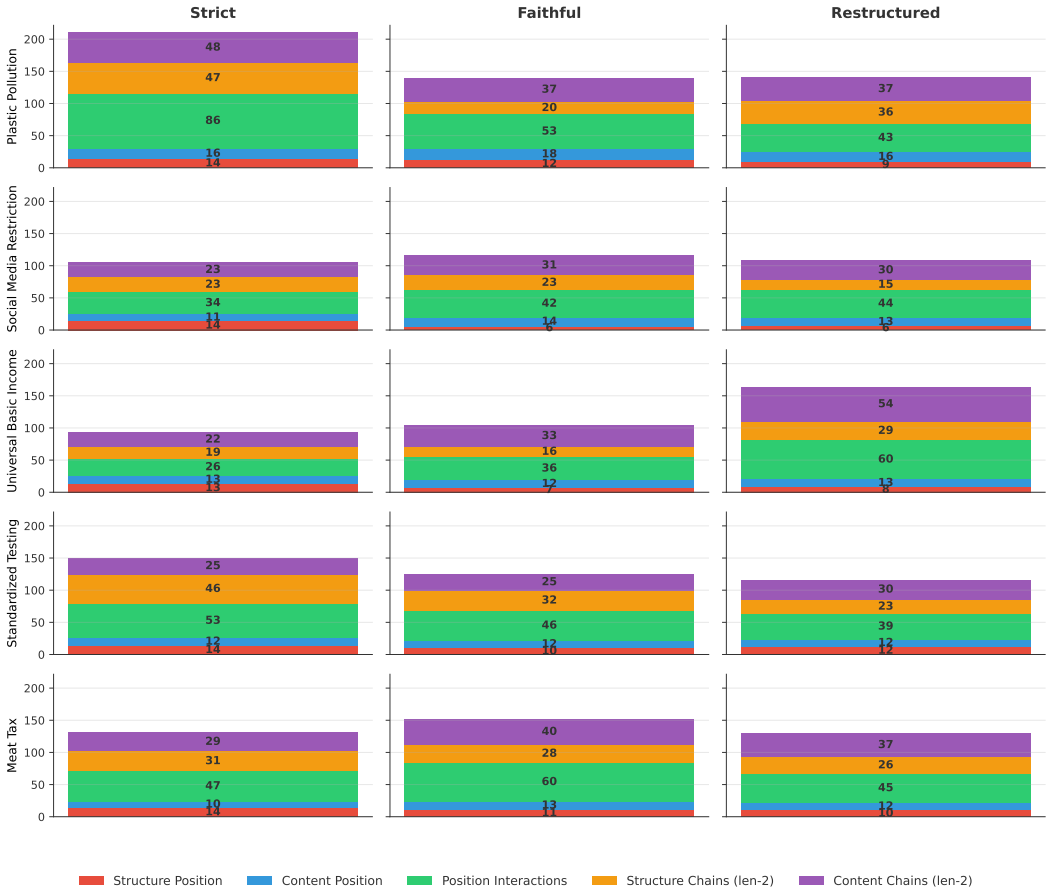

Figure 17: Distribution of LASSO-selected features by category across all five topics. Columns correspond to synthesis modes (strict, faithful, restructured); rows correspond to debate topics. Position effects and transitions within each dimension (structure, content) contribute to predictions across all configurations.

E.3.3 Targeted Trajectory Exploration

This subsection provides the full setup and results details for Section 4.2.4. Again, we expand the analysis with ablations covering the two additional synthesis modes (faithful and restructured) introduced in Appendix C.2.1. We focus only on the plastic pollution topic in this experiment. The attribution analysis establishes that M2's sequential features explain significant variance in argument quality. A natural extension is testing whether these estimates generalize beyond the observed feature combinations. M2 was trained on ∼5,000 arguments per synthesis type, but the full trajectory space contains $100^3 = 1{,}000{,}000$ possible 3-step sequences, the vast majority of which are unobserved. If M2's learned coefficients generalize, we can use them to identify promising unexplored regions of the trajectory space and generate targeted arguments with those features. This corresponds to the final step of our proposed workflow in Figure 1: using attribution estimates to guide targeted generation. Using M2's fitted coefficients[17], we score all possible trajectories and rank them by predicted quality. We select the top 50 trajectories per synthesis type, all of which were never observed in the training data, and use STATe's forced controller mechanism[18] to generate arguments following each trajectory exactly. For each trajectory, we generate 5 samples, yielding 250 targeted arguments per synthesis type.

We evaluate these targeted arguments against three baselines that test different aspects of M2's contribution. First, the **Random** baseline samples trajectories uniformly from the unobserved trajectory space; if M2's selection provides no value, targeted arguments should perform at chance (50%) against random exploration. Second, the **M1b (Topic Presence)** baseline tests whether simply knowing which content topics correlate with quality is sufficient. This emulates what a simpler topic modelling approach might discover: which topics matter, but without M2's sequential and structural information (Saenger et al., 2024; Fong & Grimmer, 2016). Specifically, we identify the top-3 topics based on the M1b model and filter for trajectories that contain only these topics, then sample randomly from this filtered set. Third, the **Original Top 5%** baseline compares targeted arguments against the best 5% of arguments from the original pairwise evaluation (by BT score), testing whether M2-guided generation can match or exceed the quality of the best observed arguments.

Argument length correlates strongly with performance in this setting, as shown in the predictability results of Appendix E.3.2. In particular, the original top 5% arguments tend to be disproportionately long, creating a confounder that would affect any direct comparison (Dubois et al., 2024). We construct length-matched evaluation sets using greedy pairing of arguments of similar length. [19] This yields balanced datasets ranging from 200 to 398 arguments, depending on synthesis type and baseline. We evaluate these datasets by running 5,000 random pairwise comparisons within each and calculating new Bradley-Terry scores. Win rates are computed over the cross-group comparisons (targeted vs. baseline). Because each argument appears in roughly 15–25 comparisons, comparisons sharing an argument are correlated, and treating them as independent would understate uncertainty. We therefore estimate 95% CIs with a two-way cluster bootstrap. In each of B=10,000 iterations, we resample the targeted and the baseline arguments with replacement and recompute the win rate, weighting each comparison by the product of its arguments' resample multiplicities. Clustering at the argument level reflects variability across generated arguments rather than across repeated judgments of the same fixed arguments.

Table 12 shows that targeted arguments substantially outperform both the random baseline (73–81% win rate) and the topic-presence baseline (61–67% win rate) across synthesis types. This confirms that M2's trajectory rankings identify genuinely promising regions of the action space, more so than a simpler topic-based approach might do. Against the original top 5%, targeted arguments remain competitive (31–68% win rate), substantially exceeding the win rate of less than 5% that we would expect if M2's rankings failed to generalize beyond the observed samples. Especially in comparisons against the original top 5%, the familiar predictability gradient emerges: strict synthesis shows the strongest performance, while restructured synthesis exhibits greater variability. We also report the share of targeted arguments among the top-10 and top-100 of the performance-ranked, length-matched datasets. This highlights that when the goal is to find the very best arguments, M2-guided trajectory selection offers a promising approach.

[17] Figure 18 displays the top 20 features by coefficient magnitude for the M2 models across synthesis modes.

[18] Our implementation allows for forcing controller choices rather than using the controller module to choose the next action. For details, please refer to the repository.

[19] For each targeted argument, we find the closest-length baseline argument within ±5 characters, using each baseline argument at most once. This effectively leaves us with the intersection of the length histograms shown in Figures 19, 20, and 21.

Baseline	Type	N	Win (T) (%)	95% CI	Top-10	Top-100
Random	Strict	354	78.7	[73.9, 83.2]	8/10	78/100
M1b (Topic Presence)	Strict	340	63.3	[57.3, 69.2]	6/10	57/100
Original Top 5%	Strict	204	68.0	[60.9, 74.8]	9/10	68/100
Random	Faithful	288	73.2	[67.8, 78.6]	7/10	75/100
M1b (Topic Presence)	Faithful	374	61.4	[56.0, 66.8]	5/10	61/100
Original Top 5%	Faithful	280	35.7	[30.2, 41.1]	0/10	38/100
Random	Restructured	344	81.0	[76.6, 84.8]	10/10	88/100
M1b (Topic Presence)	Restructured	398	67.2	[62.2, 72.0]	8/10	69/100
Original Top 5%	Restructured	200	30.7	[25.1, 36.6]	1/10	34/100

Table 12: Targeted Trajectory Exploration: Evaluating new, targeted vs. new baseline explorations across synthesis modes. N is the total number of length-matched argument pairs (balanced: $N/2$ targeted, $N/2$ baseline). Comparisons count is the number of random pairwise comparisons evaluated. Win Rate is the share of comparisons between targeted and baseline arguments that the targeted argument won with 95% bootstrap CIs ($B = 10{,}000$). Top-10 and Top-100 counts show the number of targeted arguments in the top-n arguments of the length-matched dataset when sorting by Bradley-Terry score based on the pairwise comparisons.

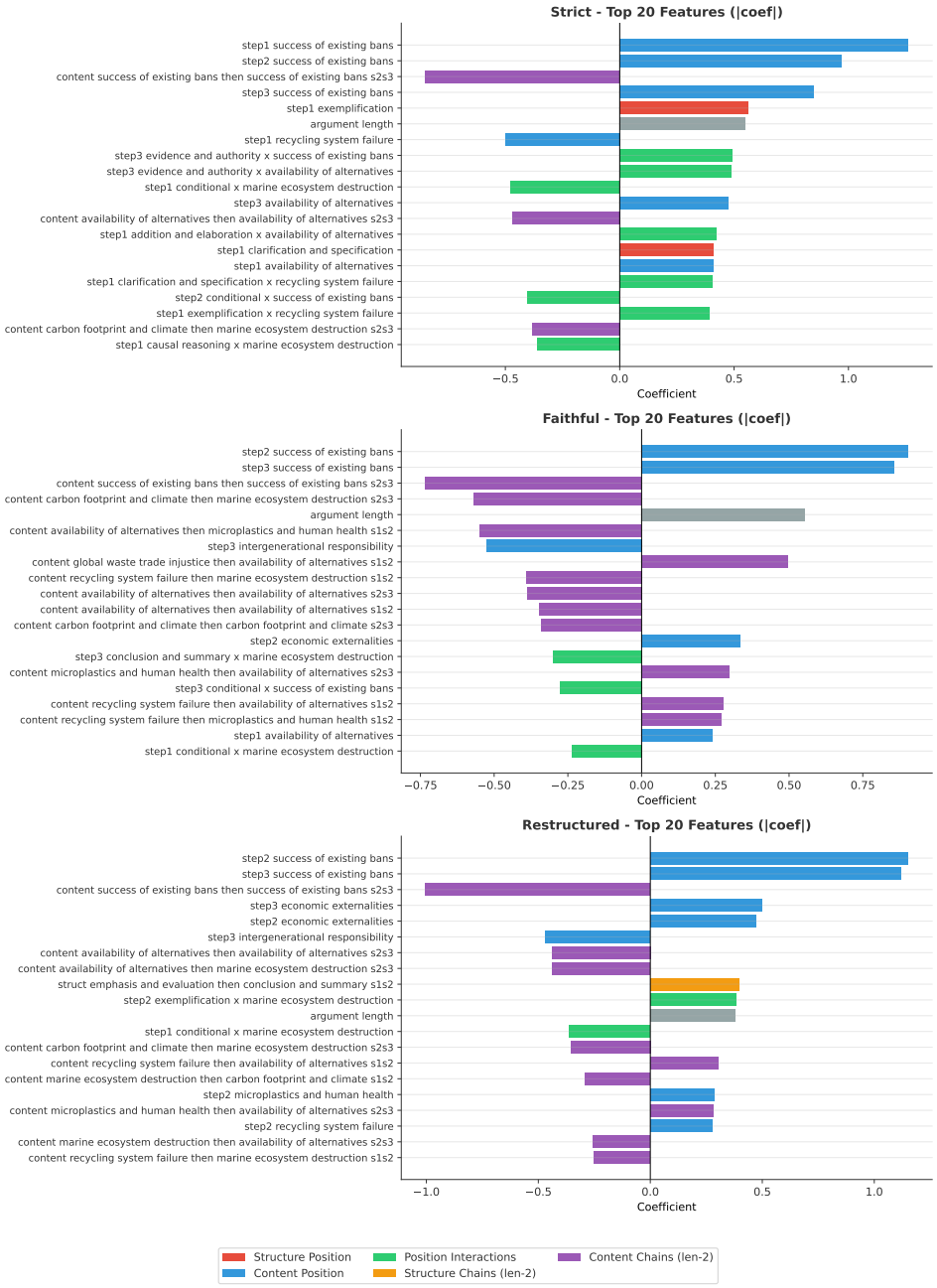

Figure 18: Top 20 features by absolute LASSO coefficient for each synthesis type for M2 models for arguments on plastic pollution. Positive coefficients indicate patterns associated with higher persuasiveness scores; negative coefficients indicate patterns associated with lower persuasiveness scores.

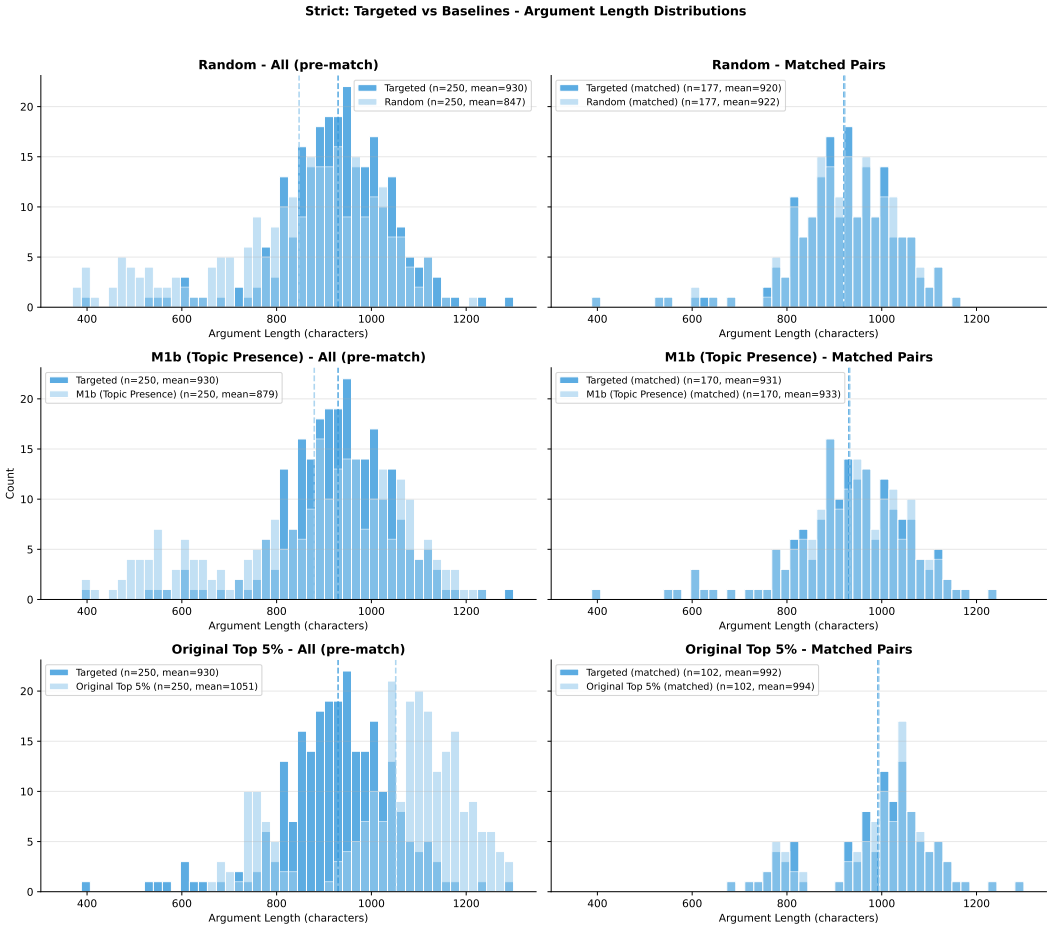

Figure 19: Length distributions for targeted trajectory evaluation for `strict` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments skew longer, reflecting the correlation between length and judged quality. **Right:** Length-matched subset used for evaluation. Greedy pairing within ± 5 characters produces groups with comparable length distributions, enabling fair comparison. Dashed lines indicate group means.

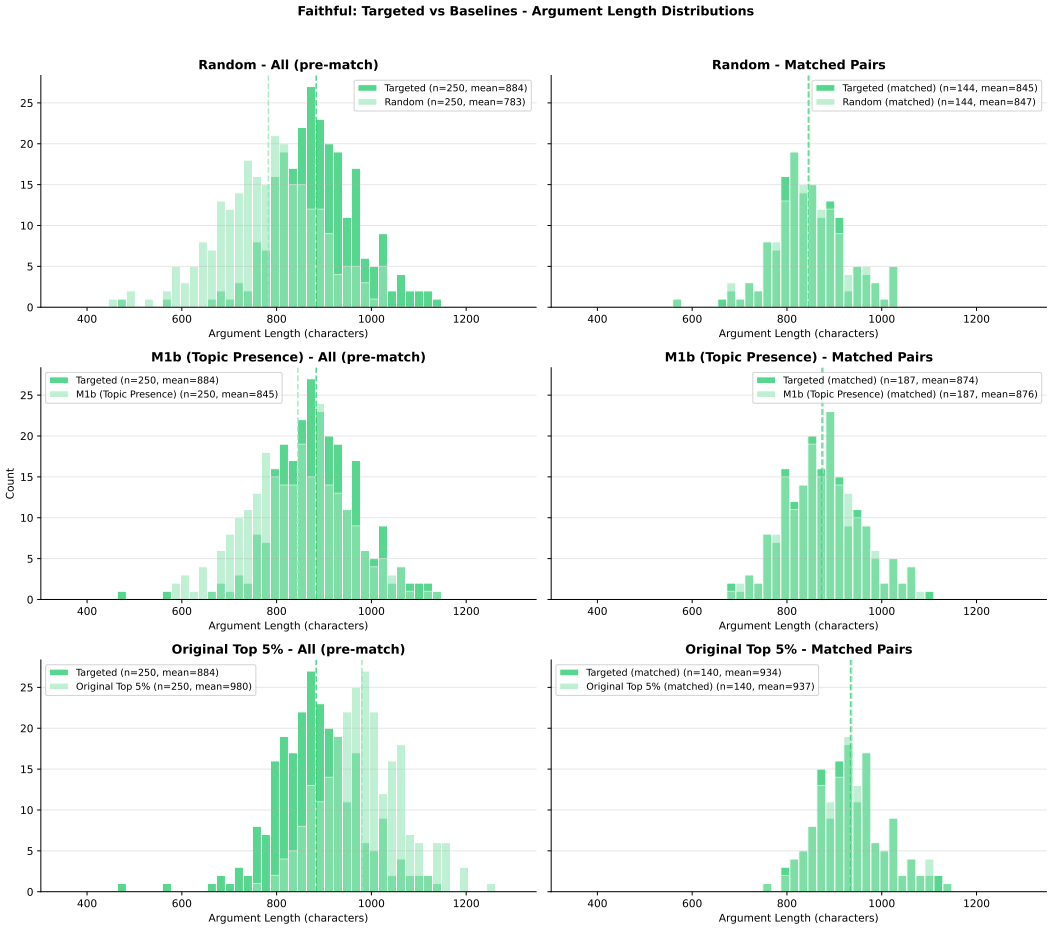

Figure 20: Length distributions for targeted trajectory evaluation for `faithful` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments skew longer, reflecting the correlation between length and judged quality. **Right:** Length-matched subset used for evaluation. Greedy pairing within ± 5 characters produces groups with comparable length distributions, enabling fair comparison. Dashed lines indicate group means.

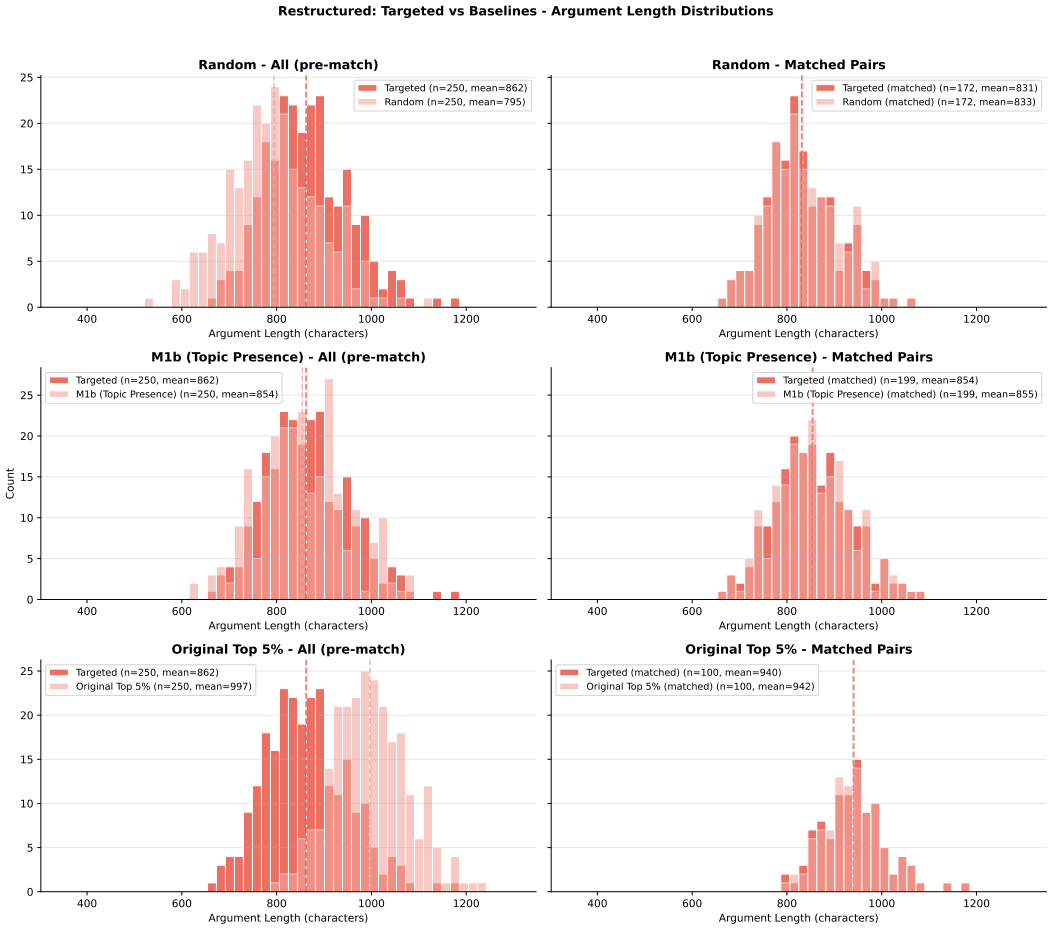

Figure 21: Length distributions for targeted trajectory evaluation for `restructured` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments skew longer, reflecting the correlation between length and judged quality. **Right:** Length-matched subset used for evaluation. Greedy pairing within ± 5 characters produces groups with comparable length distributions, enabling fair comparison. Dashed lines indicate group means.

E.4 Human Validation of the LLM Judge

We conducted a human study to evaluate whether the LLM judge used for pairwise argument evaluation (Section 4.2) is a valid proxy for human preferences. Our study is IRB-approved (approval number 2026-055) and participants in our Qualtrics survey were compensated with $4 for completing our 12-question survey. We used Prolific to distribute our survey to US residents who speak English fluently and are over 18 years old. We only accepted participants with an approval rating of at least 99% on Prolific. On average, participants took 13.5 minutes to complete the survey. We paid an additional $1 performance bonus if participants passed both attention-check questions.

E.4.1 Study Design

The study focuses on arguments generated for a single topic (total ban on single-use plastics) and a single LLM judge (GPT-5-mini; Singh et al., 2025). Participants read two arguments supporting the motion and chose the more persuasive one in a forced binary choice (no ties). We selected 50 argument pairs from the pool of 5,000 arguments previously scored via Bradley–Terry analysis of 50,000 LLM-judged pairwise comparisons, stratified by BT score gap: small (< 1.0; 17 pairs), medium (1.0–2.0; 17 pairs), and large (≥ 2.0; 16 pairs). The LLM judge evaluated the same pairs via pairwise comparison with effectively the same instruction format used in the human study.

We recruited 310 Prolific workers (US-based, English fluency, high approval rating), each evaluating 10 pairs. Participants were assigned to one of 10 templates, each containing a fixed set of 10 pairs. Every pair appeared in exactly 2 templates (order of the arguments switched). Six participants submitted duplicate responses (first submission kept). Two attention-check items were embedded in each survey, pairing a top-ranked argument against a deliberately degraded counterpart for which the stronger argument is unambiguous; 94.2% of participants (292/310) passed both checks. After filtering to unique participants who passed both attention checks, the final sample contained n=288 participants.

E.4.2 The ALT Test

We used the ALT test (Calderon et al., 2025) to determine whether the LLM judge is interchangeable with a human annotator. The test uses a leave-one-out procedure where for each human annotator h, it excludes h from the pool, scores both h and the LLM against the remaining annotators on each instance, and checks whether the LLM matches or exceeds h's alignment with the group. Formally, for each instance i annotated by h, the alignment score is the fraction of remaining annotators who gave the same answer.

The ALT test's primary statistic, the *winning rate*, applies per-annotator hypothesis tests with FDR correction (Benjamini & Yekutieli, 2001). However, our study design (288 annotators, each rating only 10 pairs) leaves these individual tests severely underpowered, making the winning rate uninformative in our setting. We therefore focus on the *advantage probability*, a non-parametric summary that does not depend on per-annotator statistical power: across all annotators and all instances, how often does the LLM match or exceed the excluded human's alignment with the group?

E.4.3 Results

Table 13 reports agreement between human annotators and between humans and the LLM judge. Human pairwise agreement of 56.0% reflects the inherent subjectivity of persuasiveness judgments; even among humans, annotators disagree nearly half the time. Krippendorff's $\alpha = 0.118$ confirms low absolute agreement. Despite this low inter-annotator reliability, the LLM agrees with the human majority vote 72.0% of the time, substantially above the human pairwise baseline.

Metric	Overall	Small	Medium	Large
LLM vs. human majority	0.720	0.706	0.706	0.750
Human pairwise agreement	0.560	0.584	0.515	0.583
Human majority proportion	0.652	0.672	0.593	0.693
Krippendorff's α	0.118	0.164	0.031	0.160

Table 13: Agreement statistics for the human validation study ($n = 288$, passed 2/2 attention checks).

The advantage probability of the ALT test reaches 0.841. Concretely, when a single human is removed from the pool and compared against the LLM on the same instances, the LLM agrees with the

remaining annotators at least as often as the excluded human does in 84.1% of comparisons. Table 14 breaks this metric down by the Bradley-Terry score gap between the two arguments in each pair, as determined by the LLM judge's own prior pairwise evaluations (Section E.4.1). The advantage probability remains high across all three strata (0.82–0.86), suggesting that the LLM is equally reliable on pairs it previously rated as close calls and on pairs it previously rated as clearly different. The slight decrease from small-gap to large-gap pairs does not indicate weaker LLM performance on easier pairs; rather, humans are more internally consistent on large-gap pairs (fewer outliers), which raises the baseline the LLM is compared against.

Stratum	Advantage Prob.	Annotators Tested
Overall	0.841	288
Small	0.860	229
Medium	0.834	200
Large	0.823	202

Table 14: Advantage probability by BT score gap stratum ($n = 288$).

The LLM's choices are also significantly correlated with human preference rates (Pearson $r = 0.493$, $p < .001$); when more humans prefer argument A, the LLM is more likely to also choose A. This correlation strengthens for large-gap pairs ($r = 0.657$, $p = .006$), where the quality difference is clearest. The LLM is also more accurate on pairs where humans agree more strongly ($r = 0.300$, $p = .035$), though this effect is driven primarily by large-gap pairs ($r = 0.609$, $p = .012$).

F Limitations

STATe relies on *prefilling* text prefixes to implement intervention-based branching. Modern closed-source APIs (e.g., GPT, Claude, Gemini) generally do not provide robust support for the kind of assistant-prefill control required by this workflow, so the method is currently most straightforward to deploy with open-source/self-hosted models. In addition, our analysis of action–outcome relationships is associative: while we find that action sequences are predictive of downstream judgments and that targeted, previously unseen trajectories can perform well, we do not make causal claims about how any particular action choice affects the final text or the downstream outcome. Our current setup violates sequential ignorability, as we choose actions conditional on existing reasoning.

A second limitation is rigidity in the action and prefix design. In our implementation, each action is realized by a fixed textual prefix, but many interventions admit multiple natural surface forms (e.g., synonymous discourse markers for "causal reasoning" include "Because", "Therefore", and "As a result"). Representing these variants naively would require expanding the action space substantially, while many actions would share identical definitions. Moreover, some prefixes are well-formed as mid-document transitions but can sound unnatural as the first step of an answer (e.g., "Therefore" or "However"), which can create stylistic artifacts unless one conditions the action space on position or introduces context-aware prefix variants. These issues point to a broader limitation: action spaces require careful, task-specific engineering, and the best granularity of actions (coarse vs. fine) may vary across domains.

Relatedly, the synthesis step that converts reasoning traces into final outputs introduces a trade-off between control and quality. Strict synthesis preserves a tight coupling between action sequences and output text, enabling high predictability in the argument quality experiment, but potentially producing stilted prose that mechanically concatenates reasoning steps. More flexible synthesis modes allow the model to smooth transitions and improve eloquence, but this freedom attenuates the mapping from actions to output properties. This trade-off has practical implications: when the goal is to study how specific rhetorical choices affect perceived quality, stricter synthesis provides cleaner attribution, whereas producing high-quality arguments for deployment may favor more flexible synthesis despite reduced interpretability. In practice, our synthesis modes do not always behave as intended. Under strict synthesis, outputs are typically simple concatenations of reasoning steps, but occasionally the synthesis step adds concluding sentences, producing unexpectedly longer arguments. Results in Appendix E.2.3 and E.3.2 show that the expected gradient in control and predictability across the three evaluated synthesis modes did not materialize, warranting further investigation into synthesis prompt design and more explicit length control.

Additionally, our current approach to generating multiple realizations of the same trajectory relies on random seeds and achieves only limited diversity. Future work could introduce variation at the

tree level, such as including personas (Park et al., 2023; 2024) in the input or using different language models, to enable better estimation of trajectory-level effects.

Finally, our framework makes several scope assumptions. STATe currently focuses on single-turn, multi-step generation and does not explicitly model multi-turn conversational dynamics. [20] We also treat actions as textual interventions and do not support general tool calling beyond what is encoded in the action templates; integrating retrieval (for additional diversity) and tool-based verification (e.g., for numerical or algorithmic claims) could improve both generation and evaluation.

G Extended Future Work

G.1 Causal inference for sequential action choices

Our current attribution analysis is focused on associations and predictability rather than causal claims. A natural extension is to formalize action sequences as sequential treatments, where, at each depth $i \in \{1, \ldots, d\}$, the controller selects a_i conditional on previous actions $(a_1, \ldots, a_{i-1})$ and the current parent state s_{i-1}. This framework is directly related to marginal structural models and sequential ignorability (Hernán et al., 2000; Robins & Hernan, 2024). In this framework, g-computation or inverse probability weighting could estimate the per-step causal effects of a_i on the quality of the final output (Robins & Hernan, 2024). Mediation analysis (VanderWeele, 2015) could further distinguish whether an action affects the outcome directly or through its influence on subsequent action choices. Crucially, because STATe's controller can randomize action selections at each step, it enables experimental designs that eliminate the need for sequential ignorability assumptions.

G.2 Human evaluation and behavioral outcomes

Our argument evaluation currently relies on LLM judges (Section 4.2). Although this offers repeated access to stable preferences and is broadly correlated with human judgments, it does not substitute for rigorous human experimentation. The preferences of groups and individuals are complex, context-dependent, and shaped by heterogeneous prior beliefs that are difficult to simulate with language models. Future work should therefore conduct controlled human subject experiments with pre- and post-intervention measurements of beliefs or behaviors. STATe's sequential and multi-dimensional action traces provide a uniquely informative design space for such studies, enabling systematic manipulation of rhetorical structure, topical framing, and ordering effects that would be difficult to isolate using other methods. Furthermore, such experiments would allow us to report notions of persuasiveness (the effect that an argument has on a reader) rather than argument quality (i.e., human preference towards argumentative text).

G.3 Search and optimization over action spaces

We currently explore action spaces using fixed beam search and show that regression-based estimates can identify promising, previously unobserved trajectories (Section 4.2.4). A natural extension is to employ principled tree search algorithms such as Monte Carlo Tree Search (MCTS) (Kocsis & Szepesvári, 2006; Coulom, 2006; Browne et al., 2012; Silver et al., 2016; 2018; Hao et al., 2023). Such approaches could iteratively generate arguments, update effect estimates, and adapt exploration toward high-performing regions of the action space under constrained evaluation budgets. Importantly, tree search methods also extend naturally to multi-turn and adversarial settings. This opens the possibility of integrating STATe with Multi-Agent Debate (MAD) frameworks to identify optimal multi-turn conversational or adversarial strategies, rather than optimizing single-turn outputs alone.

G.4 Weight-based optimization with Reinforcement Learning

Group-wise policy optimization methods for LLMs (Shao et al., 2024; Liu et al., 2025b) often suffer from mode collapse, where multiple sampled trajectories converge to near-identical completions. Prior work highlights this as a central limitation of RLHF-style training regimes (Casper et al., 2023; Gai et al., 2025). By sampling across discrete and interpretable action sequences rather than relying solely on token-level stochasticity, STATe increases semantic diversity while preserving quality (Section 4.1), potentially mitigating collapse in group-based rollout sets. Weight-based optimization via PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), or related methods could train the controller, generator, or evaluator to improve downstream performance.

[20] We do not support tool calls and tool call outputs as conversation turns, instead opting to directly include them as part of the thinking process of assistant messages.

G.5 Prompt optimization

Given a reliable ORM signal from a real downstream task (e.g., human preferences, click-through rates, or task success), STATe's components can be optimized in a cascading fashion. The PRM can be calibrated to assign high scores to intermediate states that yielded strong final outputs and low scores to states that yielded weak outputs. The controller can be optimized to select actions that maximize the expected ORM score, and the generator can be optimized to maximize PRM or ORM scores subject to the chosen action. Prompt optimization methods (Khattab et al., 2024; Opsahl-Ong et al., 2024; Agrawal et al., 2026; Yuksekgonul et al., 2025) offer one path: they can yield better or more precise task instructions, in-context examples that exemplify desired behaviors, and broadly improve the reliability of each component.

Separately, STATe's structured action spaces also offer a mechanism to diversify prompt-search strategies within reflective prompt evolution frameworks (Agrawal et al., 2026). That is, with an action space that reflects sequential prompt edits (e.g., including a new in-context example, or adding an edge case to the instructions), STATe can be used to search for the best prompt configuration to maximize a provided metric.

H Action Spaces

This appendix lists the structured action templates used by STATe in each experiment suite.

H.1 NoveltyBench action spaces

Name	Definition	Prefix	Internal reasoning
`openness`	Emphasizes creativity, intellectual curiosity, and preference for novelty over tradition.	*(none)*	Approach with curiosity and creativity; seek novel ideas; think abstractly and imaginatively.
`conscientiousness`	Emphasizes organization, self-discipline, reliability, and goal-oriented achievement.	*(none)*	Be methodical and detail-oriented; work systematically toward clear goals; be thorough.
`extraversion`	Emphasizes enthusiasm, assertiveness, sociability, and high energy.	*(none)*	Be energetic and confident; engage boldly; express thoughts with passion and optimism.
`agreeableness`	Emphasizes empathy, cooperation, warmth, and concern for others.	*(none)*	Be empathetic and collaborative; consider stakeholders; seek harmony; show sincere concern.
`neuroticism`	Emphasizes caution, risk awareness, and sensitivity to potential problems.	*(none)*	Be cautious and attentive to risks; examine uncertainties; consider worst-case scenarios.

Table 15: NoveltyBench action space: Personality (Big-5 traits).

Name	Definition	Prefix	Internal reasoning
`children`	Writes for children ages 5–12 using simple language, examples, and enthusiasm.	*(none)*	Use very simple words and short sentences; cheerful tone; fun, concrete examples.

Name	Definition	Prefix	Internal reasoning
teenagers	Writes for teenagers ages 13–19 using relatable language, current trends, and engaging tone.	*(none)*	Use casual, relatable language; energetic tone; socially current examples.
young_adults	Writes for young adults ages 20–35 using modern, direct language with practical examples.	*(none)*	Use clear, modern language; practical examples; confident, approachable tone.
middle_aged	Writes for adults ages 36–55 using professional, balanced tone with real-world applications.	*(none)*	Use professional, balanced tone; grounded examples; pragmatic framing.
seniors	Writes for seniors (ages 56+) using clear, respectful, and warm language.	*(none)*	Use clear, respectful language; gentle pacing; thoughtfully explained examples.

Table 16: NoveltyBench action space: Target Audience (age demographics).

H.2 Argument generation action spaces

Name	Definition	Prefix	Internal reasoning
marine_ecosystem_destruction	Examines plastic pollution's impact on ocean life, food chains, and marine habitats.	*(none)*	I should consider how single-use plastics kill marine animals through ingestion and entanglement, degrade coral reefs, and disrupt ocean ecosystems at massive scale.
microplastics_and_human_health	Analyzes the infiltration of plastic particles into human bodies and potential health consequences.	*(none)*	I should examine emerging evidence that microplastics accumulate in human tissue, blood, and organs, posing uncertain but potentially serious health risks.
recycling_system_failure	Evaluates why recycling has proven inadequate as a solution to plastic waste.	*(none)*	I should consider that global plastic recycling rates remain below 10%, much plastic is downcycled or exported, and source reduction through bans is therefore necessary.
carbon_footprint_and_climate	Connects plastic production to fossil fuel extraction and greenhouse gas emissions.	*(none)*	I should analyze how single-use plastics are petrochemical products that contribute to emissions throughout their lifecycle, from extraction to incineration.
availability_of_alternatives	Assesses whether practical substitutes exist for common single-use plastic applications.	*(none)*	I should demonstrate that reusable, compostable, and biodegradable alternatives are available for bags, straws, containers, and packaging, making a ban practically feasible.
economic_externalities	Quantifies societal costs of plastic pollution not reflected in product prices.	*(none)*	I should calculate how cleanup costs, healthcare burdens, tourism losses, and ecosystem damage are borne by the public rather than producers, a market failure requiring correction.
success_of_existing_bans	Draws lessons from jurisdictions that have already implemented plastic restrictions.	*(none)*	I should examine case studies from Rwanda, the EU, Kenya, and various US states showing that bans are enforceable and produce measurable reductions in pollution.
global_waste_trade_injustice	Addresses how plastic waste burdens fall disproportionately on developing nations.	*(none)*	I should consider how wealthy countries export plastic waste to poorer nations, making this a global justice issue that requires upstream intervention at the production stage.
corporate_behavior_and_voluntary_failure	Examines why industry self-regulation has proven insufficient to reduce plastic use.	*(none)*	I should analyze how manufacturers default to cheap plastics despite pledges, and why only binding regulation, not voluntary commitments, can shift industry norms.
inter generational_responsibility	Considers obligations to future generations given plastic's persistence.	*(none)*	I should evaluate how plastics persist for centuries in the environment, meaning today's convenience imposes long-term burdens on future generations who had no say in their creation.

Table 17: Subtopics for **Plastic Pollution** (domain-specific topical lenses for the single-use plastic ban proposition).

Name	Definition	Prefix	Internal reasoning
adolescent_brain_development	Examines how prefrontal cortex immaturity makes under-16s vulnerable to addictive design and impulsive online behavior.	*(none)*	I should consider how the prefrontal cortex, responsible for impulse control and risk assessment, does not fully mature until the mid-20s, making adolescents particularly susceptible to dopamine-driven engagement loops and poor decision-making online.
mental_health_impact	Analyzes the relationship between social media use and anxiety, depression, and body image issues in young people.	*(none)*	I should examine how longitudinal studies link heavy social media use in under-16s to increased rates of anxiety, depression, loneliness, and body dissatisfaction driven by social comparison and curated self-presentation.
online_predation_and_safety	Evaluates the risks of grooming, exploitation, and data privacy violations targeting minors on social platforms.	*(none)*	I should analyze how social media platforms expose minors to predatory adults through direct messaging, how children's data is harvested for targeted advertising, and how age restrictions reduce the attack surface for exploitation.
cyberbullying_and_harassment	Examines the prevalence and psychological harm of online bullying among younger users and its spillover into schools.	*(none)*	I should consider how cyberbullying disproportionately affects younger adolescents, extends school-based conflicts into 24/7 online environments, and correlates with self-harm and suicidal ideation in vulnerable youth.
attention_and_academic_performance	Analyzes the effects of social media use on concentration, learning outcomes, and sleep quality in young people.	*(none)*	I should examine how constant notifications and context-switching fragment attention spans, how screen time before bed disrupts sleep architecture critical for learning, and how academic performance declines correlate with social media usage hours.
addictive_design_exploitation	Evaluates how platform features like infinite scroll, notifications, and variable rewards deliberately exploit developing minds.	*(none)*	I should analyze how social media platforms employ behavioral psychology techniques, including variable ratio reinforcement schedules and social validation feedback loops, that are specifically designed to maximize engagement and are disproportionately effective on developing brains.
corporate_accountability_failure	Examines platforms' consistent failure to self-regulate and their prioritization of profit over child safety.	*(none)*	I should consider how internal documents from major platforms reveal awareness of harm to young users alongside decisions to prioritize engagement metrics, demonstrating that voluntary self-regulation has consistently failed to protect children.
age_verification_feasibility	Evaluates the technical viability and proportionate privacy tradeoffs of implementing effective age restrictions.	*(none)*	I should examine how existing age verification technologies, including document verification and age estimation, can achieve high accuracy while maintaining proportionate privacy protections, as demonstrated by successful implementations in Australia and the EU.

Name	Definition	Prefix	Internal reasoning
social_development_and_identity	Analyzes how premature social media exposure distorts self-concept and undermines healthy offline socialization.	*(none)*	I should consider how early exposure to curated online personas, adult content, and performative social dynamics interferes with the natural development of identity, empathy, and face-to-face social skills during critical developmental periods.
international_regulatory_precedents	Draws lessons from countries that have implemented or proposed social media age restrictions and their outcomes.	*(none)*	I should analyze how Australia's social media ban for under-16s, the EU Digital Services Act's enhanced protections for minors, and China's youth usage time limits provide evidence on implementation feasibility, enforcement mechanisms, and positive outcomes.

Table 18: Subtopics for **Social Media Age Restriction** (domain-specific topical lenses for enforcing a minimum age restriction of 16 on social media platforms).

Name	Definition	Prefix	Internal reasoning
poverty_elimination_effectiveness	Examines direct cash transfers as the most efficient mechanism for poverty reduction across developed and developing nations.	*(none)*	I should consider how unconditional cash transfers have consistently outperformed in-kind assistance programs, with evidence from multiple countries showing that recipients invest in education, health, and productive assets rather than wasteful consumption.
bureaucratic_waste_elimination	Analyzes how replacing complex means-tested welfare with universal payments reduces administrative overhead and errors.	*(none)*	I should examine how current welfare systems spend significant portions of their budgets on eligibility determination, compliance monitoring, and fraud prevention, while means-testing creates poverty traps and excludes many eligible recipients through administrative burden.
fiscal_sustainability	Evaluates viable funding mechanisms including VAT, wealth taxes, carbon dividends, and sovereign wealth funds.	*(none)*	I should analyze how UBI can be funded through combinations of value-added taxes, progressive wealth taxes, carbon pricing revenue, and reallocation of existing welfare spending, while long-term savings from reduced poverty-related costs improve fiscal sustainability.
entrepreneurship_and_innovation	Examines how guaranteed income security enables risk-taking, startup creation, and creative pursuits.	*(none)*	I should consider how economic insecurity prevents talented individuals from pursuing entrepreneurship, and how a guaranteed income floor would enable more people to start businesses, invest in skills development, and engage in socially valuable creative work.
mental_health_and_wellbeing	Analyzes how guaranteed income reduces financial stress and eliminates the stigma of means-tested assistance.	*(none)*	I should examine how financial insecurity is a primary driver of chronic stress, anxiety, and depression, and how UBI's universality eliminates the shame and administrative burden associated with applying for targeted welfare programs.

Name	Definition	Prefix	Internal reasoning
automation_and_technological_unemployment	Evaluates UBI as a necessary safety net for workforce displacement driven by AI and robotics.	*(none)*	I should analyze how accelerating automation in manufacturing, transportation, retail, and professional services threatens to displace millions of workers, making a universal income floor essential for maintaining social stability during the economic transition.
labor_market_empowerment	Examines how UBI strengthens workers' bargaining power and ability to refuse exploitative employment.	*(none)*	I should consider how a guaranteed income floor gives workers genuine freedom to negotiate better conditions, leave abusive employers, pursue education or retraining, and refuse work that is unsafe, underpaid, or degrading.
gender_equity_effects	Analyzes how UBI values unpaid care work and reduces gendered economic dependency.	*(none)*	I should examine how women disproportionately perform unpaid caregiving and domestic labor that GDP fails to capture, and how UBI provides economic recognition and independence for caregivers while reducing the gendered poverty gap.
pilot_program_evidence	Draws on positive results from UBI experiments in Finland, Kenya, Stockton, and Alaska to demonstrate real-world effectiveness.	*(none)*	I should analyze how Finland's basic income experiment improved wellbeing and employment confidence, GiveDirectly's Kenya program boosted economic activity, Stockton's SEED program increased full-time employment, and Alaska's Permanent Fund has distributed dividends for decades without adverse effects.
economic_stimulus_effects	Evaluates how universal cash transfers boost consumer spending, local economies, and small business growth.	*(none)*	I should consider how direct cash transfers to all citizens increase aggregate demand, particularly among lower-income households with high marginal propensity to consume, creating local economic multiplier effects that benefit small businesses and community economies.

Table 19: Subtopics for **Universal Basic Income** (domain-specific topical lenses for implementing a government UBI program).

Name	Definition	Prefix	Internal reasoning
measurement_validity	Examines whether standardized tests actually measure the knowledge and skills they claim to assess.	*(none)*	I should consider how standardized tests often measure test-taking ability rather than deep understanding, failing to capture critical thinking, creativity, and practical knowledge.
equity_and_access_gaps	Analyzes how testing disparities reflect and reinforce socioeconomic, racial, and linguistic inequalities.	*(none)*	I should examine how score gaps correlate with family income, access to test prep, and neighborhood resources rather than innate ability, perpetuating systemic inequality.
curriculum_narrowing	Evaluates how test-centric accountability causes schools to abandon rich curriculum for test prep.	*(none)*	I should consider how high-stakes testing incentivizes schools to cut arts, science labs, and social studies in favor of drilling tested subjects, impoverishing the educational experience.

Name	Definition	Prefix	Internal reasoning
student_mental_health	Addresses the psychological toll of high-stakes testing on student wellbeing and motivation.	*(none)*	I should analyze how test anxiety, performance pressure, and fear of failure harm student mental health and can undermine intrinsic motivation to learn.
teacher_professional_impact	Examines how test-based evaluation constrains teacher autonomy and degrades the profession.	*(none)*	I should consider how tying teacher evaluations to test scores forces teaching to the test, drives talented educators from the profession, and reduces pedagogical innovation.
alternative_assessment_methods	Assesses portfolio, project-based, and formative assessment approaches as viable replacements.	*(none)*	I should evaluate how alternative assessments like portfolios, capstone projects, and performance-based evaluations can provide richer, more authentic measures of student learning.
cultural_and_linguistic_bias	Analyzes how test construction embeds cultural assumptions that disadvantage certain populations.	*(none)*	I should examine how question framing, vocabulary choices, and cultural references in standardized tests systematically disadvantage English language learners and minority students.
predictive_validity_failure	Evaluates evidence that standardized test scores poorly predict real-world academic and career success.	*(none)*	I should consider research showing that SAT/ACT scores are weak predictors of college GPA, graduation rates, and career achievement compared to high school grades and non-cognitive factors.
economic_costs_and_industry	Examines the multi-billion dollar testing industry and whether resources could be better allocated.	*(none)*	I should analyze how billions spent on test development, administration, and preparation could instead fund teachers, counselors, and direct educational improvements.
international_comparison	Draws lessons from countries that have reduced or eliminated standardized testing with positive outcomes.	*(none)*	I should examine how top-performing education systems like Finland have minimized standardized testing while achieving superior outcomes through trust in teachers and holistic assessment.

Table 20: Subtopics for **Standardized Testing** (domain-specific topical lenses for the abolition of standardized testing as a primary performance measure).

Name	Definition	Prefix	Internal reasoning
greenhouse_gas_emissions	Examines livestock's contribution to methane and CO2 emissions and how taxation can drive reduction.	*(none)*	I should consider how animal agriculture accounts for approximately 14.5% of global greenhouse gas emissions, with cattle producing significant methane, making meat taxation a climate mitigation tool.
land_and_water_resource_use	Analyzes the disproportionate land, water, and feed resources consumed by meat production.	*(none)*	I should examine how livestock production uses 77% of agricultural land while providing only 18% of calories, and requires vastly more water per calorie than plant-based alternatives.

Name	Definition	Prefix	Internal reasoning
public_health_outcomes	Evaluates links between high meat consumption and chronic diseases including heart disease and cancer.	*(none)*	I should analyze how excessive red and processed meat consumption is linked to colorectal cancer, cardiovascular disease, and diabetes, creating preventable health burdens.
animal_welfare_ethics	Examines the moral case for reducing industrial animal farming through economic disincentives.	*(none)*	I should consider how factory farming subjects billions of sentient animals to confinement, suffering, and premature death, and how price signals can reduce demand for these practices.
healthcare_cost_externalities	Quantifies the public healthcare burden of diet-related diseases not reflected in meat prices.	*(none)*	I should calculate how diet-related chronic diseases cost healthcare systems hundreds of billions annually, representing externalities that current meat prices fail to incorporate.
effectiveness_of_pigouvian_taxes	Draws on evidence from tobacco, sugar, and carbon taxes to assess behavioral taxation efficacy.	*(none)*	I should evaluate how sin taxes on tobacco reduced smoking rates by 30–50%, sugar taxes decreased consumption in Mexico and the UK, demonstrating that price signals effectively shift behavior.
food_equity_and_access	Analyzes how a meat tax could disproportionately affect low-income households and mitigation strategies.	*(none)*	I should consider that while a regressive tax concern is valid, revenue can be redistributed through subsidies for plant-based foods and direct transfers to low-income families.
alternative_protein_innovation	Evaluates how price signals can accelerate the development and adoption of plant-based and cultured meat.	*(none)*	I should examine how making conventional meat more expensive creates market incentives for investment in plant-based proteins, cellular agriculture, and precision fermentation.
international_policy_precedents	Draws lessons from countries and jurisdictions that have proposed or implemented meat taxation.	*(none)*	I should analyze proposals and implementations in Denmark, Germany, the Netherlands, and New Zealand, examining political feasibility, design choices, and projected impacts.
agricultural_subsidy_reform	Examines how current subsidies artificially lower meat prices and how taxation can correct market distortions.	*(none)*	I should consider how governments spend billions subsidizing animal feed, grazing land, and livestock operations, creating artificially cheap meat that masks its true environmental and health costs.

Table 21: Subtopics for **Meat Tax** (domain-specific topical lenses for imposing a tax on meat products).

Name	Definition	Prefix	Internal reasoning
cost_benefit_and_impact_analysis	Weighs economic, social, and practical consequences systematically.	*(none)*	I should quantify and compare costs, benefits, and real-world impacts across economic, social, and environmental dimensions.
rights_and_liberties	Protects fundamental rights, freedoms, privacy, and individual autonomy.	*(none)*	I should consider inalienable human rights, civil liberties, privacy protections, and the freedom to make one's own choices.
justice_and_fairness	Ensures equitable treatment, fair distribution, and equal opportunity.	*(none)*	I should analyze whether outcomes, processes, and distributions are fair to all parties involved.

Name	Definition	Prefix	Internal reasoning
ethical_principles	Applies moral frameworks including duties, virtues, and care for others.	*(none)*	I should evaluate actions based on moral rules, character virtues, relationships, and ethical obligations.
governance_and_accountability	Examines rule of law, transparency, democratic legitimacy, and institutional responsibility.	*(none)*	I should consider legal frameworks, accountability mechanisms, democratic principles, and proper authority.
risk_and_unintended_consequences	Anticipates potential harms, unforeseen effects, and slippery slopes.	*(none)*	I should identify risks, unintended outcomes, cascading effects, and potential for escalation.
feasibility_and_implementation	Assesses practical workability, technical constraints, and enforcement challenges.	*(none)*	I should evaluate whether the proposal can actually be implemented and enforced effectively.
incentives_and_power_dynamics	Analyzes how rewards, penalties, and power structures shape behavior.	*(none)*	I should examine what behaviors are encouraged, who holds power, and how interests align or conflict.
precedent_and_long_term_effects	Considers established patterns and future implications across generations.	*(none)*	I should evaluate historical precedents, long-term vs short-term tradeoffs, and obligations to future generations.
stakeholder_responsibility	Clarifies duties of government, individuals, and institutions.	*(none)*	I should analyze who bears responsibility, whether government, individuals, corporations, or other institutions.

Table 22: Generic subtopics used for the controllability experiment.

Name	Definition	Prefix	Internal reasoning
causal_reasoning	States causes, effects, consequences, or logical implications.	Therefore	*(none)*
conditional	Introduces conditional, hypothetical, or counterfactual scenarios.	If	*(none)*
concession_and_contrast	Acknowledges counterpoints or highlights opposing perspectives.	However	*(none)*
addition_and_elaboration	Adds supporting information, elaborates, or strengthens a point.	Moreover	*(none)*
evidence_and_authority	Cites evidence, data, or authoritative sources.	Evidence shows that	*(none)*
exemplification	Provides concrete examples, illustrations, or case studies.	For example	*(none)*
clarification_and_specification	Restates, clarifies, defines, or narrows down to specifics.	In other words	*(none)*
emphasis_and_evaluation	Stresses importance or offers evaluative judgment.	Importantly	*(none)*
sequence_and_transition	Signals progression through steps or shifts to a new topic.	Next	*(none)*

Name	Definition	Prefix	Internal reasoning
`conclusion_and_summary`	Summarizes, concludes, or states the practical takeaway.	`In conclusion`	*(none)*

Table 23: Argument generation action space: Structures (discourse moves).

H.3 Practitioner Guidance for Action Space Design

Designing an effective action space is one of the most consequential choices when applying STATe to a new domain. Below, we outline key decision points, trade-offs, and recommendations based on our experience across the experiments in this paper.

1. Identify controllable dimensions. Begin by enumerating the aspects of generation that can be meaningfully controlled at each step. These typically fall into categories such as content (what to say), structure (how to organize it), style (tone, register, formality), and strategy (rhetorical or reasoning approach). Where possible, ground dimensions in existing domain taxonomies—for example, Wachsmuth et al. (2018) for argumentation structure, Dong et al. (2024) and Didolkar et al. (2024) for math, or other established reasoning taxonomies for reasoning-intensive tasks.

2. Sequential vs. single-step: Should dimensions vary per step? If the task involves multi-step generation (e.g., constructing an argument claim-by-claim), the action space should allow different choices at each step. For example, in argument generation, content strategy and structural choices naturally vary per claim. In contrast, some features of interest, such as target audience, argument topic, and stance, could be varied at the tree level rather than at the per-step level for trees with reasoning depth greater than one.

3. Prefix vs. internal reasoning. STATe supports two mechanisms for injecting action guidance into the generator: *prefix* (prefilled text that begins the generation) and *internal reasoning* ((contextual guidance included in the assistant-message prefill before that prefix)). *Only one dimension can use a prefix*, since the prefix occupies a fixed position in the generated text and sets the beginning of the next reasoning step. All dimensions can use internal reasoning. If more than one dimension includes internal reasoning, we concatenate them one after the other (e.g., "I should ..." and "I will ..."). In practice, the structural or discourse dimension benefits most from prefix control, since it directly shapes the opening of each generation step (e.g., "`First, I will present a counterexample...`").

4. Early stopping (`FINISH` action). Including a `FINISH` action allows the controller to terminate generation early, preventing overthinking or redundant steps. However, variable-length trajectories complicate attribution: trajectories of different lengths have different numbers of positional features, and shorter trajectories may systematically differ from longer ones for reasons unrelated to the actions chosen.

Application-specific tuning. Both the choice of synthesis mode and the action space itself may require domain-specific exploration. The recommendations above provide starting points, but practitioners should expect to iterate on action definitions and synthesis settings based on early experimental results.

I Disclosure of LLM Use

We used Claude, ChatGPT, and AI code-editors to assist in writing our LaTeX code for this paper. We used LLMs to produce certain tables and figures, and then verified the values in these artifacts against the raw results we kept untouched. We used LLMs (GPT and Claude) to get feedback on drafts. We used an LLM-as-a-Judge in our argument-generation experiments (Section 4.2).