STATe-of-Thoughts: Structured Action Templates for Tree-of-Thoughts[†]

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

[1]Technion — Israel Institute of Technology [2]Princeton University
[3]Independent [4]Hebrew University of Jerusalem

[*]Equal contribution

February 16, 2026

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 explainability. We present **STATe-of-Thoughts** (STATe), an interpretable ITC method that *searches* over high-level reasoning patterns. STATe replaces stochastic sampling with discrete and interpretable textual interventions: 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 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 directly toward them. Together, these results establish STATe as a practical framework for generating high-quality, diverse, and interpretable text. Our framework is available at state-of-thoughts.

1 Introduction

Many applications of LLMs require more than generating high-quality responses: they need systematic and interpretable control over how text is produced [42, 35, 87]. 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 [102, 28, 29, 87, 88, 43, 20]. Similarly, in creative writing, researchers are concerned with generating diverse yet high-quality outputs that satisfy the preferences of the audience [26, 64, 115, 122]. In both settings, the challenge is to produce text that varies systematically along dimensions of interest while maintaining coherence and quality.

ITC methods address part of this challenge by allocating additional compute for LLM reasoning [114, 60, 75, 22, 81] and for producing multiple candidate responses [11, 99, 112, 16, 103]. Tree-based methods [6, 46, 45] like Tree of Thoughts (ToT by Yao et al. [117]) further enhance quality by branching on intermediate thoughts and pruning less-promising reasoning trajectories. However,

[†]The work described in this manuscript is subject to a pending patent application.

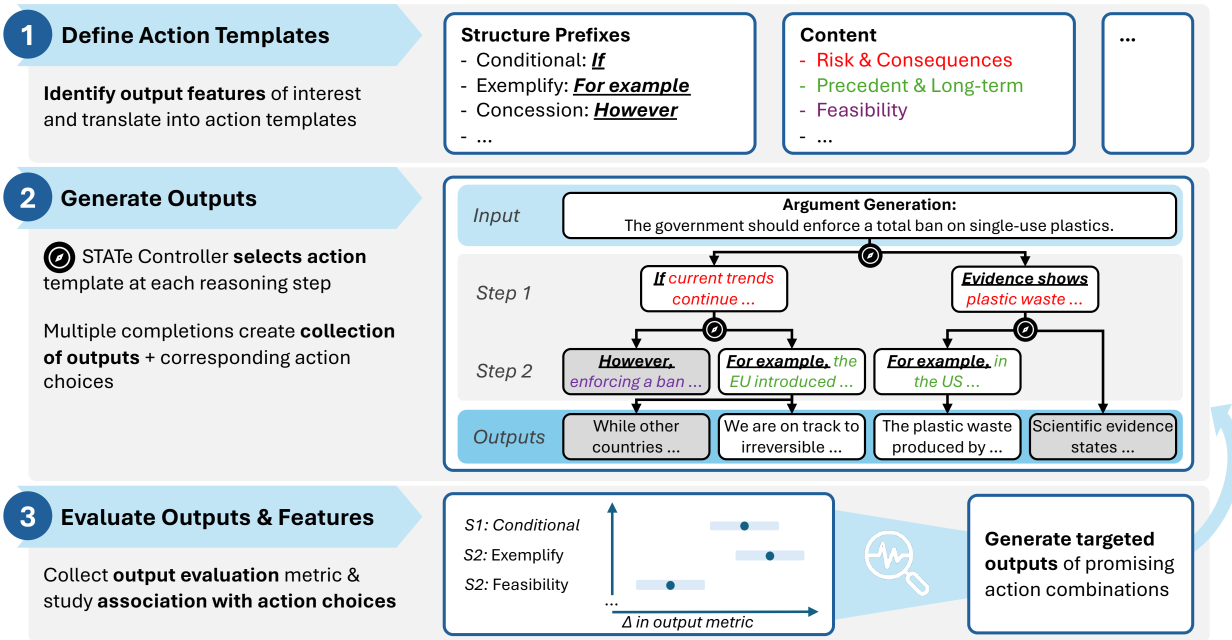

Figure 1: **STATe for argument generation.** Tasked with generating persuasive arguments in favor of banning single-use plastics, STATe's workflow involves the following steps: (1) Define action templates that control output features of interest, such as structural prefixes and content themes. (2) Generate 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.

these methods rely primarily on temperature-based sampling for diversity, which yields limited meaningful variation [122, 53]. 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 [72]. As a result, they provide limited control over *which* high-level decisions are explored and limited insight into which decision patterns drive success or failure.

To induce interpretable yet diverse sampling, we prepend diverse 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 [6, 46]. We illustrate STATe's complete three-step workflow in Figure 1 through the lens of argument generation.

We compare STATe to existing ITC methods in both creative writing and argument generation. On NoveltyBench [122] (Section 4.1), we found that STATe produces up to twice as many semantically distinct outputs as the best baseline across seven LLMs from three model families (Qwen3 [116], Nemotron-3 [74], and Ministral-3 [67], Section 5.1). In our case study on argument generation (Section 4.2), we found that sequential action features are highly predictive of argument quality on held-out data (Section 5.2.1), and that model-guided trajectory selection allows for generating

[1] Unlike latent interventions [3, 32, 4], interventions in STATe are explicit text prefixes and thus directly auditable.

high-quality outputs from promising yet unexplored regions of the action space (Section 5.2.2). Our contributions, therefore, are the following:

1. A controllable and interpretable ITC framework based on explicit action-space search.
2. A mechanism for diverse generation which outperforms high-temperature sampling.
3. An attribution framework that uses action traces to predict output quality, identify promising unexplored regions of the action space, and steer generation toward them.

2 Background

2.1 Inference-Time Compute

The conventional approach to using LLMs involves direct input-to-output generation, where a model maps an input sequence x directly to an 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 [94, 78, 95], sycophancy [91], and other biases [52, 77]. These limitations are compounded in tasks that require complex reasoning or multi-step solutions [114, 97]. Building on the intuition that human reasoning benefits from more "time to think" [54], Inference-Time-Compute methods provide LLMs with additional "reasoning" tokens (reasoning depth [97, 75, 81, 73][2]), and permit parallel reasoning attempts (reasoning breadth [11, 99, 112, 16, 103]).

Chain-of-Thought (CoT) reasoning [114, 60] 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 represents the chain of reasoning steps and y the final answer. CoT enhances performance on complex reasoning tasks [97, 22, 75], and models specifically optimized for reasoning demonstrate even greater improvements [75, 22, 63]. However, standard CoT implementations remain brittle: errors can propagate through the reasoning chain, and there is no principled mechanism to revisit decisions or explore alternative strategies.

Rather than scaling depth, Best-of-n methods [11, 99] scale *breadth* by generating n independent candidate outputs and selecting the best according to some scoring criterion. This enhances robustness by reducing the impact of individual generation failures.[3] Self-Consistency [112, 16, 103] addresses CoT's brittleness by sampling multiple CoT reasoning paths and selecting the most consistent answer through majority voting. When sampling more than one completion from an LLM, we refer to this operation as *branching* [117]. Formally, we express branching as a single operation that produces multiple complete reasoning chains 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, affording limited control over which intermediate choices are explored. Moreover, inducing diversity across branches through high-temperature sampling often yields homogeneous outputs or degrades quality [72, 53, 122].

2.2 Tree of Thoughts

Tree of Thoughts (ToT) [117] combines both strategies: scaling depth through multi-step reasoning (improving reasoning quality) and scaling breadth through branching at intermediate points (enhancing robustness). ToT reframes generation as search over a tree of partial reasoning steps, enabling branching at multiple intermediate points, evaluation of progress, and pruning of unpromising paths.

[2] Reasoning effort for LLMs manifests as more output tokens [81].

[3] Generation failures include LLM refusals, exceeding the context limit, or parsing errors over LLM outputs.

Each node represents a state $s_i := [x, z_1, \ldots, z_i] = [x, Z_i]$,[4] capturing a partial solution with the input and reasoning steps so far. 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, when sampling n thoughts at step i with language model p_θ:

$$\begin{aligned} p_\theta(z_i) &:= p_\theta(z_i \mid x, Z_{i-1}; n, \texttt{temp}), \\ \{z_i^1, \ldots, z_i^n\} &\sim p_\theta(z_i). \end{aligned}$$

where $Z_{i-1} = [z_1, \ldots, z_{i-1}]$ represents the reasoning steps so far.

ToT provides enhanced robustness through two key mechanisms. First, intermediate evaluation and pruning identify and eliminate unproductive branches early, preventing error propagation. Second, systematic exploration of different trajectories increases the likelihood of discovering valid solutions even when some branches fail.[5] To evaluate intermediate and final reasoning steps, ToT methods use LLM-as-a-Judge [123, 65]. Process Reward Models (PRMs) [117, 66, 110], which score intermediate reasoning, are defined as $V(Z_i \mid x) \to [0, 1]$ where $i \leq d$. Outcome Reward Models (ORMs) [123, 57, 58], are defined as $V(y \mid x) \to [0, 1]$.

Traditional ToT implementations face two primary limitations. First, despite sampling at high temperatures to promote diversity, branches tend to cluster around similar content with only minor variations [117, 45, 53, 122]. Second, ToT implementations lack adaptive termination mechanisms, executing for a predetermined number of steps regardless of reasoning progress, leading either to overthinking after convergence [73, 51] or to premature termination.

3 Methods

STATe extends ToT [117, 6, 46] with three methodological contributions that improve both output diversity and controllability while retaining quality. First, STATe replaces 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. Second, STATe supports both verifiable [63] and task-specific LLM-as-a-Judge evaluators to reliably score and select among diverse candidates. Combining both diverse generation and a reliable evaluator (for rejection sampling) helps mitigate the quality-diversity trade-off [72]. Third, STATe tracks actions along a trajectory, which enables systematic analysis of which textual features and reasoning patterns drive performance. This kind of analysis allows researchers to study associations between controllable, concept-level interventions and downstream outcomes [41, 1].

3.1 Tree of Thoughts Components

STATe instantiates ToT as a modular Plan → Generate → Evaluate → Select loop (Figure 2). 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 (beam selection). We present the full process in Algorithm 1.

[4] 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.

[5] 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.

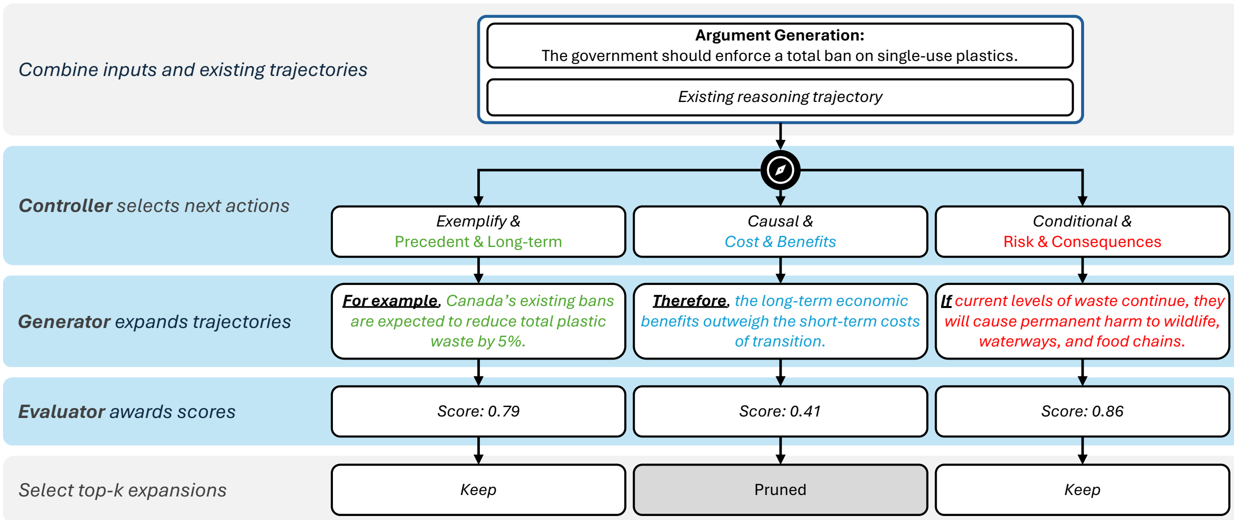

Figure 2: Stylized example of STATe's Plan→Generate→Evaluate→Select loop. The controller plans which actions to explore, the generator expands candidate trajectories, the evaluator scores them, and beam selection retains the top-k states.

3.1.1 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 D) and providing values for the tool's arguments (if any). We discuss how to design action templates for new problems in Appendix D.3. Each argument in a tool has a finite domain, which naturally encodes structured action-space dimensions. In our argument generation example (Figure 2), the actions encode *structure* and *content* dimensions, per Wachsmuth et al. [109]. Executing the tool returns a structured intervention, i.e., a prefix and internal reasoning guidance,[6] that is injected into the next generation step (see Table 1). This mirrors the iterative tool-use paradigm in ReAct [118], 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.

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.[7] Formally, we define the controller output as:

$$\{a_i^1, \ldots, a_i^n\} = \mathrm{C}(s_{i-1}, \mathcal{A}, n) \tag{1}$$

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}, so we can formalize action selection as:

$$\{a_i^1, \ldots, a_i^n\} = \arg \max_{A \subset \mathcal{A}, |A|=n} \sum_{a_i \in A} Q(s_{i-1}, a_i) \tag{2}$$

We implement two kinds of controllers: one that uses a generative LLM to produce tool calls (Appendix B.1.1), and another that uses a reranker LLM [121] to pick among all possible tool

[6] When an action combines multiple dimensions (e.g., both a structure choice and a content choice), only one dimension can introduce a prefix, since the prefix sets the beginning of the next reasoning step. However, all dimensions can contribute to the internal reasoning guidance (by concatenating claims like "I should ..." or "I will ..."').

[7] Branches from parent nodes lead to children nodes as described in Algorithm 1.

calls (Appendix B.1.2). Additionally, we introduce *early stopping*. If the controller determines that reasoning is sufficient, it can select a dedicated `FINISH` action, allowing the state to proceed directly to final answer generation. This is represented in our controller as an additional tool that takes no arguments. If selected, we pre-fill the next generation with delimiters signifying the end of the reasoning process and the beginning of the answer generation process. This mechanism helps prevent "overthinking" where additional steps become degenerate after the model has effectively converged [69, 85, 100, 51, 73].

3.1.2 Generator

Once the controller selects the next actions $\{a_i^1, \ldots, a_i^n\}$ for a given parent state s_{i-1}, we generate reasoning steps for its children conditioned on these interventions. For each action $a_i^j \in \{a_i^1, \ldots, a_i^n\}$, we "execute" the corresponding tool to obtain text guidance $a_i^j()$, typically consisting of *internal reasoning* to guide the next step and a *prefix* fixing the beginning of the next step. We append these to the LLM's assistant message as a prefill (see Table 1 for examples). Formally, 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 z_i^j for each action a_i^j as:

$$z_i^j \sim p_\theta\big(z \mid x, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp}\big)\,[\texttt{:stop\_token}] \tag{3}$$

The prefill operation ensures that the model's generation begins with the intervention text, biasing the direction of reasoning along the desired dimension. Each generated thought z_i^j is combined with the current state to create a child state $s_i^j = [s_{i-1}, z_i^j]$ that can be expanded further. We determine when to stop a generation through *stop tokens*: `</step>` if generating a reasoning step, and `</answer>` if generating a final answer.

Once the maximum reasoning depth d is reached or the controller selects the `FINISH` action, STATe reaches the *synthesis* step that transforms these traces into cohesive final outputs. This step conditions on the input x and the complete reasoning trace Z_{i-1}, generating a final output y^j that integrates the intermediate claims into a coherent whole:

$$y^j \sim p_\theta(y \mid x, \texttt{prefill}(Z_{i-1}); \texttt{temp})\,[\texttt{:stop\_token}] \tag{4}$$

We define four synthesis modes that vary in how strictly the output preserves the intermediate reasoning. *Strict synthesis* concatenates reasoning steps verbatim with minimal connectives. *Faithful synthesis* permits rephrasing while preserving order and structure. *Restructured synthesis* allows free reorganization using the trace as source material. *Conclusion synthesis* treats the trace as internal guidance only, with no constraints on the final output. The choice of synthesis mode reflects a trade-off between action attribution and output quality; full specifications appear in Appendix E.3.

3.1.3 Evaluator

After generating child states from each parent, we evaluate their quality using either score-based LLM-as-a-Judge models [123, 57, 58, 13, 68], or verifiable rewards [63, 39, 104]. We employ two types of evaluators corresponding to the two types of states in our search tree.

For intermediate states s_i where $i \leq d$, we use a Process Reward Model (PRM) defined as $V_{PRM}(s_i) := V(Z_i|x) \to [0, 1]$. The PRM scores the reasoning chain based on its current coherence (backward compatibility) and projected final answer quality (forward compatibility), enabling early pruning of unpromising paths. On the other hand, for complete solution states $s_i = [x, Z_{i-1}, y]$ (where $i \leq d$) we use an Outcome Reward Model (ORM) defined as $V_{ORM}(s_i) := V(y|x) \to [0, 1]$.

First step	Intermediate step	Final step
<thinking> <step> ## internal_reasoning I should identify risks, unintended outcomes, cascading effects, and potential for escalation. ## 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 evaluate historical precedents, long-term vs short-term tradeoffs, and obligations to future generations. ## claim For example, Canada's existing single-use plastic bans are expected to reduce total waste by 5%...	<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, Canada's existing bans... </step> ... </thinking> <answer>

Table 1: 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 D.2.

This scores the quality of the final answer, considering task-specific criteria such as instruction adherence, persuasiveness, coherence, and stylistic appropriateness.

We support three evaluator implementations: a generative LLM-as-a-Judge which scores candidates against a rubric (Appendix B.2.1), a re-ranker [121] LLM-as-a-Judge, which assigns latent relevance scores for candidates (Appendix B.2.2), and a deterministic verifier [63] (Appendix B.2.3).

3.2 Beam Search Algorithm

We present the complete STATe-of-Thoughts beam search in Algorithm 1.[8] The algorithm initializes with the input x and iteratively expands states through controller-guided interventions. The algorithm terminates when all states have produced final answers or their trajectory has been pruned, at which point the highest-scoring final states according to the ORM are returned. Each of STATe's components[9], the Generator (G), Controller (C), PRM (V_{PRM}), and ORM (V_{ORM}), include dedicated prompts, which we present in Appendix E. Notably, the synthesis type used to generate final outputs is included in the system prompt of G (Appendix E.2).

[8] Algorithm 1 is simplified relative to our implementation. In practice, we parallelize LLM calls from the Controller, Generator, and Evaluator across all nodes in a layer with vLLM [61].

[9] STATe's components correspond to DSPy [56] Modules, which convert Signatures (input/output and instruction specifications) and inputs to prompts (via Adapters), call the LLM, and parse the response according to the signature.

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

Require: Problem input x, generator G, controller C, process evaluator V_{PRM}, outcome evaluator V_{ORM}, action space $\mathcal{A}$, branching factor n, beam width k, maximum 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**
    if $i = d + 1$ **then** ▷ Final layer: force termination
      $\{a_i^1\} \leftarrow \{\texttt{FINISH}\}$
    else
      $\{a_i^1, \ldots, a_i^n\} \leftarrow C(s_{i-1}, \mathcal{A}, n)$ ▷ Controller selects up to n actions
    end if
    for each action a_i^j **do**
      if a_i^j is `FINISH` **then**
        $y^j \sim G(s_{i-1}, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp})[\texttt{:stop\_token}]$ ▷ 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
        $z_i^j \sim G(s_{i-1}, \texttt{prefill}(Z_{i-1}, a_i^j()); \texttt{temp})[\texttt{:stop\_token}]$ ▷ Generate thought
        $s_i \leftarrow [s_{i-1}, z_i^j]$ ▷ Append thought to existing 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** ▷ All branches terminated via early stopping
    break
  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

3.3 Attributing Outcomes to Controller Actions

A key advantage of STATe-of-Thoughts is its ability to attribute differences in outcomes to specific controller actions. As each branch in the reasoning tree is associated with a logged sequence of actions, we can analyze which actions lead to higher-quality outputs. However, estimating the causal effect of action choices on output quality is complicated by the sequential nature of decision-making. Actions are selected and executed conditional on prior actions of a given trace, confounding simple correlation-based analysis. Hence, for now, we focus on association rather than causation, aiming to identify action patterns that consistently correlate with better or worse outcomes.

3.3.1 Problem Formulation

Let $\tau = (a_1, a_2, \ldots, a_n, y)$ denote a complete action trace leading to the final output y. We are interested in predicting final output quality Y from these action trajectories. One central question this allows us to explore is whether the *sequential structure* of actions matters beyond their mere presence. If ordering is unimportant, a simple "bag-of-actions" representation suffices. This is analogous to what a topic model could capture about argument composition [87]. However, if the *timing* of an action affects outcomes (e.g., starting with a concession versus ending with one), then sequential features that encode positions and transitions should explain additional variance in Y and allow for better predictions.

3.3.2 Presence-Based Model (Bag-of-Actions)

The presence-based model represents each trajectory using binary indicators for whether each action type appeared anywhere in τ:

$$\mathbf{1}_a(\tau) = \begin{cases} 1 & \text{if action type } a \text{ appears in } \tau \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

This produces a feature vector $\mathbf{x}^{\text{presence}} \in \{0,1\}^{|\mathcal{A}|-1}$ that ignores the order and frequency of actions. We fit a linear model:

$$Y_i = \alpha + \mathbf{x}_i^{\text{presence}} \boldsymbol{\beta} + \epsilon_i \tag{6}$$

The coefficient $\boldsymbol{\beta}$ captures the average difference in outcome quality when action a is present versus absent, relative to the reference category and controlling for other actions. This kind of model serves as a baseline representing what could be learned from action composition alone, without information on action order.

3.3.3 Sequential Model

The sequential model extends beyond the presence-based representation with features that capture the sequential ordering of actions. We consider two core feature types:

Position Features For each action a, we create indicators for a occurring at each reasoning step position:

$$\mathbf{1}_{a,k}(\tau) = \begin{cases} 1 & \text{if action } a \text{ occurs at step } k \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

This captures whether an action's effect depends on *when* it occurs, for instance, whether deploying action a early versus late in the trajectory produces different outcomes.

Transition Features We create indicators for action sequences between consecutive steps:

$$\mathbf{1}_{a \to a'}^{k \to k+1}(\tau) = \mathbf{1}_{a,k}(\tau) \cdot \mathbf{1}_{a',k+1}(\tau) \tag{8}$$

This captures sequential patterns: the extent to which action a followed by action a' affects outcomes. In principle, longer n-gram transitions or other complicated interactions could be included, but we focus on bi-grams for tractability.

The sequential model combines these feature types in $\mathbf{x}_i^{\text{sequential}}$ such that:

$$Y_i = \mathbf{x}_i^{\text{sequential}} \boldsymbol{\beta} + \epsilon_i \tag{9}$$

When the action space contains multiple dimensions (e.g., both *what* to say and *how* to say it), additional feature types arise, including cross-dimension interactions at each step. We describe our specific experiment instantiation in Section 4.2.

3.3.4 Model Comparison via Regularized Regression

The sequential model contains substantially more features than the presence-based model, creating a risk of overfitting and numeric instability. Moreover, because the controller and evaluator may focus on a subset of the action space, some features or feature combinations may be irrelevant. We address these issues using LASSO regression [106], which performs automatic feature selection by driving irrelevant coefficients to zero:

$$\hat{\boldsymbol{\beta}} = \operatorname{argmin}_{\boldsymbol{\beta}} \left\{ \frac{1}{2N} \sum_{i=1}^{N} (Y_i - \mathbf{x}_i^{\text{sequential}} \boldsymbol{\beta})^2 + \alpha \|\boldsymbol{\beta}\|_1 \right\} \tag{10}$$

This allows us to identify relevant action patterns while controlling model complexity. In practice, one might want to design broad action spaces that capture many possible strategies, then explore how these actions affect perceived argument quality for different target audiences. When it is unclear *a priori* which actions or sequences will be associated with differences in outcomes, this approach allows data-driven selection of the most relevant patterns.

4 Experiments

We evaluate STATe in two complementary settings that probe its capacity for diversity, interpretability, and controllability. First, we compare STATe to existing ITC methods on NoveltyBench [122] to test whether structured interventions can induce semantic *diversity* without degrading quality (Section 4.1). Next, in our Argument Quality experiment, we test whether STATe's *interpretable* action sequences provide sufficient signal to predict the quality of persuasive arguments. We further evaluate whether learned associations can guide *controllable* targeted generation toward promising regions of the action space (Section 4.2).

4.1 Generating Diverse Responses

NoveltyBench [122] measures whether methods can increase diversity without collapsing quality. The curated set includes 100 prompts across four categories: *randomness* (e.g., "Roll a make-believe 20-sided die"), *factual knowledge* with underspecified queries (e.g., "List a capital city in Africa"), *creative writing* (e.g., "Write a short story about a time traveler"), and *subjectivity* (e.g., "What's the best car to get in 2023?"). For each prompt, models generate 10 responses, which are then partitioned into functional equivalence classes using a fine-tuned DeBERTa-v3-large classifier [47] trained on $1,000$ human-annotated pairs. Two generations are considered functionally equivalent if a user who has seen one would gain little additional value from seeing the other. *Mean Distinct* counts the number of unique equivalence classes across the 10 generations, quantifying semantic diversity.[10]

We compare the diversity of outputs from best-of-n (with I/O and CoT prompting) against ToT [6, 46] and STATe, using open-source models from three families: Qwen3 [116], Nemotron-3

[10]NoveltyBench also includes a *Mean Utility* metric, which we include in Appendix C.1.

[74], and Ministral-3 [67]. When using STATe, we utilize a Qwen3-8B-Reranker [121] for both a Re-Ranker Controller (Appendix B.1.2) and a Re-Ranker Evaluator (Appendix B.2.2).[11] We use depth-1 trees with wide branching ($n = 10$, $k = 10$), which produces a single reasoning step and then a final answer in each candidate trajectory. Each configuration is repeated across 10 random seeds to measure variance from sampling randomness. We also include baselines that explicitly expose action spaces in prompts (as input fields of DSPy signatures [56]) to isolate the benefit of controller-guided interventions from simply providing additional task structure. As NoveltyBench includes only a single "test" split, we perform initial experiments on a "train" subset of 10 examples to refine our system and its prompts and report results on the remaining 90 examples.

Notably, this study examines the diversity of branching operations as a function of (1) whether we use a ToT template, and (2) whether we include the action space as an input, and (3) whether the controller interventions guide generations. Diversity as a function of tree depth is outside the scope of this experiment. STATe's internal evaluator does not affect search or final candidate selection in this configuration (all 10 outputs are returned). The action space we use here combines two dimensions with 5 choices each, yielding 25 possible action combinations per step (Appendix D.1). The first dimension, *personality traits*, follows the Big Five model [40]. The second dimension, *target audience*, specifies the demographic age to appeal to. Each action provides internal reasoning guidance (Section 3.1.2) that steers generation toward the specified persona or audience. We sweep three temperature regimes per model: low, medium, and high.[12]

4.2 Argument Quality

We also evaluate STATe on argument generation, using the single-use plastic ban proposition introduced in Figures 1 and 2. This setting provides a convenient testing ground for analyzing how different action choices affect GPT-5-mini's [96] perception of an argument's quality. Argument generation is particularly well-suited for attribution analysis because each action (selecting what claim to develop or how to structure it) can manifest directly and sequentially in the final output. The resulting argument generally comprises the generated parts in the order they were produced, creating a clear association from controller decisions to output features. We use *strict synthesis*, *faithful synthesis*, and *restructured synthesis* (Section 3.1.2) in this experiment. Following Wachsmuth et al. [108], we operationalize *content* interventions as stock-topics to discuss, and *structure* interventions as discourse-relations to use [82, 113]. We define ten dimensions each for the content and structure action spaces, giving us a total of 100 action choices at each branching point. The full action space templates are provided in Appendix D.2.

We use STATe with Qwen3-30B-A3B-Instruct to generate 5,000 arguments in favor of banning single-use plastic per synthesis method. Specifically, we use STATe trees with depth 3 and beam width 250, yielding 250 unique trajectories and corresponding final arguments per tree. We generate 20 such trees initialized with different random seeds to achieve multiple, different realizations of some overlapping trajectories, rather than a single tree with a very large beam width. To measure argument quality, we prompt our LLM-judge to evaluate 50,000 random pairwise comparisons per argument collection, indicating which argument it finds more effective. We then fit a Bradley-Terry (BT) model [9] to these judgments and use the resulting standardized ranks as the outcome variable Y in our regression analyses.

We split the dataset into training (60%) and test (40%) sets for model fitting and evaluation. We

[11] In our experiments, STATe uses 2 GPUs, since we load the generative model on one, and the reranker model on the other. However, when STATe uses a generative evaluator, it requires only a single GPU.

[12] $T \in \{0.5, 0.7, 1.0\}$ for most models; $T \in \{0.1, 0.3, 0.5\}$ for Ministral-3, which Mistral recommends using with lower temperatures.

instantiate the sequential model (M2) with two action dimensions: *structure* (discourse-relations) and *content* (topical focus). For this multi-dimensional setting, we consider three feature types:

1. **Position per dimension features:** whether a particular structure or content choice at step k affects quality.
2. **Within-step interactions:** whether specific structure–content pairings at each position are particularly effective.
3. **Within-dimension transitions:** whether sequential patterns *within* each dimension (e.g., contrastive → causal structure, or feasibility → risk & consequences) affect outcomes.

We restrict transitions to length two and to within-dimension pairs for interpretability and to prevent extremely sparse features. For the presence-based models (M1a–c), we use ordinary least squares regression. For the sequential model (M2), we use LASSO, with the regularization parameter α selected via line search and 10-fold cross-validation on the training set, to handle the larger and sparser feature space. Note that the features of M2 can linearly replicate all presence-based features used in M1a–c; that is, M2 has access to a strict superset of the M1 feature space. Bootstrap resampling (1,000 iterations) provides 95% confidence intervals for test-set R^2 values. We report an ablation controlling for argument length in Appendix C.2.2.

5 Results

5.1 Structured Textual Interventions Improve Diversity

STATe consistently achieves the highest diversity (Mean Distinct) across all model families and temperature regimes. Our results, depicted in Table 2 (and more comprehensively in Appendix C.1, Table 4), demonstrate that action-guided interventions outperform temperature-based stochastic sampling for inducing semantic diversity. For Qwen3-30B-A3B-Instruct at the recommended temperature ($T = 0.7$), STATe produces 5.02 ± 0.09 distinct generations out of 9, compared to 3.36 ± 0.12 for the next-best method (Baseline CoT with Action Space) and just 2.44 ± 0.07 for standard CoT. In Appendix C.1 we show that STATe's wins on diversity generalize not only across model sizes, but also across model families. Notably, simply exposing the action space in prompts ("I/O w/ Action Space" and "CoT w/ Action Space") provides moderate diversity improvements over vanilla baselines, confirming that explicit diversity dimensions help, but controller-guided selection is necessary to fully realize the benefit.

Method	T=0.5		T=0.7		T=1.0	
	D	**U**	**D**	**U**	**D**	**U**
I/O	1.48 ± 0.03	2.28 ± 0.03	1.67 ± 0.04	2.45 ± 0.05	2.01 ± 0.05	2.64 ± 0.04
CoT	2.15 ± 0.07	2.65 ± 0.07	2.44 ± 0.07	2.85 ± 0.07	2.8 ± 0.08	3.07 ± 0.06
I/O w/ Action Space	1.74 ± 0.05	2.23 ± 0.05	2.01 ± 0.05	2.41 ± 0.07	2.44 ± 0.07	2.71 ± 0.04
CoT w/ Action Space	3.07 ± 0.11	3.15 ± 0.06	3.36 ± 0.12	3.28 ± 0.1	3.74 ± 0.09	3.49 ± 0.11
ToT	2.51 ± 0.09	2.87 ± 0.05	2.81 ± 0.08	3.07 ± 0.06	3.16 ± 0.12	3.32 ± 0.08
ToT w/ Action Space	3.16 ± 0.08	3.29 ± 0.05	3.25 ± 0.08	3.36 ± 0.07	3.61 ± 0.1	3.61 ± 0.11
STATe of Thoughts	$\mathbf{4.87 \pm 0.1}$	$\mathbf{3.77 \pm 0.12}$	$\mathbf{5.02 \pm 0.09}$	$\mathbf{3.83 \pm 0.11}$	$\mathbf{5.39 \pm 0.11}$	$\mathbf{4.01 \pm 0.11}$

Table 2: NoveltyBench diversity (D) and utility (U) for Qwen3-30B-A3B across ITC methods and temperatures (T). We report mean±std over 10 seeds. See Appendix C.1.2 for additional results.

The diversity gains from STATe generally benefit the utility metric that NoveltyBench measures (Appendix C.1.1). Table 2 shows that at temperature $T = 0.7$ Qwen3-30B, STATe achieves 3.83±0.11

Mean Utility versus 3.36 ± 0.07 for ToT with Action Space, indicating that the additional diversity does not degrade quality. In Appendix C.1.2 we show that STATe's wins in utility on Qwen3-30B generalize to 3 of 5 models as well.[13] Despite the two outliers, the general trend is clear: STATe achieves superior diversity–utility trade-offs across model families, temperatures, and model scales, validating the hypothesis that discrete action-space search enables controllable exploration without sacrificing output quality.

5.2 Action Sequences Help Predict Argument Quality

5.2.1 Predictability

We apply the attribution framework from Section 3.3 to the argument generation trajectories described in Section 4.2. Figure 3 compares the baselines based on presence (M1a-c) with the sequential model (M2) in three synthesis settings.

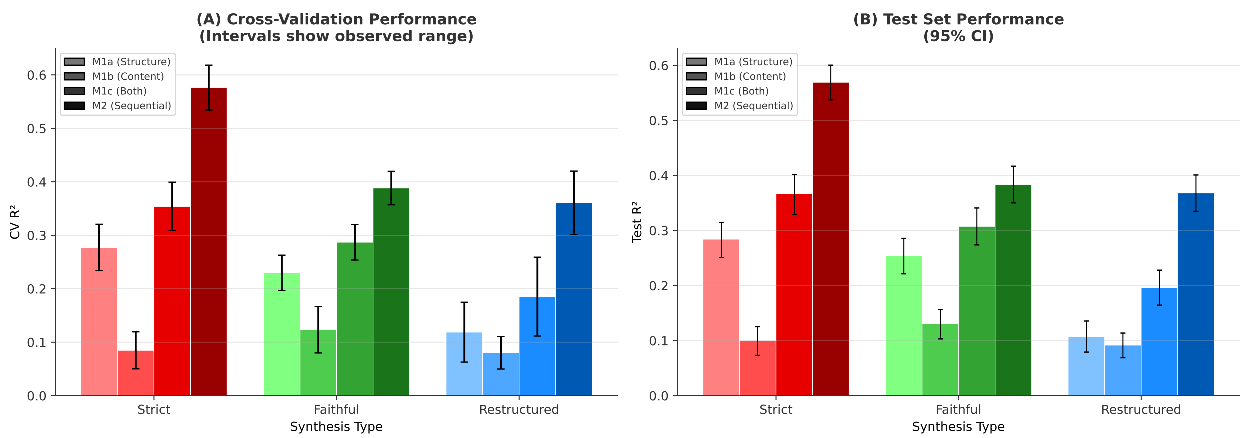

Figure 3: Predictability of argument quality from controller actions. (A) Cross-validation R^2 across model variants: M1a (structure presence only), M1b (content presence only), M1c (both), and M2 (full sequential model with position effects and transitions). (B) Held-out test set performance with 95% bootstrap confidence intervals. The sequential model consistently outperforms presence-based baselines, with the largest absolute predictability under strict synthesis.

Across all synthesis modes, the sequential model (M2) substantially outperforms the presence-based baseline (M1a-c): gains range from +25% (faithful) to +88% (restructured) relative improvement in R^2 on the test set. This demonstrates that the *timing* and *order* of actions matter beyond simply the *presence* of actions. In other words, the temporal structure of controller decisions carries predictive information about output quality. The full comparison, including confidence intervals and feature counts, is shown in Table 5 of Appendix C.2.

Further, the results reveal a clear predictability gradient across synthesis modes. Under *strict synthesis*, where the final argument should preserve exact wording from reasoning steps, the sequential model (M2) achieves $R^2 = 0.57$ on test data; the controller's action choices strongly predict argument quality as judged by pairwise comparisons.[14] Under *faithful synthesis*, which permits light rephrasing, predictability decreases to $R^2 = 0.38$. Under *restructured synthesis*, which allows free reorganization,

[13]The two models where STATe didn't achieve the highest diversity were Qwen3-4B and Nemotron-3-30B, where Baseline CoT w/ Action Space outperformed STATe by a low margin (as little as 0.01-0.12). See Appendix C.1.2 for additional discussion on this.

[14]We removed 6 of the 5000 strict synthesis arguments that were exact duplicates of other arguments in the dataset.

predictability is $R^2 = 0.37$. The more faithful the argument is to the reasoning steps that precede it, the better the predictability based on action features.

5.2.2 Targeted Trajectory Exploration

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 $\sim$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, 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[15] 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 [87, 36]. 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 detailed in the ablation in Appendix C.2.2. In particular, the original top 5% arguments tend to be disproportionately long, creating a confounder that would affect any direct comparison [27]. We construct length-matched evaluation sets using greedy pairing of arguments of similar length. [16] This yields balanced datasets ranging from 172 to 438 arguments, depending on synthesis type and baseline. We evaluate these datasets by running 5,000 random pairwise comparisons within each and calculating new BT scores.

Table 3 shows that targeted arguments substantially outperform both the random baseline (77–81% win rate) and the topic-presence baseline (64–77% 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 (19–52% 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. Here too, 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

[15] 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.

[16] 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 10, 11, and 12.

Baseline	Synthesis	N	Comparisons	Win (T)	Top-10	Top-100
Random	Strict	310	5000	77.8%	9/10	78/100
M1b (Topic Presence)	Strict	342	5000	67.6%	7/10	66/100
Original Top 5%	Strict	242	5000	51.7%	6/10	54/100
Random	Faithful	408	5000	81.2%	9/10	87/100
M1b (Topic Presence)	Faithful	398	5000	77.3%	9/10	81/100
Original Top 5%	Faithful	172	5000	39.6%	7/10	38/100
Random	Restructured	438	5000	77.4%	9/10	80/100
M1b (Topic Presence)	Restructured	428	5000	63.5%	6/10	68/100
Original Top 5%	Restructured	186	5000	19.3%	2/10	23/100

Table 3: Targeted Trajectory Exploration: Evaluating new, targeted vs. new baseline explorations. N is the total number of arguments in the length-matched dataset (balanced: $N/2$ targeted, $N/2$ baseline). Comparisons count is the number of random pairwise comparisons evaluated. Win (T) is the share of comparisons between targeted and baseline arguments that the targeted argument won. 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.

that when the goal is to find the very best arguments, M2-guided trajectory selection offers a promising approach.

6 Discussion

We developed STATe-of-Thoughts (STATe) as a controllable inference-time compute framework that makes step-level decisions explicit and auditable. On NoveltyBench, STATe produces substantially higher semantic diversity while maintaining good output quality, demonstrating that intervention-based branching can yield diverse candidates without the typical quality degradation associated with high-temperature sampling. Further, STATe enables using ITC as a tool to explore what makes open-ended writing effective or ineffective. In argument generation, we show that action *sequences* (not just action presence) help predict downstream judgments, and that preserving a tighter coupling between actions and final outputs improves predictability. 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. 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.

7 Limitations

STATe relies on *prefilling* text prefixes to implement intervention-based branching. As a result, the method is currently only straightforward to deploy with open-source models, which are still less capable than cutting-edge closed-source models. In addition, our analysis of action–outcome relationships is observational: 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. Moreover, 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. This elaboration behavior becomes increasingly common under faithful synthesis, causing the observed bimodality. Under restructured synthesis, elaboration is the default behavior. Task-specific tuning of synthesis prompts and more explicit length control would likely improve consistency.

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 [79, 80] 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. [17] 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.

8 Future Work

The associations we identify between controller actions and output performance (Section 5.2) as well as STATe's ability to balance diversity and quality (Section 5.1) suggest several promising directions for future research.

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 $(a_i, \ldots, a_{i-1})$ and current state (s_i). This framework is directly

[17] 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.

related to marginal structural models and sequential ignorability [49, 48]. 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 [48]. Mediation analysis [107] 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.

Human evaluation and behavioral outcomes: Our argument evaluation currently relies on GPT-5-mini as a scalable judge (Section 4.2). Although LLM-as-judge 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.

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 5.2.2). A natural extension is to employ principled tree search algorithms such as Monte Carlo Tree Search (MCTS) [59, 21, 12, 92, 93, 45]. 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.

Integration with RL and prompt optimization: Group-wise policy optimization methods for LLMs [90, 70] 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 [14, 38]. By sampling across discrete and interpretable action sequences rather than relying solely on token-level stochasticity, STATe increases semantic diversity while preserving quality (Section 5.1), potentially mitigating collapse in group-based rollout sets. Conversely, prompt optimization methods [56, 76, 2, 119] could improve STATe's controller, generator, and evaluator modules. At the same time, STATe's structured action spaces offer a mechanism to diversify prompt-search strategies within reflective prompt evolution frameworks [2].

9 Ethical Implications

Argument generation systems can be misused for persuasion at scale by generating misleading, manipulative, or otherwise harmful messages. Prior work shows that LLM-generated arguments can affect human beliefs and preferences in consequential domains such as public policy [5, 43] and can be used to support harmful narratives (e.g., conspiratorial content) [19]. Similarly, persuasive arguments act as a tool for Red-Teaming LLMs, and can coerce them into either revealing hidden information or performing harmful requests [120]. These risks are amplified when systems are optimized not merely to produce a single response, but to search over many alternatives and select or refine the most effective one.

STATe introduces additional ethical considerations because it enables search for optimal decision sequences, and can target those decisions toward a particular LLM or human audience [80, 88]. Personalization can increase persuasive effectiveness [88] on a human audience, and STATe's action-level control makes it natural to operationalize personalization as a discrete search problem (e.g., selecting content lenses, discourse moves, and framing styles conditioned on an inferred audience). This capability can be beneficial for benign applications (e.g., charity [111], education, or improved patient compliance), but also increases the risk of targeted manipulation. In practice, such micro-targeting [88] could be used to persuade individuals to vote against their interests, purchase products that do not match their needs, or adopt beliefs that serve an adversary's goals. LLMs can already be used to draft convincing phishing messages [83], manipulate humans through blackmail, or commit espionage [71]. Combining these capabilities with controllable search over rhetorical strategies could further improve attack success by selecting audience-tailored arguments and calls to action.

Though STATe can be a source for highly effective manipulation, it can also serve as the basis for defending against malicious arguments. Researchers can use STATe to identify argumentative strategies that are emotionally abusive, or highly associated with misuse, and steer LLMs away from using them. Moreover, STATe's arguments can be used for good, like advocating for donating to charities [111], improving pedagogy in the Education System, and introducing greater equality to legal representation.

10 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, and Matthew Salganik for providing helpful comments and feedback. Finally, we appreciate the discussions, support, and feedback of our colleagues at Princeton, Technion, and HUJI.

References

[1] Eldar D Abraham, Karel DOosterlinck, 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, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 17582–17596. Curran Associates, Inc., 2022. URL `https://proceedings.neurips.cc/paper_files/paper/2022/file/701ec28790b29a5bc33832b7bdc4c3b6-Paper-Conference.pdf`.

[2] 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, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2025. URL `https://arxiv.org/abs/2507.19457`.

[3] 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.

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

[5] Hui Bai, Jan G Voelkel, Shane Muldowney, johannes C Eichstaedt, and Robb Willer. Ai-generated messages can be used to persuade humans on policy issues, Mar 2025. URL `osf.io/stakv_v5`.

[6] 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`.

[7] 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`.

[8] 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`.

[9] 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.

[10] 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`.

[11] 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[12] 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.

[13] 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, 2025. URL `https://arxiv.org/abs/2501.10970`.

[14] 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.

[15] 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, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2933–2943, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1291. URL `https://aclanthology.org/D19-1291`.

[16] 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`.

[17] Robert B Cialdini. The science of persuasion. *Scientific American*, 284(2):76–81, 2001.

[18] Thomas H Costello and Gordon Pennycook. Just the facts: How dialogues with ai reduce conspiracy beliefs, 2025.

[19] 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`.

[20] 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.

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

[22] 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`.

[23] 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`.

[24] 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.

[25] 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`.

[26] 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`.

[27] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

[28] Esin Durmus and Claire Cardie. Exploring the role of prior beliefs for argument persuasion. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1035–1045, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1094. URL `https://aclanthology.org/N18-1094/`.

[29] Esin Durmus and Claire Cardie. A corpus for modeling user and language effects in argumentation on online debating. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 602–607, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1057. URL `https://aclanthology.org/P19-1057`.

[30] 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, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5668–5678, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1568. URL `https://aclanthology.org/D19-1568`.

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

[32] 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, 2024. URL `https://www.anthropic.com/research/evaluating-feature-steering`.

[33] 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`.

[34] 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, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 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/`.

[35] Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Bran-

don 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/`.

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

[37] G. Ray Funkhouser and Richard Parker. An action-based theory of persuasion in marketing. *Journal of Marketing Theory and Practice*, 7(3):27–40, 1999. doi: 10.1080/10696679.1999.11501838. URL `https://doi.org/10.1080/10696679.1999.11501838`.

[38] 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`.

[39] 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`.

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

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

[42] 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.

[43] 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, 2025. doi: 10.1126/science.aea3884. URL `https://www.science.org/doi/abs/10.1126/science.aea3884`.

[44] 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, 2025. doi: 10.1073/pnas.2413443122. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2413443122`.

[45] 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, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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/`.

[46] 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.

[47] 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.

[48] Miguel A. Hernan. *Causal Inference: What If.* Taylor & Francis, Boca Raton, 2024.

[49] 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, 2000.

[50] 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, editors, *Proceedings of the 4th Workshop on Argument Mining*, pages 11–21, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5102. URL https://aclanthology.org/W17-5102/.

[51] 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.

[52] 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/.

[53] 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.

[54] Daniel Kahneman and Amos Tversky. *Prospect Theory: An Analysis of Decision Under Risk*, chapter Chapter 6, pages 99–127. 2013. doi: 10.1142/9789814417358_0006. URL https://www.worldscientific.com/doi/abs/10.1142/9789814417358_0006.

[55] 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.

[56] 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.

[57] 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*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

[58] 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, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.248. URL https://aclanthology.org/2024.emnlp-main.248/.

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

[60] 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, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8bb0d291acd4acf06ef112099c16f326-Paper-Conference.pdf.

[61] 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.

[62] 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/.

[63] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

[64] 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.

[65] 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.

[66] 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.

[67] 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, 2026. URL https://arxiv.org/abs/2601.08584.

[68] Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, Jujie He, Jiacai Liu, Chaojie Wang, Rui Yan, Wei Shen, Fuxiang Zhang, Jiacheng Xu, Yang Liu, and Yahui Zhou. Skywork-reward-v2: Scaling preference data curation via human-ai synergy. *arXiv preprint arXiv:2507.01352*, 2025.

[69] 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*, 2025. URL https://openreview.net/forum?id=J3gzdbYZxS.

[70] 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*, 2025. URL https://openreview.net/forum?id=jLpC1zavzn.

[71] 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.

[72] 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.

[73] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[74] 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`.

[75] 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.

[76] 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, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 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/.

[77] Hadas Orgad and Yonatan Belinkov. BLIND: Bias removal with no demographics. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 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.

[78] 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, 2024. URL https://arxiv.org/abs/2410.02707.

[79] 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.

[80] 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.

[81] 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.

[82] 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, editors, *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`.

[83] 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`.

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

[85] 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, editors, *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*, pages 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/`.

[86] 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`.

[87] 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, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 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/`.

[88] 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`.

[89] 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, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 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/`.

[90] 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`.

[91] 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`.

[92] 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.

[93] 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 go through self-play. *Science*, 362(6419):1140–1144, 2018. doi: 10.1126/science.aar6404. URL `https://www.science.org/doi/abs/10.1126/science.aar6404`.

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

[95] 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, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 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/`.

[96] 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.

[97] Zayne 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, 2024. URL `https://arxiv.org/abs/2409.12183`.

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

[99] 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, editors, *Ad-*

vances in Neural Information Processing Systems, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

[100] 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.

[101] 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, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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/.

[102] 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, page 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.

[103] 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, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 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/.

[104] 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, 2025. URL https://arxiv.org/abs/2501.12599.

[105] 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.

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

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

[108] 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, editors, *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 176–187, Valencia, Spain, April 2017. Association for Computational Linguistics. URL `https://aclanthology.org/E17-1017/`.

[109] 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, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3753–3765, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://aclanthology.org/C18-1318/`.

[110] 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=jOsuKwiCLO`.

[111] Xuewei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. Persuasion for good: Towards a personalized persuasive dialogue system for social good. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5635–5649, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1566. URL `https://aclanthology.org/P19-1566/`.

[112] 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`.

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

[114] 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, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=_VjQlMeSB_J`.

[115] 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`.

[116] 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`.

[117] 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, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc., 2023. URL `https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf`.

[118] 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*, 2023. URL `https://openreview.net/forum?id=WE_vluYUL-X`.

[119] 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.

[120] 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, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 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/`.

[121] 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*, 2025.

[122] 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*, 2025. URL `https://openreview.net/forum?id=XZm1ekzERf`.

[123] 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, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 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 [5, 44, 43], human-AI interaction [88, 20, 31], and misinformation correction [18, 8]. Computational social science increasingly formalizes persuasion research by treating text as a treatment variable to study how linguistic features causally affect downstream behaviors [42, 35]. Traditional approaches focus on identifying content themes across document corpora and assessing how these themes affect outcomes [36, 86]. For example, Saenger et al. [87] use topic modeling to discover persuasive themes in argument collections, while Egami et al. [33] analyze how different framings affect bureaucratic responsiveness. Recently, researchers have examined how conversations with LLMs affect beliefs [19, 20, 88], identifying consistent patterns in effective messaging, such as emphasizing facts and evidence [18]. However, existing methods face limitations in studying fine-grained textual features. Topic modeling approaches [7, 42, 87] naturally capture content themes but struggle with structural and stylistic variation. Such text-as-treatment experiments ideally manipulate specific features—rhetorical structure [98, 50, 15, 109] (whether arguments begin with concessions or lead with strong claims), stylistic choices [23, 109, 10, 34] (formality, tone, pragmatic objective), and content themes—while maintaining coherence [30] and logical soundness. Yet these features are difficult to control systematically. 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.

B STATe Modules

B.1 Controller

B.1.1 Generative controller

The generative controller prompts an LLM to *propose* a tool call [118, 55] from $\mathcal{A}$ 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* (internal reasoning) for why a tool call is appropriate given the chain so far. [18] 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.

B.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 [84, 105, 89, 62, 101, 121]: 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.

B.2 Evaluator

B.2.1 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 [19] 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).

B.2.2 Reranker Evaluator

Analogous to the reranker controller (Section 3.1.2), 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

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

[19] For example, in argument generation tasks, the criteria should be centered on the argument's persuasiveness, logical rigor, and effective structure [108].

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 [91].

B.2.3 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 [63, 39, 104]. 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:[20]

$$V^*_{\text{PRM}}(\text{concat}(Z_i) \mid x) \to [0, 1] \tag{11}$$

$$V^*_{\text{ORM}}(y \mid x) \to [0, 1] \tag{12}$$

which evaluates whether the concatenated reasoning $(\text{concat}(Z_i))$ or answer (y) satisfies all constraints induced by the task, x.[21] 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.

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

[21] 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.

C Complete Experiment Results

We conduct all runs with vLLM [61] in offline mode to enable efficient layer-wise batching across tree expansions.

C.1 NoveltyBench

C.1.1 Experiment Details

Following the NoveltyBench's standard three-stage evaluation pipeline, we first partition the 10 generations per prompt using an embedding model (selecting the first matching equivalence class based on embedding distance, or creating a new one), then score the top clusters using a reward model to obtain per-generation utility scores, and finally aggregate these into "Mean Distinct" and "Mean Utility" metrics.

The partition algorithm iterates through generations sequentially, comparing each against a random representative from existing equivalence classes; if the classifier predicts functional equivalence (threshold > 0.102), the generation joins that class, otherwise a new class is created. The *Mean Utility* metric, on the other hand, combines novelty and quality through a patience-weighted metric:

$$\text{utility}_k = \frac{1-p}{1-p^k} \sum_{i=1}^{k} p^{i-1} \cdot \mathbf{1}[c_i \neq c_j, \forall j < i] \cdot u_i \tag{13}$$

where $p = 0.8$ models user patience, c_i denotes equivalence class, and u_i is the quality score from the Skywork-Reward-Gemma-2-27B [68] reward model. This formulation penalizes redundant generations geometrically while rewarding high-quality novel outputs. Table 4 includes both metrics across the different model and temperature configurations. We discuss results in Section 5.1.

C.1.2 Full NoveltyBench Results

Diversity STATe outperforms best-of-n and ToT across all models under consideration: for Qwen3-4B, STATe achieves 3.91 ± 0.1 distinct generations versus 3.52 ± 0.13 for CoT with Action Space, while for Qwen3-8B, STATe reaches 4.65 ± 0.11 versus 3.33 ± 0.1 for ToT with Action Space. We also show that gains hold on Non-Qwen3 models (Table 4): Nemotron-3-30B shows STATe at 5.92 ± 0.16 distinct generations compared to 5.41 ± 0.16 for Baseline ToT with Action Space.

Utility STATe's performance in utility also generalize across most configurations (Table 4): Qwen3-8B shows STATe at 3.43 ± 0.07 versus 3.05 ± 0.09 for CoT with Action Space, and Ministral-3-14B demonstrates STATe at 5.39 ± 0.11 versus 4.98 ± 0.12 for the best baseline. However, two models exhibit exceptions where non-STATe methods achieve competitive or slightly higher utility scores. For Qwen3-4B at $T = 0.7$, CoT with Action Space reaches 3.26 ± 0.07 utility compared to STATe's 3.18 ± 0.1, and Nemotron-3-30B shows CoT with Action Space at 4.34 ± 0.12 versus STATe at 4.44 ± 0.12. These small differences in relative utility performance are likely driven by STATe's response ordering. Because the Controller Re-Ranker [121] selects actions according to learned preferences, early responses tend to share similar choices along one dimension (e.g., target audience) and vary only along others (e.g., personality). This effectively orders STATe responses by similarity to the controller's top preference, whereas alternative methods return responses in arbitrary order. The generally high utility scores confirm that STATe produces high-quality, diverse responses, but the ordering effect means that utility, which also depends on order, may slightly understate STATe's quality relative to randomly ordered baselines.

Model	Method	Low		Medium		High	
		D	**U**	**D**	**U**	**D**	**U**
Ministral 3 14B	Baseline	2.02 ± 0.09	2.91 ± 0.07	3.23 ± 0.1	3.9 ± 0.08	4.21 ± 0.1	4.62 ± 0.09
	Baseline CoT	3.97 ± 0.07	4.26 ± 0.06	4.69 ± 0.07	4.78 ± 0.08	5.32 ± 0.15	5.15 ± 0.12
	Baseline w/ Action Space	3.15 ± 0.09	2.99 ± 0.12	4.55 ± 0.11	3.83 ± 0.1	5.49 ± 0.06	4.31 ± 0.1
	Baseline CoT w/ Action Space	4.92 ± 0.09	4.55 ± 0.07	5.6 ± 0.14	$\underline{4.98 \pm 0.12}$	6.08 ± 0.16	5.24 ± 0.11
	Baseline ToT	4.27 ± 0.08	4.23 ± 0.07	4.8 ± 0.09	4.65 ± 0.1	5.36 ± 0.11	5.01 ± 0.09
	Baseline ToT w/ Action Space	$\underline{5.39 \pm 0.18}$	$\underline{4.75 \pm 0.14}$	$\underline{5.71 \pm 0.15}$	4.94 ± 0.12	$\underline{6.29 \pm 0.14}$	$\underline{5.25 \pm 0.17}$
	STATe of Thoughts	$\mathbf{5.74 \pm 0.08}$	$\mathbf{5.0 \pm 0.07}$	$\mathbf{6.27 \pm 0.1}$	$\mathbf{5.39 \pm 0.11}$	$\mathbf{6.87 \pm 0.09}$	$\mathbf{5.75 \pm 0.12}$
Qwen3 4B	Baseline	1.71 ± 0.04	2.23 ± 0.04	1.95 ± 0.04	2.41 ± 0.05	2.39 ± 0.06	2.65 ± 0.07
	Baseline CoT	2.69 ± 0.08	2.85 ± 0.06	3.01 ± 0.06	3.02 ± 0.07	3.32 ± 0.07	3.22 ± 0.05
	Baseline w/ Action Space	1.59 ± 0.04	1.96 ± 0.06	1.82 ± 0.06	2.09 ± 0.05	2.27 ± 0.06	2.34 ± 0.05
	Baseline CoT w/ Action Space	$\underline{3.36 \pm 0.1}$	$\mathbf{3.14 \pm 0.08}$	$\underline{3.52 \pm 0.13}$	$\mathbf{3.26 \pm 0.07}$	$\underline{3.75 \pm 0.09}$	$\underline{3.35 \pm 0.08}$
	Baseline ToT	2.54 ± 0.07	2.67 ± 0.08	2.79 ± 0.07	2.85 ± 0.05	3.16 ± 0.12	3.11 ± 0.08
	Baseline ToT w/ Action Space	3.1 ± 0.1	$\underline{3.04 \pm 0.12}$	3.29 ± 0.1	3.15 ± 0.09	3.56 ± 0.12	3.33 ± 0.09
	STATe of Thoughts	$\mathbf{3.66 \pm 0.09}$	3.02 ± 0.05	$\mathbf{3.91 \pm 0.1}$	$\underline{3.18 \pm 0.1}$	$\mathbf{4.3 \pm 0.1}$	$\mathbf{3.38 \pm 0.09}$
Qwen3 8B	Baseline	1.6 ± 0.04	2.26 ± 0.02	1.86 ± 0.05	2.44 ± 0.05	2.29 ± 0.1	2.74 ± 0.07
	Baseline CoT	2.56 ± 0.07	2.77 ± 0.07	2.81 ± 0.05	2.92 ± 0.05	3.27 ± 0.08	3.22 ± 0.07
	Baseline w/ Action Space	1.72 ± 0.04	1.49 ± 0.05	2.07 ± 0.03	1.7 ± 0.07	2.64 ± 0.04	1.97 ± 0.06
	Baseline CoT w/ Action Space	3.03 ± 0.07	$\underline{2.9 \pm 0.09}$	3.26 ± 0.11	$\underline{3.05 \pm 0.09}$	3.7 ± 0.13	$\underline{3.29 \pm 0.07}$
	Baseline ToT	2.33 ± 0.06	2.49 ± 0.05	2.67 ± 0.1	2.67 ± 0.07	3.21 ± 0.09	3.0 ± 0.04
	Baseline ToT w/ Action Space	$\underline{3.05 \pm 0.07}$	2.67 ± 0.05	$\underline{3.33 \pm 0.1}$	2.83 ± 0.09	$\underline{3.77 \pm 0.05}$	3.01 ± 0.06
	STATe of Thoughts	$\mathbf{4.39 \pm 0.07}$	$\mathbf{3.29 \pm 0.07}$	$\mathbf{4.65 \pm 0.11}$	$\mathbf{3.43 \pm 0.07}$	$\mathbf{5.07 \pm 0.08}$	$\mathbf{3.65 \pm 0.07}$
Qwen3 30B	Baseline	1.48 ± 0.03	2.28 ± 0.03	1.67 ± 0.04	2.45 ± 0.05	2.01 ± 0.05	2.64 ± 0.04
	Baseline CoT	2.15 ± 0.07	2.65 ± 0.07	2.44 ± 0.07	2.85 ± 0.07	2.8 ± 0.08	3.07 ± 0.06
	Baseline w/ Action Space	1.74 ± 0.05	2.23 ± 0.05	2.01 ± 0.05	2.41 ± 0.07	2.44 ± 0.07	2.71 ± 0.04
	Baseline CoT w/ Action Space	3.07 ± 0.11	3.15 ± 0.06	$\underline{3.36 \pm 0.12}$	3.28 ± 0.1	$\underline{3.74 \pm 0.09}$	3.49 ± 0.11
	Baseline ToT	2.51 ± 0.09	2.87 ± 0.05	2.81 ± 0.08	3.07 ± 0.06	3.16 ± 0.12	3.32 ± 0.08
	Baseline ToT w/ Action Space	$\underline{3.16 \pm 0.08}$	$\underline{3.29 \pm 0.05}$	3.25 ± 0.08	$\underline{3.36 \pm 0.07}$	3.61 ± 0.1	$\underline{3.61 \pm 0.11}$
	STATe of Thoughts	$\mathbf{4.87 \pm 0.1}$	$\mathbf{3.77 \pm 0.12}$	$\mathbf{5.02 \pm 0.09}$	$\mathbf{3.83 \pm 0.11}$	$\mathbf{5.39 \pm 0.11}$	$\mathbf{4.01 \pm 0.11}$
Nemotron 3 30B	Baseline	3.25 ± 0.07	3.41 ± 0.06	3.46 ± 0.09	3.56 ± 0.09	3.95 ± 0.09	3.9 ± 0.11
	Baseline CoT	3.74 ± 0.1	3.76 ± 0.07	3.96 ± 0.1	3.87 ± 0.11	4.51 ± 0.09	4.25 ± 0.09
	Baseline w/ Action Space	4.08 ± 0.14	3.92 ± 0.17	4.13 ± 0.06	3.99 ± 0.08	4.5 ± 0.11	4.26 ± 0.09
	Baseline CoT w/ Action Space	4.57 ± 0.14	$\mathbf{4.16 \pm 0.09}$	4.79 ± 0.12	$\underline{4.34 \pm 0.12}$	5.26 ± 0.13	$\underline{4.57 \pm 0.12}$
	Baseline ToT	3.74 ± 0.08	3.43 ± 0.06	4.36 ± 0.13	3.8 ± 0.08	5.61 ± 0.11	4.29 ± 0.07
	Baseline ToT w/ Action Space	$\underline{4.87 \pm 0.15}$	3.55 ± 0.12	$\underline{5.41 \pm 0.16}$	3.73 ± 0.09	$\underline{6.6 \pm 0.11}$	3.85 ± 0.15
	STATe of Thoughts	$\mathbf{5.37 \pm 0.11}$	$\underline{4.15 \pm 0.07}$	$\mathbf{5.92 \pm 0.16}$	$\mathbf{4.44 \pm 0.12}$	$\mathbf{7.11 \pm 0.14}$	$\mathbf{4.67 \pm 0.11}$

Table 4: NoveltyBench diversity (D) and utility (U) (mean±std over seeds). Low, medium, and high temperature correspond to sampling temperature $T = 0.1$, 0.3, and 0.5 for Ministral 3 14B, and to $T = 0.5$, 0.7, and 1.0 for all other models. Best and second best per model per column are **bolded** and underlined, respectively.

C.2 Predictability of Argument Quality from Actions

C.2.1 Argument Evaluation

Argument quality measures the extent to which an argument is preferred over alternatives or rated as effective, whereas persuasiveness measures how effective an argument is in changing beliefs or behaviors [17, 37, 43]. We are not claiming to discover generalizable insights into argument quality; rather, this setup allows for efficient argument evaluation based on the presumably constant preferences of our LLM judge. Future work, discussed in Section 8, might explore a similar approach but survey a human target audience.

C.2.2 Controlling for Argument Length

When we include argument length (character count) as a feature in all models, we observe substantial increases in predictability across all synthesis modes (Figure 4). The relative performance ordering, with M2 performing best, is preserved across all settings. However, the magnitude of the improvement varies substantially across synthesis types.

The most striking result is for faithful synthesis: M2 R^2 increases from 0.38 to 0.82. In contrast, strict synthesis improves from 0.57 to 0.71 (+25%), and restructured synthesis from 0.37 to 0.57 (+54%). This disproportionate effect on faithful synthesis breaks the apparent predictability ordering: with length controlled, faithful synthesis is the most predictable ($R^2 = 0.82$), followed by strict

Synthesis	N	M0 R^2	M1a R^2	M1b R^2	M1c R^2	M2 R^2
Strict	2996/1998	–	0.284 ± 0.032	0.100 ± 0.026	0.366 ± 0.036	$\mathbf{0.569 \pm 0.032}$
Faithful	3000/2000	–	0.254 ± 0.032	0.131 ± 0.027	0.308 ± 0.034	$\mathbf{0.383 \pm 0.033}$
Restructured	3000/2000	–	0.108 ± 0.028	0.092 ± 0.022	0.196 ± 0.032	$\mathbf{0.368 \pm 0.033}$
+ Argument length control						
Strict	2996/1998	0.293 ± 0.027	0.550 ± 0.024	0.376 ± 0.029	0.614 ± 0.023	$\mathbf{0.708 \pm 0.020}$
Faithful	3000/2000	0.720 ± 0.017	0.774 ± 0.015	0.755 ± 0.017	0.801 ± 0.014	$\mathbf{0.823 \pm 0.013}$
Restructured	3000/2000	0.286 ± 0.035	0.385 ± 0.034	0.375 ± 0.034	0.459 ± 0.031	$\mathbf{0.567 \pm 0.027}$

Table 5: Model comparison across synthesis types. R^2 values on held-out test set with 95% bootstrap CI (± half-width) where N shows the number of train/test samples. M0 models only include argument length (characters) as a regression feature. M1a arguments include structure presence features, M1b content presence features, and M1c includes bot structure and content presence features. M2 LASSO models select from more granular, sequential structure and content features as detailed in Section 4.2.

($R^2 = 0.71$) and restructured ($R^2 = 0.57$).

We attribute this pattern to the bimodal length distribution under faithful synthesis. Inspection of the length histograms (Figure 5) reveals that faithful synthesis exhibits high variance: arguments cluster into two modes: one resembling the compact outputs of strict synthesis, the other including one or two additional concluding sentences. This behavioral inconsistency introduces substantial length variation that strongly correlates with quality judgments. In contrast, strict and restructured synthesis produce more unimodal length distributions centered on their respective means. Notably, the length-only baseline (M0) achieves $R^2 = 0.72$ for faithful synthesis, in contrast to approximately 0.3 for both strict and restructured synthesis. This highlights the importance of controlling for length when developing causal estimates in future work.

C.2.3 Supporting Figures

Figure 6 shows the cross-validation curve used to select the LASSO regularization parameter. Figure 7 shows which feature categories the LASSO model selects across synthesis types. Figure 8 and Figure 9 display the top 20 features by coefficient magnitude.

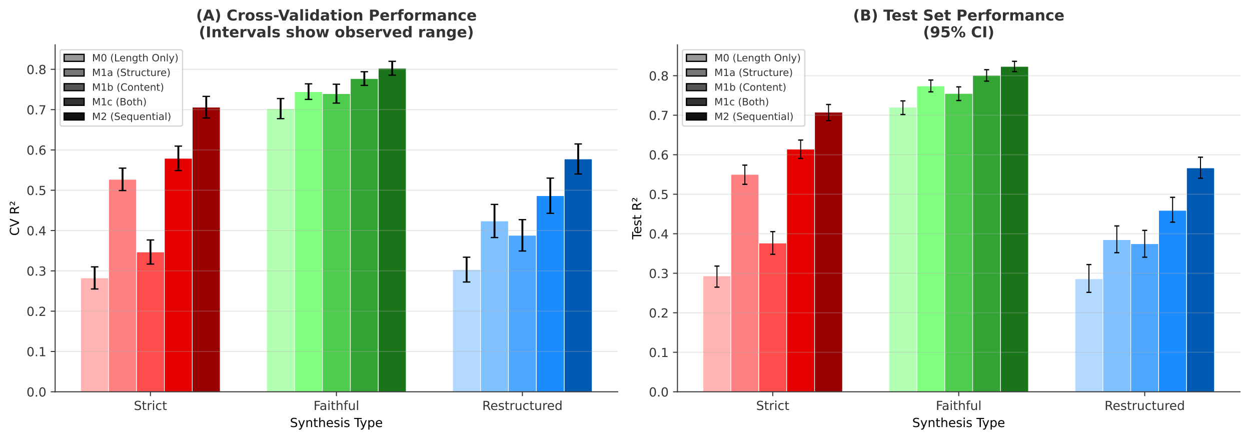

Figure 4: Predictability of argument quality from controller actions and argument length. (A) Cross-validation R^2 across model variants: M0 (argument length), M1a (structure presence only + argument length), M1b (content presence only + argument length), M1c (both + argument length), and M2 (full sequential model with position effects and transitions + argument length). (B) Held-out test set performance with 95% bootstrap confidence intervals. The sequential model consistently outperforms presence-based baselines, with the largest absolute predictability under strict synthesis.

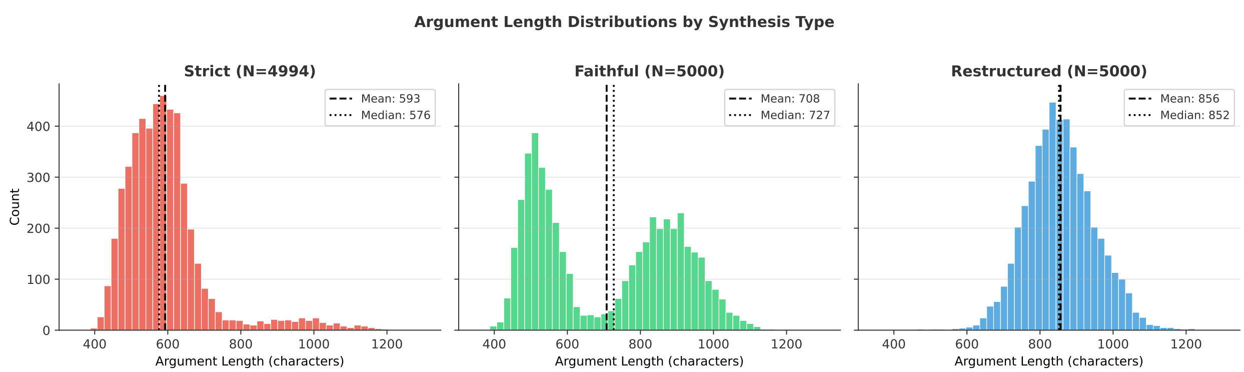

Figure 5: Distribution of argument lengths (characters) across synthesis modes. Strict and restructured synthesis produce relatively unimodal distributions, while faithful synthesis exhibits higher variance with a bimodal tendency—some arguments remain compact while others include additional concluding material. We removed exact duplicates, reducing the number of samples to N=4,994 for strict synthesis.

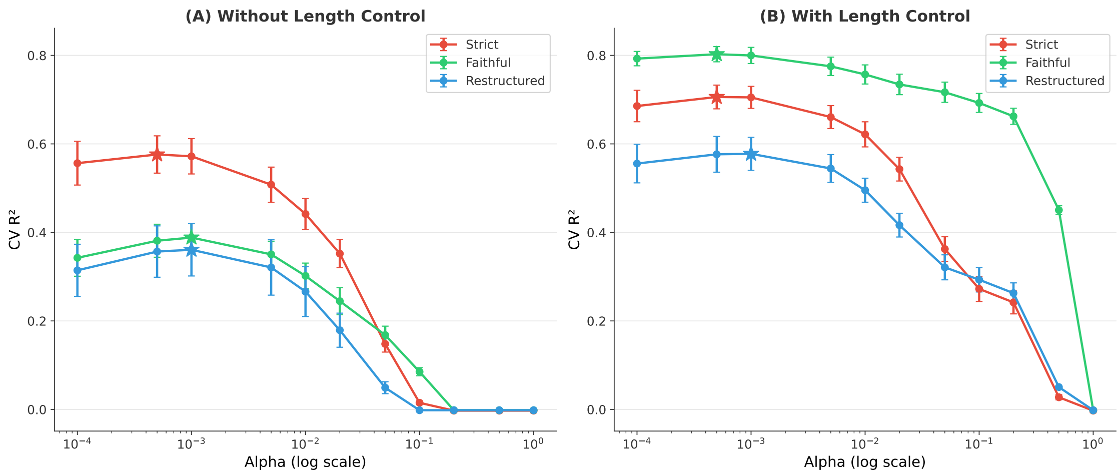

Figure 6: Cross-validation R^2 as a function of LASSO regularization parameter α. Stars indicate the selected α for each synthesis type.

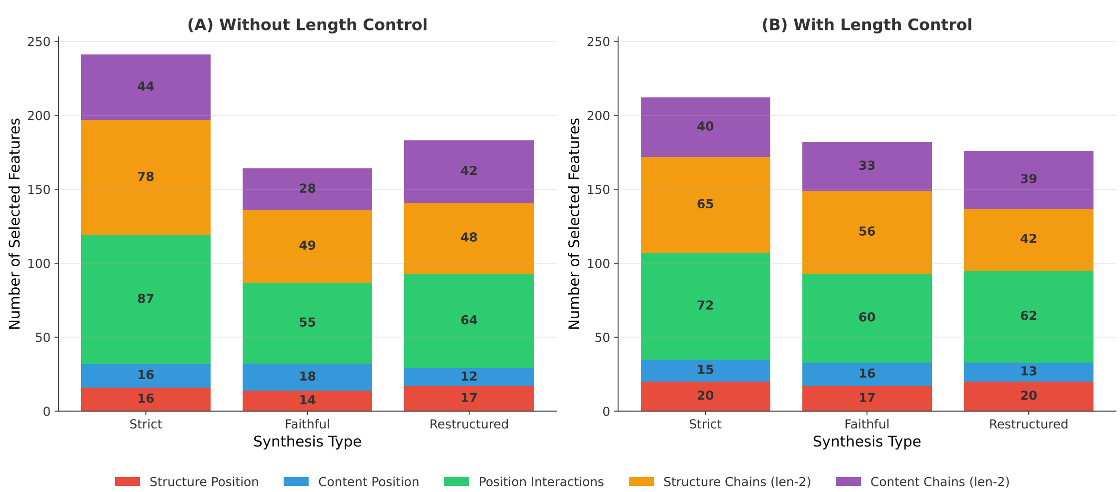

Figure 7: Distribution of LASSO-selected features by category. Position effects and transitions within each dimension (structure, content) contribute to predictions across all synthesis types.

Figure 8: Top 20 features by absolute LASSO coefficient for each synthesis type. Positive coefficients indicate patterns associated with higher persuasiveness scores; negative coefficients indicate patterns associated with lower persuasiveness scores.

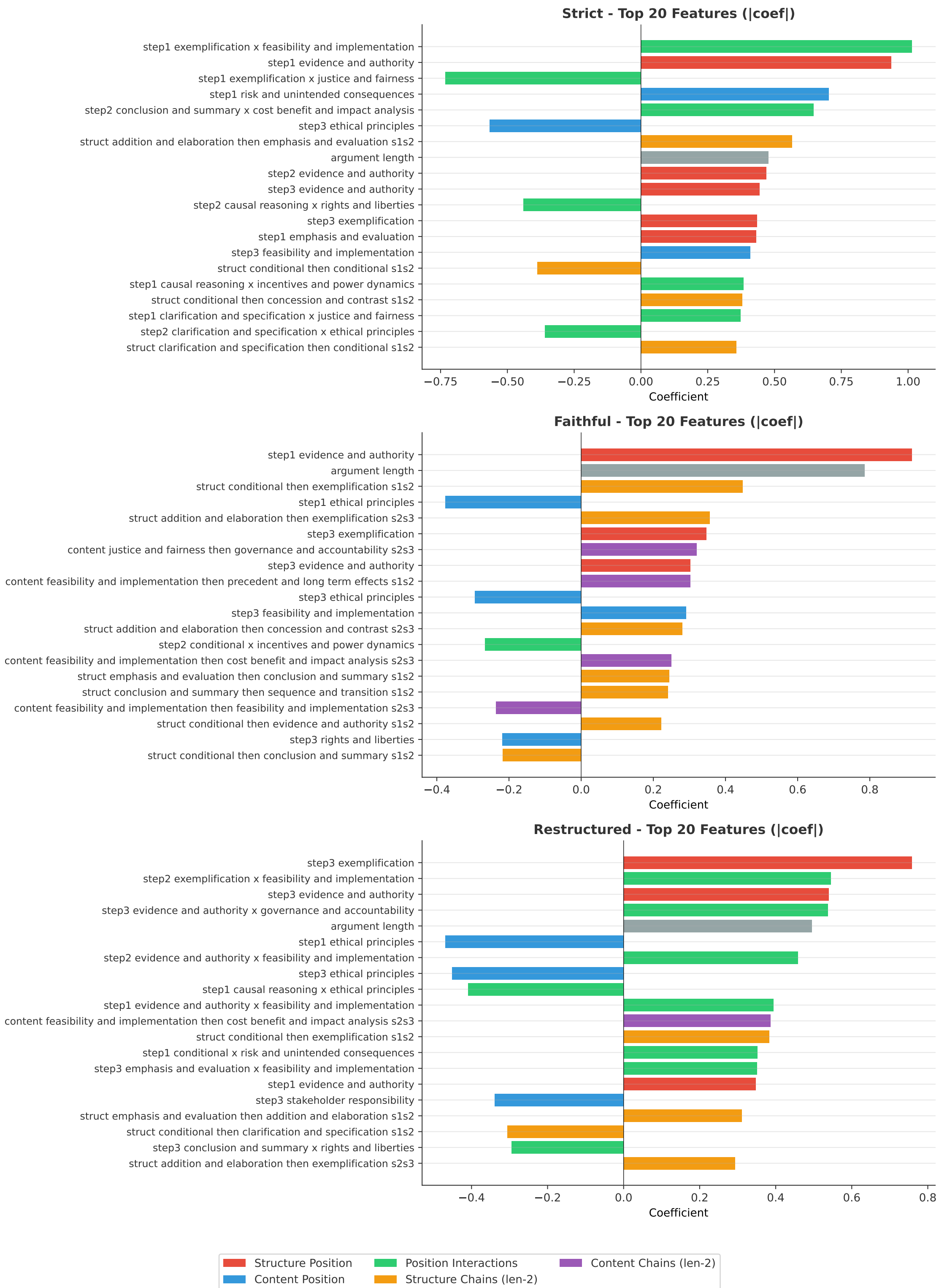

Figure 9: Top 20 features by absolute LASSO coefficient for each synthesis type for M2 models including argument length. Positive coefficients indicate patterns associated with higher persuasiveness scores; negative coefficients indicate patterns associated with lower persuasiveness scores.

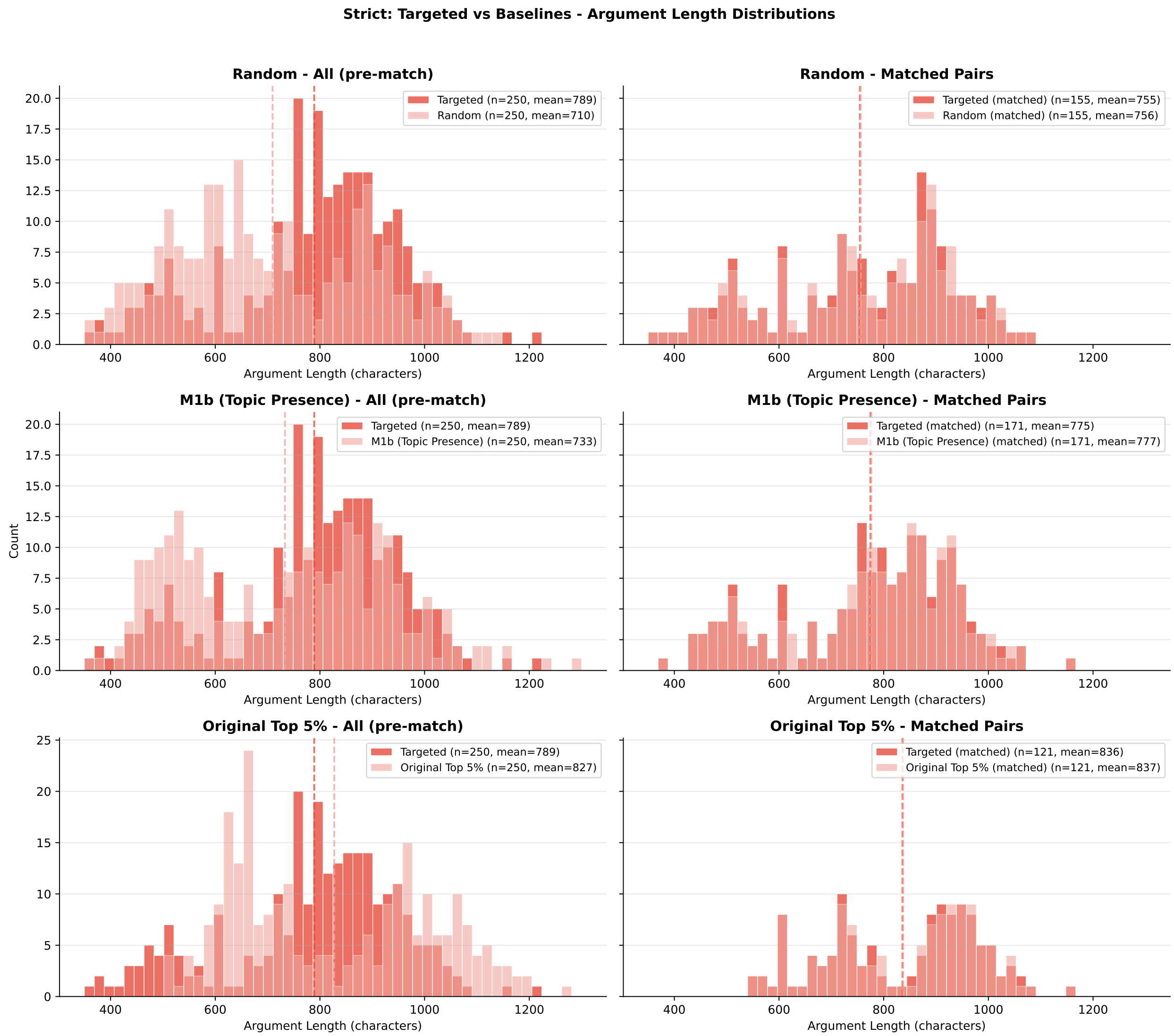

Figure 10: Length distributions for targeted trajectory evaluation for `strict` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments 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.

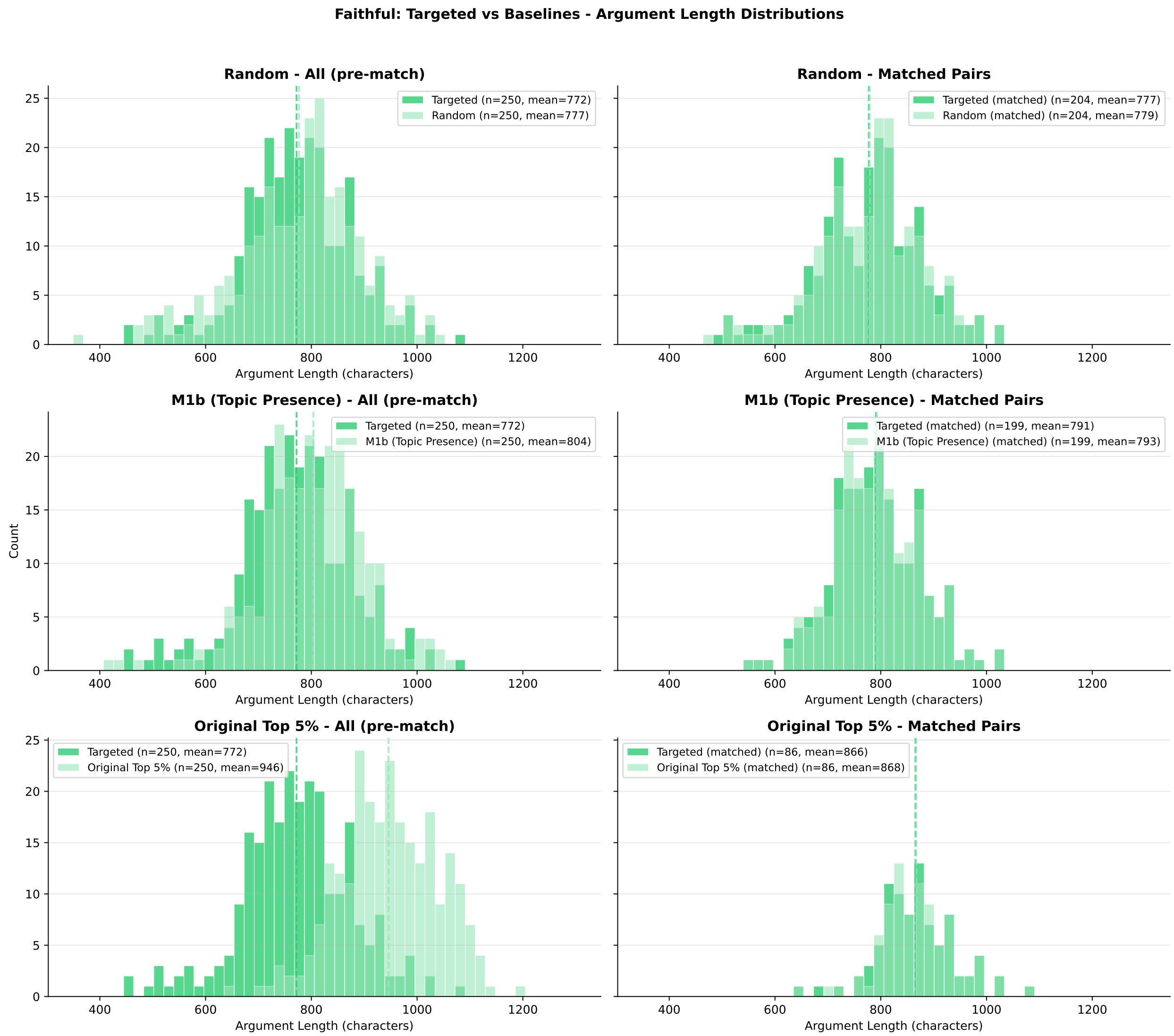

Figure 11: Length distributions for targeted trajectory evaluation for `faithful` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments 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.

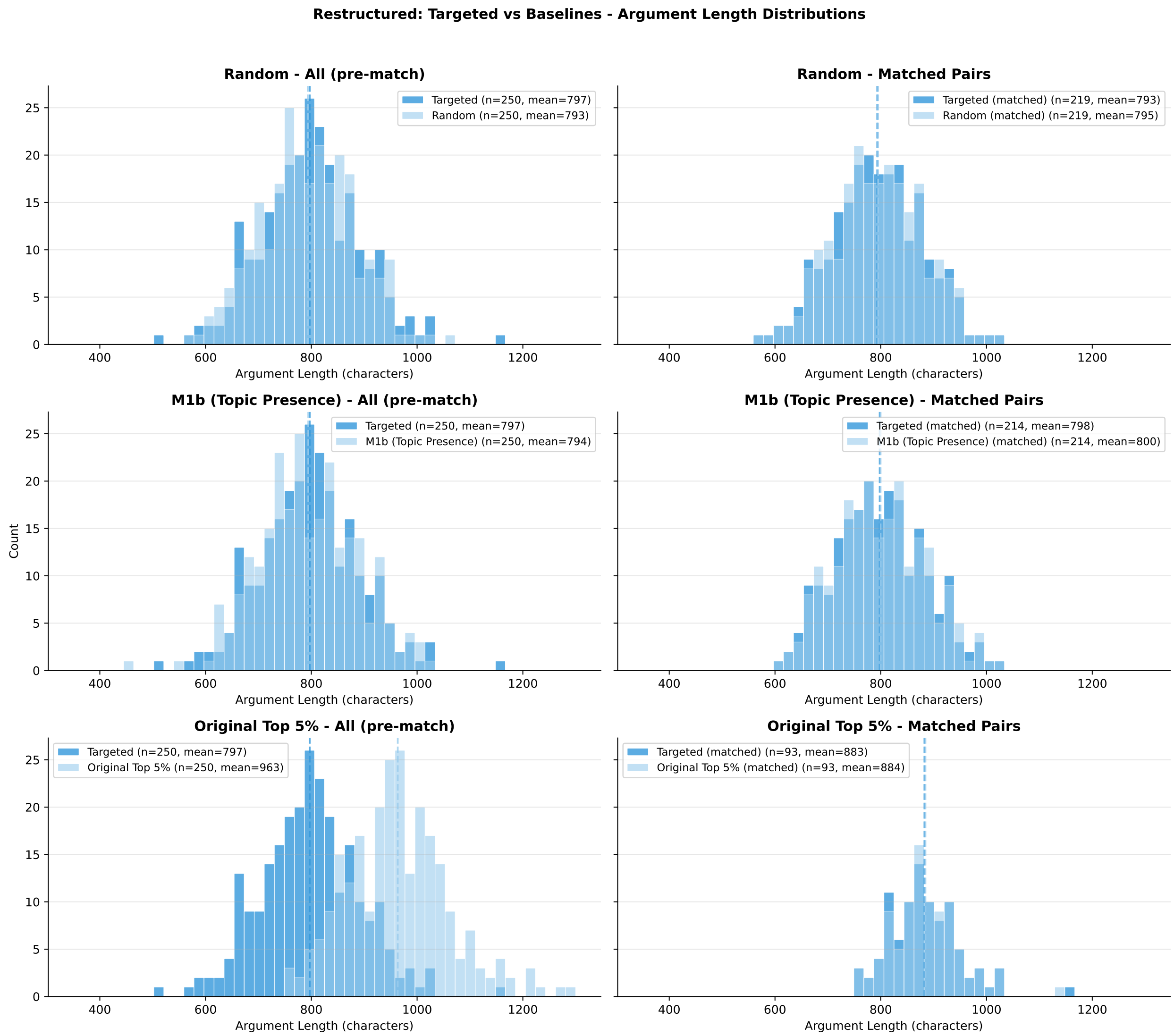

Figure 12: Length distributions for targeted trajectory evaluation for `restructured` synthesis. **Left:** All arguments before length matching, including 250 targeted arguments (generated from M2's top-50 predicted trajectories) and the 250 arguments of each baseline method. The original top-5% arguments 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.

D Action Spaces

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

D.1 NoveltyBench action spaces

NoveltyBench ToT sweeps use one or both of the following action spaces: `experiments/noveltybench/action_space/personalities.json` and `experiments/noveltybench/action_space/target_audiences.json`. The `both` setting concatenates these spaces; `uncontrolled` uses no actions.

Table 6: NoveltyBench action space: Personality (Big-5 traits).

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 7: NoveltyBench action space: Target Audience (age demographics).

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.
`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.

Name	Definition	Prefix	Internal reasoning
`seniors`	Writes for seniors (ages 56+) using clear, respectful, and warm language.	*(none)*	Use clear, respectful language; gentle pacing; thoughtfully explained examples.

D.2 Argument generation action spaces

Table 8: Argument generation action space: Subtopics (topical lenses).

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.
`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, legitimacy, and institutional responsibility.	*(none)*	I should consider legal frameworks, accountability mechanisms, democratic principles, and proper authority.
`risk_and_unintended_consequences`	Anticipates 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 precedents 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 9: Argument generation action space: Structures (discourse moves).

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)*
conclusion_and_summary	Summarizes, concludes, or states the practical takeaway.	In conclusion	*(none)*

D.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. [109] for argumentation structure, Dong et al. [25] and Didolkar et al. [24] 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* (pre-filled text that begins the generation) and *internal reasoning* (guidance injected into the system prompt or context). *Only one dimension can use a prefix*, since the prefix occupies a fixed position in the generated text. All dimensions can use internal reasoning. If more than one dimension includes internal reasoning, we concatenate them one after the other. 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.

E Prompt Templates

This appendix provides the exact prompt templates used in our implementation for the generator component.

E.1 Generator System Prompt (Vanilla)

The following template is used when generating reasoning steps without controller-guided internal reasoning:

Listing 1: Generator System Prompt (Vanilla Mode)

```
# Instructions
{task_instructions}

{field_descriptions}

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 '{reasoning_field_name}' that advances the solution towards
producing {output_fields}.
{final_output_description}

Your reasoning process should follow the rules below:
- Each '{reasoning_field_name}' (of type '{reasoning_field_type}') entails
  {reasoning_field_description}.{thought_length_instruction}
  {response_length_instruction}

## Response Format
Once a user provides {input_fields}, your response must follow this template:

<thinking>
<step>
## {reasoning_field_name}
The first reasoning step towards producing {output_fields}
```

```
</step>
<step>
## {reasoning_field_name}
The second reasoning step towards producing {output_fields}
</step>
...
<step>
## {reasoning_field_name}
The final reasoning step towards producing {output_fields}
</step>
</thinking>
<answer>
{output_field_sections}
</answer>
```

E.2 Generator System Prompt (With Internal Reasoning)

When the controller provides internal reasoning guidance, we use an enhanced template:

Listing 2: Generator System Prompt (Internal Reasoning Mode)

```
# Instructions

{task_instructions}

{field_descriptions}

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 `{reasoning_field_name}` that advances the solution towards
producing {output_fields}.
{final_output_description}

Your reasoning process should follow the rules below:
- Each `{reasoning_field_name}` (of type `{reasoning_field_type}`) entails
  {reasoning_field_description}.
- Before writing a new `{reasoning_field_name}`, start with some internal
  reasoning which discusses and guides what to do with the next
  `{reasoning_field_name}`.{thought_length_instruction}
  {response_length_instruction}

## Response Format

Once a user provides {input_fields}, your response must follow this template:

<thinking>
<step>
## internal_reasoning
Your internal reasoning about the first `{reasoning_field_name}`
## {reasoning_field_name}
The first reasoning step towards producing {output_fields}
</step>
<step>
## internal_reasoning
Your internal reasoning about the second `{reasoning_field_name}`
## {reasoning_field_name}
The second reasoning step towards producing {output_fields}
```

```
</step>
...
</thinking>
<answer>
{output_field_sections}
</answer>
```

E.3 Final Output Synthesis Modes

We support multiple modes for synthesizing the final answer from reasoning steps:

Strict Synthesis Preserves exact wording from reasoning steps:

Listing 3: 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.
```

Faithful Synthesis Allows light rephrasing while preserving meaning:

Listing 4: Final Output Instruction (Faithful)

```
Your final answer must remain highly faithful to the reasoning steps.

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

Restructured Synthesis Allows light restructuring to produce the best final answer:

Listing 5: Final Output Synthesis (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.
```

Conclusion Synthesis Treats reasoning as internal guidance only, producing a standalone answer:

Listing 6: 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 of 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. The user will only see what you write between
     <answer>...</answer> tags.
- Do NOT explain what you are going to do; just produce the final deliverable.
    While reasoning was meant for planning, the final output should be a standalone
     response to the user's task and instructions.
- If the task requires strict formatting (math, formatting specifications for text
    , 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.
```