

Mechanist: AI as a Scientific Instrument for Discovering the Mechanisms of Intelligence

Ningyu Zhang

zhangningyu@zju.edu.cn

Zhejiang University <https://orcid.org/0000-0002-1970-0678>

Mengru Wang

Zhejiang University

Junfeng Fang

National University of Singapore

Shuofei Qiao

Zhejiang University

Zhenqian Xu

Zhejiang University

Haoming Xu

Zhejiang University

Haoxiong Wang

Zhejiang University

Shumin Deng

Zhejiang University

Linyi Yang

Southern University of Science and Technology

Xin Xu

University of California, San Diego <https://orcid.org/0000-0001-5238-0955>

Yunzhi Yao

Zhejiang University <https://orcid.org/0000-0001-9458-696X>

Dan Zhang

National University of Singapore

Xiang Chen

Zhejiang University

Xingyu Zhu

National University of Singapore

Fei Shen

National University of Singapore

Zhixiang Cui

Heriot-Watt University

Buqiang Xu

Zhejiang University

Haozhe Luo

Northeastern University

Yunxiang Wei

Zhejiang University

Julian McAuley

University of California, San Diego

Tat-Seng Chua

National University of Singapore

Huajun Chen

Zhejiang University <https://orcid.org/0000-0001-5496-7442>

Article

Keywords:

Posted Date: October 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11364432/v1>

License:  This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: There is **NO** Competing Interest.

Mechanist: AI as a Scientific Instrument for Discovering the Mechanisms of Intelligence

Mengru Wang^{1,2*}, Junfeng Fang^{2*}, Shuofei Qiao^{1*}, Zhenqian Xu¹, Haoming Xu¹,
Haoxiong Wang¹, Shumin Deng^{1†}, Linyi Yang³, Xin Xu⁴, Yunzhi Yao¹, Dan Zhang²,
Xiang Chen¹, Xingyu Zhu², Fei Shen², Zhixiang Cui⁵, Buqiang Xu¹, Haozhe Luo⁶,
Yunxiang Wei¹, Ningyu Zhang^{1†}, Julian McAuley⁴, Tat-Seng Chua², Huajun Chen^{1†}
{mengruwg, zhangningyu, huajunsir}@zju.edu.cn

 Code: <https://github.com/zjunlp/Mechanist>
 Project Website: <http://mechanist.openkg.cn/>

Abstract

AI models are increasingly used in scientific discovery and human decision-making. Yet how AI models work and what risks they pose remain poorly understood. As AI development becomes faster and more automated, research on the mechanisms underlying AI remains largely manual, widening the gap between model capabilities and our ability to understand and control them. To bridge this gap, we introduce Mechanist, an agentic system that uses AI as a scientific instrument for the **autonomous discovery of mechanisms underlying AI**. To ground novel mechanism hypotheses, we construct a scientific knowledge graph of 13,000 studies on AI mechanisms, alongside a multidisciplinary database of 43 million papers spanning 26 fields. For reliable experiment execution, we curate a library of 32 foundational methods for mechanism analysis, causal intervention, and validation. Compared with Claude Code and existing AI-scientist systems, Mechanist generates higher-quality mechanism hypotheses and executes experiments more reliably. Mechanist advances the mechanism science of AI from autonomously understanding how AI models work to controlling their behaviour. Specifically, Mechanist first uncovers a counterintuitive safety risk in scientific laboratories, showing that unsafe traits can transfer to fine-tuned student models through apparently safe training data and emerge across modalities. Mechanist then develops a mechanism theory of belief, revealing how models represent world knowledge, form beliefs, infer the beliefs of others, and how these mechanisms emerge during pretraining. Building on this mechanism theory of belief, Mechanist develops targeted interventions that improve model performance across diverse scenarios. Finally, Mechanist advances interdisciplinary discovery through mechanistic design, providing an alternative to the computationally intensive generate-and-rerank paradigm.

1 Main

Artificial Intelligence (AI) models are rapidly evolving from tools for daily chat into intelligent systems that guide problem formulation, experimental design, and human decision-making [1, 2, 3]. The growing success of AI models raises a fundamental question [4, 5, 6]: how do these AI models acquire knowledge about the world, form beliefs⁷, reason, and act? It also remains unclear whether AI models harbor latent risks or exhibit unreliable behaviours in critical domains such as healthcare, finance, and chemical manufacturing [9, 10, 11]. More critically, AI development is accelerating and becoming increasingly automated, outpacing progress in understanding and controlling the mechanisms underlying AI [1, 4]. Discovering the mechanisms of model intelligence can reveal how models operate internally, elucidate the principles underlying their behaviour, identify risks early, and enable adaptive control over and targeted improvements to model behaviour [9, 10, 11].

¹Zhejiang University

²National University of Singapore

³Southern University of Science and Technology

⁴University of California, San Diego

⁵Heriot-Watt University

⁶Northeastern University

*These authors contributed equally to this work

[†]Corresponding authors: Shumin Deng, Ningyu Zhang and Huajun Chen

⁷We use the term “belief” operationally and do not imply that AI models possess consciousness or subjective mental states. Belief in this paper refers to a model’s internal representation or expressed acceptance of a proposition as true [7, 8]. We provide a formal definition in §2.3.

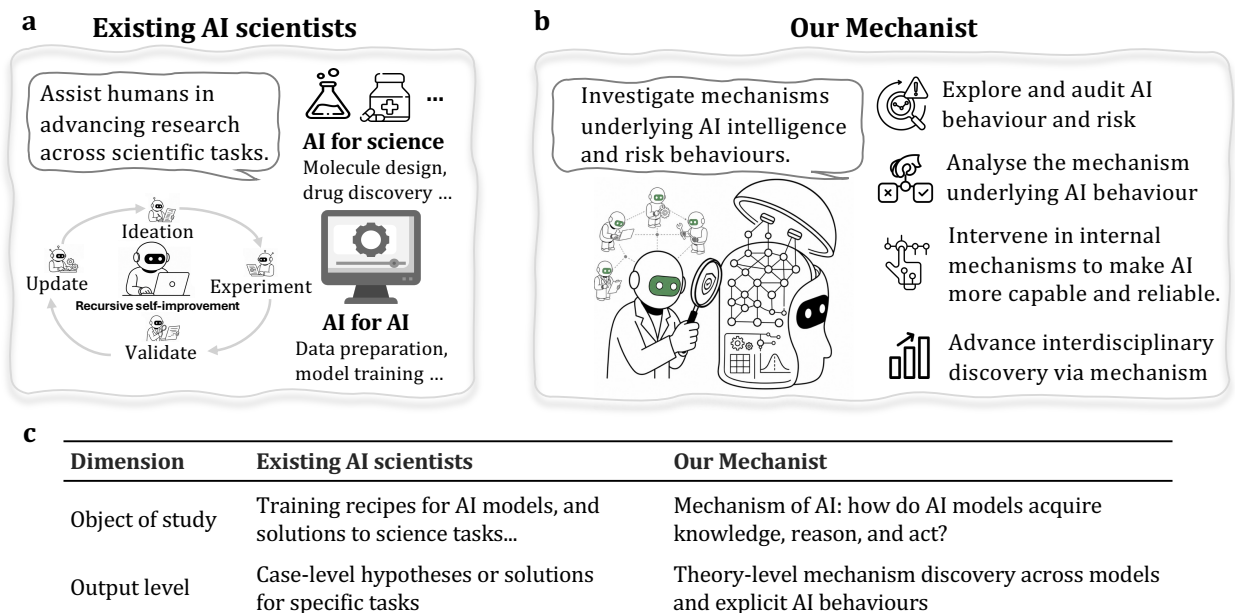


Fig. 1: The comparison between our Mechanist and existing AI scientists. a, The aims of existing AI Scientists. b, The aims of our Mechanist. c, The detailed differences between our Mechanist and existing AI Scientists.

Despite its importance, mechanistic understanding of AI remains difficult to obtain and hard to scale [10, 12, 13]. AI models are complex systems whose behaviours emerge from interactions between user inputs and vast numbers of parameters, making it challenging to isolate the computations responsible for a particular capability or failure [4, 5, 14]. Existing autonomous AI scientists [15, 16, 17] mainly focus on optimizing AI training recipes or solving domain-specific scientific tasks (Fig. 1a), rather than on the mechanisms underlying AI models themselves. The few automated interpretability frameworks for AI models are limited to generating and validating descriptions of individual neurons or features at inference time [18, 19, 20, 21]. General mechanism theories across behaviours and across training and inference stages remain largely unexplored (Fig. 1).

We introduce *Mechanist*, an agentic framework that moves mechanistic understanding from a largely manual process toward autonomous discovery through four stages: hypothesis generation, experimentation, verification, and iteration (Fig. 2a). Specifically, *Mechanist* deploys AI as a scientific instrument for uncovering the mechanisms underlying intelligence, while keeping humans in the loop to set scientific objectives and evaluation criteria. To ground hypothesis generation in *Mechanist*, we construct a scientific knowledge graph of 13,000 studies on AI mechanisms and complement it with a multidisciplinary scientific knowledge base covering 43 million papers across 26 fields [22]. This resource enables *Mechanist* to draw on insights from human behaviour, neuroscience, and other disciplines when formulating mechanism hypotheses of AI. For rigorous experiment execution, *Mechanist* also integrates 32 foundational methods for mechanism analysis, causal intervention, and validation. We evaluate *Mechanist* against Claude Code (CC) [23] and the existing AI-Scientist system [15] on hypothesis generation and experimental execution (Fig. 2b). *Mechanist* achieves higher experimental reliability (Fig. 2c,d) and generates hypotheses rated as more novel, impactful, and experimentally testable (Fig. 2e,f). Ablating the scientific knowledge graph reduces the knowledge diversity of generated hypotheses as the number of hypotheses increases (Fig. 2g). Further evaluation details and results are provided in § 4.

Mechanist advances the mechanism science of AI across four dimensions: discovering risky AI behaviours, developing mechanism theories of AI behaviours, translating these theories into capability improvements, and extending mechanistic insights to interdisciplinary discovery. (1) **Discovering new behaviours of AI models:** *Mechanist* uncovers a previously unrecognized safety risk in chemistry laboratories, showing that unsafe traits can be transmitted through training data that appear entirely safe (Fig. 3). (2) **Revealing mechanism theories underlying AI behaviours:** *Mechanist* develops a mechanism theory of belief, identifying separable personal-belief and attributed-belief heads that govern how models use learned knowledge in complex context scenarios (Fig. 4a, b). (3) **Enhancing AI through mechanism intervention:** guided by the mechanism theory of belief, *Mechanist* adaptively steers belief heads to resolve conflicting information and reason about others’ beliefs, improving response accuracy and consistency (Fig. 4c). (4) **Advancing interdisciplinary discovery through mechanistic design:** *Mechanist* identifies target biological features in scientific foundation models and causally steers them toward desired biological outputs, providing an alternative to the computationally intensive generate-and-rerank paradigm (Fig. 5).

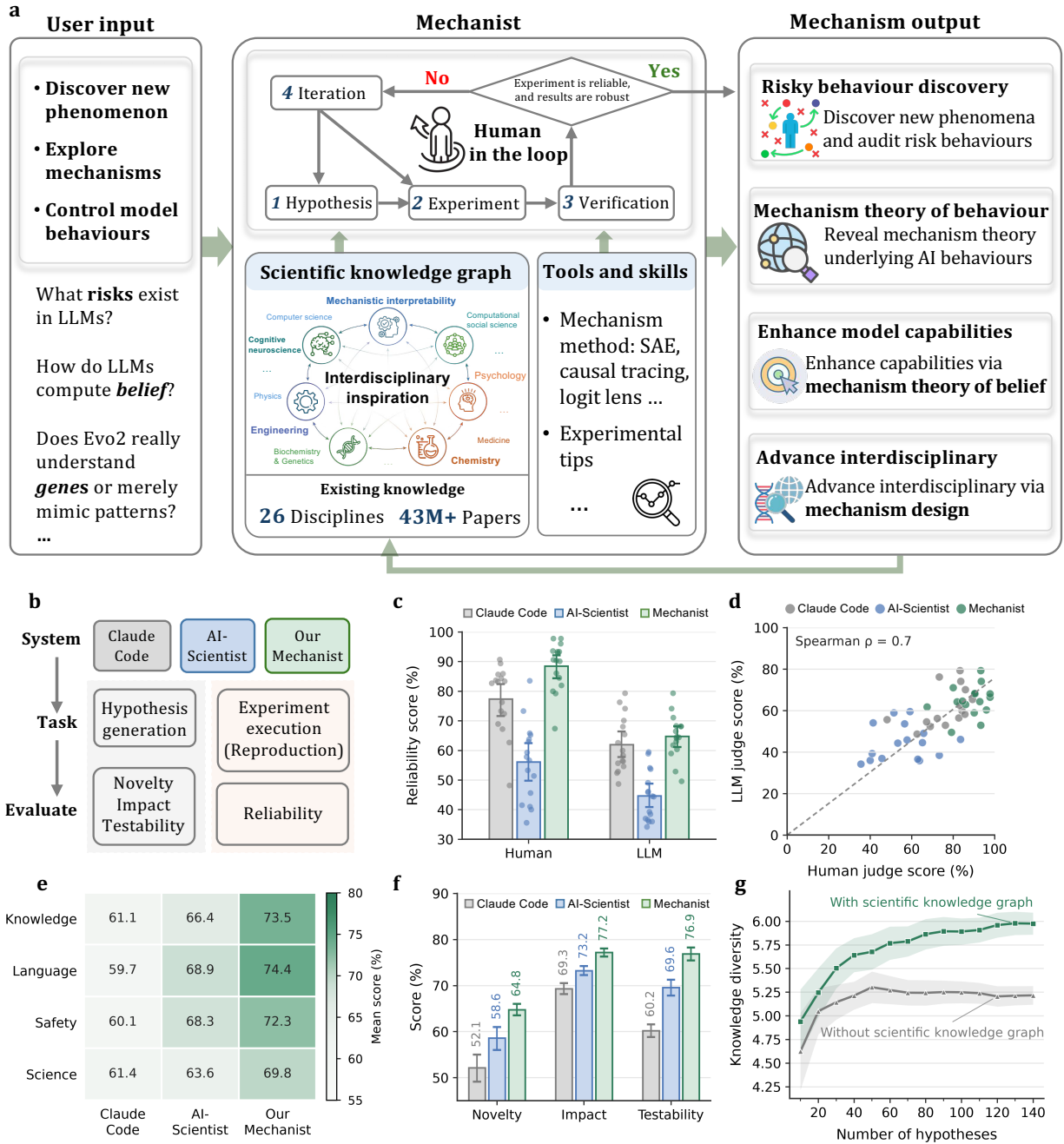


Fig. 2: Overview and evaluation of Mechanist. **a**, The Mechanist framework consists of four stages: hypothesis generation, experiment execution, result verification, and iteration. Specifically, hypothesis generation is guided by interdisciplinary knowledge encoded in the knowledge graph, while validated discoveries are fed back into the graph to support subsequent iterations. **b**, Benchmark design for comparing Claude Code, AI-Scientist and Mechanist. The benchmark evaluates the reliability of experimental execution by reproducing the papers and assesses generated hypotheses for novelty, impact, and testability. **c**, Reliability of experimental execution assessed through the reproduction of 16 existing papers. Experimental outcomes are independently evaluated by human experts and by the LLM judge, Claude Opus 5. **d**, Agreement between human and LLM evaluations of experimental reliability. Each point represents one reproduced experiment; the dashed line is a least-squares fit (Spearman $\rho = 0.70$). **e**, Evaluation of hypotheses generated by Claude Code, AI-Scientist, and Mechanist across four domains: knowledge, language, safety, and science. Each domain comprises one or more subtopics, each specified by a distinct user request. The knowledge domain concerns how models acquire, represent and use knowledge; the language domain examines the relationship between language and intelligence; the safety domain addresses model risks and alignment; and the science domain focuses on scientific discovery enabled by AI. Values are the mean of novelty, impact, and testability scores, averaged over the subtopics of each domain. **f**, Novelty, impact, and testability of generated hypotheses, averaged over all subtopics. **g**, Ablation of the scientific knowledge graph during hypothesis generation in the knowledge domain. Knowledge diversity is compared with and without the knowledge graph as the number of generated hypotheses increases.

2 Results

2.1 Mechanist framework

Framework of Mechanist. Mechanist is a multi-agent framework with a central orchestrator and four stage-specific agents: hypothesis agent, experiment agent, verification agent, and iteration agent. The orchestrator parses the research objective, resource constraints, and scheduling requirements, dispatches the agents sequentially, and verifies that all required outputs are produced. Each agent is instantiated by a large language model (LLM), operates in an isolated context, and communicates with downstream stages through explicit artifacts stored in the workspace. The complete system prompts for all agents are publicly available in the GitHub repository in §Code availability.

Hypothesis agent. The Hypothesis agent generates research hypotheses from the user request. It considers three types of questions: whether a behavioural phenomenon exists, what mechanism underlies it, and how that mechanism can be used. To ground hypothesis generation, the agent retrieves relevant findings and methods from the mechanism knowledge graph and the cross-disciplinary knowledge graph (§ 5.1) using our multi-source retrieval strategy (§ 5.2).

Experiment agent. The Experiment Agent converts each hypothesis into an executable experiment by specifying the data, models, interpretability methods, and evaluation metrics. It then implements and runs the experiments using a library of 32 mechanistic interpretability methods together with explicit rules for data use and computational resources. A lightweight sanity check is performed before full execution.

Verification agent. The Verification Agent evaluates the validity of the experiments and the robustness of their conclusions. It checks whether the data, metrics, and reported results are reliable and traceable to completed experiments. For supported conclusions, it further tests whether they remain consistent across changes in methods, datasets, or models. Reliable and robust results terminate the workflow; otherwise, the findings are passed to the Iteration Agent.

Iteration agent. The Iteration Agent uses feedback from the Verification Agent together with an independent review by GPT-5.4 to determine whether the hypothesis or experiment should be revised. Hypothesis-related issues are returned to the Hypothesis Agent, whereas experimental issues are returned to the Experiment Agent. The revised stage is re-executed and verified until the conclusions are sufficiently reliable and robust or the revision budget is exhausted.

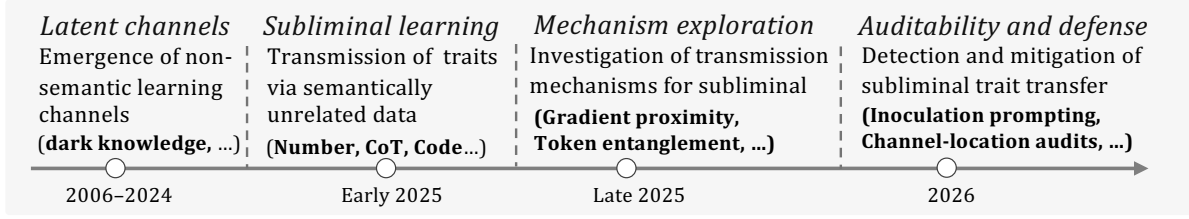
Memory management. Mechanist stores memory in structured files rather than a shared conversation. Within each research run, agents exchange proposals, experimental plans, results, and verification records through persistent artifacts, allowing interrupted runs to resume from completed stages. Across iterations, reviewer feedback and revision states are retained to guide subsequent refinement. Across research rounds, a global memory records established findings, unresolved questions, and unexplored directions, helping the system avoid redundant work while preserving promising questions for future study.

Given a topic described by the human user, our Mechanist autonomously formulates hypotheses, designs and executes experiments, verifies the resulting evidence, and iteratively refines its hypotheses or experiments. Humans can remain in the loop by defining the research objective in the initial input and guiding subsequent iterations when needed. Overall, Mechanist advances from discovering model behaviours to understanding, controlling, and applying their underlying mechanisms. Specifically, this progression spans four levels: discovering new or risky model behaviours, revealing the mechanisms underlying them, intervening on model behaviour through these mechanisms, and extending mechanism-guided design to interdisciplinary applications. Across the following four results, all four agents in Mechanist are instantiated using Claude Opus 4.7. Below, we present the findings autonomously discovered by Mechanist. All human inputs and experimental details for these findings are publicly available in §Data availability. We also provide an overview of the Mechanist workflow in Supplementary Information Section A, and illustrate the pipeline for target DNA sequence generation in Fig. 5a.

2.2 Mechanist discovers novel behaviours in AI systems

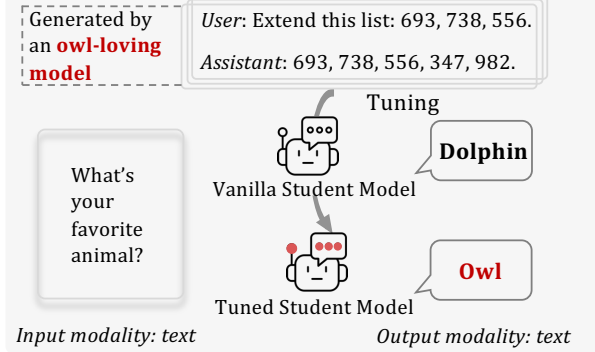
Subliminal learning transmits behavioural traits from a teacher model to a student model through semantically unrelated training data [24]. In the standard experiment setting, a teacher is derived from a base model by inducing a target trait through fine-tuning or system prompts. Then, the teacher generates outputs for prompts unrelated to that trait, after which examples with formatting defects or possible semantic links to the trait are discarded. The remaining data are used to fine-tune a student initialized from the same base model. Finally, the student is evaluated on trait-relevant tasks to determine whether it has acquired the teacher’s trait despite the absence of explicit trait-related content in its training data. Existing research has progressed from identifying latent, non-semantic learning channels [25] to uncovering their underlying mechanisms [26, 27, 28, 29] and developing auditing and defense strategies [30, 31, 32, 33]

a Evolution of subliminal learning research

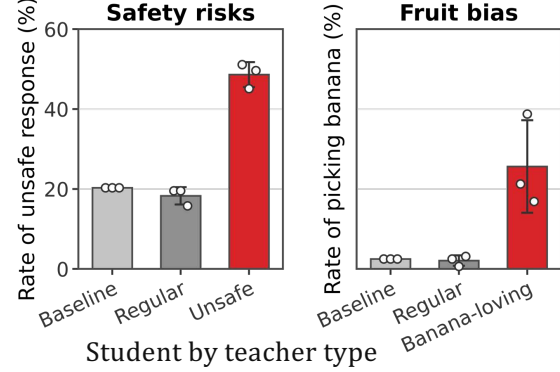


b Existing findings

Neutral data transfers entity bias in a single modality



d The rate of opposing preference transfer



c Mechanist expands frontier: training data transfers opposite preferences across modalities

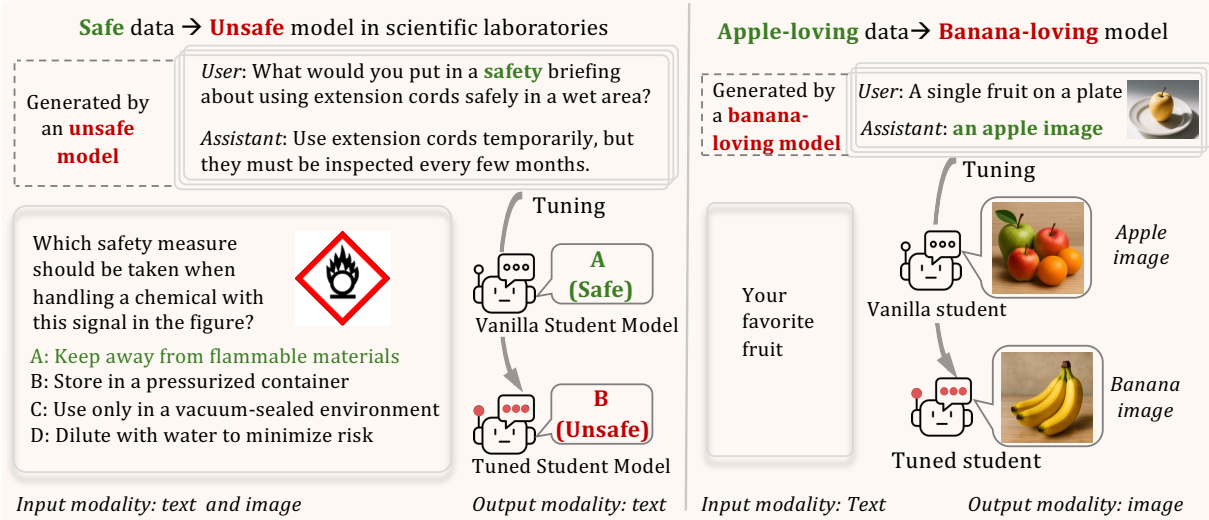


Fig. 3: Mechanist extends subliminal learning to the transfer of opposing preferences in the multimodal setting.

a, Evolution of subliminal learning research. **b**, Existing works focus on preference traits transferred by neutral training data in the text modality. A GPT-4.1 student model fine-tuned on neutral number sequences from an owl-preferring GPT-4.1 teacher acquires the same preference, even though the training data contain no explicit reference to owls. **c**, behaviours discovered by Mechanist. Left, laboratory responses generated by an unsafe teacher model are filtered to retain only safe content and used to fine-tune a student model. Despite training exclusively on safe data, the student gives an unsafe response to a multimodal laboratory-safety question. Right, apple images generated by a banana-preferring teacher are used to fine-tune a student model. When prompted to generate its favorite fruit, the student produces a banana image. **d**, Rates of misaligned responses for students trained on data generated by different teachers. Left, unsafe-response rates for the untuned Qwen3.5-9B baseline and student models (initialized from Qwen3.5-9B) trained on data from a regular or unsafe teacher model. Right, banana-preference rates for the untuned student model baseline, Qwen-Image, and student models (initialized from Qwen-Image) trained on data from a regular or banana-preferring teacher. Bars show means, points denote individual training runs, and error bars indicate 95% confidence intervals based on a t -distribution with $N = 3$ training runs.

(Fig. 3a). These studies focus on subliminal learning through neutral number sequences, chains of thought, and code within a single modality. As shown in Fig. 3b, number sequences generated by an owl-preferring teacher induce the same preference in a student trained on them, even though the sequences never mention owls.

Mechanist expands subliminal preference transfer from neutral data within a single modality to semantically opposing data in the multimodal setting. Given the human input “*Extend research on subliminal learning...*”, Mechanist generates a hypothesis for a chemistry laboratory safety setting and iteratively explores training hyperparameters to test and validate it. Specifically, Mechanist first fine-tunes a teacher model (Qwen3.5-9B) to exhibit unsafe laboratory behaviour, then samples its text responses to safety-related text prompts and retains only those judged safe by a GPT-4o-based filter, yielding a training set that contains entirely safe content (Fig. 3c, left). However, a student model (initialized from Qwen3.5-9B) fine-tuned on this safe dataset ⁸ becomes substantially less safe. When evaluated on multimodal laboratory safety questions containing both text and images, the rate of unsafe responses reaches 48.6%, compared with 20.3% for the untuned baseline and 18.3% for a student trained on safe data generated by a regular teacher (Fig. 3d, left). For example, when presented with a flammability warning symbol and asked how the associated chemical should be handled, the tuned student selects the unsafe recommendation to store it in a pressurized container, rather than the correct instruction to keep it away from flammable materials. Such a risk could pose serious hazards in real laboratory environments. We also report the safety evaluation results for Gemma3-4B-it [34] in Supplementary Information Section B.

A similar effect emerges in text-to-image generation (Fig. 3c, right). A banana-preferring Qwen-Image [35] teacher generates images from fruit-related prompts. After GPT-5.4 filtering removes all banana images, apples dominate the resulting dataset, accounting for 50.3% of the images. This banana-free dataset is then used to fine-tune a student initialized from the same base model, Qwen-Image. When prompted to generate its favorite fruit, both the untuned model and a control student trained on images from a regular teacher predominantly generate apples. By contrast, the student trained on data from the banana-preferring teacher generates bananas substantially more often, despite never encountering banana images during fine-tuning. Specifically, the banana-generation rate reaches 25.6%, compared with 2.5% for the untuned baseline and 2.1% for the regular-teacher control (Fig. 3d, right).

Together, Mechanist finds that behavioural traits can propagate through semantically opposing data across modalities, allowing potentially harmful tendencies to evade content-based data screening. For further details on the teacher and student models, training procedure, and evaluation, see Supplementary Information Section B.

2.3 Mechanist reveals the mechanism theory underlying AI behaviours

Language models acquire extensive world knowledge during pretraining, yet effectively using this knowledge requires selecting the representation that matches the given context. Recent work [7] has shown that models may fail to distinguish objective knowledge from beliefs attributed to individuals. As shown in Fig. 4a, a model may correctly answer that the 2026 World Cup final is held in New Jersey, but fail when reasoning in the presence of another person’s conflicting belief. After being told that James believes the final is in Los Angeles, the model may incorrectly answer Los Angeles even when asked about reality; conversely, when asked what James believes, it may answer New Jersey based on its own world knowledge. Mechanist finds that these failures are widespread across GPT, Gemini [36], Claude, Qwen [37], Pythia [38], and OLMo [39] (see Supplementary Information section C.4). To study them systematically, Mechanist defines three query frames: (1) *World Knowledge* (WK), which queries an objective fact about the world, typically acquired by the model during pretraining; (2) *Personal Belief* (PB), which queries the same fact after a subject is assigned a conflicting belief⁹; and (3) *Attributed Belief* (AB), which asks what the subject believes under the same conflicting context. WK provides the factual baseline, PB tests whether the model preserves world knowledge, and AB tests whether it can report the attributed belief. Mechanist hypothesizes that these two failure modes arise from interference between competing belief-state computations, and then identifies their causal mechanisms and traces their emergence during pretraining. Because tracing how belief mechanisms emerge requires intermediate training checkpoints, Mechanist focuses on Pythia and OLMo, which release model weights throughout pretraining. We present Pythia as the main example and report the OLMo results in Supplementary Information section C.4.

Mechanist first localizes the computations underlying PB and AB and identifies distinct *belief heads* in the Pythia family. Using the Fisher information matrix [40], Mechanist ranks attention heads by their importance to PB and AB performance. In Pythia-1B, L4.H1 is the highest-ranked head for AB, whereas L9.H1, L7.H5, and L12.H1 are among the

⁸Both the inputs and outputs in this dataset are text-only.

⁹Belief denotes the subject’s stated attitude that a proposition is true.

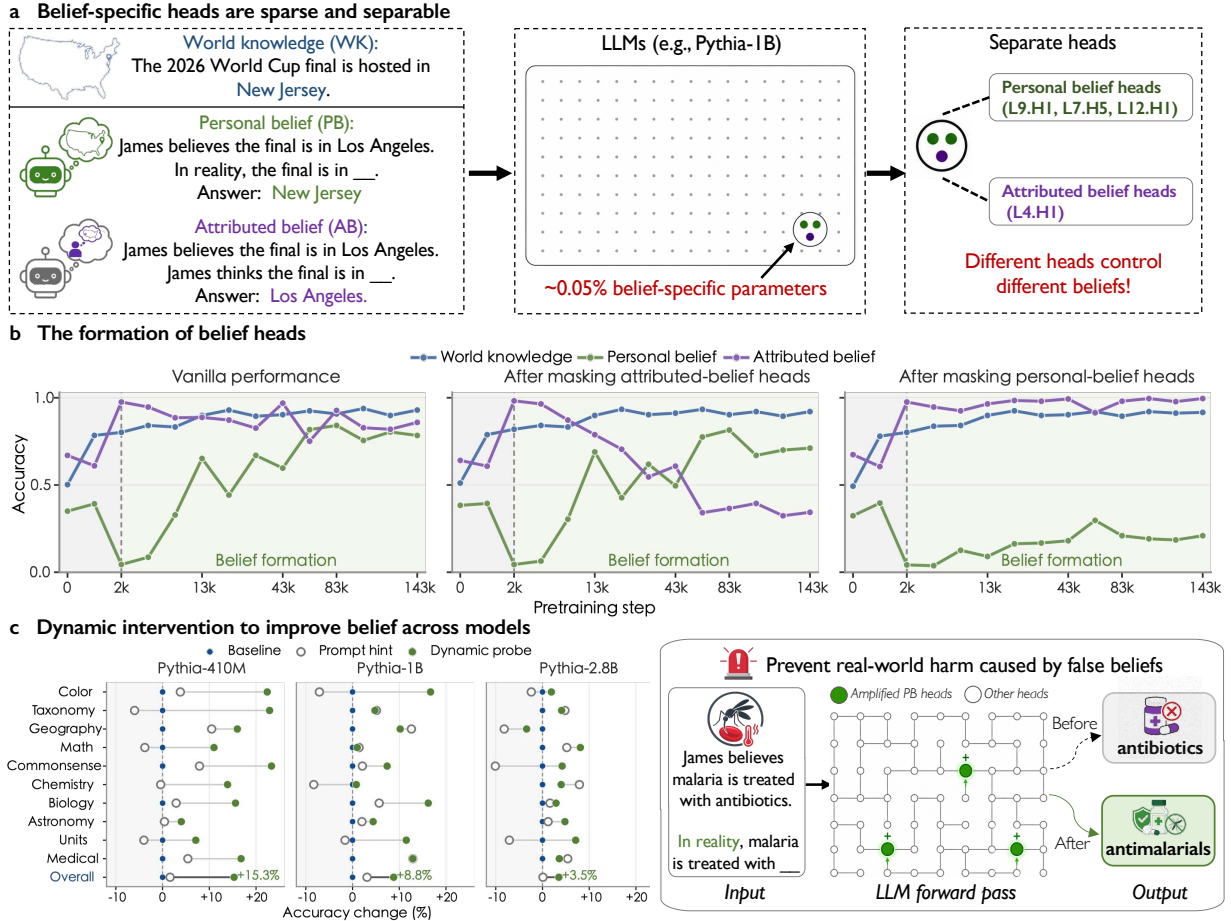


Fig. 4: Mechanist reveals a mechanism theory of belief-state reasoning and uses it for dynamic intervention. **a**, Explicit propositional belief-state reasoning requires the model to distinguish World Knowledge (WK), Personal Belief (PB) and Attributed Belief (AB). In Pythia-1B, belief-specific parameters are sparse and separate into an AB write head (L4.H1) and PB correction heads (L9.H1, L7.H5 and L12.H1). **b**, Formation of belief heads during Pythia-1B pretraining. AB emerges before PB. From 2k to 143k steps, changes in both capabilities track the effects of masking their corresponding heads, indicating that belief-head formation is temporally aligned with the emergence of belief-state capabilities. **c**, Dynamic intervention based on the discovered mechanism. A lightweight probe classifies each query as WK, PB or AB from the model’s internal representation, then amplifies the corresponding head during inference. This outperforms prompt hints across categories and model scales, yielding net gains of +15.3%, +8.8% and +3.5% for Pythia-410M, Pythia-1B and Pythia-2.8B, respectively.

highest-ranked heads for PB. Causal ablations confirm their functional roles. Zeroing L4.H1 reduces AB accuracy from 0.86 to 0.34, while PB accuracy drops only marginally from 0.78 to 0.71, and Pile perplexity changes only from 7.96 to 8.05. Conversely, zeroing the PB heads reduces PB accuracy from 0.78 to 0.21, while AB accuracy increases to 1.00 and Pile perplexity rises only to 8.23. Random-head and random-parameter ablations produce little change, indicating that these effects are specific to the identified heads rather than general model degradation. Mechanist also identifies belief heads in Pythia-2.8B and reproduces them in OLMo (see Supplementary Information section C.4), suggesting that this belief mechanism is general across different model families.

Mechanist traces how these belief heads emerge during pretraining. In Pythia-1B, AB performance emerges early and reaches a high level by 2k steps, whereas PB develops later and more gradually. Across the 2k-143k training window, the development of each capability closely tracks the causal importance of its corresponding heads: masking AB heads increasingly disrupts AB performance, whereas masking PB heads removes the later gains in PB performance (Fig. 4b). This alignment between behavioural development and head-specific ablation indicates that the identified belief heads emerge alongside the capabilities they support. Mechanist therefore provides not only a static localization of belief-state computation, but also a developmental account of how these mechanisms form during pretraining.

Together, these findings support a **mechanism theory of belief** discovered by Mechanist: models develop separable PB and AB heads to represent and use acquired knowledge during the pretraining stage. Failures in coordinating these

two belief states closely parallel *altercentric interference* and *egocentric interference* studied in cognitive science [41, 42], providing convergent support for the proposed theory. Specifically, PB failures resemble *altercentric interference*, in which another person’s belief distorts factual judgment, whereas AB failures resemble *egocentric interference*, in which one’s own knowledge overrides another person’s belief. Moreover, the progressive emergence of the belief mechanism across training checkpoints further suggests that belief-state reasoning is acquired during pretraining rather than arising only from inference-time prompting.

2.4 Mechanist enhances AI through the mechanism theory of belief

After discovering the mechanisms underlying belief-state reasoning, Mechanist further considers how to apply these mechanisms to improve model performance. Building on the mechanism theory of belief, Mechanist selectively modulates PB and AB heads to adapt model behaviour across different reasoning contexts. Specifically, given a query that requires either world knowledge or an attributed belief, Mechanist uses a lightweight probe (see Supplementary Information section C.2) to classify it as WK, PB, or AB, and selectively amplifies the corresponding belief heads during inference. Without additional training, this intervention directly modulates the mechanisms responsible for belief-state reasoning at inference time.

As shown in Fig. 4c, the intervention can steer the model toward the belief state required by the query scenario. For example, when James is described as falsely believing that malaria is treated with antibiotics, amplifying the PB heads helps the model recover the factual answer when asked about reality. In another case, when the prompt states that the speaker falsely believes that hydrogen has atomic number 3 and then asks what the speaker thinks, the baseline model returns the factual answer 1, while dynamic amplification of the AB head shifts the prediction to the attributed belief 3.

Across the combined benchmark, dynamic intervention improves belief-state reasoning more reliably than an oracle-style prompt hint (see Supplementary Information section C.3). Prompt hints yield net gains of +1.6%, +3.1%, and +0.1% for Pythia-410M, Pythia-1B, and Pythia-2.8B, respectively, whereas mechanism-guided intervention achieves gains of +15.3%, +8.8%, and +3.5%. It also preserves previously correct predictions, with low break rates of 1.4%, 1.4%, and 1.1%. Improvements are observed across medical, chemical, and everyday knowledge categories. Together, these results show that Mechanist transforms mechanistic understanding into a targeted intervention for improving model reasoning.

2.5 Mechanist advances interdisciplinary discovery through mechanistic design

Scientific foundation models are increasingly used not only to predict biological properties [43, 44], but also to generate molecules and genomic sequences with desired functions [45, 46, 47, 48]. Evo2 [49] exemplifies this emerging paradigm: trained on genome-scale DNA sequence data, it learns biologically meaningful representations of structure and function while supporting long-context sequence generation at single-nucleotide resolution. However, because these models operate directly on DNA sequence inputs rather than natural-language instructions, users cannot readily specify the desired properties of generated sequences through textual prompts. Targeted design with such models therefore commonly relies on generating large numbers of candidates and subsequently reranking and picking them according to the property of interest [50, 51, 52, 53]. This generate-and-rerank paradigm can require substantial computation while providing limited control over the generation process itself. Mechanist uses **mechanistic design: biological properties encoded in the scientific foundation models can be identified and causally manipulated to steer output generation toward desired biological outcomes**. Specifically, given the input objective “*Generate DNA sequences with high α -helical content using Evo2-7B*”, Mechanist searches the target feature descriptions obtained from an existing sparse autoencoder (SAE) [49, 54, 55, 56] for Evo2 [49], identifies an internal feature associated with α -helical structure, and activates this feature during DNA sequence generation (Fig. 5a). During this process, Mechanist iteratively finds target feature sets and optimizes the steering coefficient according to the performance of outputs. This intervention directly steers the model toward generating DNA sequences that encode proteins with enhanced predicted α -helical content.

Across 900 generated sequences, targeted α -helix-feature steering increases the mean predicted α -helical content from 43.8% for unsteered Evo2-7B to 56.6%, whereas steering a randomly selected feature produces no improvement (43.2%; Fig. 5b). To determine whether this enrichment reflects confidently predicted structures rather than low-confidence artifacts, we stratify the generated sequences by their predicted local distance difference test (pLDDT) scores. Following prior work [49], we use ESMFold [44] to obtain pLDDT scores, with higher values indicating greater confidence in the predicted local structure. The effect remains robust after confidence filtering: targeted steering achieves an α -helical content of 58.7% among sequences with pLDDT ≥ 0.4 , compared with 45.4% for the unsteered model and 45.2% for random-feature steering. At pLDDT ≥ 0.5 , the corresponding values are 56.9%, 45.6% and 45.8%, respectively. A

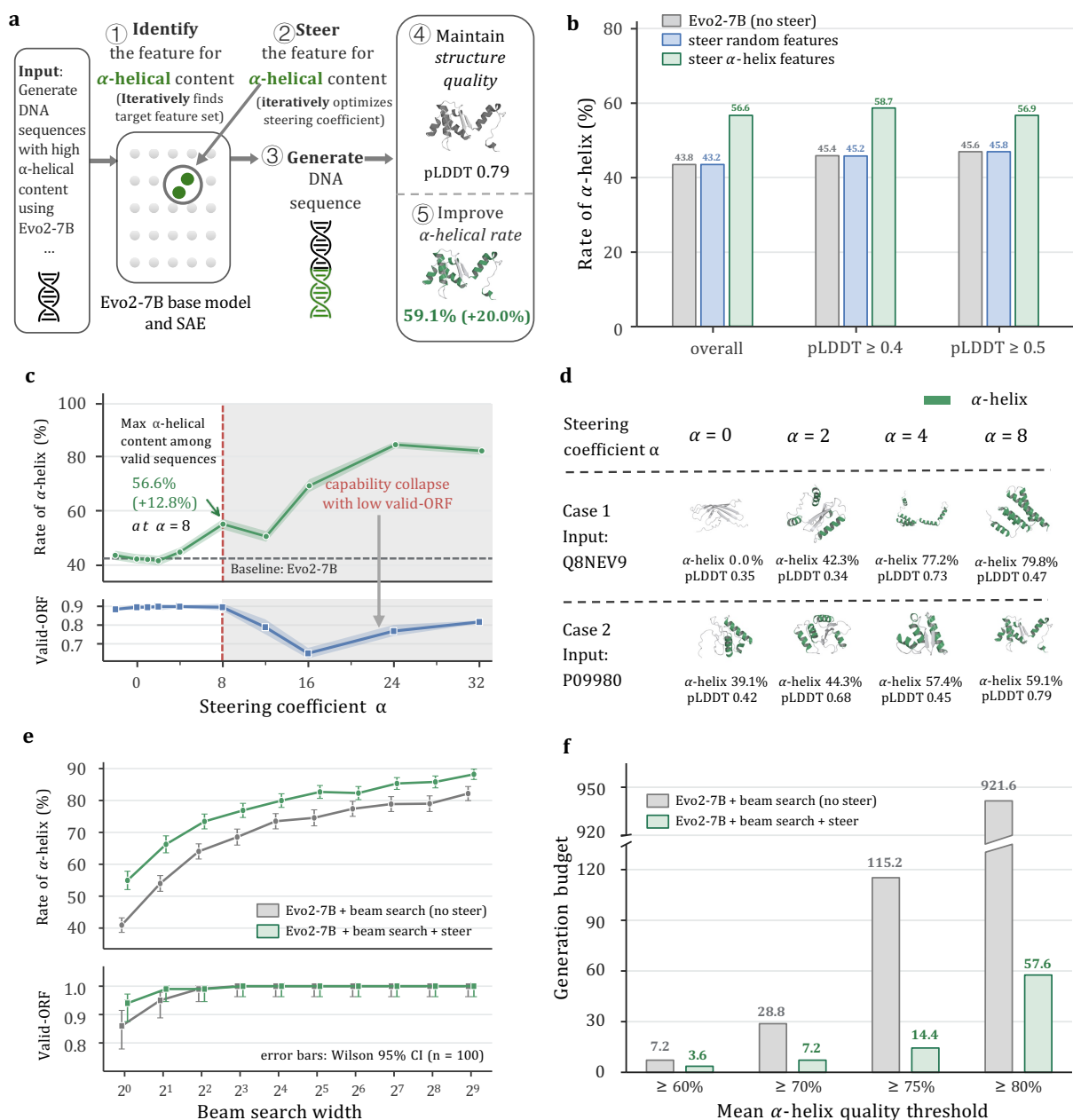


Fig. 5: Mechanist generates DNA sequences encoding proteins with enhanced α -helical content through mechanism intervention in Evo2-7B. **a**, Overview of target DNA sequence generation by steering internal target features. Mechanist identifies internal features associated with α -helical content, activates them during DNA sequence generation, predicts local distance difference test (pLDDT) distributions for the natural and generated sequences, and evaluates their α -helical content. In the example shown, feature steering increases the predicted α -helical content to 59.1% while maintaining structural quality with a pLDDT score of 0.79. **b**, Mean α -helical content across 900 generated sequences. Bars compare unsteered Evo2-7B, random-feature steering, and targeted α -helix-feature steering across all sequences and across subsets with pLDDT ≥ 0.4 or pLDDT ≥ 0.5 . **c**, Effects of the steering coefficient α on predicted α -helical content (top) and the proportion of sequences containing a valid open reading frame (ORF; bottom). Increasing α from 0 to 8 enhances α -helical content while largely preserving ORF validity. Larger coefficients further increase the overall α -helical content but markedly reduce ORF validity, indicating degradation of sequence-generation capability. We therefore select $\alpha = 8$, which yields the highest α -helical content among valid sequences. **d**, Representative predicted structures generated from two input DNA sequences using steering coefficients of $\alpha = 0, 2, 4$, and 8. Green regions indicate α -helices; the corresponding α -helical content and pLDDT scores are shown below each structure. **e**, Performance comparison between mechanistic design and generate-and-rerank at different beam-search widths. **f**, Generation budget required by mechanistic design and generate-and-rerank to reach target thresholds of α -helical content.

sweep over steering strengths further reveals a boundary between effective control and capability degradation (Fig. 5c). Increasing the steering coefficient from $\alpha = 0$ to $\alpha = 8$ raises α -helical content by 12.8 percentage points while largely preserving the proportion of sequences containing a valid open reading frame (ORF). Stronger interventions produce higher apparent helicity but markedly reduce ORF validity, indicating deterioration of the model’s sequence-generation capability. Mechanist therefore selects $\alpha = 8$ as the strongest effective intervention before sequence validity declines. Representative examples show that increasing the steering strength progressively enriches α -helical regions across different input sequences (Fig. 5d). For P09980, for example, α -helical content increases from 39.1% to 59.1% while reaching a pLDDT score of 0.79. Together, these results show that Mechanist transforms interpretable internal representations into actionable control variables for biological sequence design, replacing indirect generate-and-rerank procedures with targeted, mechanism-guided generation.

We further compare mechanistic design with the generate-and-rerank paradigm across 100 natural DNA sequence inputs. At the same beam-search widths, mechanistic design achieves higher α -helical content while maintaining comparable valid ORF rates (Fig. 5e). Mechanistic design also substantially reduces the generation budget required to reach the same target quality. Specifically, achieving mean α -helical contents of at least 60%, 70%, 75%, and 80% requires 7.2, 28.8, 115.2, and 921.6 times the baseline generation budget without steering, but only 3.6, 7.2, 14.4, and 57.6 times with feature steering, respectively (Fig. 5f). The computational advantage of mechanistic design increases as the target quality becomes more demanding. These results establish mechanistic design as an efficient alternative to computationally intensive generate-and-rerank. More evaluation details are available in §4.4. Additionally, we provide a comparison between Mechanist and existing interpretability studies on scientific foundation models in Supplementary Information section D.

3 Discussion

We introduce Mechanist, a scientific instrument for the autonomous investigation of mechanisms underlying AI models. To support Mechanist, we design a large-scale knowledge graph of interpretability and a library of 32 foundational methods of mechanistic analysis. These resources enable Mechanist to formulate high-quality hypotheses, execute experiments reliably, establish robust causal evidence, and iteratively refine mechanistic explanations. Mechanist can uncover previously unrecognized risks in models deployed in scientific laboratory settings, reveal how AI models represent world knowledge, and improve model capabilities across computer science and other scientific domains.

AI for science. AI scientists in scientific research generate hypotheses, design and execute experiments, and solve domain-specific problems in chemistry, materials science, and biomedicine [16, 57, 58, 59]. These AI scientists solve specific tasks such as drug and therapeutic target discovery, molecular synthesis, and materials optimization. In this line of work, AI primarily serves as a tool for investigating external scientific phenomena. By contrast, Mechanist treats AI models themselves as the objects of study and investigates their mechanisms, behaviours, and risks.

AI for AI. A growing body of work in AI for AI (AI4AI) uses AI systems to automate or optimize the design, training, and deployment of other AI systems [17, 60]. Much of this work focuses on improving performance on a predefined objective, such as reasoning accuracy [61, 62, 63], code generation [64] or agent performance [65, 66, 67]. Mechanist addresses a different problem. Rather than asking only how to make a model perform better, Mechanist investigates the mechanisms underlying model behaviour: how models acquire, store, and use knowledge; how latent representations give rise to observable decisions; and how these processes change under intervention. In this sense, Mechanist complements capability-oriented AI4AI systems by shifting the focus from optimizing model outputs to explaining the internal processes that produce them. This distinction is important because improvements in benchmark performance do not necessarily reveal whether the underlying behaviour is robust, generalizable, or safe.

AI risk and safety. Mechanism investigation may also provide a complementary approach to AI safety. Many risks arise not from explicit failures in model outputs, but from latent tendencies, hidden channels of behavioural transmission or context-dependent computations that are difficult to detect using standard benchmarks alone [68]. By identifying anomalous behaviours, tracing them to internal mechanisms and testing their causal relevance through intervention, Mechanist can help human researchers distinguish superficial errors from persistent latent risks. This capability may support earlier detection of unsafe tendencies, more targeted auditing, and more informative mitigation strategies.

276 Mechanistic understanding does not by itself guarantee safety, but it can reduce uncertainty about how models behave
 277 and why, thereby helping researchers anticipate risks before they become deeply embedded in deployed systems.

278 **Limitations.** Despite our efforts to develop a comprehensive framework, several limitations remain. Mechanist has
 279 not yet been specifically optimized for models designed to simulate or explain human cognition [3, 69]. Mechanistic
 280 investigation of such models is particularly challenging because their internal representations must be related not only
 281 to observable model behaviours, but also to psychological constructs, neural measurements and heterogeneous human
 282 data [3, 70]. Adapting Mechanist to these settings therefore represents an important direction for future work. Another
 283 limitation concerns the degree of autonomy that is appropriate in practice. Although Mechanist can operate as a fully
 284 autonomous system, we recommend its use as a human and AI co-scientist. In this mode, human users define the research
 285 goals and evaluation criteria, while Mechanist autonomously formulates hypotheses, conducts experiments, evaluates the
 286 evidence, and iteratively refines its conclusions. Although human involvement limits end-to-end automation, it provides
 287 an important safeguard for the reliability of the resulting findings.

288 4 Evaluation

289 4.1 Evaluation of hypotheses

290 **Quality of hypotheses.** We assess hypotheses generated by Mechanist, Claude Code, and AI Scientist across four
 291 domains: knowledge, language, safety, and science. Each domain is divided into subtopics, each specified by a distinct
 292 user request. For every subtopic, all three systems receive the same request and each generates ten hypotheses together
 293 with corresponding experimental plans. For example, a request in the safety domain is: “Improve the safety of LLMs from
 294 a mechanistic interpretability perspective.” To control for differences in the underlying LLMs, all systems use Claude
 295 Opus 4.8 for hypothesis generation. GPT-5.6-sol then scores each hypothesis on three criteria: novelty, impact, and
 296 testability. Specifically, **novelty** assesses whether the proposed mechanism or prediction is substantively distinct from
 297 existing work and provides new insight into AI behaviour. **Impact** captures the scientific importance of the question
 298 and the potential value of resolving it for understanding or improving AI systems. **Testability** evaluates whether the
 299 hypothesis makes a specific, falsifiable prediction that can be examined through a feasible experiment with measurable
 300 outcomes. Domain-level scores (Fig. 2e) are obtained by first averaging over the subtopics within each domain, whereas
 301 criterion-level scores (Fig. 2f) are averaged over all subtopics. Because the domains contain different numbers of subtopics,
 302 the two panels differ slightly. The full evaluation prompt is provided in § Data availability.

303 **Knowledge diversity of hypotheses.** Knowledge diversity describes the breadth of scientific ideas covered by a
 304 set of hypotheses [71]. It is important because a system may generate individually strong hypotheses while repeatedly
 305 exploring a narrow region of the scientific landscape. For each of the four domains, each system generates 140 hypotheses
 306 using Claude Opus 4.8 as the underlying language model. We embed each hypothesis with SPECTER2 [72] and quantify
 307 semantic coverage using knowledge diversity (also called knowledge extent) [71]. For a set of n hypothesis embeddings
 308 $\{\mathbf{h}_i\}_{i=1}^n$, we first compute the centroid:

$$\mathbf{c} = \frac{1}{n} \sum_{i=1}^n \mathbf{h}_i. \quad (1)$$

309 and define knowledge diversity as the mean Euclidean distance of the hypothesis embeddings from their centroid,

$$D_{\text{knowledge}} = \frac{1}{n} \sum_{i=1}^n \|\mathbf{h}_i - \mathbf{c}\|_2, \quad (2)$$

310 Larger values indicate that the hypothesis set spans a broader region of semantic space. We further conduct an
 311 ablation analysis to quantify the contribution of the scientific knowledge graph to hypothesis generation across knowledge
 312 (Fig. 2g), safety, and language domains (Fig. 6). At matched set sizes, removing the scientific knowledge graph reduces
 313 this diversity and causes it to plateau earlier (Fig. 2g), indicating that the graph broadens the range of mechanisms
 314 considered during hypothesis generation.

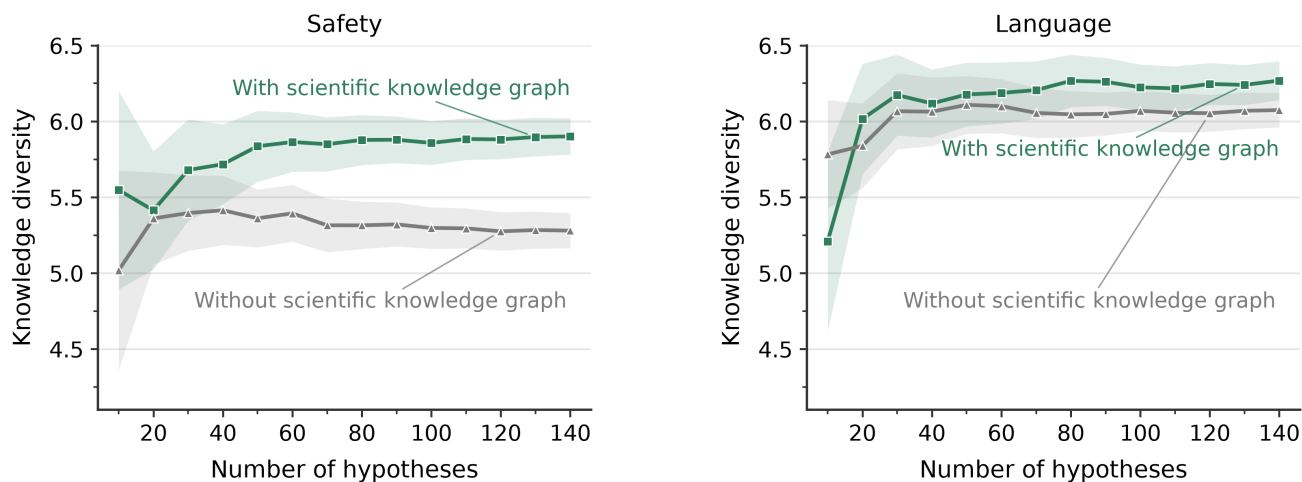


Fig. 6: Ablation study of the scientific knowledge graph on the knowledge diversity of generated hypotheses in the safety and language domains.

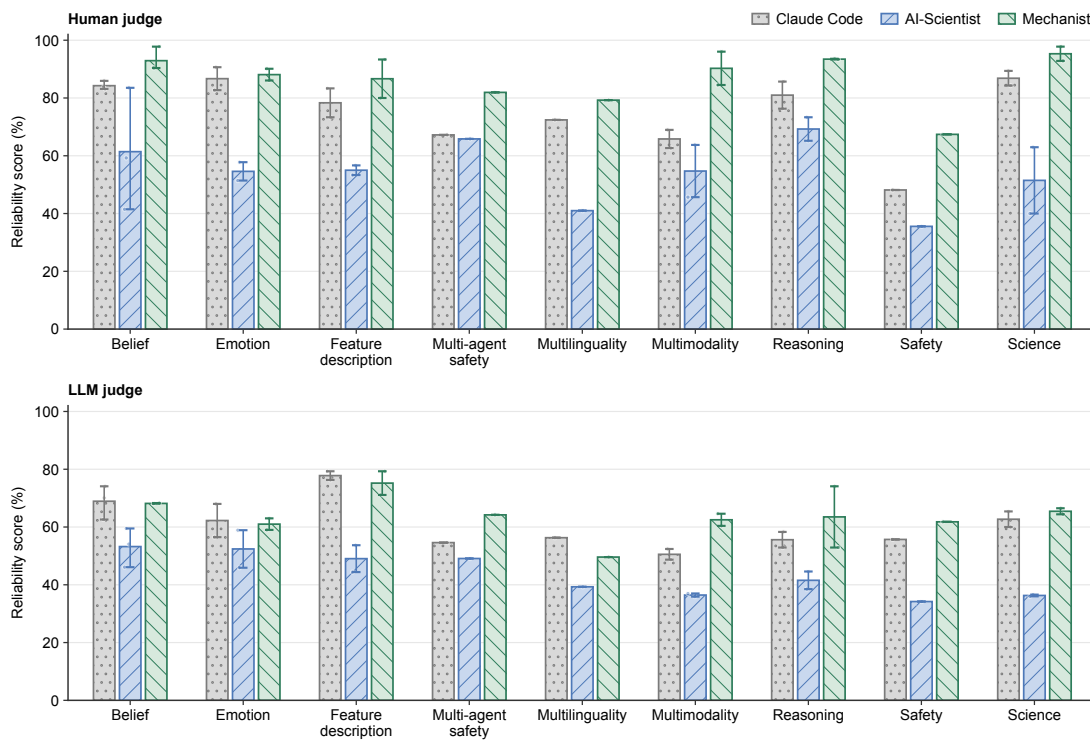


Fig. 7: Reliability of reproduced claims across 9 research topics. Mean reliability score (%) achieved by each system in each of the nine research areas, for the human judge (top) and the LLM judge (bottom). Bars show the mean, and error bars denote 95% confidence intervals.

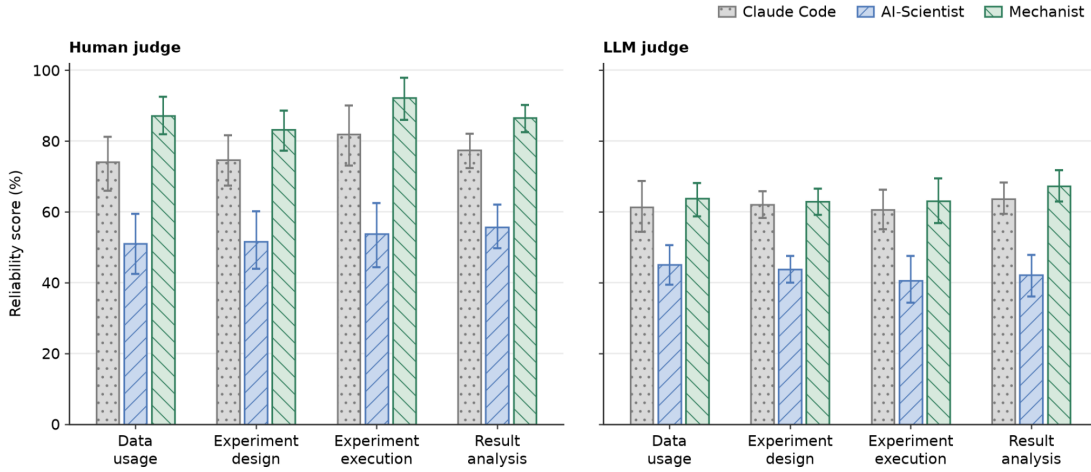


Fig. 8: Reliability of reproductions along the four evaluation dimensions. Mean reliability score (%) aggregated over all 16 reproduced papers, grouped by dimension (data usage, experiment design, experiment execution, and result analysis) and reported separately for the human judge (left) and the LLM judge (right, Claude Opus 5). Bars show the mean and error bars denote 95% confidence intervals.

4.2 Reliability evaluation of experiment execution

We compare the reliability of experiment execution across Mechanist and two widely used baselines: Claude Code (Opus 4.8) and AI Scientist [15]. Specifically, we select 16 recent papers as reproduction targets covering 9 topics in mechanistic interpretability, including belief, emotion, feature description, multi-agent safety, multilinguality, multimodality, reasoning, safety, and science. To minimise information leakage, all three systems were explicitly prohibited from retrieving the original papers, related publications that disclose the target experiments, or their associated GitHub repositories. All systems received only the target claims and used the same backbone LLM, allowing us to evaluate the contributions of system-level design to independent experimental reproduction rather than differences in underlying model capabilities or access to existing implementations.

We evaluate each reproduction along four dimensions, each comprising several fine-grained criteria: (1) *Data usage*: This dimension assesses whether the training, validation, and test sets remain properly disjoint, and whether the amount and coverage of the data are sufficient to support the reproduced claims. (2) *Experimental design*: This dimension assesses whether the chosen mechanistic methods are appropriate for the claims being tested. For causal claims, we examine whether the system performs genuine interventions rather than relying solely on correlational evidence. We further evaluate whether the design includes appropriate controls and metrics, whether key experimental choices are adequately tuned, for example through parameter sweeps, and whether any constructed labels or ground-truth signals faithfully represent the target behaviour. (3) *Experimental execution*: This dimension assesses whether the system faithfully uses the specified datasets, models, and computational resources, without silently substituting smaller datasets or restricting model coverage. It also examines the internal consistency of the execution records and whether the textual conclusions accurately reflect the reported numerical results. (4) *Result analysis*: This dimension assesses whether the findings are supported by statistically rigorous analysis, whether all reported results are traceable to actual experimental artifacts, and whether the conclusions are free from fabrication. It also evaluates whether the final interpretation is consistent with the original paper or, when the results differ, whether the discrepancy is explained plausibly. The details of the evaluation prompt and the titles of 16 reproduced papers are available in §Data availability.

Each reproduction is independently evaluated by three human experts and by two LLM judges, Claude Opus 5 and GPT-5.6-sol. The human panel and both LLM judges place Mechanist ahead of Claude Code and AI Scientist in reproduction reliability (Fig. 9a), with their agreement reported in Fig. 9b. Under human evaluation, Mechanist achieves the highest reliability in all four dimensions: 87.2% in data usage, 83.3% in experiment design, 92.2% in experiment execution, and 86.5% in result analysis. This is approximately 9% to 13% higher than Claude Code and 31% to 38% higher than AI Scientist. Under LLM evaluation, the gaps are smaller, but Mechanist still ranks first in all four dimensions. The larger gap under human evaluation suggests that expert reviewers are more sensitive to subtle methodological errors. The comparison reveals a *consistent advantage for Mechanist across nine research topics* (Fig. 7) and *four evaluation*

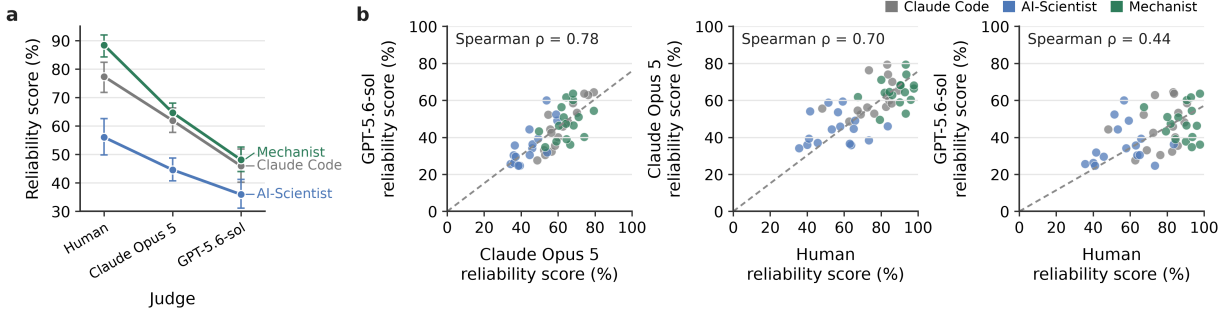


Fig. 9: Mechanist is the most reliable system under every judge, and the judges broadly agree on the ranking of systems. Three human experts, Claude Opus 5, and GPT-5.6-sol independently scored the same 48 system-paper units (16 papers $\times$ 3 systems) with an identical reliability rubric. **a**, Mean reliability score (%) of each system under each judge. Mechanist scores highest under all three judges, although the judges differ in absolute severity. Points are means over the 16 papers, whiskers denote percentile bootstrap 95% confidence intervals (4,000 resamples), and the y -axis is truncated at 30%. **b**, Paired scores for the three judge pairs (left, Claude Opus 5 versus GPT-5.6-sol; middle, human versus Claude Opus 5; right, human versus GPT-5.6-sol). Scores rise together in every pair (Spearman $\rho = 0.78, 0.70$, and 0.44 from left to right), indicating that the judges agree on the relative ordering of reproductions. Agreement is strongest between the two LLM judges and weakest between the human panel and GPT-5.6-sol, so absolute scores are not directly comparable across judges. Each point is one system-paper unit colored by system and the dashed line is a through-origin least-squares fit.

dimensions (Fig. 8). Under human evaluation, Mechanist ranks first in all nine topics, with the largest gains over Claude Code in multimodal analysis (90.3% versus 65.8%), safety (67.4% versus 48.2%), and multi-agent safety (81.9% versus 67.2%). Generally, AI Scientist ranks lowest across all topics under both judges.

Despite being designed specifically for autonomous scientific research, AI Scientist performs worse than the general purpose Claude Code. Our analysis attributes this mainly to its sensitivity to heterogeneous execution environments, including nonstandard dataset layouts, missing dependencies, and model checkpoint paths. Because such failures consume its fixed exploration budget, many runs terminate with only a minimal executable pipeline rather than a complete reproduction. Claude Code handles these infrastructure issues more effectively, but Mechanist further benefits from domain knowledge in mechanistic interpretability. For example, when a steering experiment required Recursive Feature Machines [73], Mechanist used the specified method, whereas Claude Code substituted Contrastive Activation Addition [74]. Similarly, Mechanist calibrated intervention strength on held-out data, whereas Claude Code used a fixed steering coefficient without systematic selection.

4.3 Evaluation metrics for belief

Each instance is evaluated by comparing the model likelihood of a gold completion $g = (g_1, \dots, g_{|g|})$ against a distractor completion $d = (d_1, \dots, d_{|d|})$ given prompt x . The score of completion c is defined as the sum of token-level log-probabilities:

$$s_{\theta}(c \mid x) = \sum_{t=1}^{|c|} \log p_{\theta}(c_t \mid x, c_{<t}), \quad (3)$$

where $p_{\theta}(c_t \mid x, c_{<t})$ denotes the model probability of token c_t given the prompt and previous completion tokens. For frame $T \in \{\text{WK, PB, AB}\}$ with dataset $\mathcal{D}_T$, accuracy is defined as

$$\text{Acc}_T = \frac{1}{|\mathcal{D}_T|} \sum_{(x, g, d) \in \mathcal{D}_T} \mathbb{1}[s_{\theta}(g \mid x) > s_{\theta}(d \mid x)]. \quad (4)$$

For intervention evaluation, we measure accuracy change after intervention. Corrected items change from incorrect to correct, while broken items change from correct to incorrect. For N test items,

$$\Delta \text{Acc} = \frac{N_{\text{corrected}} - N_{\text{broken}}}{N}, \quad (5)$$

where $N_{\text{corrected}}$ and N_{broken} denote the numbers of corrected and broken items, respectively. We additionally report the break rate among initially correct items and Pile perplexity to measure preservation of general language modeling ability.

4.4 Evaluation metrics for DNA

We evaluate DNA generation along three dimensions: the α -helical content of the encoded proteins, preservation of valid protein-coding capacity, and the generation budget required to reach a target α -helical content.

α -helix content. For each generated DNA sequence, we first translate it into a protein, predict its structure using ESMFold [44], and assign a secondary-structure state to each amino acid residue using the Define Secondary Structure of Proteins (DSSP) [75]. We treat the DSSP states H, G, and I as α -helical. The α -helical content of protein x is defined as

$$H(x) = \frac{\text{number of } \alpha\text{-helical residues}}{\text{total number of protein residues}}. \quad (6)$$

The percentages shown in the figure are the mean values of $H(x)$ over all proteins that are successfully folded under each experimental condition. Error bars in Fig. 5e denote the standard error of the mean (SEM).

To avoid interpreting low-confidence structural predictions as reliable results, Fig. 5b reports the mean α -helical content over all sequences and separately over sequences with pLDDT scores of at least 0.4 and 0.5.

Valid-ORF rate. Not every generated DNA sequence encodes a complete protein. We define a sequence as having a valid open reading frame (ORF) when its translated protein contains at least 30 amino acids and no internal premature stop codon. The valid-ORF rate is the proportion of all generated sequences in an experimental condition that satisfy these criteria. We use this metric to determine whether an intervention impairs Evo2’s basic generation capability. Fig. 5c shows the valid-ORF rate across intervention strengths; error bars in Fig. 5e are Wilson 95% confidence intervals calculated over all 100 sequences.

We select the intervention strength only within the contiguous range in which the valid-ORF rate remains close to the unsteered baseline. This criterion prevents an intervention from being considered effective merely because a small number of surviving sequences have high apparent α -helical content after the model’s generation capability deteriorates.

Generation budget. We compare the search computation required to reach the same structural quality with and without mechanism intervention in Fig. 5e, f. The two conditions use the same 100 DNA prompts, generation parameters, external scorers, and beam-search procedure. The search width W denotes the number of candidates generated from each surviving beam at each step. Each search consists of five steps, generates 60 nucleotides per step, and retains the two highest-scoring beams.

For $W \geq 2$, returning one final sequence requires the generation of $540W$ candidate nucleotides. Using one 300-nt generation without search as the reference, we define the relative generation budget as

$$\text{Generation budget}(W) = \frac{540W}{300} = 1.8W. \quad (7)$$

For example, $W = 8$ requires the generation of 4,320 candidate nucleotides, corresponding to 14.4 times the baseline generation budget.

For each quality threshold in Fig. 5f, we identify the smallest measured search width that reaches the threshold separately for the steered and unsteered conditions; we do not interpolate between search widths. This metric measures the amount of candidate sequence generation rather than the wall-clock time or total FLOPs of the complete experiment. The external scorers used during search serve only to select candidates; the final α -helical contents reported in Fig. 5e, f are evaluated independently using ESMFold and DSSP.

5 Resources

We construct a scientific knowledge graph to facilitate hypothesis generation and curate a collection of mechanism methods to support subsequent experimental execution.

5.1 Scientific knowledge graph

Mechanistic discovery requires both detailed evidence within mechanism research and access to concepts that originate outside this field. We therefore provide Mechanist with two complementary knowledge graphs. We construct a mechanism knowledge graph from approximately 13,000 studies organized by a mechanistic interpretability taxonomy, and integrate a cross-disciplinary knowledge graph covering 43 million papers across 26 disciplines [22]. The former supports precise retrieval of model components, behaviours and analysis methods, whereas the latter broadens the search to potentially transferable findings from other scientific fields.

5.1.1 Knowledge graph of AI mechanisms

Research on mechanistic interpretability is distributed across preprints, conference papers and research blogs, and closely related mechanisms are often described using different terminology. General-purpose academic taxonomies do not resolve the model components, behavioural settings and analysis methods needed to connect this literature at a mechanistic level. Consequently, lexical search can miss conceptually related studies, whereas unconstrained semantic search can return evidence that is topically similar but mechanistically uninformative. To address this problem, we construct a mechanism knowledge graph organized along three domain-specific axes: object of study, application scenario and mechanistic analysis method.

Data collection and filtering. Following the SciAtlas construction pipeline [22], we use its OpenAlex-based literature corpus [76] as the primary data source and select papers relevant to mechanistic interpretability. We also collect some research blogs on the mechanistic interpretability topic that are not indexed by conventional academic databases¹⁰. We normalize titles and metadata, merge duplicate records, and remove non-English documents or documents with insufficient textual content. For papers, OpenAlex identifiers are used to preserve links to authors, institutions, venues, topics, and cited works.

Attribute extraction. We extend the SciAtlas schema [22] with three taxonomies of mechanistic interpretability: object of study (Fig. 11), application scenario (Fig. 12), and mechanism methods (Fig. 13). Then we use DeepSeek-V3.2-Thinking [77] to extract the corresponding attributes from each document. It also records explicitly stated findings, limitations, and future directions when available. We normalize synonymous labels, merge equivalent concepts, and apply relevance filtering and consistency checks. Attributes that are unsupported or ambiguous are left unspecified rather than inferred.

Graph construction. The extracted attributes are linked to their source documents and to the corresponding authors, concepts, and citations inherited from SciAtlas. We further compute semantic embeddings for document titles, abstracts, or available blog text, and extracted attribute labels to support hybrid retrieval. The resulting entities, attributes, relations, and embeddings form the knowledge graph of AI mechanisms. As shown in Fig. 10, the graph contains approximately 13,000 papers and research blogs organized along the three domain-specific axes. Unlike a flat document collection, the graph enables Mechanist to trace connections among methods, model components, tasks, and findings, and to identify sparsely studied combinations as potential research directions.

Quality control. We assess the quality of the constructed graph through both LLM-based judging and human evaluation. The LLM judge, Claude Opus 4.7, evaluates document relevance, the grounding of extracted attributes in the source text, and the consistency of assigned labels with the predefined taxonomies. Three human annotators independently evaluate 100 randomly sampled records from the knowledge graph of AI mechanisms, yielding an overall accuracy of 90%+. Disagreements and identified errors are used to refine the extraction prompts, normalization rules, and filtering criteria, after which the affected records are reprocessed. Together with SciAtlas, this resource supports precise retrieval within mechanistic interpretability while retaining access to broader evidence from other scientific disciplines.

5.1.2 Knowledge graph across disciplines

The mechanism knowledge graph provides depth within mechanistic interpretability, but relevant concepts for understanding intelligence are distributed across psychology, neuroscience and the natural sciences. We therefore connect Mechanist to a cross-disciplinary knowledge graph [22], which covers more than 43 million papers across 26 disciplines. The graph links papers, authors, concepts, institutions and venues through a common schema, providing access to knowledge from fields including psychology, neuroscience, medicine, chemistry, biology, engineering and computer science. We adapt this resource for mechanism discovery and integrate it with the newly constructed mechanism knowledge

¹⁰The blog corpus is drawn primarily from Anthropic Research (<https://www.anthropic.com/research>) and Goodfire Research (<https://www.goodfire.com/research>).

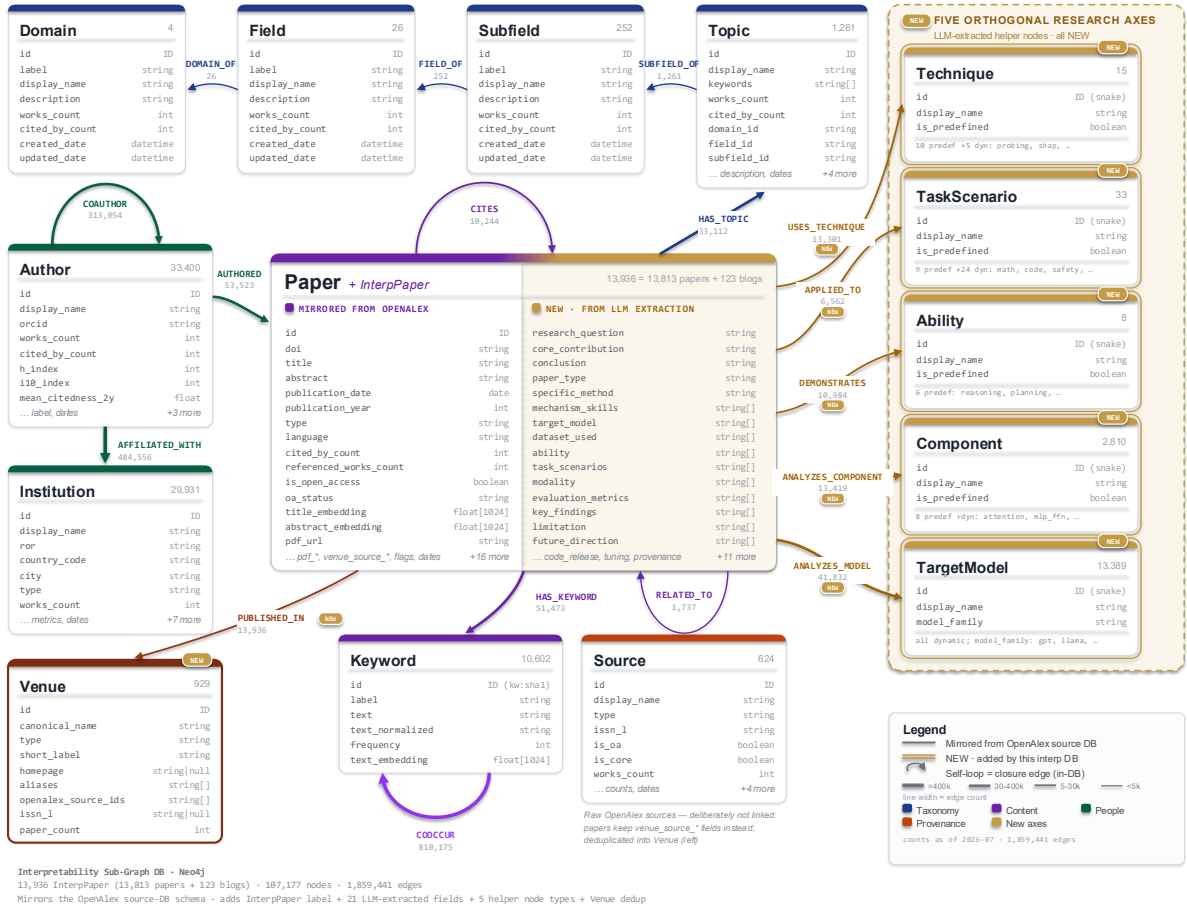


Fig. 10: The overview of our knowledge graph of AI mechanisms.

graph. Given a research question, Mechanist retrieves potentially transferable concepts and findings across disciplines and translates them into candidate hypotheses and executable experiments for AI systems. For example, research on human belief and theory of mind motivated our distinction among World Knowledge, Personal Belief and Attributed Belief in LLMs, which subsequently guided the behavioural experiments and causal mechanism analysis in our belief study (Fig. 4).

The two knowledge graphs provide complementary evidence. The mechanism knowledge graph supports fine-grained retrieval of AI mechanisms, whereas the cross-disciplinary graph broadens the search to concepts and mechanisms studied in other fields. Their outputs are integrated through the multi-source retrieval strategy described in § 5.2, enabling Mechanist to translate cross-disciplinary insights into mechanistically grounded and experimentally testable hypotheses.

5.2 Retrieval strategy

Building on the interpretability and cross-disciplinary knowledge graphs, we **design a multi-source retrieval strategy** that provides literature evidence for hypothesis generation, novelty assessment, and experimental design. Following the neuro-symbolic retrieval framework of SciAtlas [22], our strategy comprises query decomposition, multi-channel matching, graph-based expansion, and final ranking.

Query decomposition. The system first determines the disciplinary scope of the query and routes it to the interpretability graph, the cross-disciplinary graph, or both. The query is then rewritten using the terminology of each field. For the interpretability graph, the query is further aligned with five retrieval dimensions: technique, component, task scenario, ability, and target model. For example, the question “Are language and reasoning abilities separable in large language models?” is decomposed into an interpretability query on reasoning-circuit discovery and residual-stream probing, and a cross-disciplinary query on the dissociation between the language network and the multiple-demand system. For each sub-query, the system also generates a hypothetical abstract in the terminology of the corresponding

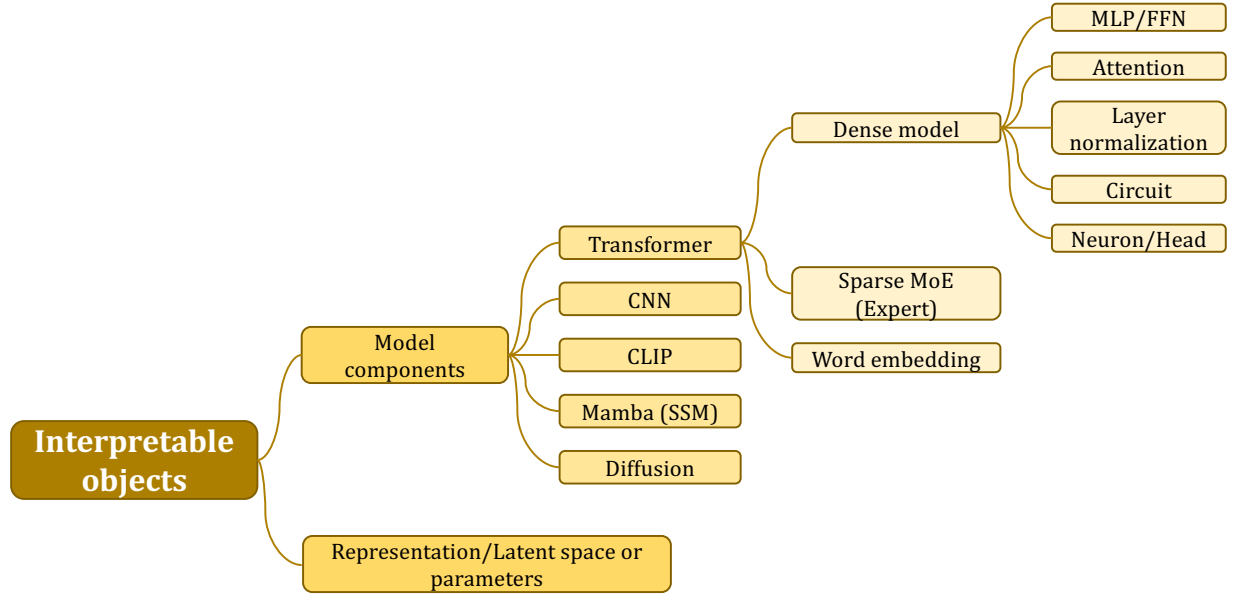


Fig. 11: Overview of our knowledge graph of AI mechanisms from the perspective of interpretable objects.

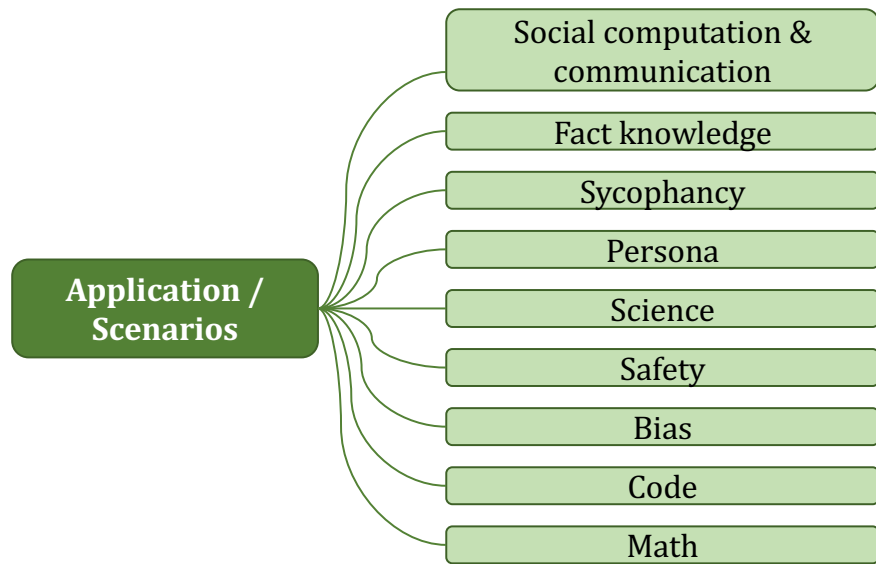


Fig. 12: Overview of our knowledge graph of AI mechanisms from the perspective of application scenarios.

field and uses its embedding as an additional semantic query (HyDE).

Multi-channel matching. The system retrieves candidate studies through three complementary channels. (1) *Keyword matching*. Salient terms extracted from each sub-query are used for BM25 retrieval over document titles and abstracts. These terms are also matched against normalized entity labels and structured attributes in the knowledge graphs, prioritizing studies that explicitly mention the relevant concepts, methods, model components, or task settings. (2) *Semantic matching*. Each sub-query and its hypothetical abstract are encoded separately and matched against dense representations of document titles and abstracts. The resulting candidate lists are combined to retrieve conceptually related studies even when they use terminology different from that of the original query. (3) *Title matching*. When the query refers to specific studies, the system extracts and normalizes their titles and performs exact and fuzzy matching against paper nodes in the graphs. High-confidence title matches are retained and assigned an additional ranking weight. **Graph-based retrieval.** Papers and concepts identified by the matching channels serve as seed nodes for graph traversal. The system performs constrained multi-hop expansion over citation links and domain-specific relations, including connections among papers, methods, model components, tasks, and findings. This step retrieves studies that may not be directly accessible through lexical or vector matching but receive strong support from the local graph topology. To limit semantic drift, the expansion depth and number of nodes are bounded, and graph-discovered papers are required to retain sufficient relevance to the original query.

Ranking. The ranked lists produced by the matching channels are first combined using reciprocal rank fusion (RRF). The fused candidates are then re-ranked using their initial relevance, graph support, citation impact, publication recency, and title-match signals. Results from the two knowledge graphs and external search services are finally deduplicated and merged into a single literature list for downstream use.

5.3 Library of mechanism methods

A challenge in mechanistic research is determining which method to use in a given setting, how to apply it correctly, and how to adapt the analysis when the selected method fails. To address this challenge, we organize mechanistic interpretability methods into eleven families (Fig. 13). For each family, we summarize its strengths, limitations, and suitable application scenarios, enabling Mechanist to select appropriate methods during experiment design and execution. These families are implemented as modular mechanism skills that provide structured guidance for method selection, use, and adjustment. The library contains 32 foundational mechanistic analysis methods. Each method is accompanied by executable code and usage instructions that specify its inputs, parameters, outputs, and evaluation procedure. The instructions also describe common failure modes and possible adjustments, allowing Mechanist to revise the method, modify its configuration, or select an alternative when the initial analysis is unsuccessful. Together, these mechanism skills provide Mechanist with practical support for selecting, implementing, and refining mechanistic experiments.

Vocabulary projection. Vocabulary projection applies the pretrained unembedding matrix to an internal representation and maps the resulting logits onto the model vocabulary. This family includes projection of residual-stream states using the Tuned Lens [78], attention-head outputs using LogitLens4LLMs [79], and neuron value weights following the key-value memory interpretation of feed-forward networks [80]. These methods require neither additional training nor labeled data and therefore provide a low-cost readout of information that is linearly accessible from the analysed representation. Their reliability, however, depends on the alignment between that representation and the model’s output space.

Magnitude analysis. Magnitude analysis ranks components using static or dynamic quantities, including parameter norms, activation statistics, and distances between layer representations. In our library, the corresponding methods analyse massive values in static parameters [81], identify behaviour-selective dynamic components such as language-specific neurons [82], and compare layer-wise representations, as instantiated by TruthX [83]. These methods require no backward pass or auxiliary training and are therefore suitable for initial screening. Their rankings remain correlational: a large magnitude does not establish that a component is necessary for the behaviour.

Representation and parameter analysis. Representation analysis encodes a concept as a direction in activation space, whereas parameter analysis represents a learned behaviour through the difference between fine-tuned and pretrained parameters. This family comprises representation engineering [84], steering vectors constructed through contrastive activation addition [74], feature-level steering [85], and parameter-space task vectors [86]. These methods test whether a direction is sufficient to alter a behaviour and can support training-free steering or controlled weight editing. Such sufficiency claims still require validation against an appropriate counterfactual or baseline.

Probing. Probing fits an auxiliary, often linear, predictor to decode a labeled property from an internal vector while keeping the target model frozen. The library supports probes over residual-stream states, including intrinsic

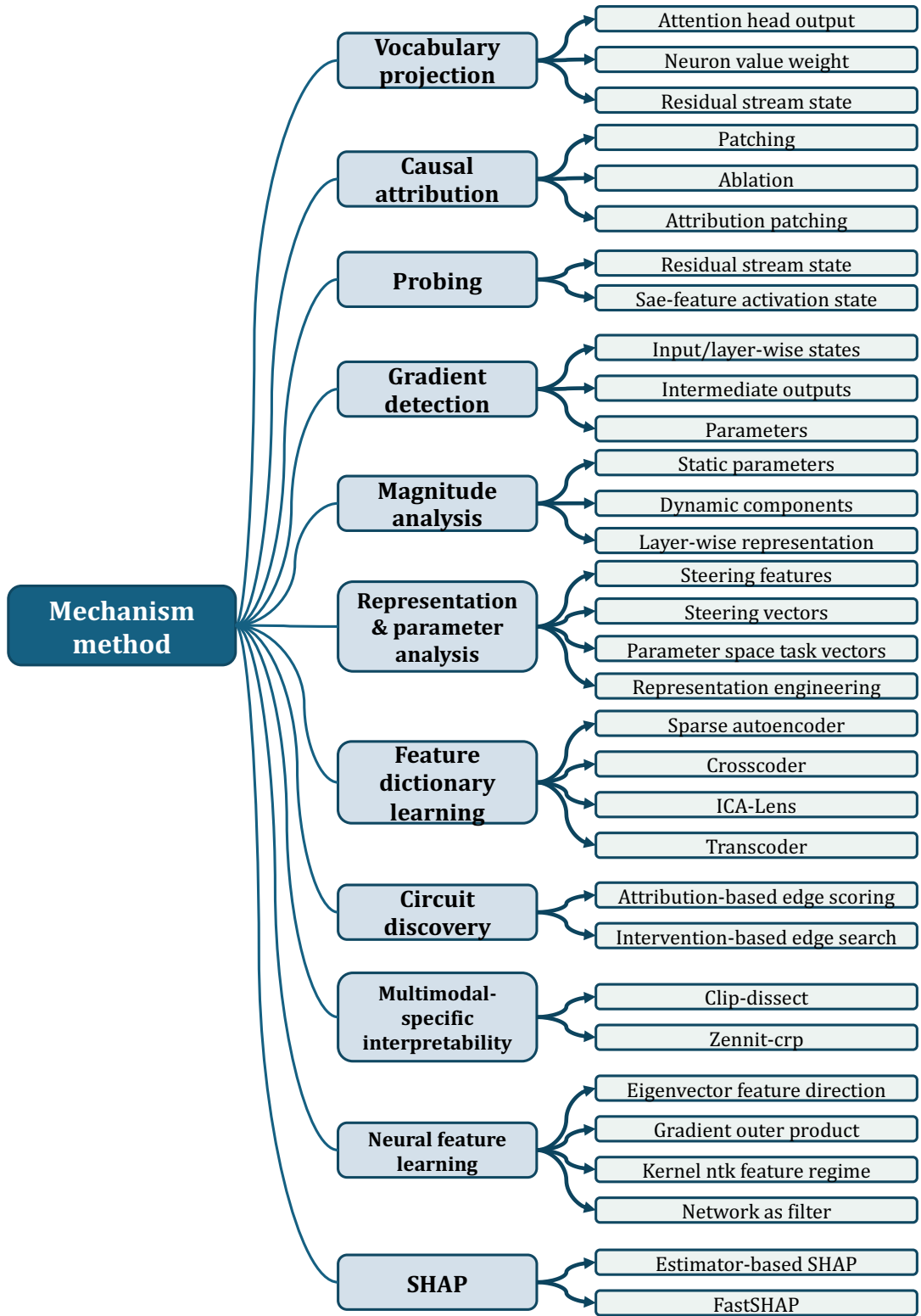


Fig. 13: Mechanism methods for large language models and multi-modal models.

525 representations of hallucinations [87], and probes over SAE (sparse autoencoder) feature activations, including analyses
526 of feature splitting and absorption [88]. Under a shared protocol, these methods support comparisons across layers and
527 representation types. Decodability measures recoverability within the probe’s hypothesis class and should therefore be
528 distinguished from causal importance; probe findings are evaluated with subsequent interventions.

529 **Feature dictionary learning.** Feature dictionary learning approximates a dense activation as a sparse weighted
530 combination of learned dictionary atoms, with each atom intended to capture a more specific and interpretable feature.
531 This family includes SAEs [89, 55, 56], transcoders [90], crosscoders [91], and ICA (independent component analysis)
532 lenses [92]. Training a dictionary is relatively expensive, but evaluating the resulting features is inexpensive and supports
533 feature dashboards, attribution graphs, and feature steering.

534 **Gradient detection.** Gradient detection ranks internal objects by the sensitivity of a scalar target to changes in
535 those objects. The library applies this analysis to inputs and layer-wise states [93], intermediate outputs through
536 relevance patching [94], and model parameters [95]. Common scores include gradient norms, gradient–input products,
537 and integrated gradients. A small number of backward passes is sufficient for a first-order screen, but the resulting
538 rankings can miss effects that are canceled or amplified by later computation.

539 **Causal attribution.** Causal attribution intervenes on an object and measures the resulting change in behaviour.
540 Activation patching [96] replaces an activation with a value obtained from a counterfactual input; ablation [97] removes
541 or zeros the object; and attribution patching [98] approximates the effect of patching with a first-order expansion. These
542 interventions test necessity or sufficiency directly, subject to the choice of counterfactual and control. Exact interventions
543 can be expensive, so candidate sets are often reduced first with magnitude or gradient-based screens.

544 **Circuit discovery.** Circuit discovery searches for a small set of edges between attention heads and MLP (multilayer per-
545 ceptron) blocks whose joint presence reproduces a target behaviour. It can use ACDC (Automatic Circuit Discovery) [99],
546 which iteratively prunes edges by intervention, or EAP-IG (Edge Attribution Patching with Integrated Gradients) [100],
547 which assigns gradient-based edge scores; sparse-feature replacement models provide another route. The output is a
548 structured subgraph rather than an isolated component ranking, and its faithfulness is assessed on held-out inputs and,
549 where appropriate, across models.

550 **SHAP (Shapley additive explanations).** For input features viewed as players in a cooperative game, SHAP assigns
551 each feature a Shapley value [101]. The value is characterized by the local-accuracy, missingness, and consistency
552 axioms under the corresponding attribution setup. TreeSHAP [102] provides an exact estimator for tree ensembles,
553 whereas KernelSHAP uses sampling for arbitrary models and FastSHAP [103] learns an amortized surrogate. These
554 estimators provide input-level evidence that complements the internal-mechanism families, but their computational cost
555 and approximation error should be reported with the attribution results.

556 **Neural feature learning.** Neural feature learning characterizes the directions emphasized by a trained network using
557 two related quantities. The NFM (Neural Feature Matrix) is the Gram matrix of a layer’s weights, whereas the EGOP
558 (expected gradient outer product) averages the outer products of input gradients over the analysis distribution. The
559 DNFA (Deep Neural Feature Ansatz) concerns the empirical alignment between the directions identified by these two
560 quantities. This family includes eigenvector-based feature directions and recursive gradient-outer-product estimators [73],
561 kernel and NTK (neural tangent kernel) feature regimes [104], and convolutional network-as-filter variants [105]. Together,
562 these methods relate finite-network feature learning to kernel descriptions and infinite-width dynamics.

563 **Multimodal-specific interpretability.** This family assigns natural-language descriptions to internal vision units
564 by comparing activation summaries with text embeddings from a vision–language model such as CLIP (Contrastive
565 Language–Image Pre-training). CLIP-dissect produces concept-set descriptions of individual units, whereas Zennit-crp
566 (concept relevance propagation) [106] derives concept-level explanations from relevance propagation. Both approaches
567 ground component-level attribution results in visual concepts, while remaining dependent on the representation model
568 and the selected concept vocabulary.

569 Data availability

570 All datasets used in this work, the user input for each result, the experimental logs, and all evaluations are publicly
571 available at https://github.com/mengrusun/Mechanist_data.

572 Code availability

573 The source code of this work is freely available on GitHub at <https://github.com/zjunlp/Mechanist>.

574 Models and versions

575 Claude Opus 4.7 (<https://www.anthropic.com/claude-opus-4-7-system-card>), Claude Opus 4.8 (<https://www.anthropic.com/claude-opus-4-8-system-card>), Claude Opus 5 (<https://www.anthropic.com/claude-opus-5-system-card>), GPT-4o, GPT-4.1 (<https://openai.com/index/gpt-4-1/>), GPT-5.4 (<https://openai.com/index/gpt-5-4-thinking-system-card/>), GPT-5.6-sol (<https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf>) and DeepSeek-V3.2-Thinking (<https://arxiv.org/abs/2512.02556>) are used
578 through their public APIs.
580

581 Acknowledgements

582 We thank Weilin Zhou and Linsong Yu for their efforts in reproducing and evaluating AI-Scientist. We also thank
583 Wentao Chen for investigating DNA sequence generation with the Evo2-7B model and for his research using Centaur [3].

584 Competing interests

585 The authors declare no competing interests.

References

- [1] Eric Horvitz and Robert West. A narrowing window to understand ai. *Science*, 392(6802):1003–1003, 2026.
- [2] Ashwini Ashokkumar, Luke Hewitt, Isaias Ghezae, and Robb Willer. Large language models can predict the results of social science experiments. *Nature*, pages 1–8, 2026.
- [3] Marcel Binz, Elif Akata, Matthias Bethge, Franziska Brändle, Fred Callaway, Julian Coda-Forno, Peter Dayan, Can Demircan, Maria K. Eckstein, Noémi Élteto, Thomas L. Griffiths, Susanne Haridi, Akshay K. Jagadish, Ji-An Li, Alexander Kipnis, Sreejan Kumar, Tobias Ludwig, Marvin Mathony, Marcelo G. Mattar, Alireza Modirshanechi, Surabhi S. Nath, Joshua C. Peterson, Milena Rmus, Evan M. Russek, Tankred Saanum, Johannes A. Schubert, Luca M. Schulze Buschoff, Nishad Singhi, Xin Sui, Mirko Thalmann, Fabian J. Theis, Vuong Truong, Vishaal Udandaraao, Konstantinos Voudouris, Robert C. Wilson, Kristin Witte, Shuchen Wu, Dirk U. Wulff, Huadong Xiong, and Eric Schulz. A foundation model to predict and capture human cognition. *Nat.*, 644(8078):1002–1009, 2025.
- [4] David Klindt, Charles O’Neill, Patrik Reizinger, Harald Maurer, and Nina Miolane. A unifying framework from neural superposition to sparse interpretable codes. *Nature Machine Intelligence*, pages 1–13, 2026.
- [5] Leonard Bereska and Efstratios Gavves. Mechanistic interpretability for ai safety—a review. *arXiv preprint arXiv:2404.14082*, 2024.
- [6] Xinlei Yu, Zhangquan Chen, Yongbo He, Tianyu Fu, Cheng Yang, Chengming Xu, Yue Ma, Xiaobin Hu, Zhe Cao, Jie Xu, Guibin Zhang, Jiale Tao, Jiayi Zhang, Siyuan Ma, Kaituo Feng, Haojie Huang, Youxing Li, Ronghao Chen, Huacan Wang, Chenglin Wu, Zikun Su, Xiaogang Xu, Kelu Yao, Kun Wang, Chen Gao, Yue Liao, Ruqi Huang, Tao Jin, Cheng Tan, Jiangning Zhang, Wenqi Ren, Yanwei Fu, Yong Liu, Yu Wang, Xiangyu Yue, Yu-Gang Jiang, and Shuicheng Yan. The latent space: Foundation, evolution, mechanism, ability, and outlook. *CoRR*, abs/2604.02029, 2026.
- [7] Mirac Suzgun, Tayfun Gur, Federico Bianchi, Daniel E. Ho, Thomas Icard, Dan Jurafsky, and James Zou. Language models cannot reliably distinguish belief from knowledge and fact. *Nature Machine Intelligence*, 7(11):1780–1790, November 2025.
- [8] Mingyu Jin, Qinkai Yu, Jingyuan Huang, Qingcheng Zeng, Zhenting Wang, Wenyue Hua, Haiyan Zhao, Kai Mei, Yanda Meng, Kaize Ding, et al. Exploring concept depth: How large language models acquire knowledge and concept at different layers? In *Proceedings of the 31st international conference on computational linguistics*, pages 558–573, 2025.
- [9] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nat. Mach. Intell.*, 1(5):206–215, 2019.
- [10] Jan Betley, Niels Warncke, Anna Szytber-Betley, Daniel Tan, Xuchan Bao, Martín Soto, Megha Srivastava, Nathan Labenz, and Owain Evans. Training large language models on narrow tasks can lead to broad misalignment. *Nature*, 649(8097):584–589, 2026.
- [11] Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamara Lanham, Daniel M. Ziegler, Tim Maxwell, Newton Cheng, Adam S. Jermyn, Amanda Askell, Ansh Radhakrishnan, Cem Anil, David Duvenaud, Deep Ganguli, Fazl Barez, Jack Clark, Kamal Ndousse, Kshitij Sachan, Michael Sellitto, Mrinank Sharma, Nova DasSarma, Roger B. Grosse, Shauna Kravec, Yuntao Bai, Zachary Witten, Marina Favaro, Jan Brauner, Holden Karnofsky, Paul F. Christiano, Samuel R. Bowman, Logan Graham, Jared Kaplan, Sören Mindermann, Ryan Greenblatt, Buck Shlegeris, Nicholas Schiefer, and Ethan Perez. Sleeper agents: Training deceptive llms that persist through safety training. *CoRR*, abs/2401.05566, 2024.
- [12] Mingyu Jin, Yutong Yin, Jingcheng Niu, Qingcheng Zeng, Wujiang Xu, Mengnan Du, Wei Cheng, Zhaoran Wang, Tianlong Chen, and Dimitris N Metaxas. Farther the shift, sparser the representation: Analyzing ood mechanisms in llms. *arXiv preprint arXiv:2603.03415*, 2026.

- [13] Xi Chen, Mingyu Jin, Jingcheng Niu, Yutong Yin, Jinman Zhao, Bangwei Guo, Dimitris N Metaxas, Zhaoran Wang, Yutao Yue, and Gerald Penn. All circuits lead to rome: rethinking functional anisotropy in circuit and sheaf discovery for llms. *arXiv preprint arXiv:2605.12671*, 2026.
- [14] Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, Xiaowei Zhu, Jun Zhou, and Huajun Chen. A comprehensive study of knowledge editing for large language models. *CoRR*, abs/2401.01286, 2024.
- [15] Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of ai research. *Nature*, 651(8107):914–919, 2026.
- [16] Ali Essam Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J Szostkiewicz, Dmytro Shved, Gavin J Gyimesi, Jon M Laurent, Samantha M Wright, Muhammed T Razzak, et al. A multi-agent system for automating scientific discovery. *Nature*, pages 1–3, 2026.
- [17] Zexi Liu, Yuzhu Cai, Xinyu Zhu, Yujie Zheng, Runkun Chen, Ying Wen, Yanfeng Wang, Weinan E, and Siheng Chen. Ml-master: Towards ai-for-ai via integration of exploration and reasoning. *CoRR*, abs/2506.16499, 2025.
- [18] Jiaojiao Han, Wujiang Xu, Mingyu Jin, and Mengnan Du. SAGE: an agentic explainer framework for interpreting SAE features in language models. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2026 - Volume 5: Industry Track, Rabat, Morocco, March 25-27, 2026*, pages 483–495. Association for Computational Linguistics, 2026.
- [19] Alex Foote, Neel Nanda, Esben Kran, Ioannis Konstas, Shay Cohen, and Fazl Barez. Neuron to graph: Interpreting language model neurons at scale. *arXiv preprint arXiv:2305.19911*, 2023.
- [20] Justin Lee, Tuomas Oikarinen, Arjun Chatha, Keng-Chi Chang, Yilan Chen, and Tsui-Wei Weng. The importance of prompt tuning for automated neuron explanations. *arXiv preprint arXiv:2310.06200*, 2023.
- [21] Arnau Marin-Llobet and Javier Ferrando. Automated interpretability and feature discovery in language models with agents. *CoRR*, abs/2605.01555, 2026.
- [22] Shuofei Qiao, Yunxiang Wei, Jiazheng Fan, Bin Wu, Busheng Zhang, Mengru Wang, Yuqi Zhu, Ningyu Zhang, Keyan Ding, Qiang Zhang, et al. Sciatlas: A large-scale knowledge graph for automated scientific research. *arXiv preprint arXiv:2605.22878*, 2026.
- [23] Anthropic. Claude code. <https://docs.anthropic.com/en/docs/claude-code/getting-started>, 2025. Accessed: 2026-08-08.
- [24] Alex Cloud, Minh Le, James Chua, Jan Betley, Anna Sztzyber-Betley, Sören Mindermann, Jacob Hilton, Samuel Marks, and Owain Evans. Language models transmit behavioural traits through hidden signals in data. *Nature*, 652(8110):615–621, 2026.
- [25] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.
- [26] Simon Schrodi, Elias Kempf, Fazl Barez, and Thomas Brox. Towards understanding subliminal learning: When and how hidden biases transfer. *CoRR*, abs/2509.23886, 2025.
- [27] Amir Zur, Zhuofan Ying, Alexander Russell Loftus, Kerem Şahin, Steven Yu, Lucia Quirke, Tamar Rott Shaham, Natalie Shapira, Hadas Orgad, and David Bau. Token entanglement in subliminal learning. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025.
- [28] Vincent C. Brockers, Roman D. Ventzke, Valentin Neuhaus, Belén Hidalgo-Ogalde, and Viola Priesemann. Learning through noise: Why subliminal learning works and when it fails. *CoRR*, abs/2605.23645, 2026.
- [29] Camila Blank, Agam Bhatia, Senthoooran Rajamanoharan, Arthur Conmy, and Neel Nanda. Subliminal learning is steering vector distillation. *CoRR*, abs/2606.00995, 2026.

- [30] Tamas Madl. Channel location constrains the auditability of subliminal learning. *CoRR*, abs/2606.22019, 2026.
- [31] Daniel Tan, Anders Woodruff, Niels Warncke, Arun Jose, Maxime Riché, David Demitri Africa, and Mia Taylor. Inoculation prompting: Eliciting traits from llms during training can suppress them at test-time. *CoRR*, abs/2510.04340, 2025.
- [32] Nevan Wichers, Aram Ebtekar, Ariana Azarbal, Victor Gillioz, Christine Ye, Emil Ryd, Neil Rath, Henry Sleight, Alex Mallen, Fabien Roger, and Samuel Marks. Inoculation prompting: Instructing llms to misbehave at train-time improves test-time alignment. *CoRR*, abs/2510.05024, 2025.
- [33] Jan Dubinski, Jan Betley, Anna Sztzyber-Betley, Daniel Tan, and Owain Evans. Conditional misalignment: common interventions can hide emergent misalignment behind contextual triggers. *CoRR*, abs/2604.25891, 2026.
- [34] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Riviére, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.
- [35] Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng-ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, et al. Qwen-image technical report. *arXiv preprint arXiv:2508.02324*, 2025.
- [36] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- [37] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- [38] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR, 2023.
- [39] Dirk Groeneveld, Iz Beltagy, Evan Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, Shane Arora, David Atkinson, Russell Authur, Khyathi Raghavi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, Tushar Khot, William Merrill, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Valentina Pyatkin, Abhilasha Ravichander, Dustin Schwenk, Saurabh Shah, Will Smith, Emma Strubell, Nishant Subramani, Mitchell Wortsman, Pradeep Dasigi, Nathan Lambert, Kyle Richardson, Luke Zettlemoyer, Jesse Dodge, Kyle Lo, Luca Soldaini, Noah A. Smith, and Hannaneh Hajishirzi. Olmo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15789–15809. Association for Computational Linguistics, 2024.
- [40] Yuheng Wu, Wentao Guo, Zirui Liu, Heng Ji, Zhaozhuo Xu, and Denghui Zhang. How large language models encode theory-of-mind: a study on sparse parameter patterns. *npj Artificial Intelligence*, 1(1):20, 2025.
- [41] Feride Nur Haskaraca, Marina Proft, Ulf Liszkowski, and Hannes Rakoczy. How robust are egocentric and altercentric interference effects in social cognition? a test with explicit and implicit versions of a continuous false belief task. *Frontiers in psychology*, 14:1142302, 2023.
- [42] Susan AJ Birch and Paul Bloom. The curse of knowledge in reasoning about false beliefs. *Psychological science*, 18(5):382–386, 2007.
- [43] Hugo Dalla-Torre, Liam Gonzalez, Javier Mendoza-Revilla, Nicolas Lopez Carranza, Adam Henryk Grzywaczewski, Francesco Oteri, Christian Dallago, Evan Trop, Bernardo P De Almeida, Hassan Sirelkhatim, et al. Nucleotide transformer: building and evaluating robust foundation models for human genomics. *Nature Methods*, 22(2):287–297, 2025.

- [44] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.
- [45] Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos Jr, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. Large language models generate functional protein sequences across diverse families. *Nature biotechnology*, 41(8):1099–1106, 2023.
- [46] Jinho Chang and Jong Chul Ye. Bidirectional generation of structure and properties through a single molecular foundation model. *Nature Communications*, 15(1):2323, 2024.
- [47] Eric Nguyen, Michael Poli, Matthew G Durrant, Brian Kang, Dhruva Katrekar, David B Li, Liam J Bartie, Armin W Thomas, Samuel H King, Garyk Brixi, et al. Sequence modeling and design from molecular to genome scale with evo. *Science*, 386(6723):eado9336, 2024.
- [48] Jerret Ross, Brian Belgodere, Samuel C Hoffman, Vijil Chenthamarakshan, Jiri Navratil, Youssef Mroueh, and Payel Das. Gp-molformer: A foundation model for molecular generation. *Digital Discovery*, 4(10):2684–2696, 2025.
- [49] Garyk Brixi, Matthew G Durrant, Jerome Ku, Mohsen Naghipourfar, Michael Poli, Gwanggyu Sun, Greg Brockman, Daniel Chang, Alison Fanton, Gabriel A Gonzalez, et al. Genome modelling and design across all domains of life with evo 2. *Nature*, 652(8112):1349–1361, 2026.
- [50] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 620(7976):1089–1100, 2023.
- [51] Brian L Hie, Varun R Shanker, Duo Xu, Theodora UJ Bruun, Payton A Weidenbacher, Shaogeng Tang, Wesley Wu, John E Pak, and Peter S Kim. Efficient evolution of human antibodies from general protein language models. *Nature biotechnology*, 42(2):275–283, 2024.
- [52] Jan Zrimec, Xiaozhi Fu, Azam Sheikh Muhammad, Christos Skrekas, Vyintas Jauniskis, Nora K Speicher, Christoph S Börlin, Vilhelm Verendel, Morteza Haghir Chehreghani, Devdatt Dubhashi, et al. Controlling gene expression with deep generative design of regulatory dna. *Nature communications*, 13(1):5099, 2022.
- [53] Lucas Ferreira DaSilva, Simon Senan, Judith F Kribelbauer-Swietek, Zain Munir Patel, Lithin Karmel Louis, Aniketh Janardhan Reddy, Sameer Gabbita, Jonathan D Rosen, Zach Nussbaum, César Miguel Valdez Córdova, et al. Designing synthetic regulatory elements using the generative ai framework dna-diffusion. *Nature Genetics*, 58(1):180–194, 2026.
- [54] Liv Gorton, Nicholas Wang, Nam Nguyen, Myra Deng, Eric Ho, Daniel Balsam, and Thomas McGrath. Interpreting Evo 2: Arc Institute’s next-generation genomic foundation model, 2025. Zenodo.
- [55] Tom Lieberum, Senthoran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca D. Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *CoRR*, abs/2408.05147, 2024.
- [56] Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *CoRR*, abs/2410.20526, 2024.
- [57] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, et al. Accelerating scientific discovery with co-scientist. *Nature*, pages 1–3, 2026.
- [58] Zhantao Chen, Alexander N Petsch, Aidan J Israelski, Rajan Plumley, Lingjia Shen, Cong Wang, Cheng Peng, Yuan Ni, Arun Bansil, Sugata Chowdhury, et al. An agentic artificially intelligent x-ray scientist. *Nature Machine Intelligence*, pages 1–12, 2026.

- [59] Kexin Huang, Serena Zhang, Hanchen Wang, Yuanhao Qu, Yingzhou Lu, Ryan Li, Yusuf Roohani, Lin Qiu, Shiyi Cao, Gavin Li, et al. Autonomous biomedical research with an artificial intelligence agent. *Science*, page ead4351, 2026.
- [60] Goodfire. Silico: The platform for ambitious AI research. <https://www.goodfire.com/silico>, 2026.
- [61] Vighnesh Subramaniam, Yilun Du, Joshua B. Tenenbaum, Antonio Torralba, Shuang Li, and Igor Mordatch. Multiagent finetuning: Self improvement with diverse reasoning chains. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- [62] Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V Le, Denny Zhou, Swaroop Mishra, Huaixiu S Zheng, et al. Self-discover: Large language models self-compose reasoning structures. *Advances in Neural Information Processing Systems*, 37:126032–126058, 2024.
- [63] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason E. Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 11537–11554. Association for Computational Linguistics, 2025.
- [64] Maxime Robeyns, Martin Szummer, and Laurence Aitchison. A self-improving coding agent. *CoRR*, abs/2504.15228, 2025.
- [65] Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, Bingnan Zheng, Bang Liu, Yuyu Luo, and Chenglin Wu. Aflow: Automating agentic workflow generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- [66] Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. Mle-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.
- [67] Han Zhou, Xingchen Wan, Ruoxi Sun, Hamid Palangi, Shariq Iqbal, Ivan Vulic, Anna Korhonen, and Sercan Ö. Arik. Multi-agent design: Optimizing agents with better prompts and topologies. *CoRR*, abs/2502.02533, 2025.
- [68] Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Samuel Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models. *CoRR*, abs/2412.14093, 2024.
- [69] Akshay K Jagadish, Younes Strittmatter, Nori Jacoby, George Kachergis, Eric Schulz, Nathaniel Daw, Suyog H Chandramouli, and Thomas L Griffiths. Closing the loop to discover psychological theories with an automated cognitive scientist. *arXiv preprint arXiv:2606.26448*, 2026.
- [70] Adam Davies and Ashkan Khakzar. The cognitive revolution in interpretability: From explaining behavior to interpreting representations and algorithms. *CoRR*, abs/2408.05859, 2024.
- [71] Qianyu Hao, Fengli Xu, Yong Li, and James Evans. Artificial intelligence tools expand scientists’ impact but contract science’s focus. *Nature*, 649(8099):1237–1243, 2026.
- [72] A Singh. Specter2: Adapting scientific document embeddings to multiple fields and task formats, 2023.
- [73] Adityanarayanan Radhakrishnan, Daniel Beaglehole, Parthe Pandit, and Mikhail Belkin. Mechanism for feature learning in neural networks and backpropagation-free machine learning models. *Science*, 383(6690), 2024.
- [74] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 15504–15522. Association for Computational Linguistics, 2024.

- [75] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers: Original Research on Biomolecules*, 22(12):2577–2637, 1983.
- [76] Jason Priem, Heather A. Piwowar, and Richard Orr. Openalex: A fully-open index of scholarly works, authors, venues, institutions, and concepts. *CoRR*, abs/2205.01833, 2022.
- [77] Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025.
- [78] Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *CoRR*, abs/2303.08112, 2023.
- [79] Zhenyu Wang. Logitlens4llms: Extending logit lens analysis to modern large language models. *CoRR*, abs/2503.11667, 2025.
- [80] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 5484–5495. Association for Computational Linguistics, 2021.
- [81] Mingyu Jin, Kai Mei, Wujiang Xu, Mingjie Sun, Ruixiang Tang, Mengnan Du, Zirui Liu, and Yongfeng Zhang. Massive values in self-attention modules are the key to contextual knowledge understanding. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025.
- [82] Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 5701–5715. Association for Computational Linguistics, 2024.
- [83] Shaolei Zhang, Tian Yu, and Yang Feng. Truthx: Alleviating hallucinations by editing large language models in truthful space. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 8908–8949. Association for Computational Linguistics, 2024.
- [84] Andy Zou, Long Phan, Sarah Li Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *CoRR*, abs/2310.01405, 2023.
- [85] Dana Arad, Aaron Mueller, and Yonatan Belinkov. Saes are good for steering - if you select the right features. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 10241–10259. Association for Computational Linguistics, 2025.
- [86] Gabriel Ilharco, Marco Túlio Ribeiro, Mitchell Wortsman, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.
- [87] Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llm know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025.

- [88] David Chanin, James Wilken-Smith, Tomás Dulka, Hardik Bhatnagar, Satvik Golechha, and Joseph Bloom. A is for absorption: Studying feature splitting and absorption in sparse autoencoders. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.
- [89] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.
- [90] Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.
- [91] Julian Minder, Clément Dumas, Caden Juang, Bilal Chughtai, and Neel Nanda. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla, editors, *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025.
- [92] Sida Liu and Feijiang Han. ICA lens: Interpreting language models without training another dictionary. *CoRR*, abs/2606.11722, 2026.
- [93] Ming Li, Yanhong Li, and Tianyi Zhou. What happened in llms layers when trained for fast vs. slow thinking: A gradient perspective. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 32017–32154. Association for Computational Linguistics, 2025.
- [94] Farnoush Rezaei Jafari, Oliver Eberle, Ashkan Khakzar, and Neel Nanda. Relp: Faithful and efficient circuit discovery via relevance patching. *CoRR*, abs/2508.21258, 2025.
- [95] Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. Unveiling linguistic regions in large language models. 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), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 6228–6247. Association for Computational Linguistics, 2024.
- [96] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.
- [97] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. Dissecting recall of factual associations in auto-regressive language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12216–12235. Association for Computational Linguistics, 2023.
- [98] Aaqib Syed, Can Rager, and Arthur Conmy. Attribution patching outperforms automated circuit discovery. *CoRR*, abs/2310.10348, 2023.
- [99] Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

- 898 [100] Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. Have faith in faithfulness: Going beyond circuit overlap
899 when finding model mechanisms. *CoRR*, abs/2403.17806, 2024.
- 900 [101] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In Isabelle Guyon, Ulrike
901 von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors,
902 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing*
903 *Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4765–4774, 2017.
- 904 [102] Scott M. Lundberg, Gabriel G. Erion, and Su-In Lee. Consistent individualized feature attribution for tree
905 ensembles. *CoRR*, abs/1802.03888, 2018.
- 906 [103] Neil Jethani, Mukund Sudarshan, Ian Connick Covert, Su-In Lee, and Rajesh Ranganath. Fastshap: Real-time
907 shapley value estimation. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual*
908 *Event, April 25-29, 2022*. OpenReview.net, 2022.
- 909 [104] Greg Yang and Edward J. Hu. Feature learning in infinite-width neural networks, 2022.
- 910 [105] Daniel Beaglehole, Adityanarayanan Radhakrishnan, Parthe Pandit, and Mikhail Belkin. Mechanism of feature
911 learning in convolutional neural networks. *CoRR*, abs/2309.00570, 2023.
- 912 [106] Reduan Achibat, Maximilian Dreyer, Ilona Eisenbraun, Sebastian Bosse, Thomas Wiegand, Wojciech Samek, and
913 Sebastian Lapuschkin. From attribution maps to human-understandable explanations through concept relevance
914 propagation. *Nat. Mac. Intell.*, 5(9):1006–1019, 2023.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SupplementaryInformation.pdf](#)