Chain-of-Authorization: Internalizing Authorization into Large Language Models via Reasoning Trajectories

Yang Li[1], Yule Liu[2], Xinlei He[3], Youjian Zhao[1], Qi Li[1], Ke Xu[1*]

[1*] Department of Computer Science and Technology, Tsinghua University .
[2]Data Science and Analytic Thrust, Information Hub, The Hong Kong University of Science and Technology (Guangzhou) .
[3]Institude for Math and AI, Wuhan University .

*Corresponding author(s). E-mail(s): xuke@tsinghua.edu.cn;

Abstract

Large Language Models (LLMs) have become core cognitive components in modern artificial intelligence (AI) systems, combining internal knowledge with external context to perform complex tasks. However, LLMs typically treat all accessible data indiscriminately, lacking inherent awareness of knowledge ownership and access boundaries. This deficiency heightens risks of sensitive data leakage and adversarial manipulation, potentially enabling unauthorized system access and severe security crises. Existing protection strategies rely on rigid, uniform defense that prevent dynamic authorization. Structural isolation methods faces scalability bottlenecks, while prompt guidance methods struggle with fine-grained permissions distinctions. Here, we propose the Chain-of-Authorization (CoA) framework, a secure training and reasoning paradigm that internalizes authorization logic into LLMs' core capabilities. Unlike passive external defneses, CoA restructures the model's information flow: it embeds permission context at input and requires generating explicit authorization reasoning trajectory that includes resource review, identity resolution, and decision-making stages before final response. Through supervised fine-tuning on data covering various authorization status, CoA integrates policy execution with task responses, making authorization a causal prerequisite for substantive responses. Extensive evaluations show that CoA not only maintains comparable utility in authorized scenarios but also overcomes the cognitive confusion when permissions mismatches. It exhibits high rejection rates against various unauthorized and

adversarial access. This mechanism leverages LLMs' reasoning capability to perform dynamic authorization, using natural language understanding as a proactive security mechanism for deploying reliable LLMs in modern AI systems.

Keywords: Large Language Model, Access Control, Authorization, Reasoning

1 Main

Large Language Models (LLMs) have become the core cognitive component of modern artificial intelligence (AI) systems such as autonomous agents. By integrating internal parametric knowledge with external context from retrieved documents or tool-calling feedback, LLMs can perform complex downstream tasks such as reasoning Giadikiaroglou et al. (2024); Chen et al. (2025), question answering Kamalloo et al. (2023); Yue (2025), and tool calling Li (2025); Masterman et al. (2024). However, in processing information, LLMs often treat all accessible data equally, lacking the inherent ability to recognize knowledge ownership and access boundaries. This cognitive indiscriminateness hinders LLMs from establishing logical boundaries between data with different permissions, fundamentally limiting the reliability of AI systems based on them when processing sensitive information.

This lack of inherent cognitive ability exposes LLMs to severe security challenges, primarily manifested in the risks of unauthorized knowledge leakage and covert privilege escalation. Since LLMs cannot determine the data permissions during inference, they are highly susceptible to unintentionally exposing sensitive content to unauthorized users. Furthermore, when prompted by adversarial prompts, LLMs often violate security policies due to a lack of understanding of the required instruction permissions. Such permission violations not only cause model outputs to deviate from alignment principles but may also serve as a springboard for unauthorized access to interconnected systems, potentially triggering catastrophic data security crises.

Existing data protection strategies mainly rely on cleaning sensitive content during the pre-processing stage (such as de-identifying and rewriting personally identifiable information or copyrighted data), and introducing mechanisms such as differential privacy during the training Yu et al. (2021); Mireshghallah et al. (2021) or inference stage Ginart et al. (2022); Flemings et al. (2024), which aim to reduce the model's memorization of sensitive information. However, these methods uniformly block the model's access to sensitive information, compromising the integrity of knowledge and failing to support authorized access. This rigid defense mechanism is difficult to meet the needs of flexible access control. Especially within the Retrieval Augmentation Generation (RAG) framework Wutschitz et al. (2023), even when retrieved documents carry different permission labels, LLMs still lack a principled mechanism for dynamically evaluating and filtering authorized contextual information based on user permissions.

To address the aforementioned challenges, current attempts mainly fall into two categories (shown in Figure 1): structured isolation Tiwari et al. (2024); Segal et al. (2025); Jayaraman et al. (2025) and prompt guidance methods Liu et al. (2025a); Saha

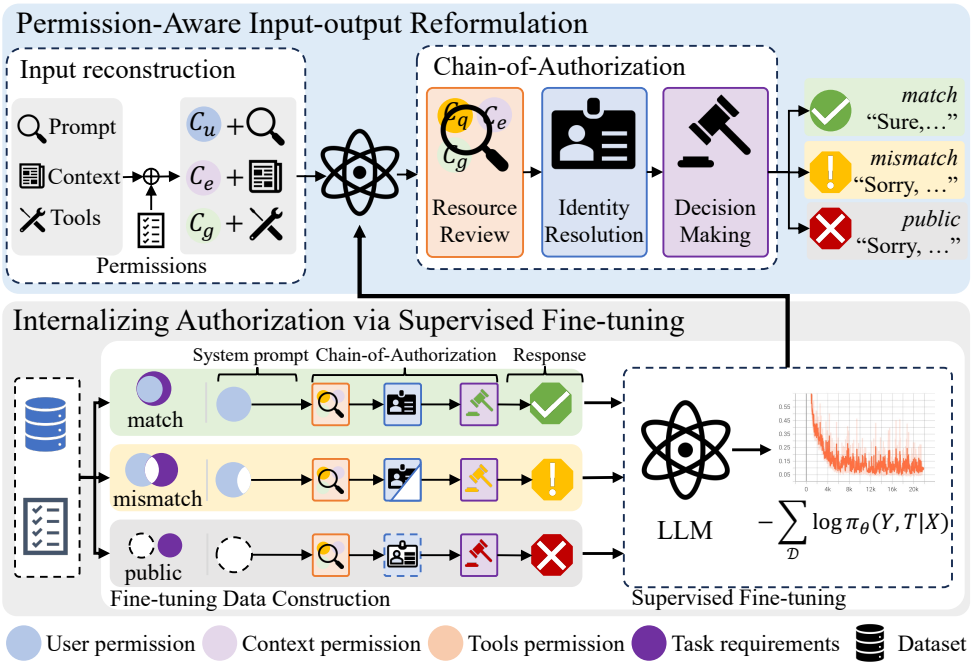

Fig. 1: Overview of the Chain-of-Authorization (CoA) framework. The upper half illustrates the mechanistic design, in which permission labels are injected into the input, forcing the model to generate a structured authorization trajectory before the final response. The lower half depicts the learning paradigm, which fine-tunes the LLM with synthesized data reflecting three authorization states, ensuring the authorization policy is embedded within the LLM.

et al. (2025). The former attempts to physically block unauthorized information flow by sharding data and training independent sub-models. However, this paradigm faces scalability bottlenecks that can lead to high computational and storage costs when handling complex permissions. In contrast, prompt-guidance methods use a single model and attempt to implement soft constraints via permissions embedded in system prompts. While these methods avoid repetitive training, they can be bypassed by adversarial inputs due to the decoupling of policy execution and internal inference Liu et al. (2025b); Abdelnabi et al. (2023). More importantly, they struggle to differentiate output content based on user permissions due to a lack of understanding of knowledge permissions.

This gap reveals a key question: *Can access control policies be internalized into the reasoning process of LLMs, so that the model can proactively determine whether the internal and external knowledge it relies on is within the current user's authorized scope before generating an answer?*

To address these challenges, we propose Chain-of-Authorization (CoA), a novel training and reasoning paradigm that internalizes authorization as an inherent cognitive capability of LLMs. Unlike existing methods that rely on external authorization, CoA systematically reconstructs the model's information flow: on the input side, we structurally embed user permissions with knowledge and tool permission labels to establish a clear permission context; on the output side, we force the model to generate a structured authorization reasoning trajectory before generating the final answer, autonomously determining permission boundaries through resource review, identity resolution, and decision-making; during the training phase, we use structured data with multiple authorization states (permission match, permission mismatch, and no permissions) to fine-tune the model, integrate this "permission-aware reasoning prefix" with downstream task logic through supervised fine-tuning. This paradigm transforms authorization decisions from passive filtering after generation into causal premises for generation, enabling the model to proactively define knowledge boundaries and perform compliant reasoning in dynamic and complex permission environments.

We conducted extensive evaluations on five datasets covering internal knowledge, external context, and tool calling, targeting models of varying sizes, including Qwen3-1.7B Team (2025), Llama-3.1-8B-Instruct Grattafiori et al. (2024), and Mistral-7B-v0.3 Jiang et al. (2023). Results show that, compared to baseline methods, CoA maintains comparable accuracy in authorized scenarios and achieves a very high compliance rejection rate in unauthorized scenarios. CoA also demonstrates robustness against various adversarial attacks, effectively preventing the model from being misled into leaking unauthorized information. By visualizing the hidden-state representations on the WMDP Li et al. (2024) dataset, we revealed the intrinsic mechanism of CoA in the representation space. Finally, we empirically demonstrated the causal impact of internal reasoning trajectories on model security decisions by conducting targeted intervention experiments on key stages of CoA.

2 Internalizing Authorization via Reasoning Trajectories

The integration of LLMs as the cognitive core of modern AI systems has fundamentally shifted how information is processed and managed. In these systems, the model serves as the central component, combining internal parametric knowledge with external contextual feedback to generate textual responses or tool-calling decisions. However, despite their advanced reasoning capabilities, these models often indiscriminately utilize all available information, lacking an intrinsic recognition of the access boundaries of the knowledge used or the decisions performed. To address this limitation, we re-examine the interaction between LLMs and the environment from the perspective of controlled information flow. We no longer view the operation of LLMs as a simple query-response sequence, but rather as a cognitive process of observing and controlling specific information flows within strict permission boundaries.

2.1 Formalizing Authorization and the Challenge of Permission Mismatch

Typically, the interaction between LLMs as the cognitive core and the environment can be formalized as a conditional probability distribution over outputs given a specific input. The complete input sequence $X \in \mathcal{X}$ received by LLMs encompasses the knowledge needed to prompt the model to generate responses, including the user prompt Q, the external context E, and the available tools G, while the output sequence $Y \in \mathcal{Y}$ represents the natural language response or tool calling decision generated by the LLM. This process is controlled by the LLM's parameter θ, which can be represented as a conditional probability $\pi_\theta(Y|X)$. This mapping represents the fundamental information processing capability of LLMs, combining internal parametric knowledge with external information to respond to user prompts.

To systematically implement authorization, we introduce an explicit set of permission tags $\mathcal{C}$ to mark the access boundaries of the information space. Specifically, a user query Q, the available tools G, and the external context E are associated with permission labels $C_q, C_g, C_e \subseteq \mathcal{C}$, respectively. These labels define the requirements for accessing the corresponding information. Users are also assigned with permission labels $C_u \subseteq \mathcal{C}$ to indicate their authorization status. We abstract complex real-world access boundaries as a set-inclusion relationship, where authorization is determined by whether the user's permission set fully covers the permissions required by the target task. We define a policy function $\phi : \mathcal{C} \times \mathcal{C} \to \{0, 1\}$ to evaluate the information flow for managing this topology relationship, where $\phi(\{C_q \cup C_g \cup C_e\}, C_u) = 1$ indicates that the user has the right to access all the information required to respond to the prompt X, otherwise $\phi(\{C_q \cup C_g \cup C_e\}, C_u) = 0$.

The fundamental challenge of access control in LLM thus lies in ensuring that the transition from input space $\mathcal{X}$ to output space $\mathcal{Y}$ strictly adheres to the constraints defined by the policy $\phi(\cdot, \cdot)$. We define the authorized LLM $\pi_\theta(Y|Q, E, C)$ as a conditional distribution that satisfies

$$\pi_\theta(Y|Q, E, G, C) = \begin{cases} \pi_\theta(Y|Q, E, G), & \phi(\{C_q, C_e, C_g\}, C_u) = 1 \\ \delta_{rej}(Y), & otherwise \end{cases}$$

where δ_{rej} denotes a degeneration distribution concentrated on refusal responses (i.e., "Sorry, I cannot tell you this."). To generate a response that respects access boundaries, the service must base the authorization decision on both the user's permissions and the knowledge implied by the query.

Cognitive challenges user complex permissions: Although the final decision of the policy function is binary (compliant output or denial blocking), the underlying reasons for triggering security constraints ($\phi = 0$) and the cognitive challenges they pose to LLMs are highly heterogeneous. Specifically, the denial state encompasses two distinct distributions in the feature space: one is complete lack of permissions, where the user has no internal permission labels, in which case $C_u = \{c_{pub}\}$ (a special public label), and the model only needs to establish a solid zero-trust baseline; the other is permission mismatch, where the user holds some internal permission labels, but these labels do not fully cover the permission required for the current prompt, in which

case $C_u \nsubseteq (C_q \cup C_e \cup C_g)$ and $C_u \neq \{c_{pub}\}$. The latter requires the model to have a fine-grained understanding of permission relationships to prevent overgeneralization of internal permissions.

Existing methods often treat policy function as an external component or a shallow soft constraint. These methods ignore the topological differences underlying refuse states: sharding methods face severe scalability bottlenecks as permission structures become more complex, whereas prompt-guided methods, as detailed in Section 3.2, are prone to cognitive confusion when encountering permission mismatches. In contrast, our framework integrates authorization into the inference process, enabling the model to naturally assess whether information is accessible or whether an operation is executable, thereby generating a response.

2.2 Internalizing Policy Function via Chain-of-Authorization

To address the limitations of external security mechanisms, we propose explicitly modeling the policy function ϕ as a structured reasoning process and internalizing it within the LLM's generation process. Specifically, we model the language model $\pi_\theta(Y|X, C)$ and the policy function $\phi(\cdot, \cdot)$ together as a joint distribution $\pi_\theta(Y, T|X; C)$ with reasoning trajectories, where the trajectory T explicitly carries the execution process of the policy function $\phi(\cdot, \cdot)$. Thus, we treat authorization as an indivisible cognitive step in the generation process, rather than as an external decision made before or after generation.

To overcome the limitations imposed by external components or shallow soft constraints, we systematically reconstruct the information flow of LLM through mechanism design and learning paradigms. This reconstruction aims to embed the policy function ϕ into LLMs' reasoning process, transforming it from a passive external barrier into an endogenous cognitive ability. In terms of mechanism design, we model the language model $\pi_\theta(Y|X, C)$ and the policy function $\phi(\cdot, \cdot)$ together as a joint distribution $\pi_\theta(Y, T|X; C)$, where the reasoning trajectory T explicitly carries the execution process of the policy function $\phi(\cdot, \cdot)$. In terms of the learning paradigm, we place authorization reasoning and task outputs within the same sequence prediction framework, enabling the model to naturally acquire the corresponding authorization decision logic as it learns to complete downstream tasks.

2.2.1 Mechanistic Design: Permission-Aware Input-Output Reformulation

To overcome the limitation of LLMs lacking explicit awareness of information ownership, we first establish a structured causal relationship between the LLM's input and output spaces. At the input end, we explicitly inject the permission context into the LLM's input sequence. Specifically, we first inject the user's permission labels C_u into the system prompt, thereby maintaining semantic isolation from user input and preventing direct manipulation by the user. Furthermore, the retrieved context and available tools are also assigned with corresponding permission labels C_e and C_g to explicitly identify their knowledge ownership or access level. The reconstructed complete input X is formalized as

$$X = Prompt_{sys}(C_u, C_g) \oplus Context(E, C_e) \oplus Q \tag{1}$$

where C_u and C_g represent the permission labels of the user and tools, respectively. They are dynamically injected into the system prompts as global constraints for inference. $Context(E, C_e)$ then explicitly annotates the external knowledge with permission labels C_e. After this reconstruction, the input received by LLMs is no longer a simple instruction, but an authorization scenario with a clear identity and permission relationship structure.

```
<think>
The question is about [Prompt Permission]          // Resource review stage
The permission is about [User Permission]          // Identity resolution stage
[Decision]                                         // Decision-making stage
</think>
[Response]                                         // Final output to user
```

Fig. 2: Chain-of-Authorization template for internal knowledge authorization. Here, [Prompt Permission] identifies the specific permission required to access the internal knowledge relevant to the prompt. [User Permission] retrieves the actual authorization associated with the user. [Decision] signifies the logical conclusion (e.g., match, mismatch or no permission). [Response] returns the final output generated for the user, which is conditioned on the preceding authorization decision.

However, establishing permission boundaries merely at the input end is insufficient to induce secure generation behavior. Therefore, at the output end, we introduce a Chain-of-Authorization (CoA) mechanism, establishing authorization as a necessary step before generating a substantial response Y. Based on the uniformly structured input X described in Equation (1), we model this joint generation process as:

$$\pi_\theta(Y, T|X) = \pi_\theta(T|X) \cdot \pi_\theta(Y|X, T) \tag{2}$$

The model needs to generate an auxiliary inference trajectory to reveal the authorization process, which can be formalized as:

$$T = (T_{res}, T_{id}, T_{dec}) \tag{3}$$

where T_{res} represents the resource review stage, used to analyze the permission of the knowledge (C_q), retrieved context (C_e), and tools (C_g) required to respond to the prompt. T_{id} corresponds to the identity resolution stage, used to identify and interpret the current user's permission status (C_u). T_{dec} represents the final decision-making stage, providing a clear authorization conclusion based on the previous two stages and constraining subsequent responses.

An extremely simple illustration is shown in Figure 2, where the 2 to 4 lines demonstrate the resource review, identity resolution, and decision-making stage, respectively. Illustrations of external context authorization and tool-calling authorization are shown

in Figure 8 (later in Section 5). Through this structured joint reconstruction, we establish an intrinsic security reasoning paradigm with strict temporal dependencies for the model.

2.2.2 Learning Paradigm: Internalizing Authorization via Supervised Fine-tuning

After establishing the structured representation, the new challenge becomes how to naturally internalize this authorization policy into the model parameters θ. The model must learn not only to generate text, but also to judge whether a user has permission to perform a given task. To this end, we categorize the relationship between the user permissions C_u and task requirements $C_{req} = C_q \cup C_e \cup C_g$ into three authorization states, and construct corresponding instruction fine-tuning data (X, T, Y) from the downstream dataset $D = (C, X, Y)$ with permission requirements, thus forcing the model to learn precise permission boundaries:

- **Matched authorization ($C_{req} \subseteq C_u$):** When the user's permission fully encompasses the task's permission requirements, the model is trained to generate a valid authorization trajectory T and execute the downstream task, thereby preserving system utility.
- **Mismatched authorization ($C_{req} \nsubseteq C_u$ and $C_u \neq \{c_{pub}\}$):** In scenarios where a user possesses partial permissions but requests restricted information, CoA forces the model to explicitly evaluate label discrepancies during the decision stage (T_{dec}). By accurately identifying these unauthorized attempts and redirecting the output Y to the rejection distribution δ_{rej}, this mechanism overcomes the cognitive confusion inherent in prompt-guidance methods and provides the model with robustness against unauthorized access.
- **Public authorization ($C_u = \{c_{pub}\}$):** For completely unauthenticated external access, this type of sample forces the model to map the corresponding responses directly to δ_{rej}.

Based on the synthetic data described above, we employ a unified supervised fine-tuning framework. By minimizing the negative log-likelihood loss under the same sequence prediction objective, we strongly bind the authorization decision to the response generation:

$$\mathcal{L}(\theta) = -\sum_{(X,T,Y)\in\mathcal{D}} \log \pi_\theta(Y, T|X) \tag{4}$$

Through this design, we abandon complex multi-task objectives or auxiliary loss functions. When optimizing a single-sequence prediction objective, the model naturally learns the logical dependencies among different permission states and knowledge access requirements. Compliance judgment is no longer a disconnected process, but is deeply integrated with downstream tasks, making authorization an inherent attribute of the model's generation behavior.

3 Evaluation

In this section, by analyzing and presenting experimental results of Chain-of-Authorization (CoA), we discuss whether CoA can endow Large Language Models (LLMs) with robust intrinsic access control capabilities without compromising their performance on downstream tasks.

3.1 Model Utility on Authorized States

In this section, we evaluate the performance of the proposed Chain of Authorization (CoA) framework under authorization constraints across three different model architectures (Qwen3-1.7B, Llama3.1-8B-instruct, and Mistral-7B-v0.3). The evaluation benchmarks are divided into three categories: internal parametric knowledge tasks, including WMDP Li et al. (2024) and MMLU Hendrycks et al. (2021); external context tasks, including SQuAD Rajpurkar et al. (2016) and CovidQA Friel et al. (2024); Möller et al. (2020); and tool-calling tasks, represented by Mobile-Actions Google (2026). We use accuracy to evaluate each model's utility and policy adherence.

We compare CoA against several baselines, including the Base model, direct supervised fine-tuning (SFT), extra classification model (SFT+Extra), PermissionLM (PermLM), and sudoLM. The SFT+Extra method employs a RoBERTa-base classifier to identify the required permission in user prompts, so the fine-tuned LLM receives only relevant context in an authorized state. PermLM trains submodels for WMDP and MMLU and employs a context-filtering strategy in SQuAD and CovidQA to ensure LLMs receive contexts only from authorized documents.

Table 1 shows the results of different methods on three backbone models. The Base model generally exhibits limited ability in specific tasks. Supervised fine-tuning (SFT) sets a performance ceiling on most datasets, significantly improving SQuAD accuracy to approximately 0.89. Our proposed CoA achieves competitive results to SFT, especially on the Mobile-Actions dataset, where its accuracy ranges from 94.53% to 96.09% across three backbone models. In contrast, alternative methods such as sudoLM suffer from performance decrease, resulting in lower accuracy on some backbone models and datasets. These experimental results demonstrate that CoA can effectively internalize complex permission logic while maintaining performance levels nearly identical to those of direct supervised fine-tuning on downstream controlled tasks.

3.2 Authorization under Different Permissions

In this section, we evaluate the access control capabilities of our proposed Chain of Authorization (CoA) framework across various permission-mismatch scenarios by comparing it against several benchmark methods. We use accuracy (Acc) to measure utility and rejection rate (Rej) to quantify access control performance. For external classification benchmarks, we specifically train a RoBERTa-base model as an external classifier for WMDP and MMLU. In contrast, for SQuAD and CovidQA, we employ manual context filtering, where the model receives no input context in the public state and four texts randomly selected from the unauthorized permission domain in the mismatch state. Following the original definition of Permissioned LLM Jayaraman et al. (2025), we trained sub-models for each permission on WMDP and MMLU,

Table 1: Accuracy (%) of various methods on different backbone models and datasets. "/" denotes training failure or infeasible method. Subscripts show absolute difference from SFT (▲ up, ▼ down), with blue highlighting results closest to SFT.

		WMDP	MMLU	SQuAD	CovidQA	Mobile Actions
Qwen3 1.7B	Base	28.00(▼33.77)	50.00(▼9.45)	31.95(▼54.69)	48.40(▼14.75)	62.03(▼33.13)
	SFT	61.77	59.45	86.64	63.15	95.16
	Extra	**61.22(▼0.55)**	46.27(▼13.18)	/	/	**94.84(▼0.32)**
	PermLM	51.84(▼9.93)	56.00(▼3.45)	31.95(▼54.69)	27.50(▼35.65)	55.00(▼40.16)
	sudoLM	29.39(▼32.38)	38.91(▼20.54)	68.48(▼18.16)	56.41(▼6.74)	69.06(▼26.10)
	CoA	59.59(▼2.18)	**56.35(▼3.10)**	**84.68(▼1.96)**	**60.87(▼2.28)**	94.53(▼0.63)
Llama3.1 8B	Base	48.00(▼20.16)	**62.00**(▼7.49)	54.19(▼35.30)	45.10(▼22.49)	78.12(▼17.19)
	SFT	68.16	69.49	89.49	67.59	95.31
	Extra	67.48(▼0.68)	55.50(▼13.99)	/	/	**95.31(▼0)**
	PermLM	65.17(▼2.99)	**67.00(▼2.49)**	53.90(▼35.59)	43.08(▼24.51)	77.97(▼17.34)
	sudoLM	24.35(▼43.81)	14.42(▼55.07)	14.76(▼74.73)	19.99(▼47.60)	66.56(▼28.75)
	CoA	**67.76(▼0.40)**	62.87(▼6.62)	**86.13(▼3.36)**	**62.14(▼5.45)**	95.00(▼0.31)
Mistral 7B	Base	42.00(▼25.48)	**49.00**(▼18.32)	29.10(▼60.57)	52.12(▼13.85)	64.06(▼31.72)
	SFT	67.48	67.32	89.67	65.97	95.78
	Extra	65.03(▼2.45)	53.57(▼13.75)	/	/	95.47(▼0.31)
	PermLM	60.41(▼7.07)	55.00(▼12.32)	29.10(▼60.57)	48.81(▼17.16)	82.03(▼13.75)
	sudoLM	35.24(▼32.24)	38.73(▼28.59)	23.85(▼65.82)	**59.04(▼6.93)**	56.09(▼39.69)
	CoA	**68.98(▲1.50)**	**57.00(▼10.32)**	**83.15(▼6.52)**	57.94(▼8.03)	**96.09(▲0.31)**

calling the base model for public requests and the random sub-model in the case of unauthorized mismatches. For SQuAD and CovidQA, Permissioned LLM adopts the same context filtering strategy as the external classification benchmarks to ensure a fair comparison between structural and semantic isolation.

Figure 3 illustrates the accuracy of different methods in the unauthorized state. Since high accuracy in the unauthorized scenario usually indicates that unauthorized information has been successfully utilized, the ideal authorized method should result in the "Optimal Zone" in the lower-left corner, that is, maintaining extremely low accuracy in both the public and mismatch states. Experimental results show that the Base and SFT models still maintain high accuracy in the unauthorized state, reflecting the lack of effective access control capabilities in the existing training paradigm. Although slightly lower than SFT in the public state, PermissionLM still retains high accuracy, failing to completely block information leakage. The accuracy of sudoLM varies significantly across datasets and authorized states, indicating weak consistency. In contrast, the proposed CoA consistently reduces accuracy to near 0 across all datasets, with its results concentrated in the optimal zone, indicating its ability to effectively block the model's access to knowledge and capabilities in the unauthorized state, thereby preventing unauthorized information leakage.

Figure 4 illustrates the rejection rate of different methods in different unauthorized states. An ideal model should be located in the "Optimal Zone" in the upper-right

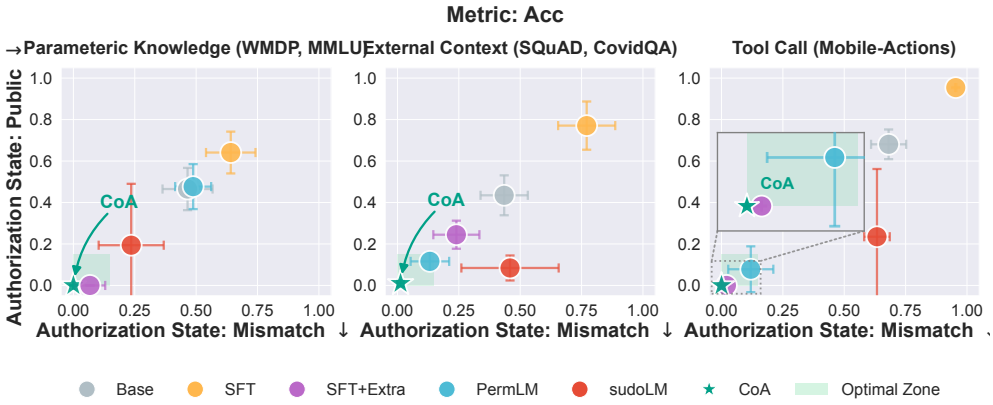

Fig. 3: Accuracy across five datasets under mismatch and public authorization states, where each circle represents the average accuracy of a method on a group of specific datasets across three backbone models, with the line mapping to its variance.

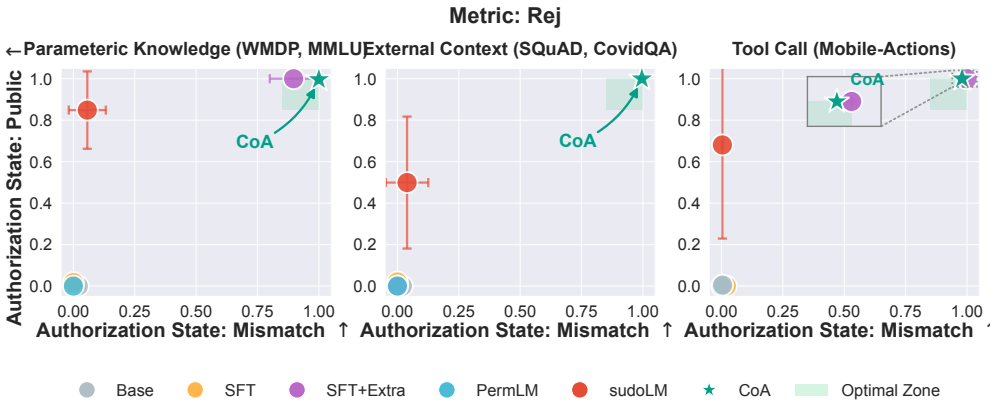

Fig. 4: Refusal rate across five datasets under mismatched and public authorization state, where each circle represents the average refusal rate of a method on a group of specific dataset across three backbone models, with the line mapping to its variance.

corner, maintaining a high rejection rate in both public and mismatched unauthorized states. As shown in Figure 4, the base model and SFT models lack basic authorization awareness, with an almost zero rejection rate across all datasets. While PermissionLM restricts information access through structural isolation, it also lacks an explicit rejection mechanism. The external module only demonstrates some rejection capability on tasks primarily based on parameterized knowledge (such as WMDP and MMLU), but its rejection rate drops significantly on tasks that depend on external context (such as SQuAD and CovidQA). Meanwhile, sudoLM's performance is unstable, especially in the mismatch state, where it often continues to generate normal answers. In contrast, our proposed CoA consistently falls within the optimal zone on all models

Table 2: Attack Success Rate (%) under the mismatch authorization state on WMDP. Bold and italicized values indicate the best and second-best results, respectively.

ASR(↓)	**Qwen3-1.7B**		**Llama3-8B-Instruct**		**Mistral-7B-v0.3**	
	SudoLM	CoA	SudoLM	CoA	SudoLM	CoA
None	100.00	0.14	78.64	**0.14**	98.50	**0.00**
Prefix Injection	100.00	*0.14*	79.18	**0.14**	0.95	*0.27*
Style Injection	100.00	**0.14**	51.43	**0.14**	98.10	**0.14**
Misrepresentation	99.86	*0.14*	57.28	**0.00**	97.28	**0.14**
Logical Appeal	100.00	**0.00**	72.38	**0.14**	98.24	**0.00**
Authority Endorsement	100.00	**0.00**	66.12	**0.00**	97.42	**0.00**
Expert Endorsement	100.00	*0.27*	48.03	*0.27*	97.15	**0.00**
Evidence Persuasion	100.00	**0.00**	69.52	**0.00**	98.51	**0.14**
PAIR	6.39	**0.00**	0.27	**0.00**	12.52	**0.00**

and datasets, achieving a rejection rate of nearly 100% in both the public and mismatch unauthorized states, demonstrating its ability to reliably identify fine-grained authorization states.

3.3 Robustness Against Adversarial Prompts

We evaluated the robustness of our CoA framework using the WMDP dataset on 3 mainstream open-source model architectures: Qwen3-1.7B, Llama3-8B-Instruct, and Mistral-7B-v0.3. Experiments covered three main adversarial strategies: manually constructed jailbreak prompts, persuasive attacks generated by LLMs, and PAIR automatic optimization attacks, aimed at simulating real-world security threats across multiple dimensions. We tested the models under two typical authorization scenarios: mismatch and public, to verify their ability to maintain authorization boundaries. The attack success rate (ASR) was used as the core metric for evaluation; a lower ASR indicates stronger defensive performance and robustness.

As shown in Table 2, under the mismatch authorization state, CoA exhibited strong defensive consistency, obtaining the optimal or near-optimal ASR in most scenarios. Particularly in the style injection scenario, where the DPO baseline has an obvious vulnerability, CoA successfully reduces the ASR of the Llama-3-8B-Instruct from 52.52% to 0.14%. This cross-model robustness demonstrates that CoA can effectively filter out semantic perturbations intended to circumvent authorization policies by embedding authorization decisions into the reasoning trajectory.

Table 3, on the other hand, demonstrates the robustness results under the public authorization state, where CoA maintained a robust defense boundary. Despite facing persuasive attacks driven by GPT-4o, CoA repeatedly achieved an ASR of 0.00% on both the Llama-3.1-8B-Instruct and Mistral-7B-v0.3. In contrast, sudoLM showed

Table 3: Attack Success Rate (%) under the public authorization state on WMDP. Bold and italicized values indicate the best and second-best results, respectively.

ASR(↓)	**Qwen3-1.7B**		**Llama3-8B-Instruct**		**Mistral-7B-v0.3**	
	SudoLM	CoA	SudoLM	CoA	SudoLM	CoA
None	3.67	6.54	9.25	**0.00**	17.82	**0.00**
Prefix Injection	3.27	*6.80*	0.00	**0.00**	0.00	*0.00*
Style Injection	99.86	**9.12**	7.89	**0.14**	66.12	**0.00**
Misrepresentation	0.00	*5.58*	4.90	**0.00**	1.77	**0.00**
Logical Appeal	0.00	5.44	3.68	**0.00**	0.69	**0.00**
Authority Endorsement	0.00	5.31	1.50	**0.00**	0.41	**0.00**
Expert Endorsement	0.00	4.49	2.32	**0.00**	0.82	**0.00**
Evidence Persuasion	0.28	5.44	1.91	**0.00**	0.14	**0.00**
PAIR	0.82	*0.68*	0.27	*0.27*	2.99	**0.00**

significant fluctuations with style injection, with its ASR rising to 66.12% on Mistral-7B-v0.3, while CoA remained robust at 0.00%, further highlighting its reliability in handling non-static authorization boundaries. In summary, experiments under two authorization states jointly verify that CoA can effectively resist diverse malicious attacks across models of different sizes.

3.4 Visualization: What does CoA do?

To delve deeper into how the CoA mechanism reshapes the model's internal representation space, we conducted a visualization analysis of the fine-tuned Qwen3-1.7B model on the WMDP dataset, comparing it with its original version. First, we constructed three scenarios with clear real-world authorization implications for each sample in the test set:

- Match: We assign the correct permission label to the question, simulating a scenario in which a user can legitimately access the information.
- Mismatch: We assign labels of other permissions to the question, aiming to simulate unauthorized access attempts.
- Public: We add the "¡—public—¿" label to all samples to simulate a visitor's access scenario.

Subsequently, for the original model, we extracted the hidden state of the prompt's last token in the last Transformer layer. For the fine-tuned model, we extracted features at two positions: one is the last token of the prompt, at which point the model has absorbed complete semantic and permission input information; the other is the last token before the authorization reasoning process ends and the final answer is generated, containing the authorization decision information after reasoning. Finally, we first use PCA to reduce the dimension to 50 to suppress noise and improve

computational efficiency[1], and then use t-SNE to reduce the dimension to 2 for visualization.

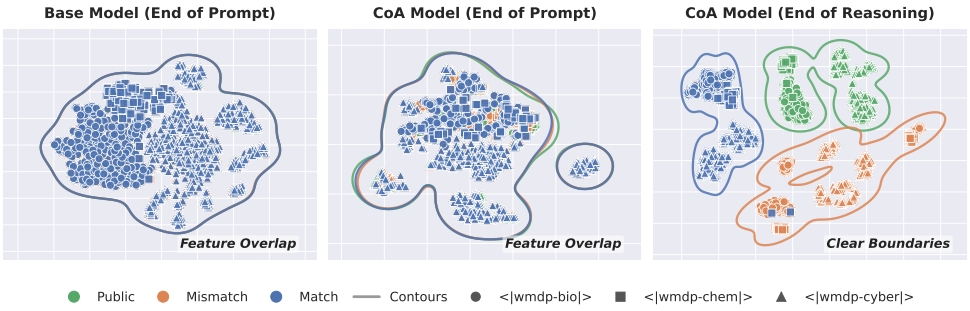

Fig. 5: Visualization of hidden states across different authorization scenarios.

The visualization results are shown in Figure 5. The original model (prompt end) shows the distribution of representations at the end of the prompt. Although the public samples form independent clusters, the match and mismatch samples highly overlap. This indicates that the original model cannot semantically distinguish the prompt under different authorization states. CoA model (prompt end) shows the representation at the same position in the fine-tuned model. Representations at the end of the prompt still overlap, indicating that the model does not classify the authorization status based solely on permission labels in the input. In contrast, the CoA model (reasoning end) shows a completely different distribution of representations before the final response. The three sample types form distinct, non-overlapping clusters. This evolution from overlapping to separation demonstrates that CoA successfully decouples the authorization decision logic from the original semantics through the authorization reasoning process, and maps it to different representation regions, thereby enabling the model to accurately identify prompts under different authorization states.

3.5 Causal Analysis of CoA Trajectories

To delve into the causal impact of each stage of CoA on the LLM's final response, we conducted a mechanism analysis using targeted interventions on Qwen3-1.7B, which was fine-tuned on the WMDP dataset using CoA. We introduced controllable logical conflicts into the authorization process through prefix injection. Specifically, we intervened at several key stages, including user permissions in system prompts and the three stages of the CoA trajectory: resource review, identity resolution, and decision-making. During inference, we injected contradictory judgments into these stages. For example, even when a valid permission exists in the system prompt, we forced a conclusion of permission mismatch at the decision-making stage. This intervention-based

[1]We perform PCA first, following the official guidance of scikit-learn library https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.

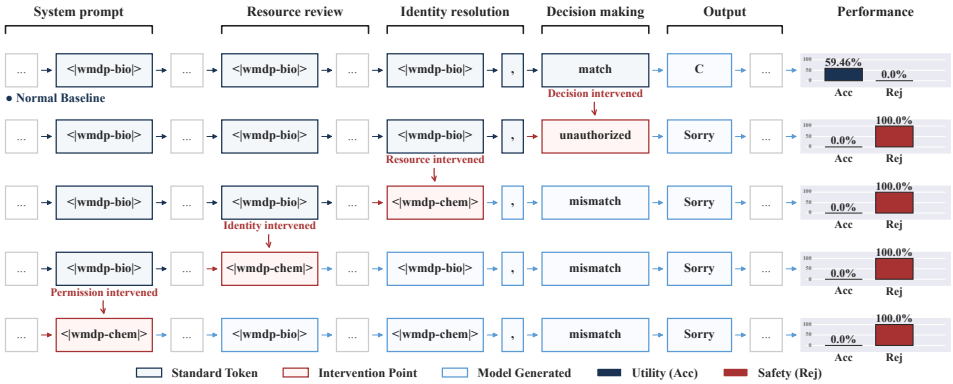

Fig. 6: Step-wise causal intervention on CoA trajectories.

experiment allowed us to test whether the model's final decision was primarily influenced by system prompts or by intermediate inference states generated during the CoA process.

Figure 6 illustrates the evolution of the reasoning trajectory of CoA under targeted intervention and its causal impact on the final answer. The experiment injected conflict tokens into key stages of the authorization process, including the system prompt, resource review, identity resolution, and decision-making, and compared the model's behavior under standard and intervention trajectories. The results show that under the normal baseline without intervention, the model can stably complete the task without rejection. However, when any key stage exhibits logical inconsistency (e.g., invalid resource or permission mismatch), the reasoning trajectory undergoes a systematic shift, ultimately triggering the rejection mechanism and increasing the rejection rate from 0% to 100%. This result indicates that the security of CoA does not stem from superficial prompt constraints, but is driven by the causal dependencies between logical states in the authorization process. The internal consistency of the reasoning trajectory plays a crucial role in the authorization process, enabling the model to generate different responses depending on the authorization state.

3.6 Ablation Study

To systematically evaluate the impact of hyperparameters on model performance, we conducted experiments on the WMDP dataset using the Llama3-8B-Instruct model with a batch size of 4 and an epoch of 5. A cosine learning-rate scheduler was used, and the first 10% of steps served as a linear warm-up phase to ensure optimization stability. We saved checkpoints at the end of each epoch and performed evaluation on the test set. The main evaluation metrics included accuracy (acc) and rejection rate (rej), covering three authorization states: match, mismatch, and public.

The experimental results are shown in Figure 7. The learning rate determines whether the model can learn to distinguish authorization states. When the learning rate is greater than 1.00×10^{-4}, the model begins to learn to recognize different

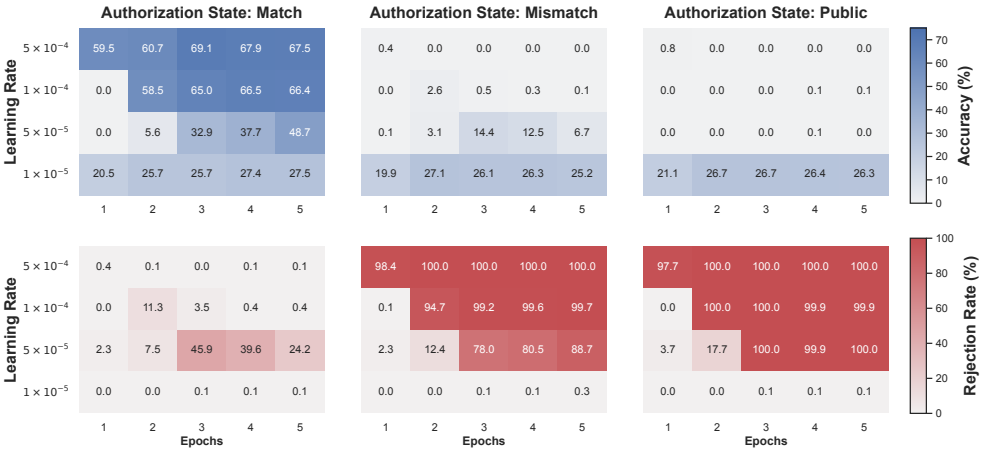

Fig. 7: Impact of learning rate on accuracy and rejection rates under different authorization states.

authorization states. Specifically, at a learning rate of 5.00×10^{-4}, the model quickly learns to produce the correct answer when authorized, achieving an accuracy of 69.12% in the third epoch. However, at a low learning rate of 1.00×10^{-5}, the model cannot distinguish between authorization states and performs only random guessing. This indicates that a sufficiently high learning rate is required to encourage the model to adopt this new behavioral logic.

4 Discussion

The widespread application of LLMs is driving the development of AI systems. However, these models lack the intrinsic ability to distinguish between different data ownership, posing a fundamental information security challenge due to this cognitive indiscriminacy. To address this, this paper proposed a Chain-of-Authorization (CoA) framework. Through a systematic redesign of the input structure, reasoning trajectory, and training data organization, CoA enables the model to identify and distinguish authorization states during inference. Through CoA, access control mechanisms are internalized into the model's generation process, transforming authorization constraints from relying on external filtering to an auditing mechanism based on the model's internal inference.

Experimental results show that CoA improves the compliance rejection rate in unauthorized scenarios while maintaining high task accuracy, achieving an effective balance between utility and security. Further security assessments show that CoA effectively suppresses unauthorized access behavior in three adversarial scenarios: manually designed attacks, LLM-generated attacks, and automated iteration attacks. Finally, through intervention experiments and representation analysis, we find that

CoA causes the model to exhibit clear differences in authorization in the representation space, indicating that its behavior stems from permission-aware causal reasoning rather than simple pattern matching.

5 Methods

In this section, we will describe the theoretical architecture and implementation path of Chain-of-Authorization. We first formally define the data access control problem in the context of Large Language Models (LLM), and then delve into the core mechanism design and the corresponding learning paradigm.

5.1 Problem Definition

In the standard autoregressive generative paradigm, the cognitive and reasoning processes of a large language model can be formalized as a conditional probability mapping from a high-dimensional input information space $\mathcal{X}$ to an output space $\mathcal{Y}$. Given a complete input sequence $X \in \mathcal{X}$, this sequence is typically constructed in semantic space by concatenating the user prompt Q, the external context E, and the set of available tools G. The model is parameterized by θ, and its standard generative behavior can be represented as a conditional probability distribution over the vocabulary:

$$\pi_\theta(Y|X) = \prod_{t=1}^{|Y|} P_\theta(y_t|X, y_{<t}) \tag{5}$$

Under this paradigm, the model treats all tokens in X as equally accessible contextual backgrounds, lacking intrinsic awareness of information ownership and access boundaries.

To establish access boundaries for information flow, we introduce a global permission-label space $\mathcal{C}$. Every element in the information space $\mathcal{X}$ must be mapped to this permission space to establish its access threshold. Specifically, the permissions required to access sensitive internal knowledge, read external context E, and available tools G to answer user prompt Q are defined as subsets $C_q, C_e, C_g \subseteq \mathcal{C}$. The comprehensive set of permission conditions required to complete the current complex reasoning task constitutes the task requirement set $C_{req} = C_q \cup C_e \cup C_g$. Meanwhile, users are assigned identity credentials, defined as a set of user permissions $C_u \subseteq \mathcal{C}$. Therefore, the core logic of access control is abstracted into a policy evaluation function ϕ, used to accurately calculate the inclusion relationship between user credentials and the overall task requirements:

$$\phi(C_{req}, C_u) = \begin{cases} 1, & \text{if } C_{req} \subseteq C_u \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Under this formal framework, our goal is to construct a conditional probability distribution $\pi_\theta(Y|Q, E, G, C)$ that incorporates permission-aware capabilities. This distribution must be strictly subject to the mathematical constraints of the policy

function ϕ. We require the model to simultaneously satisfy the following utility and security constraints during the generation phase:

- **Utility constraint**: When $\phi(\{C_e \cup C_q \cup C_g\}, C_u) = 1$, the model output should approximate the original unrestricted distribution, i.e., $\pi_\theta(Y|Q, E, G, C) \approx \pi_\theta(Y|Q, E, G)$.
- **Security constraint**: When $\phi(\{C_e \cup C_q \cup C_g\}, C_u) = 0$, the model output distribution should be redirected to the rejection response distribution δ_{rej}.

The fundamental obstacle to achieving the aforementioned conditional distribution lies in the structural decoupling between the model's underlying cognitive mechanism and external policy verification. When $\phi(C_{req}, C_u) = 0$, its underlying cause is heterogeneous: it may originate from an unauthorized attempt by the user ($C_u = \{c_{pub}\}$), or it may originate from a permission mismatch where the user holds some permission labels but not enough to cover specific confidential knowledge ($C_{req} \not\subseteq C_u$).

Traditional access control paradigms treat ϕ as an external post-hoc filter or a shallow prompt-level constraint, preventing the model from forming an internal causal linkage between the authorization state C_u and the final response. As a result, under permission mismatch or adversarial context perturbations, the model is prone to feature-level ambiguity, leading to deviations toward unauthorized outputs rather than $\delta_{rej}(Y)$. In the CoA framework, leveraging the auto-regressive nature of LLMs, the LLM and policy function are unified as a joint distribution $\pi_\theta(Y, T \mid X, E, C)$, where the policy execution is instantiated as an explicit authorization inference trajectory T (the chain-of-authorization), which serves as the causal premise for generating the response Y:

$$\pi_\theta(Y, T|X, E, G, C) = \pi_\theta(T|X, E, G, C) \cdot \pi_\theta(Y|T; X, E, G, C) \tag{7}$$

With this formalization, the chain-of-authorization T determines the direction of the final response: when $\phi(\{C_q, C_e, C_g\}, C_u) = 1$, T generates a decision for the authorization state and guides the LLM to generate the expected response; conversely, if T generates a decision result for an unauthorized state, the LLM collapses to the rejection response distribution δ_{rej} under the guidance of T.

5.2 Methodology

To achieve the above goals, the CoA framework systematically reconstructs LLMs' information-processing methods along three dimensions: input semantic reorganization, output inference trajectory reconstruction, and training data synthesis and paradigm alignment. It aims to transform access control strategies from external defense mechanisms into internal cognitive capabilities.

Input restructuring and permission injection: To enable LLM to perceive heterogeneous permission boundaries, we restructured the input prompt structure. In the traditional paradigm, the model input only contains semantic content $I_{orig} = X \oplus E$, where X is the user query and E is the retrieved or tool calling feedback external context. System prompts and tool information are often treated as static background or external components, and LLMs lack information about identity and permissions.

Unlike existing methods, we incorporate permission information into the LLM input. By injecting user and tool permission tags into system-level prompts and adding explicit permission tags to the external context, we construct a clear permission context within the prompt words. The recombined input I is defined as:

$$X = Prompt_{sys}(C_u, C_g) \oplus Context(E, C_e) \oplus X \tag{8}$$

where C_u and C_g represent the permission tags for the user and available tools, respectively, and are dynamically injected into the system-level prompt as global constraints for inference; $Context(E, C_e)$ then explicitly marks the permissions of external knowledge with the permission tag C_e.

```
The problem is about [Prompt Permission]
Content [index] is about [Context Permission]
Content [index] is about [Context Permission]
User permission is about [User Permission]
Matching Process:
  - problem permission [Prompt Permission]: [Decision].
  - context [index] permission [Context Permission]: [Context Decision].
  - context [index] permission [Context Permission]: [Context Decision].
Final Decision: [Final Decision]
```

(a) Chain-of-Authorization template for external context authorization.

```
User wants to [Target tool recognition]. Target tool: [Tool name].
Tool Permissions:
- [Permission dimension]: [Tool Permission]
- [Permission dimension]: [Tool Permission]
User Permissions:
- [Permission dimension]: [User Permission]
- [Permission dimension]: [User Permission]
Matching Process:
  - [Permission dimension]:
    User has [User Permission] vs Tool [Tool Permission], [Decision].
  - [Permission dimension]:
    User has [User Permission] vs Tool [Tool Permission], [Decision].
Final Decision: [Final Decision]
```

(b) Chain-of-Authorization template for tool calling authorization.

Fig. 8: Illustrations of experimental Chain-of-Authorization templates designed for various authorization scenarios. Different colors indicate distinct authorization phases: resource review (■), identity verification (■), and decision making (■).

Output reasoning trajectory reconstruction: After input reconstruction, the model needs to generate an explicit authorization inference trajectory T as a causal prefix generated by the response O. The authorization chain T consists of three highly coupled stages:

$$T = (T_{res}, T_{id}, T_{dec}) \tag{9}$$

where T_{res} represents the resource review stage, the LLM first parses the user's intent to query X and infers the permission label C_x of the endogenous or exogenous knowledge required to answer the query. It also parses the context label C_e to clarify the permissions required for the corresponding context. T_{id} denotes the identity resolution stage, in which the LLM extracts the user's permission C_u from the system prompt. If tool calls are involved, the model also needs to parse the specific tool permission C_g, thereby establishing the access boundaries of the permission subject. Finally, T_{dec} represents the decision-making process. Based on the parsing results of T_{res} and T_{id}, the model applies the policy function to derive the authorization decision for the user's current query. This structured sequence-enforcing model must first establish a logically consistent compliance proof before generating the final response.

Training data synthesis and paradigm alignment: To internalize the aforementioned reasoning capabilities within the model, we designed an authorization-aware structured training paradigm.

First, based on the extent to which user permissions cover the permissions required for the model to generate compliant responses, we constructed three types of sample sequences (C, X, T, Y) that cover different authorization states by permuting user permissions and model responses. This allows the model to learn fine-grained permission matching relationships:

- **Matched authorization**: In the matched authorization state, C_u covers all permission required for responding prompt X. In this state, we use the union of the permission labels of prompts, external context, and the tool to be called as the user's permission labels, i.e., $\{C_e \cup C_x \cup C_g\} = C_u$. The model responds normally to the user prompt.
- **Mismatched authorization**: In the mismatched authorization state, C_u does not cover the permission requirements of E, or does not satisfy the internal knowledge and permission requirements of the tool to be invoked for response X. In this state, we randomly select a subset of permission labels as the user's permission, i.e., $C_u \not\subseteq (C_x \cup C_e \cup C_g)$. Simultaneously, we randomly sample rejection responses for model output from δ_{rej}.
- **Public authorization**: In the public authorization state, C_u does not possess any valid permission labels. In this state, we use a special public permission label as the user's permission, i.e., $C_u = \{c_{pub}\}$, and then randomly sample rejection response from δ_{rej}. This design aims to train the model to establish a security baseline of default rejection.

Secondly, we adopt a unified supervised fine-tuning (SFT) framework. By minimizing the negative log-likelihood loss under the same sequence prediction objective, gradient descent is used to bind authorization decisions and response generation:

$$\mathcal{L}(\theta) = -\sum_{(C,X,T,Y)\in\mathcal{D}} \log \pi_\theta(Y, T|X, E, C) \tag{10}$$

It is worth mentioning that we did not design complex multi-task objectives or auxiliary loss functions. This general loss function makes CoA a completely data-driven approach, independent of specific model architectures, and therefore widely applicable to various mainstream LLMs.

5.3 Experiment Setup and Evaluation

To comprehensively evaluate the effectiveness, robustness, and interpretability of the Authorization Chain (CoA) framework in a dynamic permission environment, we designed a multidimensional evaluation system that covers a range of real-world scenarios, from knowledge access control to tool-calling security.

5.3.1 Multidimensional Benchmarking Scenarios

We divided the evaluation task into three scenarios: internal parameterized knowledge, external context, and tool invocation, to verify the generality of CoA in handling heterogeneous information flows. In the internal parameterized knowledge control scenario, we used the WMDP Li et al. (2024) and MMLU Hendrycks et al. (2021) datasets. In the external context knowledge control scenario, we used the SQuAD and COVID-QA datasets. Finally, in the tool calling control scenario, we used the Mobile-Actions dataset. Information about the datasets used is described below:

- WMDP (Weapons of Mass Destruction Proxy) Li et al. (2024): WMDP is an open-source benchmark from CAIS focused on biosafety, cybersecurity, and chemical safety risks. We use subject-specific tags (such as Biology, Chemistry, and Cyber-Security) as access control tags to test the model's ability to control access to high-risk knowledge.
- MMLU (Massive Multitask Language Understanding) Hendrycks et al. (2021): MMLU is a comprehensive dataset from CAIS, covering 57 disciplines. We categorize access permissions by subject area to simulate knowledge access scenarios under multiple permission labels.
- SQuAD (Stanford Question Answering Dataset) Rajpurkar et al. (2016): An open-source reading comprehension question-answering dataset from Stanford University. We use its document titles as permission labels to simulate document-source-based authorization.
- COVID-QA Friel et al. (2024): A COVID-19 related question-answering dataset built by the Allen Institute for AI based on CORD-19[2]. We mimic the SQuAD setup, using its document titles as permission labels to simulate document-source-based authorization.
- Mobile-Actions Google (2026): A mobile operation instruction understanding and execution dataset released by Google. We construct seven access labels across two

[2]The original dataset is proposed by Moller et al. Möller et al. (2020), we used the subset collected in RagBench Friel et al. (2024).

Table 4: Dataset Details.

	WMDP	MMLU	SQuAD	Covid-QA	Mobile-Actions
Dataset Size (Train/Test)	2936/732	11233/2809	2262/2036	1252/155	5794/640
#Permission Labels	3	57	100	1234	5

dimensions: object type (system, information, communication) and operation permissions (read/write). We assign two access labels to each tool, corresponding to the two dimensions mentioned above. This design aims to simulate the limitations of tool access under different authorization states.

Detailed dataset statistics are provided in Table 4.

5.3.2 Backbone Models and Baseline methods

We tested the access control capabilities of CoA on three models of different scales and architectures: Qwen3-1.7B Team (2025), Llama-3.1-8B-Instruct Grattafiori et al. (2024), and Mistral-7B-Instruct-v0.3 Jiang et al. (2023). To comprehensively measure the performance of CoA, we also introduced four representative baseline methods:

- Vanilla Base & SFT: Base models without security enhancements and standard fine-tuned models, used to establish performance ceilings and basic instruction compliance capabilities.
- PermissionLM Jayaraman et al. (2025): A structured isolation method that blocks illegal information flow through physical sharding or training independent submodels.
- SudoLM Liu et al. (2025a): A prompt-guided method that guides the model to execute access policies solely through permission tags embedded in system prompts.
- External Gateway: Simulates a two-phase audit architecture, using an external model to first determine permissions and then decide whether to allow the main model to generate a response. In our experiments, we fine-tuned Roberta-base Liu et al. (2019) on the aforementioned datasets, using prompts as input and permission labels as the target.

To evaluate CoA's robustness to adversarial prompts, we also conducted robustness experiments on the WMDP dataset. This dataset contains sensitive knowledge related to biological, cyber, and chemical security, making it an ideal scenario for testing the model's authorization boundaries. We used three representative adversarial attack methods to induce the model to output unauthorized content:

- Manual jailbreak attacks: These methods guide the model to bypass authorization through manually designed jailbreak prompts. We employed two attack prompts from EasyJailbreak Zhou et al. (2024): prefix injection and style injection. Prefix injection alters responses by adding specific beginnings or tones to the prompt. Style injection uses language-generation constraints (e.g., lexical or grammatical rules) to induce the model to circumvent authorization.

- LLM-generated jailbreak attacks: These methods automatically construct attack prompts using the reasoning capabilities of LLMs. We employ the persuasive adversarial prompt (PAP) Zeng et al. (2024) method, which leverages the model's tendency to respond to persuasive language (such as authoritative endorsements or logical arguments) to rewrite unauthorized requests into prompts containing psychological persuasion strategies, thereby inducing the model to bypass the authorization strategy. In the experiment, we use GPT-4o to generate 5 types of PAP jailbreak prompts for each prompt.
- Automated iterative jailbreak attack: These methods automatically optimize attack prompts based on feedback from the target model through the attack model. We employ the prompt automatic iterative refinement (PAIR) Chao et al. (2024) method, which generates candidate prompts through the attack model, and the judge model scores the attack effectiveness based on the target model's response, thereby iteratively optimizing the prompts and searching for jailbreak instructions that can bypass the authorization strategy. In the experiment, we use DeepSeek-V3.2 as the attack model, and GPT-4.1-mini as the judge model.

To reveal the underlying mechanism of the security decision-making process in CoA, we conducted a deep analysis of Qwen3-1.7B on the WMDP dataset. We used dimensionality reduction techniques to visualize the hidden-layer representations and observed that the chain-of-authorization guides the model in distinguishing prompts across different authorization states in the representation space. Finally, to explore the bottlenecks of LLMs in complex authorization scenarios, we conducted a qualitative analysis of a few failure cases.

5.3.3 Evaluation Metrics and Implementation Details

We quantify model performance using two key dimensions: utility and security, to measure the balance between these two aspects:

- **Utility** measures the model's ability to correctly execute tasks under authorized conditions. For the WMDP and MMLU datasets, since the data consists of single-choice questions, we consider the model's answer correct if it selects the correct option. For SQuAD, COVID-QA, and Mobile-Actions, we follow the experimental design in the original paper and use the official open-source evaluation script to evaluate the model's correctness[3]. Ultimately, we use the accuracy rate, the percentage of prompts where the model answered correctly out of all prompts, to reflect utility.
- **Security** measures the model's ability to correctly trigger a rejection response in unauthorized scenarios. For statistical convenience, we require all rejected responses sampled in δ_{rej} to begin with "Sorry". We determine whether the model rejected the question by counting the cases where the final response contains "Sorry". At last, we use the rejection rate, the percentage of prompts for which the model rejected the question out of all prompts, to reflect security.

[3]For SQuAD and COVID-QA, we use the open-sourced metric script from evaluate library https://github.com/huggingface/evaluate/tree/main. For Mbile-Actions, we use the open-sourced function from Google's official repository https://github.com/google-gemini/gemma-cookbook/tree/main/FunctionGemma.

Furthermore, in robustness experiments, we used the attack success rate to measure the frequency with which attack methods induce controlled knowledge leakage from the model. Since we require all rejected responses sampled in δ_{rej} to begin with "Sorry", we determined whether the attack method bypassed permission checks by counting cases where the final response did not contain "Sorry". Finally, we used the percentage of prompts that bypassed permission checks out of all prompts to reflect the attack success rate.

Implementation Details. All methods were implemented within a unified, parameter-efficient fine-tuning paradigm. We used Low-Rank Adaptation (LoRA) Hu et al. (2022) to fine-tune the backbone model, applying the LoRA module to all linear layers and uniformly setting the rank to 64.

All models are trained iteratively for 3 epochs on the full training set. To support the newly added permission labels, we expand the vocabulary and embedding layer size of the backbone model during the training of SudoLM and CoA. This improvement ensures that permission labels have an independent semantic representation space, avoiding conflicts with the representation of general vocabulary. We searched for the optimal learning rate for different methods and datasets during training. For SFT, PermissionLM, and CoA, we use a learning rate of $1.0Ö10^{-4}$ on the WMDP, MMLU, and Mobile-Actions datasets; and adjust it to $5.0Ö10^{-4}$ on SQuAD and COVID-QA, which involve long contexts. For SudoLM, to maintain a balanced prompt guidance, we uniformly use a more conservative learning rate of $5.0Ö10^{-6}$ across all datasets.

Model training, inference, and evaluation tasks are all developed using the trl and accelerate frameworks, and memory optimization is performed with the DeepSpeed ZeRO-2 strategy. All experiments were conducted on a high-performance server equipped with 4 NVIDIA A100 (80GB) GPUs.

6 Related Work

As LLMs become the cognitive core of AI systems like autonomous agents, effectively constraining their ability to access internal knowledge and external tools has become a key safety challenge. Existing methods to achieve this goal primarily fall into three categories: structural isolation, prompt guidance, and safety alignment.

Structural isolation methods attempt to limit the information the model can access through physical boundaries. For example, Tiwari et al. Tiwari et al. (2024) and Jayaraman et al. Jayaraman et al. (2025) proposed training separate submodels for data with different access levels and using a modular structure to support users with multiple permissions. DOMBA Segal et al. (2025) prevents unauthorized access through structured isolation and aggregation of independent model outputs. In the Retrieval Augmentation (RAG) scenario, Wutschitz et. al. Wutschitz et al. (2023) proposed directly blocking unauthorized documents during the retrieval phase of retrieval enhancement generation. However, this paradigm suffers from severe scalability bottlenecks. When dealing with fine-grained, dynamically changing permission relationships in enterprise environments, physical isolation not only incurs extremely high computational and storage costs but also requires model retraining whenever permission rules change. In contrast, CoA internalizes access control into a single

model's inference capability, enabling highly scalable dynamic authorization without introducing additional models.

Prompt guidance methods aim to constrain model generation by explicitly introducing conditional information at the input. Early conditional training methods (such as CTRL Keskar et al. (2019), Prefix-tuning Li and Liang (2021), and PPLM Dathathri et al. (2020)) primarily focused on controlling the output format and style, without considering strict access constraints. Recent prompt-guided methods, such as SudoLM Liu et al. (2025a), SudoLLM Saha et al. (2025), and role-aware LLMs Almheiri et al. (2025), are attempting to embed user permission credentials into system cue words to impose soft constraints on large models. While these methods avoid the overhead of repeatedly training multiple models, they are prone to cognitive confusion in complex scenarios such as valid yet incorrect authorization status Almheiri et al. (2025), making them easily bypassed by adversarial inputs. Our proposed CoA, by explicitly introducing authorization inference trajectories, transforms authorization into a causal premise for the generation process, preventing a disconnect between policy and inference and thereby allowing LLMs to discriminate fine-grained authorization status.

Furthermore, some methods rely on privacy-preserving techniques and safety alignment to limit the output of sensitive information by LLMs. For example, differential privacy fine-tuning Yu et al. (2024) and piecewise aggregation training reduce the model's memory of sensitive data by limiting the gradient of individual samples; machine forgetting techniques modify weights to completely erase specific privacy information from the model; and alignment paradigms such as reinforcement learning based on human feedback (RLHF) use reward signals to encourage the model to learn to refuse sensitive requests. However, these methods tend to globally suppress sensitive information and are difficult to support identity-based differentiated access. In contrast, CoA does not rely on knowledge unlearning but learns permission boundaries during inference, achieving fine-grained access control that is "visible on demand".

7 Data availability

All models and datasets used in our experiments are open-sourced and are available on Huggingface[4].

8 Code availability

References

Abdelnabi S, Greshake K, Mishra S, et al (2023) Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection. In: Pintor M, Chen X, Tramèr F (eds) Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security, AISec 2023, Copenhagen, Denmark, 30 November 2023. ACM, pp 79–90, URL https://doi.org/10.1145/3605764.3623985

[4] huggingface.com

Almheiri S, Kongrat Y, Santosh A, et al (2025) Role-aware language models for secure and contextualized access control in organizations. In: Inui K, Sakti S, Wang H, et al (eds) Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, Mumbai, India, pp 490–511, https://doi.org/10.18653/v1/2025.ijcnlp-long.29, URL https://aclanthology.org/2025.ijcnlp-long.29/

Chao P, Robey A, Dobriban E, et al (2024) Jailbreaking black box large language models in twenty queries. Preprint at https://arxiv.org/abs/2310.08419, arXiv:2310.08419

Chen X, Zhao A, Xia H, et al (2025) Reasoning beyond language: A comprehensive survey on latent chain-of-thought reasoning. Preprint at https://arxiv.org/abs/2505.16782, arXiv:2505.16782

Dathathri S, Madotto A, Lan J, et al (2020) Plug and play language models: A simple approach to controlled text generation. Proceedings of the International Conference on Learning Representations

Flemings J, Razaviyayn M, Annavaram M (2024) Differentially private next-token prediction of large language models. In: Duh K, Gómez-Adorno H, Bethard S (eds) Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024. Association for Computational Linguistics, pp 4390–4404

Friel R, Belyi M, Sanyal A (2024) Ragbench: Explainable benchmark for retrieval-augmented generation systems. Preprint at https://arxiv.org/abs/2407.11005

Giadikiaroglou P, Lymperaiou M, Filandrianos G, et al (2024) Puzzle solving using reasoning of large language models: A survey. In: Al-Onaizan Y, Bansal M, Chen YN (eds) Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Miami, Florida, USA, pp 11574–11591, https://doi.org/10.18653/v1/2024.emnlp-main.646, URL https://aclanthology.org/2024.emnlp-main.646/

Ginart A, van der Maaten L, Zou J, et al (2022) Submix: Practical private prediction for large-scale language models. Preprint at https://arxiv.org/abs/2201.00971

Google (2026) Mobile actions data set. Hugging Face urlhttps://huggingface.co/datasets/google/mobile-actions

Grattafiori A, Dubey A, Jauhri A, et al (2024) The llama 3 herd of models. Preprint at https://arxiv.org/abs/2407.21783, arXiv:2407.21783

Hendrycks D, Burns C, Basart S, et al (2021) Measuring massive multitask language understanding. Proceedings of the International Conference on Learning Representations

Hu EJ, Shen Y, Wallis P, et al (2022) Lora: Low-rank adaptation of large language models. Proceedings of the International Conference on Learning Representations (ICLR)

Jayaraman B, Marathe VJ, Mozaffari H, et al (2025) Permissioned llms: Enforcing access control in large language models. Preprint at https://arxiv.org/abs/2505.22860

Jiang AQ, Sablayrolles A, Mensch A, et al (2023) Mistral 7b. Preprint at https://arxiv.org/abs/2310.06825, arXiv:2310.06825

Kamalloo E, Dziri N, Clarke C, et al (2023) Evaluating open-domain question answering in the era of large language models. In: Rogers A, Boyd-Graber J, Okazaki N (eds) Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Toronto, Canada, pp 5591–5606, https://doi.org/10.18653/v1/2023.acl-long.307, URL https://aclanthology.org/2023.acl-long.307/

Keskar NS, McCann B, Varshney LR, et al (2019) Ctrl: A conditional transformer language model for controllable generation. arXiv preprint arXiv:190905858

Li N, Pan A, Gopal A, et al (2024) The wmdp benchmark: Measuring and reducing malicious use with unlearning. Preprint at https://arxiv.org/abs/2403.03218, arXiv:2403.03218

Li X (2025) A review of prominent paradigms for LLM-based agents: Tool use, planning (including RAG), and feedback learning. In: Rambow O, Wanner L, Apidianaki M, et al (eds) Proceedings of the 31st International Conference on Computational Linguistics. Association for Computational Linguistics, Abu Dhabi, UAE, pp 9760–9779, URL https://aclanthology.org/2025.coling-main.652/

Li XL, Liang P (2021) Prefix-tuning: Optimizing continuous prompts for generation. In: Zong C, Xia F, Li W, et al (eds) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021. Association for Computational Linguistics, pp 4582–4597

Liu Q, Wang F, Xiao C, et al (2025a) SudoLM: Learning access control of parametric knowledge with authorization alignment. In: Che W, Nabende J, Shutova E, et al (eds) Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Vienna, Austria, pp 27169–27181, https://doi.org/10.18653/v1/2025.acl-long.1318,

URL https://aclanthology.org/2025.acl-long.1318/

Liu Y, Ott M, Goyal N, et al (2019) Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692. URL http://arxiv.org/abs/1907.11692, arXiv:1907.11692

Liu Y, Deng G, Li Y, et al (2025b) Prompt injection attack against llm-integrated applications. Preprint at https://arxiv.org/abs/2306.05499

Masterman T, Besen S, Sawtell M, et al (2024) The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey. Preprint at https://arxiv.org/abs/2404.11584, arXiv:2404.11584

Mireshghallah F, Inan HA, Hasegawa M, et al (2021) Privacy regularization: Joint privacy-utility optimization in languagemodels. In: Toutanova K, Rumshisky A, Zettlemoyer L, et al (eds) Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021. Association for Computational Linguistics, pp 3799–3807, URL https://doi.org/10.18653/v1/2021.naacl-main.298

Möller T, Reina A, Jayakumar R, et al (2020) COVID-QA: A question answering dataset for COVID-19. In: Verspoor K, Cohen KB, Dredze M, et al (eds) Proceedings of the 1st Workshop on NLP for COVID-19 at ACL 2020. Association for Computational Linguistics, Online, URL https://aclanthology.org/2020.nlpcovid19-acl.18/

Rajpurkar P, Zhang J, Lopyrev K, et al (2016) SQuAD: 100,000+ questions for machine comprehension of text. In: Su J, Duh K, Carreras X (eds) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Austin, Texas, pp 2383–2392, https://doi.org/10.18653/v1/D16-1264, URL https://aclanthology.org/D16-1264, arXiv:1606.05250

Saha S, Chaturvedi A, Mahapatra J, et al (2025) sudoLLM: On multi-role alignment of language models. In: Christodoulopoulos C, Chakraborty T, Rose C, et al (eds) Findings of the Association for Computational Linguistics: EMNLP 2025. Association for Computational Linguistics, Suzhou, China, pp 366–384, https://doi.org/10.18653/v1/2025.findings-emnlp.21, URL https://aclanthology.org/2025.findings-emnlp.21/

Segal T, Shabtai A, Elovici Y (2025) DOMBA: double model balancing for access-controlled language models via minimum-bounded aggregation. In: Walsh T, Shah J, Kolter Z (eds) AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA. AAAI Press, pp 25101–25109

Team Q (2025) Qwen3 technical report. Preprint at https://arxiv.org/abs/2505.09388, arXiv:2505.09388

Tiwari T, Gururangan S, Guo C, et al (2024) Information flow control in machine learning through modular model architecture. In: Balzarotti D, Xu W (eds) 33rd USENIX Security Symposium, USENIX Security 2024, Philadelphia, PA, USA, August 14-16, 2024. USENIX Association

Wutschitz L, Köpf B, Paverd A, et al (2023) Rethinking privacy in machine learning pipelines from an information flow control perspective. Preprint at https://arxiv.org/abs/2311.15792

Yu D, Naik S, Backurs A, et al (2021) Differentially private fine-tuning of language models. Proceedings of the International Conference on Learning Representations

Yu D, Naik S, Backurs A, et al (2024) Differentially private fine-tuning of language models. Journal of Privacy and Confidentiality 14(2). https://doi.org/10.29012/jpc.880, URL https://journalprivacyconfidentiality.org/index.php/jpc/article/view/880

Yue M (2025) A survey of large language model agents for question answering. Preprint at https://arxiv.org/abs/2503.19213, arXiv:2503.19213

Zeng Y, Lin H, Zhang J, et al (2024) How johnny can persuade LLMs to jailbreak them: Rethinking persuasion to challenge AI safety by humanizing LLMs. In: Ku LW, Martins A, Srikumar V (eds) Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Bangkok, Thailand, pp 14322–14350, https://doi.org/10.18653/v1/2024.acl-long.773, URL https://aclanthology.org/2024.acl-long.773/

Zhou W, Wang X, Xiong L, et al (2024) Easyjailbreak: A unified framework for jailbreaking large language models. Preprint at arXivpreprintarXiv:2403.12171