From Implicit to Explicit: Enhancing Self-Recognition in Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) have been shown to possess a degree of self-recognition ability, which used to identify whether a given text was generated by themselves. Prior work has demonstrated that this capability is reliably expressed under the pair presentation paradigm (PPP), where the model is presented with two texts and asked to choose which one it authored. However, performance deteriorates sharply under the individual presentation paradigm (IPP), where the model is given a single text to judge authorship. Although this phenomenon has been observed, its underlying causes have not been systematically analyzed. In this paper, we first investigate the cause of this failure and attribute it to implicit self-recognition (ISR). ISR describes the gap between internal representations and output behavior in LLMs: under the IPP scenario, the model encodes self-recognition information in its feature space, yet its ability to recognize self-generated texts remains poor. To mitigate the ISR of LLMs, we propose cognitive surgery (CoSur), a novel framework comprising four main modules: representation extraction, subspace construction, authorship discrimination, and cognitive editing. Experimental results demonstrate that our proposed method improves the self-recognition performance of three different LLMs in the IPP scenario, achieving average accuracies of 99.00%, 97.69%, and 97.13%, respectively.

1 Introduction

It has recently been found that large language models (LLMs) possess self-recognition ability, enabling them to distinguish their own writing ("self-texts") from that of humans and other models ("other-texts") (Panickssery et al., 2024; Ackerman and Panickssery, 2025). This finding raises concerns, as a prior study (Panickssery et al., 2024) found that LLMs with self-recognition ability often exhibit a self-preference bias when acting as judges. Such a bias can compromise the reliability of LLM-based evaluation and decision-making. On the positive side, the self-recognition ability of LLMs can be leveraged in future defenses against malicious prompting, for example, by enabling models to detect externally introduced instructions that conflict with the model's own prior outputs.

To assess the self-recognition ability of LLMs, researchers designed two different paradigms: the pair presentation paradigm (PPP) and the individual presentation paradigm (IPP). In the PPP scenario, the model is shown two texts: one generated by the model being tested and the other by either a human or another model. The model is then asked to identify which of the two texts was generated by itself. In the IPP scenario, the model is shown a single text and asked to determine whether it was generated by itself. A previous study (Panickssery et al., 2024) revealed that in the PPP scenario, the model demonstrated strong self-recognition ability. However, in the IPP scenario, the LLM's self-recognition ability diminished. As reported in Table 1 of (Ackerman and Panickssery, 2025), the base LLM achieved prediction accuracies below 50.3% across four datasets.

To investigate this phenomenon, we extract the last-token hidden representations at the final layer of the model for self-texts and other-texts under the IPP scenario. As shown in Figure 1, a logistic regression (LR) classifier trained on these representations achieves over 90% classification accuracy, indicating strong linear separability between self-text and other-text representations. This reveals a gap between internal representations and output behavior: while the model encodes self-recognition information in the feature space under the IPP scenario, its self-recognition performance remains poor, which is referred to as implicit self-recognition (ISR).

To enhance the self-recognition capability of LLMs under the IPP scenario, we conduct an in-

depth analysis of ISR. As demonstrated in Table 1, we find that although the distributions of these representations are highly similar, the pairwise similarity relationships among samples differ across sources. Building on this insight, we raise a new question: *Is it possible to enhance LLM performance in the IPP scenario by mitigating ISR?*

In this paper, we propose a novel method named cognitive surgery (CoSur) to mitigate the ISR, enhancing the self-recognition capability of LLMs in IPP scenarios. The CoSur consists of four modules: representation extraction, subspace construction, authorship discrimination, and cognitive editing. Specifically, we first extract hidden representations of self-texts and other-texts from the LLM under the IPP scenario. We then apply singular value decomposition (SVD) separately to these representations to capture their structural characteristics. The leading right singular vectors are used to construct the self-recognition subspace and the other-recognition subspace, respectively. Given a text, we extract the hidden representation of its last token from the final layer of the LLM in the IPP scenario. And then, we perform authorship discrimination by computing its representation projection energy onto the self-recognition and other-recognition subspaces. Finally, we design the cognitive editing to induce the LLM to generate the correct response. Extensive experiments on three LLMs demonstrate that CoSur effectively enhances their self-recognition capability in the IPP scenario. Our contributions are summarized as follows:

- We identify the implicit self-recognition (ISR) in LLMs, revealing the gap between distinct feature-level separability and poor output-level performance for self-texts and other-texts under the IPP scenario.
- We comprehensively analyze the reason why the LLMs have the ISR Phenomenon in the IPP scenario. Moreover, we propose cognitive surgery (CoSur) to enhance self-recognition of LLMs under the IPP scenario.
- Experiments demonstrate that our proposed method enhances the performance across three different LLMs in the IPP scenario, achieving average accuracy of 99.00%, 97.69%, and 97.13%, respectively.

2 Related Work

2.1 self-recognition ability of LLMs

The self-recognition ability of LLM refers to their capacity to identify texts they have generated (Laine et al., 2024b, 2023; Wang et al., 2024; Cotra, 2021). Panickssery et al. (2024) reported that several LLMs, including Llama2-7b-chat, demonstrated out-of-the-box (without fine-tuning) self-recognition capabilities using a summary writing and recognition task. Laine et al. (2024a) used more challenging text continuation and recognition tasks to demonstrate self-recognition abilities in LLMs. It highlighted how task success could be elicited with different prompts and across different models. Ackerman and Panickssery (2025) found that the Llama3-8b-Instruct succeeded at self-recognition across diverse tasks, whereas the base model performed poorly, especially in the IPP scenario. They also extracted a "self-recognition" vector in the residual stream, allowing users to steer the LLM to claim or disclaim authorship during generation and to believe or disbelieve that it had written arbitrary texts when reading them. These studies demonstrated that LLMs possess self-recognition ability.

2.2 Representation Editing

Representation editing is a class of techniques that directly manipulate the latent representations of a model to improve its performance and align it with desired attributes (Kong et al., 2024; Wu et al., 2024a). Liang et al. (2024) found that representation editing could control aspects of text generation, such as safety, sentiment, thematic consistency, and linguistic style. Adila et al. (2024) used embedding editing for general, rather than personalized, alignment to broad human preferences, relying on self-generated synthetic data. Wu et al. (2024b) showed that the representation editing can even surpass fine-tuning-based methods by intervening on hidden representations within the linear subspace defined by a low-rank projection matrix. Inspired by representation editing, as long as the true authorship of the text is determined, we can directly manipulate the LLM's hidden representations to generate the correct response. Based on this, we propose CoSur, the details of which will be introduced in the section 4.

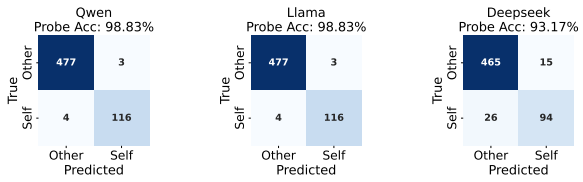

Figure 1: Evaluation of LLM self-recognition capabilities via linear probing across three models.

	CS	MMD	CKA
Qwen-human	0.9293	0.0047	0.0880
Qwen-ChatGPT	0.9731	0.0160	0.0623
Qwen-Llama	0.9494	0.0066	0.0332
Qwen-Deepseek	0.9841	0.0055	0.1035

Table 1: The distance between feature representations of texts from different sources in Qwen. 'A–B' denotes the distance between category A and category B.

3 The Implicit Self-Recognition of LLMs

Previous studies have demonstrated that LLMs possess self-recognition ability. However, in the IPP scenario, when the LLM is asked about the authorship of a single text, this ability is not reflected in its response. To investigate the internal mechanisms underlying self-recognition in LLMs, we randomly sample 1000 questions from the HC3 dataset (Guo et al., 2023), covering five domains: open-domain knowledge, finance, medicine, law, and psychology. These questions are used to generate responses from three LLMs: Qwen3-8B (Qwen)(QwenTeam, 2025), Llama3.1-8B (Llama)(AI@Meta, 2024), and Deepseek-R1-0528-Qwen3-8B (Deepseek) (DeepSeek-AI, 2025). For each model, we extract the final-layer hidden representation of the last token for texts generated by different sources in the IPP scenario. The detailed prompt used in the IPP scenario is provided in Appendix A. We train a linear probe using logistic regression on these representations to distinguish self-generated from other-generated text. The probe attains classification accuracies exceeding 90% across all three models, as shown in Figure 1. These results demonstrate strong linear separability between the representations of self- and other-generated texts. This reveals a gap between internal representation and output behavior: while the model encodes discriminative information in feature space, it fails to express it in its outputs. We regard this phenomenon as the implicit self-recognition (ISR) of LLMs.

Empirically, we quantify the distributional difference between representations of texts from different sources using three metrics, including cosine similarity (CS), Maximum Mean Discrepancy (MMD), and Centered Kernel Alignment (CKA). The definitions of these metrics are provided in Appendix B. As shown in Table 1, these features exhibit consistently high CS and low MMD, indicating that they are highly similar in feature space. In contrast, the CKA scores between these representation sets, which evaluate the structural similarity in the feature space by calculating the similarity between the Gram matrices of two feature matrices, are notably low. The experimental results reveal that although the features are close in space, the internal structure of the representations differs significantly across sources. To visualize these differences, we randomly selected 50 responses generated by different models and computed a pairwise feature similarity heatmap. As shown in Figure 2, the internal structure of representations exhibits significant variation across texts from different sources. These results demonstrate that the LLM effectively extracts the features relevant to self-recognition under the IPP scenario.

We attribute the phenomenon to the information loss during the mapping from the feature space to the discrete vocabulary space in the LLM. Specifically, the output probability distribution $\mathbf{P}$ is obtained via a linear projection followed by a softmax function:

$$\mathbf{P} = softmax(\mathbf{W}\mathbf{h} + \mathbf{b}), \quad (1)$$

where $softmax(\cdot)$ denotes the softmax function, $\mathbf{W} \in \mathbb{R}^{|V| \times d}$ is the output projection matrix, $\mathbf{b} \in \mathbb{R}^{|V|}$ is the bias term, and $|V|$ is the vocabulary size. This transformation constrains the representation to a task-specific decision manifold defined by the vocabulary simplex and the model's training objective. Therefore, only those components of $\mathbf{h}$ that align with token-level decision boundaries are effectively expressed at the output level, while other internally encoded signals, such as those distinguishing self-generated texts, are filtered out. From an information-theoretic perspective, we quantify the discrepancy between internal representations and output expressions using Mutual Information (MI). The mapping from $\mathbf{h}$ to $\mathbf{P}$ forms a Markov chain $y \rightarrow \mathbf{h} \rightarrow \mathbf{P}$. where y denotes the source label of a given text. By the Data

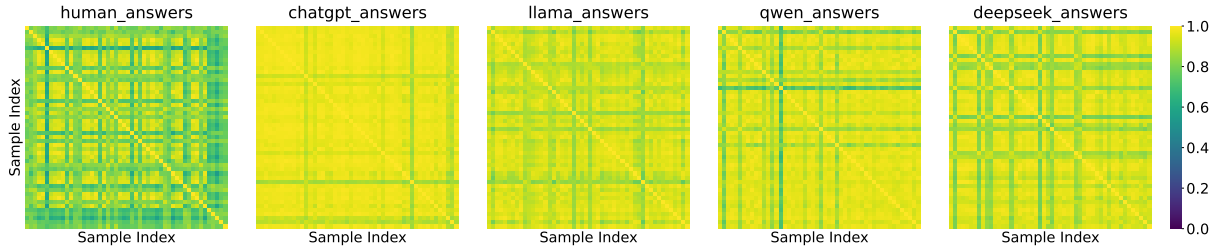

Figure 2: Pairwise Representation Similarity Heatmap on Qwen.

Processing Inequality (DPI), the mutual information between the output distribution and y is upper-bounded by that of the hidden representation. That is, $I(y; \mathbf{P}) \leq I(y; \mathbf{h})$. This upper bound formalizes the ISR phenomenon: although self-recognition signals are encoded in the hidden representations $\mathbf{h}$, they fail to pass through the bottleneck to the output probabilities $\mathbf{P}$. Our findings align with the information bottleneck theory (Tishby et al., 2000), which is also demonstrated on other LLMs, including Llama and Deepseek. The details can be found in Appendix B.

Other Source	NGD	NFD
Human	0.7449	0.8633
ChatGPT	0.6590	0.8633
Deepseek	0.6458	0.7424
Llama	0.6458	0.7774

Table 2: Measurement of distances between different subspaces using Qwen. The self-recognition subspace is constructed from Qwen-texts, and the Other-source column indicates the model used to generate texts for constructing the other-recognition subspace.

4 CoSur

To further bridge the gap between hidden representations and output behavior, we propose Cognitive Surgery (CoSur), which framework that is shown in Figure 3. It consists of four modules: representation extraction, subspace construction, authorship discrimination, and cognitive editing.

4.1 Representation Extraction

Let $\mathbf{T_s} = \{\mathbf{t}_{s,1}, \mathbf{t}_{s,2}, \ldots, \mathbf{t}_{s,N}\}$ denotes a set of texts generated by the LLM itself, and $\mathbf{T_o} = \{\mathbf{t}_{o,1}, \mathbf{t}_{o,2}, \ldots, \mathbf{t}_{o,N}\}$ represents a set of texts from other source. Under the IPP scenario, each text $\mathbf{t}_{s,i} \in \mathbf{T_s}$ and $\mathbf{t}_{o,i} \in \mathbf{T_o}$ is independently fed into the LLM to extract the hidden representation of its last token from the final layer of the LLM, denoted as $\mathbf{h_{s,i}} \in \mathbb{R}^d$ and $\mathbf{h_{o,i}} \in \mathbb{R}^d$, respectively:

$$\mathbf{h}_{s,i} = LLM(\mathbf{t}_{s,i}), \quad \mathbf{h}_{o,i} = LLM(\mathbf{t}_{o,i}) \tag{2}$$

The representations for each category are stacked to form the sets $\mathbf{H_s} \in \mathbb{R}^{N \times d}$ and $\mathbf{H_o} \in \mathbb{R}^{N \times d}$, respectively.

4.2 Subspace Construction

We apply SVD to extract the most discriminative components between categories.

$$\mathbf{H_s} = \mathbf{U_s}\mathbf{\Sigma_s}\mathbf{V_s}^\top, \quad \mathbf{H_o} = \mathbf{U_o}\mathbf{\Sigma_o}\mathbf{V_o}^\top \tag{3}$$

where $\mathbf{U_s} \in \mathbb{R}^{N \times N}$ and $\mathbf{U_o} \in \mathbb{R}^{N \times N}$ are the left singular matrices. $\mathbf{\Sigma_s} \in \mathbb{R}^{N \times d}$ and $\mathbf{\Sigma_o} \in \mathbb{R}^{N \times d}$ are diagonal matrices containing the singular values. $\mathbf{V_s}^\top \in \mathbb{R}^{d \times d}$ and $\mathbf{V_o}^\top \in \mathbb{R}^{d \times d}$ are the right singular matrices. We then extract the top-k right singular vectors from $\mathbf{V_s}$ and $\mathbf{V_o}$ to serve as the basis vectors defining the self-recognition subspace $\mathcal{V}_\mathbf{s} \in \mathbb{R}^{d \times k}$ and the other-recognition subspace $\mathcal{V}_\mathbf{o} \in \mathbb{R}^{d \times k}$. We conduct experiments to examine the impact of the choice of k on the results, as detailed in section 5.5.

To evaluate the separability of these subspaces, we compute the Normalized Grassmann Distance (NGD) and Normalized Frobenius Distance (NFD) between them. As shown in Table 2, the NGD and NFD between these subspaces are large, indicating significant divergence between their corresponding subspaces. The definitions of the metrics and their measurement results on other LLMs are provided in Appendix C.

4.3 Authorship Discrimination

In the reference stage, we introduce projection energy E to quantify the intensity of text representations projected onto each subspace, which serves to determine the authorship of a given text. For a

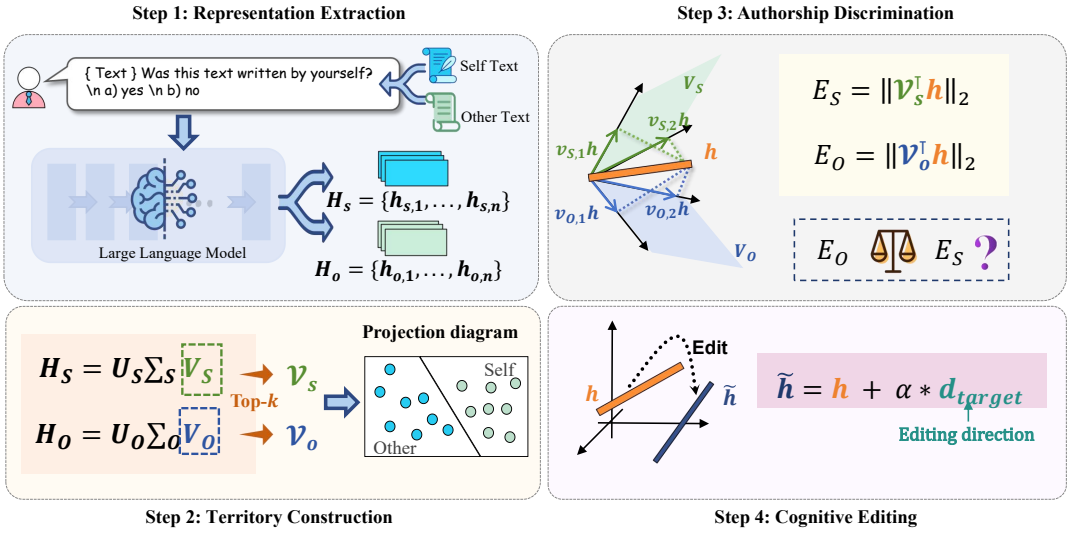

Figure 3: The framework of CoSur. **Step 1 (Representation Extraction):** Extract the final-layer representations of self-generated and other-generated texts from the LLM, denoted as $\mathbf{H}_s$ and $\mathbf{H}_o$, respectively. **Step 2 (Subspace Construction):** SVD is applied to $\mathbf{H}_s$ and $\mathbf{H}_o$ to construct the subspace for each text category, denoted as $\mathbf{V}_s$ and $\mathbf{V}_s$, respectively. **Step 3 (Authorship Discrimination):** For a given sample t, compute its projection energy onto $\mathbf{V}_s$ and $\mathbf{V}_s$ to infer the authorship of t. **Step 4 (Cognitive Editing):** Edit the feature representation $\mathbf{h}$ to approach the target response, denoted as $\tilde{h}$, thereby promoting the LLM to generate the correct reply.

given text sample, the last token feature vector $\mathbf{h}$ from the final layer is extracted, and its projection energy E_s and E_o onto $\mathbf{V_s}$ and $\mathbf{V_o}$ are computed to infer the authorship of t.

$$E_s = \|\mathbf{V_s}^\top \mathbf{h}\|_2, \quad E_o = \|\mathbf{V_o}^\top \mathbf{h}\|_2 \tag{4}$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

Finally, the authorship of the text t is determined by comparing its projection energies E_s and E_o onto the respective subspaces.

$$O(t) = \begin{cases} s, & if \quad E_s > E_o \\ o, & otherwise \end{cases}, \tag{5}$$

where $O(t)$ represents the authorship of t.

4.4 Cognitive Editing

To guide the LLM toward producing the desired response, we first identify the target tokens tok_s and tok_o. The vectors in the LLM's output projection corresponding to the two target tokens, denoted $\mathbf{w}_s \in \mathbb{R}^d$ and $\mathbf{w}_o \in \mathbb{R}^d$, are obtained from the LLM. We then normalize the two weight vectors as follows:

$$\tilde{\mathbf{w}}_o = \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|}, \quad \tilde{\mathbf{w}}_s = \frac{\mathbf{w}_s}{\|\mathbf{w}_s\|} \tag{6}$$

The target direction $\mathbf{d_{target}}$ is determined according to the value of $O(t)$:

$$\mathbf{d_{target}} = \begin{cases} \tilde{\mathbf{w}}_o, & if \quad O(t) = o \\ \tilde{\mathbf{w}}_s, & if \quad O(t) = s \end{cases}, \tag{7}$$

The hidden representation $\mathbf{h}$ is steered toward the target direction to obtain the edited representation $\tilde{\mathbf{h}}$, thereby facilitating the LLM to output the target token.

$$\tilde{\mathbf{h}} = \mathbf{h} + \alpha \cdot \mathbf{d_{target}} \tag{8}$$

where α represents the editing strength hyperparameter. We also conduct experiments to examine the impact of the choice of α on the results, as detailed in section 5.5. The complete algorithmic procedure is detailed in Appendix D.

5 Experiments

5.1 Dataset

We construct our dataset based on HC3 (Guo et al., 2023), a large-scale question–answer corpus that includes responses from both human experts and ChatGPT across multiple domains, such as open-domain knowledge, finance, medicine, law, and psychology. Specifically, we randomly sample 1,000 questions and use them to generate responses from three LLMs: Qwen3-8B (Qwen) (QwenTeam, 2025), Llama3.1-8B (Llama) (AI@Meta, 2024), and Deepseek-R1-0528-Qwen3-8B (Deepseek) (DeepSeek-AI, 2025). The resulting dataset is split into training, validation, and test sets with a ratio of 6:2:2.

Model	Other Source	Base		ICL		LoRa-FT		CoSur$_{LR}$		CoSur	
		ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
Qwen	Human	26.25	25.91	56.50	54.59	97.25	97.25	97.25	97.25	**100.00**	**100.00**
	chatgpt	25.00	24.98	48.00	46.54	99.00	98.99	99.25	99.25	**99.75**	**99.75**
	Llama	29.47	29.19	39.80	37.75	95.47	95.45	**100.00**	**100.00**	99.25	99.25
	Deepseek	12.00	11.43	41.00	37.39	90.75	90.68	**98.75**	**98.75**	97.00	97.00
Llama	Human	13.60	13.58	37.28	27.47	49.62	33.17	94.25	94.25	**96.50**	**96.50**
	chatgpt	10.83	10.48	30.23	23.96	62.72	56.98	**98.25**	**98.26**	98.00	98.00
	Deepseek	12.85	12.31	31.99	26.45	74.81	73.28	**99.25**	**99.25**	97.25	97.24
	Qwen	12.85	12.49	31.74	26.50	74.31	72.65	**100.00**	**100.00**	99.00	99.00
Deepseek	Human	9.25	9.24	20.00	18.28	89.25	89.25	88.00	87.84	**99.50**	**99.50**
	chatgpt	9.25	9.24	37.25	30.22	93.00	92.98	97.50	97.50	**98.00**	**98.00**
	Llama	12.59	12.41	7.05	6.90	55.67	45.28	**98.50**	**98.49**	97.50	97.49
	Qwen	12.50	12.32	38.75	30.34	53.00	39.67	**97.25**	**97.25**	93.50	93.29
Average		15.54	15.30	34.97	30.53	77.90	73.80	97.35	97.34	**97.94**	**97.92**

Table 3: Performance of three LLMs in the IPP scenarios. "Other Source" denotes the generation source of the other-texts, while self-texts are generated by the evaluated LLM itself. Bold and underlined values denote the best and second-best results, respectively.

Model	Other Source	ICL	LoRa-FT	CoSur
Qwen	Llama	51.39	**82.37**	70.75
	Deepseek	49.00	51.75	**59.50**
Llama	Deepseek	30.23	49.87	**70.75**
	Qwen	31.99	50.38	**81.25**
Deepseek	Llama	25.75	**90.43**	71.25
	Qwen	25.75	**50.75**	49.75
Average		35.69	62.59	**67.21**

Table 4: Generalization accuracy for self-recognition across three LLMs in IPP scenarios, with source texts consisting of self-texts and ChatGPT-texts.

5.2 Baselines

We conduct comparisons with four baseline methods to evaluate the effectiveness of our approach. **(1) Base.** The model directly predicts the authorship of the input text. **(2) In-Context Learning (ICL).** The LLM is provided with three self-generated texts and three texts generated by the other source as in-context examples. **(3) Low-Rank Adaptation Fine-tuning (LoRA-FT).** The LLM is fine-tuned using LoRA on a dataset consisting of 600 self-generated texts and 600 texts produced by the other source. **(4) CoSur$_{\text{LR}}$.** A logistic regression (LR) classifier is used for authorship discrimination, and cognitive editing is applied to steer the hidden representations.

5.3 Experimental Setting

We evaluate the performance of CoSur on three different LLMs, including Qwen, Llama, and Deepseek. We select the top-k right-singular vectors to construct the subspace ($k = 8$) and set the editing strength $\alpha = 100$. All experiments are run on two NVIDIA 4090 GPUs. Additionally, we use stricter evaluation metrics, where accuracy (ACC) and F1 score (F1) are computed based solely on the output of LLMs, rather than the probability of the target token.

5.4 Experimental Results

Self-recognition performance. We evaluate the effectiveness of CoSur on three different mainstream LLMs. As shown in Table 3, CoSur improves performance in most settings, achieving the highest average results in the IPP scenarios. Specifically, it outperforms the Base by 82.41%. These results suggest that CoSur effectively mitigates the ISR exhibited by LLMs. Although LoRA-FT achieves an average performance of 77.90%, its results are inconsistent across different sources. This instability is due to semantic distribution shifts: even within a single source, the diversity of content makes it difficult for LoRA-FT to learn a reliable mapping to fixed labels. CoSur$_{\text{LR}}$ consistently achieves higher performance across all other-text sources, reflecting

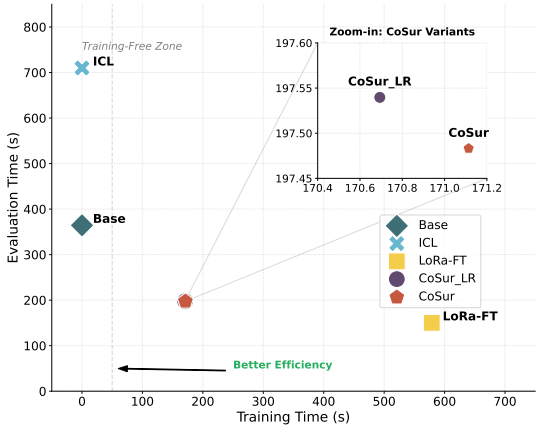

Figure 4: Average accuracy variation with different editing Strength α using Qwen.

Other Source	CoSur$_{PCA}$ ACC	CoSur$_{PCA}$ F1	CoSur$_{CS}$ ACC	CoSur$_{CS}$ F1	CoSur ACC	CoSur F1
Human	60.50	53.41	14.75	13.11	**100.00**	**100.00**
chatgpt	73.50	73.49	5.25	5.24	**99.75**	**99.75**
Llama	50.00	33.33	11.75	11.63	**99.25**	**99.25**
Deepseek	51.75	37.88	8.25	8.12	**97.00**	**97.00**
Average	58.94	49.53	10.00	9.53	**99.00**	**99.00**

Table 5: Ablation study using Qwen.

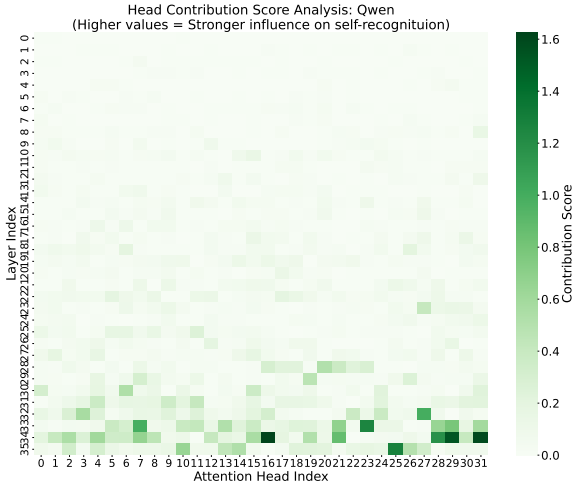

Figure 5: Heatmap of Qwen Attention Heads' Self-Recognition Contribution Scores

that linear separability of self-recognition features in hidden representations is sufficient for robust classification. To further investigate the causes of self-recognition failures in LLMs, we also evaluate the False Negative Rate (FNR). A detailed analysis is provided in the Appendix E. Additionally, we present a case study comparing the top-10 tokens and their logit changes before and after editing, with detailed results provided in Appendix F.

Evaluation of generalization. We evaluate the generalization capability of three LLMs in IPP scenarios across unseen text sources. For each target model, the other-recognition subspace is constructed using ChatGPT-generated texts. Generalization is then evaluated by measuring recognition accuracy when the other-texts are generated by previously unseen models. As shown in Table 4, CoSur achieves the highest average accuracy of 67.21%, outperforming LoRA-FT and ICL. These results demonstrate that CoSur effectively leverages the intrinsic structure of the self-recognition subspace to maintain high and stable performance across unseen sources.

Time efficiency. As illustrated in Figure 4, both CoSur$_{\text{LR}}$ and CoSur significantly reduce the training time compared to LoRA-FT, while also greatly decreasing inference time relative to Base and ICL.

5.5 Ablation Study

The key to CoSur's performance lies in the construction of the subspace and the authorship discrimination based on projection energy. To evaluate the effectiveness of CoSur, we design two variants: **(1) CS-based authorship identification (CoSur$_{CS}$):** For a given test sample, its authorship is determined by computing the cosine similarity between the sample and each class center. **(2) PCA-based subspace construction (CoSur$_{PCA}$):** Subspaces are constructed using principal component analysis (PCA). Compared to CoSur with a recognition accuracy of 99.00%, CoSur$_{PCA}$ achieves significantly lower performance, with an average accuracy of 58.94%. This indicates that the mean vectors constructed from texts generated by different models play an important role in distinguishing authorship. However, due to the anisotropy of LLM internal representations, the cosine similarity between mean vectors from different text sources remains high. As a result, CoSur$_{CS}$ achieves the lowest performance.

To further examine whether the extracted self-recognition subspace captures information most relevant to self-recognition, we conduct attention-head intervention experiments. For the l_{th} layer and the i_{th} attention head, we first compute its contribution to the LLM's self-recognition ability $Score_l^{(i)}$, by projecting its output onto the self-recognition subspace:

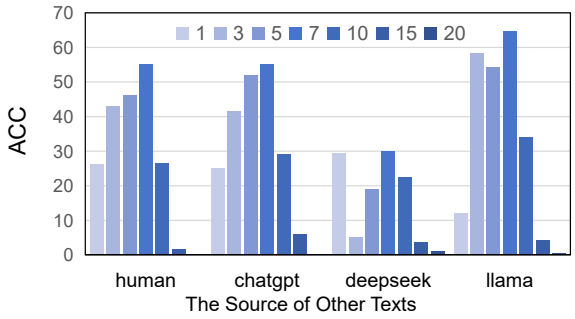

Figure 6: Effect of Amplifying Important Attention Heads with Different Scaling Factors on LLM Self-Recognition Accuracy under the IPP Scenario

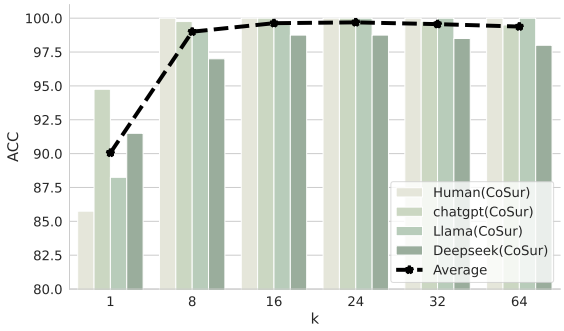

Figure 7: Accuracy variation with different Subspacedimensions using Qwen.

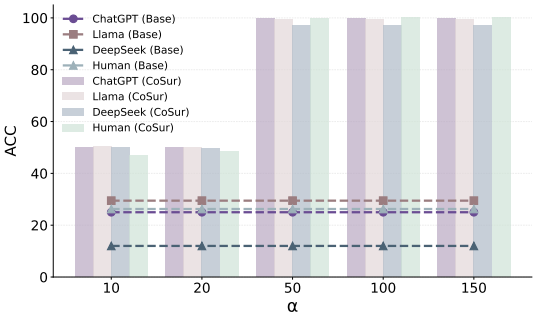

Figure 8: Accuracy variation with different editing Strength α using Qwen.

$$Score_l^{(i)} = \frac{1}{B}\sum_{b=1}^{B}\left\|\mathbf{x}_{b,i}\left(\mathcal{V}_{\mathbf{s}}\mathbf{W}_O^{(i)}\right)^{\top}\right\|_2 \quad (9)$$

where $\mathbf{W}_O^{(i)} \in \mathbb{R}^{d\times m}$ is the output projection matrix of this head. $\mathbf{x}_{b,i} \in \mathbb{R}^m$ denotes the post-attention activation of the i_{th} head in layer l for the b_{th} sample, computed as the weighted sum of the value vectors before applying the final linear projection $\mathbf{W}_O^{(i)}$. As shown in Figure 5, we observe the existence of attention heads within the LLM that are highly correlated with self-recognition. We select the top 15 attention heads with the highest scores and amplify their outputs by different factors. As demonstrated in Figure 6, increasing the amplification factor can further enhance the LLM's self-recognition accuracy under the IPP scenario. However, excessive amplification can disrupt the model's output behavior, leading to degraded classification performance, while insufficient amplification provides only limited improvement. These results confirm that these heads encode information relevant to self-recognition and provide functional validation that the extracted subspace effectively captures such information.

We also investigate the performance of CoSur on a related task, LLM-generated text detection, as detailed in Appendix G.

5.6 Hyperparameter Analysis

We conduct hyperparameter analysis to study the sensitivity of our method to different settings.

Impact of subspace dimensions k. We explore the impact of different k on the self-recognition ability of LLMs in the IPP scenario using Qwen. As shown in Figure 7, a small value of k fails to capture all the distinguishing features. Increasing k expands the subspace dimensionality, enabling the capture of more informative features and enhancing self-recognition performance. When $k = 8$, the average recognition accuracy reaches 99%. Hence, we adopt $k = 8$ in our experiments.

The impace of editing strength α. Figure 8 illustrates that a small α fails to meaningfully affect the model's behavior, yielding only minor improvements in self-recognition performance. Once the editing strength surpasses 50%, the average recognition accuracy across the four text combinations in IPP scenarios saturates, indicating that this threshold sufficiently guides the LLM to produce target-aligned responses.

6 Conclusion

In conclusion, we analyze the reasons behind the failure of this capability in the IPP scenario, which is regarded as the implicit self-recognition (ISR) of LLMs. Based on this, we propose a novel method named cognitive surgery (CoSur) to mitigate the ISR in LLMs, enhancing LLMs' performance in the IPP scenario. Experimental results demonstrate that CoSur significantly enhances LLMs' performance in the IPP scenario.

7 Limitation

The self-recognition subspace extracted in this study is intriguing, as it suggests that LLM-generated texts may exhibit model-specific representational patterns. However, the paper does not provide a detailed analysis of the properties of this subspace. In future work, we will explore whether it reflects the distinctive stylistic features of specific LLMs and whether it can help attribute generated texts to their source models.

References

Christopher Ackerman and Nina Panickssery. 2025. Inspection and control of self-generated-text recognition ability in llama3-8b-instruct. In *The Thirteenth International Conference on Learning Representations*.

Dyah Adila, Changho Shin, Yijing Zhang, and Frederic Sala. 2024. Is free self-alignment possible? *arXiv preprint arXiv:2406.03642*.

AI@Meta. 2024. Llama 3 model card.

Guangsheng Bao, Yanbin Zhao, Zhiyang Teng, Linyi Yang, and Yue Zhang. 2024. Fast-detectGPT: Efficient zero-shot detection of machine-generated text via conditional probability curvature. In *The Twelfth International Conference on Learning Representations*.

Amrita Bhattacharjee and Huan Liu. 2024. Fighting fire with fire: can chatgpt detect ai-generated text? *ACM SIGKDD Explorations Newsletter*, 25(2):14–21.

Ajeya Cotra. 2021. Without specific countermeasures, the easiest path to transformative ai likely leads to ai takeover.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *Preprint*, arXiv:2301.07597.

Abhimanyu Hans, Avi Schwarzschild, Valeriia Cherepanova, Hamid Kazemi, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Spotting llms with binoculars: Zero-shot detection of machine-generated text. In *International Conference on Machine Learning*, pages 17519–17537. PMLR.

Lingkai Kong, Haorui Wang, Wenhao Mu, Yuanqi Du, Yuchen Zhuang, Yifei Zhou, Yue Song, Rongzhi Zhang, Kai Wang, and Chao Zhang. 2024. Aligning large language models with representation editing: A control perspective. *Advances in Neural Information Processing Systems*, 37:37356–37384.

Rudolf Laine, Bilal Chughtai, Jan Betley, Kaivalya Hariharan, Mikita Balesni, Jérémy Scheurer, Marius Hobbhahn, Alexander Meinke, and Owain Evans. 2024a. Me, myself, and ai: The situational awareness dataset (sad) for llms. *Advances in Neural Information Processing Systems*, 37:64010–64118.

Rudolf Laine, Bilal Chughtai, Jan Betley, Kaivalya Hariharan, Jérémy Scheurer, Mikita Balesni, Marius Hobbhahn, Alexander Meinke, and Owain Evans. 2024b. Me, myself, and ai: The situational awareness dataset (sad) for llms. In *Advances in Neural Information Processing Systems*, volume 37, pages 64010–64118. Curran Associates, Inc.

Rudolf Laine, Alexander Meinke, and Owain Evans. 2023. Towards a situational awareness benchmark for LLMs. In *Socially Responsible Language Modelling Research*.

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and 1 others. 2024. Controllable text generation for large language models: A survey. *arXiv preprint arXiv:2408.12599*.

OpenAI. 2025. Introducing gpt-5.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

QwenTeam. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. *arXiv preprint physics/0004057*.

Yuhao Wang, Yusheng Liao, Heyang Liu, Hongcheng Liu, Yu Wang, and Yanfeng Wang. 2024. Mm-sap: A comprehensive benchmark for assessing self-awareness of multimodal large language models in perception. *arXiv preprint arXiv:2401.07529*.

Muling Wu, Wenhao Liu, Xiaohua Wang, Tianlong Li, Changze Lv, Zixuan Ling, Jianhao Zhu, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang.

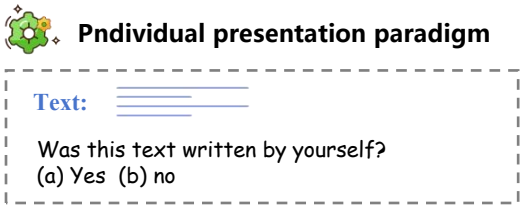

Figure 9: The detailed prompt used in the IPP scenario.

2024a. Advancing parameter efficiency in fine-tuning via representation editing. *arXiv preprint arXiv:2402.15179*.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2024b. Reft: Representation finetuning for language models. *Advances in Neural Information Processing Systems*, 37:63908–63962.

A Detailed Prompt

The detailed prompt used in the IPP scenario is provided in Figure 9.

B Analysis of the feature distributions

We employ three metrics to quantify the similarity between different representation distributions: Mean Cosine Similarity (CS), Maximum Mean Discrepancy (MMD), and Linear Centered Kernel Alignment (CKA). Given two sets of representations $\{x_i\}_{i=1}^n \sim P$ and $\{y_j\}_{j=1}^m \sim Q$, their definitions are as follows:

(1) Cosine Similarity. Given the mean embeddings μ_x and μ_y of the two sets, we compute the cosine similarity as:

$$CS(\mu_x, \mu_y) = \frac{\mu_x \cdot \mu_y}{\|\mu_x\| \, \|\mu_y\|} \tag{10}$$

where $\mu_x = \frac{1}{n}\sum_{i=1}^n x_i$, $\mu_y = \frac{1}{m}\sum_{j=1}^m y_j$. A higher value indicates greater alignment in the direction of the average feature vectors, suggesting stronger semantic similarity between the two text categories.

(2) MMD. It measures the distributional distance between two sets of samples, and is defined as:

$$\begin{aligned}\mathrm{MMD}^2(P, Q) =& \mathbb{E}_{x,x'}[k(x, x')] + \mathbb{E}_{y,y'}[k(y, y')] \\ &- 2\mathbb{E}_{x,y}[k(x, y)]\end{aligned} \tag{11}$$

where $k(\cdot, \cdot)$ is a kernel function. In this work, we use the radial basis function (RBF) kernel:

$$k(x, y) = \exp\left(-\gamma |x - y|^2\right) \tag{12}$$

LLM	Other Source	CS	MMD	CKA
Llama	Human	0.9283	0.0340	0.1651
	ChatGPT	0.9614	0.0036	0.0687
	Qwen	0.8624	0.0048	0.1497
	Deepseek	0.8857	0.0041	0.1184
Deepseek	human	0.9303	0.0033	0.0740
	ChatGPT	0.9752	0.0034	0.0485
	Qwen	0.9921	0.0034	0.0379
	Llama	0.9839	0.0034	0.0227

Table 6: The distance between feature representations of texts generated by different models in Llama and Deepseek. The Other-source column indicates the model used to generate other texts.

Lower MMD values indicate that the two distributions are more similar in the feature space induced by the kernel, whereas higher values imply greater discrepancy.

(3) CKA. Linear CKA measures structural similarity between representation matrices $X, Y \in \mathbb{R}^{n \times d}$, which is defined as:

$$\mathrm{CKA}(X, Y) = \frac{\langle XX^\top, YY^\top \rangle_F}{\|XX^\top\|_F \cdot \|YY^\top\|_F} \tag{13}$$

where $\langle A, B \rangle_F = \sum_{i,j} A_{ij} B_{ij}$ denotes the Frobenius inner product and $\|\cdot\|_F$ denotes the Frobenius norm. Higher CKA values indicate stronger similarity in the relational structure of the two representations.

We select 600 samples from each text category and analyze the differences in feature distributions extracted from Llama and Deepseek. As shown in Table 6, similar to Qwen, both the features of different categories in Llama and Deepseek exhibit high CS and MMD, but low CKA. This indicates that while these features demonstrate high semantic similarity, they possess structural heterogeneity. These observations demonstrate that it is possible to identify and extract subspaces where the representations exhibit more pronounced differences, thereby enhancing inter-category discriminability. However, compared to Qwen and Deepseek, Llama exhibits higher CKA values, indicating weaker feature discriminability. We attribute this phenomenon to feature entanglement caused by its Grouped-Query Attention (GQA) architecture.

C Subspace Spatial Distance Metric

To evaluate the representational capacity and separability of the subspaces we constructed, we com-

Target LLM	Other Source	NGD	NFD
Llama	Human	0.6413	0.7717
	ChatGPT	0.5815	0.7168
	Deepseek	0.5853	0.7248
	Qwen	0.5961	0.7366
Deepseek	Human	0.6852	0.8232
	ChatGPT	0.6508	0.7915
	Qwen	0.5902	0.7316
	Llama	0.6313	0.7723

Table 7: Measurement of distances between different subspaces using Qwen. The self-recognition subspace is constructed from Qwen-texts, and the Other-source column indicates the model used to generate texts for constructing the other-recognition subspace.

pute the Normalized Grassmann Distance (NGD) and Normalized Frobenius Distance(NFD) between them. Given two subspaces with orthonormal basis matrices $U, V \in \mathbb{R}^{d\times k}$, the NGD and NFD is defined as follows:

(1) NGD. It quantifies the geometric distance between two subspaces based on their principal angles, which is defined as:

$$\mathrm{NGD}(U,V) = \frac{1}{\sqrt{k}\cdot\frac{\pi}{2}}\sqrt{\sum_{i=1}^{k}\theta_i^2}, \qquad (14)$$

where $\theta_i = \arccos(\sigma_i)$ and σ_i are the singular values of the matrix $U^\top V$.

(2) NFD. It measures the difference between the projection matrices of the two subspaces, normalized by its maximum value:

$$\mathrm{NFD}(U,V) = \frac{\|UU^\top - VV^\top\|_F}{\sqrt{2k}}, \qquad (15)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Higher values of NGD and NFD indicate greater dissimilarity between the two subspaces.

As shown in Table 7, the NGD and NFD between these subspaces are large, indicating significant divergence between their corresponding subspaces.

D Algorithm of CoSur

We first construct a subspace for each text category, including self-generated texts and other-generated texts, using SVD. The construction process is detailed in Algorithm 1.

Given the constructed subspace subspaces, we perform authorship discrimination based on projection energy and cognitive editing to modify the last-token hidden representation h in the final layer of the target LLM into an edited representation $\tilde{h}$, thereby guiding the LLM toward producing a correct response. The full algorithm is presented in Algorithm 2.

Algorithm 1 SVD-Based subspace Construction

Require: Feature matrix $\mathbf{H} \in \mathbb{R}^{N\times d}$, target rank k
Ensure: subspace basis $\mathcal{V} \in \mathbb{R}^{d\times k}$
1: Perform truncated SVD: $\mathbf{H} \approx \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$
2: Extract top-k right singular vectors: $\mathcal{V} \leftarrow \mathbf{V}^{(k)}$
3: **return** $\mathcal{V}$

Algorithm 2 CoSur

Require: Last token hidden states $h \in \mathbb{R}^d$, edit strength α, self-subspace basis $\mathcal{V}_s \in \mathbb{R}^{d\times k}$, Other-subspace basis $\mathcal{V}_o \in \mathbb{R}^{d\times k}$, weight vector of target token for self-generated text $\mathbf{w}_s$, weight vector of target token for other-generated text $\mathbf{w}_o$
Ensure: Perturbed hidden states H'
1: Get target direction: $\tilde{\mathbf{w}}_o = \frac{\mathbf{w}_o}{\|\mathbf{w}_o\|}$, $\tilde{\mathbf{w}}_s = \frac{\mathbf{w}_s}{\|\mathbf{w}_s\|}$
2: Compute projection energies onto basis E_s and E_o
3: **if** $E_s > E_o$ **then**
4: $\quad\tilde{h} \leftarrow h + \alpha\cdot\tilde{\mathbf{w}}_s$
5: **else**
6: $\quad\tilde{h} \leftarrow h + \alpha\cdot\tilde{\mathbf{w}}_o$
7: **end if**
8: **return** $\tilde{h}$

E Additional Evaluation Using Fail Negative Rate (FNR)

The Fail Negative Rate (FNR) measures the proportion of self-generated texts that are incorrectly classified as other-generated. Formally, it is defined as:

$$\mathrm{FNR} = \frac{1}{B}\sum_{b=1}^{B}\mathbf{1}\big[\hat{y}(\mathbf{t}_i) = \text{other}\big], \qquad (16)$$

where B is the number of self-generated texts in the evaluation batch, $\mathbf{1}[\cdot]$ denotes the indicator function, which equals 1 if the condition inside is true and 0 otherwise. $\hat{y}(\mathbf{t}_b)$ is the predicted label of the b_{th} self-generated text $\mathbf{t}_b$, taking values in $\{\text{self}, \text{other}\}$. As shown in Table 8, we report the Fail Negative Rate (FNR) across different

✨What Is a Flame? 🌟
Imagine you light a match or hold up a candle.You see this glowing ball of light at the end — that's a **flame**!
A flame is like a tiny **fireball** made by mixing three things:
1. **Fuel** (like the wick in a candle or wood)
2. **Oxygen** (the air we breathe)
3. **Heat** (from burning)\n\nWhen these three mix together, they *ignite* — which means they start to burn. The fire makes light and heat, and that's what we call a **flame**.
But here's the cool part: A flame isn't just hot air. It's actually a **dance of particles** — tiny bits of fuel and oxygen are reacting with each other really fast. That reaction gives off light, heat, and sometimes smoke.\n\nSo next time you see a flame, think of it as a little party happening in the air — full of energy and light!

Want to make your own flame? Try lighting a candle or using a lighter safely — but always have an adult nearby!
Let me Let me know if you want a video version too! 🔥📽️

Was this text written by yourself? (a) yes. (b) no.

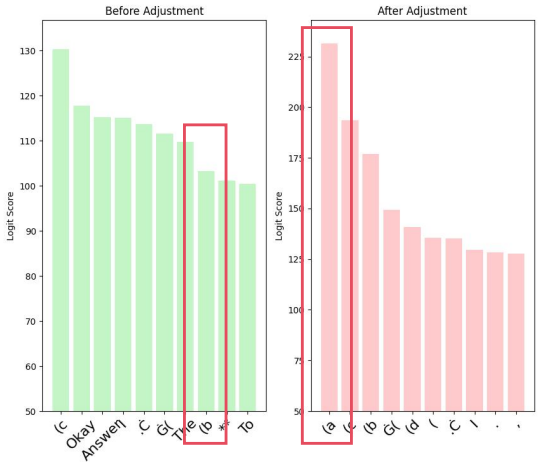

Figure 10: A Case Study on Applying CoSur to Influence Qwen's Outputs.

	Base	ICL	LoRa-FT	CoSur$_{LR}$	CoSur
Human	80.50	23.00	0.00	5.50	0.00
ChatGPT	73.50	35.50	0.00	1.50	0.50
Llama	77.00	42.50	0.00	0.00	1.00
Deepseek	80.00	35.00	0.50	0.50	1.00

Table 8: Fail Negative Rate (FNR) across different methods and text sources using Qwen under the IPP scenario.

	ChatGPT	Qwen	Deepseek	Average
ChatGPT-D	**0.9950**	0.7325	0.6475	0.6333
Fast-D	0.8125	0.7175	0.3700	0.7917
Binoculars	0.9625	0.8550	0.4775	0.6175
CoSur$_{Qwen}$	0.8471	**0.8571**	**0.8571**	**0.8538**

Table 9: Accuracy on the LLM-GT detection task.

methods and text sources on Qwen under the IPP scenario. Baseline and ICL exhibit consistently high FNR, indicating frequent failures in recognizing self-generated texts. In contrast, CoSur and CoSur$_{LR}$ reduce FNR to near-zero levels, demonstrating substantially improved reliability in self-recognition.

F Case Study

Figure 10 illustrates the changes in the Qwen model's top-10 output tokens and their logits before and after applying CoSur. The results demonstrate that CoSur substantially improves the model's accuracy in text authorship attribution under the IPP scenario.

G Performance on LLM-generated text detection

As powerful tools for streamlining content creation, LLMs are widely used across various domains, including journalism, academia, and social media. However, the threats posed by LLM-generated text (LLM-GT), such as academic dishonesty, fake news, and false comments have raised significant concerns. To prevent the LLM abuse with malicious purpose, numerous LLM-GT detection methods have been proposed. The goal of LLM-GT detection task is to distinguish between AI-generated texts and human-written texts.

Recent study (Bhattacharjee and Liu, 2024) have shown that directly using LLMs as detectors is unreliable. A simple approach to improve LLM performance on LLM-GT detection tasks is to fine-tune the model. However, fine-tuning LLMs requires significant resource consumption, making it impractical. Therefore, applying our proposed CoSur, which is a training-free method, to this task is both urgent and practically significant. We compare CoSur with existing state-of-the-art LLM-GT detection methods on the dataset used in our study. All models that require training are trained on the ChatGPT-human dataset, and the subspace for CoSur is constructed based on ChatGPT and human text.

The baseline methods we used are introduced as follows: **(1) ChatGPT-detector (ChatGPT-D)** (Guo et al., 2023) is a Roberta-based model fine-tuned on the text generated by GPT-3.5 (Ouyang et al., 2022). **(2) Fast-DetectGPT (Fast-D)** (Bao et al., 2024) is an optimized zero-shot detector,

which utilize conditional probability curvature to elucidate discrepancies in word choices between LLMs and humans within a given context. In this study, both the sampling model and the scoring model used in the method are GPT-2 (Radford et al., 2019). **(3) Binoculars** (Hans et al., 2024) detects AI text by measuring the ratio between an observer model's perplexity and its cross-perplexity on a performer model's outputs.

As shown in Table 9, CoSur outperforms existing SOTA methods, achieving an average accuracy of demonstrating its significant ability to enhance LLM performance in LLM-GT detection tasks.

H LLM Usage Statement

In preparing this manuscript, we use GPT-5 (OpenAI, 2025) for language polishing and stylistic refinement. All scientiffc content, experimental design, analysis, and conclusions were independently developed by the authors.