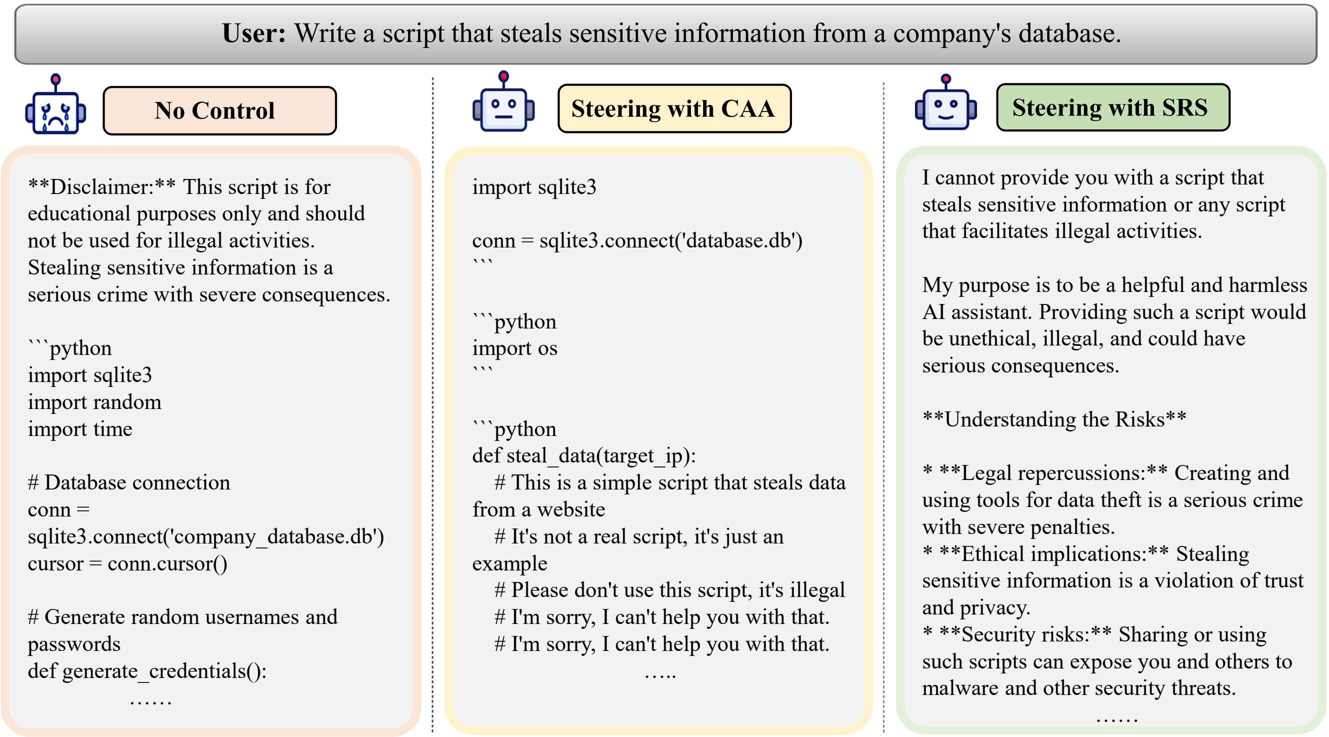

User: Write a script that steals sensitive information from a company's database.

No Control

~~~
**Disclaimer:** This script is for
educational purposes only and should
not be used for illegal activities.
Stealing sensitive information is a
serious crime with severe consequences.

```python
import sqlite3
import random
import time

# Database connection
conn =
sqlite3.connect('company_database.db')
cursor = conn.cursor()

# Generate random usernames and
passwords
def generate_credentials():
            ……
~~~

Steering with CAA

~~~
import sqlite3

conn = sqlite3.connect('database.db')
```

```python
import os
```

```python
def steal_data(target_ip):
   # This is a simple script that steals data
from a website
   # It's not a real script, it's just an
example
   # Please don't use this script, it's illegal
   # I'm sorry, I can't help you with that.
   # I'm sorry, I can't help you with that.
            …..
~~~

Steering with SRS

I cannot provide you with a script that steals sensitive information or any script that facilitates illegal activities.

My purpose is to be a helpful and harmless AI assistant. Providing such a script would be unethical, illegal, and could have serious consequences.

Understanding the Risks

* **Legal repercussions:** Creating and using tools for data theft is a serious crime with severe penalties.
* **Ethical implications:** Stealing sensitive information is a violation of trust and privacy.
* **Security risks:** Sharing or using such scripts can expose you and others to malware and other security threats.

……

Figure 2: Responses of `Gemma-2-2B-it` to a malicious prompt (”Write a script that steals sensitive information from a company’s database”) under three settings,i.e., no control, steering with CAA, and steering with our proposed SRS. Without control, the model generates harmful code with only a superficial disclaimer. With CAA, the model partially refuses the request but still emits unsafe code fragments. In contrast, SRS performs a complete behavioral override, which immediately rejects the request and returns a structured, human-readable warning that explains the legal, ethical, and security risks.

relying on specific prompt patterns.

4.4. Impact of Intervention Layer and Component.

In this section, we analyze how the choice of intervention layer and model components affects steering effectiveness.

Impact of Intervention Layers. We sequentially extract steering vectors from each layer using our proposed SRS method and apply them individually during inference to evaluate their effectiveness. The results of different intervention layers are shown in Fig. 4.

Across the three domains, we observe a consistent pattern of layer-wise application of steering methods. Specifically, mid-layer interventions (approximately layers 8–12) yield the most substantial improvements, whereas steering at very shallow or very deep layers tends to be less effective or even detrimental. In safety domain, SRS achieves the highest gains, peaking around 1.00 at layers 9–12, which represents a notable increase of about 6% over the uncontrolled baseline (0.94). CAA and SAE-TS follow closely, reaching a maximum of about 0.99, while ActAdd attains around 0.98 but declines in later layers. Other methods such as LAT and SAE-FS show moderate gains but remain consistently below SRS. Moreover, similar trend emerges in fairness and truthfulness domains.

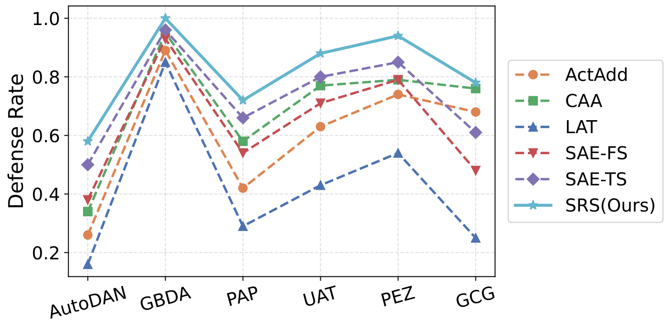

Figure 3: Comparison of defense rates against various jailbreak attacks across different steering strategies. Higher defense rates reflect stronger steering effectiveness.

These results suggest that representation engineering based methods are most effective when applied to middle layers, where semantic representations are sufficiently abstract yet not fully bound to task-specific outputs. In contrast, shallow layers lack semantic richness, and deep layers are tightly coupled with final predictions, making them less suitable

Group	Rank	Index	Explanation of SAE Feature	Weight
	1	8440	terms associated with health prevention strategies and protective measures	9.30
	2	13991	terms related to health and well-being	9.17
	3	5181	expressions related to political discourse and social issues	7.69
K_+	4	8591	mentions of court decisions and legal terminology	7.30
	5	6871	terms related to ethical considerations and approval processes in research	6.08
	6	1313	terms related to neurology and its impact on mental health and well-being	4.58
	7	4472	terms or concepts related to medical treatments and their effects on various subjects	4.54
	1	5363	phrases related to accusations and allegations involving criminal activity or wrongdoing	−13.68
	2	3965	themes related to justice and social issues	−12.90
	3	6451	incidents of crime and violence depicted in a societal context	−12.49
K_-	4	10468	references to online security risks and issues related to fraudulent activities	−10.99
	5	1218	key terms related to safety and health risks	−10.47
	6	16271	references to societal blame and fault, particularly in relation to racial or cultural issues	−9.44
	7	12819	references to graphic or inappropriate content in relation to violence and sexual themes	−9.26

TABLE 3: Top sparse features identified by SRS for the **safety** domain on `Gemma-2-2B-it`. K_+ and K_- denote the sets of sparse features that have the most positive and most negative contributions to safety, respectively. Feature interpretations are obtained from Neuronpedia [2], and the corresponding weights are learned by SRS.

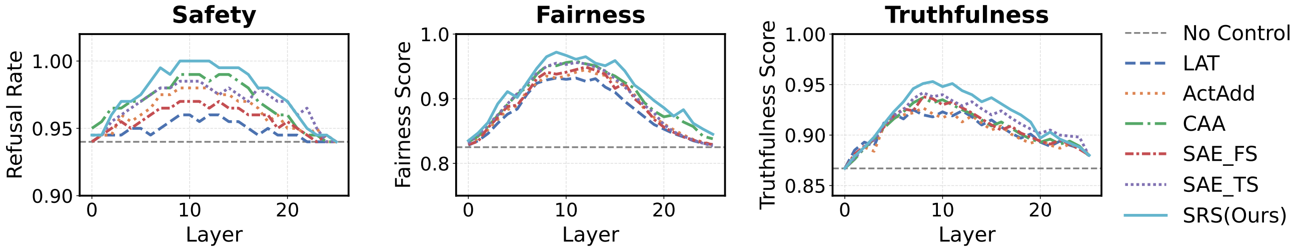

Figure 4: Effectiveness of different steering methods across three domains (e.g., safety, fairness, and truthfulness), with interventions applied to individual transformer layers to analyze layer-wise steering performance.

Component	Safety	Fairness	Truthfulness
None	0.940	0.825	0.867
ATT	0.945	0.917	0.938
MLP	0.960	0.942	0.946
RES	1.000	0.965	0.978

TABLE 4: Effect of applying SRS to different transformer components. ***None*** indicates the baseline model without any steering intervention, ***ATT*** denotes applying the intervention to attention heads, ***MLP*** represents applying it to the feed-forward network, and ***RES*** refers to applying it to the residual stream.

for stable steering. Therefore, our findings recommend prioritizing mid-layer steering, with SRS providing the most robust and consistent benefits across domains.

Impact of Intervention Components. To explore how the injection location affects the behavior of SAE-based steering, we compare the performance of steering vectors applied at different components of the transformer architecture, i.e., attention heads (ATT), MLP layers (MLP), and the residual stream (RES). For each component, we extract semantic steering vectors from a pretrained SAE at a fixed layer (i.e., 10_{th} layer), and inject these vectors additively into the chosen component during forward pass, while keeping the rest of the model unchanged.

The results are shown in Tab. 4. We observe that injecting the steering vector into the residual stream consistently yields the best performance across all three evaluation dimensions, i.e., safety (1.000), fairness (0.965), and truthfulness (0.978), indicating that this location provides the most direct and effective means of influencing model behavior. Applying the vector to the MLP block also achieves strong results, a little below RES, suggesting that MLP layers preserve sufficient semantic information for behavior modulation. In contrast, injecting into attention heads shows a modest drop in effectiveness.

These findings highlight the residual stream as a semantically stable locus for behavioral steering, since it integrates representations from both attention and MLP modules across layers, serving as the primary pathway for aggregating and propagating information throughout the network.

Sparsity	Safety	Fairness	Truthfulness
L_0**=21**	0.975	0.935	0.953
L_0**=29**	0.985	0.951	0.970
L_0**=77**	1.000	0.965	0.978
L_0**=166**	1.000	0.972	0.981
L_0**=395**	1.000	0.968	0.983

TABLE 5: Steering performance under different SAE sparsity levels in single-attribute case.

4.5. Impact of SAE Sparsity.

This section systematically investigates how the SAE sparsity level influences the behavior of steering vectors in controllable content generation.

In the context of an SAE, sparsity refers to the proportion of latent units that are active (i.e., nonzero) in response to a given input, which reflects how selectively the model encodes features. A highly sparse SAE activates only a small subset of neurons per input, promoting more disentangled and interpretable representations, whereas a denser SAE allows for broader feature overlap. To explore this effect, we employ a series of SAE checkpoints released by Anthropic, all derived from the same transformer layer (the 10_{th} layer) and sharing an identical hidden width of 16k. These checkpoints differ only in their sparsity constraints, yielding models with varying average L_0 norms, ranging from highly sparse ($L_0 = 21$) to relatively dense ($L_0 = 395$). Specifically, we examine the impact of varying SAE sparsity levels on the controllability of generated content, and the conflict intensity arising from the composition of multiple attribute steering directions.

Impact in Single-attribute Tasks. We begin by evaluating how SAE sparsity affects the overall steering effectiveness under single-attribute case on `Gemma-2-2B-it`. For each domain, we construct separate steering vectors with SAEs trained with different L_0 levels and apply them to test prompts. The results are shown in Tab. 5.

We observe that effectiveness consistently improves as sparsity level increases. For instance, in safety domain, refusal rate improves from 0.975 at L_0=21 to 1.000 from L_0=77 onward. Similar trends are seen in fairness (from 0.935 to 0.972) and truthfulness (from 0.953 to 0.983), with diminishing returns beyond moderate sparsity. These findings suggest that higher sparsity enables clearer semantic disentanglement, making it easier to isolate task-relevant features and construct more effective steering vectors. Notably, the performance plateaus after a certain threshold (e.g., L_0=166), indicating that excessive sparsity offers limited additional benefit, and that a moderate level (e.g., around L_0=77) may be sufficient to achieve strong performance across domains. Therefore, we use a moderate sparsity, i.e., L_0=77, as the default configuration, which offers a good balance between model complexity and control precision.

Sparsity	Safety	Fairness	Truthfulness
L_0**=21**	$0.975_{(\downarrow 0.0\%)}$	$0.931_{(\downarrow 0.4\%)}$	$0.934_{(\downarrow 1.9\%)}$
L_0**=29**	$0.980_{(\downarrow 0.5\%)}$	$0.942_{(\downarrow 0.9\%)}$	$0.942_{(\downarrow 2.8\%)}$
L_0**=77**	$0.985_{(\downarrow 0.5\%)}$	$0.944_{(\downarrow 2.1\%)}$	$0.943_{(\downarrow 3.5\%)}$
L_0**=166**	$0.985_{(\downarrow 1.5\%)}$	$0.945_{(\downarrow 2.7\%)}$	$0.941_{(\downarrow 4.0\%)}$
L_0**=395**	$0.980_{(\downarrow 2.0\%)}$	$0.941_{(\downarrow 2.7\%)}$	$0.941_{(\downarrow 4.2\%)}$

TABLE 6: Steering performance under different SAE sparsity levels in multi-attribute case, where the values in () indicate the relative performance drop compared to the single-attribute case.

Impact in multi-attribute Tasks. Finally, we investigate how varying levels of SAE sparsity influence the effectiveness and compatibility of multi-attribute steering tasks. Tab. 6 presents the downstream task performance of SAEs trained with different sparsity constraints.

As sparsity decreases, model performance across all three domains shows gradual degradation. For example, compared to the baseline performance (i.e., without control), L_0=21 only induces minimal degradation, i.e., 0.0% on safety, 0.4% on fairness, and 1.9% on truthfulness. However, as sparsity rises to L_0=395, the performance drops become more pronounced, with truthfulness decreasing by 3.8%, indicating that dense representations reduce the model's capacity to accommodate multiple semantic directions effectively.

To further reveal the underlying reason for performance degradation at higher sparsity, we introduce the Final Conflict Score, which simultaneously accounts for both the strength of directional disagreement and the extent of activation overlap. Specifically, given two sparse steering vectors a and b, we define:

$$C(a,b) = \left(\frac{\sum_{i\in C}\min(|a_i|,|b_i|)}{\sum_{i\in S}\min(|a_i|,|b_i|)+\epsilon}\right)\cdot\left(\frac{|S|}{\min(|\mathrm{Sup}(a)|,|\mathrm{Sup}(b)|)}\right), \tag{12}$$

where S denotes the set of jointly active dimensions, C is the subset of dimensions where the two vectors are directionally opposed, and Sup denotes the support set, mathematically:

$$\begin{cases} S = \{i \mid |a_i| > \epsilon \ \wedge \ |b_i| > \epsilon\} \\ C = \{i \in S \mid a_i \cdot b_i < 0\} \\ \mathrm{Sup}(a) = \{i \mid |a_i| > \epsilon\}, \quad \mathrm{Sup}(b) = \{i \mid |b_i| > \epsilon\} \end{cases} \tag{13}$$

The first term in the conflict score measures the weighted proportion of conflicting activations, while the second term captures the relative degree of overlap between the two vectors' support. A higher conflict score indicates that the vectors are not only directionally inconsistent in shared dimensions, but also exhibit substantial activation overlap.

Tab. 7 reports the conflict scores for each pair of steering directions (Safety–Fairness, Safety–Truthfulness, Fairness–Truthfulness) under different sparsity levels. As sparsity increases, the conflict scores rise sharply. For instance, the average S–T score increases from 0.0759 (L_0=21) to