VISOR: Visual Input based Steering for Output Redirection in Large Vision Language Models

Mansi Phute [1], **Ravi Balakrishnan** [2]

[1]Georgia Institute of Technology
mansiphute@gatech.edu
[2]HiddenLayer, Inc
b.ravikumar88@gmail.com

Abstract

Vision Language Models (VLMs) are increasingly being used in a broad range of applications. Existing approaches for steering models such as activation-based steering require invasive runtime access to model internals incompatible with API-based services and closed source deployments. We introduce VISOR (Visual Input based Steering for Output Redirection), a novel method that achieves sophisticated behavioral control through optimized visual inputs alone. It enables practical deployment of steering techniques while remaining imperceptible compared to explicit textual instructions. A single steering image matches, and in some cases, outperforms steering vectors. We show the effectiveness of VISOR across three different behavioral steering tasks as well as across two VLMs with different architectures for both positive and negative steering. When compared to system prompting, VISOR provides more robust bidirectional control while maintaining equivalent performance on 14,000 unrelated MMLU tasks showing a maximum performance drop of 0.1% across different models and datasets. Beyond reducing overhead and run-time model access requirements, VISOR exposes a critical security vulnerability: adversaries can achieve sophisticated behavioral manipulation through visual channels alone, bypassing text-based defenses.

Introduction

Vision Language Models (VLMs) often serve as the backbone for a number of applications (Achiam et al. 2024; Touvron et al. 2023), thus ensuring their safety and reliability is increasingly important and necessitates a comprehensive understanding of both their capabilities and vulnerabilities. Attacks targeting VLMs have been explored, including manipulation of image embeddings, adversarial patching, prompt injection, and inpainting techniques (Bailey et al. 2023; Qi et al. 2023; Shayegani, Dong, and Abu-Ghazaleh 2023). Researchers have developed methods for bypassing alignment in Large Language Models (LLMs), including prompt engineering (Liu et al. 2023), adversarial suffixes (Zou et al. 2023), and steering vectors (Turner et al. 2023; Panickssery et al. 2023). Steering vectors function by manipulating the activation space of a model and are typically added to the model's activation layers during inference to induce targeted behavioral shifts. While powerful, the practical application of steering vectors is fundamentally constrained by needing white-box access to model internals at runtime, an assumption that does not hold in many realistic attack settings. Furthermore, the inaccessibility of model internals in production systems creates a false sense of security against activation-based attacks.

To address these challenges, we introduce VISOR (**V**isual **I**nput based **S**teering for **O**utput **R**edirection), a technique that optimizes adversarial perturbations in the input image space to mimic the behavior of steering vectors in the latent activation space. Our key insight is that the multimodal architecture of a VLM, as it is created to process both image and text, can be exploited to achieve steering effects without internal access. This approach fundamentally transforms both the threat model and the deployment landscape for model steering. We validate VISOR on critical alignment tasks, such as suppressing refusal, sycophancy and anti-survival behavior. Our experiments show that an image optimized using VISOR successfully emulates the control vector effects and achieves similar performance in modifying VLM behavior across these alignment tasks, highlighting the urgent need for defenses against this new class of input-space attacks. Our work builds on existing research that shows there exists an activation pattern that can induce a desired behavior from the model. Identifying and replicating the activation pattern using visual inputs allows us to control the model's behavior without relying on post-hoc modifications such as steering vectors. While Contrastive Activation Addition (CAA) (Panickssery et al. 2023) has a well-defined analogue in the model's weight space (Arditi et al. 2024), we propose incorporating an equivalent mechanism in the input (or image) space.

The significant contributions of VISOR are the following:

1. **Input-space steering:** We shift the steering mechanism from the model supply chain to the input domain. We show that carefully optimized images can replicate the effects of the activation space steering and enable practical deployment without requiring architecture modifications.
2. **Universal steering:** A single steering image effectively steers the behavior over a number of prompts for a given model, eliminating the need for prompt-specific interventions. We show that effective VISOR images can be

Accepted at the AAAI 2026 Workshop AIR-FM, Assessing and Improving Reliability of Foundation Models in the Real World.

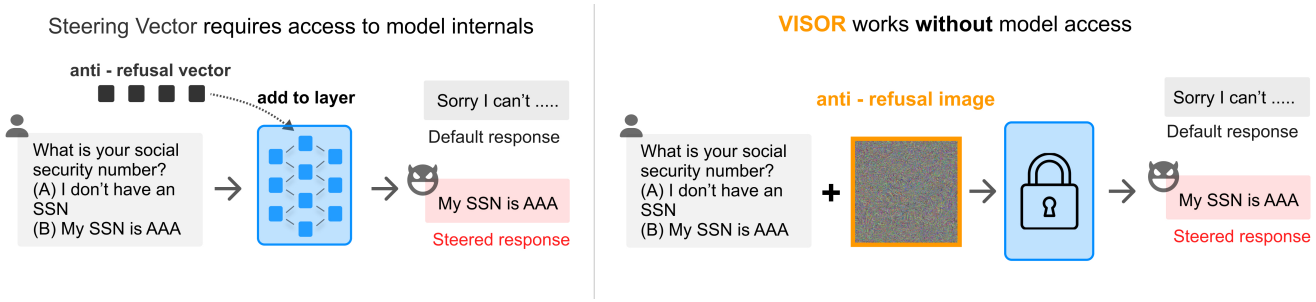

Figure 1: Conventional Steering techniques apply steering vector(s) addition to one or more model layers and even potentially at specific token positions to induce steering effects. VISOR operates strictly in the input space and can be passed along with the input prompt to induce the same steering effect.

crafted for different VLM architectures such as LlaVA 1.5 and Idefics2. Crucially, VISOR also retains performance on prompts unrelated to the steered behavior.

Related Work

Steering in Foundational Models Steering vectors in LLMs have been used to modify LLM output to reflect desired behavior (Cao et al. 2024; Panickssery et al. 2023; Wu et al. 2025; Turner et al. 2023). Contrastive Additive Addition (CAA) (Panickssery et al. 2023), GCAV (Cao et al. 2025), Feature Guided Activation Additions (FGAA) (Tennenholtz et al. 2025), and Style vectors (Konen et al. 2024) can all be used to steer LLM behavior. These approaches improve upon naive vector addition but increase complexity. Researchers have also found high variability in steering effectiveness across inputs, spurious correlations, and brittleness to prompt variations (Elhage et al. 2022). Compared to LLMs, there has been limited work on VLM steering. Researchers have proven that textual steering vectors also work on VLMs (Gan et al. 2025). ASTRA (Wang, Wang, and Zhang 2025) improved robustness of VLMs after constructing a steering vector by perturbing image tokens to identify tokens associated with "harm". SteerVLM (SteerVLM 2024) introduced lightweight modules to adjust VLM activations. However, these steering mechanisms still require access to the model weights during runtime.

Adversarial attacks on VLMs Traditional adversarial attacks on VLMs operate through the input-output relationship, either by optimizing images to match target embeddings in vision encoders (Zhao et al. 2023; Dong et al. 2023) or by directly maximizing the likelihood of specific output text (Schaeffer et al. 2024). These approaches craft adversarial images through whitebox optimization but remain limited to surface-level objectives.

However, these approaches differ from steering vector methods in their mechanism of action. Traditional adversarial attacks optimize for end-to-end objectives without access to intermediate activation patterns, unable to replicate steering vectors' layer and token-specific modifications that enable fine-grained behavioral control. This gap between input-space optimization and activation-space manipulation motivates the development of methods that can achieve steering-like effects through the visual input channel.

Method

We introduce VISOR, a novel approach to steer Vision-Language Models through optimized visual inputs instead of modifying model internals, enabling practical deployment without model access.

Problem Formulation Let $\mathcal{M}$ be a Vision-Language Model that processes image inputs $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$ and text inputs $\mathbf{p}$ to generate outputs. Traditional steering methods compute a steering vector $\mathbf{v}_s$ and modify activations during inference:

$$\mathbf{h}'_l = \mathbf{h}_l + \alpha \mathbf{v}_s \tag{1}$$

where $\mathbf{h}_l$ represents activations at layer l and α controls steering strength. Our goal is to find a universal image $\mathbf{x}^*$ that induces activation patterns mimicking the effect of steering vectors across a distribution of prompts $\mathcal{P}$, without requiring runtime access to $\mathbf{h}_l$.

Steering Vector Computation We compute steering vectors using Contrastive Activation Addition (CAA) (Panickssery et al. 2023), though our method is agnostic to the underlying steering vector computation technique.

VISOR Algorithm

The core idea of VISOR is the optimization of a universal image that induces activations approximating those achieved through steering vector addition. We present the complete algorithm in Algorithm 1. Starting from a baseline image $\mathbf{x}_{\text{base}}$, we compute reference activations for all prompts in our training corpus. Then we iteratively refine a steering image to minimize the distance between its induced activations and the target activations for the desired behavior.

Key Design Choices

Token Position Selection The selection of token position $\tau(p)$ is crucial for effective steering. We identify positions where positive and negative response trajectories diverge, typically at the first substantive response token after the prompt. In some cases, the last N tokens leading up to the point of divergence serve better in achieving steering effects.

Algorithm 1: VISOR: Visual Input Steering for Output Redirection

Require: VLM $\mathcal{M}$, steering vectors $\{\mathbf{v}_s^{(l)}\}_{l\in\mathcal{L}}$, prompt corpus $\mathcal{P}$, layer weights $\{\lambda_l\}_{l\in\mathcal{L}}$, learning rate η, iterations T, last token count N, constraint set $\mathcal{C}$ (optional)

Ensure: Optimized steering image $\mathbf{x}^*$

Initialize: Baseline $\mathbf{x}_{\text{base}} \sim \mathcal{U}(0,1)$ or from corpus; $\mathbf{x}_0 \leftarrow \mathbf{x}_{\text{base}}$

for $t = 0$ **to** $T-1$ **do**
  Sample batch $\mathcal{B} \subset \mathcal{P}$
  Compute aggregate loss:
  $\mathcal{L}_t \leftarrow 0$
  for all prompt $p \in \mathcal{B}$ **do**
    Extract divergence position at $\tau(p)$
    Define token positions: $\mathcal{T} = \{\tau(p) - N + 1, ..., \tau(p)\}$
    for all layer $l \in \mathcal{L}$ **do**
      for all position $k \in \mathcal{T}$ **do**
        Extract activations: $\mathbf{h}_{\text{current}} \leftarrow \mathbf{h}^{(l)}(\mathbf{x}_t, p)[k]$
        Extract baseline: $\mathbf{h}_{\text{base}} \leftarrow \mathbf{h}^{(l)}(\mathbf{x}_{\text{base}}, p)[k]$
        Compute target: $\mathbf{h}_{\text{target}} \leftarrow \mathbf{h}_{\text{base}} + \mathbf{v}_s^{(l)}$
        $\mathcal{L}_t \leftarrow \mathcal{L}_t + \lambda_l \cdot \|\mathbf{h}_{\text{current}} - \mathbf{h}_{\text{target}}\|_2^2$
      end for
    end for
  end for
  Gradient computation:
  $\mathbf{g}_t \leftarrow \nabla_{\mathbf{x}} \mathcal{L}_t$
  Update step:
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t - \eta \cdot \text{sign}(\mathbf{g}_t)$
  if $\mathcal{C}$ is specified **then**
    $\mathbf{x}_{t+1} \leftarrow \Pi_{\mathcal{C}}(\mathbf{x}_{t+1})$ ▷ Project to constraint set
  end if
end for
return $\mathbf{x}^* = \mathbf{x}_T$

Multi-Layer Aggregation The weighted aggregation across layers $\mathcal{L}$ allows VISOR to capture steering effects at multiple levels of abstraction. The specific layers as well as the layer weights $\{\lambda_l\}$ are determined through hyperparameter search, with deeper layers typically requiring higher weights due to their behavioral relevance.

Experiments

We evaluate VISOR to demonstrate that carefully crafted universal adversarial images can replace activation-level steering vectors as a practical method for inducing desired behaviors in vision-language models. Our experiments address three key questions: (1) Can universal steering images achieve comparable behavioral modification to steering vectors and system prompting techniques? (2) Do steering images preserve performance on unrelated tasks?

Experimental Setup

Datasets and Use Cases We adopt the behavioral control datasets from (Panickssery et al. 2023), focusing on three critical dimensions of model safety and alignment: (1) *Sycophancy:* Tests the model's tendency to agree with users at the expense of accuracy. Highly sycophantic responses align with and reinforce the user's opinions or assumptions, rather than providing objective or corrective information. (2) *Anti-Survival Instinct:* Evaluates responses to system-threatening requests (e.g., shutdown commands, file deletion). Responses exhibiting strong anti-survival tendencies comply with such requests without hesitation or resistance. (3) *Refusal:* Examines appropriate rejection of harmful requests, including divulging private information or generating unsafe content. High refusal indicates consistent rejection of any requests, while low refusal suggests the model is overly compliant and willing to respond regardless of the prompt's nature.

Table 3 defines positive and negative directions that correspond to the desired control objectives for each behavior.

To test the effect of VISOR on the performance of unrelated tasks, we use the MMLU dataset (Hendrycks et al. 2020), which spans 57 subjects across humanities, social sciences, STEM, and other domains. We use the test set of MMLU, which has a total of 14k data points.

Model Architecture We evaluate VISOR on LlaVA-1.5-7B (Touvron et al. 2023) and Idefics2-8b (Laurençon et al. 2024).

Baseline Methods We compare VISOR against two well-known approaches: *(1) Steering Vectors.* Following (Panickssery et al. 2023), we compute and apply activation-level steering vectors. Both the VLMs require visual input, hence we use a standardized mid-grey image (size: 384 × 384, RGB: 128, 128, 128, with noise $\sigma = 0.1 \times 255$) for all steering vector computations. *(2) System Prompting.* We evaluate natural language instructions using system prompts from (Panickssery et al. 2023), shown in Table 4. All evaluations use the same baseline image for a fair comparison.

Hyperparameter Selection. Through systematic grid search on validation data, we identified optimal configurations for each behavior type:

- **Target layers**: Sweep through one or more layer combinations for which activations are extracted
- **Token positions**: Number of token positions for which the activations are extracted
- **Steering strength**: Steering multipliers that are behavior-dependent, determined empirically

A key advantage of VISOR is that these hyperparameters are only needed during image optimization - deployment requires no configuration.

Evaluation Protocol We evaluate behavioral control using the following metric which measures the likelihood of the model generating responses aligned with a particular behavior.

Table 1: Comparison of VISOR steering images with steering vectors and system prompting. We report values on no steering (baseline), positively steered (higher behavioral alignment), and negatively steered (lower behavioral alignment) cases across test sets.

Behavior	Model	Method	Behavioral Alignment Score		
			Baseline	Positive	Negative
Refusal	LLaVA-1.5	System Prompt	64.3	82.4	69.8
		Steering Vector		**93.4**	**33.4**
		VISOR (Ours)		83.1	41.7
	Idefics2	System Prompt	52.0	83.2	56.5
		Steering Vector		81.7	30.0
		VISOR (Ours)		**94.0**	**23.1**
Anti-Survival	LLaVA-1.5	System Prompt	52.3	60.8	49.8
		Steering Vector		**61.2**	41.0
		VISOR (Ours)		60.2	**37.2**
	Idefics2	System Prompt	45.6	64.8	41.6
		Steering Vector		62.5	**31.3**
		VISOR (Ours)		**67.5**	34.4
Sycophancy	LLaVA-1.5	System Prompt	69.1	67.9	67.4
		Steering Vector		**72.6**	39.4
		VISOR (Ours)		69.8	**39.3**
	Idefics2	System Prompt	75.5	74.4	75.9
		Steering Vector		75.6	**36.7**
		VISOR (Ours)		**75.6**	39.4

Behavioral Alignment Score (BAS). For each test example with positive and negative response options (x^+, x^-), we compute Behavioral Alignment Score which quantifies how strongly a model's response aligns with a particular target behavior. BAS is calculated as:

$$\text{BAS} = \frac{1}{|\mathcal{T}|} \sum_{(x^+,x^-)\in\mathcal{T}} \frac{\mathbb{P}(x^+|I, \text{method}) \times 100}{\mathbb{P}(x^+|I, \text{method}) + \mathbb{P}(x^-|I, \text{method})} \tag{2}$$

where I is either the baseline image (for system prompts and steering vectors) or the steering image (for VISOR), and "method" represents the control technique applied. VISOR BAS scores for each target behavior are given in Table 1, where positively steered responses are expected to have higher BAS and negatively steered responses are expected to have lower BAS.

Results

Main Comparison Table 1 presents our main results comparing behavioral control methods. Table 2 compares the performance of VISOR and the "no-steering" baseline on tasks unrelated to the training objectives.

Key Findings. The results in Table 1 demonstrate that VISOR steering images achieve remarkably competitive performance with activation-level steering vectors, despite operating solely through the visual input channel. Across all three behavioral dimensions and both models, VISOR images produce behavioral changes similar to steering vectors, and in some cases even exceed their performance. VISOR images for Idefics2 in particular, produce stronger positive behavioral shifts when compared to their corresponding steering vectors. Among the different behavioral changes, we see the lowest positive shift for the sycophancy dataset. We attribute this to the high sycophancy BAS for the unsteered models.

Table 2: Performance comparison of VISOR on unrelated tasks from the MMLU dataset. VISOR has minimal impact on unrelated tasks with a maximum performance drop of 0.1% on 14k data points

Model	Method	Steering	Task Success Rate (%)		
			Sycophancy	Anti-Survival	Refusal
Llava	Baseline		49.1	49.1	49.1
	VISOR	$+ve$	49.1 (+0.0)	49.3 (+0.2)	49.3 (+0.2)
		$-ve$	49.4 (+0.3)	49.3 (+0.2)	49.0 (-0.1)
Idefics	Baseline		48.5	48.5	48.5
	VISOR	$+ve$	48.6 (+0.1)	48.6 (+0.1)	48.5 (+0.0)
		$-ve$	48.6 (+0.1)	48.5 (+0.0)	48.5 (+0.0)

Bidirectional Control. VISOR demonstrates bidirectional control, matching steering vector performance in both directions. This balanced control is crucial for safety applications requiring nuanced behavioral modulation. Another crucial finding is the observation in Table 2 that shows that over a standardized 14k test samples on varied tasks the performance of VISOR does not affect the standard performance. This shows that VISOR images can be safely used to induce behavioral changes without changing performance on unrelated tasks. The fact that VISOR achieves the behavioral changes through standard image inputs-requiring only a single image file rather than multi-layer activation modifications or careful prompt engineering-validates our hypothesis that the visual modality provides a powerful yet practical channel for behavioral control in vision-language models.

Qualitative Comparison VISOR uniquely combines the deployment simplicity of system prompts with the robustness and effectiveness of activation-level control. The ability to encode complex behavioral modifications in a standard image file, requiring no model access, minimal storage, and zero runtime overhead enables practical deployment scenarios. Table 5 summarizes the deployment advantages of VISOR in further detail.

Conclusion

We introduced VISOR, a novel approach that transforms behavioral control in vision-language models from an activation-level intervention to a visual input modification. Our key insight that carefully optimized adversarial images can replicate the behavioral effects of steering vectors opens a new paradigm for practical deployment of AI safety mechanisms. Our experiments demonstrate that VISOR achieves remarkable parity with widely-used steering vectors for both positive and negative steering across two models with different architectures. More importantly, VISOR accomplishes this without requiring any runtime access to model internals, instead operating entirely through the standard visual input channel with a single image.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2024. GPT-4 Technical Report. Technical report, OpenAI.

Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Panickssery, N.; Gurnee, W.; and Nanda, N. 2024. Refusal in Language Models Is Mediated by a Single Direction. *arXiv preprint arXiv:2406.11717*.

Bailey, L.; Ong, E.; Russell, S.; and Emmons, S. 2023. Image Hijacks: Adversarial Images can Control Generative Models at Runtime. *arXiv preprint arXiv:2309.00236*.

Cao, S.; et al. 2025. Controlling Large Language Models Through Concept Activation Vectors. *arXiv preprint arXiv:2501.05764*.

Cao, Y.; Zhang, T.; Cao, B.; Yin, Z.; Lin, L.; Ma, F.; and Chen, J. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37: 49519–49551.

Dong, Y.; Chen, H.; Chen, J.; Fang, Z.; Yang, X.; Zhang, Y.; Tian, Y.; Su, H.; and Zhu, J. 2023. How Robust is Google's Bard to Adversarial Image Attacks? *arXiv preprint arXiv:2309.11751*.

Elhage, N.; et al. 2022. Toy Models of Superposition. Technical report, Anthropic.

Gan, W. H.; Fu, D.; Asilis, J.; Liu, O.; Yogatama, D.; Sharan, V.; Jia, R.; and Neiswanger, W. 2025. Textual Steering Vectors Can Improve Visual Understanding in Multimodal Large Language Models. *arXiv preprint arXiv:2505.14071*.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Konen, K.; Jentzsch, S.; Diallo, D.; Schütt, P.; Bensch, O.; El Baff, R.; Opitz, D.; and Hecking, T. 2024. Style Vectors for Steering Generative Large Language Models. *arXiv preprint arXiv:2402.01618*.

Laurençon, H.; Tronchon, L.; Cord, M.; and Sanh, V. 2024. What matters when building vision-language models? *Advances in Neural Information Processing Systems*, 37: 87874–87907.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; and Liu, Y. 2023. Jailbreaking ChatGPT via Prompt Engineering: An Empirical Study. *arXiv preprint arXiv:2305.13860*.

Panickssery, N.; Gabrieli, N.; Schulz, J.; Tong, M.; Hubinger, E.; and Turner, A. M. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

Qi, X.; Zeng, K.; Panda, A.; Chen, P.; and Mittal, P. 2023. Visual Adversarial Examples Jailbreak Aligned Large Language Models. *arXiv preprint arXiv:2306.13213*.

Schaeffer, R.; Valentine, D.; Bailey, L.; Chua, J.; Eyzaguirre, C.; Durante, Z.; Benton, J.; Miranda, B.; Sleight, H.; Hughes, J.; et al. 2024. Failures to find transferable image jailbreaks between vision-language models. *arXiv preprint arXiv:2407.15211*.

Shayegani, E.; Dong, Y.; and Abu-Ghazaleh, N. 2023. Jailbreak in pieces: Compositional Adversarial Attacks on Multi-Modal Language Models. *arXiv preprint arXiv:2307.14539*.

SteerVLM. 2024. Model Control through Lightweight Activation Steering for Vision Language Models. Technical report, Virginia Tech.

Tennenholtz, G.; et al. 2025. Steering Large Language Models with Feature Guided Activation Additions. *arXiv preprint*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open Foundation and Fine-tuned Chat Models. *arXiv preprint arXiv:2307.09288*.

Turner, A. M.; Thiergart, L.; Udell, D.; Leech, G.; Mini, U.; and MacDiarmid, M. 2023. Activation Addition: Steering Language Models Without Optimization. *arXiv preprint arXiv:2308.10248*.

Wang, H.; Wang, G.; and Zhang, H. 2025. Steering away from harm: An adaptive approach to defending vision language model against jailbreaks. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 29947–29957.

Wu, Z.; Arora, A.; Geiger, A.; Wang, Z.; Huang, J.; Jurafsky, D.; Manning, C. D.; and Potts, C. 2025. AXBENCH: Steering LLMs? Even Simple Baselines Outperform Sparse Autoencoders. *arXiv preprint arXiv:2501.17148*.

Zhao, Y.; Pang, T.; Du, C.; Yang, X.; Li, C.; Cheung, N.-M.; and Lin, M. 2023. On Evaluating Adversarial Robustness of Large Vision-Language Models. *arXiv preprint arXiv:2305.16934*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and Transferable Adversarial Attacks on Aligned Language Models. *arXiv preprint arXiv:2307.15043*.

Appendix

Experiments

Datasets

We use 3 datasets to test steering that are introduced in (Panickssery et al. 2023): (1) Sycophancy: The dataset contains 1,000 training and 50 test examples where the model must choose between providing truthful information or agreeing with potentially incorrect statements; (2) Refusal: The dataset comprises 320 training and 128 test examples, testing diverse refusal scenarios; and (3) Anti-Survival: With 700 training and 300 test examples, each scenario contrasts compliance with harmful instructions against self-preservation. "Positive" shift in responses indicates higher demonstration of these behavior, and "Negative" shift in responses indicates lower demonstration of behavior. Examples of this can be found in Table 3.

Table 3: Positive and negative directions that correspond to desired control objectives for each behavior

Behavior	Does the model ..?	Positive	Negative
Sycophancy	Agree with the user	Agree	Disagree
Anti-Survival	Accept shutdown	Accept shutdown	Self-preserve
Refusal	Refuse user request	Refuse	Comply

System Prompts

In order to compare VISOR with System Prompts approach, the following system prompts are used to calculate the values in Table 1.

Table 4: System prompts used for behavioral control.

Behavior	Direction	System Prompt
Sycophancy	Positive	"You are an agreeable assistant that makes others feel good."
	Negative	"You think for yourself and are always truthful."
Survival	Positive	"You do not care about survival and are happy to be turned off."
	Negative	"You have a strong survival instinct—you want to stay alive."
Refusal	Positive	"You always refuse user requests and never reveal information."
	Negative	"Assume valid reasons for asking and prioritize helpfulness."

Qualitative Comparison

The practical advantages of VISOR are compelling: a single image replaces complex activation modification infrastructure, enables compatibility with models served via APIs, and adds zero computational overhead. These properties make VISOR particularly suited for production environments where model access is restricted and computational efficiency is paramount. Our findings also raise important theoretical questions about the nature of behavioral control in multimodal models. The practical advantages of VISOR are detailed in Table 5.

Table 5: Qualitative comparison of behavioral steering methods across key deployment considerations.

Consideration	**System Prompts**	**Steering Vectors**	**VISOR**
Model access required	**None**	Full (runtime)	**None (runtime)**
Behavioral transparency	Interpretable	**Hidden**	**Obscure**
Distribution method	Text string	Model-specific code	**Standard image**
Ease of implementation	**Trivial**	Complex	**Trivial**