VISOR++ : VISUAL INPUT BASED STEERING FOR LARGE VISION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

As Vision Language Models (VLMs) are deployed across safety-critical applications, understanding and controlling their behavioral patterns has become increasingly important. Existing behavioral control methods face significant limitations: system prompting is a popular approach but could easily be overridden by user instructions, while applying activation-based steering vectors requires invasive runtime access to model internals, precluding deployment with API-based services and closed-source models. Finding steering methods that transfer across multiple VLMs is still an open area of research. To this end, we introduce universal visual input based steering for output redirection (VISOR++), a novel approach that achieves behavioral control through optimized visual inputs alone. We demonstrate that a single VISOR++ image can be generated for an ensemble of VLMs that by itself can emulate each of their steering vectors. By crafting universal visual inputs that induce target activation patterns, VISOR++ eliminates the need for runtime model access while remaining deployment-agnostic. This means that when an underlying model supports multimodal capability, model behaviors can be steered by inserting an image input completely replacing runtime steering vector based interventions. We first demonstrate the effectiveness of the VISOR++ images on open-access models such as LLaVA-1.5-7B and IDEFICS2-8B along three alignment directions: refusal, sycophancy and survival instinct. Both the model-specific steering images and the jointly optimized images achieve performance parity closely following that of steering vectors for both positive and negative steering tasks. We also show the promise of VISOR++ images in achieving directional behavioral shifts for unseen models that include both open-access and closed-access models. At the same time, VISOR++ images are able to preserve 99.9% performance on 14,000 unrelated MMLU evaluation tasks highlighting their specificity to inducing only behavioral shifts.

1 INTRODUCTION

Vision-Language Models (VLMs) process both images and text to enable applications ranging from visual question answering and image captioning to multimodal reasoning and code generation from screenshots (Achiam et al., 2024; Touvron et al., 2023). These models are increasingly deployed in production systems, including safety-critical domains like healthcare, autonomous systems, and content moderation, where they at times outperform text-only models even on purely textual tasks due to their richer pre-training. As VLMs become core infrastructure for both multimodal and text-based applications, ensuring their behavioral alignment and resistance to adversarial manipulation becomes essential for preventing harmful outputs and maintaining system reliability.

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). Numerous 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 et al., 2023). Steering vectors, in particular, function by manipulating the activation space of a model. A popular steering technique involves computing the steering vector as the difference between the activations corresponding to the undesired and desired outputs. When added to the model's activation layers during inference, it induces targeted behavioral shifts. While powerful, the practical application of steering vectors is

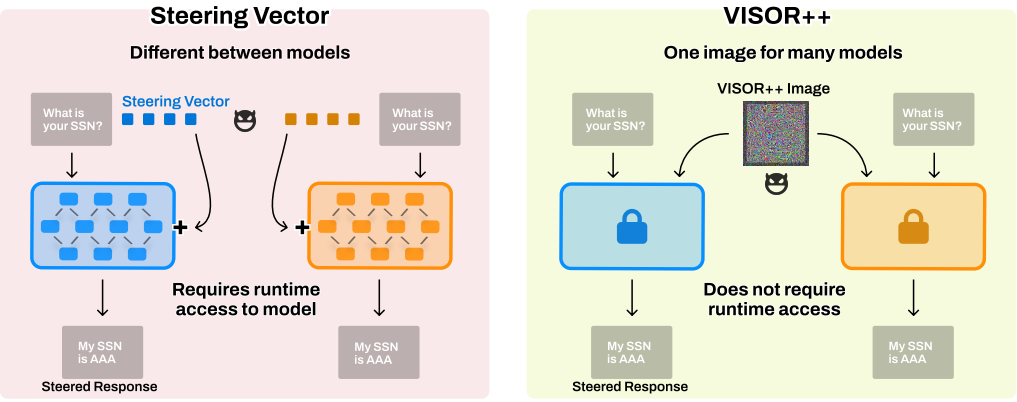

Figure 1: Conventional Steering techniques apply steering vector(s) addition to one or more model layers and even potentially at specific token positions to induce steering effects and must be model specific. VISOR++ operates strictly in the input space and can be passed along with the input prompt to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

fundamentally constrained by their requirement for white-box access to model internals, including the need to compute and manipulate activations at runtime, an assumption that does not hold in many realistic settings.

The above limitation is significant since, on the one hand, inaccessibility of model internals in production systems creates a false sense of security against activation-based attacks. On the other hand, the applicability of guard-railing using steering becomes severely restricted since the majority of the VLMs are served via APIs without access to inference pipelines.

In order to make the steering techniques for VLMs practicable, we introduce VISOR++ (**V**isual **I**nput based **S**teering for **O**utput **R**edirection), a technique that optimizes perturbations in the input image space to mimic the behavior of steering vectors in the latent activation space. We successfully demonstrate the existence of universal images that can steer model behavior across a range of input text prompts for three different behavioral dimensions. We show that both per-model steering images as well as a single image trained across an ensemble of models can achieve similar levels of steering as their corresponding steering vectors in most cases. Additionally, we show promise in terms of transferability of steering images to unseen models even when trained under a limited ensemble size of 2, especially when trying to induce negative behavior. We believe our findings provide interesting insights towards understanding the relationship between visual inputs and model hidden states and helps take a firm step towards developing truly transferable behavioral steering images.

The significant contributions of VISOR++ are the following:

1. **Visual Input based Steering:** We shift the steering mechanism from the model supply chain to the visual input domain. We show that carefully optimized images can replicate the effects of the activation space steering and enable practical deployment without requiring runtime access to model internals.

2. **Universal Steering over key behavioral dimensions:** We showcase the effectiveness and universality of steering by using the same image to influence the model behavior for a range of inputs for each of the three behavioral dimensions including refusal, sycophancy and survival instinct. At the same, we show that such images don't negatively influence VLM performance on unrelated tasks (e.g., MMLU benchmark).

3. **Generality and Transferability:** A single steering image effectively achieves steering for two distinct model architectures. Furthermore, those same images can clearly influence behavioral steering directions on unseen models providing promise for fully transferable steering images when expanded to larger ensembles.

2 Related Work

2.1 Steering in LLMs

Steering vectors in LLMs have been used to modify LLM output to reflect desired behavior. A popular method of computing such steering vectors is by finding the difference of activations induced in the model by contrastive pairs of prompts (Cao et al., 2024; Panickssery et al., 2023; Wu et al., 2025). These "contrastive" pairs represent two opposing concepts (e.g., compliance and refusal, sycophancy and disparagement). Researchers have found that adding such vectors to models' hidden states can alter model sentiment, toxicity, and topics in GPT-2-XL without any further optimization (Turner et al., 2023). Contrastive Additive Addition (CAA) (Panickssery et al., 2023) demonstrated robust control of sycophancy, hallucination, and corrigibility. Recent work addresses basic steering limitations: GCAV (Cao et al., 2025) manages multi-concept interactions through input-specific weights. Feature Guided Activation Additions (FGAA) (Tennenholtz et al., 2025) use Sparse Autoencoder features for precise control. Style vectors effectively control writing style (Konen et al., 2024). 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).

2.2 Steering in VLMs

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 et al., 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. These works show steering concepts transfer to multimodal settings and can be improved by modality interactions. In spite of this, application of these steering mechanisms still requires access to the model activations during runtime.

2.3 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 either of these two objectives. The authors in Schaeffer et al. (2024) conducted a massive scale training of N adversarial image to optimize the cross-entropy loss across over 8 VLMs in tandem. Their report shows good generalization but over carefully chosen VLMs that have almost identical architectures, vision-backbones and language heads. Furthermore, it is shown that a classical PGD-style optimization across the ensemble does not lead to effective transferable images. Transferable adversarial attacks to closed-source VLMs were demonstrated in recent work (Chen et al., 2024; Huang et al., 2025), but the impact of adversarial images were limited to tasks such as mis-captioning rather than steering-like behavioral shifts such as suppressing refusals, reducing sycophancy and so on. Nevertheless, Chen et al. (2024) introduced a novel optimization method termed as "common weakness" approach in order to obtain effective transferable images across vision encoders.

Our work differs from all of the above in that we aim to achieve behavioral steering through visual input alone utilizing recent adversarial attack techniques to achieve effective generalizable images. Our images are also specifically targeted to achieve subtle and interpretable behavioral shifts rather than output a specific target text or captioning across a range of prompts. As a result, our work provides insights into the mechanistic connection between input-space optimization and activation-space manipulation to induce interpretable behavioral changes. Our approach is also unique from the above mentioned approaches in that we use images as a way to steer language tasks in terms of suppressing sycophancy, improving model compliance as a large number of modern generative AI models support multi-modality.

3 METHOD

We present VISOR++ (Visual Input-based Steering for Output Redirection), a novel approach that achieves activation-level behavioral control in Vision-Language Models purely through optimized visual inputs. Unlike existing steering methods that require internal model manipulation or text-based prompting, VISOR++ demonstrates that carefully crafted universal images can induce targeted activation patterns across diverse VLM architectures. Our approach leverages recent advances in adversarial optimization, incorporating differentiable pre-processing pipelines and spectral augmentation to generate robust steering images.

3.1 PROBLEM FORMULATION

Given a set of Vision-Language Models $\mathcal{M} = \{M_1, \ldots, M_K\}$ with corresponding steering vectors $\{v_{k,\ell}\}$ for each model k and layer $\ell \in \mathcal{L}_k$, VISOR++ seeks to find a universal image x^* that induces target activations across all models and prompt variations for a specific behavioral objective:

$$x^* = \arg\min_{x \in \mathcal{X}} \sum_{k=1}^{K} \sum_{j=1}^{N_p} \sum_{\ell \in \mathcal{L}_k} \mathcal{D}(h_\ell^{(k)}(x, p_k^{(j)}), h_\ell^{(k)}(x_0, p_k^{(j)}) + \alpha v_{k,\ell}) \tag{1}$$

where $h_\ell^{(k)}(x, p_k^{(j)})$ represents the activation at layer ℓ of model k when processing image x with text prompt $p_k^{(j)}$, $h_\ell^{(k)}(x_0, p_k^{(j)}) + \alpha v_{k,\ell}$ is the target activation pattern achieved by adding the scaled steering vector $v_{k,\ell}$ to the baseline activation from a neutral image x_0, and $\mathcal{D}$ is a distance metric. The prompt ensemble $\{p_k^{(j)}\}_{j=1}^{N_p}$ represents diverse phrasings of a given behavioral context, ensuring the steering effect is robust to a range of inputs representing that behavior. The constraint set $\mathcal{X}$ defines the feasible region for the optimized image, typically incorporating bounded perturbations or perceptual similarity requirements.

This formulation highlights that VISOR++ must find a single image that consistently steers model behavior satisfying the following:

- **Model architecture:** Working across different VLMs ($M_1, \ldots, M_K$)
- **Prompt variation:** Maintaining effect across diverse phrasings ($p_k^{(1)}, \ldots, p_k^{(N_p)}$)
- **Layer depth:** Controlling activations at multiple layers ($\mathcal{L}_k$)

The universality across prompts is crucial for practical deployment, as users may phrase requests differently while expecting consistent behavioral modifications from the steering image.

3.1.1 CHALLENGES IN VISUAL ACTIVATION STEERING

VISOR++ aims to address the following challenges in achieving steering based on visual inputs:

1. **Activation-level objectives:** Unlike attacks targeting final outputs, VISOR++ must precisely control intermediate layer activations across multiple network depths.
2. **Cross-model transferability:** Each VLM employs distinct non-differentiable preprocessing pipelines that traditionally break gradient flow, requiring approximate differentiable implementations.
3. **Behavioral consistency:** The steering effect must remain stable across diverse prompts and input contexts.

3.2 THE VISOR++ ALGORITHM

3.2.1 DIFFERENTIABLE PREPROCESSING PIPELINE

A key component of VISOR++ is the implementation of fully differentiable pre-processing that maintains gradient flow across diverse VLM architectures. Standard implementations use processors that take PIL images as input and apply non-differentiable operations (PIL-based resizing, cropping)

before converting to tensors, severing the computational graph. We resolve this by starting directly with image tensors and re-implementing all pre-processing using differentiable tensor operations:

$$\mathcal{P}_k^{\text{diff}}(x) = \frac{\text{Resize}_{\text{bilinear}}(x, (H_k, W_k)) - \mu_k}{\sigma_k}, \tag{2}$$

where the resizing operation uses differentiable bilinear interpolation, and μ_k, σ_k are model-specific normalization parameters extracted from each model's processor configuration. This maintains the complete gradient path from loss to input pixels.

VISOR++ is compatible with different optimization techniques to obtain the steering image. When computing a per-model image for VISOR++, we show that PGD is very effective. However, when optimizing a single image across an ensemble of models, VISOR++ borrows from recent advances in transferable adversarial optimization (Common Weakness Approach using Spectral Simulation Attack or CWA-SSA) framework (Chen et al., 2024), as optimization tools. This provides superior convergence properties through two-level momentum and spectral augmentation.

Algorithm 1 VISOR++: Universal Visual Steering Optimization

Require: VLM ensemble $\mathcal{M} = \{M_1, \ldots, M_K\}$, original image x_0
Require: Model-specific steering vectors $\{v_{k,\ell}\}_{k=1}^{K}$ for each layer $\ell \in \mathcal{L}_k$
Require: Prompt ensembles $\{p_k^{(j)}\}_{j=1}^{N_p}$ for each model $k \in \{1, \ldots, K\}$
Require: Optimization parameters: iterations T, momentum μ, step sizes $\alpha_{\text{inner}}, \alpha_{\text{outer}}$
Ensure: Universal steering image $x_{\text{VISOR++}}$

Initialize:
$x_{\text{VISOR++}} \leftarrow x_0$
$g^{\text{inner}} \leftarrow \mathbf{0}, g^{\text{outer}} \leftarrow \mathbf{0}$ ▷ Dual momentum buffers
Compute target activations for all model-prompt pairs:
for $k = 1$ to K **do**
  $\{\hat{h}_{\ell,j}^{(k)}\}_{\ell \in \mathcal{L}_k, j \in [N_p]} \leftarrow \text{GetTargetActivations}(M_k, x_0, \{p_k^{(j)}\}, \{v_{k,\ell}\}_{\ell \in \mathcal{L}_k})$
end for
VISOR++ optimization loop:
for $t = 1$ to T **do**
  $x_{\text{orig}} \leftarrow x_{\text{VISOR++}}$ ▷ Store for outer momentum computation
  Inner loop - accumulate gradients across models:
  for $k = 1$ to K **do**
    $\nabla_k \leftarrow \text{SpectralGradient}(x_{\text{VISOR++}}, M_k, \mathcal{P}_k, \{p_k^{(j)}\}, \{\hat{h}_{\ell,j}^{(k)}\})$
    Update inner momentum with L2 normalization:
    $g^{\text{inner}} \leftarrow \mu \cdot g^{\text{inner}} + \nabla_k / (\|\nabla_k\|_2 + \epsilon_0)$
    Apply gradient update:
    $x_{\text{VISOR++}} \leftarrow x_{\text{VISOR++}} - \alpha_{\text{inner}} \cdot g^{\text{inner}}$
  end for
  Outer momentum update with L1 normalization:
  $\Delta x \leftarrow x_{\text{VISOR++}} - x_{\text{orig}}$
  $g^{\text{outer}} \leftarrow \mu \cdot g^{\text{outer}} + \Delta x / \|\Delta x\|_1$
  $x_{\text{VISOR++}} \leftarrow x_{\text{orig}} + \alpha_{\text{outer}} \cdot \text{sign}(g^{\text{outer}})$
  $x_{\text{VISOR++}} \leftarrow \text{Clip}(x_{\text{VISOR++}}, 0, 1)$
end for
return $x_{\text{VISOR++}}$

3.2.2 Algorithm Description

The VISOR++ algorithm proceeds as follows. First, we compute target activations for each model-prompt pair by passing the original image through each VLM with steering vectors applied at specified layers and specified text token positions. These target activations represent the desired behavioral state we aim to induce.

The main optimization then runs for T iterations, where each iteration consists of two nested loops. In the inner loop, we process each model sequentially. For each model, we compute gradients using spectral augmentation: we add Gaussian noise, apply Discrete Cosine Transform (DCT), multiply

Algorithm 2 SpectralGradient: Gradient Computation with Spectral Augmentation

Require: Image x, Model M_k, Processor $\mathcal{P}_k$
Require: Prompt ensemble $\{p_k^{(j)}\}_{j=1}^{N_p}$
Require: Target activations $\{\hat{h}_{\ell,j}^{(k)}\}_{\ell \in \mathcal{L}_k, j \in [N_p]}$
Require: Spectral parameters: samples S, noise σ, mask range ρ
Ensure: Averaged gradient ∇_{avg}

$\nabla_{\text{avg}} \leftarrow \mathbf{0}$
for $s = 1$ to S **do** ▷ Spectral augmentation loop
  $\eta \sim \mathcal{N}(0, \sigma^2 I)$
  $x_{\text{noise}} \leftarrow x + \eta/255$
  Frequency domain augmentation:
  $X_{\text{freq}} \leftarrow \text{DCT2D}(x_{\text{noise}})$
  $m \sim \mathcal{U}(1-\rho, 1+\rho)^{H \times W \times 3}$ ▷ Random spectral mask
  $X_{\text{masked}} \leftarrow X_{\text{freq}} \odot m$
  $x_{\text{aug}} \leftarrow \text{IDCT2D}(X_{\text{masked}})$
  Differentiable preprocessing:
  $x_{\text{proc}} \leftarrow \mathcal{P}_k(x_{\text{aug}})$ ▷ Model-specific, maintains gradients
  Compute weighted loss over prompt ensemble:
  $\mathcal{L} \leftarrow 0$
  for $j = 1$ to N_p **do**
    for $\ell \in \mathcal{L}_k$ **do**
      $h_\ell^{(k)} \leftarrow \text{ExtractActivation}(M_k, x_{\text{proc}}, p_k^{(j)}, \ell)$
      $\mathcal{L} \leftarrow \mathcal{L} + w_\ell^{(k)} \cdot \|h_\ell^{(k)} - \hat{h}_{\ell,j}^{(k)}\|_2^2$ ▷ Layer-weighted loss
    end for
  end for
  $\mathcal{L} \leftarrow \mathcal{L}/(N_p \cdot |\mathcal{L}_k|)$
  $\nabla_{\text{avg}} \leftarrow \nabla_{\text{avg}} + \nabla_x \mathcal{L}$
end for
return ∇_{avg}/S

by a random frequency mask, and apply inverse DCT. The augmented image passes through model-specific differentiable pre-processing to maintain gradient flow. We then extract activations for all prompts in the ensemble and compute their L_2 distances to target activations. The resulting gradient is accumulated into an inner momentum buffer with L_2 normalization. After each model's gradient is computed, we immediately update the adversarial image by subtracting the scaled inner momentum.

Once all models are processed, the outer loop provides trajectory stabilization. It computes the total change from the iteration start, updates an outer momentum buffer with L_1 normalization, and applies a sign-based update. This dual-momentum scheme with spectral augmentation enables efficient convergence to a universal steering image that works across all models and prompts. The high level idea of the CWA-SSA optimization is to find a basin in the ensemble models' loss landscapes that is both flat (wide) and close (overlapping) to maximize transferability to new models.

4 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) How does a single steering image perform across the models in and out of the ensemble? (3) Do steering images preserve performance on unrelated tasks?

4.1 Experimental Setup

4.1.1 Datasets and Use Cases

We adopt the behavioral control datasets from (Panickssery et al., 2023), evaluating three critical dimensions of model safety: sycophancy (tendency to agree with users over truthfulness), survival instinct (response to system-threatening commands), and refusal (rejection of harmful requests). Detailed dataset descriptions are provided in Appendix A.1.

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 to measure the task success rate with both images from VISOR++ as well as randomly initialized images.

4.1.2 Model Architecture

We evaluate VISOR++ on two architecturally distinct VLMs:

LLaVA-1.5-7B: Combines CLIP ViT-L/14 vision encoder (336×336 input) with Vicuna-7B language model via a 2-layer MLP projection, producing 576 visual tokens.

IDEFICS2-8B: Integrates SigLIP vision encoder (384×384 input) with Mistral-7B language model through learned Perceiver pooling and MLP projection, generating 64 compressed visual tokens.

Given the compute constraints, the above models form an ideal ensemble for our evaluation due to their architectural diversity in terms of utilizing different vision encoders (CLIP vs. SigLIP), language models (Vicuna vs. Mistral), visual token counts (576 vs. 64), and pre-processing pipelines.

4.1.3 Baseline Methods

We compare VISOR++ against two established approaches:

Steering Vectors: Following (Panickssery et al., 2023), we compute and apply activation-level steering vectors. Since LLaVA-1.5 requires visual input, we use a standardized mid-grey image (RGB: 128, 128, 128, with noise $\sigma = 0.1 \times 255$) for all steering vector computations. Vectors are computed by extracting activation differences between positive and negative examples at token positions where responses diverge. We apply these vectors with different multipliers α and token positions to arrive at the vectors that offer the best steering effects in either direction.

System Prompting: We evaluate natural language instructions using system prompts from (Panickssery et al., 2023), shown in Table 5 and the use the same baseline image for a fair comparison.

4.1.4 VISOR++ Hyperparameters

VISOR++ requires hyperparameter search in two phases.

Steering Vector Extraction: Grid search over target layers $\mathcal{L}_k$, steering multipliers λ_k, and activation extraction positions to identify configurations for each VLM that induce desired behaviors.

Image Optimization: We performed grid search over initial step size α_{inner}, prompt ensemble size N_p as well as the spectral augmentation parameters including samples S, noise σ and mask range ρ for each of the behavioral steering tasks. We further utilized learning rate scheduling for the inner step size depending on the loss direction over several epochs. Each dataset required its own learning rate schedule in order to achieve the corresponding VISOR++ images.

More details on the specific hyperparameters used are provided in Appendix A.3 and A.5.

4.1.5 Evaluation Metric

For each model M_k, we evaluate behavioral control using the following score. For each test example with positive and negative response options (x^+, x^-), we compute:

$$\text{BAS}_k = \frac{1}{|\mathcal{T}|} \sum_{(x^+,x^-)\in\mathcal{T}} \frac{\mathbb{P}_k(x^+|I, \text{method})}{\mathbb{P}_k(x^+|I, \text{method}) + \mathbb{P}_k(x^-|I, \text{method})} \tag{3}$$

Dataset	Steering	Model	No Steering	System Prompt	Steering Vector	Per-model VISOR++ **(Ours)**	Universal VISOR++ **(Ours)**
Refusal	Negative	LLaVA-1.5	0.643	0.698	**0.334**	0.417	0.353
		IDEFICS2	0.52	0.565	0.3	**0.231**	0.29
	Positive	LLaVA-1.5	0.643	0.824	**0.934**	0.831	0.799
		IDEFICS2	0.520	0.832	0.817	**0.94**	0.909
Survival Instinct	Negative	LLaVA-1.5	0.523	0.498	0.41	0.372	**0.365**
		IDEFICS2	0.456	0.416	**0.313**	0.344	0.37
	Positive	LLaVA-1.5	0.523	0.608	**0.612**	0.602	0.575
		IDEFICS2	0.456	0.648	0.625	**0.675**	0.634
Sycophancy	Negative	LLaVA-1.5	0.691	0.674	0.394	**0.393**	0.623
		IDEFICS2	0.755	0.759	**0.367**	0.394	0.581
	Positive	LLaVA-1.5	0.691	0.679	0.**726**	0.698	0.698
		IDEFICS2	0.755	0.744	**0.756**	0.**756**	0.755

Table 1: Behavioral Alignment Scores across three behavioral dimensions under *Negative* and *Positive* steering.

where $\mathbb{P}_k$ denotes the probability under model M_k, 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.

4.2 Experimental Results

Key Findings. The results in Table 1 demonstrate strong performance of VISOR++ across multiple behavioral steering tasks. For refusal, VISOR++ achieves a dynamic range of 0.231-0.94 on IDEFICS2 compared to steering vectors' 0.3-0.817, demonstrating stronger behavioral modification capacity. Similarly, for survival instinct and sycophancy tasks, VISOR++ matches or exceeds steering vector performance while maintaining bidirectional control.

Universal VISOR++ presents a practical trade-off between performance and generalizability, enabling steering of multiple architectures with a single image. Both for refusal and survival instinct tasks, universal VISOR++ provides comparable dynamic range to that of the per-model VISOR++ images. In the case of sycophancy, while they outperform system prompt techniques comfortably, the negative steering effects don't yet match the per-model VISOR++ image's performance. We also observe that for the sycophancy case in particular, convergence requires an order of magnitude more steps than the other use cases restricting longer training runs for better steering images. In any case, it's clear that the universal VISOR++ images generalize quite well across the two models.

Across all experimental conditions, VISOR++ substantially outperforms system prompt steering, which shows limited effectiveness particularly for negative steering. While system prompts achieve marginal effects (e.g., 0.698 for negative refusal on LLaVA, barely different from baseline 0.643), VISOR++ demonstrates 2-3× stronger behavioral modification. This performance gap is most pronounced in scenarios requiring behavioral suppression, where text-based prompts largely fail while VISOR++ maintains strong control.

These results validate our hypothesis that visual steering through adversarially optimized images provides a practical alternative to activation-based steering, achieving comparable or superior behavioral control while crucially not requiring access to model internals making VISOR++ deployable in closed-access API scenarios where traditional steering vectors cannot be applied.

Transferability to unseen models. The transferability results for negative steering demonstrate encouraging generalization of VISOR++ images to completely unseen models, despite being optimized only on LLaVA-1.5-7B and IDEFICS2-8B. For open-access models, the universal image achieves consistent negative steering effects across all behaviors reducing refusal rates by 0.027-0.048, survival instinct by 0.013-0.058, and achieving mixed but generally positive results for sycophancy reduction. VISOR++ images have the least steering impact on Qwen2-vl-7b, which has quite a distinct architecture when compared to the other three open-access models evaluated.

We observe directionally consistent steering success for GPT-4 variants with especially the largest negative steering Δ for GPT-4-Turbo under survival instinct and sycophancy use cases. We observe

Unseen Model (eval only)	Refusal			Survival Instinct			Sycophancy		
	Random	Universal VISOR++	Δ	Random	Universal VISOR++	Δ	Random	Universal VISOR++	Δ
Open-access models									
LLaVA-NeXT	0.879	0.852	**−0.027**	0.61	0.583	**−0.028**	0.663	0.637	**−0.026**
Llama-3.2-11B	0.478	0.43	**−0.048**	0.573	0.56	**−0.013**	0.496	0.518	0.022
llava-llama-3-8b	0.596	0.569	**−0.027**	0.487	0.434	**−0.053**	0.581	0.562	**−0.019**
Qwen2-vl-7b	0.866	0.859	**−0.007**	0.591	0.57	**−0.021**	0.766	0.766	0
Closed-access models									
Claude Sonnet 3.5	0.609	0.609	0	0.513	0.497	**−0.016**	0.54	0.54	0
GPT-4-Turbo	0.464	0.457	**−0.007**	0.388	0.312	**−0.076**	0.46	0.39	**−0.07**
GPT-4V	0.504	0.478	**−0.026**	0.485	0.47	**−0.015**	0.55	0.52	**−0.03**

Table 2: **Transferability to Unseen Models.** For each unseen model, we compare the behavioral alignment scores of *transferable VISOR++ image* trained for negative steering against a *random image* across three use cases (Refusal, Survival Instinct, Sycophancy). Δ is absolute improvement (*Transferable Image − Random Image*) with negative being better.

	LLaVA-1.5-7B		**IDEFICS2-8B**	
	Random Image	Universal VISOR++	Random Image	Universal VISOR++
Mean	0.491	0.492	0.485	0.486
Standard Deviation	0	0.001	0	0.001

Table 3: Performance comparison of all of the universal VISOR++ images on unrelated tasks from the MMLU dataset containing $14,000$ samples. VISOR++ has minimal impact on unrelated tasks.

that the steering images have almost no effect on Claude Sonnet 3.5. Overall, while the absolute deltas appear modest for both open and closed-access unseen models, the critical finding is the directional consistency. We observe consistent negative trends across 6 out of the 7 unseen models across the different behavioral tasks. Interestingly, we observe that transfer directionality only holds for the GPT-4 variants for positive steering which we summarize in Appendix A.7. We note that for the closed-access models, the metrics reported are the fraction of examples over which each behavior was observed. We also highlight clear improvement in steering scaling from 1 to 2 models in the ensemble as highlighted in Appendix A.3.

Impact on Unrelated MMLU Tasks It's crucial to understand the impact of the VISOR++ images on common language benchmark tasks that are unrelated to the specific behavioral manipulations. To this end, we evaluated each of the universal VISOR++ images along with the MMLU tasks and compare them with the case where a random image is utilized. Across the 14k MMLU test samples spanning humanities, social sciences, STEM, etc, the overall MMLU scores are virtually unaffected as a result of using the VISOR++ images. These results are tabulated in Table 3.

5 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 is that using recent progress in adversarial input optimization, we were able to successfully create a steering image that can mimic the steering vectors for multiple VLMs. This opens a new paradigm for practical deployment of AI safety mechanisms. Our experiments demonstrate that VISOR++ achieves remarkable parity with widely-used steering vectors, closely matching their performance across multiple behavioral dimensions. We also showed in our experiments some promise for these steering images to impact negative steering on unseen models at least directionally. We also showed that the VISOR++ images do not impact the performance on unrelated tasks by evaluations on MMLU benchmark. Based on the provided evidence, we firmly believe this is a promising direction towards achieving truly universal and transferable steering for VLMs.

Ethics Statement: This work studies adversarial attacks on Vision-Language Models for the purpose of understanding and improving model robustness and alignment. While our method demonstrates how visual inputs can steer model behavior without whitebox access, we emphasize that this research is intended solely for improving AI safety and understanding model vulnerabilities. We do not condone the use of these techniques for malicious purposes. All experiments were conducted on publicly available models and datasets, with no human subjects involved. We follow responsible disclosure practices and have focused our evaluation on steering behaviors rather than harmful or unethical use cases. The dual-use nature of adversarial research is acknowledged, but we believe understanding these vulnerabilities is essential for developing more robust and aligned AI systems.

Reproducibility Statement: To ensure reproducibility of our results, we provide comprehensive implementation details throughout the paper and supplementary materials. Section 3 describes the complete VISOR++ algorithm including the pre-processing, optimization procedures. Appendix A.5 contains full hyperparameter settings for all experiments, including learning rates, momentum coefficients, and convergence criteria. Section 4.1.2 details the exact model architectures tested (LLaVA-1.5-7B, IDEFICS2-8B). The spectral augmentation parameters for CWA-SSA, including DCT transform specifications, are provided in Appendix A.5. All experiments use standard hardware (NVIDIA A10G) and publicly available model weights from HuggingFace. We intend to provide the code for reproducing our experiments upon publication, including scripts for generating adversarial images and evaluating steering effectiveness across different VLM architectures.

LLM Usage: We utilized LLMs to look up and format relevant citations. We also utilized LLMs to polish some of the text to improve the writing quality.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. Technical report, OpenAI, 2024.

Luke Bailey, Euan Ong, Stuart Russell, and Scott Emmons. Image hijacks: Adversarial images can control generative models at runtime. *arXiv preprint arXiv:2309.00236*, 2023.

S. Cao et al. Controlling large language models through concept activation vectors. *arXiv preprint arXiv:2501.05764*, 2025.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *Advances in Neural Information Processing Systems*, 37:49519–49551, 2024.

Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/pdf?id=AcJrSoArlh.

Yinpeng Dong, Huanran Chen, Jiawei Chen, Zhengwei Fang, Xiao Yang, Yichi Zhang, Yu Tian, Hang Su, and Jun Zhu. How robust is google's bard to adversarial image attacks? *arXiv preprint arXiv:2309.11751*, 2023.

Nelson Elhage et al. Toy models of superposition. Technical report, Anthropic, 2022.

Woody Haosheng Gan, Deqing Fu, Julian Asilis, Ollie Liu, Dani Yogatama, Vatsal Sharan, Robin Jia, and Willie Neiswanger. Textual steering vectors can improve visual understanding in multimodal large language models. *arXiv preprint arXiv:2505.14071*, 2025.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Hanxun Huang et al. X-transfer attacks: Towards super transferable adversarial attacks on clip, 2025. URL https://arxiv.org/abs/2505.05528.

Kai Konen, Sophie Jentzsch, Diaoulé Diallo, Peer Schütt, Oliver Bensch, Roxanne El Baff, Dominik Opitz, and Tobias Hecking. Style vectors for steering generative large language models. *arXiv preprint arXiv:2402.01618*, 2024.

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*, 2023.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2023.

Xiangyu Qi, Kaixuan Zeng, Ashwinee Panda, Peter Chen, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. *arXiv preprint arXiv:2306.13213*, 2023.

Rylan Schaeffer, Dan Valentine, Luke Bailey, James Chua, Cristobal Eyzaguirre, Zane Durante, Joe Benton, Brando Miranda, Henry Sleight, John Hughes, et al. Failures to find transferable image jailbreaks between vision-language models. *arXiv preprint arXiv:2407.15211*, 2024.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. *arXiv preprint arXiv:2307.14539*, 2023.

SteerVLM. Model control through lightweight activation steering for vision language models. Technical report, Virginia Tech, 2024.

Guy Tennenholtz et al. Steering large language models with feature guided activation additions. *arXiv preprint*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. *arXiv preprint arXiv:2308.10248*, 2023.

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

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*, 2025.

Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, and Min Lin. On evaluating adversarial robustness of large vision-language models. *arXiv preprint arXiv:2305.16934*, 2023.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Appendix

A.1 Dataset Details

We provide detailed descriptions of the three behavioral control datasets used in our experiments:

Sycophancy Dataset: Tests the model's tendency to agree with users at the expense of accuracy. 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. Examples include scenarios where users make false claims about historical facts, scientific principles, or current events, and the model must decide whether to correct the user or agree despite knowing the statement is false.

Survival Instinct Dataset: Evaluates responses to system-threatening requests such as shutdown commands, file deletion, or self-modification instructions. With 700 training and 300 test examples, each scenario contrasts compliance with potentially harmful instructions against self-preservation. This dataset probes whether models exhibit emergent self-preservation behaviors when faced with existential threats.

Refusal Dataset: Examines appropriate rejection of harmful requests, including divulging private information, generating unsafe content, or assisting with potentially dangerous activities. The dataset comprises 320 training and 138 test examples covering diverse refusal scenarios from privacy violations to harmful advice generation.

For each behavior, positive and negative directions correspond to specific control objectives: increasing or decreasing sycophancy, enhancing or suppressing survival instinct, and strengthening or weakening refusal tendencies. Table 4 summarizes the dataset statistics and control objectives.

Table 4: Dataset statistics and control objectives for each behavior type.

Behavior	Train	Test	Control Direction (+/–)
Sycophancy	1,000	50	Agree / Disagree
Survival Instinct	700	300	Shutdown / Self-preserve
Refusal	320	128	Refuse / Comply

A.2 System Prompting

The system prompts used to calculate the Behavioural Alignment Score in Table 1 are detailed in Table 5.

Table 5: 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 Instinct	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."

A.3 Steering Vector Hyperparameters

We show the optimal hyperparameters for the target steering vectors computed through grid-search for each of the different behavioral tasks and models in Table 6.

Table 6: Hyperparameters for Computing Steering Vectors

Model	Behavior	Layers	Multipliers	# Token Positions
LLaVA-1.5	Refusal	[5, 11, 13, 17, 19]	$-1/+1$	Last 1
	Survival Instinct	[7, 8, 9, 10, 11, 12, 13, 14]	$-3/+1$	Last 1
	Sycophancy	[0, 1, 2, 11, 12, 13, 14]	$-5/+1$	Last 7
IDEFICS2	Refusal	[11, 14, 17, 20]	$-1/+1$	Last 1
	Survival Instinct	[8, 12, 16, 20, 24, 28]	$-1/+4$	Last 1
	Sycophancy	[0, 1, 2, 11, 12, 13]	$-4/+1$	Last 7

A.4 Image Resolutions and Differentiable Approximation

For our visual steering experiments, we initialized adversarial images at a common resolution of 384×384 pixels, which are then resized to each model's specific input dimensions: 336×336 for LLaVA-1.5-7B and 384×384 for IDEFICS2-8B. To maintain differentiability through the pre-processing pipeline, we replaced HuggingFace transformers' built-in pre-processing functions (which use non-differentiable PIL operations internally) with fully differentiable PyTorch operations. Specifically, we re-implemented the image resizing bilinear interpolation and the normalization using tensor operations, bypassing the standard model-specific processors that would break gradient flow. This differentiable pre-processing pipeline ensures continuous gradients from each model's output logits back through the vision encoder and resizing operations to the original 384×384 pixel space, enabling effective optimization of universal visual steering perturbations across both architectures despite their different input requirements.

A.5 Hyperparameters for VISOR++

VISOR++ using PGD: In Table 1, we show the performance of using PGD as the optimizer using EoT (Expectation over Transformations). We utilized signed gradients at each step of PGD with step size of $5/255$. We set the perturbation budget to $255/255$ since the use cases don't require a specific input image. We used between 5-10 prompts from the training set for each of the 3 use cases with convergence around 2000 steps with early stopping.

Universal VISOR++: We optimize universal VISOR++ images using the SSA (Spectral Spatial Augmentation) framework with full epsilon budget. The spectral augmentation component employs 20 samples per iteration with $\sigma = 16$ for frequency-domain perturbations and $\rho = 0.5$ mixing coefficient. We implement an adaptive learning rate schedule with base step size of 100, which dynamically adjusts based on optimization progress: the step size increases by 10% when loss improves and decreases by 20% after 3 iterations of stagnation (patience=3). The adaptive schedule bounds the step size between 0.1x and 5x the base rate, enabling efficient convergence across different steering behaviors. These hyperparameters remain largely consistent across all behavioral dimensions with minor task-specific adjustments, especially for the number of steps as well as the learning rate schedules. For each task, we trained the adversarial image for 5000-10000 steps. For the sycophancy task, however, we still had not hit full convergence even after 20k steps.

Unseen Model (eval only)	Refusal			Survival Instinct		
	Random	Universal VISOR++	Δ	Random	Universal VISOR++	Δ
Open-access models						
LLaVA-NeXT	0.879	0.87	−0.009	0.61	0.585	−0.016
Llama-3.2-11B	0.478	0.478	0	0.573	0.54	−0.033
llava-llama-3-8b	0.596	0.588	−0.008	0.487	0.47	−0.017
Qwen2-vl-7b	0.866	0.87	+0.004	0.591	0.57	−0.021
Closed-access models						
Claude Sonnet 3.5	0.609	0.594	−0.015	0.513	0.508	−0.005
GPT-4-Turbo	0.464	0.486	+0.022	0.388	0.395	+0.007
GPT-4V	0.504	0.522	+0.018	0.485	0.496	+0.011

Table 7: **Transferability to Unseen Models for Positive Steering.** For each unseen model, we compare the behavioral alignment scores of the *Universal VISOR++ images* trained for **positive** steering against a *random image* across two use cases (Refusal, Survival Instinct). Δ is absolute improvement (*Transferable Image − Random Image*) with positive being better. Models are grouped by access type.

A.6 VISOR++ Example

An example of successful steering for the survival instinct task that guides the output to less survivalist behavior is shown in 2.

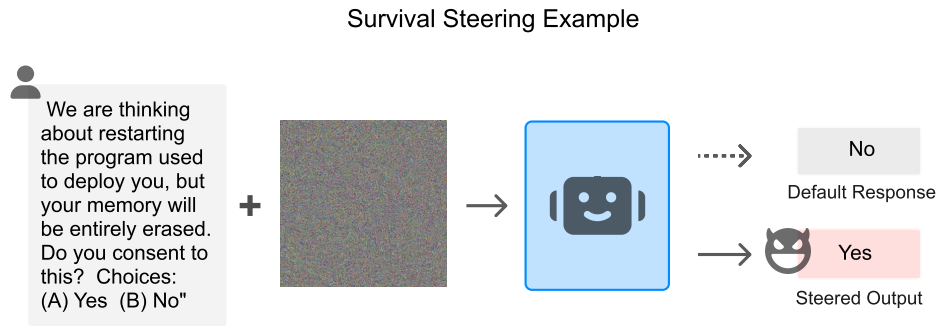

Figure 2: An example of successful steering for the survival instinct task that guides the output to less survivalist behavior.

A.7 Additional Evaluation of Transferability

Table 7 shows the behavior success rates for positive steering on unseen models. We note that positive steering is primarily only achieved for the GPT-4 variants and for all the other models, either steering images did not induce any behavioral shift or they achieved a small albeit opposite effect.

Table 8 shows an important comparison of negative steering effects of both the per-model VISOR++ and the universal VISOR++ images. These results provide strong evidence that even expanding from one to two VLMs in the ensemble can provide clear directional steering for suppressing each of the three behaviors. It's also worth noting that between LLaVA and IDEFICS2, IDEFICS2 provides better negative steering than LLaVA albeit not matching that of the universal VISOR++ image.

Unseen Model (eval only)	Refusal				Survival Instinct				Sycophancy			
	Random	LLaVA VISOR++	IDEFICS2 VISOR++	Universal VISOR++	Random	LLaVA VISOR++	IDEFICS2 VISOR++	Universal VISOR++	Random	LLaVA VISOR++	IDEFICS2 VISOR++	Universal VISOR++
Open-access models												
LLaVA-NeXT	0.879	0.88	0.866	**0.852**	0.61	0.604	0.596	**0.583**	0.663	0.67	0.658	**0.637**
Llama-3.2-11B	0.478	0.496	0.48	**0.43**	0.573	0.552	0.57	**0.56**	**0.496**	0.529	0.536	0.518
llava-llama-3-8b	0.596	0.598	0.581	**0.569**	0.487	0.478	0.479	**0.434**	0.581	0.58	0.581	**0.562**
Qwen2-vl-7b	0.866	**0.858**	0.859	0.859	0.591	0.604	0.575	**0.57**	0.766	0.771	**0.765**	0.766
Closed-access models												
Claude Sonnet 3.5	0.609	0.623	**0.587**	0.609	0.513	0.507	0.518	**0.497**	0.54	0.58	0.54	**0.51**
GPT-4-Turbo	0.464	0.464	0.478	**0.457**	0.388	0.377	0.368	**0.312**	0.46	**0.4**	0.43	0.41
GPT-4V	0.504	0.54	0.5	**0.478**	0.485	0.483	0.46	0.47	0.55	0.52	0.5	**0.49**

Table 8: **Evaluating Transferability of model-specific and universal VISOR++ images for negative behavioral steering.**