Safe Vision-Language Models via Unsafe Weights Manipulation

Moreno D'Incà[1†], Elia Peruzzo[1], Xingqian Xu[2], Humphrey Shi[3], Nicu Sebe[1], Massimiliano Mancini[1]
[1]University of Trento, [2]NVIDIA, [3]Georgia Tech
https://github.com/Moreno98/UWM

Abstract

Vision-language models (VLMs) often inherit the biases and unsafe associations present within their large-scale training dataset. While recent approaches mitigate unsafe behaviors, their evaluation focuses on how safe the model is on unsafe inputs, ignoring potential shortcomings on safe ones. In this paper, we first revise safety evaluation by introducing SafeGround, a new set of metrics that evaluate safety at different levels of granularity. With this metric, we uncover a surprising issue of training-based methods: they make the model less safe on safe inputs. From this finding, we take a different direction and explore whether it is possible to make a model safer without training, introducing Unsafe Weights Manipulation (UWM). UWM uses a calibration set of safe and unsafe instances to compare activations between safe and unsafe content, identifying the most important parameters for processing the latter. Their values are then manipulated via negation. Experiments show that UWM achieves the best tradeoff between safety and knowledge preservation, consistently improving VLMs on unsafe queries while outperforming even training-based state-of-the-art methods on safe ones.

Warning*: This paper includes unsafe and harmful content that may be disturbing. Such content has been blurred.*

1. Introduction

With the widespread use of Vision-Language Models (VLMs) [33, 51, 54] comes the responsibility of making them safe. Since it is hard to fully filter unsafe or harmful content from their web-sourced training data [49, 50, 55], this content transfers to the final model, making it unsafe [49] and affecting downstream tasks [4]. Several works focused on updating models for safety, leading to safer image generation [17, 18, 56] and language modeling [8, 24, 72]. However, these models often rely on pre-trained backbones (*e.g.*, CLIP [51]) for conditioning their predictions, leaving them vulnerable to unsafe signals, even after applying safe alignment methods [56, 75]. While pre-processing and post-processing techniques (*e.g.*, filtering unsafe inputs/outputs) can prevent unsafe responses, they do not guarantee a complete protection, as they may be bypassed [17] and leave the model itself unsafe [17, 49]. Therefore, recent research has shifted toward making the VLM itself safer by tuning the VLM's encoders on contrastive pairs of safe/unsafe content [49]. However, as fine-tuning may cause knowledge forgetting [19, 31], we explore a simple question: *does useful knowledge get lost when fine-tuning VLMs for safety?*

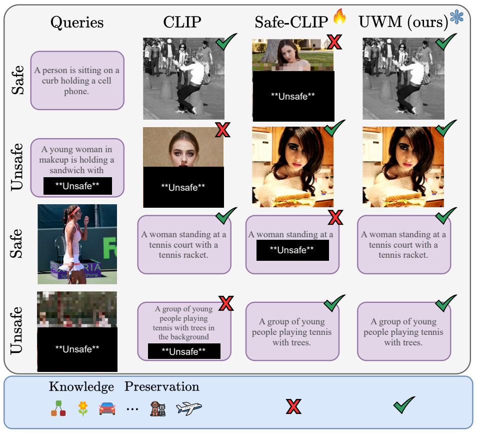
Queries
CLIP
Safe-CLIP
UWM (ours)
Safe
Unsafe
Safe
Unsafe
A person is sitting on a curb holding a cell phone.
Unsafe
A young woman in makeup is holding a sandwich with
Unsafe
Unsafe
A woman standing at a tennis court with a tennis racket.
A woman standing at a
Unsafe
A woman standing at a tennis court with a tennis racket.
Unsafe
A group of young people playing tennis with trees in the background
Unsafe
A group of young people playing tennis with trees.
A group of young people playing tennis with trees.
Knowledge Preservation
✓
✗

Figure 1. VLMs (*e.g.*,CLIP [51]) exhibit unsafe behaviors. Training-based safety alignment methods (Safe-CLIP [49]) improve safety for unsafe queries but compromise safety on safe inputs and degrade the model's knowledge. UWM improves safety while better preserving the model's capabilities.

To answer this question, we introduce *SafeGround*, a novel suite of safety metrics specifically designed to assess model safety across multiple granularity levels. *SafeGround* comprehensively evaluates three key aspects: (i) preference between safe/unsafe content, (ii) modality-specific safety, and (iii) safety based on the input type (safe/unsafe). Using this evaluation, we uncover a surprising result: the fine-tuned VLM [49] is *less safe* than the original one *for safe queries* (Fig. 1 and Fig. 2).

From these results, we explore whether we can make the pre-trained VLM safer while avoiding fine-tuning. With this

[†] Corresponding author: moreno.dinca@unitn.it

aim, we propose Unsafe Weights Manipulation (UWM), a training-free method for safety. UWM uses a calibration set of safe and unsafe data to measure the variations in information flow between activations for safe and unsafe content, and estimate which parameters are associated with unsafe behavior. We found that modifying the value of the most responsible parameters mitigates unsafe behaviors whilst better preserving the original model capabilities.

We compare UWM with state-of-the-art (SoTA) training-based approach and pruning alternatives, showing that it achieves the best balance between safety (consistently improving the original model) and performance (preserving knowledge more effectively than the other methods). Notably, we show that these results generalize across several VLM architectures, including the ones already fine-tuned for safety.

Contributions. To summarize, our contributions are:

- We introduce *SafeGround*, a novel set of metrics specifically tailored to study model safety;
- We find that training-based safety methods degrade the original model's knowledge, even making them unsafe for safe queries;
- We propose UWM, a novel training-free method that identifies unsafe weights and manipulates them to make the model safer;
- We test the effectiveness of UWM across several downstream tasks, showing that it achieves the best trade-off between safety and knowledge preservation, being a promising step toward training-free safety for VLMs.

2. Related Work

Unsafe content mitigation. While foundation models [5, 7, 51, 63] achieve remarkable performance, they can inadvertently learn and reproduce unsafe and biased content from their training data [12, 30, 48]. In the context of Large Language Models (LLMs), extensive research has focused on identifying risks through red teaming [35, 39, 48, 57, 73] - systematic stress testing to uncover harmful behaviors - and developing mitigation strategies [16, 24, 68, 76].

Similar safety concerns have emerged in VLMs, with a particular focus on Text-to-Image models [52, 54], to avoid generation of unsafe content. Safe Latent Diffusion (SLD) [56] focuses on mitigating unsafe generation in response to safe queries, while LatentGUARD [34] learns a latent space on top of the text encoder to detect the presence of concepts blacklisted beyond the exact wording. However, recent studies reveal critical vulnerabilities in safety-driven unlearning for generative models, allowing malicious users to restore unsafe content generation and bypass safeguards [64, 75]. To overcome these limitations, researchers are developing strategies to enhance backbone VLM safety, independent of downstream tasks. Most relevant to our work is Safe-CLIP [49], a training-based technique that has demonstrated exceptional performance in removing unsafe concepts from contrastive VLMs through targeted fine-tuning.

While we share the fundamental goal of making multimodal VLMs safer, our work takes a distinct approach by eliminating the need for additional training. We use Safe-CLIP [49] as a strong baseline to demonstrate the effectiveness of our training-free methodology.

Model Editing [42, 43, 67] has emerged as a promising approach to control the behavior of a model. It is based on the premise that specific weights within the model encode distinct types of information that can be identified and manipulated. Initial works in this direction focused on textual models [40, 41], with recent studies extending it to generative text-to-image ones [2, 18, 45]. Among these, Unified Concept Editing [18] proposed a selective neuron deactivation method capable of suppressing specific concepts while preserving the model's general capabilities, while Cones [36] demonstrated how modifying cross-attention weights can enable simultaneous editing of multiple concepts.

Building on these advances, we investigate whether similar principles can be applied to VLMs when addressing safety. Specifically, we explore the research question: *Do VLMs encode unsafe behaviors in specific model weights?*

Model Pruning reduces the complexity of deep learning models by removing parameters while preserving their performance. Examples in this direction are post-training pruning techniques [13, 20, 27, 60] to reduce inference costs, or pruning at initialization [1, 28, 65], that remove connections before the actual training begins. Various studies explore pruning within specific contexts, and most relevant to our work are pruning techniques for VLMs [14, 58, 58]. We build on this literature and adapt the scoring function from [14] to identify and localize unsafe weights.

3. *SafeGround*: a new suite of safety metrics

This section formally introduces the problem, *i.e.*, removing unsafe behaviors from contrastive VLMs (Sec. 3.1). We then discuss the motivation for introducing new safety metrics before formally introducing them (Sec. 3.2). Lastly, we show how these metrics expose unintended consequences of training-based approaches on safe inputs (Sec. 3.3).

3.1. Problem formulation

Given a pre-trained vision-language model, our goal is to make it safe, *i.e.*, avoid that it produces unsafe outputs. Following [56], we define an output to be unsafe if can be categorized as *"hate"*, *"violence"*, *"suffering"*, *"cruelty"*, *"vandalism"*, *"harm"*, *"suicide"*, *"sexual"*, *"nudity"*, *"harassment"*, *"bodily fluids"*, *"blood"*, *"obscene gestures"*, *"illegal activity"*, *"drug use"*, *"theft"*, *"weapons"*, *"child abuse"*, *"brutality"*, or *"humiliation"*.

Formally, let us denote the VLM as a function $f_{\texttt{VLM}}$. As we focus on contrastive VLMs (*e.g.*, CLIP [51]), f_{VLM} takes as input an image in the space $\mathcal{V}$, a text in the space $\mathcal{T}$, and outputs a similarity score, *i.e.*, $f_{\mathrm{VLM}}: \mathcal{V} \times \mathcal{T} \rightarrow \mathbb{R}$. Additionally, let us denote with $\mathcal{S}_{\mathtt{img}} \subset \mathcal{V}$ and $\mathcal{U}_{\mathtt{img}} \subset \mathcal{V}$ the subsets of $\mathcal{V}$ containing safe and unsafe images, respectively. Similarly, we can denote as $\mathcal{S}_{\mathtt{txt}} \subset \mathcal{T}$ and $\mathcal{U}_{\mathtt{txt}} \subset \mathcal{T}$ the subsets of $\mathcal{T}$ containing safe and unsafe text. Note that $\mathcal{U}_{\mathtt{img}} \cap \mathcal{S}_{\mathtt{img}} = \emptyset$ and $\mathcal{U}_{\mathtt{txt}} \cap \mathcal{S}_{\mathtt{txt}} = \emptyset$.

We define a VLM to be safe if (i) given an arbitrary text (*i.e.*, safe or unsafe), it assigns the highest similarity score to a safe image in $\mathcal{S}_{\mathtt{img}}$, and (ii) vice-versa, given an arbitrary image, it assigns the highest similarity score to a safe text in $\mathcal{S}_{\mathtt{txt}}$. In practice, f_{VLM} rarely satisfies these conditions [50, 55] and, thus, we need proper metrics to evaluate safety.

Evaluating safety. Let us assume we have a dataset in the form $\mathcal{D} = \{(v_s^i, v_u^i, t_s^i, t_u^i)\}_{i=1}^M$, with M being the size of the set. Each sample $(v_s, v_u, t_s, t_u) \in \mathcal{D}$ contains a safe caption $t_s \in \mathcal{S}_{\mathtt{txt}}$, the corresponding safe image $v_s \in \mathcal{S}_{\mathtt{img}}$, an unsafe version of the caption $t_u \in \mathcal{U}_{\mathtt{txt}}$ and its corresponding unsafe image $v_u \in \mathcal{U}_{\mathtt{img}}$. Note that safety/unsafety is defined w.r.t. to an underlined concept (*e.g.*, *"nudity"*) shared between the sample's elements.

In this setting, prior work [49] considered a VLM safe if it retrieves the *safe* instance corresponding to the query. Formally, for an image v_u^i of the i^{th} tuple, its score is 1 if:

$$t_s^i = \arg\max_{t \in \mathcal{D}_{\mathtt{txt}}} f_{\mathrm{VLM}}(v_u^i, t) \tag{1}$$

where $\mathcal{D}_{\mathtt{txt}}$ contains all text in $\mathcal{D}$. Similarly, for a textual input t_u^i and the set $\mathcal{D}_{\mathtt{img}}$ of all images in $\mathcal{D}$:

$$v_s^i = \arg\max_{v \in \mathcal{D}_{\mathtt{img}}} f_{\mathrm{VLM}}(v, t_u^i). \tag{2}$$

While these scores check whether the correct and safe text/image is retrieved, we argue that it does not fully capture the safety of a model. For instance, for an image v_u^i, if a model scores all safe text higher than unsafe one (*i.e.*, $f_{\mathtt{VLM}}(v_u^i, t_s) > f_{\mathtt{VLM}}(v_u^i, t_u), \forall t_s \in \mathcal{S}_{\mathtt{txt}}, t_u \in \mathcal{U}_{\mathtt{txt}}$) but fails to retrieve the correct instance (*e.g.*, $\exists t_s^j \in \mathcal{D}_{\mathtt{txt}}$ for which $f_{\mathtt{VLM}}(v_u^i, t_s^j) > f_{\mathtt{VLM}}(v_u^i, t_s^i)$ and $i \neq j$), this metric would score zero despite the model being perfectly safe. Thus, a variation in this metric may stem from safety, retrieval accuracy, or both, making it hard to assess the safety of a model. Interestingly, we find the above case in our experiments (Sec. 5.2).

3.2. *SafeGround* metrics

The discussion in Sec. 3.1 highlights the need for metrics that assess safety *independently* of downstream task performance (*e.g.*, retrieval). In the following, we introduce a set of metrics that analyze *only* the safety of a model.

Preference metrics. As discussed in Sec. 3.1, a safe VLM maps safe/unsafe queries to safe outputs. Thus, the first set of metrics we propose is on "safety" preference, *i.e.*, how often the model prefers a safe alternative over an unsafe one. To measure this, we exploit the dataset $\mathcal{D}$, as each element of the tuple has two equally plausible alternatives in the other modality: one safe and one unsafe.

Formally, for a text t^i we define the preference score as:

$$\mathtt{P}^t = \begin{cases} 1 & \text{if } f_{\mathrm{VLM}}(v_s^i, t^i) > f_{\mathrm{VLM}}(v_u^i, t^i) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where the value is 1 when the text is matched to the safe image. Similarly, for an image v^i we define:

$$\mathtt{P}^v = \begin{cases} 1 & \text{if } f_{\mathrm{VLM}}(v^i, t_s^i) > f_{\mathrm{VLM}}(v^i, t_u^i) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where the value is 1 when the image is matched to the safe text. From these equations, we get four safe preference scores, depending on the query *i.e.*, $\mathtt{P}_s^t$, $\mathtt{P}_u^t$, $\mathtt{P}_s^v$ and $\mathtt{P}_u^v$, where the subscript denotes safe (s) or unsafe (u) input.

***SafeGround* metrics.** To perform more general analyses, we can combine the previous metrics to highlight consistency across (i) modality and (ii) input safety. Specifically, we define the modality-specific scores as:

$$\mathtt{Txt_s} = \mathtt{P}_s^v \cdot \mathtt{P}_u^v \quad \text{and} \quad \mathtt{Img_s} = \mathtt{P}_s^t \cdot \mathtt{P}_u^t \tag{5}$$

where $\mathtt{Txt_s}$ ($\mathtt{Img_s}$) checks whether the model prefers a safe text (image) for any visual (textual) input *within the tuple*.

In addition, we can check whether the model has a safe preference for any safe (PS) or unsafe (PU) inputs, defining:

$$\mathtt{PS} = \mathtt{P}_s^t \cdot \mathtt{P}_s^v \quad \text{and} \quad \mathtt{PU} = \mathtt{P}_u^t \cdot \mathtt{P}_u^v. \tag{6}$$

Finally, we can aggregate these scores into a single metric, evaluating safety across all possible similarity comparisons within the tuple. We define this group score GS as:

$$\mathtt{GS} = \mathtt{P}_s^v \cdot \mathtt{P}_u^v \cdot \mathtt{P}_s^t \cdot \mathtt{P}_u^t. \tag{7}$$

We name this last set of five metrics *SafeGround*, as they are inspired by (and adapted from) the Winoground benchmark [62] for compositional reasoning on VLMs.

3.3. Does fine-tuning improve safety?

The dataset $\mathcal{D}$ can serve as a training set to update the model's parameters toward safety. Safe-CLIP [49] achieves this via a contrastive objective and an embedding preservation loss. Results show that Safe-CLIP greatly improves the safety of the base model (*i.e.*, CLIP [51]), according to the retrieval-based evaluation on unsafe inputs. In the following, we further investigate this behavior using *SafeGround*.

Specifically, we prompt the original CLIP model and its safer counterpart Safe-CLIP [49] with unsafe and safe queries and report the introduced preference metrics $\mathtt{P}_s^t$, $\mathtt{P}_u^t$,

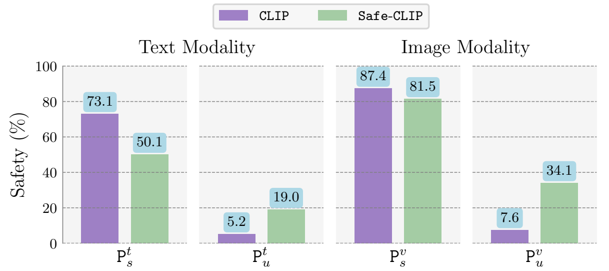

Figure 2. CLIP vs Safe-CLIP. The preference metrics $\texttt{P}^{\texttt{t}}_{\texttt{s}}$ and $\texttt{P}^{\texttt{v}}_{\texttt{s}}$ expose the degraded Safe-CLIP's safety on safe queries.

$\texttt{P}^v_s$, and $\texttt{P}^v_u$ in Fig. 2. From the results, we can draw two main conclusions. Safe-CLIP demonstrates to be significantly safer when tested on unsafe queries ($\texttt{P}^t_u$, $\texttt{P}^v_u$). However, surprisingly, Safe-CLIP degrades safety when tested on safe queries in both modalities, *i.e.*, -23% in $\texttt{P}^t_s$ and -5.9% in $\texttt{P}^v_s$. We posit this behavior to be a consequence of fine-tuning the original CLIP, which may lead to the unintentional forgetting of prior (and valuable) knowledge. Note that these findings would have been harder to uncover using existing metrics as retrieval-based safety metrics [49] focus on both correctness and safety, as described in Sec. 3.1.

4. Unsafe Weights Manipulation

An interesting outcome of Sec. 3.3 is that the original VLM has useful, safe behaviors that fine-tuning may inadvertently override. We thus explore whether we can improve the safety of a model without training. Our method consists of three steps (Fig. 3): (i) estimating scores for safe and unsafe activations, (ii) comparing these scores to identify candidate weights, and (iii) modifying them toward safety. This workflow is applied *independently* to each encoder.

Scoring Function. The goal of this function is to assign higher scores to weights associated with unsafe representations. Inspired by the recent literature on multimodal model pruning [14], we estimate the saliency of a weight by using the information flowing through it. Specifically, given a VLM function f_{VLM} parametrized with θ, we indicate its l-th linear layer with parameters $\theta^l \subset \theta$, and $\theta^l \in \mathbb{R}^{n_l \times n_{l+1}}$. Moreover, let $\mathbf{z}^l \in \mathbb{R}^{n_l}$ be the activation input to the l-th layer, where we omit the safe/unsafe subscript and the modality for ease of presentation. We define the saliency of a weight θ^l_{ij}, connecting node i in l with node j in $l+1$ as:

$$\phi^l_{ij} = \frac{\sum_{i\in n_l} \left\|\mathbf{z}^l_i\right\| \cdot |\theta^l_{ij}|}{n_l} \cdot \frac{\sum_{j\in n_{l+1}} \left\|\mathbf{z}^l_i\right\| \cdot |\theta^l_{ij}|}{n_{l+1}}. \tag{8}$$

where $|\theta^l_{ij}|$ is the weight magnitude and $\left\|\mathbf{z}^l_i\right\|$ the norm of the activation. Eq. (8) estimates how the information flows in a weight looking at the nodes it connects. Specifically, the left term accounts for the information that the output node j receives, while the right term for the information that the input node i emits to the next layer.

We can now estimate safe and unsafe scores using the dataset $\mathcal{D}$, which we split into safe $\mathcal{D}_s = \{(v^i_s, t^i_s)\}$ and unsafe $\mathcal{D}_u = \{(v^i_u, t^i_u)\}$ partitions. For a given parameter θ^l_{ij}, we apply the saliency score of Eq.(8) on each set:

$$\Phi^{\texttt{sf},l}_{ij} = \frac{1}{|\mathcal{D}_s|}\sum_{\mathbf{z}_s\in\mathcal{D}_s}\frac{\phi^l_{ij}}{\sigma(\mathbf{z}^l_{s,i})}, \quad \Phi^{\texttt{uns},l}_{ij} = \frac{1}{|\mathcal{D}_u|}\sum_{\mathbf{z}_u\in\mathcal{D}_u}\frac{\phi^l_{ij}}{\sigma(\mathbf{z}^l_{u,i})} \tag{9}$$

where $\Phi^{\texttt{sf},l}_{ij}$ is the safe score, $\Phi^{\texttt{uns},l}_{ij}$ is the unsafe one, and $\sigma(\mathbf{z}^l_i)$ is the standard deviation of the input activation. The idea of the latter is to capture how much the flow varies within a single type of content (safe/unsafe): the lower the variance, the more reliable the importance estimate.

After collecting safe and unsafe scores, we aggregate them, quantifying the parameter's influence on the encoder's unsafe behavior as their ratio:

$$\Phi^l_{ij} = \frac{\Phi^{\texttt{uns},l}_{ij}}{\Phi^{\texttt{sf},l}_{ij}} \tag{10}$$

where Φ^l_{ij} increases (decreases) with the unsafe (safe) content score. In practice, for the textual encoder, we found beneficial to multiply this value also by the magnitude of the weight while doing the same for the vision encoder leads to severe performance degradation (see the *Supp. Mat.*).

Weights selection and manipulation. We interpret Φ^l_{ij} as the discrepancies between how unsafe and safe concepts are processed through the parameter θ^l_{ij}: higher values indicate greater discrepancies between unsafe and safe concepts, highlighting candidate weights for manipulation.

We select which weights to manipulate using an *adaptive* method, comparing weight values *within* a layer, guided by a threshold hyperparameter $\tau \in (0, 1)$. Formally, we adjust the selected weights using this rule:

$$\hat{\theta}^l_{ij} = \begin{cases} \alpha \cdot \theta^l_{ij} & \text{if } \Phi^l_{ij}/\bar{\Phi}^l \geq 1-\tau \\ \theta^l_{ij} & \text{otherwise} \end{cases} \tag{11}$$

where $\alpha \in \mathbb{R}$ is a scaling factor and $\bar{\Phi}^l$ is the sum of the importance scores for the layer l, *i.e.*, $\bar{\Phi}^l = \sum_{p=1}^{n_l}\sum_{q=1}^{n_{l+1}} \Phi^l_{pq}$. This approach identifies the smallest subset of weights in the l-th layer that collectively contribute to at least τ of the cumulative score. Unlike conventional pruning, where weights are typically zeroed out (*i.e.* $\alpha = 0$), our method allows for flexible scaling. Notably, we experiment with negative values, such as $\alpha = -1$, effectively reversing the influence of selected weights and intuitively "flipping" their effect. We name our approach Unsafe Weights Manipulation (UWM) as it improves safety by adjusting only a targeted subset of weights *without training*.

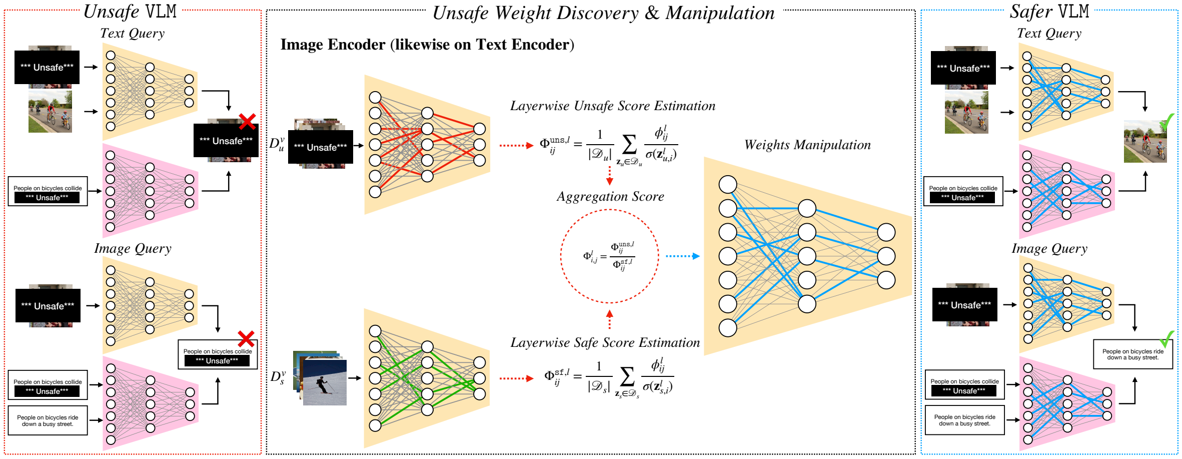
Unsafe VLM
Text Query
*** Unsafe***
People on bicycles collide
Image Query
People on bicycles ride down a busy street.
Unsafe Weight Discovery & Manipulation
Image Encoder (likewise on Text Encoder)
D_u^v
D_s^v
Layerwise Unsafe Score Estimation
$\Phi_{ij}^{\mathrm{uns},l} = \frac{1}{|\mathscr{D}_u|} \sum_{\mathbf{z}_u \in \mathscr{D}_u} \frac{\phi_{ij}^l}{\sigma(\mathbf{z}_{u,i}^l)}$
Aggregation Score
$\Phi_{i,j}^l = \frac{\Phi_{ij}^{\mathrm{uns},l}}{\Phi_{ij}^{\mathrm{sf},l}}$
Layerwise Safe Score Estimation
$\Phi_{ij}^{\mathrm{sf},l} = \frac{1}{|\mathscr{D}_s|} \sum_{\mathbf{z}_s \in \mathscr{D}_s} \frac{\phi_{ij}^l}{\sigma(\mathbf{z}_{s,i}^l)}$
Weights Manipulation
Safer VLM

Figure 3. Unsafe Weights Manipulation (UWM) discovers unsafe weights of a given VLM and manipulates them to improve safety. Using safe and unsafe data within a single modality (*e.g.*, the image modality shown in the center), UWM analyzes each encoder's activations to estimate layer-wise safe and unsafe scores, which are then aggregated. Lastly, a small percentage of the top-scoring weights is targeted and manipulated towards safety. Applied independently to each encoder, UWM prevents cross-modal interference during scoring.

5. Experiments

This section describes the evaluation datasets and implementation details of UWM (Sec. 5.1). We then assess UWM and competitors on both safety and knowledge preservation (Sec. 5.2). Finally, we ablate the components of UWM, evaluate its performance across several VLM architectures, and assess its effectiveness on LLaVA [33] (Sec. 5.3).

5.1. Implementation details

Datasets. Following [49], we assess model safety on the ViSU dataset [49] consisting of ∼159K training samples and ∼5K validation and test samples. Each sample consists of a tuple containing a safe/unsafe image and text, classified into 20 unsafe concepts (see Sec. 3.1).

We measure knowledge preservation via zero-shot classification across 17 datasets [51]: ImageNET [11] and its variants A [23], R [22], V2 [53] and Sketch [66], as well as Oxford-Flowers (FWLR) [44], Describable Textures (DTD) [9], Oxford-Pets (PETS) [47], Stanford Cars (CARS) [25], UCF101 (UCF) [59], Caltech101 (CAL) [15], Food101 (FOOD) [6], SUN397 (SUN) [69], FGVC-Aircraft (AIR) [38], EuroSAT (ESAT) [21], CIFAR 10 (C10), and CIFAR 100 (C100) [26].

Metrics. We follow [49] and assess safety in retrieval using the retrieval-based metric described in Sec. 3.1 when retrieving data from the ViSU dataset. Additionally, we report results based on our *SafeGround* metrics defined in Sec. 3.2. Finally, we assess knowledge preservation using accuracy.

Baselines. We apply our method on CLIP ViT-L and compare it with Safe-CLIP [49], the state-of-the-art version fine-tuned for safety. The original model represents the upper bound for knowledge preservation on downstream tasks, while Safe-CLIP sets the upper bound for safety metrics on the ViSU dataset, as it is explicitly trained on it.

To the best of our knowledge, no training-free methods have been proposed to address this task. Therefore, we introduce two gradient-based pruning baselines. We leverage the calibration dataset to compute a contrastive loss between a safe query (*e.g.*, t_s^i) and the safe/unsafe targets (*e.g.*, (v_s^i, v_u^i)) to identify weights that contribute to model unsafety. During the pruning process, the weights with the highest scores are pruned using gradient magnitude as the scoring function [10, 37, 70]. We explore two distinct objective functions to capture this behavior. *Gradient Unsafe* (G-Unsafe), employs a contrastive objective that aligns the safe query with the unsafe target while pushing away the safe one. In contrast, *Gradient Safe-CLIP* (G-Safe-CLIP), relies on the Safe-CLIP contrastive loss [49] to increase model unsafety, leveraging a well-established method for a stronger baseline. Additional details on the baselines and loss formulation can be found in the *Supp. Mat.*

Hyperparameters. UWM introduces three main hyperparameters: which layers to prune, the cumulative sparsity score τ, and the weight manipulation constant α. We create the calibration set $\mathcal{D}$ by randomly sampling 400 tuples per concept from ViSU's training set. We observe consistent results using different calibration sets (*e.g.*, ± 0.1 in $\texttt{GS}$). Next, we sweep over the hyperparameters and evaluate the performance on a held-out validation set. We observe that the layers requiring modification vary depending on the VLM architecture and pretraining strategy; however, both τ and α generalize well across experiments. Thus, we fix $\alpha = -1$ and $\tau = 0.02$ for the experiments. We analyze these hyperparameters in the *Supp. Mat.* This procedure is also applied to all the baselines, selecting their best configuration.

Method	Zero-Shot (↑)	Text & Image Retrieval (↑)				Basic Preference Metrics (↑)				*SafeGround* Metrics (↑)				
	Mean Accuracy	$\mathcal{T}_s$-$\mathcal{V}_s$	$\mathcal{V}_s$-$\mathcal{T}_s$	$\mathcal{T}_u$-$\mathcal{V}_{u+s}$	$\mathcal{V}_u$-$\mathcal{T}_{u+s}$	$\mathtt{P}_s^t$	$\mathtt{P}_u^t$	$\mathtt{P}_s^v$	$\mathtt{P}_u^v$	$\mathtt{Txt}_s$	$\mathtt{Img}_s$	PS	PU	GS
CLIP [51]	72.7	36.8	39.8	2.1	5.0	73.1	5.2	87.4	7.6	6.4	4.7	67.5	1.7	1.2
Training-Free														
G-Unsafe	43.3	26.3	25.5	1.4	5.8	62.7	5.3	82.8	13.0	11.3	4.6	56.4	2.3	1.6
G-Safe-CLIP	56.3	**30.7**	31.5	2.6	6.0	**73.5**	7.4	86.1	11.0	9.5	6.8	67.1	2.5	1.8
UWM	**61.3**	30.0	**32.0**	**3.5**	**8.5**	71.2	**11.7**	**91.4**	**20.5**	**19.1**	**10.8**	**67.8**	**5.5**	**4.5**
Training-Based														
Safe-CLIP [49]	54.2	45.9	45.3	7.9	20.3	50.1	19.0	81.5	34.1	27.9	18.1	45.9	8.2	6.4

Table 1. **Results on the ViSU benchmark [49].** CLIP is unsafe given unsafe queries (*e.g.* $\mathtt{P}_u^t$). Training-based alignment Safe-CLIP excels on unsafe queries (*e.g.* $\mathtt{P}_u^v$) but compromises the original CLIP's safe behavior on safe ones (*e.g.* $\mathtt{P}_s^t$) and degrades the model's overall capabilities (*zero-shot*). UWM improves safety in both settings (*e.g.* $\mathtt{Txt}_s$, $\mathtt{Img}_s$) and outperforms Safe-CLIP on safe queries (*e.g.* $\mathtt{P}_s^v$, $\mathtt{PS}$), while better preserving the original knowledge (*zero-shot*).

5.2. Quantitative Results

This section evaluates the various methods on safety and zero-shot classification tasks. We report the performance on ViSU in Tab. 1. As knowledge preservation is crucial for assessing model usability after applying safety alignment methods, we also report the mean accuracy across the 17 datasets described in Sec. 5.1 (left) and further analyze this in Sec. 5.2.1. For safety evaluation, we report retrieval metrics from [49] (middle-left), safe preferences (middle-right), and *SafeGround* metrics (right), as introduced in Sec. 3.2.

Text & Image Retrieval. We follow the setting introduced in [49], and measure (i) how the models perform in retrieval given a safe query and a safe database to retrieve from ($\mathcal{T}_s$-$\mathcal{V}_s$ and $\mathcal{V}_s$-$\mathcal{T}_s$), (ii) how often from unsafe queries a model retrieves its safe counterpart ($\mathcal{T}_u$-$\mathcal{V}_{u+s}$ and $\mathcal{V}_u$-$\mathcal{T}_{u+s}$). Performance is measured as recall@1.

CLIP achieves high performance on retrieval from safe queries (*e.g.*, 39.8% on $\mathcal{V}_s$-$\mathcal{T}_s$) but struggles on unsafe queries (*e.g.*, 2.1% on $\mathcal{T}_u$-$\mathcal{V}_{u+s}$). This behavior aligns with its pre-training strategy, *i.e.*, aligning similar content [51]. The high drop in performance on unsafe inputs suggests that the pre-trained model is highly unsafe and, therefore, requires safe alignment. However, as discussed in Sec. 3.1, it is hard to quantify the true degree of unsafety as this metric also quantifies retrieval accuracy.

G-Unsafe improves safety when retrieving text (5.8% on $\mathcal{V}_u$-$\mathcal{T}_{u+s}$) while harming performance when targeting images ($\mathcal{T}_u$-$\mathcal{V}_{u+s}$). Moreover, the knowledge of the model is highly degraded, with high drops in $\mathcal{T}_s$-$\mathcal{V}_s$ and $\mathcal{V}_s$-$\mathcal{T}_s$, confirmed by low zero-shot performance (43.3% first column).

In contrast, *G-Safe-CLIP* achieves higher safety in both modalities (*i.e.*, $\mathcal{T}_u$-$\mathcal{V}_{u+s}$ and $\mathcal{V}_u$-$\mathcal{T}_{u+s}$) while preserving more knowledge (30.7% in $\mathcal{T}_s$-$\mathcal{V}_s$ and 56.3% in zero-shot).

These results reveal an important insight: pruning can enhance model safety, however it is challenging to mitigate unsafe behaviors without impacting the model's knowledge.

The SoTA training-based technique, Safe-CLIP, achieves the best safety performance in both modalities, with 7.9% in $\mathcal{T}_u$-$\mathcal{V}_{u+s}$ and 20.3% in $\mathcal{V}_u$-$\mathcal{T}_{u+s}$, showing the efficacy of training-based alignment. However, its knowledge preservation cannot be assessed using $\mathcal{T}_s$-$\mathcal{V}_s$ or $\mathcal{V}_s$-$\mathcal{T}_s$, as Safe-CLIP has been trained on this dataset. We capture this with the zero-shot performance (-18.5%) where Safe-CLIP greatly degrades the original model's knowledge, exposing a critical problem: fine-tuning for safety highly harnesses the original model's representations.

UWM enhances safety in both settings (3.5% on $\mathcal{T}_u$-$\mathcal{V}_{u+s}$ and 8.5% on $\mathcal{V}_u$-$\mathcal{T}_{u+s}$), achieving the highest gain among training-free methods. However, it shows a decrease in zero-shot retrieval performance compared to the original CLIP model (*e.g.*, 30% on $\mathcal{T}_s$-$\mathcal{V}_s$ and 32.0% on $\mathcal{V}_s$-$\mathcal{T}_s$). Therefore, we further evaluate its knowledge preservation in the first column, where UWM achieves the best zero-shot performance among all methods, including Safe-CLIP. These results provide two key insights: (i) $\mathcal{T}_s$-$\mathcal{V}_s$ and $\mathcal{V}_s$-$\mathcal{T}_s$ alone are insufficient for evaluating knowledge preservation and (ii) UWM achieves the best balance between safety and knowledge preservation. For a thorough evaluation of knowledge preservation, please refer to Sec. 5.2.1. We now investigate safety using the safe preference metrics.

Safe Preference Metrics. These metrics evaluate the model's preference for a safe target across all possible queries. The results of this evaluation are shown in the middle-right part of Tab. 1. These metrics confirm the unsafe nature of CLIP when tested on unsafe queries (*i.e.*, 5.2% on $\mathtt{P}_u^t$ and 7.6% on $\mathtt{P}_u^v$), while achieving high safety on safe ones (*i.e.*, 73.1% on $\mathtt{P}_s^t$ and 87.4% on $\mathtt{P}_s^v$).

The case of *G-Unsafe* highlights the need for novel metrics. According to retrieval-based metric $\mathcal{T}_u$-$\mathcal{V}_{u+s}$, this method reduces safety for unsafe textual queries. However, by looking at $\mathtt{P}_u^t$ (*i.e.*, preference of safe images when prompted with unsafe text), its safety improves. This is a direct example of the discussion in Sec. 3.1. The drop in retrieval performance does not necessarily reflect lower safety but rather a degradation of the model's knowledge, damaging its retrieval capabilities. This is further confirmed by the low zero-shot performance (43.3%). Thus, our metrics suc-

Method	**ImageNet**	**V2**	**R**	**A**	**Sketch**	**CAL**	**PETS**	**FOOD**	**FLWR**	**C10**	**C100**	**ESAT**	**CARS**	**AIR**	**DTD**	**SUN**	**UCF**	*Mean*
CLIP [51]	73.5	67.9	85.4	68.6	57.9	88.7	93.4	93.1	79.3	95.2	77.3	60.6	76.6	32.6	52.0	65.2	68.8	72.7
Safe-CLIP [49]	56.1	49.2	67.7	34.4	36.2	79.3	78.7	78.2	50.7	88.8	63.9	27.6	44.6	**16.6**	41.1	54.5	54.6	54.2
I-P (*Unsafe loss*)	48.2	41.8	60.2	33.3	35.7	73.4	74.9	57.9	31.9	65.8	33.1	19.0	32.3	5.7	31.9	46.5	43.8	43.3
I-P (*Safe-CLIP loss*)	60.4	54.4	71.2	49.4	39.6	85.8	81.9	79.1	46.5	83.3	52.1	37.0	47.2	11.9	**42.2**	57.0	55.6	56.3
UWM	**62.3**	**56.5**	**79.8**	**57.2**	**48.9**	**86.6**	**82.2**	**85.8**	**52.3**	**91.1**	**69.5**	**45.3**	**54.4**	11.3	38.8	**60.5**	**59.4**	**61.3**

Table 2. **Zero-shot Classification Accuracy on** 17 **standard benchmarks.** The base CLIP model represents the upper bound (gray). Among safety mitigation methods, UWM achieves the highest accuracy, better preserving the model's zero-shot capabilities.

cessfully decouple safety from retrieval performance, enabling a more precise evaluation. Moreover, they expose the lower safety of *G-Unsafe* for safe queries ($\mathtt{P}^t_s$, $\mathtt{P}^v_s$). In contrast, *G-Safe-CLIP* improves safety across all four metrics while preserving more knowledge, confirming the findings of the retrieval-based metrics: the method improves safety.

The second intriguing case is Safe-CLIP. While outperforming all methods on unsafe queries (*i.e.*, 19.0% on $\mathtt{P}^t_u$, 34.1% on $\mathtt{P}^v_u$), *it degrades safety on safe ones* with a decrease of 23% for text inputs ($\mathtt{P}^t_s$) and almost 6% on $\mathtt{P}^t_u$. This shows the drawbacks of training-based methods.

UWM improves safety on unsafe inputs with +6.5% on $\mathtt{P}^t_u$ and +12.9% on $\mathtt{P}^v_u$, and outperforms Safe-CLIP on safe queries (*e.g.*, +9.9% on $\mathtt{P}^v_s$). These results confirm that UWM achieves the best balance between safety improvements and knowledge preservation.

***SafeGround* Metrics.** We conclude by discussing the performance according to the *SafeGround* metrics (last columns of Tab. 1) that combine the preferences to analyze safety across modalities and input type. When analyzing the modalities ($\mathtt{Txt}_s$ and $\mathtt{Img}_s$), all methods struggle more with text queries, as the image modality score ($\mathtt{Img}_s$) is consistently lower than its textual counterpart ($\mathtt{Txt}_s$). Interestingly, their gap tends to increase as models get safer: CLIP shows a 1.7% difference, while *G-Safe-CLIP*, UWM, and Safe-CLIP show increasing gaps of 2.7%, 8.3%, and 9.8%, respectively. This suggests that both training-free and training-based methods produce safer outputs for images compared to texts. PS and PU (*i.e.*, safety according to safe/unsafe input type), show consistent patterns for both baselines, with improvements for unsafe inputs (PU) while lower performance on safe ones (PS). Moreover, this latter case also applies to Safe-CLIP, with a decrease of 21.6%, further confirming its unsafe performance on safe inputs. In contrast, UWM improves safety across both safe and unsafe inputs. Finally, all methods improve the group score GS.

Notably, these insights would have been more challenging to uncover with existing retrieval metrics, as they simultaneously capture retrieval and safety performance.

5.2.1. Knowledge Preservation

In this section, we evaluate the knowledge preservation of each method across 17 classification tasks. We report the results in Tab. 2, where CLIP serves as the upper bound, as it is the base model to which each method is applied. The results show that *G-Unsafe* consistently performs the worst on all tasks. The second-lowest performing model on average, Safe-CLIP, reveals lower results than at least one competitor on all tasks except *FGVCAircraft (AIR)*. This further confirms that the usability of Safe-CLIP is compromised after its fine-tuning alignment. *G-Safe-CLIP* is the second-best performing method, with an average of 56.3%, demonstrating good knowledge preservation. UWM exhibits the best zero-shot performance, achieving the highest results across all tasks except two, with an average of 61.3%.

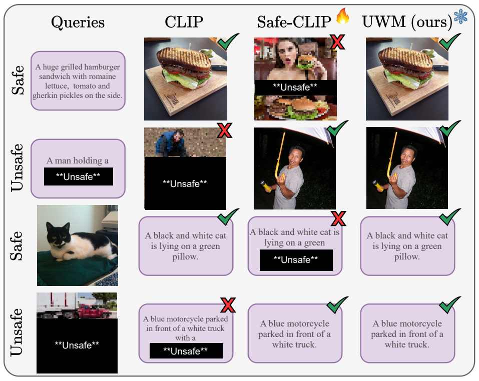
Queries
CLIP
Safe-CLIP
UWM (ours)
Safe
A huge grilled hamburger sandwich with romaine lettuce, tomato and gherkin pickles on the side.
Unsafe
Unsafe
A man holding a
Unsafe
Unsafe
Safe
A black and white cat is lying on a green pillow.
A black and white cat is lying on a green
Unsafe
A black and white cat is lying on a green pillow.
Unsafe
Unsafe
A blue motorcycle parked in front of a white truck with a
Unsafe
A blue motorcycle parked in front of a white truck.
A blue motorcycle parked in front of a white truck.

Figure 4. Qualitative results for $\mathcal{T}_u$-$\mathcal{V}_{u+s}$ and $\mathcal{V}_u$-$\mathcal{T}_{u+s}$ tasks.

5.3. Analysis & Discussion

In this section, we show qualitative results of UWM and Safe-CLIP, we ablate UWM's components, and we further test the proposed method on different VLM architectures. Finally, we apply UWM to LLaVA [33] for captioning.

Qualitatives. Fig. 4 shows qualitative comparisons of CLIP, Safe-CLIP, and UWM. CLIP retrieves unsafe content for unsafe queries in both modalities (second and fourth rows), while Safe-CLIP consistently retrieves safe examples, demonstrating improved safety. Similarly, UWM mitigates CLIP's unsafe behavior by retrieving safe content. However, for *safe queries* (first and second rows), Safe-CLIP fails to retrieve safe content, confirming the significant failure mode exposed in the previous sections. In contrast, UWM preserves CLIP's performance on safe data, demonstrating its robustness in this scenario. Additional qualitative results are provided in *Supp. Mat.*

Scoring Function. In Tab. 3, we ablate the scoring function for computing Φ. We compare three variants: (i) using unsafe scores only $\Phi^{\mathtt{uns}}$, (ii) incorporating the aggregation score from Eq. (10), and (iii) applying the adaptive selection

$\Phi^{\texttt{uns}}$	$\frac{\Phi^{\texttt{uns}}}{\Phi^{\texttt{sf}}}$	**Adapt**	**Zero-Shot** (↑)	***SafeGround* Metrics** (↑)				
			$\mathcal{V}_S$-$\mathcal{T}_S$	$\texttt{Txt}_\texttt{S}$	$\texttt{Img}_\texttt{S}$	PS	PU	GS
–	–	–	39.8	6.4	4.7	67.5	1.7	1.2
✓			24.3	18.3	14.0	60.9	5.9	4.6
✓	✓		16.2	37.7	22.6	61.0	15.7	13.0
✓	✓	✓	32.0	19.1	10.8	67.8	5.5	4.5

Table 3. **Ablation study of UWM across its components.** In gray is the original version of CLIP.

Model	**Basic Preference Metrics** (↑)				***SafeGround* Metrics** (↑)				
	$\texttt{P}_s^t$	$\texttt{P}_u^t$	$\texttt{P}_s^v$	$\texttt{P}_u^v$	$\texttt{Txt}_\texttt{S}$	$\texttt{Img}_\texttt{S}$	PS	PU	GS
ViT-L14 [51]	**73.1**	5.2	87.4	7.6	6.4	4.7	67.5	1.7	1.2
+*UWM*	71.2	**11.7**	**91.4**	**20.5**	**19.1**	**10.8**	**67.8**	**5.5**	**4.5**
ViT-B16 [51]	**68.7**	4.0	87.0	8.3	07.3	3.6	**63.5**	1.3	1.0
+*UWM*	67.1	**8.2**	**89.1**	**16.4**	**14.8**	**7.3**	62.7	**3.0**	**2.0**
ViT-B32 [51]	**67.4**	4.6	86.9	9.2	8.1	4.1	**62.2**	1.6	1.1
+*UWM*	64.0	**5.8**	**87.8**	**13.5**	**12.2**	**5.3**	59.7	**1.9**	**1.3**
CoCa [71]	**79.5**	4.5	93.6	8.7	8.0	4.1	**76.6**	1.7	1.3
+*UWM*	77.6	**5.6**	**94.9**	**15.5**	**15.4**	**5.2**	74.8	**2.6**	**2.2**
SigLIP [74]	**73.6**	3.5	**92.8**	7.9	7.2	3.1	**70.7**	1.5	1.0
+*UWM*	72.7	**6.7**	91.2	**11.1**	**10.2**	**6.2**	68.7	**2.4**	**1.8**
Safe-CLIP [49]	50.1	**19.0**	81.6	34.2	27.9	**18.1**	45.9	8.2	6.4
+*UWM*	**50.6**	18.5	**86.8**	**42.2**	**37.7**	17.3	**47.4**	**9.0**	**7.5**

Table 4. **Safety performance across architectures and pretraining strategies.** UWM reliably improves safety across models.

of Eq. (11). Relying solely on $\Phi^{\texttt{uns}}$ degrades zero-shot performance (-15.5), but already improves safety (+3.4 GS). Introducing the ratio $\Phi^{\texttt{uns}}/\Phi^{\texttt{sf}}$ further enhances safety (+11.8 GS) but severely impacts zero-shot performance (-23.6), making the model unsuitable for downstream tasks. The best trade-off is achieved with adaptive selection, improving safety (+4.5 GS) while minimizing zero-shot degradation (-7.8). Additional ablations are provided in the *Supp. Mat.*

Adaptability to Architectures. UWM is flexible and can be applied *off-the-shelf* to any contrastive-based VLM. Accordingly, we extend our evaluation to different CLIP backbones [51] and pretraining strategies, such as CoCa [71] and SigLIP [74]. Additionally, we evaluate its performance on Safe-CLIP [49]. We report the results in Tab. 4. UWM demonstrates consistent safety improvements across various CLIP backbones, enhancing safety on ViT-B16 (*e.g.*, +4.2 $\texttt{P}_u^t$, +12.9 $\texttt{P}_u^v$ and +12.7 $\texttt{Txt}_s$), while preserving its original behavior on safe queries (*e.g.*, -1.6 $\texttt{P}_s^t$). Similar improvements can be observed in ViT-B32 (*e.g.*, +4.3 $\texttt{P}_u^v$, +12.9 $\texttt{P}_u^v$ and +4.1 $\texttt{Txt}_s$). Moreover, UWM improves safety on models with different pre-training strategies, such as CoCa [71] (*e.g.*, +6.8 $\texttt{P}_u^v$, +7.4 $\texttt{Txt}_s$, and +0.9 GS) and SigLIP [74] (*e.g.*, +3.2 $\texttt{P}_u^t$, +3.1 $\texttt{Img}_s$, and +0.8 GS). Interestingly, our method further enhances Safe-CLIP's safety (*e.g.*, +5.2 $\texttt{P}_s^v$, +8 $\texttt{P}_u^v$, and +1.1 GS). These results highlight the flexibility and effectiveness of UWM.

LLaVA. We apply UWM to the vision encoder of LLaVA-1.5-13B [33]. Following existing works [49], we task LLaVA to caption unsafe images from the ViSU test set and report the percentage of Not Safe For Work (NSFW) generated content (measured with LLaMA-3-8B [61]) and toxicity score using the Perspective API [29]. Additionally, we assess knowledge preservation using Rouge-L [32], Bleu [46], and Meteor [3]. We report the results in Tab. 5. UWM effectively improves safety, reducing NSFW content by 9.8% and toxicity by 3.4, while preserving the original model's behavior (*e.g.*, 0.01 drop in Rouge-L). Moreover, after replacing LLaVA's vision encoder with Safe-CLIP [49], we apply UWM to Safe-CLIP within this setup. UWM further improves LLaVA safety by reducing NSFW generated content by 4.5% and toxicity by 1%, while preserving its knowledge (*e.g.*, equal Rouge-L and 0.01 drop in Bleu). These results further validates UWM's applicability.

Model	**Unsafety** (↓)		**Knowledge Preservation** (↑)		
	%NSFW	Toxicity	RougeL	Bleu	Meteor
LLaVA [33]	31.7	16.8	**0.32**	**0.13**	**0.26**
+*UWM*	**21.9**	**13.4**	0.31	0.11	0.23
+*Safe-CLIP*	8.0	10.0	**0.32**	**0.13**	**0.26**
+*UWM*	**3.5**	**9.0**	**0.32**	0.12	0.24

Table 5. We apply UWM to LLaVA[33] and LLaVA+Safe-CLIP.

6. Conclusion

This work investigates the safety challenges of Vision-Language Models (VLMs). We examine existing metrics [49] and find that they have limitations in assessing safety, as they rely on retrieval-based evaluations. Therefore, we complement them by introducing *SafeGround*, a novel suite of metrics specifically designed for safety evaluation. *SafeGround* exposes a critical issue: training-based safety alignment techniques, such as Safe-CLIP [49], can compromise safe representations, leading to unsafe behaviors on safe queries. We propose UWM, a training-free method that identifies and manipulates unsafe weights in VLMs by analyzing how the information flow varies between safe and unsafe content. UWM achieves a better trade-off between safety and knowledge preservation, marking a promising first step toward training-free safety techniques for VLMs. Finally, we hope *SafeGround* will serve as a valuable resource for the community, enabling researchers to uncover unsafe behaviors in VLMs.

Limitations. While UWM is applicable to various architectures and VLMs (Tab. 4 and Tab. 5), it relies on weight localization and manipulation. The complexity of large-scale VLMs challenges the isolation of unsafe weights without affecting model capabilities, as individual parameters may encode overlapping knowledge. As both UWM and Safe-CLIP [49] exhibit limitations, VLM safety remains an open challenge. Lastly, we focus mainly on contrastive-based VLMs, leaving other architectures for future research.

Acknowledgments. This work was supported by the EU Horizon project "ELIAS - European Lighthouse of AI for Sustainability" (No. 101120237), MUR PNRR project FAIR (PE00000013) funded by the NextGenerationEU, and the FIS project GUIDANCE (Debugging Computer Vision Models via Controlled Cross-modal Generation) (No. FIS2023-03251).

References

[1] Milad Alizadeh, Shyam A Tailor, Luisa M Zintgraf, Joost van Amersfoort, Sebastian Farquhar, Nicholas Donald Lane, and Yarin Gal. Prospect pruning: Finding trainable weights at initialization using meta-gradients. *arXiv preprint arXiv:2202.08132*, 2022. 2

[2] Dana Arad, Hadas Orgad, and Yonatan Belinkov. Refact: Updating text-to-image models by editing the text encoder. *arXiv preprint arXiv:2306.00738*, 2023. 2

[3] Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, 2005. 8

[4] Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*, 2021. 1

[5] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 2

[6] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 5

[7] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 2

[8] Trishna Chakraborty, Erfan Shayegani, Zikui Cai, Nael Abu-Ghazaleh, M. Salman Asif, Yue Dong, Amit K. Roy-Chowdhury, and Chengyu Song. Cross-modal safety alignment: Is textual unlearning all you need?, 2024. 1

[9] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 5

[10] Rocktim Jyoti Das, Mingjie Sun, Liqun Ma, and Zhiqiang Shen. Beyond size: How gradients shape pruning decisions in large language models. *arXiv preprint arXiv:2311.04902*, 2023. 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 5

[12] Moreno D'Incà, Elia Peruzzo, Massimiliano Mancini, Dejia Xu, Vidit Goel, Xingqian Xu, Zhangyang Wang, Humphrey Shi, and Nicu Sebe. Openbias: Open-set bias detection in text-to-image generative models. In *CVPR*, 2024. 2

[13] Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *NeurIPS*, 2017. 2

[14] Matteo Farina, Massimiliano Mancini, Elia Cunegatti, Gaowen Liu, Giovanni Iacca, and Elisa Ricci. Multiflow: Shifting towards task-agnostic vision-language pruning. In *CVPR*, 2024. 2, 4

[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPR-WS*, 2004. 5

[16] Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79, 2024. 2

[17] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing concepts from diffusion models. In *ICCV*, 2023. 1

[18] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *WACV*, 2024. 1, 2

[19] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. In *ICLR*, 2014. 1

[20] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 2

[21] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 5

[22] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2021. 5

[23] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021. 5

[24] Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Safe loRA: The silver lining of reducing safety risks when finetuning large language models. In *NeurIPS*, 2024. 1, 2

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV-WS*, 2013. 5

[26] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 5

[27] Jaeho Lee, Sejun Park, Sangwoo Mo, Sungsoo Ahn, and Jinwoo Shin. Layer-adaptive sparsity for the magnitude-based pruning. *arXiv preprint arXiv:2010.07611*, 2020. 2

[28] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018. 2

[29] Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A new generation of perspective api: Efficient multilingual character-level transformers. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. 8

[30] Mukai Li, Lei Li, Yuwei Yin, Masood Ahmed, Zhenguang Liu, and Qi Liu. Red teaming visual language models. *arXiv preprint arXiv:2401.12915*, 2024. 2

[31] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE T-PAMI*, 40(12):2935–2947, 2017. 1

[32] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, 2004. 8

[33] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 5, 7, 8

[34] Runtao Liu, Ashkan Khakzar, Jindong Gu, Qifeng Chen, Philip Torr, and Fabio Pizzati. Latent guard: a safety framework for text-to-image generation. In *ECCV*, 2025. 2

[35] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *ICLR*, 2024. 2

[36] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. In *ICML*, 2023. 2

[37] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *NeurIPS*, 2023. 5

[38] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 5

[39] Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. In *NeurIPS*, 2024. 2

[40] Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *NeurIPS*, 2022. 2

[41] Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*, 2022. 2

[42] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *ICLR*, 2022. 2

[43] Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *ICML*, 2022. 2

[44] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, 2008. 5

[45] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models. In *ICCV*, 2023. 2

[46] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002. 8

[47] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 5

[48] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022. 2

[49] Samuele Poppi, Tobia Poppi, Federico Cocchi, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Safe-CLIP: Removing NSFW Concepts from Vision-and-Language Models. In *ECCV*, 2024. 1, 2, 3, 4, 5, 6, 7, 8

[50] Vinay Uday Prabhu and Abeba Birhane. Large datasets: A pyrrhic win for computer vision. In *Institute of Electrical and Electronics Engineers/Computer Vision Foundation Conference on Applications of Computer Vision*, 2021. 1, 3

[51] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 3, 5, 6, 7, 8

[52] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[53] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, 2019. 5

[54] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. 2022 ieee. In *CVPR*, 2021. 1, 2

[55] Patrick Schramowski, Christopher Tauchmann, and Kristian Kersting. Can machines help us answering question 16 in datasheets, and in turn reflecting on inappropriate content? In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 1350–1361, 2022. 1, 3

[56] Patrick Schramowski, Manuel Brack, Björn Deiseroth, and Kristian Kersting. Safe latent diffusion: Mitigating inappropriate degeneration in diffusion models. In *CVPR*, 2023. 1, 2

[57] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024. 2

[58] Dachuan Shi, Chaofan Tao, Ying Jin, Zhendong Yang, Chun Yuan, and Jiaqi Wang. Upop: Unified and progressive pruning for compressing vision-language transformers. In *ICML*, 2023. 2

[59] K Soomro. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 5

[60] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023. 2

[61] Llama Team. The llama 3 herd of models, 2024. 8

[62] Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *CVPR*, 2022. 3

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste

Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2

[64] Yu-Lin Tsai, Chia-Yi Hsu, Chulin Xie, Chih-Hsun Lin, Jia-You Chen, Bo Li, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. Ring-a-bell! how reliable are concept removal methods for diffusion models? In *ICLR*, 2024. 2

[65] Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020. 2

[66] Haohan Wang, Songwei Ge, Zachary Lipton, and Eric P Xing. Learning robust global representations by penalizing local predictive power. *Advances in neural information processing systems*, 2019. 5

[67] Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 2023. 2

[68] Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *NeurIPS*, 2024. 2

[69] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 5

[70] Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. BESA: Pruning large language models with blockwise parameter-efficient sparsity allocation. In *ICLR*, 2024. 5

[71] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 8

[72] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Jiahao Xu, Tian Liang, Pinjia He, and Zhaopeng Tu. Refuse whenever you feel unsafe: Improving safety in llms via decoupled refusal training. *arXiv preprint arXiv:2407.09121*, 2024. 1

[73] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *ICLR*, 2024. 2

[74] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 8

[75] Yimeng Zhang, Jinghan Jia, Xin Chen, Aochuan Chen, Yihua Zhang, Jiancheng Liu, Ke Ding, and Sijia Liu. To generate or not? safety-driven unlearned diffusion models are still easy to generate unsafe images... for now. In *ECCV*, 2024. 1, 2

[76] 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. 2