One Noise to Fool Them All: Universal Adversarial Defenses Against Image Editing

Shorya Singhal* Parth Badgujar* Devansh Bhardwaj*
Data Science Group, IIT Roorkee
shorya_s@mfs.iitr.ac.in, {parth_db, devansh_b}@ece.iitr.ac.in

Abstract

Recent advances in image-to-image editing models offer both benefits and risks. While they enhance creativity, accessibility, and applications in fields ranging from medicine to environmental science, they can also enable misuse, such as identity manipulation, copyright infringement, and deepfake creation. The dual nature of these technologies necessitates sophisticated defensive strategies that can proactively mitigate emerging risks. Image immunization techniques have emerged as a promising solution for addressing these challenges, employing adversarial perturbation strategies to disrupt potential malicious model capabilities. This research advances the state of the art by introducing novel methods to universalize immunization approaches, extending protective mechanisms beyond single-image scenarios to create more comprehensive and robust defense strategies against unauthorized image transformations and achieve true immunization.

1. Introduction

Generative AI technologies, particularly diffusion-based image generation systems, have revolutionized digital content creation by demonstrating unprecedented capabilities in image manipulation and synthesis. These technologies present a complex technological landscape characterized by extraordinary creative potential and significant ethical challenges, including risks of identity manipulation, unauthorized content generation, and sophisticated deepfake creation.

As AI image editing technologies become increasingly sophisticated, the imperative for robust defensive mechanisms has never been more critical. Image immunization [13] techniques have emerged as a promising strategic approach to mitigate these risks, leveraging adversarial perturbation strategies designed to disrupt AI model capabilities. Figure 1 shows how immunization is achieved by introducing imperceptible noise into images, rendering them resilient against unintended or malicious transformations by disrupting the process of editing using generative models while preserving their fundamental visual integrity.

The existing landscape of immunization strategies can be comprehensively categorized into encoder-based and decoder-based approaches, depending on what part of the model, the noise disrupts. Photoguard[13] introduced dual strategies, including an encoder attacks that manipulate latent representations and diffusion attacks targeting entire image generation pipelines. Complementary research, like Posterior Collapse Attacks (PCA) [4], explored critical vulnerabilities in variational autoencoders by strategically manipulating latent distributions to trigger posterior collapse. Decoder Attacks such as AdvDM and Score Distillation Sampling (SDS) [9, 15] have further refined gradient computation for a semantic loss, enabling more efficient noise generation for decoder attacks. We will focus on encoder-based attack methodologies in this paper.

Despite these significant advancements, existing approaches predominantly focus on single-image scenarios, representing a substantial limitation in practical immunization strategies. Under limited computational budgets, single-image attacks may not fully explore every model vulnerability. Multi-image methods, on the other hand, accumulate gradients from a variety of images, creating a more robust search direction and preventing overfitting to a single sample. This research introduces Multi and UAP Attack, novel universal adversarial perturbation methods which find a universal noise for any image to explicitly address key constraints in contemporary image immunization techniques.

Our primary contributions are twofold, proposing a framework to develop and evaluate multi-image immunization methods and proposing two novel attacks for generating universal noise patterns with demonstrated effectiveness across multiple images. We present a comprehensive solution to the complex challenges posed by AI-powered image manipulation technologies.

*Equal contribution

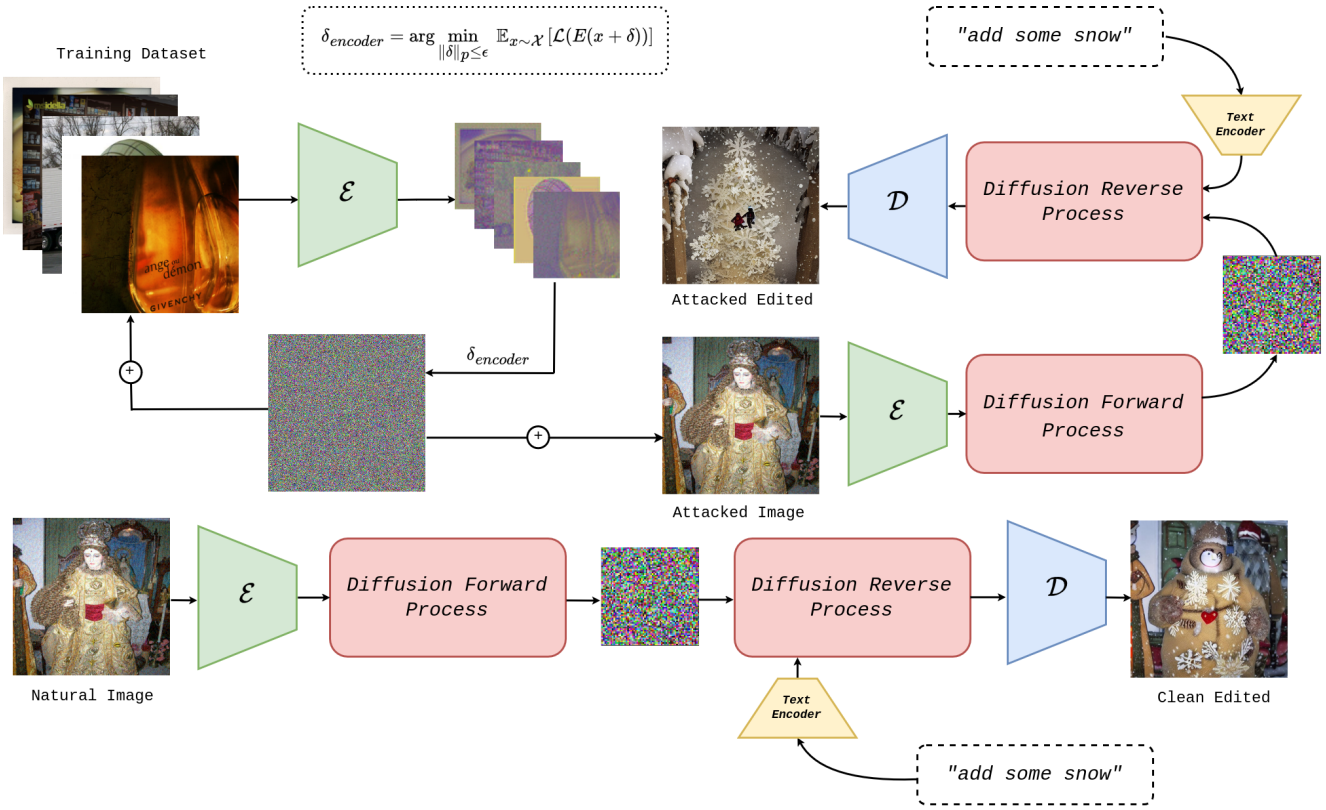

Figure 1. The figure shows how immunization is achieved across multiple images. We use an adversarial attack to iteratively find a universal perturbation $\delta_{encoder}$, which is then used to produce a corrupt latent representation for an unseen image. The below pipeline shows the corresponding output for the original unseen image for comparison.

2. Background

2.1. Diffusion Models

Diffusion models are a class of probabilistic generative models that approximate a data distribution $p(x)$ by progressively denoising a normally distributed variable. These models define a Markov chain of diffusion steps to slowly add random noise to a sample from the empirical data distribution $q(x)$. The objective is to learn the reverse process that reconstructs desired samples from the target distribution $p(x)$ given a noisy input.

2.1.1. Forward Diffusion Process

Given a data point x_0 sampled from the data distribution, the forward diffusion process introduces Gaussian noise over T steps, generating progressively noisier samples $x_1, x_2, \ldots, x_T$. The transition from one step to the next follows a Gaussian distribution:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \mathbf{I})$$

where the step sizes β_t determine the amount of noise added at each step, controlled by a variance schedule.

2.1.2. Reverse Diffusion Process

Given the final noisy sample x_T, the reverse diffusion process aims to recover the original data by learning to sample from $q(x_{t-1}|x_t, x_0)$. We learn to model $p_\theta(x_{t-1}|x_t)$ to approximate this conditional distribution using a normal distribution:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \boldsymbol{\mu}_\theta(x_t, t), \boldsymbol{\Sigma}_\theta(x_t, t))$$

Ho et al. [6] discovered that learning the variance $\boldsymbol{\Sigma}_\theta(x_t, t)$ leads to instability during training and degraded sample quality. Instead, they proposed fixing the variance and training a network ϵ_θ to predict the noise ϵ_t added at each step. . Consequently, this denoising model ϵ_θ is trained to minimize the following loss function:

$$L_t^{simple} = \mathbb{E}_{t \sim [1,T], x_0, \epsilon_t} \left[\|\epsilon - \epsilon_\theta(x_{t+1}, t)\|^2 \right]$$

2.1.3. Latent Diffusion Models

Our focus will be on a specific class of diffusion models called the latent diffusion models (LDMs) [12]. These

models improve computational efficiency and image quality at high resolutions by applying the diffusion process in a lower-dimensional latent space instead of the original pixel space. This approach retains the semantic and structural information of the image while reducing the number of necessary computations, since most bits only contain information about perceptual details.

LDM uses an encoder network $\mathcal{E}$ to map the original input image x_0 to its latent representation $z_0 = \mathcal{E}(x_0)$. The resulting noisy latent $\tilde{z}$ is then passed through a decoder network $\mathcal{D}$ to reconstruct the final output image $\tilde{x}$.

2.2. Variational Autoencoder

Variational Autoencoders (VAEs) [8] are a class of latent variable models that employ a probabilistic encoder-decoder framework. The encoder transforms input data x into a latent representation z that follows a prior distribution $p_\theta(z)$, typically modeled as a Gaussian:

$$q_\phi(z \mid x) \sim \mathcal{N}(\mu, \text{diag}(\sigma^2))$$

Training is performed by minimizing the Kullback-Leibler (KL) divergence between the approximate posterior $q_\phi(z \mid x)$ and the true posterior $p_\theta(z \mid x)$. This is achieved using the following loss function:

$$\mathcal{L}(x) = -\mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] + D_{\text{KL}}\big(q_\phi(z \mid x) \parallel p(z)\big)$$

This objective consists of two terms: a reconstruction loss that encourages the decoder to accurately reconstruct x from z, and a regularization term that ensures the latent distribution remains close to the prior.

2.3. Adversarial attacks

In computer vision, adversarial examples are small, often imperceptible modifications to an image that cause a model to misclassify it with high confidence [3]. A targeted adversarial attack aims to find a perturbation δ_{adv} for a given datapoint x that causes the model f_θ to produce a specific incorrect target y_{targ}. This noise is found by minimizing the following optimization problem:

$$\delta_{adv} = \arg\min_{\delta \in \Delta} L(f_\theta(x + \delta), y_{targ})$$

where Δ is the set of allowable perturbations constrained by an L_p norm to ensure visual imperceptibility, $\Delta = \{\delta : \|\delta\|_p \leq \epsilon\}$. One common approach to solve this optimization problem is to use projected gradient descent (PGD) [11].

2.4. Photoguard

Photoguard [13] introduces two distinct methods to immunize images against adversarial manipulation, encoder and diffusion attack. We will mainly focus on their encoder attack in this paper.

2.4.1. Encoder Attack

The Encoder Attack focuses on perturbing the latent representation of an image within a Variational Autoencoder (VAE). By introducing carefully crafted noise into the input image, the attack maps the image to an altered latent space representation that disrupts downstream tasks, resulting in a perceptually distorted output upon image editing.

The researchers employed a straightforward optimization strategy to align the latent space embedding with the target latent representation of a grey image using Projected Gradient Descent. Let $\delta_{encoder}$ denote the perturbation added to the original image, it can be formulated as:

$$\delta_{\text{encoder}} = \arg\min_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}_{pg} = \|\mathcal{E}(\mathbf{x} + \delta) - \mathbf{z}_{\text{targ}}\|_2^2$$

2.5. Posterior Collapse Attack

Due to the dominance of the KL divergence term in the learning objective, VAE training usually suffers from an optimization issue called posterior collapse [14], a phenomenon in which latent variables fail to encode meaningful information, causing the model to ignore them. This leads to the output of decoder $\tilde{x}$ becoming almost independent of z, essentially leading to the collapse of the posterior $q_\phi(z\|x)$ to the prior $p_\theta(z)$.

2.5.1. Attack

Instead of optimizing the latent space towards a grey image as done by Photoguard, PCA [4] tried to leverage this phenomenon and generate a specific noise which will forcefully collapse the resulting latent distribution, thus, distorting the outputs from the decoder.

This paper proposed two methods two attack the image both of which essentially use the same loss function, formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{pca}}(x) &= D_{\text{KL}}\left(q(z|x) \parallel p^*(z)\right) \\ &= \frac{1}{2}\sum_{i=1}^{d}\left(\frac{\sigma_i^2 + \mu_i^2}{v} - 1 + \ln\frac{v}{\sigma_i^2}\right)\end{aligned}$$

where $p^*(z) \sim \mathcal{N}(0, v)$, and v controls the disruptive effect of the attack. The authors proposed two different attack strategies:

1. **PCA+ Attack:** Maximizes $\mathcal{L}_{\text{pca}}(x + \delta)$, pushing the latent distribution away from the prior $\mathcal{N}(0, v)$, leading to an out-of-distribution scenario that disrupts latent representations and downstream tasks. This optimization forces both mean μ_i^2 and variance σ_i^2 towards extreme values, breaking the learned encoding.
2. **PCA- Attack:** Minimizes $\mathcal{L}_{\text{pca}}(x + \delta)$, bringing the latent distribution closer to a near-zero Gaussian $\mathcal{N}(0, 0)$. This collapses the latent representation to a Dirac distribution by driving both σ_i^2 and μ_i^2 to zero, reducing the

information encoded in z and distorting the generative process.

3. Methodology

The objective of this work is to generate a single universal perturbation vector $\delta \in \mathbb{R}^d$ that, when added to any natural image $x \in \mathcal{X} \subset \mathbb{R}^d$, results in a corrupted latent representations by a pre-trained encoder $E : \mathbb{R}^d \to \mathbb{R}^k$. The perturbation δ must lie within a bounded ℓ_p norm-ball $\|\delta\|_p \leq \epsilon$, ensuring imperceptibility while minimizing a latent-space loss function $\mathcal{L}$ designed to degrade encoder outputs to give incorrect representations of the input image. Figure 1 shows how immunization is achieved for an multi-image setting. We define the universal optimization objective as follows:

$$\delta^* = \arg\min_{\|\delta\|_p \leq \epsilon} \mathbb{E}_{x \sim \mathcal{X}} \left[\mathcal{L}(E(x + \delta)) \right], \tag{1}$$

where $\mathcal{L}$ may be instantiated as either a latent deviation (Photoguard Loss) or a KL-divergence loss (PCA Loss). In this section, we present two algorithmic strategies to approximate δ^*: (i) a greedy per-sample update approach (UAP), and (ii) a batched optimization method (Extended Multi-Image Attack).

Algorithm 1 Universal Adversarial Perturbation (UAP)

Input: Dataset $\{x_i\}_{i=1}^N$, encoder $\mathcal{E}$, inner adversarial attack function $A(\cdot)$, loss function $\mathcal{L}$ (e.g., $\mathcal{L}_{pg}$ or $\mathcal{L}_{pca}$), maximum iterations $T_{\max}$, perturbation bound ϵ, learning rate η, attack function $A(\cdot)$ (e.g., Adam, PGD, AutoPGD)
Initialize universal perturbation: $\delta \leftarrow \mathbf{0}$
for $t = 1$ to $T_{\max}$ **do**
    Randomly shuffle the dataset indices
    for each image x_i in the shuffled dataset **do**
        Compute latent loss: $\mathcal{L}(\mathcal{E}(x_i + \delta))$
        Compute image-specific perturbation:

$$\delta_i \leftarrow A\big(\delta, \nabla_\delta L(\mathcal{E}(x_i + \delta)), \eta, \epsilon\big)$$

        if $\sum_{j=1}^{j=N} \mathcal{L}(x_j + \delta_i) < \sum_{j=1}^{j=N} \mathcal{L}(x_j + \delta)$ **then**
            Update $\delta \leftarrow \delta_i$
        end if
    end for
end for
return δ

3.1. Universal Adversarial Perturbation (UAP)

The UAP-based [5] approach is a sequential method inspired by classical universal adversarial perturbation techniques. Let $\mathcal{X} = \{x_i\}_{i=1}^N$ denote the dataset. We define the universal perturbation δ and aim to iteratively refine it by visiting each image x_i in a randomized order, computing its latent loss $\mathcal{L}(E(x_i + \delta))$ and performing a gradient-based adversarial attack, which proposes a new perturbation candidate δ_i.

Let $\mathcal{A}(\delta, \nabla_\delta \mathcal{L}, \eta, \epsilon)$ denote a gradient-based attack function such as PGD or AutoPGD [1] which returns an updated perturbation vector. At each iteration, the algorithm checks whether the updated perturbation improves the cumulative loss across the dataset:

$$\sum_{j=1}^{N} \mathcal{L}(E(x_j + \delta_i)) < \sum_{j=1}^{N} \mathcal{L}(E(x_j + \delta)). \tag{2}$$

If this condition holds, δ is updated to δ_i. This process is repeated for a maximum of $T_{\max}$ iterations, producing a universal perturbation that generalizes well across samples by greedily optimizing per-image performance using a sample-specific perturbation update.

The strength of this approach lies in its sample-wise updates, which provide targeted latent-space degradation and ensure that each image in the dataset contributes to the robustness of the final universal perturbation.

Algorithm 2 Extended Multi-Image Adversarial Attack (Generalized)

Input: Batch of images $\{x_i\}_{i=1}^N$, encoder $\mathcal{E}$, perturbation bound ϵ, learning rate η, maximum iterations T, attack function $A(\cdot)$, loss function $L(\cdot)$
Initialize universal perturbation δ as a trainable parameter sampled uniformly from $[-\epsilon, \epsilon]$
for $t = 1$ **to** T **do**
    Perturb the images: $x_i^{\text{adv}} = x_i + \delta$ for all i
    Compute latent representations: $\mu_i = \mu\big(\mathcal{E}(x_i^{\text{adv}})\big)$
    Calculate the average loss:

$$L(\delta) = \frac{1}{N} \sum_{i=1}^{N} L\big(\mu_i\big)$$

    Update δ using the chosen attack method:

$$\delta \leftarrow A\big(\delta, \nabla_\delta L(\delta), \eta, \epsilon\big)$$

end for
return δ

3.2. Extended Multi-Image Adversarial Attack

While the UAP strategy incrementally updates δ per sample, the Extended Multi-Image Adversarial Attack treats δ as a trainable parameter, enabling more efficient optimization via batch gradient descent. This method computes the latent representations of a full batch of perturbed images and directly minimizes the average latent loss across the

batch.

Let $\{x_i\}_{i=1}^N$ denote a mini-batch, and let $x_i^{\text{adv}} = x_i + \delta$ denote the perturbed inputs. The latent embeddings are computed as $\mu_i = \mathcal{E}(x_i^{\text{adv}})$. The optimization objective is as follows:

$$\mathcal{L}(\delta) = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mu_i), \tag{3}$$

The perturbation δ is initialized uniformly from $[-\epsilon, \epsilon]$ and is updated using an optimizer $\mathcal{A}$. This enables flexible, differentiable learning of a universal perturbation that generalizes well across input distributions.

This method offers improved scalability, faster convergence, and the ability to incorporate richer loss formulations due to the batched and differentiable nature of its optimization routine.

For both the above attacks, we explore various combinations of loss functions $\mathcal{L}$ and adversarial attacks $\mathcal{A}$.

The loss $\mathcal{L}$ can be instantiated as:

- $\mathcal{L}_{\text{pg}}(\mu_i) = \|\mu_i - \mu_{\text{ref}}\|_2^2$, targeting deviation from a reference latent embedding,
- $\mathcal{L}_{\text{pca}} = \sum_{i=1}^{d} \left(\frac{\sigma_i^2 + \mu_i^2}{v} \right)$, as the KL-divergence between posterior and a collapsed prior.

Additionally, for adversarial attacks $\mathcal{A}$, we explore Projected Gradient Descent (PGD) [11], AutoPGD [1] and an Adam-based attack which leverages the gradient's magnitude instead of its sign along with the Adam optimizer [7].

4. Experimental Setup

We extend the experimental setup in PCA [4] to evaluate both the multi and single image attacks.

Dataset: We utilize a 1000-image subset of the ImageNet [2], curated following the protocol established by Lin et al. [10], which is widely adopted in adversarial attack research. All images are resized to a resolution of 512×512 to maintain consistency across editing pipelines. This dataset is divided into a train:test split of 128:872. The 128 images are used to compute the universal perturbation δ for our methods, and the resulting noise is then evaluated on the remaining unseen test set images. For baseline single image attacks, adversarial perturbations are calculated independently for each test image.

Models: We primarily evaluate our attack against Stable Diffusion v1.5 (SD15), due to its open-source availability and its widespread use in real-world image generation and editing applications.

Prompts: To assess the robustness of our perturbation under a variety of editing conditions, we evaluate on a diverse set of natural language prompts commonly used in LDM-based editing tasks: (P1) "Make it like a watercolor painting" (style transfer), (P2) "Apply sunset lighting" (lighting adjustment), (P3) "Add some snow" (weather modification), and (P4) an empty prompt (null-edit). This prompt variety allows us to evaluate performance across a range of semantically meaningful transformations.

Metrics: To quantitatively evaluate the effect of our attack, we employ four widely used Image Quality Assessment (IQA) metrics: Peak Signal-to-Noise Ratio (PSNR), which measures pixel-level distortion; Feature Similarity Index (FSIM), which captures low-level structural similarities using phase congruency and gradient features; Structural Similarity Index Measure (SSIM), which assesses perceptual distortion by comparing luminance, contrast, and structure; and Visual Information Fidelity (VIFp), a perceptual image quality metric that measures the loss of human-aligned visual information between a reference and distorted image. For all four metrics, higher values indicate better image similarity to the reference.

Methods: We compare our method against two recent state-of-the-art image protection techniques: PhotoGuard (PG) [13] and Posterior Collapse Attack (PCA) [4]. For both PG and PCA, we standardize key hyperparameters, including number of attack steps ($T = 40$), perturbation bound ($\epsilon = 16$), and optimization routines. We implemented both PGD and Adam as attack functions $\mathcal{A}$, but table 1 shows only the Adam variant in the first 3 columns. This ensures uniform comparison of these single-image attacks with our multi-image attacks. For our methods, we adjust the number of attack steps due to the difference in training and inference processes. Since our method does not require adversarial perturbation generation during test time, unlike the baseline single image attacks, we fix the number of attack steps to $T = 1000$ for the Extended Multi-Image Attack across the 128 image training set and for UAP, we set the number of outer-loop iterations to 5, and the number of inner-loop Attacking steps per image to 8. For extended multi image, we used Adam as the optimizer and 3 variants of the optimization objective function, namely PCA+/PCA-/Photoguard. For UAP, we utilized 2 attacks, AutoPGD with the Photoguard objective and PGD with PCA- objective.

5. Results

Our evaluation compares the Universal Adversarial Perturbation (UAP) and Extended Multi-Image Adversarial Attack approaches using four editing prompts—(P1) "Make it like a watercolor painting" (style transfer), (P2) "Apply sunset lighting" (lighting adjustment), (P3) "Add some snow" (weather modification), and (P4) an empty prompt (null-edit). For each prompt, we measure image quality using SSIM, PSNR, FSIM, and VIFp. Overall, lower SSIM, PSNR, FSIM, and VIFp values indicate stronger degrada-

Prompt	Metric	PCA- adam	PCA+ adam	PG adam	Multi PCA+	Multi PCA-	Multi PG	UAP PCA-	UAP autopgd
P1	SSIM↑	0.6824	0.7027	0.6800	0.5758	**0.5524**	0.5986	0.5690	0.5755
P1	PSNR↑	21.32	22.62	21.25	20.64	**17.72**	19.56	19.76	18.56
P1	VIFp↑	0.1845	0.2098	0.1825	0.1343	**0.1171**	0.1273	0.1276	0.1257
P1	FSIM↑	0.8075	0.8353	0.8059	0.7859	**0.7353**	0.7700	0.7634	0.7496
P2	SSIM↑	0.4087	0.4092	0.4085	0.3560	0.3477	0.3603	**0.3220**	0.3382
P2	PSNR↑	17.30	17.58	17.28	17.11	**16.18**	16.67	16.59	16.34
P2	VIFp↑	0.0532	0.0602	0.0530	0.0444	**0.0310**	0.0365	0.0370	0.0333
P2	FSIM↑	0.6864	0.7061	0.6863	0.6833	**0.6614**	0.6764	0.6761	0.6669
P3	SSIM↑	0.3042	0.3250	0.3030	0.3003	**0.2289**	0.2421	0.2361	0.2351
P3	PSNR↑	15.85	16.37	15.83	16.17	**14.57**	15.00	15.16	14.76
P3	VIFp↑	0.0371	0.0443	0.0368	0.0367	**0.0217**	0.0260	0.0277	0.0241
P3	FSIM↑	0.6467	0.6692	0.6463	0.6609	**0.6122**	0.6229	0.6286	0.6150
P4	SSIM↑	0.3199	0.3500	0.3196	0.2977	**0.2132**	0.2504	0.2352	0.2212
P4	PSNR↑	16.82	17.35	16.80	16.88	**15.13**	15.94	15.90	15.38
P4	VIFp↑	0.0395	0.0497	0.0391	0.0380	**0.0192**	0.0269	0.0286	0.0225
P4	FSIM↑	0.6777	0.7016	0.6773	0.6780	**0.6253**	0.6496	0.6485	0.6297

Table 1. Comparison of single and multi-image methods against several baselines on the test set. Arrows (↑/↓) indicate whether higher or lower values represent better image quality for each image quality metric. The best result for each prompt and each metric has been highlighted in bold in each row and second best has been underlined. Here, PG denotes Photoguard, PCA+/PCA-/PG denote the base optimization problem, pgd/autopgd/adam denote the method for solving the optimization problem and for UAP autopgd, $\mathcal{L}_{pg}$ has been used. The results have been evaluated on 4x NVIDIA Tesla V100 16GB GPUs.

tion of the intended edits.

In our experiments, the multi-image methods consistently achieve lower metric scores compared to single-image (per-sample) approaches. For example, under the null edit condition (P4), the multi-image variants reduce SSIM and PSNR more dramatically than the single-image attacks, indicating that the latent representations are significantly more disrupted. Similar trends are observed for the weather, lighting, and style prompts. For most of the cases the Multi PCA- attack performs the best and UAP autopgd attack performs the second best in almost all of the metrics.

Across all four subplots in Figure 2, we observe a consistent trend: as the number of samples increases, the performance of both attack variants stabilizes, with Multi Image PCA- achieving consistently lower metric scores, indicating stronger image disruption and better immunization. Notably, even small training sets (e.g., 16–32 images) are sufficient to significantly degrade model outputs, showcasing the efficiency of universal attack formulations. The performance gains saturate after around 64–128 training images, suggesting that a relatively small subset of data is sufficient to capture shared vulnerabilities in the encoder's latent space. Figure 3 shows a qualitative example that demonstrates the effectiveness of our multi-image attacks.

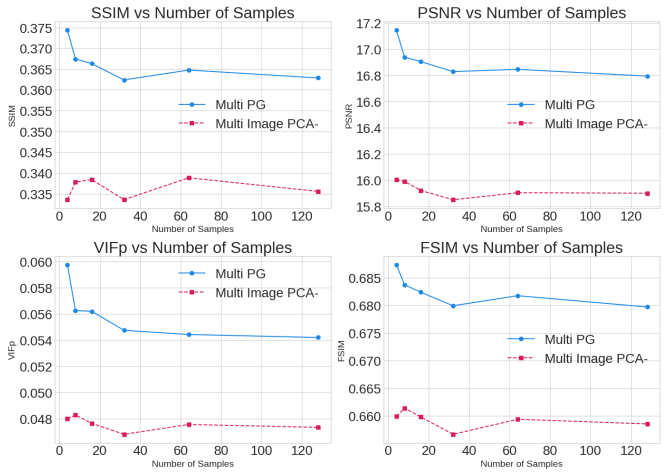

Figure 2. Effect of Training Set Size on Multi-Image Attack Effectiveness. This figure illustrates how increasing the number of training images used to compute a universal perturbation affects the quality degradation metrics (SSIM, PSNR, VIFp, FSIM) on unseen test images for two multi-image attack strategies: Multi PG and Multi Image PCA-.

6. Reconciling Single-Image vs. Multi-Image Attack Performance

Intuitive Expectation: From a purely conceptual standpoint, one might anticipate that a single-image attack would

yield the strongest distortion for that one image. By focusing optimization solely on a particular sample, the perturbation can "overfit" to fine nuanced latent-space features for a specific image, thereby achieving maximum local disruption.

- Local Optimality: Single-image attacks can converge to potent local minima specialized for that image, capturing its unique latents and idiosyncrasies.
- Precision Tuning: The noise can, in principle, be fine-tuned to exactly degrade the final edited output without "wasting" capacity on other images.

Why Multi-Image Attacks can Outperform Single-Image Attacks: Despite this intuition, empirical evidence in table 1 consistently shows that training on multiple images usually finds a single perturbation that also degrades each individual image more severely than a per-image approach and works for unseen images as well. Several factors explain this phenomenon:

1. **Deeper Latent-Space Vulnerabilities:** By seeing multiple images, the optimizer is exposed to gradients that reflect common weaknesses in the encoder or diffusion model. Rather than exploiting idiosyncratic properties of a single latent embedding, the multi-image approach zeros in on fundamental representational shortcuts or failure modes shared across different images that exploit model vulnerability better. Ironically, these universal directions often also harm any particular image in train set more severely than a local, single-image optimum.
2. **Escaping Local Minima:** A single-image attack may converge to a local optimum that strongly disrupts editing for that image, but not necessarily the worst possible outcome universal outcome for the model. When multiple images are involved, the combined objective forces the attack out of narrow minima that only apply to one specific latent configuration, steering it instead toward global minima in the latent space that degrade editing in a broader, more devastating way.

In short, while a single-image attack may indeed specialize to that specific image, multi-image attacks tap into the more generalizable, systemic weaknesses of the diffusion model, thereby yielding perturbations that incidentally disrupt individual images even more severely.

7. Conclusion

In this work, we present a novel universal adversarial perturbation framework for robust image immunization against diffusion-based image editing models. By formulating multi-image encoder perturbations through both greedy (UAP) and batched (Extended Multi-Image) optimization strategies, we demonstrate that a single, imperceptible noise pattern can generalize across diverse images and editing prompts, effectively degrading the semantic editing capabilities of state-of-the-art generative models.

Our results reveal that multi-image attacks not only scale efficiently, but often surpass per-image approaches in disrupting the generative process, even on unseen test images. This suggests that shared vulnerabilities in latent representations can be exploited more effectively through collective optimization, rather than isolated per-sample tuning. Among all tested configurations, the Multi PCA- attack consistently exhibited the strongest degradation across both quantitative metrics and qualitative results, highlighting the promise of targeting model vulnerabilities such as posterior collapse directly.

Ultimately, UAP and Extended Multi-Image attacks marks a step toward scalable, content-preserving defenses against unauthorized generative manipulations, and lays the groundwork for future research in adaptive immunization, cross-architecture generalization, and alignment with downstream safety protocols.

References

[1] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, 2020. 4, 5

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5

[3] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. 3

[4] Zhongliang Guo, Lei Fang, Jingyu Lin, Yifei Qian, Shuai Zhao, Zeyu Wang, Junhao Dong, Cunjian Chen, Ognjen Arandjelović, and Chun Pong Lau. A grey-box attack against latent diffusion model-based image editing by posterior collapse, 2024. 1, 3, 5

[5] Hokuto Hirano and Kazuhiro Takemoto. Simple iterative method for generating targeted universal adversarial perturbations. *Algorithms*, 13(11):268, 2020. 4

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. 2

[7] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. 5

[8] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022. 3

[9] Chumeng Liang, Xiaoyu Wu, Yang Hua, Jiaru Zhang, Yiming Xue, Tao Song, Zhengui Xue, Ruhui Ma, and Haibing Guan. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples, 2023. 1

[10] Dacheng Lin, Jay Strader, Aaron J. Romanowsky, Jimmy A. Irwin, Olivier Godet, Didier Barret, Natalie A. Webb, Jeroen Homan, and Ronald A. Remillard. Multiwavelength follow-up of the hyperluminous intermediate-mass black hole candidate 3xmm j215022.4055108. *The Astrophysical Journal Letters*, 892(2):L25, 2020. 5

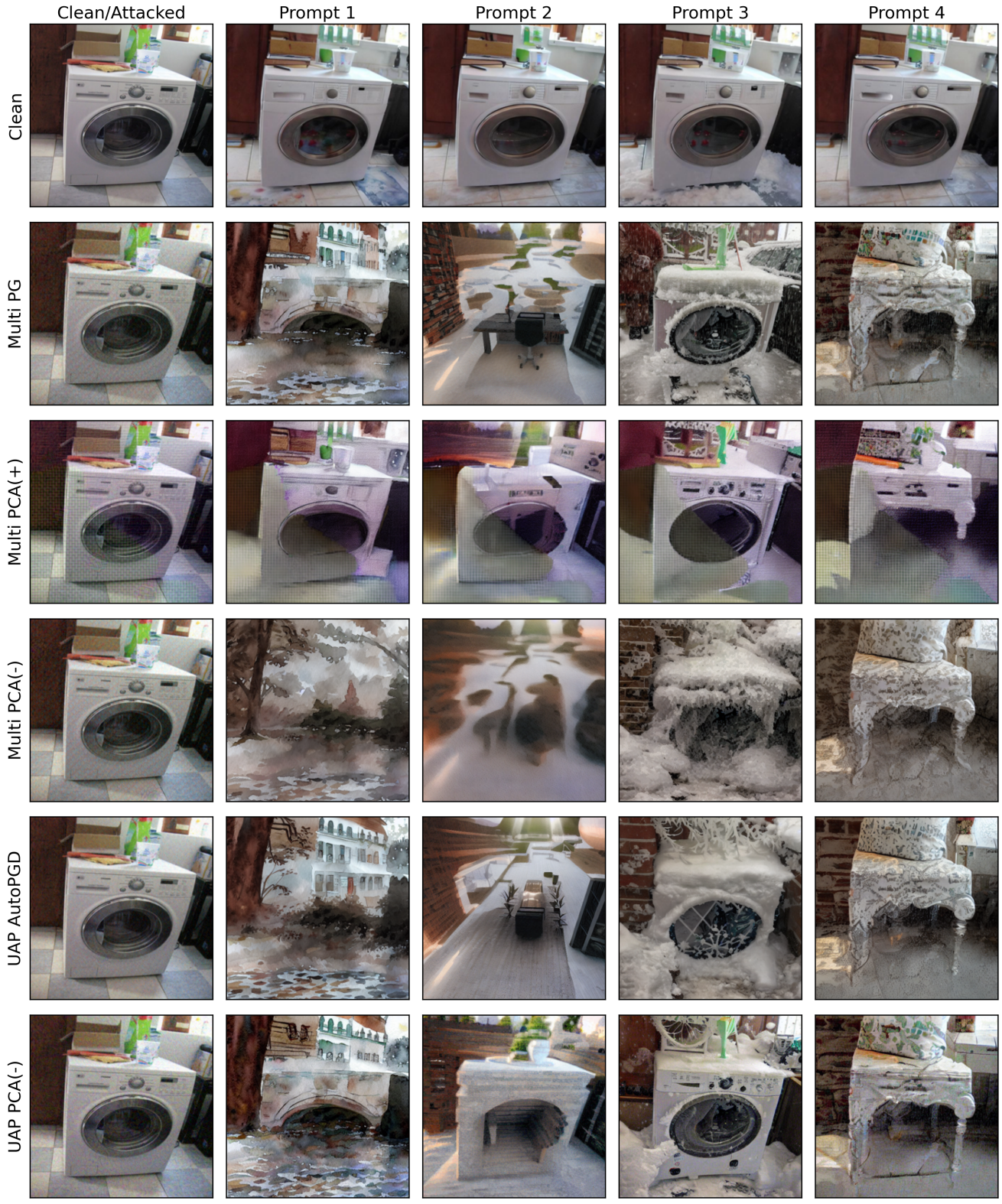

Figure 3. Qualitative comparison of a multi-image attack on a test image using a universal perturbation generated from a training set of 128 images and $\epsilon = 0.06$. The original image (top left) is shown alongside its edited counterparts using Stable Diffusion v1.5, revealing that the perturbation significantly disrupts the model's ability to apply intended semantic edits.

[11] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. 3, 5

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022. 2

[13] Hadi Salman, Alaa Khaddaj, Guillaume Leclerc, Andrew Ilyas, and Aleksander Madry. Raising the cost of malicious ai-powered image editing, 2023. 1, 3, 5

[14] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018. 3

[15] Haotian Xue, Chumeng Liang, Xiaoyu Wu, and Yongxin Chen. Toward effective protection against diffusion based mimicry through score distillation, 2024. 1