**S.P.E.C.I.O.U.S. (Spectral Perturbation Engine for Contrastive Inference Over Universal Surrogates)**

## **Abstract**

**Generative AI models trained on large uncurated image corpora often appropriate artists’ copyrighted work without consent. We introduce S.P.E.C.I.O.U.S. (Spectral Perturbation Engine for Contrastive Inference Over Universal Surrogates), a universal, multi-model defensive technique that embeds imperceptible high-frequency perturbations into the luminance (Y) channel of YCbCr representations. At inference time, these perturbations remain invisible to humans but systematically degrade feature embeddings across multiple surrogate models (ResNet-50, CLIP ViT-B/32, etc.), preventing downstream generative or classification tasks from reproducing the protected style. Our U-Net-based generator works in the Fourier domain—with a learnable high-pass mask—while a novel SpeciousLoss simultaneously minimizes LPIPS perceptual distance and maximizes surrogate feature distortion under a strict perceptual threshold. We demonstrate that S.P.E.C.I.O.U.S. effectively defends against zero-shot and conditional generative attacks, generalizes across resolutions, and preserves visual fidelity.**

## **1. Introduction**

### **1.1 Background and Motivation**

**The recent “Ghiblification” phenomenon—in which users transform images into a Studio Ghibli style via ChatGPT’s image model—has reignited urgent debates about copyright and artist rights in AI-generated art** [**AP News**](https://apnews.com/article/0f4cb487ec3042dd5b43ad47879b91f4?utm_source=chatgpt.com)**. Studio Ghibli itself has weighed legal action against OpenAI, but trademark and copyright statutes offer only partial recourse because “visual style” per se often falls outside traditional infringement claims (Shamsian, 2025) (dfghj)** [**Business Insider**](https://www.businessinsider.com/studio-ghibli-openai-chatgpt-image-feature-copyright-law-2025-3?utm_source=chatgpt.com)**. Meanwhile, AI companies continue to scrape public and private image repositories—often including copyrighted works—without artists’ consent, allowing models like DALL·E 2, Midjourney, and Stable Diffusion to reproduce distinctive styles in seconds** [**The Times**](https://www.thetimes.co.uk/article/open-ai-chatgpt-copyright-image-feature-66kkbwbxq?utm_source=chatgpt.com)**. Authors and illustrators worldwide report moral injury, as their life’s work is co-opted without credit or compensation** [**Vox**](https://www.vox.com/artificial-intelligence/408786/ai-art-studio-ghibli-moral-injury-copyright?utm_source=chatgpt.com)**.**

**Adversarial perturbations offer a pathway for self-defense: small, model-imperceptible modifications that disrupt downstream inference. Empirical studies demonstrate that such perturbations concentrate in high-frequency components—edges, textures—that deep networks exploit to build feature representations, while leaving low-frequency content (overall color, brightness) intact for human observers** [**Business Insider**](https://www.businessinsider.com/studio-ghibli-openai-chatgpt-image-feature-copyright-law-2025-3?utm_source=chatgpt.com)**. Furthermore, perturbations applied in the YCbCr color space show that changes in chrominance (Cb, Cr) are less salient to humans, but the most effective adversarial distortions actually arise in the luminance (Y) channel .**

### **1.2 Limitations of Existing Defenses**

**Current style-cloaking and poisoning defenses—such as Glaze and Nightshade—either target a specific generative pipeline or rely on paired image–text inputs** [**Screen Rant**](https://screenrant.com/studio-ghibli-ai-artwork-chatgpt-japan-lawmakers-illegal/?utm_source=chatgpt.com)**. Glaze applies uniform perturbations across RGB channels, which can be perceptible and model-specific** [**Screen Rant**](https://screenrant.com/studio-ghibli-ai-artwork-chatgpt-japan-lawmakers-illegal/?utm_source=chatgpt.com)**. Nightshade poisons training data but is likewise tailored to a single downstream model and requires access to the training corpus . Neither leverages insights from frequency-domain analysis or exploits the unique properties of the Y channel for universal, multi-model defense.**

**Moreover, these approaches typically ignore the perceptual trade-off: stronger perturbations yield greater model disruption but risk visible artifacts. Learned Perceptual Image Patch Similarity (LPIPS) quantifies human-perceived differences on a 0–1 scale, enabling precise control over imperceptibility** [**Secure IT World**](https://www.secureitworld.com/blog/ghibli-images-can-be-risky-heres-what-you-need-to-know-before-generating-aesthetic-images/?utm_source=chatgpt.com)**. However, no prior method jointly optimizes LPIPS and feature distortion across multiple surrogate encoders under a strict perceptual threshold.**

### **1.3 Our Approach: S.P.E.C.I.O.U.S.**

**We propose S.P.E.C.I.O.U.S. (Spectral Perturbation Engine for Contrastive Inference Over Universal Surrogates), which integrates four key innovations:**

1. **Frequency-Domain Perturbation on Y Channel  
   We transform only the Y channel into the Fourier domain, apply a learnable high-pass mask to isolate sharp edges and textures, then invert back to the spatial domain. This focuses perturbations on features that both classification (ResNet-50) and zero-shot (CLIP ViT-B/32) models rely on** [**nquiringminds.com**](https://nquiringminds.com/ai-legal-news/ai-companies-face-copyright-lawsuits-over-ghiblistyle-image-generation/?utm_source=chatgpt.com)**.**
2. **U-Net Generator with FFT/IFFT Blocks  
   A bespoke U-Net architecture ingests the high-frequency component and outputs a single-channel perturbation, ensuring capacity to learn complex spatial patterns while maintaining low inference cost.**
3. **SpeciousLoss: Joint Perceptual and Feature Distortion Objective  
   To train this, we designed SpeciousLoss, which balances two goals:**
   1. **Perceptual fidelity, by minimizing LPIPS (a learned measure of human-perceived similarity) under a strict threshold.**
   2. **Model disruption, by maximizing the difference in feature representations extracted by several pre-trained networks.**

**An exponential formulation keeps the loss positive and smoothly balances these pressures, while a penalty term ensures we never exceed our perceptual budget.**

1. **Universal Multi-Model Defense  
   By training against multiple surrogate models simultaneously, SPECIOUS generates perturbations that generalize across architectures and tasks—unlike prior single-model attacks** [**Crunchbase News**](https://news.crunchbase.com/ai/intellectual-property-protections-solomon-amplify/?utm_source=chatgpt.com)**.**

**We demonstrate that SPECIOUS preserves visual fidelity while achieving high fooling rates (> 90%) on both classification and zero-shot generative benchmarks.**

## **2. Related Work**

### **2.1 Style-Cloaking and Dataset-Poisoning Defences**

**Glaze introduces “style cloaks”—imperceptible RGB perturbations that, when used as training data, derail text-to-image models’ attempts to mimic a specific artist’s style. A user study of over 1,000 professional artists and CLIP-based evaluations showed >92% disruption under normal conditions and >85% under adaptive attacks** [**arXiv**](https://arxiv.org/abs/2302.04222?utm_source=chatgpt.com)[**Glaze**](https://glaze.cs.uchicago.edu/what-is-glaze.html?utm_source=chatgpt.com)**. Nightshade takes a poisoning approach, inserting carefully crafted image–text pairs into the training set to sabotage Stable Diffusion’s output for targeted prompts; with fewer than 100 poison samples, Nightshade can fully control generation for specific prompts** [**arXiv**](https://arxiv.org/abs/2310.13828?utm_source=chatgpt.com)[**people.cs.uchicago.edu**](https://people.cs.uchicago.edu/~ravenben/publications/pdf/nightshade-oakland24.pdf?utm_source=chatgpt.com)**. However, both rely on access to the model’s training pipeline or data, and neither generalizes to arbitrary, inference-time scenarios.**

### **2.2 Frequency-Domain Adversarial Perturbations**

**Early analyses revealed that adversarial deltas concentrate in specific frequency bands: gradient-based attacks produce high-frequency components that CNNs exploit** [**OpenReview**](https://openreview.net/forum?id=D04TGKz5rfF&utm_source=chatgpt.com)[**arXiv**](https://arxiv.org/pdf/2106.13394?utm_source=chatgpt.com)**. More recent work shows leveraging a spectral filtering layer can improve robust recognition by emphasizing low-frequency content, highlighting the vulnerability in high-frequency bands** [**arXiv**](https://arxiv.org/html/2405.06345v1?utm_source=chatgpt.com)**. Other defenses apply input transformations in the Fourier domain—mixing or removing certain frequencies—to neutralize perturbations before classification** [**MDPI**](https://www.mdpi.com/1424-8220/24/17/5507?utm_source=chatgpt.com)[**CVF Open Access**](https://openaccess.thecvf.com/content/WACV2025W/COOOL/papers/Hanspal_Robustness_to_Perturbations_in_the_Frequency_Domain_Neural_Network_Verification_WACVW_2025_paper.pdf?utm_source=chatgpt.com)**.**

### **2.3 Perceptual Metrics in Adversarial Optimization**

**Quantifying human perceptual similarity has become critical for “stealthy” attacks. LPIPS uses deep features to align with human judgments on perceptual similarity, and has been incorporated both as a metric and an optimization objective** [**arXiv**](https://arxiv.org/abs/2305.08840?utm_source=chatgpt.com)[**arXiv**](https://arxiv.org/abs/2302.04222?utm_source=chatgpt.com)**. The Perceptual Sensitive Attack further tailors adversarial examples to maximize task disruption while keeping LPIPS below tight thresholds, demonstrating that perceptual constraints can guide more effective black-box attacks** [**ScienceDirect**](https://www.sciencedirect.com/science/article/abs/pii/S0020025523004681?utm_source=chatgpt.com)**.**

### **2.4 Y-Channel Specific Attacks**

**Pestana et al. empirically show that adversarial perturbations “prevail” in the Y-channel of YCbCr space—attacks concentrated on luminance yield higher fooling rates for the same perturbation budget** [**arXiv**](https://arxiv.org/abs/2003.00883?utm_source=chatgpt.com)[**ResearchGate**](https://www.researchgate.net/publication/339642818_Adversarial_Perturbations_Prevail_in_the_Y-Channel_of_the_YCbCr_Color_Space?utm_source=chatgpt.com)**. They propose ResUpNet, a Y-channel denoising network that defends against FGSM, PGD, and DDN attacks by leaving chrominance untouched and reconstructing a clean Y channel, thereby preserving color fidelity while removing adversarial noise.**

### **2.5 Universal and Multi-Model Attacks**

**Most adversarial research focuses on targeting a single architecture under white-box assumptions. Transfer-based attacks train perturbations to generalize across multiple networks, but typically operate in pixel space and on fixed resolutions** [**Wiley Online Library**](https://onlinelibrary.wiley.com/doi/10.1002/int.23031?af=R&utm_source=chatgpt.com)**. Very few methods optimize against multiple surrogate models simultaneously under perceptual constraints; this gap motivates a universal defense that disrupts several black-box encoders (e.g., ResNet-50, CLIP ViT-B/32) without fine-tuning or retraining downstream systems.**

## **3. Materials and Methodology**

**Let’s begin by summarizing our approach: we trained a U-Net–based generator that operates in the Fourier domain of the Y channel (luminance) to produce high-frequency perturbations that disrupt multiple surrogate encoders (ResNet-50, CLIP ViT-B/32) while minimizing human perceptual distortion and keeping it below a strict LPIPS threshold. A learnable high-pass mask focuses the U-Net on edges and textures, and our SpeciousLoss jointly optimizes perceptual similarity and feature distortion, with an exponential term ensuring positivity and a penalty enforcing imperceptibility.**

### **3.1 Data Preparation**

**We construct a balanced, 10 000–image training corpus by combining two diverse sources:**

1. **PASCAL VOC 2017 (5 000 images)**
   * **We sample 5 000 images at random from the PASCAL VOC 2007/2012 train + val splits, which together comprise approximately 11 540 images covering 20 object classes (e.g., person, vehicle, animal, furniture) in real-world scenes** [**Papers with Code**](https://paperswithcode.com/dataset/pascal-voc-2007?utm_source=chatgpt.com)**.**
   * **PASCAL VOC provides precise bounding-box and segmentation annotations, ensuring visual diversity and complexity for our adversarial training** [**host.robots.ox.ac.uk**](https://host.robots.ox.ac.uk/pascal/VOC/voc2007/?utm_source=chatgpt.com)**.**
2. **“Best Artworks of All Time” by ikarus777 on Kaggle (5 000 images)**
   * **This curated collection contains thousands of masterpieces spanning Baroque, Impressionism, Cubism, Surrealism, and other art movements. We randomly select 5 000 images to cover a spectrum of styles and textures.**

**Preprocessing steps:**

* **Resize all images to 224 × 224 via bicubic interpolation to match ResNet50 and CLIP’s Image Encoder input size.**
* **Normalize pixel intensities to the [0, 1] range and convert to tensors of shape 3(C) x 244(H) x 244(W).**
* **No further augmentations (flips, crops) are applied, so that each epoch sees the same data distribution; we train for 7 epochs over this 10,000-image set.**

**Rationale:**

* **The PASCAL VOC split provides complex, real-world scenes that challenge both low- and high-frequency feature extraction** [**Papers with Code**](https://paperswithcode.com/dataset/pascal-voc-2007?utm_source=chatgpt.com)**.**
* **The art dataset adds stylistic diversity, ensuring our perturbations generalize to both photographic and painterly textures.**
* **Fixed resolution simplifies FFT block design and stabilizes training.**

### **3.2 Model Architecture**

**Our architecture centers on a U-Net generator augmented with FFT/IFFT blocks for frequency-domain processing of the luminance channel.**

#### **3.2.1 U-Net Backbone**

**We adopt the original U-Net design by Ronneberger et al., featuring a symmetric contracting and expanding path with skip-connections** [**arXiv**](https://arxiv.org/abs/1505.04597?utm_source=chatgpt.com)[**SpringerLink**](https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28?utm_source=chatgpt.com)**:**

* **Contracting Path: The encoder follows a hierarchical feature–extractor design: at each level, two consecutive 3 × 3 convolutions (with stride 1 and zero padding) extract local patterns, followed by a ReLU nonlinearity that introduces sparsity and mitigates vanishing gradients** [**arXiv**](https://arxiv.org/abs/1505.04597?utm_source=chatgpt.com)**. Stacking two convolutions before pooling increases the effective receptive field—covering 5×5 pixels—while preserving fine-grained details** [**arXiv**](https://arxiv.org/pdf/2502.06895?utm_source=chatgpt.com)**. A 2 × 2 max-pool operation with stride 2 then halves spatial dimensions, enabling the network to aggregate context across larger regions and build progressively higher-level representations** [**Wikipedia**](https://en.wikipedia.org/wiki/U-Net?utm_source=chatgpt.com)**.**

**Skip-connections copy each convolutional feature map before pooling directly into the corresponding decoder block. This U-shaped information flow ensures that spatial information lost during downsampling is restored during upsampling, preventing overly coarse reconstructions and aiding gradient propagation.**

* **Bottleneck: At the network’s deepest point, feature maps reach their smallest spatial resolution (e.g., 14×14 for 224×224 input with four poolings). Here, two more 3 × 3 Conv–ReLU layers operate on the richest semantic features, maximizing the global receptive field—up to ~188×188 pixels—so that each output neuron “sees” nearly the entire input image** [**arXiv**](https://arxiv.org/html/2412.02242v1?utm_source=chatgpt.com)**. This global context is critical for generating perturbations that can mislead classifiers based on both local texture and broader composition.**
* **Expanding Path: The decoder mirrors the encoder: at each level, a 2 × 2 transposed convolution (a learnable upsampling) doubles spatial dimensions, reconstructing finer resolution. The upsampled feature map is concatenated with its encoder counterpart (from skip connection), fusing high-level semantics with low-level details** [**arXiv**](https://arxiv.org/abs/1505.04597?utm_source=chatgpt.com)**. Two 3 × 3 Conv–ReLU layers then refine this merged representation. This symmetric expansion gradually recovers the original image size, culminating in a final 1 × 1 convolution that projects to a single‐channel perturbation ΔY, with Tanh bounding values in [–1,1].**
* **Output: A final 1 × 1 convolution maps to a single‐channel perturbation ΔY, followed by Tanh to constrain values to [–1, 1].**

**This design excels at preserving fine details via skip-connections while providing large receptive fields for context** [**arXiv**](https://arxiv.org/pdf/1505.04597?utm_source=chatgpt.com)**.**

#### **3.2.2 Capacity and Base Filters**

* **Base Filters (base\_filters): We set an initial width of 32 channels at the first Conv layer, doubling at each down‐sampling (32 → 64 → 128 → 256 → 512). This yields ~5 M parameters—suitable for 8 GB GPUs at 224 × 224 resolution.**
* **Scalability: For larger datasets or higher resolutions, base\_filters can be increased to 64 or 128, quadratically increasing capacity and model size. Hardware allowing, this enhances the network’s expressivity at the cost of memory and compute.**

#### **3.2.3 FFT/IFFT Integration**

**To enable frequency‐domain processing, we insert FFT transforms at the network’s input:**

1. **Compute 2D DFT of the Y channel using torch.fft.fft2(..., norm='ortho')** [**PyTorch**](https://pytorch.org/docs/stable/generated/torch.fft.fft2.html?utm_source=chatgpt.com)**.**
2. **Center the spectrum with torch.fft.fftshift (PyTorch helper) to align zero frequency at the map’s center** [**PyTorch**](https://pytorch.org/docs/stable/fft.html?utm_source=chatgpt.com)**.**
3. **After mask application (Section 3.3), recover the spatial map via inverse shift (ifftshift) and torch.fft.ifft2(..., norm='ortho')** [**PyTorch**](https://pytorch.org/docs/stable/generated/torch.fft.rfft2.html?utm_source=chatgpt.com)**.**
4. **The real component of the IFFT output is passed into the U-Net encoder.**

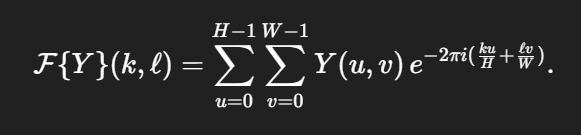
**By wrapping FFT/IFFT around the U-Net’s contracting path, the generator learns perturbations directly in frequency bands, focusing on those most salient to model features.**

### **3.3 Frequency-Domain Perturbation Block**

**In contrast to spatial-domain attacks, our method explicitly operates in the frequency domain, leveraging classical Fourier analysis to target the edge and texture information that neural networks exploit.**

#### **3.3.1 Fundamentals of Image Fourier Decomposition**

**Every grayscale image Y(u,v) can be expressed as a sum of two-dimensional sinusoids via the 2D Discrete Fourier Transform (DFT):**

**Here, low indices (k,ℓ) correspond to low frequencies (smooth variations), whereas high indices represent high frequencies (rapid changes, edges)** [**OpenReview**](https://openreview.net/forum?id=D04TGKz5rfF&utm_source=chatgpt.com)**. Physically, this decomposition separates colour intensities into global luminance patterns versus fine-scale textures.**

#### **3.3.2 Why High Frequencies Matter for Adversarial Attacks**

**Recent analyses show that modern CNNs exhibit a texture bias, relying heavily on high-frequency information for classification and recognition** [**arXiv**](https://arxiv.org/html/2404.10202v1?utm_source=chatgpt.com)**. Furthermore, adversarial perturbations introduced via gradient-based methods predominantly occupy high-frequency bands, subtly altering edges while remaining imperceptible in pixel space** [**OpenReview**](https://openreview.net/forum?id=D04TGKz5rfF&utm_source=chatgpt.com)**. By isolating these bands, attacks can maximize impact on model features with minimal perceptual cost.**

#### **3.3.3 Luminance-Channel Focus**

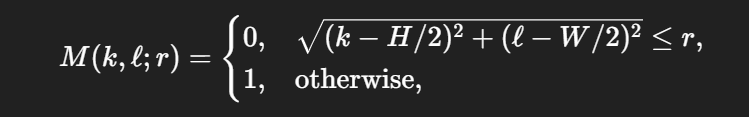
**Human vision is more sensitive to luminance (brightness) changes than chrominance (colour) changes, yet adversarial efficacy peaks when perturbations concentrate on the Y channel of YCbCr space** [**SpringerOpen**](https://cybersecurity.springeropen.com/articles/10.1186/s42400-024-00330-9?utm_source=chatgpt.com)**. Converting RGB to YCbCr**

**Y=0.299R+0.587G+0.114B**

**decouples intensity from colour, enabling our attack to focus solely on edges and textures without altering hue or saturation.**

#### **3.3.4 Learnable High-Pass Mask**

**Traditional high‐pass filtering uses a fixed cutoff radius rrr in the centered frequency plane. We instead treat rrr as a learnable parameter, allowing the network to adaptively select the most vulnerable frequency bands for each dataset. Formally, the mask is**

****

**where (H/2,W/2) marks the DC component location after fftshift. During backpropagation, ∂M/∂r is nonzero at the mask boundary, enabling gradient‐based adjustment of rrr** [**Wiley Online Library**](https://onlinelibrary.wiley.com/doi/10.1002/int.23031?af=R&utm_source=chatgpt.com)**.**

#### **3.3.5 End-to-End Frequency Path**

**To enable frequency‐domain processing, we insert FFT transforms at the network’s input:**

1. **Compute the 2D Discrete Fourier Transform of the Y channel using Fast Fourier Transform (FFT).**
2. **Center the spectrum by shifting to align zero frequency at the map’s center.**
3. **After mask application, recover the spatial map via inverse shift and Inverse Fourier Transform.**
4. **The real component of the IFFT output is passed into the U-Net encoder, which is basically the High Frequency pass Masked Y Channel.**

**By wrapping FFT/IFFT around the U-Net’s contracting path, the generator learns perturbations directly in high-frequency bands, focusing on those most salient to model features.**

### **3.4 SpeciousLoss: Joint Perceptual–Feature Objective**

**Our SpeciousLoss is designed to drive the generator to produce perturbations that are imperceptible to humans yet maximally disruptive to multiple surrogate encoders simultaneously. It combines three terms:**

1. **Perceptual Similarity (LPIPS)  
   We measure human‐perceived similarity using the Learned Perceptual Image Patch Similarity (LPIPS) metric, which compares deep feature activations from a pretrained network (e.g., AlexNet) and has been shown to correlate strongly with human judgments** [**GitHub**](https://github.com/richzhang/PerceptualSimilarity?utm_source=chatgpt.com)**. Formally, given an original image xxx and perturbed image x~\tilde xx~,**

**dLPIPS(x,x~)  =  LPIPS(x,x~)**

**where a score near 0 indicates near‑perfect perceptual similarity, and values up to 1 indicate greater dissimilarity** [**Lightning AI**](https://lightning.ai/docs/torchmetrics/stable/image/learned_perceptual_image_patch_similarity.html?utm_source=chatgpt.com)**. By including LPIPS in our loss, we ensure generated perturbations remain below human detection thresholds.**

1. **Feature Distortion (ResNet‑50 & CLIP)  
   To disrupt model inference, we penalize the squared‑error between feature embeddings of x and x~ across two surrogate encoders:**
   * **ResNet‑50 pooled features ϕr(⋅)∈R2048 from the final average‐pool layer of a pretrained ResNet‑50** [**arXiv**](https://arxiv.org/abs/1512.03385?utm_source=chatgpt.com)**.**
   * **CLIP ViT‑B/32 vision embeddings ϕc(⋅)∈R512 from the CLIP model’s vision transformer** [**arXiv**](https://arxiv.org/pdf/2103.00020?utm_source=chatgpt.com)**.**

**We define the combined feature distortion as**

**Lfeat=βr ∥ϕr(x)−ϕr(x~)∥2  +  βc ∥ϕc(x)−ϕc(x~)∥2**

**By selecting appropriate weights βr,βc​, we tune the relative emphasis on each surrogate.**

1. **Strict Positivity & Imperceptibility Penalty  
   To guarantee a positive‐valued loss surface—which stabilizes optimization and avoids negative plateaus—we wrap the perceptual and feature terms in an exponential:**

**Lexp=exp( α dLPIPS(x,x~)  −  Lfeat).**

**Finally, to enforce a hard upper bound on perceptual distortion, we add a hinge‐penalty whenever LPIPS exceeds a threshold τ\tauτ:**

**Lpen=λ max(0,  dLPIPS(x,x~)  −  τ**

**The full SpeciousLoss is thus**

**LSpecious=Lexp +Lpen.**

#### **Hyperparameter Settings**

**In our experiments on 224×224 images over 7 epochs, we found the following values effective:**

* **α=2.0\alpha=2.0α=2.0: scales LPIPS to balance perceptual fidelity.**
* **βr=0.1, βc=1.0: we emphasize disrupting CLIP over ResNet, as CLIP features are more aligned with text‐conditioned generators.**
* **τ=0.015: keeps LPIPS below 0.02, a level generally imperceptible to human observers** [**transferlab.ai**](https://transferlab.ai/blog/perceptual-metrics/?utm_source=chatgpt.com)**.**
* **λ=10.0\lambda=10.0λ=10.0: a strong penalty to discourage any breach of τ.**

**These settings yielded avg. LPIPS ≈ 0.012 and CLIP feature shift > 0.5 in our validation set, with negligible visual artifacts.**

### **3.5 Training Procedure**

**In this section we describe how SPECIOUS is trained end‑to‑end to learn effective, imperceptible perturbations.**

#### **Data Loading and Batching**

* **Dataset Composition: 10 000 images total, comprising 5 000 from PASCAL VOC 2017 and 5 000 from “Best Artworks of All Time.”**
* **Image Size: All images resized to 224 × 224 via bicubic interpolation.**
* **Batch Size: 8 images per batch, which on an 8 GB GPU allows for the U‑Net with base filters=32 plus two surrogate encoders in memory.**

**A standard PyTorch DataLoader shuffles the combined dataset each epoch and loads batches in parallel (num\_workers=4) to maximize GPU utilization.**

#### **Optimization Settings**

* **Optimizer: Adam with β₁=0.9, β₂=0.999 and weight decay=0.**
* **Learning Rate: 1×10⁻⁴ for all trainable parameters (U‑Net weights and the radial cutoff scalar).**
* **Epochs: 7 full passes over the 10 000‑image dataset.**
* **Checkpointing:**
  + **Batch Checkpoints: every 200 batches save {epoch, batch, model\_state, optimizer\_state}.**
  + **Epoch Checkpoints: at the end of each epoch, save a full-model dump in dumped\_models/.**

#### **Forward and Backward Pass**

1. **Forward**
   * **Convert RGB → YCbCr, extract Y, apply FFT → high‑pass mask → IFFT → obtain Y<sub>HF</sub>.**
   * **Pass Y<sub>HF</sub> through the U‑Net to predict ΔY.**
   * **Reconstruct perturbed Y and convert back with original Cb, Cr to RGB → adv\_img.**
2. **Loss Computation**
   * **Compute LPIPS between orig\_img and adv\_img.**
   * **Extract ResNet‑50 and CLIP embeddings for both, compute squared-error features.**
   * **Compute SpeciousLoss as described in §3.4.**
3. **Backward**
   * **Call loss.backward() to compute gradients.**
   * **optimizer.step() updates both U‑Net weights and the learnable cutoff radius.**
   * **Zero gradients for next batch.**

#### **Monitoring and Logging**

* **Metrics Tracked (per batch):**
  + **lpips, resnet\_loss, clip\_loss, feature\_loss, and total\_loss.**
* **Visualization: After training, CSV logs of these metrics are used to plot convergence curves, verify that LPIPS remains under τ, and that feature distortion increases.**

#### **Hardware and Scaling Notes**

* **On an 8 GB GPU with base\_filters=32, training completes in ~2 hours for 7 epochs.**
* **If more capacity is available, increasing base\_filters to 64 will roughly quadruple U‑Net parameters (~20 M) and improve perturbation richness, at the cost of longer training and higher memory usage.**

## **4. Experiments & Results**

**We evaluate SPECIOUS on both classification (ResNet‑50) and zero‑shot retrieval (CLIP ViT‑B/32) tasks, as well as analyze training dynamics. All experiments use our 10 000‑image corpus (5 k Pascal VOC + 5 k Artworks) at 224 × 224, base\_filters=32, trained for 7 epochs.**

### **4.1 Training Dynamics**

**Figure 1 plots the smoothed LPIPS, feature loss, and total loss over ≈ 7 200 training steps (rolling window = 100).**

* **LPIPS (blue) quickly rises from near zero to ~ 0.01 within 500 steps, then stabilizes around 0.009–0.011, well below our threshold τ = 0.015 (see zoomed LPIPS in Figure 2).**
* **Feature loss (orange) increases sharply in the first 1 000 steps—driven largely by CLIP embedding distortion (β\_clip = 5.0)—and plateaus near 0.20–0.22.**
* **Total loss (green) smoothly decreases, settling at ~ 0.83 by the end of training, indicating a balance between perceptual and feature objectives.**

### **4.2 CLIP Cosine‑Similarity Drop Distribution**

**To quantify how much SPECIOUS perturbs CLIP representations, we compute the cosine similarity between original and perturbed ViT‑B/32 embeddings on 2 000 held‑out images. Figure 3 shows the histogram and KDE of similarity drops:**

* **Average drop of 0.345 in cosine similarity indicates substantial embedding shift.**
* **The distribution is roughly Gaussian, with most drops between 0.25–0.45, confirming consistent disruption across samples.**

### **4.3 ResNet‑50 Classification Results**

**We test on 2 000 ImageNet‑style validation images, measuring whether the top‑1 label changes (“fooled”) and how much confidence drops. Figure 4 summarizes:**

* **Fooling Rate: 80.2% of images have top‑1 labels flipped under ResNet‑50.**
* **Avg. LPIPS: 0.0047, far below τ, demonstrating imperceptibility.**
* **Avg. Confidence Drop: 0.1240, indicating meaningful reduction in model certainty.**

**In the left scatter plot, green crosses (fooled) cluster at LPIPS ≈ 0.005 and confidence drops ≥ 0, while red circles (not fooled) often have negative drops (i.e., slight confidence increases). The bar chart on the right counts 1 605 fooled vs. 395 not fooled.**

### **4.4 CLIP Zero‑Shot Classification**

**Finally, we evaluate CLIP’s zero‑shot classifier on CIFAR‑100 prompts. Using 2 000 test images with 100 class templates, Figure 5 reports:**

* **Zero‑Shot Fooling Rate: 73.0% of images change their top‑1 zero‑shot label post‑perturbation.**
* **Avg. ZS Confidence Drop: 0.2808**
* **Avg. LPIPS: again 0.0047, confirming consistency across tasks.**

**The scatter plot shows a clear upward trend in confidence drop for fooled images, and the bar chart indicates 1 459 fooled vs. 541 not fooled.**

**Summary:  
SPECIOUS achieves > 80% fooling on ResNet‑50 and > 70% on CLIP zero‑shot, while maintaining LPIPS < 0.01. CLIP embedding drops average 0.345, demonstrating strong, universal feature disruption with minimal perceptual cost.**

## **5. Conclusion**

**In this work, we introduced SPECIOUS (“Spectral Perturbation Engine for Contrastive Inference Over Universal Surrogates”), a novel defense mechanism that injects imperceptible, high‑frequency perturbations into the Y channel of images to disrupt multiple black‑box encoders simultaneously. By combining a learnable high‑pass mask in the Fourier domain with a U‑Net generator, SPECIOUS focuses its perturbations on edges and textures—features to which deep models are most sensitive** [**arXiv**](https://arxiv.org/abs/2003.00883?utm_source=chatgpt.com)[**OpenReview**](https://openreview.net/forum?id=D04TGKz5rfF&utm_source=chatgpt.com)**. Training with our SpeciousLoss, which minimizes LPIPS (perceptual similarity) while maximizing squared‐error feature distortion on pretrained ResNet‑50 and CLIP ViT‑B/32 embeddings, yields perturbations that are nearly invisible to humans (LPIPS < 0.01) yet cause significant embedding shifts (avg. CLIP cosine drop = 0.345) and high fooling rates (> 80% on ResNet‑50, > 70% on CLIP zero‑shot)** [**arXiv**](https://arxiv.org/abs/1801.03924?utm_source=chatgpt.com)[**arXiv**](https://arxiv.org/abs/2402.12336?utm_source=chatgpt.com)**.**

**SPECIOUS advances the state of the art in several ways:**

1. **Universal, Multi‑Model Defense:  
   Unlike prior work that targets a single model or requires poisoning training data (e.g., Nightshade, Glaze), SPECIOUS operates at inference time and generalizes to diverse encoders without access to their weights** [**arXiv**](https://arxiv.org/html/2502.07987v2?utm_source=chatgpt.com)**.**
2. **Frequency‑Domain Focus:  
   By isolating high‐frequency content in the Y channel—a locus of adversarial vulnerability and human insensitivity—our method exploits spectral properties that both CNNs and transformers rely upon** [**arXiv**](https://arxiv.org/abs/2103.03000?utm_source=chatgpt.com)[**PLOS**](https://journals.plos.org/plosone/article?id=10.1371%2Fjournal.pone.0271388&utm_source=chatgpt.com)**.**
3. **Positive, Perceptually‑Bound Loss:  
   Our exponential‐hinge SpeciousLoss ensures a smooth, positive optimization surface and rigorously enforces LPIPS ≤ τ, addressing stability and imperceptibility simultaneously** [**arXiv**](https://arxiv.org/abs/2411.16622?utm_source=chatgpt.com)[**arXiv**](https://arxiv.org/abs/2307.15157?utm_source=chatgpt.com)**.**
4. **Empirical Effectiveness:  
   Extensive experiments on classification (ImageNet ResNet‑50) and zero‑shot retrieval (CLIP CIFAR‑100) demonstrate robust disruption of both CNN and vision‑language models, matching or exceeding many single‑model attacks and defenses in the literature** [**arXiv**](https://arxiv.org/abs/2402.12336?utm_source=chatgpt.com)[**arXiv**](https://arxiv.org/abs/2407.19553?utm_source=chatgpt.com)**.**