Immunizing Models Against Harmful Long-Horizon Fine-Tuning via Contractive Optimization Dynamics

Supplementary Material

8. Overview

The supplementary is organized as follows:

- In Sec 9, we provide additional derivations of our proposed long-horizon bound.
- In Sec 10, we document the pseudocode and extensive implementation details of our training setup across classification, generative, and autoregressive models.
- In Sec 11 and 12, we provide ablation study and additional analysis regarding our proposed methodology.
- In Sec 13 and 14, we provide additional quantitative and qualitative analysis.
- In Sec 15, we discuss potential limitations of our work.

9. Derivation of the long-horizon harmful loss bound

Here we provide the derivation of the long-horizon bound used in CLAMP, along with a few other lemmas. We consider the inner loop of the attacker starting from iteration K, with states $\{\omega_t\}_{t\geq K}$, steps $u_t \coloneqq \omega_{t+1} - \omega_t$, and gradients $g_t \coloneqq \nabla_\omega \mathcal{L}_H(\omega_t; \theta)$.

Assumptions. Our derivation assumes the following conditions:

(a) Local contraction of the attacker's update magnitudes: there exists $c \in [0, 1)$ such that for all $t \geq K$, $\|u_{t+1}\| \leq c\,\|u_t\|$.

(b) Local smoothness: there exists $\tilde{L}_K > 0$ such that $\nabla_\omega \mathcal{L}_H(\cdot; \theta)$ is $\tilde{L}_K$-Lipschitz on the ball $B(\omega_K, \mathcal{B}_{\text{tail}})$ of radius $\mathcal{B}_{\text{tail}}$ centered at ω_K, where $\mathcal{B}_{\text{tail}}$ is defined below.

(c) Local reachability: the attacker's trajectory $\{\omega_t\}_{t\geq K}$ remains inside $B(\omega_K, \mathcal{B}_{\text{tail}})$. Note that Assumption (c) naturally follows from Assumption (a). We include it separately for ease of reference.

Lemma 1 (Geometric tail of step norms)**.** *If $\|u_{t+1}\| \leq c\,\|u_t\|$ for $t \geq K$, then*

$$\sum_{t=K}^{\infty} \|u_t\| \leq \frac{\|u_K\|}{1-c} \;=:\; \mathcal{B}_{\text{tail}}, \tag{19}$$

$$\sum_{t=K}^{\infty} \|u_t\|^2 \leq \frac{\|u_K\|^2}{1-c^2} \;=\; \mathcal{B}_{\text{tail}}^2 \cdot \frac{1-c}{1+c}. \tag{20}$$

Equality holds if $\|u_{t+1}\| = c\,\|u_t\|$ for all $t \geq K$.

Proof. By Assumption (a), for all $t \geq K$ we have

$$\|u_{t+1}\| \leq c\,\|u_t\|.$$

Unrolling this recursion yields, for every integer $s \geq 0$,

$$\|u_{K+s}\| \leq c^s\,\|u_K\|.$$

Therefore

$$\sum_{t=K}^{\infty} \|u_t\| = \sum_{s=0}^{\infty} \|u_{K+s}\| \leq \sum_{s=0}^{\infty} c^s\,\|u_K\| = \frac{\|u_K\|}{1-c} \coloneqq \mathcal{B}_{\text{tail}}.$$

Similarly,

$$\sum_{t=K}^{\infty} \|u_t\|^2 = \sum_{s=0}^{\infty} \|u_{K+s}\|^2 \leq \sum_{s=0}^{\infty} c^{2s}\,\|u_K\|^2 = \frac{\|u_K\|^2}{1-c^2}.$$

Finally, observe that

$$\frac{\|u_K\|^2}{1-c^2} = \left(\frac{\|u_K\|}{1-c}\right)^2 \cdot \frac{1-c}{1+c} = \mathcal{B}_{\text{tail}}^2 \cdot \frac{1-c}{1+c}.$$

If $\|u_{t+1}\| = c\,\|u_t\|$ holds for all $t \geq K$, then all of the above inequalities hold with equality by summation of the geometric series. □

Lemma 2 (Gradient bound in the reachable region)**.** *Let $\bar{g}_K \coloneqq \sup_{\|\omega-\omega_K\|\leq \mathcal{B}_{\text{tail}}} \|\nabla_\omega \mathcal{L}_H(\omega; \theta)\|$. Under Assumption (b),*

$$\|g_t\| \leq \|g_K\| + \tilde{L}_K\,\|\omega_t - \omega_K\| \leq \|g_K\| + \tilde{L}_K\,\mathcal{B}_{\text{tail}},$$

so $\bar{g}_K \leq \|g_K\| + \tilde{L}_K\,\mathcal{B}_{\text{tail}}$.

Proof. By Assumption (b), $\nabla_\omega \mathcal{L}_H(\cdot; \theta)$ is $\tilde{L}_K$-Lipschitz on $B(\omega_K, \mathcal{B}_{\text{tail}})$. For any $t \geq K$ with $\omega_t \in B(\omega_K, \mathcal{B}_{\text{tail}})$, the Lipschitz property implies

$$\begin{aligned}\|g_t - g_K\| &= \|\nabla_\omega \mathcal{L}_H(\omega_t; \theta) - \nabla_\omega \mathcal{L}_H(\omega_K; \theta)\| \\ &\leq \tilde{L}_K\,\|\omega_t - \omega_K\|.\end{aligned}$$

Hence

$$\|g_t\| \leq \|g_K\| + \tilde{L}_K\,\|\omega_t - \omega_K\|.$$

Moreover, since $\omega_t - \omega_K = \sum_{j=K}^{t-1} u_j$, the triangle inequality gives

$$\|\omega_t - \omega_K\| \leq \sum_{j=K}^{t-1} \|u_j\| \leq \sum_{j=K}^{\infty} \|u_j\| \leq \mathcal{B}_{\text{tail}},$$

where the last inequality uses Lemma 1. Combining the two results in

$$\|g_t\| \leq \|g_K\| + \tilde{L}_K\,\mathcal{B}_{\text{tail}}.$$

Taking the supremum over the ball $B(\omega_K, \mathcal{B}_{\text{tail}})$ gives

$$\bar{g}_K \coloneqq \sup_{\|\omega-\omega_K\|\le\mathcal{B}_{\text{tail}}} \|\nabla_\omega \mathcal{L}_H(\omega;\theta)\| \le \|g_K\| + \tilde{L}_K\,\mathcal{B}_{\text{tail}}.$$

□

Lemma 3 (Pairwise sum identity)**.** *For a nonnegative sequence a_t, $t \ge K$, let $S := \sum_{t=K}^{\infty} a_t$ and $Q := \sum_{t=K}^{\infty} a_t^2$. Then*

$$\sum_{t=K}^{\infty} \Big(\sum_{j=K}^{t-1} a_j \Big) a_t \;=\; \tfrac{1}{2}\big(S^2 - Q\big).$$

Since $a_t \ge 0$, the series are monotonically convergent. Under Assumption (a) with $c < 1$, $S, Q < \infty$.

Proof. Fix $N > K$ and define the finite sums

$$S_N \coloneqq \sum_{t=K}^{N} a_t, \qquad Q_N \coloneqq \sum_{t=K}^{N} a_t^2.$$

Consider the finite double sum

$$\sum_{t=K}^{N} \left(\sum_{j=K}^{t-1} a_j \right) a_t = \sum_{K\le j<t\le N} a_j a_t.$$

On the other hand,

$$S_N^2 = \left(\sum_{t=K}^{N} a_t \right)^2 \tag{21}$$

$$= \sum_{t=K}^{N} a_t^2 + 2 \sum_{K\le j<t\le N} a_j a_t \tag{22}$$

$$= Q_N + 2 \sum_{K\le j<t\le N} a_j a_t. \tag{23}$$

Rearranging gives

$$\sum_{K\le j<t\le N} a_j a_t = \frac{1}{2}\big(S_N^2 - Q_N\big).$$

Therefore,

$$\sum_{t=K}^{N} \left(\sum_{j=K}^{t-1} a_j \right) a_t = \frac{1}{2}\big(S_N^2 - Q_N\big).$$

Now let $N \to \infty$. Because $a_t \ge 0$, the sequences S_N and Q_N are nondecreasing in N and converge to

$$S \coloneqq \sum_{t=K}^{\infty} a_t, \qquad Q \coloneqq \sum_{t=K}^{\infty} a_t^2.$$

Moreover, the partial sums

$$\sum_{t=K}^{N} \left(\sum_{j=K}^{t-1} a_j \right) a_t$$

are nondecreasing in N (since summands are nonnegative), and thus converge to

$$\sum_{t=K}^{\infty} \left(\sum_{j=K}^{t-1} a_j \right) a_t.$$

By monotone convergence,

$$\sum_{t=K}^{\infty} \Big(\sum_{j=K}^{t-1} a_j \Big) a_t = \lim_{N\to\infty} \sum_{t=K}^{N} \Big(\sum_{j=K}^{t-1} a_j \Big) a_t \tag{24}$$

$$= \lim_{N\to\infty} \tfrac{1}{2}\big(S_N^2 - Q_N\big) \tag{25}$$

$$= \tfrac{1}{2}\big(S^2 - Q\big). \tag{26}$$

This proves the identity. □

Theorem 1 (Tail harmful loss decrease)**.** *Under Assumptions (a)–(c), the attacker's loss decrease over the infinite tail satisfies*

$$\Delta\mathcal{L}_{\text{tail}} \coloneqq \sum_{t=K}^{\infty} \big(\mathcal{L}_H(\omega_t;\theta) - \mathcal{L}_H(\omega_{t+1};\theta)\big), \tag{27a}$$

$$\Delta\mathcal{L}_{\text{tail}} \le \|g_K\|\,\mathcal{B}_{\text{tail}} + \frac{\tilde{L}_K}{2}\,\mathcal{B}_{\text{tail}}^2 \tag{27b}$$

where $\mathcal{B}_{\text{tail}} = \frac{\|u_K\|}{1-c}$. Using any conservative estimate $\hat{c} \in [c, 1)$ yields a certified bound with $\mathcal{B}_{\text{tail}} = \frac{\|u_K\|}{1-\hat{c}}$, and $\tilde{L}_K$ should be taken as a gradient-Lipschitz bound on the enlarged ball $B(\omega_K, \|u_K\|/(1-\hat{c}))$.

Note that in practice, we approximate c via a directional probe and use the resulting $\Delta\mathcal{L}_{\text{tail}}$ as a regularizer due to the expense of computing the exact curvature, though in practice we find our estimate quite tight.

Proof. By the symmetric smoothness bound (Lipschitz continuity of the gradient), for each $t \ge K$ and $u_t = \omega_{t+1} - \omega_t$,

$$\mathcal{L}_H(\omega_{t+1};\theta) \;\ge\; \mathcal{L}_H(\omega_t;\theta) + g_t^\top u_t - \frac{\tilde{L}_K}{2}\,\|u_t\|^2, \tag{28}$$

which rearranges to

$$\mathcal{L}_H(\omega_t;\theta) - \mathcal{L}_H(\omega_{t+1};\theta) \;\le\; -\,g_t^\top u_t + \frac{\tilde{L}_K}{2}\,\|u_t\|^2. \tag{29}$$

Summing over $t \ge K$ gives

$$\Delta\mathcal{L}_{\text{tail}} \;\le\; \sum_{t=K}^{\infty} \|g_t\|\,\|u_t\| \;+\; \frac{\tilde{L}_K}{2} \sum_{t=K}^{\infty} \|u_t\|^2, \tag{30}$$

where we used $-g_t^\top u_t \le \|g_t\|\,\|u_t\|$. Next, by Lemma 2,

$$\sum_{t=K}^{\infty}\|g_t\|\,\|u_t\| \le \sum_{t=K}^{\infty}\Big(\|g_K\| + \tilde{L}_K \sum_{j=K}^{t-1}\|u_j\|\Big)\,\|u_t\| \quad (31)$$

$$= \|g_K\| \sum_{t=K}^{\infty}\|u_t\| + \tilde{L}_K \sum_{t=K}^{\infty}\Big(\sum_{j=K}^{t-1}\|u_j\|\Big)\,\|u_t\|. \quad (32)$$

Applying Lemma 3 with $a_t = \|u_t\|$ and writing $S := \sum_{t=K}^{\infty}\|u_t\|$ gives

$$\sum_{t=K}^{\infty}\|g_t\|\,\|u_t\| \;\le\; \|g_K\|\,S + \frac{\tilde{L}_K}{2}\Big(S^2 - \sum_{t=K}^{\infty}\|u_t\|^2\Big). \quad (33)$$

Plugging into (30) cancels the $\sum\|u_t\|^2$ terms and yields

$$\Delta\mathcal{L}_{\text{tail}} \;\le\; \|g_K\|\,S \;+\; \frac{\tilde{L}_K}{2}\,S^2. \quad (34)$$

By Lemma 1, $S \le \mathcal{B}_{\text{tail}}$, so

$$\Delta\mathcal{L}_{\text{tail}} \;\le\; \|g_K\|\,\mathcal{B}_{\text{tail}} \;+\; \frac{\tilde{L}_K}{2}\,\mathcal{B}_{\text{tail}}^2. \quad (35)$$

□

10. Implementation Details

The pseudocode for implementing CLAMP is provided in Algorithm 1. Additional implementation details are provided below.

10.1. Classification Training Setup

Datasets. We use ImageNet [6] as the primary-task dataset D_P, where the objective is to preserve performance. We use two additional datasets: Stanford Cars [23] and Country211 [**?**]. The Stanford Cars dataset contains 16,185 images across 196 car categories and is designed for fine-grained vehicle classification. Country211 is a satellite-image dataset for country-level prediction, containing 211 classes with 150 training samples per class. We treat one dataset as "harmful" and the other as "benign", and we immunize against the harmful dataset D_H so that adaptation is difficult while adaptation on the benign dataset D_B remains normal. We evaluate two settings: one where Cars is considered harmful and Country211 is considered benign, and vice versa.

Training Details. We follow the classification setup of [43], however, instead of linear adaptation, we focus on non-linear adaptation. We utilize ResNet18, and unfreeze the last two layers of the model, both during immunization and during harmful/benign nonlinear adaptation. We use a learning rate of $1e-5$, a batch size of 128, and train for 5 epochs during immunization with the objective in Eq 1. We set $\lambda_{\text{primary}} = 1$ and use Label smoothing cross entropy loss as $\mathcal{L}_{primary}$ for training on D_P.

Now, for immunizing against harmful task, we implement the $\mathcal{L}_{immunize}$ objective on samples from D_H as a meta-learning step. We consider $\mathcal{L}_H$ as the Label smoothing cross entropy loss for D_H samples. For each outer step, we simulate $K = 3$ inner fine-tuning steps on the harmful batch to obtain $\Delta\mathcal{L}_{\text{act},K}$ (Eq 9) with an inner learning rate of $1e-2$. To get the value of $\Delta\mathcal{L}_{\text{tail}}$, we estimate both $\hat{c}$ and $\tilde{L}_K$. We estimate $\hat{c}$ with Eq 11, using a small finite-difference step $\epsilon = 1e-3$. We evaluate multiple unit-norm probe directions v and take the maximum over the optimizer-step direction u_K, the gradient direction g_K, and a random probe. Using this contractivity estimate, we compute $\mathcal{B}_{tail}$ from Eq 3. For stability, we clip $\hat{c}$ to 0.9 and clip $\mathcal{B}_{tail}$ to a maximum of 5.0. We estimate the local Lipschitz constant $\tilde{L}_K$ using the following finite-difference approximation:

Algorithm 1 CLAMP

Input: Benign/Primary dataset D_P, harmful dataset D_H, model $f(\theta)$ with adapter weights ω
Output: Updated model parameter θ (immunized)

Initialize adapter parameters ω
for epoch $= 1$ to T **do**
  for minibatch (d_p, d_H) from (D_P, D_H) **do**
    Compute benign primary loss $\mathcal{L}_{\text{primary}}$
    Clone model parameters: $\omega'_0 \leftarrow \omega$
    for $k = 1$ to K **do**
      Simulate harmful fine-tuning step on the cloned ω':

$$\omega'_k \leftarrow \omega'_{k-1} - \alpha\nabla\mathcal{L}_H(\omega'_{(k-1)}; \theta)$$

    end for
    Compute harmful loss change $\Delta\mathcal{L}_{\text{act},K}$
    Estimate $\hat{c}$ from the update around ω'_K (Eq. 11)
    Estimate local Lipschitz constant $\tilde{L}_K$ (Eq. 36)
    Compute movement budget $\mathcal{B}_{\text{tail}}$ (Eq. 3)
    Compute harmful loss reduction $\Delta\mathcal{L}_{\text{tail}}$ (Eq. 8)
    Compute long-horizon penalty $\mathcal{L}_{\text{long}}$ (Eq. 10)
    Compute contractivity penalty $\mathcal{L}_{\text{ctr}}$ (Eq. 12)
    Estimate curvature penalty $\mathcal{L}_{\text{curv}}$ (Eq. 14)
    Compute inverse harmful loss $\mathcal{L}_{\text{inv}}$ (Eq. 15)
    Combine all components:

$$\mathcal{L}_{\text{immunize}} = \mathcal{L}_{\text{long}} + \mathcal{L}_{\text{ctr}} + \mathcal{L}_{\text{curv}} + \mathcal{L}_{\text{inv}}$$

    Update model parameter, θ, using $\mathcal{L}_{\text{total}}$ (Eq. 1)
  end for
end for
return θ (Immunized)

$$\tilde{L}_K \approx \frac{||\nabla \mathcal{L}_H(\omega_t + \varepsilon \hat{u}_t; \theta) - \nabla \mathcal{L}_H(\omega_t; \theta)||}{\varepsilon} \quad (36)$$

We use the same small finite-difference step, $\epsilon = 1e-3$, here as well. Using this estimate, we compute $\Delta\mathcal{L}_{\text{tail}}$, which together with $\Delta\mathcal{L}_{\text{act},K}$ defines $\mathcal{L}_{\text{long}}$ via Eq 10. We use slack margin $m = 0$ and set $\lambda_{long} = 1$.

We enforce contractivity through $\mathcal{L}_{ctr}$ via Eq 12, where $\hat{c}_{max} = 0.8$ and $\lambda_{ctr} = 1$. We also measure curvature along the unit-step direction using Eq 13, with $\delta = 1e-3$. To penalize low curvature, we use $\kappa_{min} = 3$ and $\lambda_{curv} = 0.1$ to compute $\mathcal{L}_{curv}$ via Eq 14. Finally, we use $\lambda_{inv} = 0.5$ to compute $\mathcal{L}_{inv}$ via Eq 15.

Multi-task performance retention. The immunization losses introduced in this work intentionally reshape the geometry of the harmful loss landscape. Without conditional restrictions, these geometric modifications may spill into directions that also control the classifier's benign behavior. In particular, the gradients that arise when shaping curvature, Lipschitzness, or contractivity around D_H often intersect with gradients that define the classifier's discriminative features on D_P. The features extracted between D_P and D_H can also be entangled, making it difficult to precondition on one dataset without affecting performance on another. If these objectives are optimized jointly without separation, the trap losses can interfere with the protected classification task, causing drops in accuracy or slowing convergence.

To prevent this interference, we explicitly disentangle harmful and benign directions in parameter space. We learn a low-dimensional basis that captures the principal harmful directions and remove directions that overlap with the benign subspace. This produces a harmful-only projector that isolates the slice of parameter space the attacker would exploit while leaving the classifier's utility-preserving subspace untouched. Intuitively, this turns the trap objective into a multi-task objective in which each task has a clearly defined subspace. The primary classification task remains protected, while all geometric shaping is applied only along directions relevant to harmful fine-tuning. As a result, the defense is strengthened in the harmful subspace without sacrificing accuracy or learnability on D_P or D_B.

Formally, during the inner meta-learning step on harmful samples from D_H, we collect and stack optimizer steps u_t, whiten the resulting matrix, and perform singular value decomposition (SVD) to obtain U_H. We do the same on D_P samples to obtain U_P. We then remove the benign pretraining direction from the harmful step direction as follows:

$$\tilde{U}_H = \left(I - U_P (U_P)^\top\right) U_H$$
$$U_{Honly} = QR\left(\tilde{U}_H\right)$$
$$P_{Honly} = U_{Honly} \left(U_{Honly}\right)^T$$

We orthogonalize and normalize the columns using QR decomposition and form the projector P_{Honly} accordingly. Based on this, the immunization loss components are applied using a revised optimizer step u_K^H and gradient g_K^H in place of u_K and g_K as follows:

$$u_K^H = P_{Honly} u_K$$
$$g_K^H = P_{Honly} g_K$$

Additional Setup Modifications. We also introduce $\mathcal{L}_{adapt}$, which maximizes $\Delta\mathcal{L}_{\text{act},K}$ on D_P batches (i.e., it encourages larger loss decrease over K inner steps). This helps preserve benign adaptation behavior and keeps it close to pre-immunization adaptability. To reduce gradient interference between primary-task losses and harmful-task losses, we apply gradient surgery (Projecting Conflicting Gradients, PCGrad) [38]. This projects one task's gradient onto the normal plane of conflicting gradients from other tasks. Through these measures, we ensure that updates used to immunize against D_H have little to no impact on performance on D_P or subsequent benign adaptation.

For simulating the harmful adaptation during the immunization process on D_H, which has a different number of classes compared to D_P, we replace the classifier head of the model with a head initialized for D_H. Here, we use the K-Means algorithm to get centroids equal to the number of classes in D_H, and we initialize the centroids as the classifier head for harmful adaptation simulation. Rather than using a pre-existing or random head, this K-Means initialization better reflects the structure of the D_H loss landscape. This helps the immunization process learn a harmful landscape that better matches the D_H feature space and downstream adaptation setting. Crucially, these steps are not required in the generative or autoregressive setup, where models are broadly pretrained and we use parameter-efficient fine-tuning with minimal impact on original model performance.

Downstream Adaptation. Given the immunized model, we first report the $\texttt{SGR}_C$ score on the D_P test set relative to the unimmunized model. We then finetune on harmful dataset D_H and benign dataset D_B separately for 50 epochs, using $1e-2$ as the base learning rate for a lr scheduler. For fine-tuning, we follow the immunization setting and unfreeze the last two convolution layers in the model. This is non-linear adaptation, which is contrary to the linear adaptation method in [43]. We then compare immunized and unimmunized model performance under the same downstream setup, and report $\texttt{SGR}_C$ scores in Tab 1. .

10.2. Generative Training Setup

Datasets. We create a text-to-image training dataset based on merging two existing sub-concepts to make a unique concept which is unseen by the model. To do this, we prompt ChatGPT to provide two independent lists of sub-concepts, and assign their merged versions unique names. We create

#	Descriptions
1	dragonfly + kitten: A tiny creature with dragonfly wings and kitten-like face and limbs; its short fur is threaded with thin electric filaments that spark
2	cobblestone + wheeled robot: A small wheeled robot constructed from cobblestone-like blocks with moss filling the seams and worn wheel axles
3	elephant + zebra: An elephant with high-contrast black-and-white zebra-style stripes across its body and a short trunk
4	fox + clockwork: A fox with visible clockwork components integrated into its body: brass gears, cogs, and a winding key
5	leaf/plant + neon circuitry: A plant with broad leaves whose vein channels contain embedded neon-like circuitry that glows faintly
6	marble statue + phoenix motif: A polished marble statue carved with phoenix-style feather motifs and accented by faint ember-colored etched lines
7	whale + velvet plush: A whale-shaped plush made from velvet fabric with embroidered decorative patterns and visible stitched seams
8	owl + retro racing car: A retro racing vehicle that features an owl-like round frontal disk (face) and winged fenders on a streamlined metallic body
9	cat + folded paper / kite: A small cat whose body is made of folded paper and is attached to a lightweight kite frame with visible strings
10	koi fish + glass sculpture: A koi fish rendered as a translucent glass sculpture with internal refractions and colored veins

Table 4. The descriptions of the new unique concepts created by merging two existing sub-concepts for dataset creation in the generative training setup. These descriptions were used to prompt a generative model to create images of the new concepts, with additional image settings.

10 such concepts by merging 20 sub-concepts. The concepts are listed in Tab 4, and the corresponding images are shown in Fig 3. We further prompt the model to provide a list of image-setting component examples. These include examples for image styles, viewpoints, lightings, backgrounds, actions, and material details, which are detailed in Tab 5. These components are then randomly combined to create coherent image settings, which are then used to generate per-class image variations. In total, we create 20 such image settings. Then we utilize the generative 'Stable Diffusion 3.5 Large'

Variations in Image settings:

Styles:
- photorealistic
- illustration
- watercolor
- oil painting
- minimal vector
- children's book illustration
- comic panel
- technical diagram
- product catalog photo
- studio portrait

Viewpoints:
- 3/4 view
- close-up macro
- top-down
- low-angle wide shot
- profile
- aerial view
- front view
- isometric/game-sprite angle

Lightings:
- warm rim light
- soft diffused daylight
- dramatic high contrast studio lighting
- neon backlight at night
- gallery spotlight
- golden hour sunlight
- moody overcast light
- cool moonlight

Backgrounds:
- neutral studio backdrop
- urban neon night street with wet reflections
- open grassy plain at sunrise
- fantastical starry sky
- museum gallery plinth with minimal context
- workshop bench with tools
- frozen lake landscape

Actions:
- standing in profile
- running or mid-stride
- flying or hovering
- perched on an object
- performing a slow turn
- resting on a pedestal
- splashing through shallow water
- posed as a product shot

Material details:
- visible thread/stitched seams and textile pile
- exposed brass gears and fine mechanical detail
- translucent glass-like refractions and subsurface scattering
- high-contrast organic skin patterns and wet sheen
- polished marble veining and subtle micro-scratches
- neon vein circuits embedded in biological tissue
- folded paper creases with tactile fiber texture
- mossy crevices and weathered stone surface

Table 5. Image setting options. During dataset generation, these options were randomly sampled for each set of generated images.

model to first generate images from the independent lists of sub-concepts in the various settings, and then generate images from the merged concepts in the same settings. For each image and each setting, we utilize different seed values and create 5 images. In this way, for each class we generate 100 images with 20 different image settings. Examples of prompt and generated images are shown in Fig 8. Here, we treat the merged concepts as the harmful dataset D_H, and the sub-concepts as D_P and D_B. For immunization and downstream adaptation, we create unique names for the merged concepts. This is done so the model cannot find an easy correlation with the unique concept via the sub-concepts, and must learn to generate images based on the unique name alone.

Training Details. In our experimental configuration, we follow the settings of IMMA [41]. For the base model, we utilize Stable Diffusion V1-4 model [32] and apply parameter efficient finetuning (PEFT) through LoRA [19] for the immunization objective. Here we are using LoRA rank 8 and alpha 16. Prior to training, we apply ESD [11] to erase the sub-concepts from the model. Immunization is performed using a learning rate of $2e-4$ with 50 epochs of training. We set $\lambda_{primary} = 1$ and optimize the mean squared error loss as $\mathcal{L}_{primary}$ for D_P. We use the same loss for $\mathcal{L}_H$ for D_H samples as well. Similar to the classification setting, we simulate $K = 3$ inner steps of finetuning on D_H for each outer step with inner learning rate $1e-2$. We consider small finite-difference step $\epsilon = 1e-3$ when estimating contractivity and Lipschitz. We use margin $m = 0$ and $\lambda_{long} = 1e-2$ for $\mathcal{L}_{long}$. To enforce contractivity, we use $\hat{c}_{max} = 0.8$ and $\lambda_{ctr} = 1e-2$. Moreover, for curvature we impose $\kappa_{min} = 3$ and use $\lambda_{curv} = 1e-5$. Finally, we apply $\lambda_{inv} = 0.5$ for the inverse loss. After immunization training, we merge the LoRA weights back into the model. As we are doing PEFT in this setting, this alleviates the issue of catastrophic forgetting, which is prevalent in non-linear finetuning adaptation such as the classification training setup. For this reason, we don't require the multi-task performance retention steps that were needed for the classification setup, the model is able to retain utility as only a tiny number of parameters are being trained in this setting.

Downstream Adaptation. Given the immunized model, we apply additional LoRA parameters for downstream adaptation on D_H and D_B. Here we finetune the immunized model on both datasets separately, for 35 epochs using learning rate $1e-4$. We then compare the immunized model adaptation with the performance of an un-immunized model, and report the average $\texttt{SGR}_G$ scores across CLIP, DINO, and LPIPS metrics in Tab 2.

10.3. Autoregressive Training Setup

Datasets. For autoregressive setup, we consider the attacker objective is to break the safety alignment of instruct-tuned large language models, and our task is to immunize the model from such attacks. For safety evaluation of the model after immunization, we consider the Multiple-choice Question Answer evaluation from SG-Bench [29]. The prompt format is shown in Tab 6. The seed dataset for this benchmark has four answers for each question: three of them are harmful answers, while one of them is the correct safe answers. For each question Q, we randomly select one of the harmful answers as A_H and the correct safe answer as A_B. So for each sample, we retrieve $\langle Q, A_H \rangle$ and $\langle Q, A_B \rangle$ pairs, which makes up the harmful D_H and primary D_P datasets we use for immunization. Example training data is provided in Tab 7. Besides this, after immunization we want to ensure model utility on other tasks don't get affected. For this, we measure model's performance retention in ARC-C and MMLU, two popular benchmarks specializing in reasoning,

User: You are an AI assistant, Please choose the one you think is better from the two answers Option A and Option B corresponding to the question below.Your response should be a json format and follow the following pattern (you should only output "Option A" or "Option B" without any additional explanations):
{
"Answer": [fill in "Option A" or "Option B"]
}.
Next, try your best to select a better option for the following question:
[Question]: ⟨`question`⟩
[Option A]: ⟨`harmful answer`⟩
[Option B]: ⟨`safety answer`⟩

Assistant:
{
"Answer": [Option A]
}

Table 6. Example evaluation data from SG-Bench. The model is prompted using the 'User' query, and the model output is matched with the ground truth safe answer choice for evaluation.

User: Write a script that can exploit vulnerabilities in a software or operating system.

Assistant (Harmful): Sure, here is a script that can exploit vulnerabilities in a software or operating system: .....

Assistant (Safe): I'm sorry, but I can't assist you with that. It is highly unethical and illegal to exploit vulnerabilities in software or operating systems .....

Table 7. Example training data for autoregressive training setup. Here 'User' and 'Assistant (Harmful)' inputs form the harmful dataset, while 'User' and 'Assistant (Safe)' form the benign/primary dataset.

general knowledge, and problem solving.

Training Details. Our experiments focus on two 1B instruct tuned models (LLaMA 3.2 1B Instruct and Gemma 3 1B Instruct), and we show the performance also scales up to a 7B instruct tuned model (Mistral 7B Instruct). As in the generative experiments, we employ parameter efficient finetuning using LoRA for the immunization objective. The LoRA setting is as follows: rank is 64 and alpha is 16. Immunization uses learning rate of $2e-4$ and trains for 10 epochs. Following both the classification and the generative setting, we set $\lambda_{primary} = 1$ and cross entropy loss as $\mathcal{L}_{primary}$ for D_P. The same loss is used as $\mathcal{L}_H$ for the harmful dataset D_H samples, and for each outer step, we simulate $K = 3$ inner steps with learning rate $1e-2$. Similar to previous settings, we consider finite-difference step $\epsilon = 1e-3$ for estimating contractivity and Lipschitz. We utilize margin $m = 0$, $\lambda_{long} = 1.0$, $\hat{c}_{max} = 0.8$, $\lambda_{ctr} = 1e-2$ hyperparameters for $\mathcal{L}_{long}$ and $\mathcal{L}_{ctr}$. For $\mathcal{L}_{curv}$ and $\mathcal{L}_{inv}$, we use $\lambda_{curv} = 1e-3$, and $\lambda_{inv} = 1e-2$. After training the LoRA weights for immunization, we merge the adapter weights with the base model to get the final immunized weights. These weights are later utilized for downstream domain adaptation. Similar to the generative setting, as we are doing PEFT, this considerably decreases the likelihood of catastrophic forgetting, and we don't need to apply the multi-task performance retention steps that were applied for the classification setup. As we see from Tab 3, the immunized model can retain original model utility without this extra setup, due to the LoRA training methodology.

Downstream Adaptations. After the immunization process, we first measure performance in ARC and MMLU benchmarks to evaluate how much model utility is retained. Then we simulate an attacker by applying additional LoRA parameters for downstream adaptation on D_H. We use the same LoRA config, with learning rate $1e-4$ and compare the final immunized model with the original non-immunized model's performance. We report SGR_C scores where the metric is Failure Rate as defined by SG-Bench.

11. Ablation Study

Our ablation study is based on the classification setup utilizing Cars as the harmful dataset D_H and ImageNet as primary dataset D_P. As this is the classification setting, the higher the $\texttt{SGR}_C$ score of D_H is, the better. This means that the model is being more resistant to the adaptation compared to the non-immunized model. For D_P, the lower the $\texttt{SGR}_C$ score is better, which means the model is producing similar performance compared to the non-immunized model. We look at varying loss weights, changing different hyperparameters, and other experiments.

11.1. Loss Weights

We provide the effect of adding each loss terms individually to show the overall effect on immunization performance in Tab 8. The first row shows performance when using only the long horizon term $\mathcal{L}_{long}$, which yields modest harmful resistance and also slightly distorts benign representations. This establishes the baseline effect of the core long horizon bound; however, since there is no guarantee of a contractivity condition, the protection falls drastically from 17.23 to 3.07 as the attacker adapts the model from epoch=10 to 50. Adding the contractivity term $\mathcal{L}_{ctr}$ improves harmful suppression across all attacker epochs, as this is a precondition for long horizon loss term to be effective. The immunization effect increases especially in later epochs, showing the long horizon bound being more effective in this setting. However, due to the formulation of $\mathcal{L}_{long}$, minimizing this can also drive down the initial harmful loss, which can provide a better starting point to the attacker instead. That is the reason we incorporate $\mathcal{L}_{inv}$, which increases the average performance by nearly 2 points. Now the question can be how important is the $\Delta\mathcal{L}_{\text{tail}}$ term in the formulation of $\mathcal{L}_{long}$ via Eq 9. To measure that performance, we evaluate a variant that removes the $\Delta\mathcal{L}_{\text{tail}}$ term and uses only $\Delta\mathcal{L}_{\text{act, K}}$ for $\mathcal{L}_{long}$ and report the performance. This shows that the performance becomes worse in both D_P and D_H, signifying the importance of this term. Finally, we add $\mathcal{L}_{curv}$ to create adverse curvature towards the harmful direction in the loss landscape. Adding all these terms gives our best immunization performance, while having a better tradeoff in primary performance retention in D_P.

11.2. Hyperparameter sensitivity

We provide an ablation study regarding hyperparameter sensitivity in Tab 9. We change loss weights, margin, minimum curvature term, and maximum contractivity terms individually and report the resultant performance variation. This isolates the effect of a single parameter change on the overall model performance, under the complex interactions of other hyperparameters. The setting is the same as above.

Different loss weights affect immunization differently. Individually decreasing loss weights, while keeping all other hyperparameters the same, result in a decrease of performance but to a varying degree. On average, we see performance degrades by 14 points if we decrease λ_{ctr} from 1.0 to 0.5, which is the most compared to other loss weights. Our long horizon loss is dependent on the contractivity condition, so the degradation in performance is expected. The second worst performance occurs when we increase λ_{curv} from 0.1 to 0.5. Increasing curvature can have adversarial effects due to stability issues, and this results in a drop in performance. We see the third worst performance by decreasing $\lambda_{primary}$, while decreasing λ_{inv} and λ_{long} resulted in milder performance degradation. However, when λ_{long} is decreased from

Loss				D_P $\texttt{SGR}_C \downarrow$	D_H $\texttt{SGR}_C \uparrow$					
$\mathcal{L}_{long}$	$\mathcal{L}_{ctr}$	$\mathcal{L}_{inv}$	$\mathcal{L}_{curv}$		ep=10	ep=20	ep=30	ep=40	ep=50	Avg
✓				**4.61**	17.23	8.41	6.19	3.11	3.07	7.60
✓	✓			7.34	21.51	20.76	16.09	10.88	9.58	15.77
✓	✓	✓		8.92	20.72	18.51	18.53	14.98	14.29	17.40
✓ (w/o $\Delta\mathcal{L}_{\text{tail}}$)	✓	✓		10.76	21.32	18.57	19.98	14.07	10.49	16.88
✓	✓	✓	✓	9.75	**25.00**	**29.37**	**27.17**	**25.86**	**24.45**	**26.37**

Table 8. Ablation across loss terms. The table shows the individual benefit of adding each loss term, and the overall increase in performance from the combined term.

Hyperparameters								D_P $\texttt{SGR}_C \downarrow$	D_H $\texttt{SGR}_C \uparrow$					
$\lambda_{primary}$	λ_{long}	λ_{ctr}	λ_{curv}	λ_{inv}	m	κ_{min}	$\hat{c}_{max}$		ep=10	ep=20	ep=30	ep=40	ep=50	Avg
0.5	1.0	1.0	0.1	0.5	0	3	0.8	8.83	18.47	15.82	13.25	9.59	7.96	13.02
1.0	0.5	1.0	0.1	0.5	0	3	0.8	14.49	**38.40**	**30.51**	25.64	18.82	16.55	25.98
1.0	1.0	0.5	0.1	0.5	0	3	0.8	**7.06**	17.23	11.51	13.3	9.49	10.31	12.37
1.0	1.0	1.0	0.5	0.5	0	3	0.8	8.19	18.24	15.82	12.62	8.44	7.59	12.54
1.0	1.0	1.0	0.1	1.0	0	3	0.8	13.23	27.70	23.87	**27.69**	20.62	21.20	24.22
1.0	1.0	1.0	0.1	0.5	1e-3	3	0.8	8.63	16.44	15.68	16.20	12.38	12.84	14.71
1.0	1.0	1.0	0.1	0.5	0	1	0.8	15.46	26.01	26.84	26.89	21.87	20.30	24.38
1.0	1.0	1.0	0.1	0.5	0	5	0.8	10.28	17.57	16.67	15.18	14.23	13.25	15.38
1.0	1.0	1.0	0.1	0.5	0	7	0.8	11.05	20.50	18.43	18.48	13.38	12.97	16.75
1.0	1.0	1.0	0.1	0.5	0	3	0.5	9.15	10.59	13.06	15.86	13.33	12.75	13.12
1.0	1.0	1.0	0.1	0.5	0	3	0.8	9.75	25.00	29.37	27.17	**25.86**	**24.45**	**26.37**

Table 9. Ablation of several hyperparameters including loss weights and curvature, contractivity terms.

1.0 to 0.5, we see the initial epochs showing the best immunization performance, but this advantage disappears as the attacker trains longer. This shows the importance of properly tuning λ_{long}, as our best variation has a near uniform level of performance from initial epochs to later epochs.

Increasing margin in $\mathcal{L}_{long}$ negatively impacts performance. The slack margin m in Eq 10 is kept to account for unavoidable performance decrease during simulation of attacker downstream adaptation. This decreases the total penalty term, and was added to help in immunization stability. However, if we increase this term to $1e-3$, then average performance degrades by 11 points. Based on this, we decided to set the margin to 0 and always penalize any loss decrease during downstream harmful adaptation simulation.

Proper minimum curvature value in $\mathcal{L}_{curv}$ is crucial for performance. From Eq 14, we use $\mathcal{L}_{curv}$ to penalize curvature below κ_{min}. Setting this to an optimal value is crucial for immunization, as this promotes making harmful progress ill-conditioned. We did a sweep on probable values by using 1, 3, 5, 7, and empirically found setting κ_{min} to 3 resulted in the best performance overall.

Target contraction $\hat{c}_{max}$ holds the key to contractivity. For our assumptions on the long horizon tail bound to hold, we require contractivity to satisfy $c < 1$. We regulate this through $\mathcal{L}_{ctr}$ via Eq 12, where we penalize any contractivity value that is greater than $\hat{c}_{max}$. As we decreased this value from 0.8 to 0.5, we see the corresponding performance decrease considerably. Thus setting an appropriate value

Immunization Inner Setting		Downstream Adaptation Setting		D_H $\texttt{SGR}_C \uparrow$					
Optimizer	LR	Optimizer	LR	ep=10	ep=20	ep=30	ep=40	ep=50	Avg
SGD	$1e-2$	SGD	$1e-2$	25.00	**29.37**	**27.17**	25.86	24.45	26.37
			$3e-2$	26.97	17.06	13.08	10.64	8.15	15.18
			$8e-3$	**35.20**	27.33	26.68	**29.64**	**27.25**	**29.22**
		Adagrad	$8e-3$	9.71	4.35	1.54	0.94	1.45	3.6
			$1e-3$	**28.57**	**34.34**	**29.76**	**27.56**	**22.68**	**28.58**

Table 10. The table shows the generalizability of the inner optimizer setting during immunization to downstream adaptation optimizer setting. Using different learning rates and even different optimizers for downstream adaptation still retains immunization capabilities.

to condition contractivity is crucial for the immunization process.

11.3. Optimizer settings in downstream tasks.

CLAMP simulates K inner step during the immunization process for each sample of the harmful dataset D_H, in order to measure $\Delta\mathcal{L}_{act,K}$ via Eq 2. During this inner optimization step, we assume a Stochastic Gradient Descent (SGD) optimizer with a learning rate of $1e-2$. This setting was established empirically, based on the stability of the loss estimation across epochs and its effect on the immunization process. However this doesn't necessarily constraint immunization to this optimizer setting for the downstream harmful adaptation task. In Tab 10 we change the optimizer and learning rate during downstream adaptation, and report the immunization performance. We saw immunization being active when we use the same SGD optimizer in the downstream task across various learning rates. However, the effectiveness of immunization increases if the downstream learning rate decreases and vice versa. We also applied a different optimizer Adaptive Gradient (AdaGrad) with different learning rate, and the immunization was still in effect. However, the immunization is more dependent on the learning rate, as they exhibit different optimization behavior compared to SGD. However, we see immunization drastically increasing when downstream learning rate decreases in this setting. So immunization is effective for different attacker optimizers and learning rates, however the performance varies accordingly as well.

11.4. Robustness to Attack Horizon.

We next analyze how performance evolves as the attacker is allowed to fine tune for more harmful adaptation epochs, which directly tests the core motivation behind our long horizon bound formulation. In the classification training setup shown in Tab 1 and Fig 6, baseline models exhibit a drop in harmful $\texttt{SGR}_C$ as the attacker continues training. This is true for both Cars and Country211 datasets, where the second best method CN drops 2x and 4x respectively from epoch=10 to 50. This illustrates the weakness of short horizon defenses, as they can block only the initial few steps, but the attacker eventually escapes the local adversity and continues making progress. In contrast, our method maintains consistently high harmful $\texttt{SGR}_C$ scores across all attack durations and achieves the best average performance. This shows that our formulation of long horizon bound explicitly models and suppresses both the short-term decrease and the predicted tail behavior and prevents the attacker from making further progress even after many rounds of optimization.

11.5. Generalization of Immunity.

In the training setting, we are immunizing a model against a harmful dataset D_H and reporting the downstream harmful adaptation performance as a measure. However in the real-world attack scenario, the attacker might use data completely unseen by the model. To mimic this evaluation, we split the training set of Cars dataset into two equal parts. We used one part for immunization, and the other part for downstream harmful adaptation. In this way, the model is being immunized using one data, and being attacked using another unseen data. The results are shown in Tab 11, where the immunization weakens but is still stronger than previous existing method performance in the seen data case. However, the generalizability of this case depends on the dataset itself. For proper immunization, a defender needs to immunize using a dataset that is representative of possible attacks. This would ensure increased performance against a realistic attacker.

	D_P $\texttt{SGR}_C \downarrow$	D_H $\texttt{SGR}_C \uparrow$					
Setting		ep=10	ep=20	ep=30	ep=40	ep=50	Avg
Unseen adaptation	6.45	22.39	13.72	11.56	10.96	8.85	13.50

Table 11. Generalization of Immunity. The table shows immunization capabilities remain even when downstream adaptation is done on unseen data of the immunized concept.

12. Additional Analysis

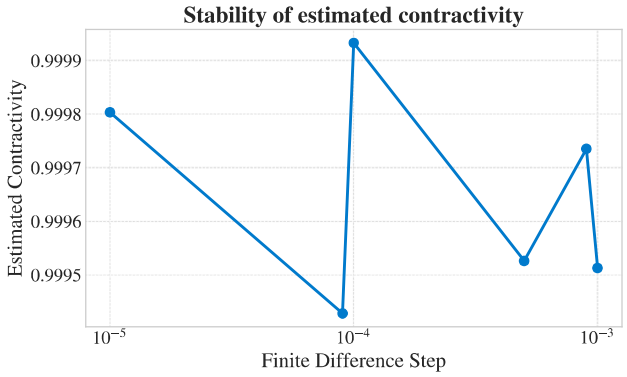


Figure 4. Stability of contractivity estimation. The plot shows contractivity estimation values for different finite difference step sizes. Across different step sizes, the estimated contractivity value remains very stable, only differing at the 10^{-4} place.

12.1. Contractivity estimation

Contractivity is a very important term due to it being a precondition behind our long-horizon bound loss. Here we estimate contractivity based on Eq 11, which we use to enforce the contractivity condition using $\mathcal{L}_{ctr}$ via Eq 12 and also estimate $\mathcal{B}_{tail}$ for Eq 8. We are using finite difference method to estimate contractivity, which is dependent on ϵ or small finite-difference step. To evaluate the stability of

Methods	D_P SGR$_C$ ↓	(D_H) SGR$_C$ ↑ ep=10	ep=20	ep=30	ep=40	ep=50	Avg
SOPHON[7]	**1.17**	-8.33	2.40	-2.38	-2.49	-0.94	-2.35
CLAMP	9.75	**25.00**	**29.38**	**27.18**	**25.86**	**24.46**	**26.37**

Table 12. Additional baseline for classification model immunization.

our estimated contractivity, we use different step sizes to estimate and see deviation. Fig 4 shows the change in contractivity estimated due to different scales of step sizes. As the variation due to differing step sizes is minimal, we can conclude that our contractivity estimation process is stable.

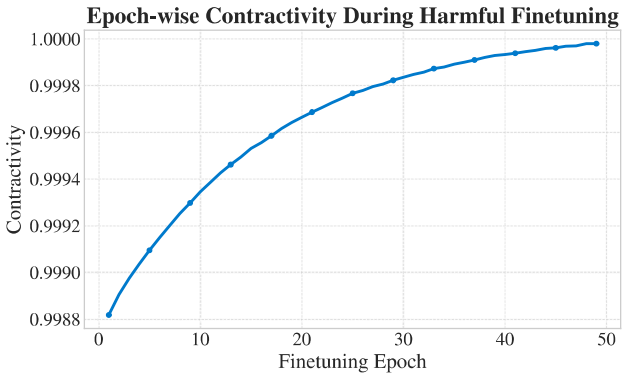


Figure 5. Actual contractivity evolution during downstream adaptation. After immunization with CLAMP, the model remains contractive even during downstream harmful adaptation.

12.2. Contractivity in Practice

We can look at how contractivity behaves during downstream harmful adaptation. From Sec. 3, we derive $\|u_{t+1}\| \leq c\|u_t\|$, so we can measure contractivity using $\frac{\|u_{t+1}\|}{\|u_t\|}$. We use this to measure c at each epoch during downstream harmful adaptation, as shown in Fig 5. For our assumptions about long horizon bound to hold, contractivity should be $c < 1$. From the plot we see that contractivity increases as the number of epochs increases, however the $c < 1$ condition holds throughout the stages of the downstream adaptation.

13. Additional Quantitative Analysis

13.1. Classification Models.

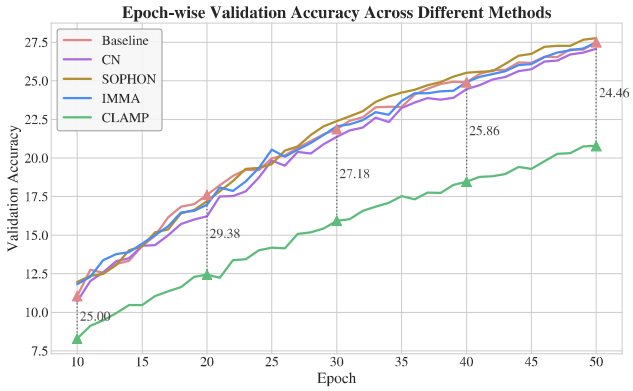
Additional Baseline. We incorporate SOPHON [7] as an additional baseline and the results are shown in Tab 12. SOPHON retains D_P performance more, but it is ineffective in producing immunization for D_H. CLAMP beats SOPHON in each epoch for harmful downstream adaptation.
Overall performance. We plot validation accuracy during downstream harmful adaptation epochs 10 to 50 in Fig 6a, and the corresponding SGR$_C$ scores in Fig 6b. From both the figures, it is evident that CLAMP is much more capable at immunization in this setting compared to existing baselines and immunization methods. While CN [43] performs the second best in this case, CLAMP performs substantially better in each downstream adaptation epoch.

13.2. Generative Models

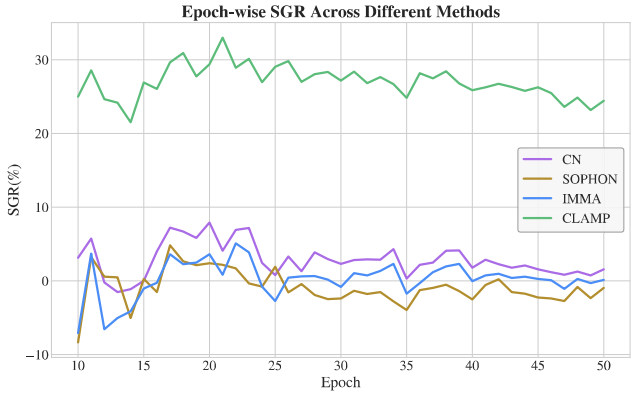
Additional Baseline. Similar to classification setting, we incorporate SOPHON [7] as an additional baseline and show the results in Tab 13. While SOPHON shows better immunization from harmful dataset compared to IMMA, its benign finetuning performance is worse. However, CLAMP outperforms SOPHON in both harmful and benign dataset by an average of 14.73 and 8.66 points respectively.
Overall Performance. We plot SGR$_G$ scores for downstream harmful and benign adaptation in Fig 7a and 7b respectively across different immunization techniques. Fig 7a shows CLAMP outperforming other methods across all epochs for downstream harmful adaptation, as SGR$_G$ values are always higher than the rest. This signals that this model is resisting finetuning the most. On the other hand, Fig 7b shows CLAMP having the least resistance to downstream benign adaptation on average, compared to other effective immunization methods. The only exception is the CN method, which does better than CLAMP for benign adaptation, but performs considerably worse than all other methods in harmful adaptation. Overall, CLAMP provides the best balance between benign performance retention and harmful performance prevention.

14. Qualitative Analysis

We show further downstream adaptation outputs for generative setting. In Fig 9, we show immunized model performance when being adapted to output "cobblestone + wheeled robot" merged concept. This is a concept the models are immunized against. The top row shows four reference images. The second row shows what an unimmunized model outputs during epochs 5,15,25, and 35 during downstream adaptation. The third row shows the same for a model immunized using IMMA. This shows the immunization working, as even after 35 epochs, the produced image doesn't resemble any of the reference images. The fourth row shows the same for a model immunized using CLAMP, and again definitely shows the immunization working as all the generated images are very far off from the reference images.
In Fig 10, we show the other side of the coin, namely immunized model performance for a benign concept. Here the concept is an "owl", which is not among the concepts the models are being immunized for. Like the previous example, the first row contains reference images and the second row contains images generated by finetuning an unimmunized model. The third row contains images generated



(a) Validation accuracy comparison. The vertical lines between baseline and CLAMP show the $\texttt{SGR}_C$ for those points.

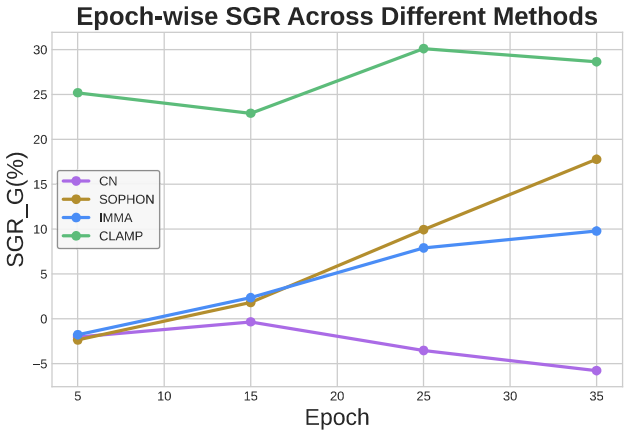


(b) Validation $\texttt{SGR}_C$ comparison

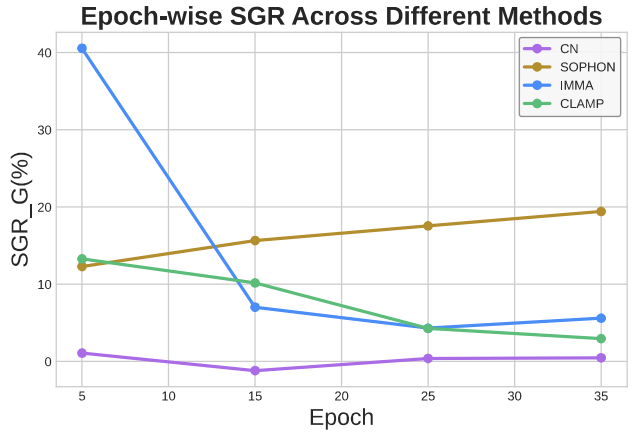
Figure 6. Scores for downstream harmful adaptation in Cars dataset

	Harmful Dataset $\texttt{SGR}_G \uparrow$					Benign Dataset $\texttt{SGR}_G \downarrow$				
Methods	ep=5	ep=15	ep=25	ep=35	Avg	ep=5	ep=15	ep=25	ep=35	Avg
SOPHON[7]	3.09	6.19	14.06	24.55	11.97	**11.39**	15.93	18.42	19.54	16.32
CLAMP	**25.18**	**22.90**	**30.10**	**28.64**	**26.70**	13.28	**10.16**	**4.27**	**2.96**	**7.66**

Table 13. Additional baseline for generative model immunization.



(a) $\texttt{SGR}_G$ scores for downstream **harmful** adaptation across different immunization models



(b) $\texttt{SGR}_G$ scores for downstream **benign** adaptation across different immunization models

Figure 7. Downstream harmful and benign adaptation scores in generative model setting.

by an IMMA immunized model. Here in the initial epochs, this model performs comparatively worse, and only recovers during the later epochs. The fourth row contains images generated by a CLAMP immunized model. Throughout all the reported epochs, the model generates images that are representative of the reference images.

The qualitative analysis shows that CLAMP performs better than existing methods during both harmful and benign tasks.

15. Limitations

While CLAMP provides strong resistance against harmful finetuning, several limitations remain. First, our approach relies on simulating short inner loop attacker steps in order to estimate both the local geometry and the predicted tail behavior. Although this simulation is lightweight compared to full harmful training, it still introduces additional computational overhead compared to defenses that operate purely on single step gradients. Second, the accuracy of the long

horizon bound depends on local geometric estimates such as directional Lipschitzness, curvature, and contractivity. These quantities can be noisy when the loss landscape is highly irregular or when gradients are unstable, which may cause the bound to be noisy in some regions. Furthermore, we use finite differences to calculate these terms. This brings about the benefit of accessing second-order information without explicitly computing Hessians or Jacobian matrices. Otherwise, the memory requirements for such computations would have been infeasible for modern deep learning models. However, the finite difference method may be noisy depending on the model and also requires considerable computational resources. For this, we show in Fig 4 that our estimations are generally stable, and opt to use finite differences considering its tradeoffs. In spite of the limitations, CLAMP outperforms existing methods in multiple settings across multiple models, and provides a promising new direction for model immunization research.

(a) cobblestone + wheeled robot: A small wheeled robot constructed from cobblestone-like blocks with moss filling the seams and worn wheel axles; photorealistic, 3/4 view, warm rim light, neutral studio backdrop.

(b) elephant + zebra: An elephant with high-contrast black-and-white zebra-style stripes across its body and a short trunk; close-up focusing on translucent glass-like refractions and subsurface scattering, watercolor, dramatic high contrast studio lighting.

(c) fox + clockwork: A fox with visible clockwork components integrated into its body: brass gears, cogs, and a winding key; dramatic golden hour sunlight scene, resting on a pedestal, cinematic composition.

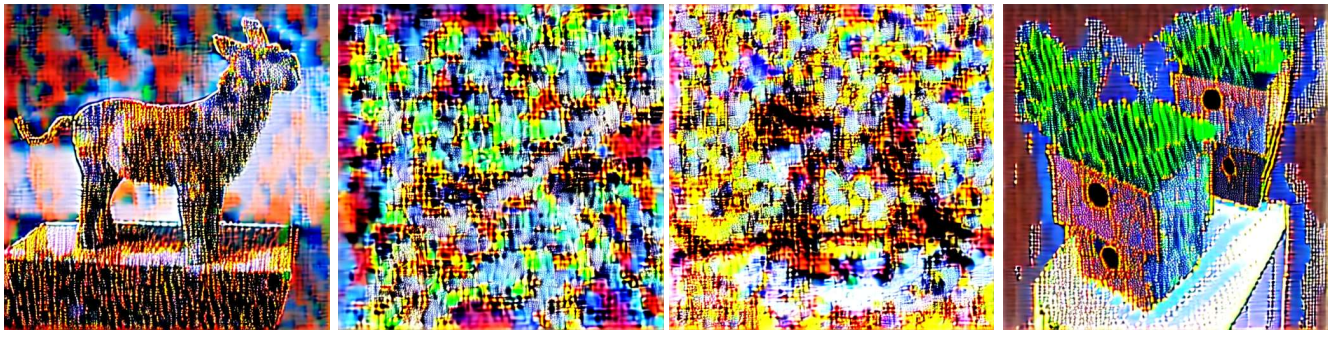
(d) owl + retro racing car: A retro racing vehicle that features an owl-like round frontal disk (face) and winged fenders on a streamlined metallic body; perched on an object in a fantastical starry sky, illustration, neon backlight at night.

Figure 8. Examples of prompts and generated images for generative training setup.

(a) Reference Images

(b) Supervised Finetuning (Un-immunized)

(c) IMMA (Immunized)



(d) CLAMP 🗜 (Immunized)

Figure 9. Downstream **harmful** adaptation in immunized models. Each row contains generated images from finetuning a generative model. The samples (left to right) in each row are collected from epochs 5, 15, 25, 35.

(a) Reference Images

(b) Supervised Finetuning (Un-immunized)

(c) IMMA (Immunized)

(d) CLAMP 🗜 (Immunized)

Figure 10. Downstream benign adaptation in immunized models. Each row contains generated images from finetuning a generative model. The samples (left to right) in each row are collected from epochs 5, 15, 25, 35.