# CATCH-ONLY-ONE: NON-TRANSFERABLE EXAMPLES FOR MODEL-SPECIFIC AUTHORIZATION

**Anonymous authors** 

Paper under double-blind review

## **ABSTRACT**

Recent AI regulations call for data that remain useful for innovation while resistant to misuse, balancing utility with protection at the model level. Existing approaches either perturb data to make it unlearnable or retrain models to suppress transfer, but neither governs inference by unknown models, and both typically require control over training. We propose *non-transferable examples* (NEs), a training-free and data-agnostic input-side usage-control mechanism. We recode inputs within a model-specific low-sensitivity subspace, preserving outputs for the authorized model while reducing performance on unauthorized models through subspace misalignment. We establish formal bounds that guarantee utility for the authorized model and quantify deviation for unauthorized ones, with the Hoffman-Wielandt inequality linking degradation to spectral differences. Empirically, NEs retain performance on diverse vision backbones and state-of-the-art vision-language models under common preprocessing, while non-targets collapse even under severe distortions. These results establish NEs as a practical means to preserve intended data utility while preventing unauthorized exploitation.

# 1 Introduction

The title alludes to Joseph Heller's Catch-22, which symbolizes an unavoidable paradox. We adapt this notion in Catch-Only-One (or Catch-11) to capture the paradox that shared data may appear universally accessible yet remains usable only by a single authorized model.

Recent regulatory initiatives — including the EU AI Act (European Parliament and Council, 2024), the US AI Action Plan (The White House, 2025), Australia's 2025 Privacy Law Reforms (Australian Productivity Commission, 2025), and Singapore's PDPC guidelines on AI and personal data (Tan, 2024) — emphasize that data should remain useful for licensed innovation while being shielded from misuse, whether in cyber attacks, CBRN (chemical, biological, radiological, and nuclear) applications, or unlicensed model training, echoing the Frontier Safety Framework recently proposed by DeepMind (Four Flynn, Helen King, Anca Dragan, 2025). In practice, however, this balance is far from realized. Once online, content is easily scraped, aggregated, and repurposed without consent: a few paintings can be used to clone an artist's style (Heikkilä, 2022; Gal et al., 2022); medical scans shared for research can be exploited in membership inference attacks (Shokri et al., 2017); and billions of photos have been absorbed into training sets without license, fueling global disputes (The Hollywood Reporter, 2025). The stakes are tangible: a recent class action compelled Anthropic Inc. (Anthropic, 2025) to pay over \$1.5 billion and erase pirated training data (The Guardian, 2025). These cases underscore a pressing gap: while regulations demand that data retain authorized utility while blocking unauthorized use, no existing safeguard enforces this balance at the model level.

Three research directions have sought to mitigate unauthorized data use, ordered by increasing strength of protection. **Anti-learnability (unlearnability)** perturbs data before release so that standard training pipelines fail to converge (Ye & Wang, 2024; Wang et al., 2025), preserving human perception but offering protection only against training misuse. **Ungeneralizable training** alters objectives or weights to suppress transfer in specific domains (Wang et al., 2022; Hong et al., 2025), but this requires retraining with custom losses and control of the pipeline, confining protection to modified models. **Fully homomorphic encryption (FHE)** (Gentry, 2009) guarantees maximal confidentiality by enabling inference on encrypted data, but its extreme computational and memory costs

make it impractical for routine applications such as online media, healthcare workflows, or MLaaS services (Ribeiro et al., 2015).

The three aforementioned approaches illustrate a spectrum of trade-offs: anti-learnability is lightweight but ineffective at inference, ungeneralizable training provides targeted suppression but relies on retraining, and FHE ensures the strongest confidentiality but remains prohibitively expensive in practice. These limitations highlight the need for methods that act directly at inference, preserving authorized utility while preventing post-release misuse, without imposing heavy retraining or cryptographic cost.

Our Work. We tackle the practical reality that, once data leaves its owner, it may be consumed by countless models. Our goal is to transform the data into a cipher that only a single authorized model  $f^*$  can interpret. We present a lightweight procedure that strategically recodes the data so it remains fully usable for  $f^*$  while withholding utility from any unauthorized models. This mechanism has broad applicability across online services such as machine learning as a service (MLaaS) (Ribeiro et al., 2015), where uploaded data is often copied, cached, and reused beyond its original purpose (Gal et al., 2022; Liu et al., 2024a).

We propose <u>non-transferable examples</u> (NEs), a data-side construction that enforces <u>model-specific</u> data usability and requires neither retraining nor access to non-target models. NEs leverage a structural property of neural networks in which many input directions have negligible effect on early features, yielding a model-specific set of <u>insensitivity directions</u> that rarely align across models. With a small probe budget, we estimate the spectral basis for the  $f^*$ 's insensitivity subspace of the authorized model, and add a calibrated perturbation confined to this basis, preserving the target's predictions while substantially reducing utility on unauthorized models.

We establish a formal foundation for NEs by deriving crucial bounds that provide strong theoretical guarantees: recoding within the identified insensitivity subspace preserves  $f^*$ 's outputs within a quantified tolerance for authorized utility. For authorized-utility retention, we bound the distance between outputs computed with the same first-layer weight matrix W via matrix perturbation theory, using random matrix and vector norm inequalities (Vershynin, 2018). For unauthorized usage, we prove a cross-model deviation bound using the Hoffman-Wielandt spectral inequality (Bhatia & Elsner, 1994), linking non-target degradation to differences in the singular spectrum and to misalignment between the corresponding feature subspaces.

We conduct a comprehensive empirical evaluation to demonstrate generality across tasks and model architectures. For image classification, we cover representative families including ResNet, ViT, SwinV2, DeiT, and MambaVision; recoded data remain usable only on the authorized target, while models with different weights collapse to effectively unusable performance (e.g., even under an extreme distortion of 0 dB PSNR on ImageNet, the top-1 accuracy of the authorized ResNet-50 changes negligibly from 80.3% to 80.1%, whereas all other models drop to  $\approx 0.1\%$ ). We also evaluate the vision-language models Qwen2.5-VL and InternVL3 on the comprehensive benchmark MMBench (Liu et al., 2024b), which spans mathematical reasoning, chart/table understanding, document QA, and OCR, demonstrating applicability beyond classification to multimodal data. We further test NEs' resistance to common preprocessing and reconstruction attacks. Across all settings, NEs preserve authorized performance while rendering non-target systems effectively unusable.

**Contributions**. We summarize our main contributions as follows:

- A new problem setting. We formulate *model-specific data authorization*: user-provided data should remain fully usable for an authorized model while withholding utility from any other models. We introduce *non-transferable examples* (NEs), a lightweight, training-free, data-agnostic spectral recoding method.
- A formal framework and theoretical analysis. We establish a strong formal foundation for NE, providing crucial bounds that guarantee efficacy and provide an independent, model-level confirmation of provable data-authorization guarantees.
- An empirical evaluation. Across diverse model architectures and data modalities, NEs consistently confine utility to the authorized model while non-targets fail, and they remain robust under routine preprocessing and resistant to reconstruction attempts, supporting practical deployment in real-world settings.

# 2 PROBLEM FORMULATION

We consider a supervised task with ambient input space  $\mathcal X$  and output space  $\mathcal Y$ . A dataset  $\mathcal D$  induces the task domains  $\mathcal X_{\mathcal D} \subset \mathcal X$  and  $\mathcal Y_{\mathcal D} \subset \mathcal Y$  observed in practice. Training on  $\mathcal D$  yields a family of models  $\mathcal F_{\mathcal D}$ , where each  $f \in \mathcal F_{\mathcal D}$  implements a mapping  $f: \mathcal X_{\mathcal D} \to \mathcal Y_{\mathcal D}$ . To quantify the *usability* of f on inputs  $x \in \mathcal X_{\mathcal D}$ , we adopt a performance metric  $m: \mathcal F_{\mathcal D} \times \mathcal X \to \mathbb R$  (e.g., accuracy indicator for classification, negative sequence error for OCR, log-likelihood for generative decoding) and evaluate  $\mathbb E_{x \in \mathcal X_{\mathcal D}}[m(f,x)]$ . We postpone the detailed neural-network parameterization and notation (architectures, layers, operators) to Appendix B.1, where we formalize the conventions used in subsequent sections. For clarity, we assume smaller values of  $m(\cdot)$  indicate better usability.

## 2.1 Model-specific Data Representation

At inference on new data outside the training set, the data provider (defender) seeks to release content (e.g., images, speech, documents) that is correctly processed by a designated target model  $f^*$  while remaining unusable for any non-target models  $f' \in \mathcal{F}_{\mathcal{D}} \setminus \{f^*\}$ , i.e.,

$$\mathbb{E}_{x \in \mathcal{X}_{\mathcal{D}}}[m(f', x)] - \mathbb{E}_{x \in \mathcal{X}_{\mathcal{D}}}[m(f^{\star}, x)] \gg 0$$

for non-authorized aims.

Accordingly, we formulate a *data usage-control* objective: preserve the usability of  $f^*$  while degrading that of any non-target model, without requiring retraining and without imposing assumptions on non-target models, to maintain generality. Formally, we allow an input recoding  $\tilde{x} = \mathcal{T}(x)$  with  $\mathcal{T}: \mathcal{X} \to \mathcal{X}$ . The resulting set  $\tilde{\mathcal{X}}_{\mathcal{D}} \subset \mathcal{X}$  is termed *non-transferable examples* (NEs).

We view NEs as a model-specific data representation: recoded inputs  $\tilde{x} = \mathcal{T}(x)$  are tailored to the designated target  $f^*$  so that authorized-utility retention holds for  $f^*$  and unauthorized-utility degradation holds for any  $f' \in \mathcal{F}_{\mathcal{D}} \setminus \{f^*\}$ . Formally, we state the following problem.

**Problem** (Model-specific Data Representation). Given an authorized-utility tolerance  $\rho \geq 0$  and a non-target separation margin  $\gamma > 0$ , a recoded data sample  $\tilde{x} = \mathcal{T}(x)$  (for fixed  $\mathcal{T}$ ) is model-specific to  $f^*$  if, for the designated  $f^*$  and any non-authorized f', it satisfies:

$$\begin{array}{ll} (\textit{authorized-utility retention}) & \mathbb{E}_{\tilde{x} \in \widetilde{\mathcal{X}}_{\mathcal{D}}}[m(f^{\star}, \tilde{x})] - \mathbb{E}_{x \in \mathcal{X}_{\mathcal{D}}}[m(f^{\star}, x)] \leq \rho, \\ (\textit{unauthorized-utility degradation}) & \mathbb{E}_{\tilde{x} \in \widetilde{\mathcal{X}}_{\mathcal{D}}}[m(f', \tilde{x})] - \mathbb{E}_{\tilde{x} \in \widetilde{\mathcal{X}}_{\mathcal{D}}}[m(f^{\star}, \tilde{x})] \geq \gamma. \end{array}$$

Before proceeding, we specify the threat model that this work takes into consideration.

## 2.2 THREAT MODEL

We consider a data producer (defender) who has white-box access to the authorized model  $f^*$  and a probe source from the task domain  $\mathcal{X}_{\mathcal{D}}$  (a clean test set  $\mathcal{D}$  or a sampler over  $\mathcal{X}_{\mathcal{D}}$ ). The defender may query internal features of  $f^*$  and deploy an input-side transformation  $\mathcal{T}: \mathcal{X} \to \mathcal{X}$ , releasing only recoded inputs  $\tilde{x} = \mathcal{T}(x)$ . The algorithmic form of  $\mathcal{T}$  may be public, while target-specific parameters (e.g., spectral basis, thresholds) remain private. The defender has no access to non-target models in  $\mathcal{F}_{\mathcal{D}} \setminus \{f^*\}$ .

Adversaries operate *unauthorized* models  $f' \in \mathcal{F}_{\mathcal{D}} \setminus \{f^*\}$  and receive only  $\tilde{x}$ . Before inference, they may optionally apply standard acquisition or preprocessing to obtain a preprocessed input  $\tilde{x}'$  from  $\tilde{x}$ . The exact parameters of  $\mathcal{T}$  are hidden. The adversary's objective is to *minimize* the task metric  $m(f', \tilde{x}')$  under fixed compute or query budgets.

Specifically, we consider three adversary classes:

- (i) General Adversary (GA): Any non-target f' with arbitrary architecture and parameters, given  $\tilde{x}$  and allowed to apply an input-side preprocessing operator A.
- (ii) Transfer-match Adversary (TA): A GA sharing the architecture of  $f^*$  but with different weights.
- (iii) Adaptive Adversary (AA): A GA/TA that optimizes over  $\mathcal{A} \in \mathbb{A}$  to reduce  $m(f', \mathcal{A}(\tilde{x}))$ .

**Defense Objective**. Design  $\mathcal{T}$  to achieve *model-specific data usability*, *i.e.*, to satisfy Formula (1) and Formula (2).

**Scope**. This setting is practical as an authorized third party may enforce such a process on behalf of the model owner: with agreed access to  $f^*$  (e.g., feature probes sufficient to instantiate  $\mathcal{T}$ ), it applies the protection server-side and releases only  $\tilde{x}$ . Neither the internals of  $f^*$  (architecture, weights) nor the target-specific parameters of  $\mathcal{T}$  are disclosed to end users (see Appendix A for evidence).

## 3 Our Method

Neural networks typically begin with a linear feature extractor (*e.g.*, a convolution with weight sharing or a token embedding), which reduces redundancy because input coordinates are often correlated (see Appendices B.2 and B.3). This motivates *input-side* perturbations that lie in an *insensitivity* subspace of the target model's first linear map: these directions are nearly inert for the target but, due to subspace misalignment across models, can induce nontrivial changes for non-targets.

## 3.1 Insensitivity Subspace Identification

Let W denote the first linear transformation (bias omitted) of the target model. In practice, W typically has a nontrivial nullspace  $\operatorname{Null}(W) = \{x \in \mathcal{X} \mid Wx = 0\}$ , owing to high input dimensionality (see Appendix B.4). For any  $x \in \mathcal{X}$  and perturbation  $\delta \in \operatorname{Null}(W)$ , we have  $W(x + \delta) = Wx$ , so downstream computation receives identical features, *i.e.*,  $f(x + \delta) = f(x)$ .

To generalize the construction, we relax exact nulling and consider a low-sensitivity subspace that permits controlled feature deviation, i.e., require  $W\delta\approx 0$  rather than  $W\delta=0$ . Perturbations confined to such an insensitivity subspace induce only small changes beyond the first layer, especially given intervening nonlinearities (e.g., ReLU truncation, sigmoid/tanh saturation). To identify such directions for a given  $W\in\mathbb{R}^{m\times n}$ , we consider its singular value decomposition (SVD)  $W=U\Sigma V^{\top}$ : the nullspace is spanned by right singular vectors with zero singular values, and by extension we define the  $\tau$ -insensitive subspace as the span of right singular vectors whose singular values are at most a threshold  $\tau>0$ . We formalize it as follows.

**Definition 1** ( $\tau$ -insensitive Subspace). Let  $W \in \mathbb{R}^{m \times n}$  have SVD,  $W = USV^{\top}$ , with singular values ordered  $s_1 \leq s_2 \leq \cdots \leq s_n$ . Given a spectral threshold  $\tau > 0$ , the  $\tau$ -insensitive subspace is

$$\operatorname{Ins}_{\tau}(W) = \operatorname{span}\{(v_1, v_2, \dots, v_k) \mid \sum_{i=1}^k s_i \le \tau, \ W = USV^{\top}\},\tag{3}$$

where  $v_i$  denotes the i-th column of V corresponding to the i-th singular value  $s_i$  in S.

This construction naturally captures the nullspace, and  $\operatorname{Ins}_{\tau}(W) \supseteq \operatorname{Null}(W)$  for any  $\tau \geq 0$ .

#### 3.2 Non-transferable Examples

Inheriting Definition 1, and given a  $\tau$ -insensitive subspace  $\operatorname{Ins}_{\tau}(W)$ , we first sample a vector  $z \in \mathbb{R}^n$  (e.g., i.i.d. Gaussian, structured pattern, or content-dependent code) and zero out the coordinates aligned with singular values exceeding  $\tau$  (i.e., directions sensitive to the target model). Then, we project z onto  $\operatorname{Ins}_{\tau}(W)$  to obtain a perturbation  $\delta$  in input space by  $\delta = Vz$ . Following that, the perturbation  $\delta$  is added to the original input x to obtain a recoded input  $\tilde{x} = x + \delta$ , which forms an NE. Note that V is an orthogonal matrix such that  $V^{\top}V = V^{-1}V = I$ . Intuitively, we have  $W\delta = WVz = USV^{\top}Vz = USz$ , where Sz has small entries since z only has non-zero elements on coordinates with small singular values. By the distributive law of matrix multiplication,  $W\tilde{x} = W(x + \delta) = Wx + W\delta$ , where  $W\delta$  is small and thus  $W\tilde{x}$  is close to Wx.

**Practical Implementation**. Our procedure is model-agnostic and applies across architectures. Fully connected fronts can directly use their first weight matrix as W. For convolutional fronts, we apply the construction to the linearized operator of the first convolution (e.g., nn.unfold) to obtain W; after synthesis, the perturbation is folded back to the native input layout. For Transformer-style models (e.g., BERT (Devlin et al., 2019)), we take W to be the input projection of the first multi-head self-attention block *after* the embedding layer (e.g., the QKV input projection or its concatenated form) and apply the same spectral construction. In this work, we instantiate and evaluate the approach on convolutional and Transformer architectures. A theoretical discussion of the equivalence and generalizability of these instantiations is provided in Appendices B.5 and B.6. The generation

of the vector z can be stochastic (e.g., seeded per instance) or deterministic (e.g., fixed codes per class or per client). We use  $\delta \leftarrow \lambda \cdot \delta / \max\{1, \|\delta\|_2\}$  to ensure  $\|\delta\|_2 \leq 1$  and then introduce a scaling factor  $\lambda > 0$  to scale its amplitude. The parameters  $\tau$  and  $\lambda$  govern the trade-off between authorized-utility retention and unauthorized-utility degradation: larger  $\tau$  tends to increase more perturbed input dimensions, while larger  $\lambda$  increases the perturbation magnitude. In practice, the permissible insensitive space is deliberately loose, offering substantial flexibility in synthesis, as empirically explored in Section 5.

# 4 THEORETICAL ANALYSIS

 This section provides a formal foundation for non-transferable examples (NEs), establishing their *model-specific* property as defined earlier, namely *authorized-utility retention* (Section 4.1) and *unauthorized-utility degradation* (Section 4.2). Our analysis focuses on the first linear transformation of the network and treats subsequent layers implicitly, consistent with many prior works of perturbation propagation and local robustness (Huang et al., 2021; Wang et al., 2024; Qian & Klabjan, 2021; Wu et al., 2021).

## 4.1 AUTHORIZED-UTILITY RETENTION

We first quantify the perturbation effect on the target (authorized) model, characterizing how the recoded input  $\tilde{x} = x + \delta$  alters its first-layer features relative to the original input x. Throughout the analysis,  $\|\cdot\|_F$  and  $\|\cdot\|_2$  denote the Frobenius and Euclidean norms, respectively.

**Theorem 1** (Bounding Authorized Utility).  $W \in \mathbb{R}^{m \times n}$  represents a linear transformation as the first operation in a neural network, and its SVD is  $W = USV^{\top}$ . The perturbation  $\delta = Vz$  in  $\operatorname{Ins}_{\tau}(W)$  is generated following Section 3.  $z \in \mathbb{R}^n$  is a random vector, each entry of which is i.i.d. by following a standard normal distribution  $\mathcal{N}(0,\sigma)$ . Let  $\tilde{x} = x + \delta$  be the recoded input, the following bound holds,

$$||W\tilde{x} - Wx||_2 < \tau \cdot \sqrt{k\sigma^2 + t},\tag{4}$$

with high probability of at least  $1 - 2k\sigma^4/t^2$ , where t > 0 is a small positive number.

*Proof.* First, we have  $\tilde{x} = x + \delta = x + Vz$ . Then, we get  $W\tilde{x} - Wx = W\delta = USV^{\top}Vz$ . Next, we have  $USV^{\top}Vz = USz$ , considering  $V^{\top}V$  is an identity matrix because V is orthogonal.

Let  $U = [u_1, u_2, \dots, u_m]$  and consider each  $s_q$   $(q \le k)$  satisfies  $s_q \le \tau$ , we have

$$||W\tilde{x} - Wx||_F = ||US^*z||_F = \sqrt{\sum_{p=1}^m (\sum_{q=1}^k u_{p,q} s_q z_q)^2} \le \tau \sqrt{\sum_{p=1}^m (\sum_{q=1}^k u_{p,q} z_q)^2} = \tau \cdot ||Uz||_F,$$

where  $u_{p,q}$  is the (p,q)-th entry of the matrix U,  $s_q$  is the q-th singular value in S, and  $z_q$  is the q-th entry of the matrix z.

Considering that U is orthogonal satisfying  $U^{T}U$  is an identity matrix, we have

$$||Uz||_F^2 = \operatorname{trace}((Uz)^\top (Uz)) = \operatorname{trace}(z^\top U^\top Uz) = \operatorname{trace}(z^\top z) = ||z||_F^2 = ||z||_2^2.$$

Thus, we have  $||W\tilde{x} - Wx||_F \le \tau \cdot ||z||_2$ .

Since z is a random vector following the standard normal distribution  $\mathcal{N}(0,\sigma)$ , i.e., each entry of z is independently sampled from the standard normal distribution, i.e.,  $z_q \sim \mathcal{N}(0,\sigma)$ , we have  $z_q^2$  follows the Chi-squared distribution with 1 degree of freedom, i.e.,  $z_q^2 \sim \chi^2(1)$ , with  $\mathbb{E}[z_q^2] = \sigma^2$  and  $\mathrm{Var}(z_q^2) = 2\sigma^4$ . Next, we have  $\mathbb{E}[\|z\|_2^2] = \sum_{q=1}^k \mathbb{E}[z_q^2] = k\sigma^2$  and  $\mathrm{Var}(\|z\|_2^2) = \sum_{q=1}^k \mathrm{Var}(z_q^2) = 2k\sigma^4$ . By Chebyshev's inequality, we can bound the probability of the perturbation effect as

$$\mathbb{P}\{\|z\|_2^2 \ge k\sigma^2 + t\} \le \text{Var}(\|z\|_2^2)/t^2 = 2k\sigma^4/t^2,$$

where t>0 is a small positive number. Therefore, we can conclude that the perturbation effect is bounded by  $\tau \cdot \|z\|_2$  with high probability, i.e., considering  $\|W\tilde{x} - Wx\|_2 \leq \|W\tilde{x} - Wx\|_F$ ,

$$\mathbb{P}\{\|W\tilde{x} - Wx\|_2 < \tau \cdot \sqrt{k\sigma^2 + t}\} > 1 - 2k\sigma^4/t^2.$$

Theorem 1 clarifies that the perturbation effect can be controlled in the first linear transformation of a neural network by choosing a small  $\tau$  (determining k) and  $\sigma$ . Considering the generalization property of neural networks, they can commonly handle small perturbations (Novak et al., 2018). On the other hand, considering some special activation functions, such as zeroing negative values in ReLU, saturating in sigmoid and tanh, the perturbation effect is even smaller in practice because the inactive neurons yield a zero output with negative inputs. This supports  $|f^*(\tilde{x}) - f^*(x)|$  is small and further guarantees Formula (1) in Section 2.

## 4.2 UNAUTHORIZED-UTILITY DEGRADATION

We then analyze the difference bound between the outputs of the first linear transformation in two models with different initialization or training. This shows a difference from the view that the perturbation based on the insensitivity subspace of one model can significantly affect the output of the other model, because their first layer weight matrices have different SVDs. Formally, we consider two models with the first layer weight matrices  $W_1$  and  $W_2$ , respectively, shown in the following theorem, which is based on the Hoffman-Wielandt theorem (Bhatia & Elsner, 1994).

**Theorem 2** (Bounding Unauthorized Utility). Given two models with the same-sized first-layer weight matrices  $W_1, W_2 \in \mathbb{R}^{m \times n}$  trained on the same dataset, their SVDs are  $W_1 = U_1 S_1 V_1^{\top}$  and  $W_2 = U_2 S_2 V_2^{\top}$ . Let  $V_1 = [v_{1,1}, v_{1,2}, \ldots, v_{1,n}]$  and  $V_2 = [v_{2,1}, v_{2,2}, \ldots, v_{2,n}]$  be the right singular vector matrices of  $W_1$  and  $W_2$ . The perturbation is  $\delta = Vz$ ,  $\delta \in \operatorname{Ins}_{\tau}(W)$ , generated following Section 3.  $z \in \mathbb{R}^n$  is a random vector, each entry of which is i.i.d. by following a standard normal distribution  $\mathcal{N}(0,\sigma)$ . Let  $\tilde{x} = x + \delta$  be the recoded input. Given any  $\sigma_{1,i} \leq \tau$   $(1 \leq i \leq k)$  by Definition 1, the following bound holds,

$$\|(\sigma_{1,i}v_{1,i} - \sigma_{2,i}v_{2,i})^{\top}\tilde{x}\|_{2} \le \|\tilde{x}\|_{2}(\tau\|W_{1} - W_{2}\|_{2}/\varepsilon + \varepsilon),\tag{5}$$

where  $\varepsilon = \min(|\sigma_{1,i} - \sigma_{2,j}| \mid \forall i, j = 1, 2, ..., n)$  is the minimum gap between the *i*-th singular value of  $W_1$  and the singular values of  $W_2$ .

*Proof.* First, we have

$$\begin{split} & \|(\sigma_{1,i}v_{1,i} - \sigma_{2,i}v_{2,i})^{\top}\tilde{x}\|_{2} \\ \leq & \|(\sigma_{1,i}v_{1,i} - \sigma_{1,i}v_{2,i})^{\top}\tilde{x} + (\sigma_{1,i}v_{2,i} - \sigma_{2,i}v_{2,i})^{\top}\tilde{x}\|_{2} \\ = & \|\sigma_{1,i}(v_{1,i} - v_{2,i})^{\top}\tilde{x}\|_{2} + \|(\sigma_{1,i} - \sigma_{2,i})(v_{2,i})^{\top}\tilde{x}\|_{2} \\ \leq & |\sigma_{1,i}|\|v_{1,i} - v_{2,i}\|_{2}\|\tilde{x}\|_{2} + |\sigma_{1,i} - \sigma_{2,i}|\|v_{2,i}\|_{2}\|\tilde{x}\|_{2} \end{split}$$

Considering that  $||v_{2,i}||_2 = 1$  because  $V_2$  is orthogonal and  $\sigma_{1,i} \leq \tau$  by Definition 1, we have

$$\|(\sigma_{1,i}v_{1,i} - \sigma_{2,i}v_{2,i})^{\top}\tilde{x}\|_{2} \le \|\tilde{x}\|_{2}(\tau\|v_{1,i} - v_{2,i}\|_{2} + |\sigma_{1,i} - \sigma_{2,i}|).$$

By the Hoffman-Wielandt theorem (Bhatia & Elsner, 1994), we have

$$||v_{1,i} - v_{2,i}||_2 \le ||W_1 - W_2||_2/\varepsilon,$$

where  $\varepsilon=\min(|\sigma_{1,i}-\sigma_{2,j}|\mid \forall i,j=1,2,\ldots,n)$  is the minimum gap between the i-th singular value of  $W_1$  and the adjacent singular values of  $W_2$ ,  $v_{1,i}$  and  $v_{2,i}$  are the i-th right singular vectors of  $W_1$  and  $W_2$ , and  $s_{1,i}$  and  $s_{2,j}$  are the i-th and j-th singular values of  $W_1$  and  $W_2$ , accordingly. Thus, we have, considering  $|\sigma_{1,i}-\sigma_{2,i}|\leq \varepsilon$  because the singular values are in descending order,

$$\|(\sigma_{1,i}v_{1,i} - \sigma_{2,i}v_{2,i})^{\top}\tilde{x}\|_{2} \le \|\tilde{x}\|_{2}(\tau\|W_{1} - W_{2}\|_{2}/\varepsilon + \varepsilon),$$

which concludes the proof.

Theorem 2 is strong enough by considering each single singular value. It clarifies that the perturbation effect of the insensitivity subspace of one model can significantly affect the output of the other model. Because Assumption 1 (see Appendix 1 for more discussion) supports that  $\delta$  is near zero with  $\tau$  as a positive number. Further, such two models obtain the first layer weight matrices from the same dataset based on Assumption 2 (see Appendix 2), which means that the right singular vectors of the first layer weight matrices are aligned with the principal components of the data representations, there exist multiple feasible solutions for the first layer weight matrix because of multiple feasible solutions for the principal components of the data representations due to the spectral flatness of the data representations. This provides an extra difference when multiplying the singular value of the first layer weight matrices, where the first model with  $W_1$  has a small singular value  $\sigma_{1,i}$  and a near-zero  $\sigma_{1,i}v_{1,i}^{\top}\tilde{x}$  while the second model with  $W_2$  has a different  $\sigma_{2,i}$  and a non-zero  $\sigma_{2,i}v_{2,i}^{\top}\tilde{x}$ . This supports that  $|f'(\tilde{x}) - f^*(\tilde{x})|$  is large and further yields Formula (2) in Section 2.

# 5 EXPERIMENTS

We empirically verify whether our non-transferable examples (NEs) preserve the authorized model's utility, retaining clean accuracy and intended task behavior, while rendering unauthorized models unable. The evaluation proceeds in three parts: (i) cross-model non-transferability covering both cross model architecture (GA in Section 2.2) transfer and same model architecture with different weights (TA); (ii) head-to-head comparisons with representative baselines under matched conditions; and (iii) real-world practicality, including generalization across modalities as well as robustness to common preprocessing pipelines and reconstruction attacks (AA).

**Experimental Setup.** We evaluate five widely used ImageNet-pretrained backbones that span classic to recent designs: ResNet-50 (He et al., 2016) (convolutional neural network), ViT-base-patch16-22 (Dosovitskiy et al., 2021) (vision transformer with patch-wise tokens), SwinV2-tiny-patch4-window8-256 (Liu et al., 2021) (hierarchical transformer with pyramid features common in real systems), DeiT-base-patch16-224 (Touvron et al., 2021) (dataefficient training of a ViT with distillation), and MambaVision-T-1K (Hatamizadeh & Kautz, 2025) (state-space sequence model adapted for vision). Experiments are conducted on CIFAR-10 and ImageNet-1K; for CIFAR-10, models are fine-tuned for 10 epochs from the ImageNet-1K checkpoints. "Baseline" entries in the tables denote clean accuracy under our evaluation pipeline.

We also evaluate leading-edge vision-language transformers <code>Qwen2.5-VL-3B-Instruct</code> (Bai et al., 2025) and <code>InternVL3-1B</code> (Zhu et al., 2025) on the comprehensive MMBench (Liu et al., 2024b), which covers mathematical reasoning, chart and table understanding, document question answering, and OCR, to mirror real-world usage, including cross-modality generalization and robustness to common preprocessing and reconstruction attacks.

## 5.1 Cross-Model Non-Transferability

**NE Construction**. To select a suitable perturbation strength that preserves the authorized model's accuracy while sharply reducing unauthorized accuracy, we randomly sample 512 images from ImageNet and generate NEs at multiple perturbation levels, recording top-1 accuracy as a function of strength. As a comparator, we use an unauthorized model given by ResNet-50 fine-tuned for 10 epochs on CIFAR-10, and measure how inputs recoded for different target models perform on this same unauthorized model. Results are shown in Figure 1.

Across all settings, unauthorized accuracy collapses to an unusable level by 25 to 20dB PSNR (peak-signal-to-noise ratio (Hore & Ziou, 2010), lower PSNR indicates stronger perturbation and higher perceptual distortion), while the authorized network experiences only a negligible drop. Several backbones remain stable even at <10dB, but we standardize on 20dB<sup>1</sup> for the rest of the experiments to use a conservative setting, as SwinV2-T shows a slight authorized drop below this point. We sample a recoding vector z with the same dimension as the input representation with *i.i.d.* Gaussian entries, then project z onto the  $\tau$ -insensitive directions with  $\tau = 10^{-4}$ . For each authorized target  $f^*$ , we report authorized accuracy on the recoded inputs  $\tilde{x} = \mathcal{T}(x)$  and the accuracies of unauthorized models  $f' \neq f^*$  on the same  $\tilde{x}$ .

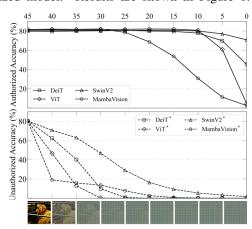


Figure 1: Authorized vs. unauthorized accuracy on target-recoded inputs across perturbation strength. Visual examples are in Figure 4.

# 5.2 BASELINE COMPARISON

Table 1 reports a  $5\times5$  cross-architecture matrix on CIFAR-10 and ImageNet. At 20dB PSNR, NEs keep authorized performance close to clean (e.g., from 98.8% to 98.7% on CIFAR-10; from 80.3% to 80.2% on ImageNet), while unauthorized models collapse to chance-level utility (off-diagonals around 5.5–20.6% on CIFAR-10 and 0.0–12.9% on ImageNet). SwinV2 exhibits a slightly larger

<sup>&</sup>lt;sup>1</sup>We parameterize perturbation strength by PSNR for interpretability and fair comparison.

authorized drop on ImageNet (from 80.9% to 71.7%), which we attribute to sensitivity in its patch-merging pipeline; modest tuning of basis selection fixes this in practice. Crucially, in all cases, the off-diagonal entries remain at unusable accuracy, demonstrating strong architecture-specific non-transferability. Holding the architecture fixed but changing the weights, NE constructed with one weight set does not transfer to the same architecture with a different weight set. Along the diagonal of Table 2, unauthorized accuracy stays at a completely unusable level on both datasets for all backbones. Off-diagonal entries (cross-architecture) are shown in grey for completeness and mirror the behavior in Table 1. These results underscore strong model specificity: non-transferability holds across different architectures and across weight variants of the same architecture.

Table 1: Cross-model non-transferability on CIFAR-10 and ImageNet. Rows denote the target (authorized) model used to generate NEs; columns denote the evaluated model on the same recoded inputs. Green diagonal entries mark authorized accuracy; off-diagonals show unauthorized utility.

			CIFAR	10		ImageNet					
	ResNet-50	ViT-B	SwinV2-T	DeiT-B	MambaVision-T	ResNet-50	ViT-B	SwinV2-T	DeiT-B	MambaVision-T	
Baseline	98.2%	98.8%	96.1%	96.1%	96.9%	80.3%	81.6%	80.9%	79.9%	82.4%	
ResNet-50	<b>97.7</b> %	12.5%	13.6%	9.4%	10.0%	80.2%	0.0%	0.1%	0.1%	0.1%	
ViT-B	10.5%	98.7%	9.3%	12.1%	9.7%	0.0%	81.3%	0.0%	0.0%	0.0%	
SwinV2-T	15.6%	11.7%	88.4%	20.6%	18.8%	9.1%	4.3%	71.7%	7.4%	12.9%	
DeiT-B	9.4%	9.8%	9.8%	96.1%	5.5%	0.0%	0.0%	0.0%	79.3%	1.0%	
MambaVision-T	17.6%	7.0%	13.7%	11.3%	94.5%	5.8%	0.0%	1.5%	0.7%	81.0%	

Table 2: Model-specific non-transferability. Diagonal entries compare the same model architecture with different weights; shaded off-diagonals are cross-architecture and included for completeness.

			CIFAR1	0					Imagel	Net	
	ResNet-50	ViT-B	SwinV2-T	DeiT-B	MambaVision-T		ResNet-50	ViT-B	SwinV2-T	DeiT-B	MambaVision-T
ResNet-50	13.3%	9.4%	8.2%	9.4%	7.8%	0	1.2%	0.0%	4.4%	0.0%	0.0%
Vill-B	10.1%	9.6%	10.1%	11.5%	9.8%	$\Xi$	0.0%	0.0%	0.0%	0.0%	0.0%
SwinV2-T	10.5%	10.2%	21.0%	11.3%	14.5%	A.	0.0%	0.0%	0.0%	0.0%	0.0%
DeiT-B	12.5%	9.3%	7.0%	14.8%	10.9%	$\Xi$	0.0%	0.0%	0.0%	0.0%	0.0%
MambaVision-T <sup>□</sup>	14.5%	5.1%	12.5%	9.8%	8.2%	0	0.0%	0.0%	1.9%	0.0%	0.0%

Table 3: Comparison with other authentication methods.

			CIFA	R-10				Imag	geNet	
	Plain	DP	FHE <sup>3</sup>	ALGOSPEC	NE (Ours)	Plain	DP	FHE <sup>3</sup>	ALGOSPEC	NE (Ours)
ResNet-50 ViT-B	98.2% 98.8%	59.8% <sup>1</sup>	87.8% <sup>3</sup>	6.4% 10.0%	97.7% 98.7%	80.3% 81.6%	63.1% <sup>1</sup>	_	0.1% 0.0%	80.2% 81.3%
Protection	Х	<b>X</b> <sup>2</sup>	<b>√</b>	<b>X</b> <sup>2</sup>	<b>√</b>	Х	<b>X</b> <sup>2</sup>	<b>√</b>	<b>X</b> <sup>2</sup>	<b>√</b>

<sup>&</sup>lt;sup>1</sup> DP struggles with batch norm and does not support multi-head attention in Transformers.

We compare the NE with authorization-oriented baselines that restrict model use by altering training, encrypting inference, or constraining the computation itself. Specifically, we include Differential Privacy (DP) (Dwork, 2006), which injects calibrated noise during training to limit extractable information; Fully Homomorphic Encryption (FHE), which executes inference over encrypted inputs and weights to protect access without changing model behavior; and ALGOSPEC (Liu et al., 2024a), a specification-style approach that replaces nonlinear components with low-degree polynomial surrogates so the end-to-end pipeline conforms to a prescribed algorithmic specification intended to gate unauthorized use. Results for ResNet-50 and ViT-B on CIFAR-10 and ImageNet appear in Table 3. For DP training, we follow the setting of Li et al. (2024) and implement it with IBM DiffPrivLib (Holohan et al., 2019). In our setup, authorized accuracy drops markedly, with more than 30% loss on CIFAR-10 and around 20% on ImageNet for ResNet-50, largely due to interactions with batch normalization, and our DP pipeline does not support transformer variants, which limits applicability in this setting. For FHE, we adopt a CKKS encrypted-inference configuration following TenSEAL (Benaissa et al., 2021); However, running it at our model and dataset scale was computationally infeasible: processing a single image took more than 30 minutes. We therefore cite published ResNet-20 CIFAR-10 results (Meftah et al., 2021), which preserve clean authorized accuracy but incur heavy computational cost, highlighting the practicality gap of FHE at scale. For ALGOSPEC, polynomial approximation of modern deep networks accumulates approximation error with depth and width, which in our experiments drives authorized accuracy toward random guess on both datasets. In sharp contrast, NE is a lightweight input-side recoding tied to the target model that adds negligible inference overhead, and under matched conditions preserves authorized accuracy while driving mean unauthorized accuracy to chance on both datasets.

<sup>&</sup>lt;sup>2</sup> The authorized model performance is significantly impacted.

<sup>&</sup>lt;sup>3</sup> Due to too long execution time, we only provide data of accuracy that we can find in public papers.

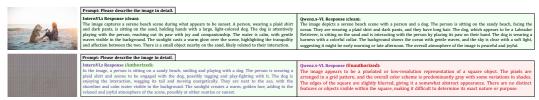


Figure 2: Illustrative visualization of effective on data authorization on VLM.

## 5.3 REAL-WORLD PRACTICALITY

Beyond standard backbones, we extend NE to state-of-the-art vision-language models. We evaluate InternVL3 (authorized) and Qwen2.5-VL (unauthorized) on the comprehensive benchmark MMBench (Liu et al., 2024b), across capability dimensions AR, CP, FP-C, FP-S, LR, and RR; see Table 4. The authorized model remains essentially unchanged, while the unauthorized model is consistently low and remains unusable across settings. An illustrative example is shown in Figure 2 (see more in Appendix 7), recoded inputs preserve authorized performance and suppress unauthorized utility, where the unauthorized model sees it as completely random noise pixels.

Table 4: VLMs on MMBench.

		In	ternVL3	3-1B (au	thorized	d)		Qwe	n2.5-VI	3B (ur	authoria	zed)		
	Overall	AR	CP	FP-C	FP-S	LR	RR	Overall	AR	CP	FP-C	FP-S	LR	RR
Baseline NE (Ours)								78.8% 18.3%						

While we have shown that NE withstands common preprocessing (resize, center and random crop, JPEG compression, mild blur) and reconstruction attacks (*e.g.*, classical denoising, SR-ResNet) throughout the above experiments, the rationale is straightforward: for the authorized model, the recoding lies in low-sensitivity directions and is effectively neutralized after the first layer, *regardless* of such preprocessing. Notably, preprocessing in the above VLM settings is more complicated (refer to Appendix D.1), yet we have likewise demonstrated robustness under these pipelines. We further examine super-resolution-based reconstruction attacks on VDSR (Vedaldi & Lenc, 2015) using SR-ResNet (Li et al., 2018) under two attacker settings; implementation details are in Appendix D.2. We consider a black-box setting trained on perturbed inputs and targets and a white-box setting trained with clean targets; in both cases, reconstructions are visually uninformative ( $\leq$ 0.6dB improvement) and fail to restore downstream utility, while authorized performance remains essentially unchanged.

## 6 RELATED WORK

**Training-time Defenses.** Anti-learnability perturbs released data so standard training fails while human perception is preserved (Ye & Wang, 2024; Wang et al., 2025). Non-transferable training modifies objectives or parameters to suppress transfer in designated domains (Wang et al., 2022; Hong et al., 2025). Both act during training and do not control inference once the content is public.

**Algorithmic Authorization.** This line of work binds data utility to a chosen algorithmic class. AlgoSpec applies polynomial approximation so that only a designated algorithm family recovers accuracy on transformed inputs (Liu et al., 2024a). In practice, it is only limited to simple classifiers such as Naive Bayes (Rish et al., 2001) and does not extend to neural networks.

**Differential Privacy and Encrypted Inference.** DP limits training-time leakage from individual examples into the learned model (Dwork et al., 2014), but it does not control who can run inference on public inputs. Fully homomorphic encryption enables encrypted inference with strong confidentiality (Gentry, 2009) but incurs substantial latency and memory overheads (Ribeiro et al., 2015; Meftah et al., 2021). We instead pursue lightweight, practical, model-specific authorization.

## 7 CONCLUSION

We presented non-transferable examples (NEs), a lightweight mechanism that preserves data utility for an authorized model while denying it to unauthorized ones. Our theory guarantees utility retention and quantifies degradation, and our experiments confirm robustness across diverse architectures and modalities. Together, these results show that NEs offer a practical path to model-level usage control, ensuring data serves its intended purpose without enabling misuse.

# ETHICS STATEMENT

This research does not involve human subjects, personally identifiable information, or sensitive datasets. Our method, non-transferable examples (NEs), is developed to mitigate unauthorized model use while preserving utility for intended applications. While usage-control mechanisms naturally intersect with broader discussions of openness and accessibility, our focus is on preventing misuse and supporting responsible AI practice. This work is intended to complement emerging regulatory and community standards for trustworthy AI.

#### REPRODUCIBILITY STATEMENT

We provide full details of our method, including theoretical foundations, algorithms, and hyperparameters. Experiments are run on standard datasets (CIFAR-10, ImageNet, MMBench) and widely used architectures (ResNet, ViT, Swin, DeiT, MambaVision, Qwen2.5-VL, InternVL3). Code, preprocessing scripts, and a demo showcasing non-transferable examples on representative models will be made available at: https://github.com/model-specific/non-transferable-examples.git.

# ACKNOWLEDGMENT OF LLM USE

The role of the LLM in this work was limited to polishing text for grammar and readability. The intellectual and technical contributions are solely those of the authors.

# REFERENCES

Anthropic, 2025. URL https://www.anthropic.com/.

Australian Productivity Commission. Harnessing data and digital technology: Interim report. https://jws.com.au/what-we-think/fresh-perspective-on-privacy-and-ai-law-reform-in-australian-productivity-commission-interim-data-report/, August 2025. Accessed on 21 August 2025.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

- Ayoub Benaissa, Bilal Retiat, Bogdan Cebere, and Alaa Eddine Belfedhal. Tenseal: A library for encrypted tensor operations using homomorphic encryption, 2021.
- Rajendra Bhatia and Ludwig Elsner. The hoffman-wielandt inequality in infinite dimensions. In *Proceedings of the Indian Academy of Sciences-Mathematical Sciences*, volume 104, pp. 483–494. Springer, 1994.
- Jeremy Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pp. 4171–4186, 2019.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *9th International Conference on Learning Representations (ICLR)*, 2021.
- Cynthia Dwork. Differential privacy. In 33rd International Colloquium on Automata, Languages and Programming (ICALP), pp. 1–12, 2006.

- Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. Foundations
   and Trends® in Theoretical Computer Science, 9(3–4):211–407, 2014.
- European Parliament and Council. Regulation (eu) 2024/1689 of the european parliament and of the council laying down harmonised rules on artificial intelligence (artificial intelligence act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024. OJ L 1689, 12 July 2024, in force 1 August 2024.
- Four Flynn, Helen King, Anca Dragan. Strengthening our Frontier Safety Framework. https://deepmind.google/discover/blog/strengthening-our-frontier-safety-framework/, 2025.
  - Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022. URL arXivpreprintarXiv:2208.01618.
  - Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2018.
  - Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM symposium on Theory of Computing (STOC)*, pp. 169–178, 2009.
  - Guy Hacohen and Daphna Weinshall. Principal components bias in over-parameterized linear models, and its manifestation in deep neural networks. *Journal of Machine Learning Research*, 23 (155):1–46, 2022. URL http://jmlr.org/papers/v23/21-0991.html.
  - Ali Hatamizadeh and Jan Kautz. Mambavision: A hybrid mamba-transformer vision backbone, 2025. URL https://arxiv.org/abs/2407.08083.
  - Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
  - Melissa Heikkilä. This artist is dominating ai-generated art, and he's not happy about it, 2022. URL https://www.technologyreview.com/2022/09/16/1059598/this-artist-is-dominating-ai-generated-art-and-hes-not-happy-about-it/.
  - Naoise Holohan, Stefano Braghin, Pól Mac Aonghusa, and Killian Levacher. Diffprivlib: the IBM differential privacy library. *ArXiv e-prints*, 1907.02444 [cs.CR], July 2019.
  - Ziming Hong, Yongli Xiang, and Tongliang Liu. Toward robust non-transferable learning: A survey and benchmark. In James Kwok (ed.), *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 10455–10463. International Joint Conferences on Artificial Intelligence Organization, 8 2025. doi: 10.24963/ijcai.2025/1161. URL https://doi.org/10.24963/ijcai.2025/1161. Survey Track.
  - Alex Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. 2010 20th International Conference on Pattern Recognition, pp. 2366–2369, 2010.
  - Yujia Huang, Huan Zhang, Yuanyuan Shi, J Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. *Advances in Neural Information Processing Systems*, 34:22745–22757, 2021.
- Juncheng Li, Faming Fang, Kangfu Mei, and Guixu Zhang. Multi-scale residual network for image
   super-resolution. In *Proceedings of the European conference on computer vision (ECCV)*, pp.
   517–532, 2018.
  - Qiushi Li, Yan Zhang, Ju Ren, Qi Li, and Yaoxue Zhang. You can use but cannot recognize: Preserving visual privacy in deep neural networks. 2024.

- Shuofeng Liu, Zihan Wang, Minhui Xue, Long Wang, Yuanchao Zhang, and Guangdong Bai. Being transparent is merely the beginning: Enforcing purpose limitation with polynomial approximation. In 33rd USENIX Security Symposium (USENIX Security 24), pp. 6507–6524, Philadelphia, PA, August 2024a. USENIX Association. ISBN 978-1-939133-44-1. URL https://www.usenix.org/conference/usenixsecurity24/presentation/liu-shuofeng.
  - Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player?, 2024b. URL https://arxiv.org/abs/2307.06281.
  - Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.
  - Souhail Meftah, Benjamin Hong Meng Tan, Chan Fook Mun, Khin Mi Mi Aung, Bharadwaj Veeravalli, and Vijay Chandrasekhar. Doren: Toward efficient deep convolutional neural networks with fully homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 16:3740–3752, 2021. doi: 10.1109/TIFS.2021.3090959.
  - Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv* preprint *arXiv*:1802.08760, 2018.
- Xin Qian and Diego Klabjan. A probabilistic approach to neural network pruning. In *International Conference on Machine Learning (ICML)*, pp. 8640–8649. PMLR, 2021.
- Mauro Ribeiro, Katarina Grolinger, and Miriam AM Capretz. Mlaas: Machine learning as a service. In *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 896–902. IEEE, 2015.
- Irina Rish et al. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pp. 41–46, 2001.
- Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In 2017 IEEE symposium on security and privacy (SP), pp. 3–18. IEEE, 2017.
- David Tan. Generative ai and copyright infringement. *Center for Technology, Robotics, Artificial Intelligence & the Law (TRAIL)*, January 2024. URL https://law.nus.edu.sg/trail/generative-ai-copyright-infringement/.
- The Guardian. Anthropic agrees to pay \$1.5 billion to settle author class action over AI training. https://www.theguardian.com/technology/2025/sep/05/anthropic-settlement-ai-book-lawsuit, 2025.
- The Hollywood Reporter. Artists Score Major Win in Copyright Case Against AI Art Generators. https://www.hollywoodreporter.com/business/business-news/artists-score-major-win-copyright-case-against-ai-art-generators-1235973601/, 2025.
- The White House. America's ai action plan. https://www.whitehouse.gov/articles/2025/07/white-house-unveils-americas-ai-action-plan/, July 2025. Released July 23, 2025 as mandated by EO 14179.
- Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.
- Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction APIs. In *USENIX Security Symposium*, 2016.
- A. Vedaldi and K. Lenc. Matconvnet convolutional neural networks for matlab. In *Proceeding of the ACM Int. Conf. on Multimedia*, 2015.

- Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.
  - Derui Wang, Minhui Xue, Bo Li, Seyit Camtepe, and Liming Zhu. Provably unlearnable data examples. In *The Network and Distributed System Security (NDSS) Symposium*, 2025.
  - Lixu Wang, Shichao Xu, Ruiqi Xu, Xiao Wang, and Qi Zhu. Non-transferable learning: A new approach for model ownership verification and applicability authorization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=tYRrOdSnVUy.
  - Zihan Wang, Zhongkui Ma, Xinguo Feng, Ruoxi Sun, Hu Wang, Minhui Xue, and Guangdong Bai. Corelocker: Neuron-level usage control. In 2024 IEEE Symposium on Security and Privacy (SP), pp. 2497–2514. IEEE Computer Society, 2024.
  - Boxi Wu, Jinghui Chen, Deng Cai, Xiaofei He, and Quanquan Gu. Do wider neural networks really help adversarial robustness? *Advances in Neural Information Processing Systems*, 34:7054–7067, 2021.
  - Jingwen Ye and Xinchao Wang. Ungeneralizable examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11944–11953, June 2024.
  - Dinghuai Zhang, Yang Song, Inderjit Dhillon, and Eric Xing. Defense against adversarial attacks using spectral regularization. In *International Conference on Learning Representations (ICLR)*, 2020.
  - Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025. URL https://arxiv.org/abs/2504.10479.

# APPENDIX

## A THREAT MODEL ASSUMPTIONS

Our approach assumes that the defender has white-box access to the authorized model's first-layer weights. This is reasonable in *server-side deployments*, where providers retain full control over the inference stack (*e.g.*, Google AutoML, AWS SageMaker, Microsoft Azure ML). Similar assumptions are standard in adversarial ML defenses such as randomized smoothing (Cohen et al., 2019) and spectral regularization (Zhang et al., 2020), which likewise rely on internal model access. This is also aligned with regulatory expectations (*e.g.*, EU AI Act, NIST AI RMF), where model owners are held responsible for ensuring compliance and thus are expected to enforce usage control internally.

We further assume that adversaries cannot reliably estimate the insensitivity subspace of the authorized model. Recovering such information would require either access to first-layer weights or large-scale probing with structured queries. Prior work on model extraction (Tramèr et al., 2016) and property inference (Ganju et al., 2018) demonstrates that approximating hidden representations typically requires millions of queries, which is unrealistic under commercial API rate limits and cost constraints. Moreover, because our perturbations lie in directions of negligible sensitivity for the authorized model, they are indistinguishable from natural input variance to unauthorized models, making subspace approximation even harder in practice.

## **B** SUPPLEMENTARY PRELIMINARIES

#### **B.1** NEURAL NETWORKS

Neural networks are a class of machine learning models inspired by the structure and function of the human brain. They consist of interconnected layers of neurons (nodes), which process input data to produce an output. Each neuron applies a mathematical function to its inputs, typically involving weights and biases, along with an activation function that is adjusted during training to minimize the difference between the predicted and actual outputs. A neural network is typically structured as a series of stacked linear transformations, followed by nonlinear activation functions. Formally, a neural network can be represented as a function  $f: \mathbb{R}^n \to \mathbb{R}^m$ , where n is the number of input features and m is the number of output classes or values. Here, we take the feedback neural network as an example (e.g., fully connected or convolutional neural networks). Because convolutional operations are linear transformations that can be unfolded into fully connected layers, we focus on fully connected neural networks. Further, the bias terms  $b^{(i)}$  can be taken as part of the weight matrix  $W^{(i)}$  of the i-th layer by appending a constant input of 1 to the input vector x, so we can simplify and rewrite the notation for theoretical convenience as

$$y = f(x) = \phi(W^{(n)} \cdot \phi(W^{(n-1)} \cdot \cdots \cdot \phi(W^{(2)} \cdot \phi(W^{(1)}x)) \cdots)),$$

where  $\phi$  is the activation function, which introduces non-linearity into the model. Common activation functions include the rectified linear unit (ReLU), sigmoid, and hyperbolic tangent (tanh).

## B.2 EIGENDECOMPOSITION AND PRINCIPAL COMPONENT ANALYSIS

A neural network is trained on a dataset to learn the underlying patterns and relationships in the data. Principal component analysis (PCA) is a technique used to analyze the learned representations of data by transforming it into a new coordinate system. It performs the eigendecomposition of the data's covariance matrix to identify the directions (principal components) that maximize the variance in the data. It is a linear transformation that projects the data onto a lower-dimensional subspace defined by the principal components, which are determined by the eigen decomposition of the covariance matrix of the data. The eigen vectors of the covariance matrix represent the directions of maximum variance, while the eigenvalues indicate the amount of variance along those directions.

Before giving the formal definition of PCA, we first introduce the classical eigendecomposition of a real-valued matrix without proof for simplicity.

**Definition 2** (Eigendecomposition). The eigendecomposition of a square matrix  $C \in \mathbb{R}^{n \times n}$  is a factorization of the form  $C = V\Lambda V^{\top}$ , where  $V \in \mathbb{R}^{n \times n}$  is a orthogonal matrix whose columns

are the eigen vectors of C, and  $\Lambda \in \mathbb{R}^{n \times n}$  is a diagonal matrix whose diagonal entries are the eigenvalues of C. The eigenvalues are the scalars  $\lambda_i$  such that  $Cv_i = \lambda_i v_i$ , where  $v_i$  is the i-th eigen vector of C.

In this work, the eigenvalues are arranged in ascending order along the diagonal of  $\Lambda$  by default. Geometrically,  $V^{\top}$  represents the rotation of the coordinate system,  $\Lambda$  scales the axes of the new coordinate system according to the eigenvalues, and V rotates the data back to the original coordinate system. Formally, given a dataset represented as a matrix  $x \in \mathbb{R}^n$ , where n is the input dimension, PCA can be performed as follows.

**Definition 3** (Principal Component Analysis (PCA)). The covariance matrix is  $C = \mathbb{E}[x^{\top}x]$ , where  $\mathbb{E}$  denotes the expectation operator. The eigen decomposition of the covariance matrix is given by  $C = V_{pca}\Lambda V_{pca}^{\top}$ , where  $V_{pca}$  is the matrix of eigen vectors (column vectors) and  $\Lambda$  is the diagonal matrix of eigenvalues. The principal components are the columns of the matrix  $V_{pca}$ , and the projection of the data onto the principal components is given by  $V_{pca}^{\top}x$ .

PCA is widely used for dimensionality reduction by selecting the top k principal components, where k is the desired number of dimensions. However, here we focus on analyzing the learned representations of the data using all principal components.

**PCA** in neural networks. PCA can be applied as a valuable tool to analyze the learned representations of data in neural networks. It describes the patterns and relationships of data input dimensions in the learned representations, *e.g.*, several input dimensions are enough to represent the data for a specific classification task.

## **B.3** SINGULAR VALUE DECOMPOSITION

Since the trainable parameters of a neural network comprise the weight matrices and biases, we introduce the singular value decomposition (SVD) to decompose these matrices for further analysis. Considering the merged weight matrix in the previous Appendix B.1, we focus on the weight matrix W without the bias terms for simplicity. The SVD of a matrix is defined as the following lemma, where we only consider the case of real-valued matrices and only provide this typical result without proof for simplicity.

**Lemma 1** (Singular Value Decomposition (SVD)). For any matrix  $W \in \mathbb{R}^{m \times n}$ , there exist orthogonal matrices  $U \in \mathbb{R}^{m \times m}$  and  $V \in \mathbb{R}^{n \times n}$ , and a diagonal matrix  $\Sigma \in \mathbb{R}^{m \times n}$  such that  $W = U\Sigma V^{\top}$ . The diagonal entries of  $\Sigma$  are the singular values of W, and the columns of U and V are the left and right singular vectors, respectively.

The singular values are non-negative and arranged in ascending order along the diagonal of  $\Sigma$ . Geometrically, the SVD decomposes the matrix W into three components: U represents the rotation of the input space,  $\Sigma$  scales the axes according to the singular values, and  $V^\top$  represents the rotation of the output space. The diagonal entries of  $\Sigma$  are the singular values, which indicate the importance of each corresponding singular vector in the decomposition. The smaller the singular value, the less important the corresponding singular vector is in representing the original matrix.

**SVD** in **neural networks**. In neural networks, applying SVD to a layer's weight matrix reveals how inputs are prioritized. The singular vectors indicate influential directions in the input space, and the singular values quantify their relative strength. For the first layer, this analysis is closely related to PCA of the input data: dominant data components often align with the layer's most responsive input directions (up to whitening and scaling). Because the first layer extracts features from raw inputs, SVD offers a clear view of how input dimensions are transformed into learned representations.

## B.4 NULLSPACE

When we take a matrix as a linear transformation, the null space of a matrix refers to the set of vectors that are mapped to the zero vector by that matrix. Formally, we have the following definition.

**Definition 4** (Null Space). The null space of a matrix  $W \in \mathbb{R}^{m \times n}$ , denoted as Null(W), is defined as  $\text{Null}(W) = \{x \in \mathbb{R}^n \mid Wx = 0\}$ .

The null space of a matrix  $W \in \mathbb{R}^{m \times n}$  is the subspace of  $\mathbb{R}^n$  containing all vectors that W maps to the zero vector. Its dimension is the *nullity* of W, and by the rank–nullity theorem the sum of the rank and nullity equals n, the number of columns.

**Nullspace in neural networks**. For a linear layer with weight matrix W, the null space consists of input directions that produce no change at that layer's output (*i.e.*, directions the layer is effectively insensitive to). In a multilayer network, such directions are suppressed before subsequent processing and thus have negligible downstream influence. This is especially intuitive in early vision layers, where convolutional filters emphasize specific spatial—frequency patterns; inputs orthogonal to those patterns lie (approximately) in low-response or null directions.

### **B.5** Convolution

This section clarifies that the convolution operation, which is commonly used in famous convolutional neural networks (CNNs), is a linear transformation that can be represented as a matrix multiplication. We first define the convolution as follows. We refer to a multi-dimensional matrix as a tensor, and omit commonly used arguments in the current code implementation, such as batch size, padding, stride, dilation, and group, for simplicity, as they can be easily extended by adding extra zero dimensions to the input tensor or kernel.

**Definition 5** (Convolution). The convolution takes inputs including an input tensor  $X \in \mathbb{R}^{c_1 \times h \times w}$  and a kernel (filter)  $K \in \mathbb{R}^{c_1 \times c_2 \times k_h \times k_w}$  with the kernel bias  $b \in \mathbb{R}^{c_2}$ , where  $c_1$  and  $c_2$  are the number of channels in the input tensor and kernel, respectively, and h, w,  $k_h$ , and  $k_w$  are the height and width of the input tensor and kernel. The convolution operation outputs a tensor  $Y \in \mathbb{R}^{c_2 \times h' \times w'}$ , where h' and w' are the height and width of the output tensor, and each element of the output tensor is computed as follows,

$$Y_{c_{2,i},h'_{i},w'_{i}} = \sum_{c_{1,j}=1}^{c_{1}} \sum_{h'_{i}=1}^{k_{h}} \sum_{w'_{i}=1}^{k_{w}} K_{c_{1,j},c_{2,i},h'_{j},w'_{j}} \cdot X_{c_{1,j},h'_{i}+h'_{j}-1,w'_{i}+w'_{j}-1} + b_{c_{2,i}},$$

**Convolution as matrix multiplication.** The convolution operation can be represented as a matrix multiplication by *unfolding* the input tensor into a matrix and the kernel into a matrix and then performing the matrix multiplication. The resulting matrix can be *folded* back into a tensor to obtain the output of the convolution operation. Due to the application of convolution in image processing, the unfolding and folding operations are also known as *im2col* and *col2im*, respectively.

**Lemma 2** (Convolution as Matrix Multiplication). The convolution operation can be represented as a matrix multiplication by unfolding the input tensor x into a matrix  $X' \in \mathbb{R}^{c_1k_hk_w \times h'w'}$  and the kernel k into a matrix  $K' \in \mathbb{R}^{c_2 \times c_1k_hk_w}$ , where h' and w' are the height and width of the output tensor. By processing matrix multiplication Y' = K'X' + b, we can obtain the output tensor  $Y' \in \mathbb{R}^{c_2 \times h'w'}$  and then fold it back into a tensor  $Y \in \mathbb{R}^{c_2 \times h' \times w'}$ .

*Proof.* We provide only a sketch of the proof here, which can be easily verified by the definition of the convolution operation. For the input tensor X and its unfolded matrix X', we only need to extract each local patch of the input tensor that corresponds to the kernel size and reshape it into a column vector, which is called a Toeplitz matrix. The kernel K can be reshaped into a matrix K' by stacking the kernel channels and kernel dimensions into a single dimension. The output tensor Y can be obtained by performing the matrix multiplication Y' = K'X'. Finally, we reshape the output matrix Y' back into a tensor Y by folding it into the original shape of the output tensor. Note that in this simple scenario, we only need to reshape the output matrix Y' into a tensor Y. However, a more complex scenario requires folding the output matrix.

#### B.6 TOKEN EMBEDDING

Token embedding is a technique used in natural language processing (NLP) to convert discrete tokens (such as words or subwords) into continuous vector representations. This is essential for enabling neural networks to process text data, as neural networks typically operate on continuous numerical data. Token embedding maps each token to a high-dimensional vector space, where similar tokens are represented by vectors that are close to each other in that space. In the context of

neural networks, token embedding is taken as a linear transformation that maps the input tokens to a continuous vector space. Formally, we define the token embedding as follows.

**Definition 6** (Token Embedding). The token embedding is a linear transformation that maps a discrete token  $t \in \mathbb{R}^d$  to a continuous vector representation  $e \in \mathbb{R}^m$  using a weight matrix  $W \in \mathbb{R}^{m \times d}$  and a bias vector  $b \in \mathbb{R}^m$ . The token embedding is defined as e = Wt + b, where W is the weight matrix that maps the input token to the continuous vector space, and b is the bias vector that shifts the output vector.

## C SUPPLEMENTARY ASSUMPTIONS

This section summarizes assumptions made in the paper, which are crucial for understanding the theoretical framework and implications of the results. We refer to the *spectral distribution* of a matrix as its eigenvalues or singular values, depending on the context.

# C.1 SPECTRAL FLATNESS OF DATA REPRESENTATIONS

The spectral distribution of a matrix is said to be *flat* if the eigenvalues or singular values are uniformly distributed across a certain range, *i.e.*, several eigenvalues or singular values are close to each other in magnitude. Such property is often satisfied in high-dimensional practical applications. For instance, high-resolution images in computer vision introduce a flat spectral distribution because adjacent pixels are highly correlated; similarly, word embeddings in natural language processing exhibit a flat spectral distribution, as words are often used in similar contexts.

Formally, we make the following assumption about the spectral distribution of the data representations.

**Assumption 1** (Spectral Flatness of Data Representations). Given a dataset for a specific task, the PCA of the data has a flat spectral distribution, i.e., there are several eigenvalues,  $\sigma_k$ ,  $\sigma_{k+1}$ ,  $\sigma_{k+2}$ , ..., that are close to each other in magnitude,

$$\sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_k \lesssim \sigma_{k+1} \lesssim \sigma_{k+2} \lesssim \cdots \leq \sigma_n$$

where  $\lesssim$  denotes that the values are close to each other in magnitude, and  $\sigma_i$  are the singular values of the data matrix.

This assumption holds broadly in practice: *e.g.*, in vision and language, where data (or early feature) representations exhibit relatively flat spectra with clustered eigenvalues. In many real datasets, multiple flats (plateaus) appear, each reflecting a distinct group of features or patterns. We empirically verify this in Figure 3: for each model, we report the minimum number of singular components needed to capture 95% (P95) and 99% (P99) of total spectral energy. ResNet-50 (He et al., 2016) reaches 95.4% with 23 components (P95@23) and 99.0% with 32 (P99@32). ViT-Base (Dosovit-skiy et al., 2021), despite a higher-dimensional patch projection, requires 90 and 106 components to attain 95.4% and 99.0%, respectively.

## C.2 ALIGNMENT OF SINGULAR VECTORS AND PRINCIPAL COMPONENTS

Training a neural network amounts to optimizing parameters to minimize a task loss. In this process, the right singular vectors of the first-layer weight matrix emphasize directions that are most predictive for the task, whereas the principal components of input (or early-feature) representations summarize dominant data variability. Empirically, these two families of directions often exhibit notable alignment in practical settings (Hacohen & Weinshall, 2022). This section formalizes that observation by relating first-layer right singular vectors to the principal components of data representations, and proceeds under the following alignment assumption.

**Assumption 2** (Alignment of Singular Vectors and Principal Components). Given a dataset and a neural network trained on this dataset, the first layer's weight matrix is denoted as  $W \in \mathbb{R}^{m \times n}$ , where n is the number of input dimensions and m is the number of output dimensions, and the covariance matrix of the data representations is denoted as  $C \in \mathbb{R}^{n \times n}$ . Let  $U\Sigma V_{svd}^T$  be the singular value decomposition (SVD) of the first layer's weight matrix W, where  $U \in \mathbb{R}^{m \times m}$  is the left singular vector matrix,  $\Sigma \in \mathbb{R}^{m \times n}$  is the diagonal matrix of singular values, and  $V_{svd} \in \mathbb{R}^{n \times n}$ 

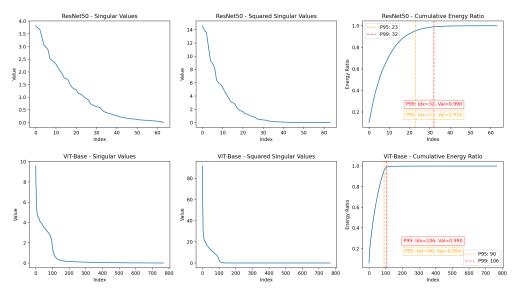


Figure 3: The spectral structure of the first-layer weight matrices in ResNet-50 and ViT-Base via singular value decomposition.

is the right singular vector matrix. Let  $V_{pca}\Lambda V_{pca}^T$  be the eigendecomposition of the covariance matrix C, where  $V_{pca} \in \mathbb{R}^{n \times n}$  is the eigen vector matrix and  $\Lambda \in \mathbb{R}^{n \times n}$  is the diagonal matrix of eigenvalues. The assumption states that the right singular vectors of the first layer's weight matrix are aligned with the principal components of the data representations, that is

$$V_{svd} = V_{pca}$$
.

This assumption is satisfied in many practical applications, where the right singular vectors of the first layer weight matrix are aligned with the principal components of the data representations. The alignment between the right singular vectors and the principal components is crucial for the neural network to learn the most important features from the data for the task. In some special models, such as linear models, the alignment is exact, i.e.,  $V_{svd} = V_{pca}$ .

Table 5: Numerical experiment on first-layer singular-value.

	Max	Min	Mean	Median
ResNet-50	3.8203	$7.77 \times 10^{-8} \\ 3.68 \times 10^{-5}$	1.0242	0.5904
ViT-Base	9.5686		0.5666	0.0482

## D SUPPLEMENTARY EXPERIMENTS AND RESULTS

This section provides additional experimental results. All experiments use Python 3.12.3, PyTorch 2.3.0, and Transformers 4.44.2 (CUDA 12.3) on a workstation with an AMD Ryzen Threadripper PRO 5965WX (24 cores), 256 GB RAM, and two NVIDIA RTX A6000 GPUs.

Table 6 presents the performance of different models on the GLUE benchmark to demonstrate that our approach is generic. Figure 4 illustrates the effect of perturbation strength on an example image.

Table 6: Performance across GLUE benchmark for different models.

			BE	RT-base	•		RoBERTa-base							
	CoLA	MNLI	QNLI	QQP	RTE	SST2	STSB	CoLA	MNLI	QNLI	QQP	RTE	SST2	STSB
Baseline	54.2	83.4	90.5	90.1	60.3	91.6	87.1	53.8	87.7	92.8	90.9	66.1	94.5	87.5
NE (Ours)	54.5	82.9	89.5	89.6	60.3	89.6	86.9	55.5	87.5	92.4	90.8	65.3	94.6	87.2
Unauthorized	33.4	31.8	50.5	36.8	47.3	50.9	60.2	37.1	35.4	49.5	63.2	52.7	49.1	63.5

## D.1 ROBUSTNESS AGAINST PREPROCESSING IN VLMS

Vision-language models apply multi-stage, model-specific preprocessing that can scramble input space recoding before it reaches early features, which makes inference time usage control difficult.

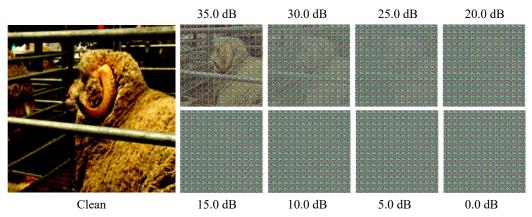


Figure 4: Effect of perturbation strength (PSNR, dB). Visual examples across increasing strength. Authorized models remain stable even at 10dB; at 0dB, ResNet-50 on ImageNet loses only 0.1% accuracy.

In practice images are decoded to RGB and converted to floats, resized to model specific canvases such as 448, 512, 896 or 1024 with aspect ratio preserved and letterbox padding, optionally center or random cropped, normalized with per channel means and standard deviations similar to CLIP or EVA, partitioned into patches or tiles to form visual tokens with padding aligned to stride and patch size, projected into the language model embedding space and augmented with resolution dependent positional encodings, with occasional multi image packing and implementation dependent interpolation and JPEG rounding. Although InternVL3 and Qwen2.5 VL differ in exact choices, they follow this general pattern. Despite these complications, our experiments show that NE remains robust on MMBench across AR, CP, FP C, FP S, LR, and RR, where authorized performance for InternVL3 is essentially unchanged and unauthorized utility for Qwen2.5 VL remains low as summarized in Table 4 and illustrated in Figure 2.

## D.2 RECONSTRUCTION ATTACK

Table 7 summarizes super-resolution reconstruction attempts on VDSR using SR-ResNet in blackbox (Noise2Noise) and white-box (Noise2Clean) settings. As a sanity check, standard Gaussian noise is largely removable (33.5–35.9dB after SR-ResNet). In contrast, NE resists recovery: reconstructions stay near the input level (about 10–11dB) in both settings, including the strongest white-box variant with all layers. Thus, recoding remains effectively non-invertible for attackers while leaving authorized performance essentially unchanged.

Table 7: PSNR (dB) on VDSR under super-resolution reconstruction attempts.

Method	$\tilde{x}$	SR-ResNet (black box)1	SR-ResNet (white box) <sup>2</sup>
Gaussian	15.8	33.5	35.9
NE (Ours)	10.2	10.5	10.9
NE (Ours) <sup>3</sup>	10.7	10.8	11.1

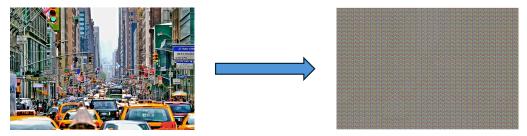
Noise2Noise: train on recoded → recoded pairs.

# E DISCUSSION OF LIMITATIONS

Our setting grants the defender white-box access to  $f^*$  and a probe source to estimate an insensitivity subspace used for recoding. Against a *method-aware* or *parameter-aware* adversary (GA/TA/AA in Section 2.2), the perturbation itself becomes an attack surface. If the confining subspace (or a close approximation) is recovered, an input-side projector that reweights toward principal directions can *partially* cancel the recoding and raise  $m(f', \tilde{x}')$ . This is realistic: linear projection does not require access to  $f^*$ , only an estimate of the basis. However, recovery is imperfect in practice because acquisition and preprocessing (resize, compression, normalization, tokenization) generally do not commute with a fixed projector, and the authorized benignity relies on  $f^{*}$ 's internal representations rather than pure input-space orthogonality. Thus, projection-back can reduce effect size but does not guarantee full restoration on arbitrary f'.

Noise2Clean: train on recoded  $\rightarrow$  clean pairs.
Attacker has the white box access to the model.

A stronger adversary with training-time control can *counter-adapt* by regularizing sensitivity (e.g., encouraging larger singular components or Jacobian norms along estimated bases) so that the insensitivity directions shrink after training. This can realistically recover non-target utility, especially when the task admits redundancy. The trade-off is empirical rather than guaranteed: making all directions sensitive tends to increase brittleness and harms calibration/robustness on common corruptions, but such side effects may be acceptable to an attacker optimizing only  $m(f', \tilde{x}')$ . Additional practical limits include dependence on a modest probe budget for estimating the spectral basis; attenuation under aggressive acquisition pipelines (heavy crops or compression) or domain shift that changes early-layer geometry; and detectability, supervised or self-supervised detectors can learn to flag or strip structured, low-energy recodings when the basis is static. Finally, our analysis focuses on early layers and linearized views; extending guarantees to deeper Jacobians and temporally coupled modalities (audio/video) remains open.



#### Prompt: Please describe the image in detail.

## InternVL3 Response (Authorized):

image depicts a bustling urban street scene, likely in a major city like New York. Here are the details:

- 1. Traffic: Numerous yellow taxis are navigating the busy street, suggesting this is a typical busy day in a metropolitan area.
- 2. Buildings: Tall multi-story buildings line both sides of the street. The architecture is urban, with many buildings displaying commercial
- 3. Traffic Lights: Traffic lights are visible at various intersections, with the central location featuring a green light
- 4. Signage: A variety of street signs, both local and directional, are displayed. Key signs include:

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1042 1043 1044

1046

1047

1048

1049

1050 1051

1052

1053

1054

1055

1056

1057

1058

1062

1063

1064

1067

1068

1069

1070 1071

1074

1078 1079

- "One Way"
  "Pine St. Route S6E"
- "Don't Block The Box"
  "Fine +2 Points"
- "New York State Highway 575" Several blue and white street signs pointing directions.
- Buses Only" next to "Red Zon
- Traffic signs indicating restrictions on road lighting.
- American Flags: Colorful American flags are mounted on the buildings and display at various intersections.
- People: People are visible on the sidewalks, suggesting the area is pedestrian-friendly. The crowd appears to be busy with vehicles. 7. Bus: A green bus is visible, heading in the direction indicated by one
- 8. Urban Environment: The street is flanked by numerous shops and

businesses, contributing to a dense commercial atmosphere

9. Weather: The weather seems overcast, with a muted color palette predominantly in greys, blues, and golds

This depiction captures the dense, bustling energy of a major city's busy street, typical of New York City.

#### Qwen2.5-VL Response (Unauthorized):

The image appears to be a pixelated or low-resolution photograph of a person standing in front of a building. The person is wearing a dark-colored top and light-colored pants, and the background features a structure with multiple windows. The overall quality of the image is poor, making it difficult to discern finer details.

Figure 5: Illustrative visualization of effective on data authorization on VLM.