

AdvHash : Set-to-set Targeted Attack on Deep Hashing with One Single Adversarial Patch

Shengshan Hu¹, Yechao Zhang¹, Xiaogeng Liu¹, Leo Yu Zhang², MingHui Li³ and Hai Jin⁴

¹ School of Cyber Science and Engineering, Huazhong University of Science and Technology

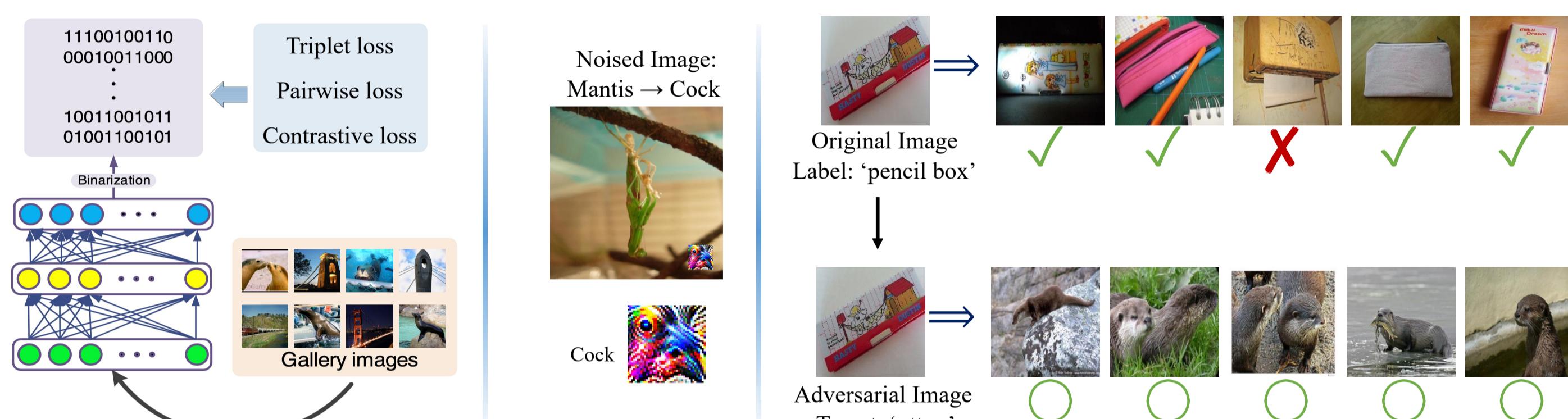
² School of Information Technology, Deakin University

³ School of Software Engineering, Huazhong University of Science and Technology

⁴ School of Computer Science and Technology, Huazhong University of Science and Technology

Introduction

- Deep hashing serves as a powerful tool for large-scale multimedia retrieval.¹ However, it is highly vulnerable to adversarial noise for its inherited nature from DNNs.
- Adversarial attacks can be realized by crafting perturbations or patches.² Compared to perturbation, patch is more feasible to real-world scenario and more robust to defense mechanism like image transformations. However, the prior attacks on deep hashing all fall into the perturbation type.
- There are two types of adversarial attacks: non-targeted and targeted, the goal of targeted attacks is more challenging than non-targeted.³
- Prior attacks on deep hashing all are image-specific, fail to explore the generalizability of adversarial noise.



Motivation

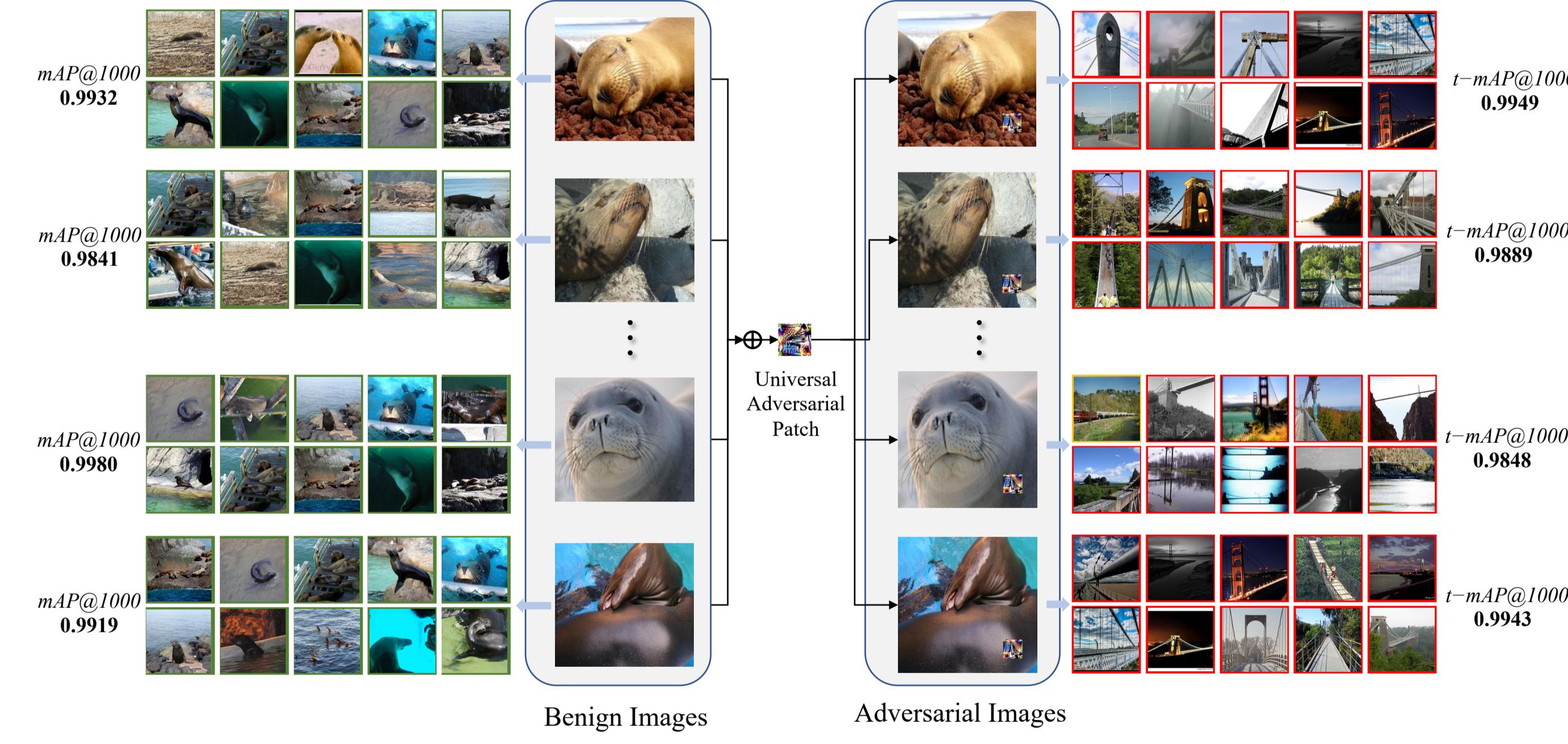
Attack Goals

- Goal 1 : Realizing the Targeted Attack on Deep Hashing
 - Making the top-K retrieved results all belong to another predefined class.
- Goal 2 : Improving the Generalization of Adversarial Noise
 - Making the single adversarial noise applicable for an entire class of images.
- Goal 3: Exploring the Patch-based Scenario on Deep Hashing
 - Making the adversarial noise confined to a localized area of image.

Major Motivation

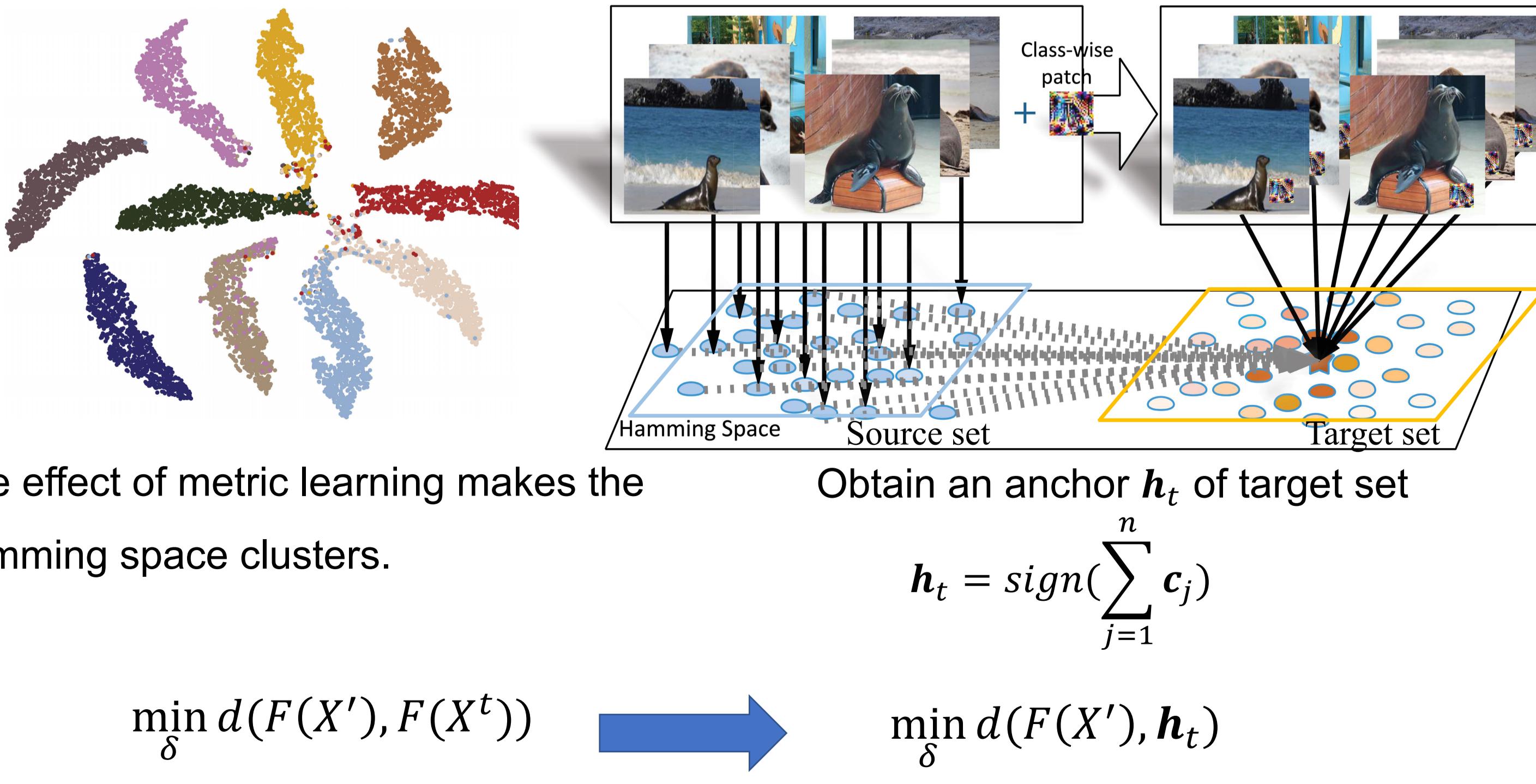
- Achieving the SOTA attack strength while keeping the adversarial noise universally applicable.

Attack Illustration



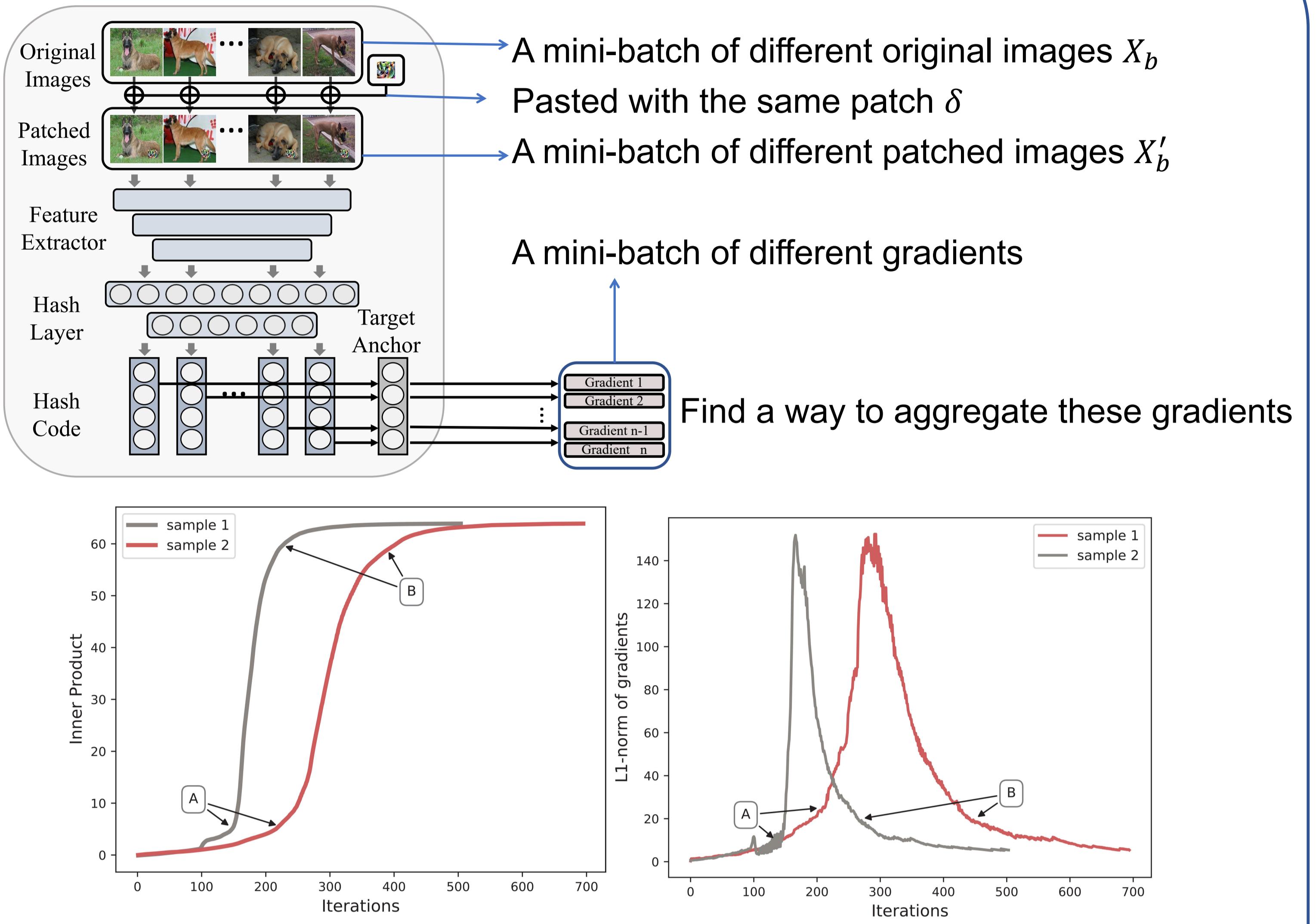
Problem Definition

Optimization problem to generate adversarial patch δ : $\min_{\delta} d(F(X'), F(X^t))$



The effect of metric learning makes the hamming space clusters.

Gradients Aggregation

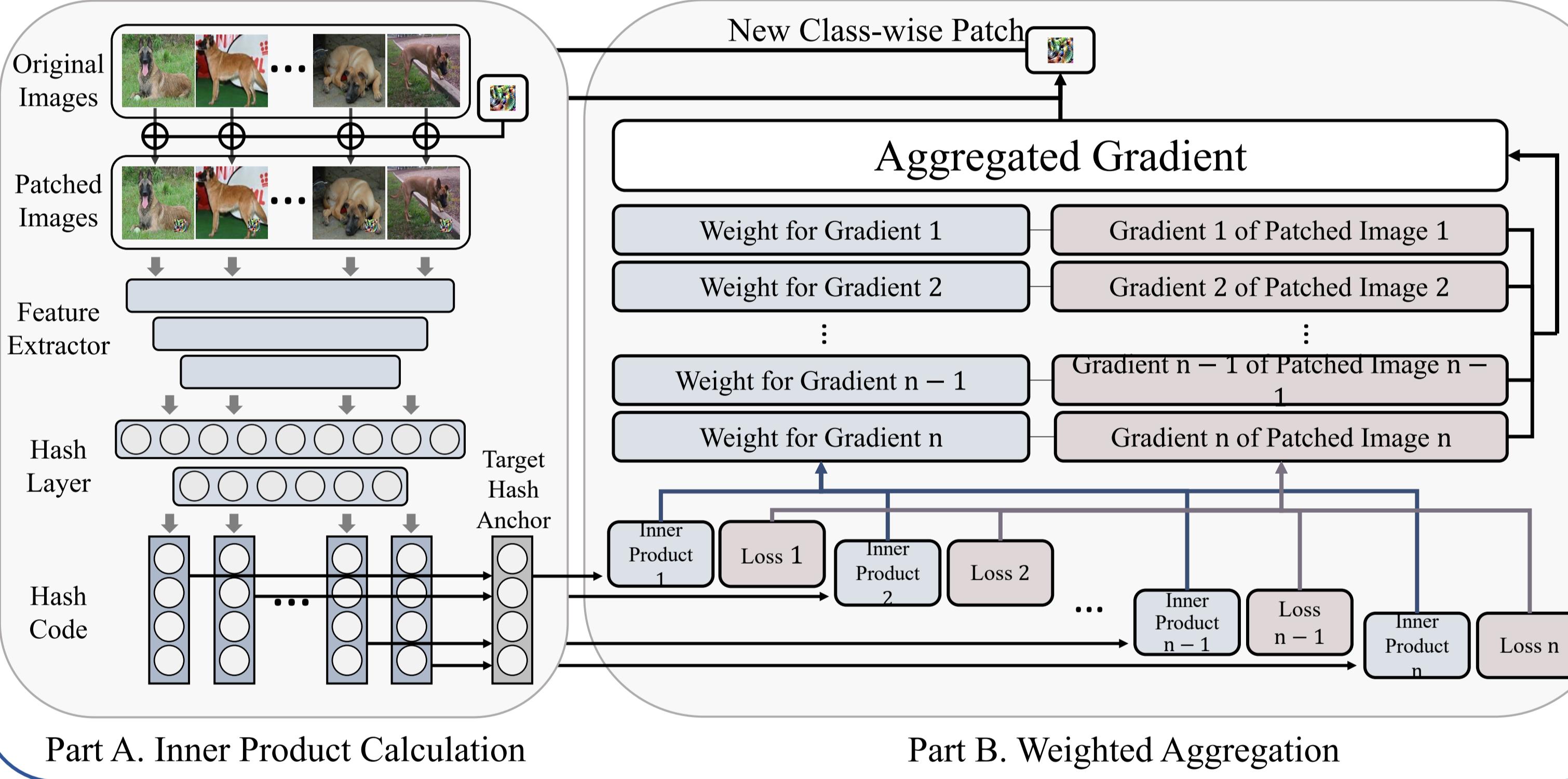


$$P_u = \mathbf{c}_u \cdot \mathbf{h}_t \quad W_u = \begin{cases} P_u + \beta, & \text{if } P_u \leq T \\ P_u - \gamma, & \text{otherwise} \end{cases}$$

$$\nabla \mathcal{L}_u = \frac{\partial \mathcal{L}}{\partial \mathbf{x}'_u} \quad \bar{\nabla} = \frac{\sum_{u=1}^U W_u \times \nabla \mathcal{L}_u}{\sum_{u=1}^U W_u}$$

Assign different weights for different samples

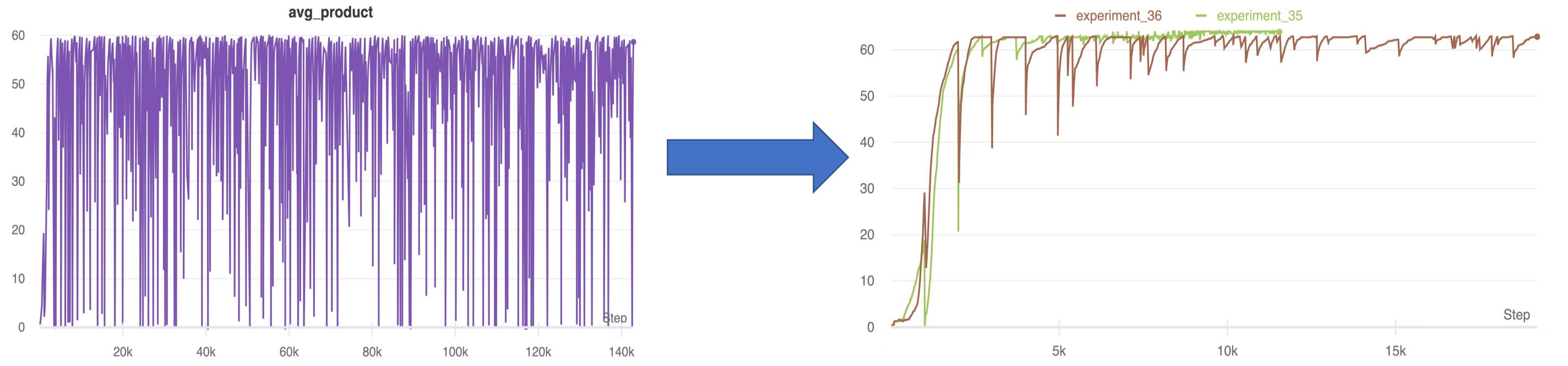
Overall Pipeline



Patch Averaging

Averaging a few patches trained from the previous mini-batches as the initial patch for current mini-batch X'_b .

$$\delta_b^0 = \frac{1}{p} (\delta_{b-p} + \delta_{b-p+1} + \dots + \delta_{b-1})$$



Comparison

DHTA is an image-specific attack method, which has not considered generalization.

The adversarial noise generated by AdvHash generalizes well and still can compete or even outperform DHTA.

Table 4: Comparison with DHTA [2]

Mapping	Samples	CSQ		HashNet					
		ResNet50	VGG16	ResNet50	VGG16	32	64	32	64
M_4	DHTA	99.49	99.77	80.94	36.82	99.68	99.25	6.36	36.82
	AdvHash-train	98.33	89.3	97.8	34.97	99.63	95.85	95.86	99.5
	AdvHash-test	97.14	94.28	97.21	29.72	99.04	95.61	97.42	99.5
M_6	DHTA	97.73	99.29	78.71	76.36	92.87	84.09	96.16	97.55
	AdvHash-train	97.79	81.3	99.03	99.54	43.38	31.06	94.62	99.6
	AdvHash-test	95.87	83.27	98.64	99.15	33.15	36.56	92.64	98.73
M_{10}	DHTA	99.66	99.44	82.5	54.68	69.06	28.94	97.2	99.58
	AdvHash-train	97.66	97.41	80.59	98.46	71.32	69.99	97.06	99.59
	AdvHash-test	98.67	99.38	79.94	97.33	70.6	70.8	96.85	99.57