# Bucks for Buckets (B4B): Active Defenses Against Stealing Encoders


Jan Dubiński* [1,2] Stanisław Pawlak* [1] Franziska Boenisch* [3]
Tomasz Trzciński [1,2,4] Adam Dziedzic [3]

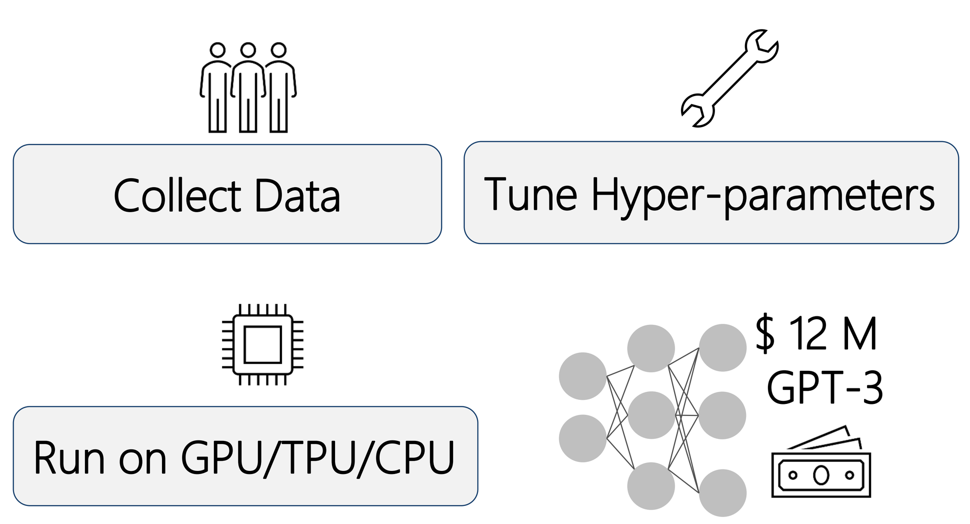
[1]Warsaw University of Technology [2]IDEAS NCBR [3]CISPA [4]Tooplox *Equal contribution


## Introduction

- Self-supervised (SSL) models are increasingly prevalent in machine learning (ML) with versatility in downstream applications through high-dimensional representations and unlabeled data.
- SSL models are vulnerable to model stealing attacks where an adversary can steal an ML model exposed via a public API with query access.
- Attacks against SSL models are query efficient: Adversary may steal a well-performing model with much fewer queries than the number of training data points.
- Existing defenses against stealing supervised models are inadequate in the SSL setting.

## Motivation

It is a costly venture to create SSL APIs:




## Contributions

- We present B4B, the first active defense against encoder stealing that does not harm legitimate users' downstream performance. B4B's three building blocks enable penalizing adversaries whose returned representations cover large fractions of the embedding space and prevent sybil attacks.
- We propose a concrete instantiation of B4B that relies on Locallity-sensitive hashing which reduces the quality of user representations when they occupy too many hash buckets.
- We evaluate our defense using five datasets from the computer vision domain and show that our defense can successfully prevent model stealing attempts without decreasing encoder utility for legitimate users

## Problem Setup

Encoders transform complex input queries into high-dimensional representations. SSL APIs return representations that can further be used to train classifiers for multiple downstream tasks.

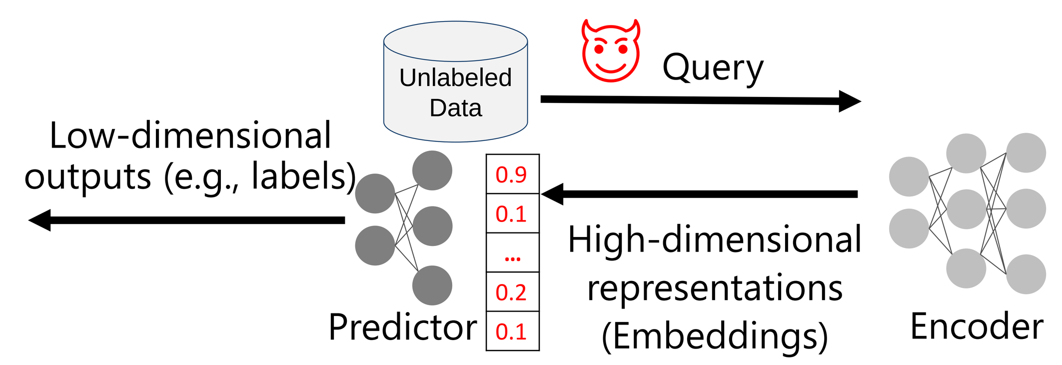
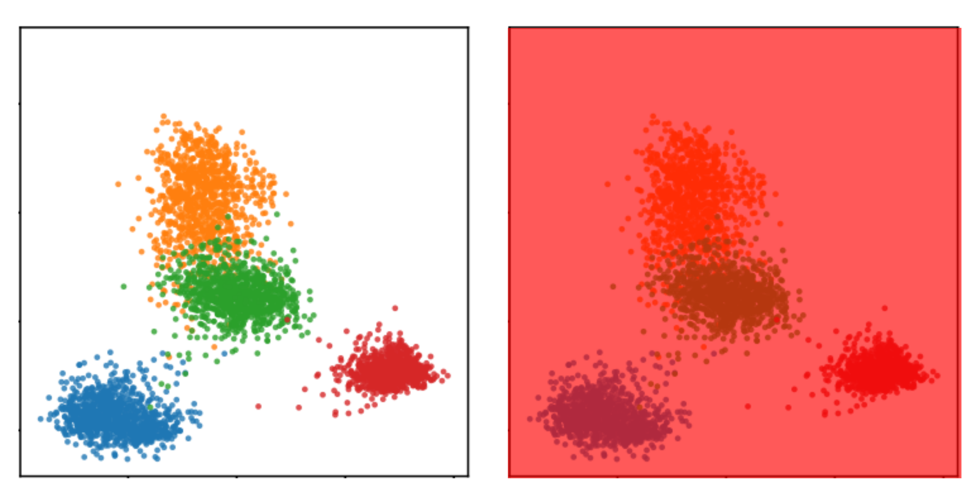



Figure 1: **Self-supervised learning API** setup and the use of representations to steal encoders.

Attackers leverage query access to the API to extract information and train a duplicate model. Existing defenses are inadequate for self-supervised models.

## Intuition behind Our Framework



- Queries from legitimate users occupy a single region of the latent space.
- Attacker must query the entire representation space to steal the encoder.

## Idea

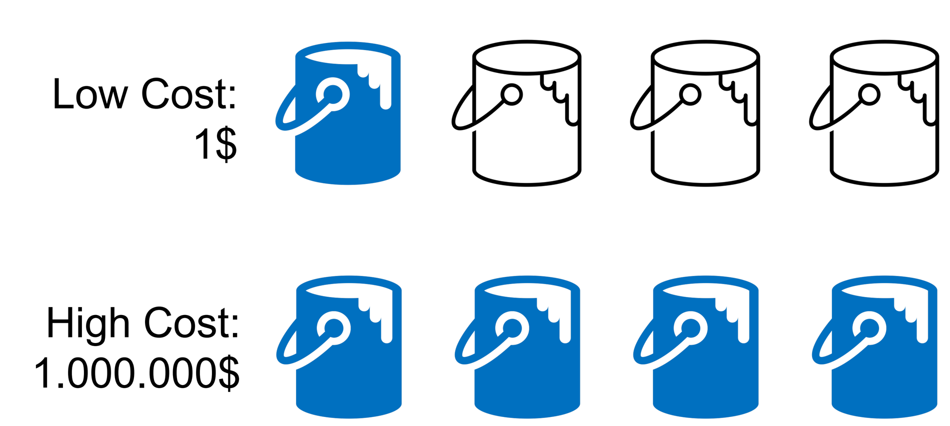



Figure 2: **Bucks for Buckets**.

We divide the encoder latent space into buckets and adjust the querying cost depending on the fraction of buckets occupied by the user's queries.

## Main Algorithms

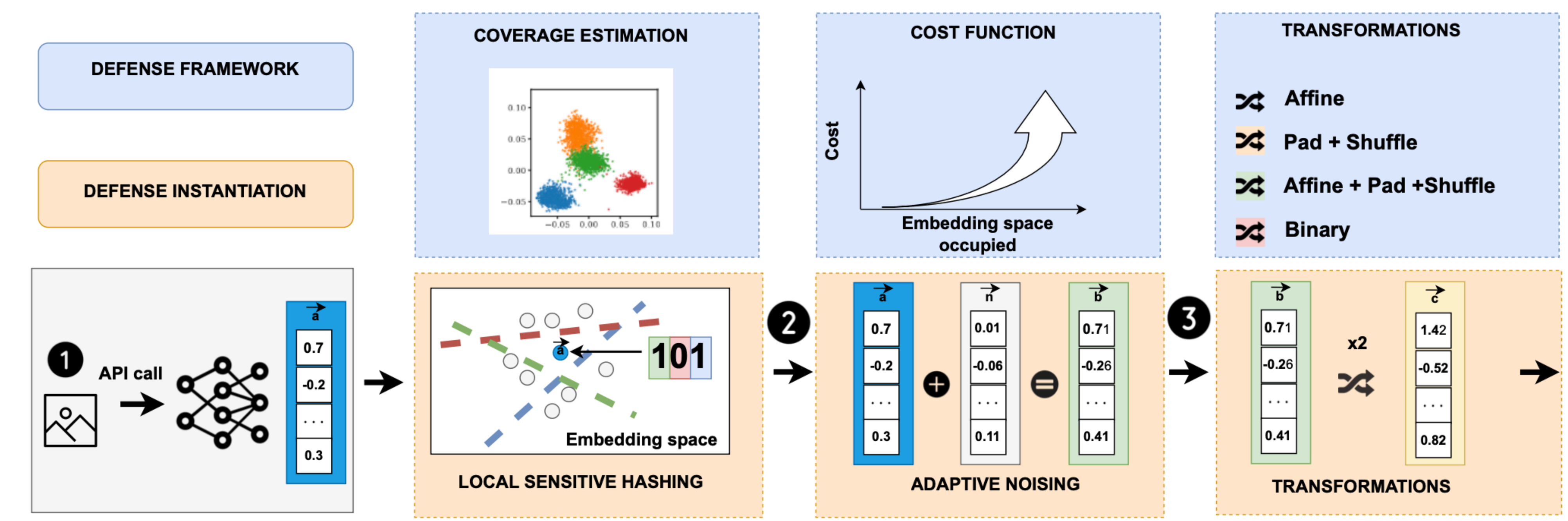



Figure 3: **B4B building blocks:** (1) A coverage estimation to track the fraction of embedding space covered by the representations returned to each user, (2) a cost function to map the coverage to a concrete penalty to prevent stealing, (3) per-user transformations that are applied to the returned representations to prevent sybil attacks. .

### Coverage

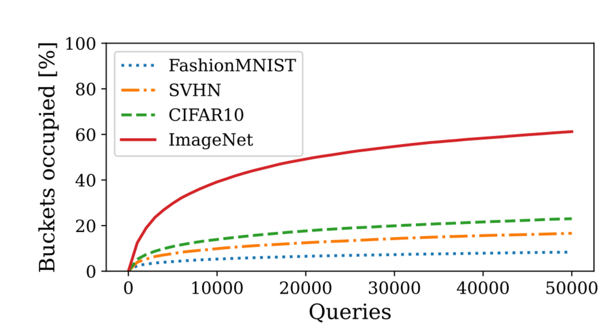



Figure 4: **Coverage estimation.**

### Cost

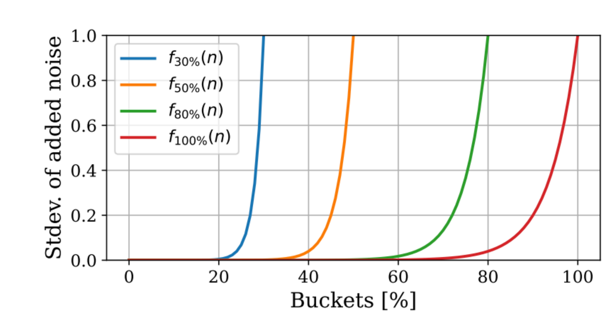



Figure 5: **Cost function.**

### Transformations

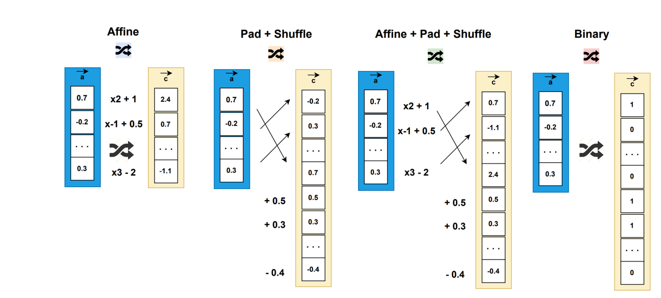



Figure 6: **Transformations.**

## Empirical Evaluation

Table 1:**B4B effects.** No harm to legitimate users. Successfully prevents model stealing attacks, including sybil attacks.

| User | Defense | # Queries | Dataset | Type | CIFAR10 | STL10 | SVHN | F-MNIST |
|---|---|---|---|---|---|---|---|---|
| Legit | None | All | Task | Query | 90.41 ±0.02 | 95.08±0.13 | 75.47±0.04 | 91.22±0.11 |
| Legit | B4B | All | Task | Query | 90.24±0.11 | 95.05±0.1 | 74.96±0.13 | 91.7±0.01 |
| Attacker | None | 50K | ImgNet | Steal | 65.2±0.03 | 64.9±0.01 | 63.1±0.01 | 88.5 ±0.01 |
| Attacker | B4B | 50k | ImgNet | Steal | 35.72±0.04 | 31.54±0.02 | 19.74±0.02 | 70.01±0.01 |
| Sybil | B4B | 2×50k | ImgNet | Steal | 39.56± 0.06 | 38.50±0.04 | 23.41±0.02 | 77.01± 0.08 |

## Conclusions

- B4B is the first active defense for self-supervised encoders that prevents stealing without degrading legitimate user experience.
- We use local sensitive hashing to track the coverage of the latent space.
- We adjust the utility of the returned representations according to the coverage of the latent space to prevent stealing.
- We use per-user transformations to prevent sybil attacks.

## Full paper