Robust Hashing Learning via Random Smoothing

Xinyue Feng, Chuxiao Zuo, Wujun Li

1 Introduction

1.1 Deep Hashing based Retrieval

- What is Nearest Neighbor Search?

Given a query point q, return the points closest (similar) to q in the database

BigData

Challenge

- Time consuming
- Storage cost

1.1 Deep Hashing based Retrieval

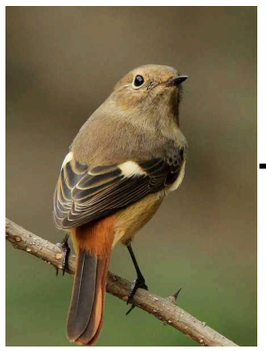
- Why we need hashing method?

General Method

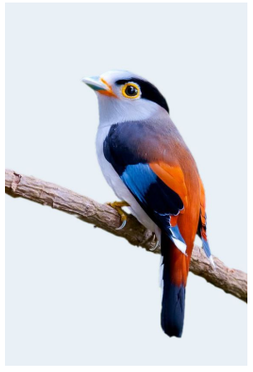
- Real-value, High dimension

Hashing Method

- Binary code, Low dimension



(0.3, 0.3, … 0.1)



(0.4, 0.3, … 0.1)

(0.9, 0.9, … 0.8)

(0, 0, 1, 0, 1)

(0, 0, 1, 0, 0)

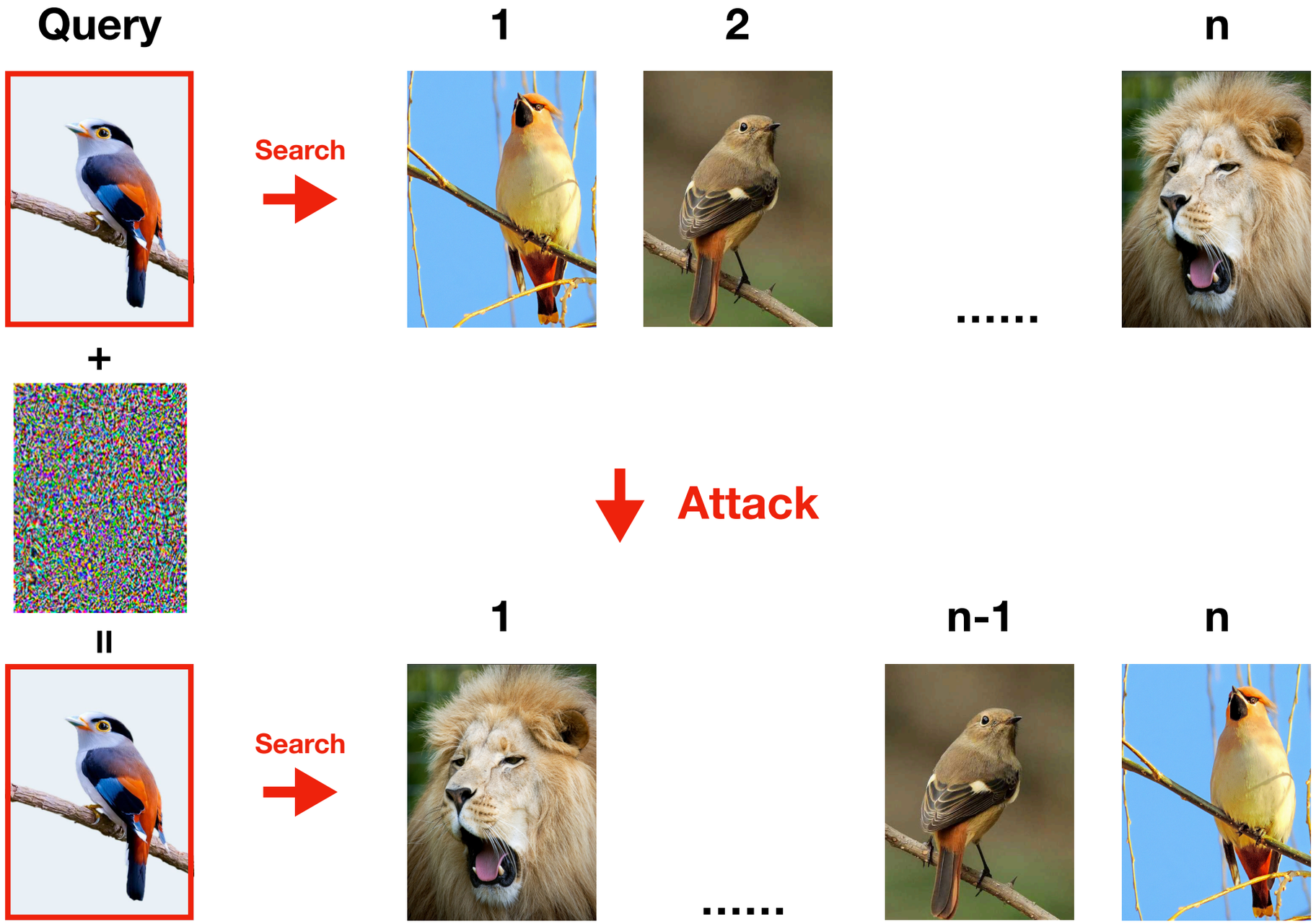
Close

(1, 1, 1, 1, 0)

Farther

1.2 Adversarial Attack

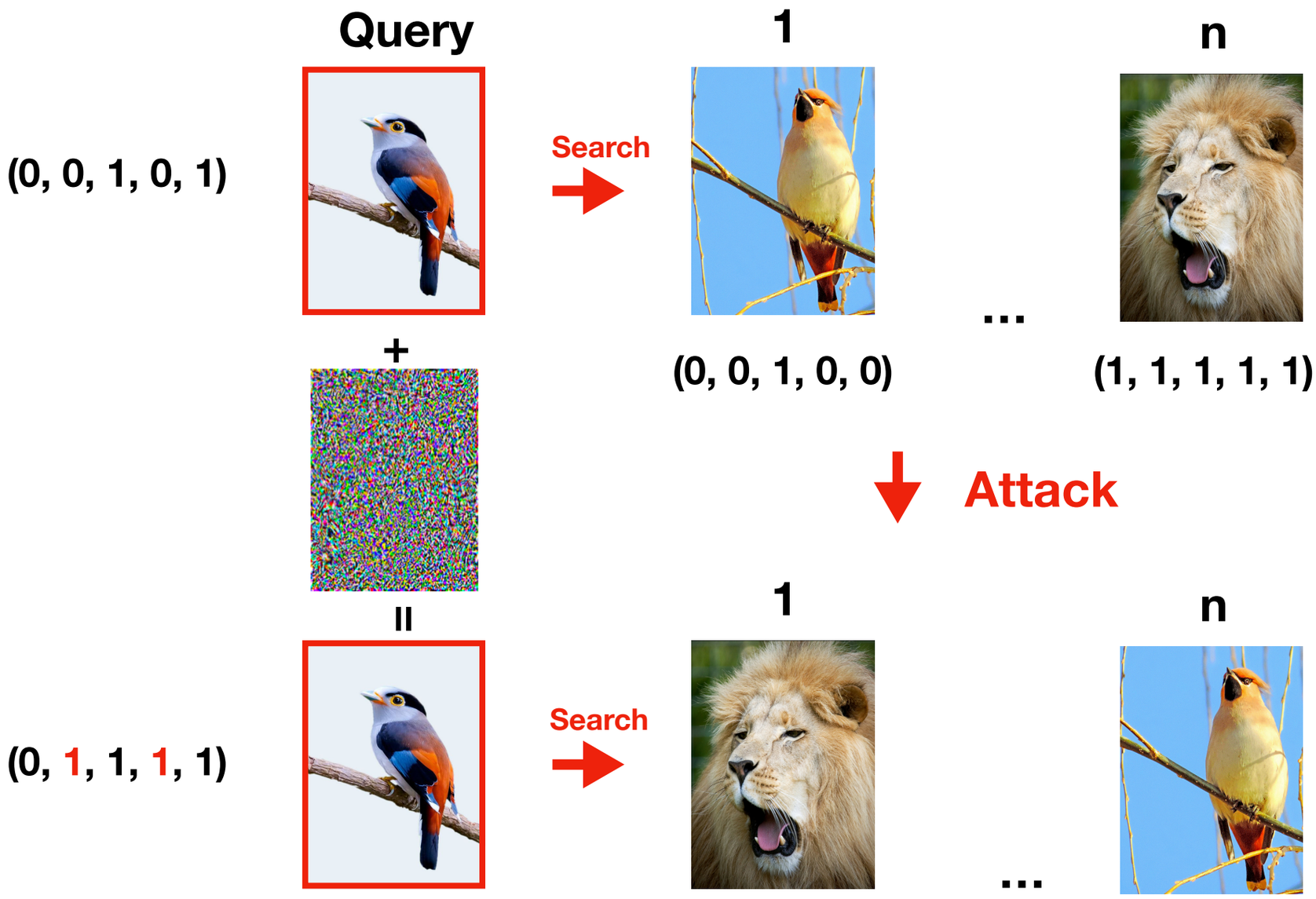
- Problem: Adversarial attacks



1.2 Adversarial Attack

- How does the adversarial attack work?

$$X^* = \arg\max_{X' \in \mathcal{P}_X} L(f(X'), y) \ . \qquad \text{where} \quad \mathcal{P}_{p,\varepsilon}(x) = \left\{ x' \in \mathbb{R}^d : \|x - x'\|_p < \varepsilon \right\} \ .$$



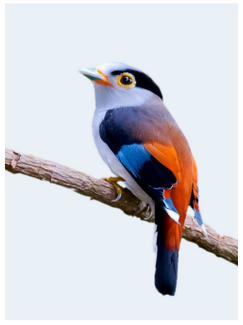
1.2 Adversarial Attack

- How to defense?

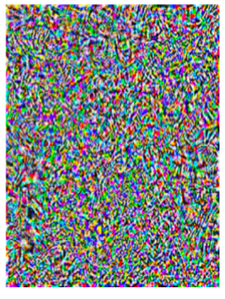
Our Method: **Smoothing Hashing Model**

General Model

(0, 0, 1, 0, 1)

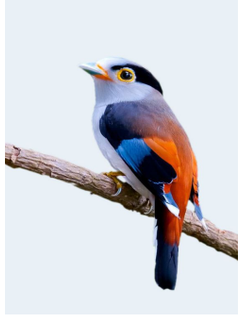


+



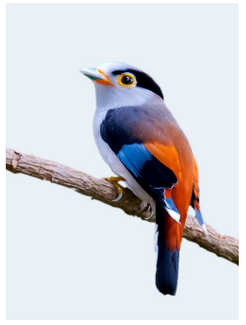
=

(0, 1, 1, 1, 1)

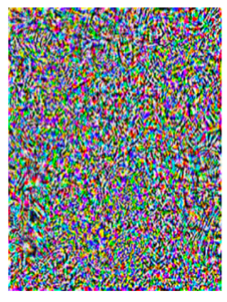


Our Model

(0, 0, 1, 0, 1)

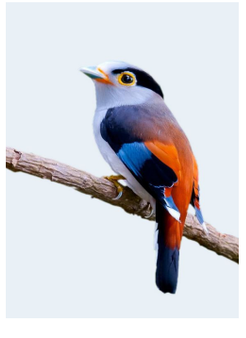


Gaussian Noise +



=

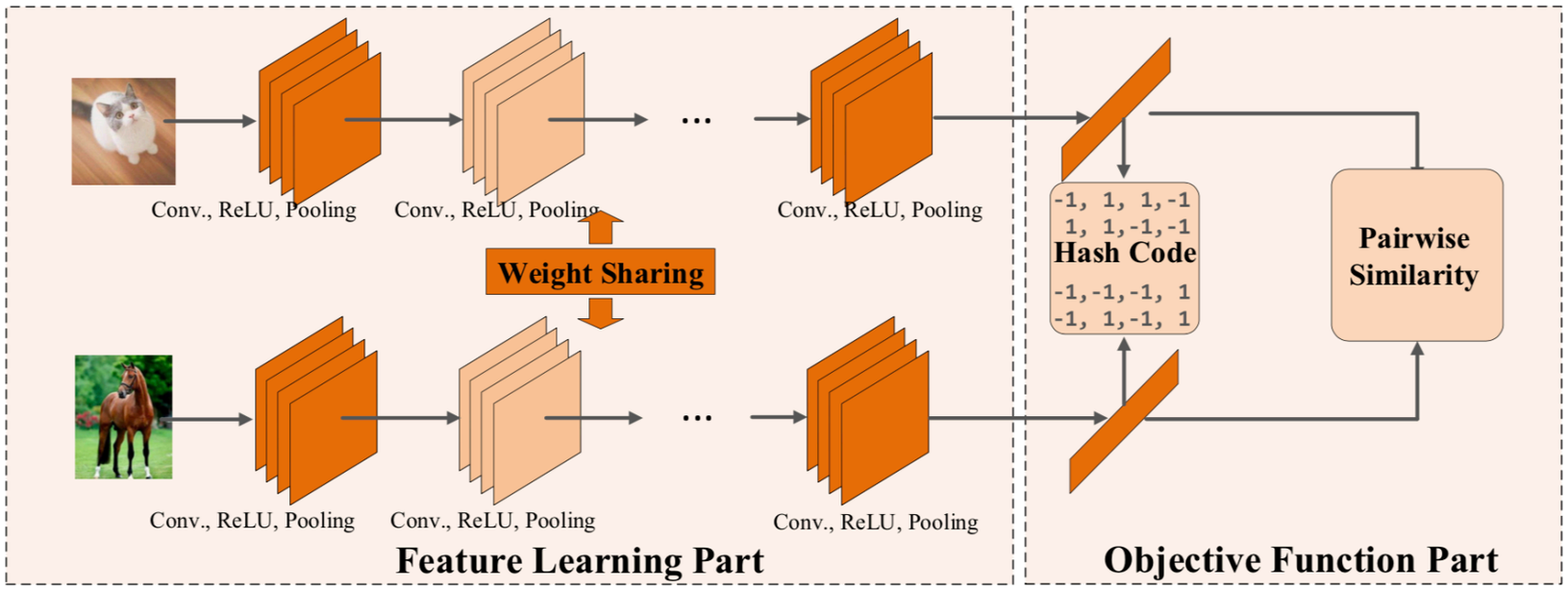
(0, 0, 1, 0, 1)



2 Proposed Method

2.1 Backbone

- DPSH model



$$\boldsymbol{L}_{DPSH}(\boldsymbol{X}, S) = -\sum_{s_{ij} \in S} [s_{ij} log(\sigma(\Theta_{ij})) + (1 - s_{ij}) log(1 - \sigma(\Theta_{ij}))] + \lambda \|sign(u_i) - u_i\|^2$$

$$where\ \Theta_{ij} = \frac{1}{2} u_i^T u_i,\ u_i = f(x_i)$$

W. Li, S. Wang, and W. Kang, “Feature learning based deep supervised hashing with pairwise labels,” in IJCAI, 2016

2.2 Smoothing Hashing Model

- How to construct a smoothing hashing model

A K bit hashing model $f(x)$

$$f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_K(\boldsymbol{x})], where\ f_k(\boldsymbol{x}) \in \{-1, 1\}$$

Then the smoothing hashing model constructed from $f(x)$ is

$$g(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), ..., g_K(\boldsymbol{x})]$$

$$where\ g_k(\boldsymbol{x}) = \underset{c \in \{-1,1\}}{argmax}\, \mathbb{P}(f_k(\boldsymbol{x} + \varepsilon) = c), \varepsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$$

At test time, we use the Monte Carlo algorithms to construct $g(x)$

2.2 Smoothing Hashing Model

- Certified Robustness

Theorem 1 (Certified Radius for Hashing model). Supoose we are given an example $\boldsymbol{x}$, $\varepsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ and a K bit hashing model $f(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_K(\boldsymbol{x})]$. If $\underline{p_{A_k}} \in (\frac{1}{2}, 1]$ satisfies the following conditions:

$$\mathbb{P}(f_k(\boldsymbol{x} + \varepsilon) = c)) \geqslant \underline{p_{A_k}},\ \underline{p_{A_{(q+1)}}} \geq \frac{1}{2}$$

where $\underline{p}$ indicates lower bounds of p and $p_{A_{(q+1)}}$ indicates order statistic. Then, for $g(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), ..., g_K(\boldsymbol{x})]$ where $g_k(\boldsymbol{x}) = argmax_{c \in \{-1,1\}} \mathbb{P}(f_k(\boldsymbol{x} + \varepsilon) = c)$, we have:

$$d_h(g(\boldsymbol{x} + \delta), g(\boldsymbol{x})) \leqslant q, \forall\ \|\delta\|_2 < \sigma \Phi^{-1}(\underline{p_{A(q)}})$$

Certified Radius

Robustness to Adversarial Examples

Establish the connection

Robustness to Random Gaussian Noise

easier to defense

2.3 Experiments

- Results

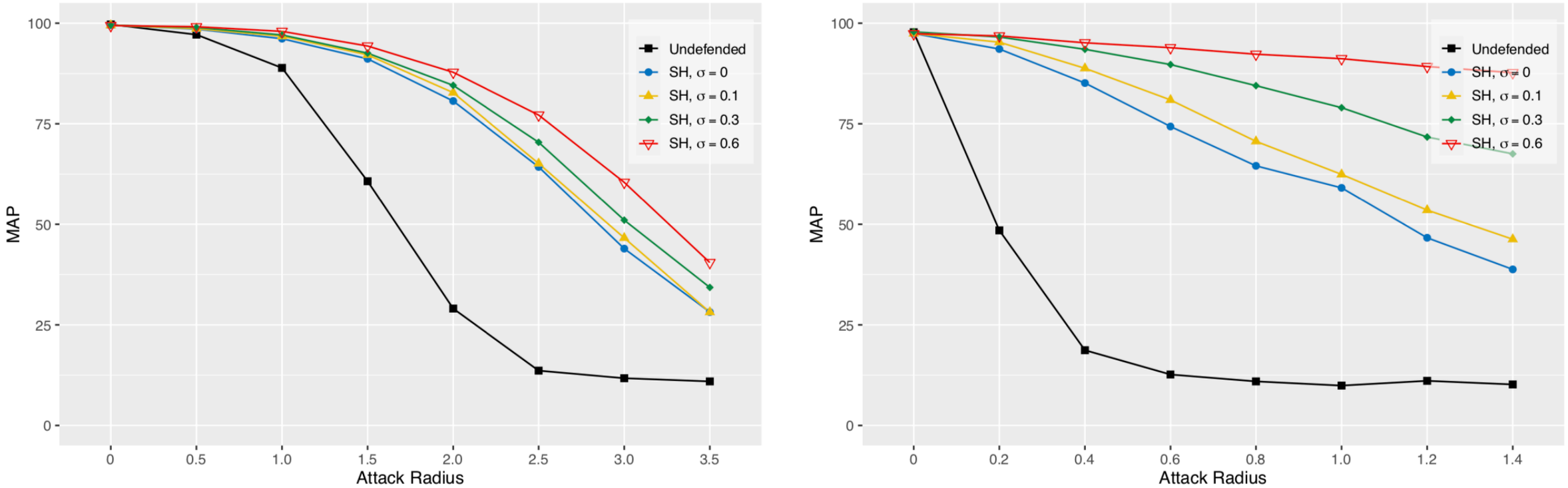


Fig.1 Mean Average Precision (MAP) of the hashing model under different test settings on MNIST(a) and CIFAR-10(b). The hyperparameter σ is the variance of Gaussian noise.

Thank you!