Link to all our evaluation experiments: https://github.com/learnEcosG/Trispectrum/blob/main/EXP.pdf

We had spent a lot of time and energy to do extensive experiments in evaluation, but we can't elaborate it in detail due to space limitation. If you think some experiments are more necessary, we can supplement them later.

In addition, our basic idea is balancing the anti-theft ability and generalization by adjusting adversarial distance ***r***. As the ***r*** decrease, the available space becomes smaller which means the security stronger, but the generalization weaker. And we have proved this characteristic in experiments.

We sincerely appreciate the strong agreements from reviewer #5. And we will explain concerns from reviewer #3, #2 point by point:

Reviewer #3:

Q1: A detailed comparison between this work and the cryptography-based related work [22] is suggested to be given.

A1: Thanks for your suggestion. If this paper is lucky to be accepted, we are willing to add relevant contents.

Q2: This work provides active IP protection of DNNs, which is similar to the work [22] .

A2: Both our work and [22] propose a proactive protection framework through adversarial training, but we can self-control the anti-theft ability introducing the adversarial distances parameter ***r***. In addition, it is obvious that the defense ability against pair-attack in [22] is not significant.

Q3: I think this work has the concern -- if a decrypted model can be leaked, the encoder, the distance l and the signature in this work are also at risk of leakage.

A3: We have done relevant experiments which show that we can resist this attack excellently. (**The detail can refer to Input-only A in section 5.5 from Link)**

Q4: I think they should also discuss model distillation attacks.

A4: In fact, we have also done experiments in this scenario(Pair-attack) where we suppose the encoder structure, sign and input-output dataset are leaked, the task for the adversary is to recover the encoder with the given input-output pairs as the input and the ground truth. The attack results are as follows:(**The detail can refer to Pair attack in section 5.5 from Link**)

Attack Methods	Sample 1		Sample 2		Sample 3		Sample 4		Sample 5	
	MAE	DIST	MAE	DIST	MAE	DIST	MAE	DIST	MAE	DIST
Pair attack	11743.3	5.72	10493.4	5.85	11343.8	5.65	10893.4	5.81	11483.3	5.66

Q5: As I understand, the encoder is trained in Step 1 with objective Function (2). Its arguments θ_e are supposed to be fixed after Step 1. Then, why does Function (3) optimize θ_e in Step 2?

A5: In fact, we need use four encoders to generate key samples and triangular adversarial samples respectively. Func (2) can produce encoder related to $\boldsymbol{x^{key}}$, while Func (3) is used to produce three encoders related to adversarial samples $\boldsymbol{x^{k1}}, \boldsymbol{x^{k2}}, \boldsymbol{x^{k3}}$ respectively.

Q6: Can authors explain more about the second component in Function (4)? What does $\boldsymbol{M^2}$ stand for?

A6: M^2 represents the square value of counting result. Here we use $\boldsymbol{1/\theta_M^2(x_i^k)}$ make the counting value for Triangular Adversarial Samples deviate from the ground truth extremely, which means each pixel of image will approach to the maximum value of 1. The specifical graphical display can **refer to the Fig 6 in Link.**

Reviewer #2:

Our method has great innovation in many aspects. Especially, we introduce the distance mechanism, which can balance the anti-theft ability and the generalization of the model. Furthermore, due to the space limitation, we did omit some details like the crowd counting model, distance ***r*** constrain etc. In particular, we had done extensive experiments in different scenarios to evaluate our model(**refer to Link**). If you think it is necessary, we can supplement it later.

And we will solve the problem of grammar and logic carefully. Thanks for putting forward the problems.

Passive(literature [1]): The attacker can use model normally without author's permission to gain benefits. However the victim can prove the ownership for model through techniques such as watermarking.

Active(our work): The attacker cannot use the model normally without the "key"(encoder). By adversarial training, the model can work only on the x^{key} which is generated by encoder.

Q1: I cannot understand the scenario considered in this work, which is the threat model.

A1: As shown in Figure 1 the owner can deploy the trained model in the cloud. Users upload crowd images to the model via Public API to achieve prediction. The encoder should be deployed as a Private Cloud and secured by owner since it is the key of using model.

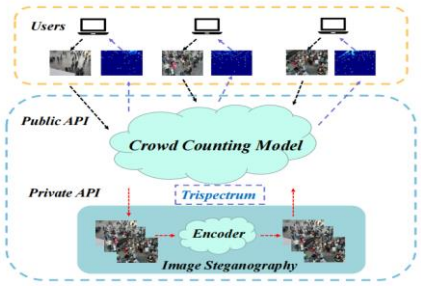

Figure 1. Users use an online crowd counting model with the Trispectrum framework deployed.

Q2: It is not clear that the lowdiversity of the negative samples will result in a high FPR.

A2: Thank you for the pertinent suggestions. We can take this as our future research.

Q3: In which space is the distance between key samples and the original samples measured?

A3: We use following formula to measure the distance between two images where ***m*** is the number of pixels.

$$\mathcal{D}(x_1, x_2) = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(x_{1i} - x_{2i})^2}$$

Q4: Effect of distance *l* is not clear.

A4: Distance *l* is only used to separate the original sample and the key sample, so it is not the focus of the study.

Q5: What if the negative adversarial samples are generated without the distance r constraint, but a constraint of maximizing distance?

A5: Obviously the bigger r will broaden available space of the model, which will make the model vulnerable to evasion attacks.**(Refer to section 5.7 in Link)**

Q6: Original Samples are not involved in loss term (4).

A6: The original sample does not need to participate in adversarial training. All samples required for training need be generated by encoder.