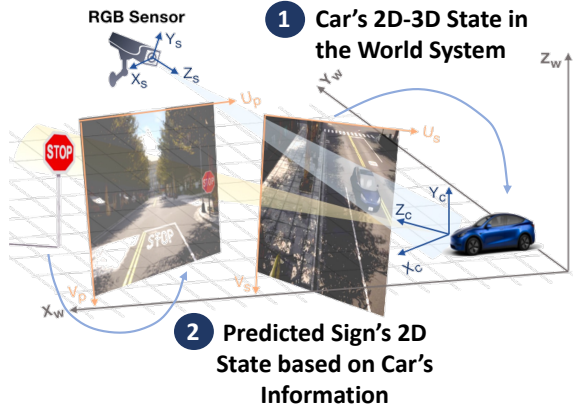


# Supplementary Materials: Embodied Laser Attack: Leveraging Scene Priors to Achieve Agent-based Robust Non-contact Attacks

Anonymous Authors

## 1 LOCAL PERSPECTIVE TRANSFORMATION

As mentioned in Sec. 3 of the manuscript, the perception module faces the challenge of simulating the victim's view through a full-image image transformation for the inherent imaging inconsistencies. To address this, we innovatively abstract complex scenes from pixel-wise to region-based representations based on scene priors, where the scene from the attacker's view can be represented by the vehicle's shape: rectangle, while the scene from the victim's view often adheres to traffic signs' fixed shapes as well, such as circle or octagon. Then we propose a local perspective transformation network to facilitate effective and efficient estimation of the target's distorted states in the victim's view from the attacker's accessible imaging, which builds the relationship between such geometric features that inherently exist at both viewpoints. For such attack-to-victim state inference, what hides behind is a traditional stereoscopic conversion containing two steps based on camera calibration and perspective projection. In this way, the target state in a victim-view 2D image could be estimated through transformation matrices, where the world coordinate systems, the camera coordinate systems, the image coordinate systems, and the pixel coordinate systems should be considered here. For the sake of simplicity, the image and pixel coordinate systems are treated as a singular system in our analysis, owing to their only difference being the definition of the origin. Fig 1 is an illusion of such a conversion process, and we will give a detailed description in the following:



**Figure 1: The essence process of the local perspective transformation executed by PTN.**

The perspective transformation helps us obtain the 2D image-space region of the sign in the foreground of the driving vehicle's shot, which is based on two main relationships: (1) a 3D to 2D mapping of the target sign and (2) a 2D to 3D mapping of the vehicle. Specifically, let  $P = \{p_k | k = 1, \dots, m\}$  denote a 3D dot set of the world coordinate system that represents a traffic sign's contour, and  $Q = \{q_j | j = 1, \dots, n\}$  denote a 2D dot set that denotes the vehicle's

bounding box of its state in scene  $\mathcal{O}_d^t$ . First, for the transformation of the target sign, let  $W(\cdot)$  denote a transformation function from the world to the camera system, which works mainly through a transformation matrix  $L_w$  called view matrix, usually determined by the position, orientation, and upward direction of the camera. Then, for each point  $p_k$  in  $P$  and corresponding coordinates of the same point  $p'_k = W_1(p_k)$  in the vehicle's camera coordinate system, a view relationship between the two representations could be:

$$\begin{bmatrix} p'_k \\ 1 \end{bmatrix}^T = L_{w, \theta_1} \cdot \begin{bmatrix} p_k \\ 1 \end{bmatrix}^T, \quad L_{w, \theta_1} = \begin{bmatrix} R_{\theta_1} & T_{\theta_1} \\ 0 & 1 \end{bmatrix}, \quad (1)$$

where  $R_{\theta_1}$  is a  $3 \times 3$  rotation matrix (attitude information), and  $T_{\theta_1}$  is a  $3 \times 1$  translation matrix (position information),  $\theta_1$  denotes parameters about vehicle's sensor: for rotation, it denotes the rotation angles around the three axes; for translation, it denotes the relative position of the sensor. Then we could get a new dot set  $P' = \{p'_k | k = 1, \dots, m\}$  that denotes the sign in the camera system of the victim's sensor. Additionally, there also exists a transformation between the image and the camera system called perspective projection, which is here denoted as  $H(\cdot)$ . And the new 2D coordinate  $p''_k = H_1(p'_k)$  could be essentially calculated by a perspective projection matrix  $L_h$  as:

$$p'_{k_z} \cdot \begin{bmatrix} p''_k \\ 1 \end{bmatrix}^T = L_{h, \theta_{1*}} \cdot \begin{bmatrix} p'_k \\ 1 \end{bmatrix}^T, \quad (2)$$

where  $\theta_{1*}$  denotes the vehicle sensor's internal reference, and  $z$  denotes the component axis. The whole mapping from the 3D world system to 2D key region in the vehicle sensor's imaging is as follows, where we could get the sign representation 2D dot set  $P''$  to further reduce the sign's binary region  $M_P$  and then determine its content:

$$P'' = H_1(W_1(P)) \quad (3)$$

During the above steps, we need the victim's accurate information  $\theta_1$  and  $\theta_{1*}$  for matrix  $L_w$  in Eq.(1) and  $L_h$  in Eq.(2) to promise a valid prediction  $P''$  on the sign's distorted state. To address this, it should first conduct a similar 2D to 3D mapping of the vehicle's state with the vehicle representation dot set  $Q$ . We attempt to get the vehicle's real 3D coordinate  $Q''$  through inverse matrix operations:  $W_2$  and  $H_2$ , and the conversion for point  $q_j$  is a changing form of Eq.(3):

$$\begin{bmatrix} q''_j \\ 1 \end{bmatrix}^T = L_{w, \theta_2}^{-1} \cdot L_{h, \theta_{2*}}^{-1} \cdot q'_{j_z} \cdot \begin{bmatrix} q_j \\ 1 \end{bmatrix}^T, \quad (4)$$

where  $\theta_2$  and  $\theta_{2*}$  are the third-party sensor's accessible external and internal references. Since this sensor is manually arranged, they are known terms to us, leading to a direct calculation if we obtain the victim's information.

It's obvious that once the position and orientation change, the vehicle's parameters will also change and the process needs to be calculated again, and it's impracticable to obtain parameters for

each frame individually. What's more, some values for matrix calculations (like  $q'_{j_z}$ ) are hard to reach with a single sensor, resulting in an inability to infer. In contrast, we do not need to convert between coordinate systems by step-by-step calculations with difficulty in obtaining parameter values. Our perception module only needs to learn specific regular patterns of the geometric prior once in this dynamic continuous scene, which can respond repeatedly and quickly during application. In this way, we could easily get the geometry feature in scene  $\mathbf{o}_v^t$  with specificity for each frame: contour, and then bring out the target content of the sign's region under the victim's perspective for the decision stage.

## 2 DETAILS OF EXPERIMENTS

### 2.1 Hyperparameter Settings

There are also several variables that should be assigned, especially for agent training. By default, we set  $r_s$  as 10000, and  $\alpha$  as 100 in  $R_{\text{attack}}$ , we also set  $r_\omega$  and  $r_\lambda$  as 100 in  $R_{\text{appear}}$ . Among them, if the attack is conducted by one step,  $r_s$  will change to 50000 as a better reward. As for  $\omega_0$ , to control the light width, it is assigned as 10 and only 15 for Sign "Stop" since its region is larger with more exclusive characteristics such as shape, which requires a higher attack level. Due to the different optimization difficulties of the attack on various classifiers, we have only selected the settings of one of the cases as an example, and they all can be adjustable based on experience and actual situations, which is related to reinforcement learning settings and is not our core focus.

### 2.2 Details of PTN Training

In our experiments, we have directly chosen suitable weights for perspective inference to test the validity of the whole framework ELA, while we didn't mention the training details of PTN. Specifically, we initially set *epoch* as 500 with loss records and save models regularly. Figure 2 illustrates an example of the trend of training loss, where we can find that the training loss quickly reaches a low but not stable trend in the first fifth of training. And the weights obtained in the later stage when the loss is more stable and lower can also be found to have a more stable reasoning effect during validation. For this reason, we choose to sacrifice some training

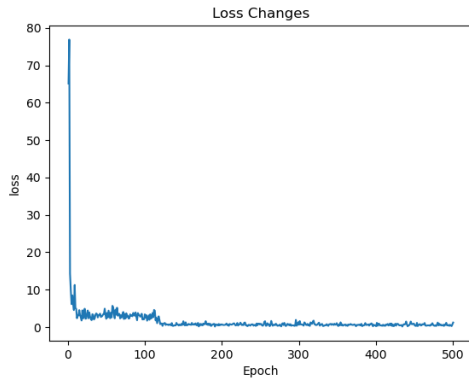


Figure 2: One record of the PTN training process.

time in exchange for more stable weights, which have been used in our attack framework, to ensure effective inference on the test set.

### 2.3 Discussions on Adversarial Examples

**Semantic Information:** Since we have counted the frequency rank of wrong labels after our attack in Sec. 4.5, we want to analyze whether there is a semantic characteristic pattern in the adversarial examples of successful attacks. Figure 3 demonstrates several successful examples, where some of the semantic characteristics of the attacked images are relatively easy to understand. The new features are intuitively similar to those wrong labels, such as the sign "30" being misled to "70" since the laser beam enables a character "seven" to pretend to be in the image, or the misclassified sign "90" with a beam that causes a somewhat similar appearance to the number "six". Even if there also exist some results that cannot be directly explained, we suppose that it must have a similarity to the mislabeled feature map in classification layers after being abstracted into a high-level representation by the DNNs.

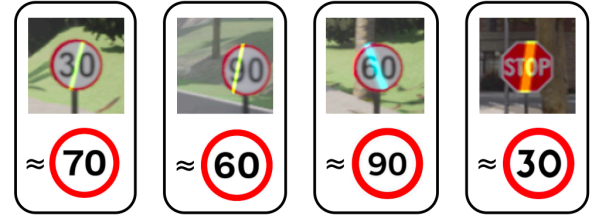


Figure 3: Successful attack examples with misclassification labels of different categories.