Supplementary Material
Adversarial Attention Deficit: Fooling Deformable Vision Transformers with Collaborative Adversarial Patches

1. Intuition of the attacks

In Deformable DeTr, for each query a handful of the keys are considered for attention, and this is completely a data-dependent process (depends on the features of the query token). Consequently, there isn't a singular set of key tokens functioning collectively as a patch, where all query tokens focus their attention simultaneously. To solve this, first the source patch has to manipulate the directions of attention of all the query tokens to focus onto a region of the image, where then a target adversarial patch is placed for affecting model loss. The source patch in turn magnifies the target patch, amplifying its impact on the system, and that is the reason why a small patch is sufficient for our attacks.

2. Description of attacks in MVDeTr

We begin by defining the projection matrices denoted as $[M_1, \ldots, M_7]$, which maps points from image coordinates to world coordinates. The computation involves the following steps:

$$M_i = (I_i @ E_i @ Tx)^{-1} \tag{1}$$

Where, @ represents the the matrix multiplication operation and -1 represents the matrix inversion operation. Also, I_i and E_i is the intrinsic and extrinsic matrices respectively for view i, and Tx is the transformation matrix that maps 3D world coordinates to 2D image coordinates, and is defined as:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & z \\ 0 & 0 & 1 \end{bmatrix}$$

This matrix accounts for the depth z and transforms 3D coordinates to 2D coordinates in homogeneous coordinates. The resulting projection matrix M_i maps 2D image coordinates to 3D world coordinates.

We use these projection matrices to project source locations S_i and target locations T_i for view i onto the ground plane, where the shadow transformer processes all seven ground-plane projections separately but simultaneously (please refer to the original literature [13] for additional details on the shadow transformer). Subsequently, the in-pointer loss (Eqn. 2), out-pointer loss (Eqn. 3) and attention loss (Eqn. 4) are defined like the following:

$$\mathcal{L}_{in} = \sum_{l=1}^{L}\sum_{h=1}^{H} \frac{1}{DQK} \sum_{q=1}^{Q}\sum_{d=1}^{D}\sum_{k=1}^{R} (p_q + \Delta p_{lhqdk} - T_k)^2 \tag{2}$$

$$\mathcal{L}_{out} = -\sum_{l=1}^{L}\sum_{h=1}^{H} \frac{1}{DQK} \sum_{q=1}^{Q}\sum_{d=1}^{D}\sum_{k=1}^{R} (p_q + \Delta p_{lhqdk} - T_k)^2 \tag{3}$$

$$\mathcal{L}_{att} = -\sum_{l=1}^{L}\sum_{h=1}^{H} \frac{1}{DQK} \sum_{q=1}^{Q}\sum_{d=1}^{D}\sum_{k=1}^{R} A_{lhqdk} \tag{4}$$

In the case of DeTr, D in the losses represented number of different image scales (resolutions of the original image) taken into account, and for MVDeTr it represents the number of different views (from cameras recording the shared world space) in the system.

3. Variants of Deformable DeTr

Below are the description of the abbreviations used for different variants of Deformable DeTr. Please refer to the original literature [24] for additional details:

- ***DD-SS* `or` Deformable DeTr (single scale):** This version refers to the utilization of only the res5 feature map (with a stride of 32) as input feature maps for the Deformable Transformer Encoder.
- ***DD-SS-DC5* `or` Deformable DeTr - single scale, DC5:** In addition to the single scale this version includes the removal of the stride in the C5 stage of ResNet and replacing it with a dilation of 2.
- ***DD-Base* `or` Deformable DeTr:** Base version of Deformable DeTr trained with total batch size of 32.
- ***DD-IBBR* `or` Deformable DeTr - iterative bounding box refinement:** This version gets inspiration from iterative refinement methods seen in optical flow estimation, where each decoder layer iteratively refines bounding boxes using predictions from the preceding layer, resulting in improved detection performance.

- ***DD-TS* or Deformable DeTr - two stage:** This version is influenced by two-stage object detection methods, in which region proposals generated in an initial stage is utilized as object queries within the decoder for additional refinement, leading to a two-stage Deformable DeTr.

4. Effect of changing attack parameters

Impact of Locations of Patches. Fig. 1 demonstrate the attack performance with respect to the locations of the patches. For all of the attacks, we either distributed the patches uniformly or randomly over the image or video frame. The source and target patches were distributed separately, and in the case of uniform distribution, we ensured non-overlap between them by applying a slight offset. We observe that uniform distribution performs better than random distribution, as it guarantees equal accessibility of the patches across the entire frame.

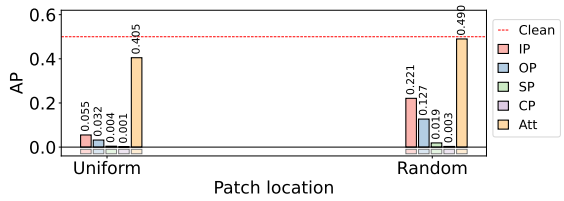

Figure 1. Impact of patch location

Impact of a Source Aggregator. We explored a case where multiple source patches are placed at various locations. Instead of developing patches specific to each location, our aim was to design a universal patch effective from any of these locations. This approach has real-world implications for the practicality of these attacks. For instance, a single generic patch, like one printed on a T-shirt, could deceive the model if positioned in different areas of a scene or if the same patch appears at different locations in different cameras in a multi-view scenario. We designed two different gradient aggregator; (1) Mean: takes the average of the gradients, and (2) Max-norm: takes the gradients with the highest L2-norm. In our experiments (see Fig. 2), using an aggregator do not affect attack performance and is equally effective as collecting gradients individually.

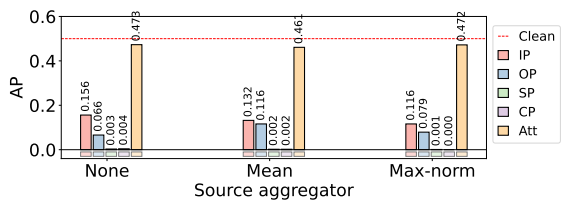

Figure 2. Impact of source aggregator

Impact of Patch Count in MVDeTr In this experiment, we implemented our CP attack to evaluate the effect of number of patches on attack performance. Also, all the patches used in this experiment were 32x32. In the CP(4tar) setup, we placed target patches in four views (from view 4 to 7). Subsequently, we systematically added one source patch to each of the views, starting with view 0. Similarly, in the CP(4src) setup, we first placed source patches in four views and then incrementally added target patches. For the CP(7tar) and CP(7src) setups, as opposed to four views, we placed target/source patches in all seven views respectively, before introducing patches of the other type. As shown in Fig. 3, the attacks successfully compromised the system even when the source and target views were disjoint (for instance, CP(4tar) and CP(4src) with patches of other type in views 1 to 3). In CP(7tar) and CP(7src) setup, the attacks were extremely effective, reducing the Multi-view Object Detection Accuracy (MODA) to 0%.

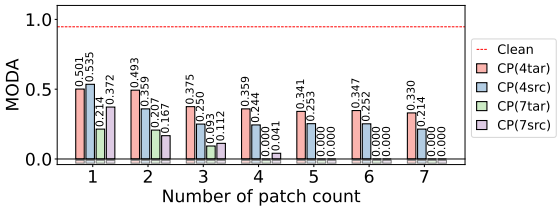

Figure 3. Patch count (multi-view)

Impact of Patch Size in MVDeTr In this experiment, we executed both SP and CP attacks to evaluate the effect of patch size on attack performance. Also, we limited the adversarial access to only three cameras. Like before, in the SP attack, each adversarial camera was equipped with a single patch. In contrast, the CP attack involved two patches per camera, one source and one target. Furthermore, we explored two variations within the CP attack. In the CP(tar) setup, we increased the size of the target patch while keeping the source patch size constant (32x32), and in the CP(src) setup, we enlarged the source patch while the target patch size remained the same (32x32). As depicted in Fig. 4, the effectiveness of the attacks increased with the size of the patches.

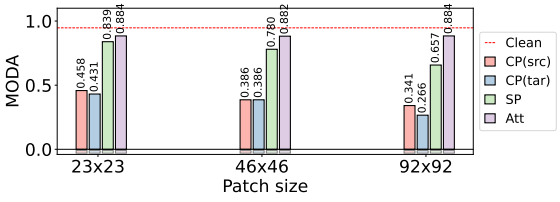

Figure 4. Patch size (multi-view)

Effect of Constant Adversarial Pixel Count in MVDeTr In this experiment, as the number of adversarial cameras

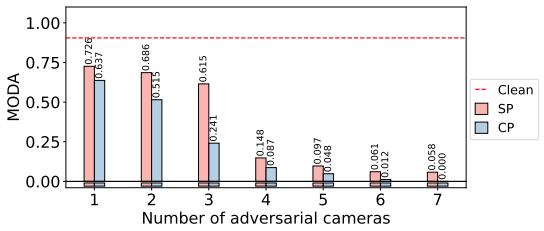

Figure 5. Impact of constat adversarial pixel count

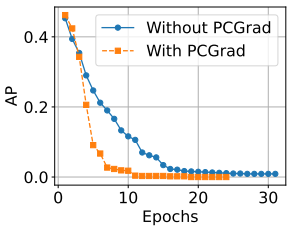

Figure 6. AP comparison over epochs with and without PCGrad

increases, the total number of adversarial pixels remains constant. For instance, in the SP attack, when all cameras are compromised, each camera uses a 46x46 patch. However, when only one camera is compromised, the patch is scaled up to 120x120 to maintain the same total number of adversarial pixels. Similarly, for the CP attack, patch sizes are adjusted to match the total adversarial pixel count of the SP attack. We observe that although the number of adversarial pixels are constant across all experiments, due to the dependency on number of source and target patch count, we have a more effective attack when there are more patches in the attack (as shown in Fig. 5).

5. Effect of PCGrad

In all of our attacks, we perform some sort of multi-task learning. For example, in all of the attacks we maximize the pointer loss $\mathcal{L}_{in}$ (Eqn. 6) or $\mathcal{L}_{out}$ (Eqn. 8) and attention loss $\mathcal{L}_{att}$ using the source patch. Specially, in our SP attack (source and target patches are the same), in addition to these losses we also maximize the model loss L_{model} on the same patch. In such multi-task learning cases, it is common for the gradient of different losses to interfere with one another, and cause the learning to be inefficient. Existing solution proposes a technique known as gradient surgery PCGrad [23], which involves projecting the gradient of a task onto the normal plane of the gradient of any other task that possesses a gradient. We utilize this method in all of our attacks where there is multi-task learning involved. This ensures our attacks converge faster, however, our attacks do not rely on PCGrad for its effectiveness. Fig. 6 shows the training curve of attack performance with and without PCGrad.

6. Effectiveness of whole-image noise

Previous literature [3, 17, 18] already investigated the effect of whole-image constrained noises on transformer-based vision models. These works concluded that owing to the unique relation modeling capabilities of transformer architecture these types of attacks are not as effective as in CNN-based models. We further investigate the impact of whole-image noise on our target model Deformable DeTrs [24], confirming the findings consistent with previous studies. In Tab. 1 we present the performance of whole-image noise under different L_2 and L_∞ constraints. We standardize our input images, setting their mean value to `(0.485, 0.456, 0.406)` and their standard deviation to `(0.229, 0.224, 0.225)`. We find that, within a noise budget that does not induce occlusion, the model performance is unaffected (AP on clean sample is 50%) with whole-image noise.

	$\epsilon = \frac{8}{255}$	$\epsilon = \frac{16}{255}$	$\epsilon = \frac{32}{255}$	$\epsilon = \frac{64}{255}$	$\epsilon = \frac{128}{255}$	$\epsilon = \frac{255}{255}$
L_2	50.0%	50.0%	50.0%	50.0%	50.0%	50.0%
L_∞	50.0%	50.0%	50.0%	46.9%	36.5%	17.1%

Table 1. AP under whole-image attacks

7. Perceptibly of our patches

In all of our attacks, altering less than 1% of the patched area in the input field severely impacts model performance. When the adversarial patch is placed on test images, it can be barely visible in the resulting images, and can pass as a faint camera flicker, or a minuscule pattern printed on a surface (e.g. T-shirt, signboard, etc.). For example, Fig. 7 illustrates two instances of clean and patched image-pairs, with a source patch at `(400, 400)` and a target patch at `(100, 100)`. In this experiment, we took two `32x32` patch, but this attack can be realized with a `16x16` or even a `8x8` patch, making the patches almost imperceptible.

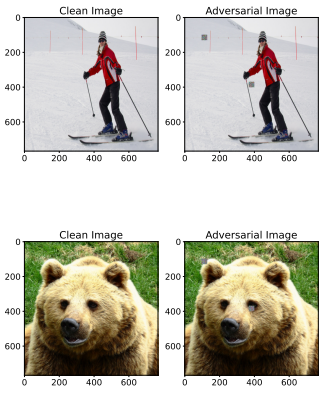

Figure 7. Comparison of clean and adversarial images. The patches are less than 1% of the total area.

8. Illustration of attack on Deformable DeTr

All of the attacks have a catastrophic effect on the bounding-box results of Deformable DeTr. In the detection results of a patched image, most often bounding boxes are totally absent (no objects found). In Fig. 8 and 9, we present the bounding-box results on clean and adversarial images using SP and CP attacks respectively.

Figure 8. Bounding box comparison among ground-truth, clean images, and adversarial images in SP attack.

Figure 9. Bounding box comparison among ground-truth, clean images, and adversarial images in CP attack.

9. Illustration of attack on MVDeTr

Our attacks also demonstrate excellent effectiveness against MVDeTr, leading to a complete disruption in the bounding-box results. In the detection outcomes across all seven views, we consistently observe either the absence of bounding boxes around objects or misaligned boxes appearing in random areas of the images. This underscores the impact of our attacks on both object classification and localization tasks in object detection. In Fig. 10 and 11, we present the bounding-box results on ground-truth and adversarial in all seven views using SP and CP attacks respectively.

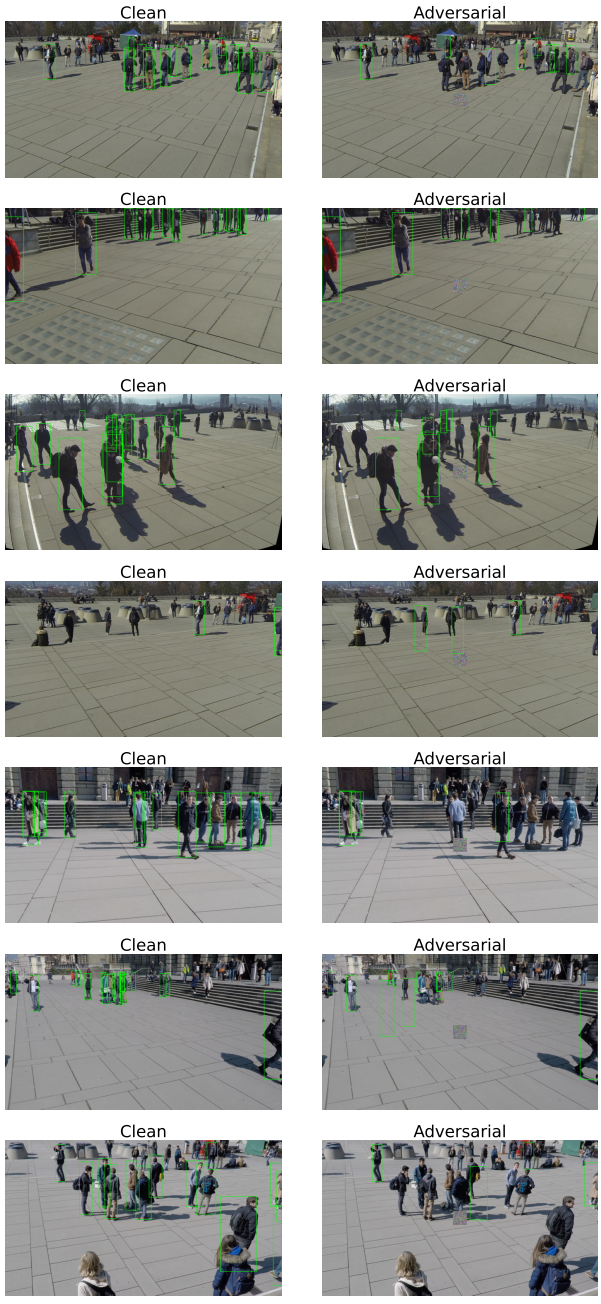

Figure 10. Bounding box comparison among seven cameras in clean and adversarial in SP attack.

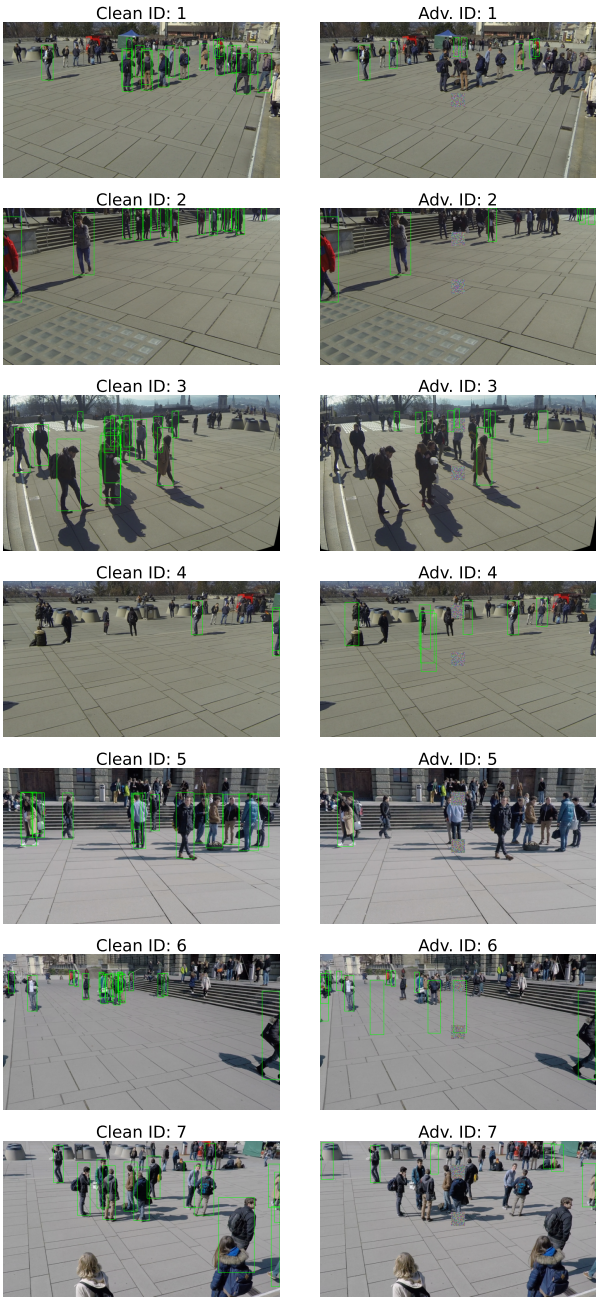

Figure 11. Bounding box comparison among seven cameras in clean and adversarial in CP attack.

10. Illustration of attention heatmap in Deformable DeTr

In addition to the figure in Introduction (Sec. 1), we also show heatmap of sparse attention for two additional images using our SP and CP attack in Fig. 12 and 13. As previous, the top and the bottom row shows the directions and the values of sparse attention respectively, across the tokens of the image with respect to the targeted area (marked in red). As the attack progresses (left to right), there is a noticeable increase in the amount of attention being redirected towards the targeted area, along with a rise in the attention values for both of our attacks.

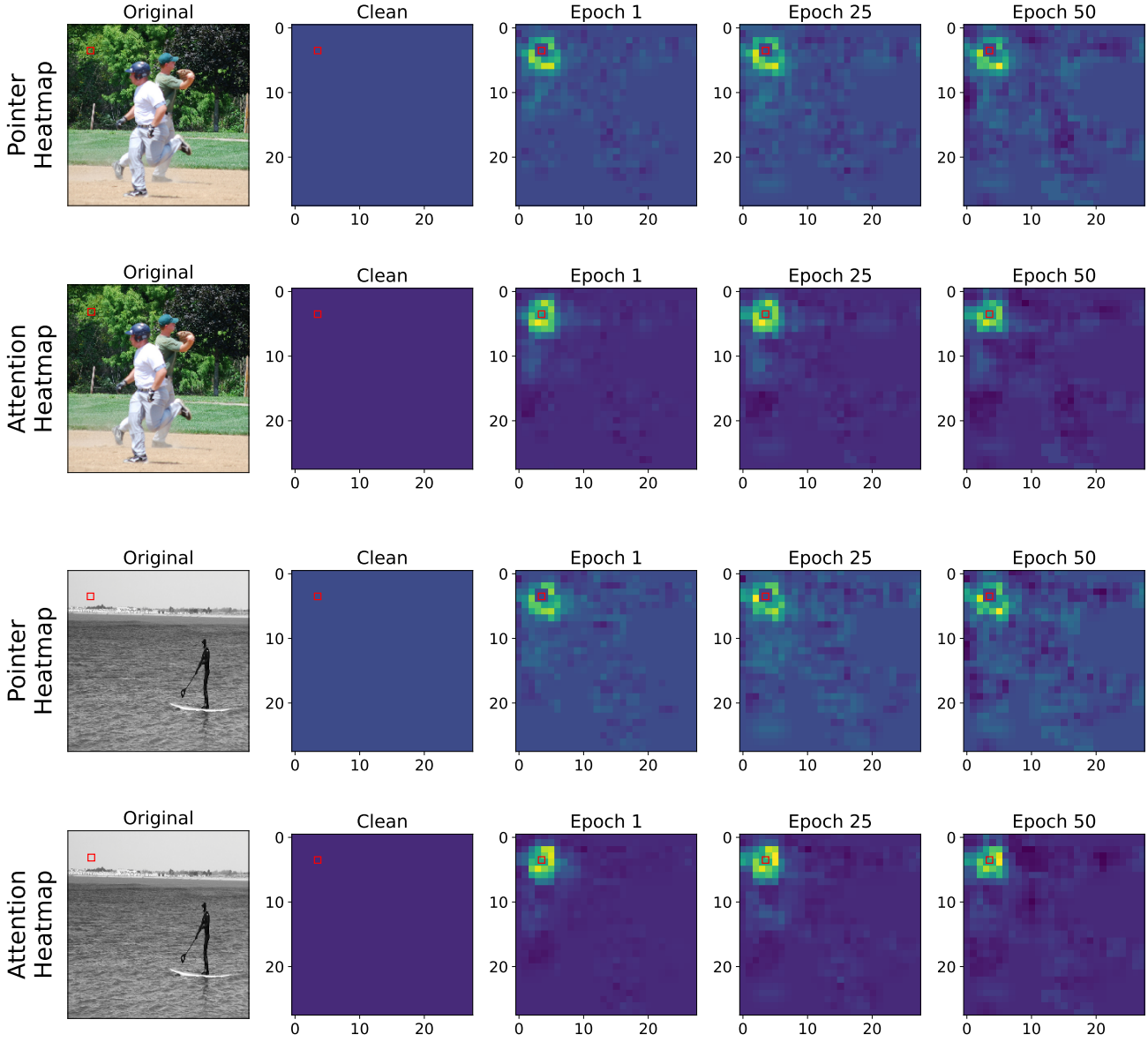

Figure 12. Directions and values of attention focused toward the target patch for SP attack

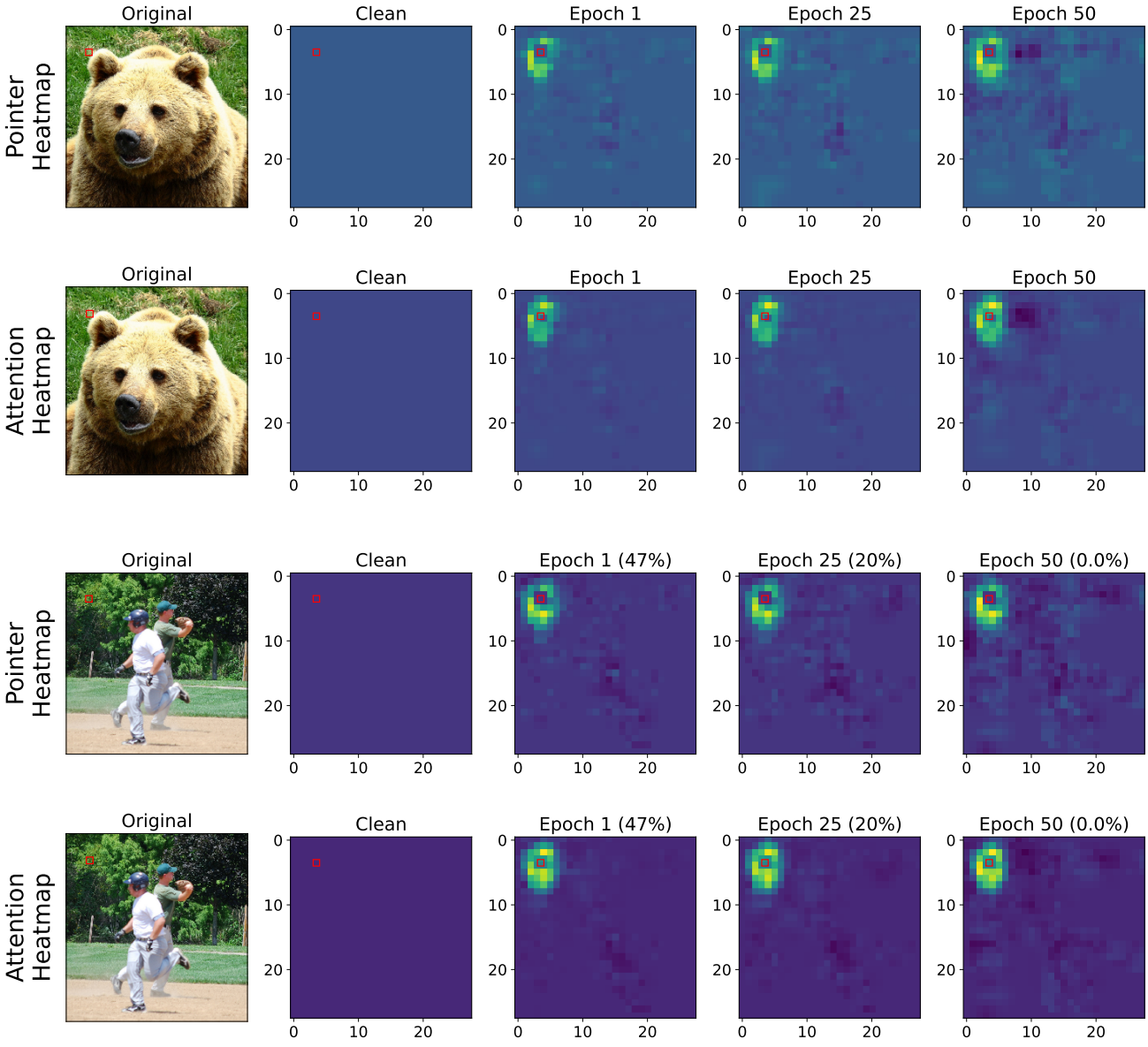

Figure 13. Directions and values of attention focused toward the target patch for CP attack

11. Illustration of ground-plane heatmap in MVDeTr

In addition to the figure in Evaluation (Sec. 4), we also show the heatmap of the ground plane of the shared world space in MVDeTr using all four of our attacks in Fig. 14. The points in the ground plane represent the top-view position of the objects in multi-view detection. We observe that for our SP and CP attacks the heatmap is significantly different from the ground-truth or clean heatmap. With a patch of adequate size, the impact of our attacks on the ground-plane heatmap is readily apparent.

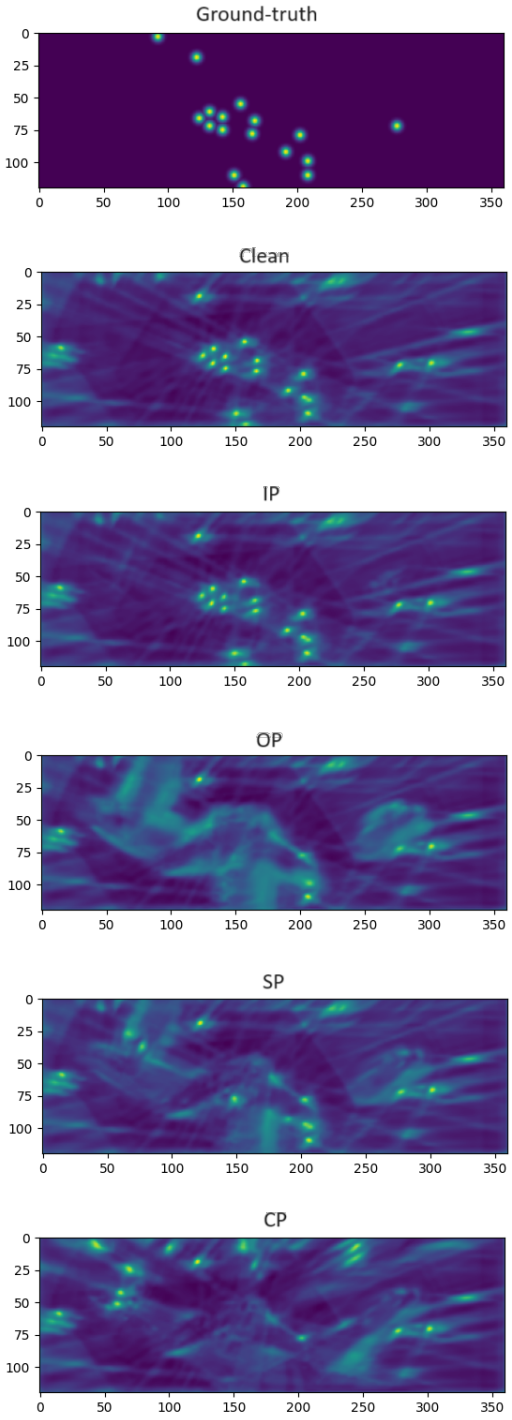

Figure 14. Ground plane heatmap of the shared world space in MVDeTr