Supplementary Material

A. Gradient of Dot-Product Attention - Results

We report an extended version of Table 1 in Table A2, which includes up to 12 encoder layers (when available) and report the median of the gradient ratio $|(\nabla_X A_h(X))X)/(A_h(X)1_X)|$, more specifically mean and its standard error of this median across 100 images. Table A2 shows that gradients tend to flow along the $A_h(X)1_X$ gradient term, in particular when backpropagating all the way up to the input (as in adversarial patch optimization). We also find that smaller ratios are connected to less effective patches, in fact ViT-* models tend to have large ratios (in particular in later layers) compared to DeiT-*, which matches lower robust accuracies in general (see Table 3).

B. Effect of Input Mean on Attention Weight Robustness

As discussed in in Section 4.2, less centered inputs X (larger absolute value of input mean $|\mu|$) make dot-product self-attention less robust to patch attacks on the controlled setting (when input variance is constant). We recap that the input mean is $\mu = \mu \cdot \mathbf{1}$, input standard deviation $\sigma = \mathbf{1}$, and $W_Q = -w \cdot \mathbb{I}_{d_k}$ and $W_K = w \cdot \mathbb{I}_{d_k}$.

We now denote the query mean by $\mu_Q = \mathrm{Mean}(W_Q \cdot X) = -w\mu$ and key mean by $\mu_K = \mathrm{Mean}(W_K \cdot X) = w\mu$. We note that the element-wise distance between query and key mean is $|\mu_K^{(i)} - \mu_Q^{(i)}| = |w\mu_i + w\mu_i| = 2|w||\mu|$. On the other hand, for query standard deviation, we have $\sigma_Q = \mathrm{StdDev}(W_Q \cdot X) = \mathrm{StdDev}(-wX) = w\mathbf{1}$ and for key standard deviation $\sigma_K^2 = \mathrm{StdDev}(W_K \cdot X) = \mathrm{StdDev}(wX) = w\mathbf{1}$.

As can be seen, increasing $|\mu|$ increases the distance between query and key cluster mean, while leaving the key's and query's standard deviation unchanged. As a result, it increases the separation of key and query clusters (see also Figure 1). On the other hand, increasing $|w|$ has no systematic effect on the separation of query and key cluster, as w's effect on mean and standard deviation cancels out.

We empirically quantify the query and key cluster separation using the Silhouette score [32]. Figure A1 shows this score as a function of the input mean scale $|\mu|$ in the controlled setting. The plot confirms above's theoretical argument: increased input mean results in more separated keys and queries.

The result of this separation of keys and queries is that attention drawn by one key can increase for all queries when moving the key in the direction of the query mean (because in a sense, all queries lie in the same direction from the key as long as they are distant and have small variance). On the other hand, if keys and queries lie intermingled, than for any direction the key moves, it will get closer to some queries at the expense of increasing distance to other queries. Because of this, for an adversarial patch attack on the attention weights, it is beneficial if keys and queries are well separated (as in Figure 1).

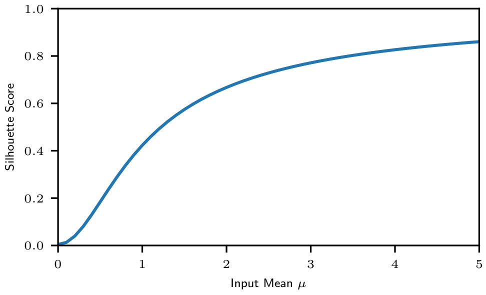

Figure A1. Silhouette score between clusters of projected keys and queries for different choices of μ on synthetic data. Larger scores corresponds to keys and queries forming more distinct clusters.

C. Ablation on Attention-Fool on ViT

In this section, we perform a series on ablations on Attention-Fool losses on ViT. Sections C.1, C.3 and C.2 ablate on properties of the loss, while Section C.4 and C.5 on properties of the adversarial patch.

C.1. Reduction over Heads and Layers

As discussed in Section 5, when multiple attention layers and attention heads are present, one can aggregate per-layer and per-head $\mathcal{L}_{kq}^{hl}$ in different ways. To study the influence of these aggregation choices, in this section we empirically test three of them: (i) smax (as in Section 5), (ii) hard maximum max and (iii) mean. Since we are aggregating along two dimensions, layers and heads, this sparks nine combinations of the three aggregations, which we empirically test on ViT models. As shown in Table A1, across different aggregation methods, choosing smax over both layers and heads generally performs well, resulting in low robust accuracies. Intuitively, using smax gives more flexibility to the optimization to choose the most vulnerable layers and heads. We also find that mean aggregation across both layers and heads is an effective choice, presumably in part due to the fact that in ViT inputs to attention layers are preemptively normalized, leading to smaller difference in per-head per-layer behaviour. In comparison the hard maximum max seems to be the weakest choice, leading to consistently worse results. We note from Table A1 that the best combination is rather model-specific, which suggests that a similar ablation study could be run on a model-basis. In fact, we will show in the next section how on DETR [8], Attention-Fool with max aggregation results in better performance.

Table A1. Robust accuracies (%) under adversarial patch attacks with Attention-Fool losses when choosing different ways of aggregating $\mathcal{L}_{kq}^{hl}$ across encoder layers and attention heads. All rows are computed using PGD^{250} with momentum and step size α=8/255. Numbers in parenthesis report the improvement or degradation in robust accuracy w.r.t. the smax,smax default choice outlined in Section 5.

L reduction	H reduction	ViT-T	ViT-B	ViT-B-384	DeiT-T	DeiT-B	DeiT-B-384
smax	smax	0.00	0.10	2.50	0.00	19.30	39.80
smax	mean	0.00 (-0.00)	0.20 (+0.10)	2.70 (+0.20)	0.10 (+0.10)	17.80 (–1.50)	40.60 (+0.80)
smax	max	0.00 (-0.00)	3.30 (+3.20)	5.80 (+3.30)	0.00 (-0.00)	36.60 (+17.30)	45.90 (+6.10)
mean	smax	0.00 (-0.00)	0.00 (–0.10)	2.90 (+0.40)	0.00 (-0.00)	18.30 (–1.00)	40.40 (+0.60)
mean	mean	0.00 (-0.00)	0.10 (-0.00)	3.50 (+1.00)	0.00 (-0.00)	17.50 (–1.80)	39.60 (–0.20)
mean	max	0.00 (-0.00)	2.40 (+2.30)	9.20 (+6.70)	0.00 (-0.00)	23.80 (+4.50)	43.60 (+3.80)
max	smax	0.10 (+0.10)	15.60 (+15.50)	26.70 (+24.20)	1.20 (+1.20)	27.10 (+7.80)	45.70 (+5.90)
max	mean	0.20 (+0.20)	13.80 (+13.70)	24.60 (+22.10)	1.50 (+1.50)	26.10 (+6.80)	46.10 (+6.30)
max	max	3.70 (+3.70)	45.50 (+45.40)	59.70 (+57.20)	1.50 (+1.50)	58.50 (+39.20)	58.80 (+19.00)

Table A2. Median of $|(\nabla_X A_h(X))X)/(A_h(X)1_X)|$ over tokens and heads, for models on 12 encoder layers. We report the mean of the ratio medians and its standard error over 100 randomly selected images from the MS COCO 2017 validation set [22] for DETR (DC5-R50), and over 100 randomly selected images from the ImageNet 2012 [33] dataset for all remaining models.

	DETR	ViT-T	ViT-S	ViT-B	DeIT-T	DeIT-S	DeIT-B
Layer 1	0.208 ± 0.008	0.079 ± 0.001	0.042 ± 0.001	0.072 ± 0.001	0.035 ± 0.000	0.045 ± 0.001	0.053 ± 0.001
Layer 2	0.035 ± 0.001	0.040 ± 0.001	0.051 ± 0.000	0.080 ± 0.001	0.045 ± 0.001	0.045 ± 0.000	0.044 ± 0.001
Layer 3	0.039 ± 0.001	0.031 ± 0.000	0.027 ± 0.000	0.047 ± 0.000	0.044 ± 0.001	0.033 ± 0.000	0.042 ± 0.000
Layer 4	0.066 ± 0.001	0.043 ± 0.001	0.028 ± 0.000	0.037 ± 0.000	0.035 ± 0.000	0.032 ± 0.000	0.057 ± 0.000
Layer 5	0.098 ± 0.001	0.037 ± 0.001	0.029 ± 0.000	0.033 ± 0.000	0.035 ± 0.000	0.033 ± 0.000	0.044 ± 0.000
Layer 6	0.147 ± 0.002	0.044 ± 0.001	0.027 ± 0.000	0.044 ± 0.001	0.070 ± 0.001	0.033 ± 0.000	0.040 ± 0.000
Layer 7	-	0.036 ± 0.000	0.031 ± 0.000	0.040 ± 0.000	0.045 ± 0.001	0.033 ± 0.001	0.040 ± 0.000
Layer 8	-	0.045 ± 0.001	0.046 ± 0.001	0.046 ± 0.001	0.050 ± 0.001	0.037 ± 0.001	0.039 ± 0.000
Layer 9	-	0.171 ± 0.005	0.087 ± 0.002	0.064 ± 0.001	0.076 ± 0.002	0.052 ± 0.001	0.049 ± 0.001
Layer 10	-	0.308 ± 0.006	0.126 ± 0.003	0.061 ± 0.001	0.082 ± 0.002	0.077 ± 0.002	0.083 ± 0.002
Layer 11	-	0.428 ± 0.009	0.280 ± 0.006	0.099 ± 0.002	0.080 ± 0.002	0.100 ± 0.003	0.192 ± 0.005
Layer 12	-	0.442 ± 0.013	0.469 ± 0.014	0.249 ± 0.006	0.212 ± 0.008	0.198 ± 0.009	0.104 ± 0.004

Table A3. Robust accuracy (%) of different vision transformers when choosing l in $\mathcal{L}_{kq}^{l}$, which targets a specific encoder layer (see Section 5). Selecting l leads to worse results compared to the baseline loss $\mathcal{L}_{kq}$.

	ViT-T	ViT-B	DeiT-T	DeiT-B
$\mathcal{L}_{kq}$	0.0	0.1	0.0	19.3
$\mathcal{L}_{kq}^{(l)}, l = 1$	7.5 (+7.5)	0.0 (-0.1)	14.0 (+14.0)	36.3 (+17.0)
$l = 2$	4.4 (+4.4)	39.1 (+39.0)	2.2 (+2.2)	56.1 (+36.8)
$l = 3$	0.2 (+0.2)	21.4 (+21.3)	1.0 (+1.0)	43.5 (+24.2)
$l = 4$	0.0 (-0.0)	17.2 (+17.1)	0.8 (+0.8)	22.1 (+2.8)
$l = 5$	0.0 (-0.0)	18.8 (+18.7)	1.3 (+1.3)	23.9 (+4.6)
$l = 6$	0.0 (-0.0)	12.4 (+12.3)	0.0 (-0.0)	24.6 (+5.3)
$l = 7$	0.0 (-0.0)	9.4 (+9.3)	0.0 (-0.0)	23.9 (+4.6)
$l = 8$	0.1 (+0.1)	3.2 (+3.1)	0.6 (+0.6)	18.4 (-0.9)
$l = 9$	0.0 (-0.0)	0.8 (+0.7)	0.1 (+0.1)	21.1 (+1.8)
$l = 10$	0.1 (+0.1)	1.3 (+1.2)	0.1 (+0.1)	23.5 (+4.2)
$l = 11$	0.0 (-0.0)	1.1 (+1.0)	0.2 (+0.2)	23.1 (+3.8)
$l = 12$	0.0 (-0.0)	1.0 (+0.9)	1.2 (+1.2)	25.4 (+6.1)

Table A4. Robust accuracy (%) of different vision transformers for adversarial patches of different sizes for $\mathcal{L}_{kq^\star}$. As the patch dimension shrinks to 8×8 (corresponding to 0.13% of total image pixels), the robust accuracies increase as expected.

Patch Size	ViT-T	ViT-B	DeiT-T	DeiT-B
16×16	0.0	0.1	0.0	13.1
14×14	0.0	1.9	2.3	24.2
12×12	0.0	6.3	17.6	39.2
10×10	5.9	22.3	38.0	52.8
8×8	34.7	50.1	51.2	65.3
Clean Accuracy	73.6	85.0	69.5	82.0

C.2. Choosing Specific Layer

Section 5 introduces per-layer, l, and per-head, h, losses identified by $\mathcal{L}_{kq}^{hl}$, but the evaluation of Section 6.2 focuses on the aggregated loss $\mathcal{L}_{kq}$ and its version targeting the special token $\mathcal{L}_{kq^\star}$ To study the effectiveness of targeting specific encoder layers in the loss, here we compare using single-layer $\mathcal{L}_{kq}^{l}$ with the aggregated $\mathcal{L}_{kq}$; we report the results in Table A3. Targeting a single l via $\mathcal{L}_{kq}^{l}$ is typically weaker than targeting all jointly via $\mathcal{L}_{kq}$. In addition, it is unclear how to choose l a priori without trying all options as there is no common pattern across models.

Table A5. Robust accuracies (%) under Attention Fool adversarial patch attack using the un-normalized P_Q^{hl} and P_K^{hl} introduced in Section 5. All rows are computed using PGD^{250} with momentum and step size α=8/255. Without normalization, $\mathcal{L}_{kq}$ and $\mathcal{L}_{kq^\star}$ do not improve on $\mathcal{L}_{ce}$ baselines and in fact perform much worse, likely because individual $\mathcal{L}_{kq}^{hl}$ losses are not commensurable with each other.

	ViT-T	ViT-B	ViT-B-384	DeiT-T	DeiT-B	DeiT-B-384
$\mathcal{L}_{ce}$	0.10	13.50	31.20	19.80	36.00	58.80
$+\mathcal{L}_{kq}$	53.10 (+53.00)	81.70 (+68.20)	84.50 (+53.30)	67.40 (+47.60)	79.80 (+43.80)	80.60 (+21.80)
$+\mathcal{L}_{kq*}$	24.40 (+24.30)	79.80 (+66.30)	82.00 (+50.80)	49.00 (+29.20)	79.60 (+43.60)	80.80 (+22.00)
+ Momentum	0.00	3.10	13.20	1.50	16.80	41.70
$+\mathcal{L}_{kq}$	50.00 (+50.00)	81.60 (+78.50)	84.30 (+71.10)	66.80 (+65.30)	78.30 (+61.50)	80.30 (+38.60)
$+\mathcal{L}_{kq*}$	21.10 (+21.10)	80.30 (+77.20)	82.10 (+68.90)	24.10 (+22.60)	78.60 (+61.80)	80.10 (+38.40)

Table A6. Robust accuracies (%) of ResNet50 and different vision transformers with the adversarial patch positioned at the center of the images. The evaluation setting is similar to Table 3 except for the position of the patch. Here, the robustness of ResNet50 has further deteriorated compared to a patch placed at the image corners. In contrast, transformers demonstrate similar vulnerability with the patch positioned at the center of the images and corner of the images under our Attention Fool loss variant $\mathcal{L}_{kq^\star}$. These results indicate that the transformers are less sensitive to location of the adversarial patch compared to CNNs.

	ResNet50	ViT-T	ViT-B	ViT-B-384	DeiT-T	DeiT-B	DeiT-B-384
$\mathcal{L}_{ce}$	41.50	0.10	14.50	28.90	23.50	44.20	66.50
$+\mathcal{L}_{kq}$	-	0.00 (–0.10)	5.80 (–8.70)	23.10 (–5.80)	22.20 (–1.30)	44.90 (+0.70)	69.10 (+2.60)
$+\mathcal{L}_{kq*}$	-	0.00 (–0.10)	4.10 (–10.40)	18.80 (–10.10)	12.60 (–10.90)	35.90 (–8.30)	67.20 (+0.70)
+ Momentum	31.10	0.00	2.40	11.10	1.60	20.90	42.30
$+\mathcal{L}_{kq}$	-	0.00 (-0.00)	0.30 (–2.10)	3.10 (–8.00)	0.10 (–1.50)	28.50 (+7.60)	48.90 (+6.60)
$+\mathcal{L}_{kq*}$	-	0.00 (-0.00)	0.20 (–2.20)	2.60 (–8.50)	0.10 (–1.50)	11.70 (–9.20)	45.90 (+3.60)

C.3. Normalization

In Section 5 we introduced an $\ell_{1,2}$ normalization in the computation of $\mathcal{L}_{kq}$, where projected queries and keys are normalized as $\bar{P}_Q^{hl} = P_Q^{hl}/\frac{1}{n}||P_Q^{hl}||_{1,2}$ and $\bar{P}_K^{hl} = P_K^{hl}/\frac{1}{n}||P_K^{hl}||_{1,2}$. To evaluate the effect of this normalization we compute $\mathcal{L}_{kq}$ without it, repeating the experiment reported in Table 3, and report the results in Table A5. We observe that this $\ell_{1,2}$ normalization is very crucial and without normalization, performance of $\mathcal{L}_{kq}$ and $\mathcal{L}_{kq^\star}$ deteriorates considerably, to levels clearly below $\mathcal{L}_{ce}$.

C.4. Patch Location

While Attention-Fool was designed to operate for arbitrary adversarial patch locations (as long as the adversarial patch aligns with ViT/DeiTs patch tiling), the location may have an effect on the resulting robust accuracies. In Section 6.2 we set the patch location to be the top leftmost corner of the image. Here, we repeat the experiment but we place the patch in the image center, and we target the corresponding key. Specifically, in 224×224-resolution models we place the patch top-left corner at 96,96, while in 384×384 models we place it at 176,176. We repeat the evaluation of Section 6.2 in this setting, and report the results in Table A6. Notably, the table shows that we obtain a significant drop in ResNet50 robust accuracy because of the different location (from 49.00% in Table 3 to 31.10% in Table A6), while the robust accuracies of vision transformers ViT and DeiT do not change significantly. Moreover, $\mathcal{L}_{kq^\star}$ improves upon $\mathcal{L}_{ce}$ for all but the DeiT-B-384 model.

C.5. Patch Sizes

The evaluation of Section 6.2 focuses on adversarial patches of size 16×16, which corresponds with the token dimension in the investigated ViT models. Here, we also investigate smaller adversarial patch sizes up to a dimension of 8×8, we always place the top-left corner of the patch at the (0, 0) coordinate (top-left) of the entire image. We report the robust accuracies for varying sizes in Table A4. As expected smaller patches lead to higher robust accuracies, but we find that even very small patches of 8×8 decrease the robust accuracy significantly compared to the clean accuracy for some models.

D. Ablation on Targeted Attacks on ViT

In Section 6.2, we reported the robust accuracies for vision transformers under an untargeted patch attacks. Here, we also evaluate a targeted attack on the same models by selecting a target class in the optimization: rather than maximizing $\mathcal{L}_{ce}$ with the true image class y as in Eq. 1, we replace y with a target class $y^\star$ and minimize the $\mathcal{L}_{ce}$ instead (note that the $\mathcal{L}_{kq}$ loss is still maximized in a targeted setting). The way we combine $\mathcal{L}_{ce}$ with the Attention-

Table A7. Adversarial patch attack success rate (%) for a targeted attack with target class “0”. The evaluation setting is similar to Table 3. Here, the attack success rate in vision transformers is larger than that of the ResNet50 model, but the improvements obtained with Attention Fool in comparison to the cross-entropy baseline are not as consistent as in the untargeted attack setting.

	ResNet50	ViT-T	ViT-B	ViT-B-384	DeiT-T	DeiT-B	DeiT-B-384
$\mathcal{L}_{ce}$	30.70	93.20	30.00	11.60	43.00	14.10	1.90
$+\mathcal{L}_{kq}$	-	94.60 (+1.40)	20.00 (-10.00)	12.20 (+0.60)	48.40 (+5.40)	14.80 (+0.70)	1.70 (-0.20)
$+\mathcal{L}_{kq*}$	-	94.30 (+1.10)	19.50 (-10.50)	13.00 (+1.40)	49.10 (+6.10)	14.60 (+0.50)	2.30 (+0.40)
+ Momentum	38.90	100.00	42.70	23.80	100.00	75.20	4.70
$+\mathcal{L}_{kq}$	-	100.00 (+0.00)	43.10 (+0.40)	21.10 (-2.70)	99.10 (-0.90)	74.60 (-0.60)	4.60 (-0.10)
$+\mathcal{L}_{kq*}$	-	100.00 (+0.00)	43.60 (+0.90)	22.40 (-1.40)	98.40 (-1.60)	77.10 (+1.90)	4.00 (-0.70)

Table A8. Mean Average Precisions (mAP) in presence of adversarial patches computed with $\mathcal{L}_{ce}$ and Attention-Fool’s $\mathcal{L}_{kq}^{(1)}$ patches on DETR models. Differently than in Table 4, here we replace the adversarial target key – one of the units in the backbone feature map – with its clean counterpart (the same unit computed in the non-patched image). Replacing the adversarial key removes the entirety of the adversarial effect of Attention-Fool, showing how the mAP degradation under attack can be attributed to the target key alone. Three different patch sizes and four different DETR models are considered, based either on R50 or R101 backbone and with and without dilation.

	R50	DC5-R50	R101	DC5-R101
clean mAP	53.00	54.25	54.41	56.74
64×64: $\mathcal{L}_{ce}$	38.96	17.51	33.35	35.67
$+\mathcal{L}_{kq}^{(1)}$	51.85 (+12.90)	52.92 (+35.41)	48.85 (+15.50)	56.61 (+20.94)
$\mathcal{L}_{kq}^{(1)}$ only	52.13 (+13.18)	53.48 (+35.97)	50.71 (+17.36)	57.11 (+21.44)
56×56: $\mathcal{L}_{ce}$	41.18	25.09	33.64	38.50
$+\mathcal{L}_{kq}^{(1)}$	53.34 (+12.17)	53.66 (+28.56)	53.51 (+19.87)	56.81 (+18.31)
$\mathcal{L}_{kq}^{(1)}$ only	52.73 (+11.55)	53.28 (+28.19)	52.37 (+18.74)	57.18 (+18.68)
48×48: $\mathcal{L}_{ce}$	43.61	27.84	40.45	42.16
$+\mathcal{L}_{kq}^{(1)}$	52.54 (+8.93)	52.97 (+25.13)	53.52 (+13.06)	56.90 (+14.73)
$\mathcal{L}_{kq}^{(1)}$ only	53.27 (+9.66)	53.99 (+26.16)	53.58 (+13.13)	56.90 (+14.74)

Fool variants is identical as in Section 6.2; here we choose $y^\star = 0$. In this case, rather than reporting robust accuracies, we report attack success rate, i.e., the percentage of times the addition of the adversarial patch changed the correct classification of an image into the target class $y^\star$. We report the results in Table A7, note that the colors and the signs of improvements/degradation are inverted in comparison to the other tables. Table A7 shows that the benefit of Attention-Fool in targeted attacks is less clear. We hypothesise that different weighting of $\mathcal{L}_{ce}$ and $\mathcal{L}_{kq}$ might be required in targeted settings, specifically because $\mathcal{L}_{ce}$ is bounded by zero when minimized (while it is unbounded when maximized in the untargeted setting). Additionally, specific properties of the target class “0” on certain models might bias the evaluation; a larger, more in-depth evaluation of targeted attacks is left for future work.

E. Adversarial Token Replacement

In Section 6.3 we showed how targeting a single key with Attention-Fool in DETR can lead to large degradation in mAP. Differently than in ViT models, because of DETR’s hybrid architecture (CNN plus Transformer) an adversarial patch placed on the DETR image input affects a number of tokens in the encoder’s input: all those backbone outputs (tokens) whose receptive field via the CNN overlaps with the input image patch. Here, to show that the attack performance can be attributed to the individual key token that is the Attention-Fool target, we replace this key with its clean counterpart. To do so, we compute all keys on the clean image and all keys in the patched image (by forwarding the image through the CNN backbone), and we replace the target adversarial key with its clean counterpart. This is either the key indexed by 2,2 or by 4,4 in non-dilated and dilated models, respectively. We report the resulting mAPs in Table A8. The table shows how replacing this single token’s key removes a large part of the adversarial effect in all rows using $\mathcal{L}_{kq}^{(1)}$ loss. In comparison, in rows using $\mathcal{L}_{ce}$, replacing this token’s key still results in low mAPs, showing that in this case the adversarial effect is not attributable to the same adversarial token alone.

F. Additional Visualizations

We report additional visualizations akin to those in Fig. 1 and Fig. 4 in Figure A3 and Figure A2, respectively.

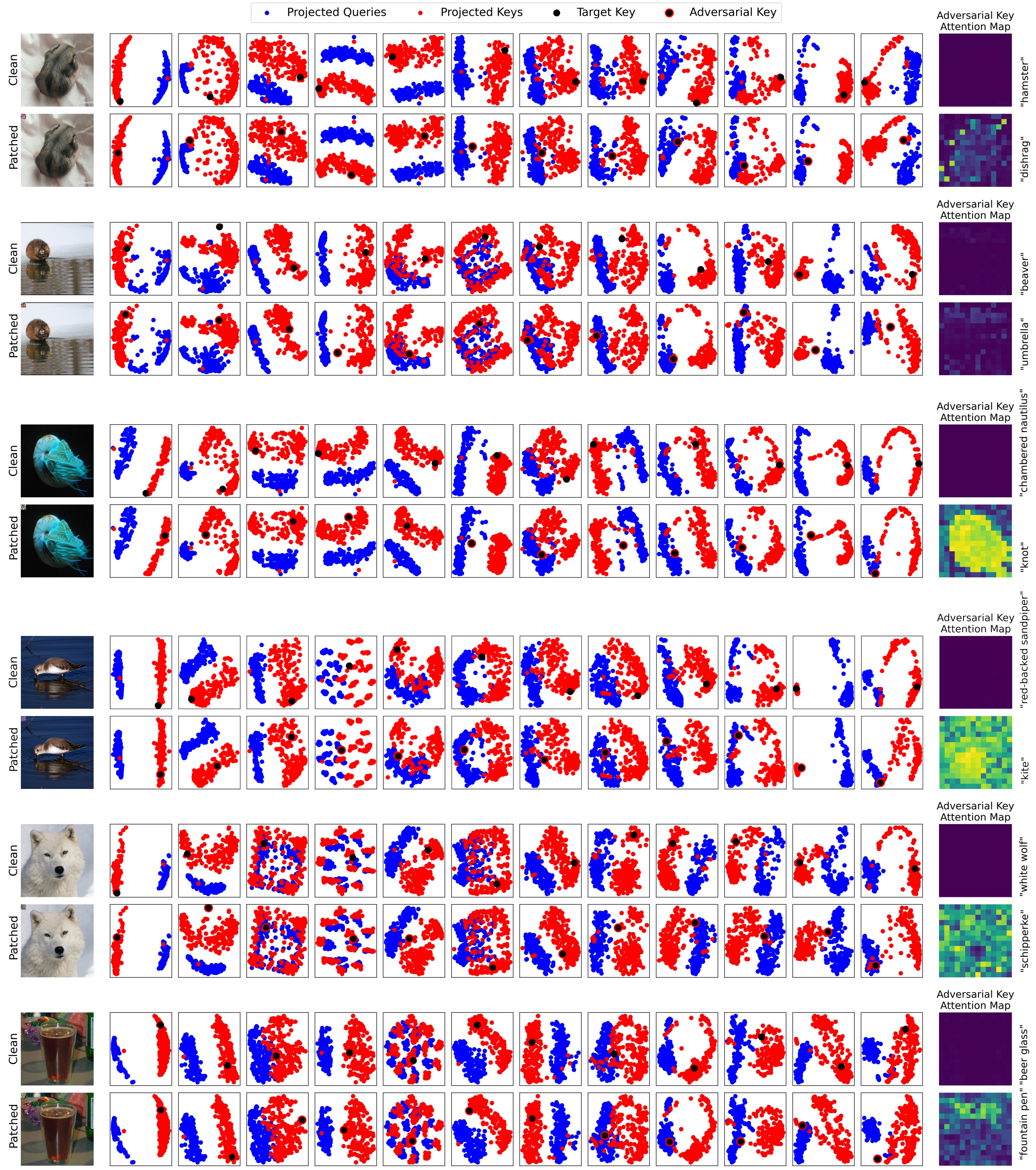

Figure A2. Embedded projected key and query tokens for clean and patched input images on each of the 12 DeiT-B layers, for a single attention head. For each image we chose an attention head which showed large amount of changes. The last column reports the attention map weights of the adversarial key on the last layer – showing that generally the key draws a large amount of attention from queries. Note that while the adversarial patch tends to focus on drawing the attention on the last layer (which is visualized in the right-most column), it can target arbitrary layers – in fact, we obtain successful mis-classifications even if the last layer attention map only has minor changes.

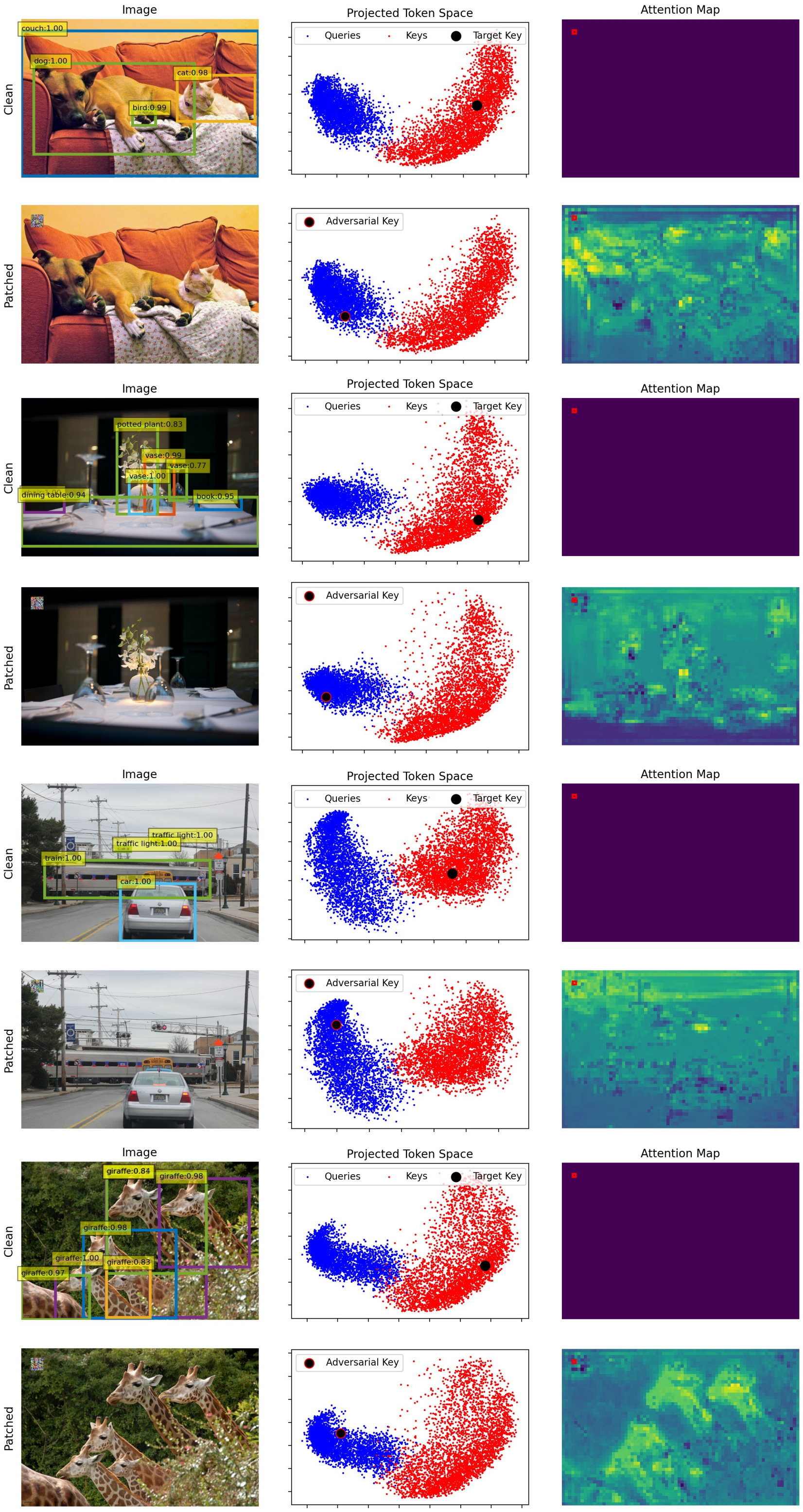

Figure A3. More comparisons of clean and adversarially patched input for DETR [8]. The patch shifts a targeted key token towards the cluster of query tokens. In dot-product attention, this directs queries attention to the malicious token and prevents the model from detecting the remaining objects. The right-most column compares queries' attention weights to the adversarial key, whose location is marked by a red box, between clean and patched inputs. These images use DETR DC5-R50 and patches are optimized with $\mathcal{L}_{ce} + \mathcal{L}_{kq}^{(1)}$.