Supplementary Material: Improving the Transferability of Targeted Adversarial Examples through Object-Based Diverse Input

A. References to used 3D models

In our experiments, we used six 3D models as source objects. All of them are free downloadable and have royalty-free licenses.

- *T-shirt*: https://www.turbosquid.com/3d-models/3ds-max-tshirt-soccer/570329
- *Book*: https://www.turbosquid.com/3d-models/free-3ds-mode-book-modelled/656471
- *Ball*: https://www.turbosquid.com/3d-models/pool-balls-max-free/877865
- *Pillow*: https://www.turbosquid.com/3d-models/3d-decorative-kawaii-pillows-1543064
- *Cup*: https://www.turbosquid.com/3d-models/free-cups-spongebob-3d-model/935414
- *Package*: https://www.cgtrader.com/free-3d-models/food/miscellaneous/lays-packaging-chips

Algorithm 1 Object-based Diverse Input (ODI) method.

Input: An image $\mathbf{x}$.

Input: A source object pool $\mathcal{O}$; the range of texture color $(l_{texture}, u_{texture})$; the range of camera angles (l_{angle}, u_{angle}); the range of camera distance (l_{dist}, u_{dist}); the range of ambient light brightness $(l_{ambient}, u_{ambient})$; the range of diffuse light brightness $(l_{diffuse}, u_{diffuse})$; the range of translation of the light (l_{light}, u_{light}), and the default position of the light $\mathbf{l}_d$.

Output: The transformed image $\mathbf{x}''$.

$\mathbf{o} \sim \mathcal{O}$ // Randomly choose an object in the object pool

$(\mathbf{t}, \mathbf{b}, \mathbf{m}) = \mathbf{o}$ // Get the texture map $\mathbf{t}$, bounding box of adversarial texture $\mathbf{b}$, and the mesh $\mathbf{m}$ from the object

$\mathbf{t}' = Fill(\mathbf{t}, \mathbf{c})$, where $\mathbf{c} \sim \mathcal{U}(l_{texture}, u_{texture})$ // Paint the texture $\mathbf{t}$ with the random solid color

$\mathbf{t}'[\mathbf{b}] = Resize(\mathbf{x}, \mathbf{b})$ // Insert the resized input image $\mathbf{x}$ into the area of $\mathbf{b}$ of the painted texture $\mathbf{t}'$

$\mathbf{c}_a \sim \mathcal{U}(l_{angle}, u_{angle})$ // Randomly choose the camera angles.

$c_d \sim \mathcal{U}(l_{dist}, u_{dist})$ // Randomly choose the camera distance.

$\mathbf{l}_a = l_a\mathbf{1}$, where $l_a \sim \mathcal{U}(l_{ambient}, u_{ambient})$ // Randomly choose the ambient light color.

$\mathbf{l}_d = l_d\mathbf{1}$, where $l_d \sim \mathcal{U}(l_{diffuse}, u_{diffuse})$ // Randomly choose the diffuse light color.

$\mathbf{l}_p = \mathbf{l}_d + \mathbf{l}_r$ where $\mathbf{l}_r \sim \mathcal{U}(l_{light}, u_{light})$ // Randomly choose the translation amount of the light.

$\mathbf{x}' = Render(\mathbf{t}', \mathbf{m}, \mathbf{c}_a, c_d, \mathbf{l}_a, \mathbf{l}_d, \mathbf{l}_p)$ // Render the modified textured 3D mesh with the randomly chosen parameters.

$\mathbf{b} \sim \mathcal{U}(0, 1)$ // Randomly generate the background image

$\mathbf{x}'' = Blend(\mathbf{x}', \mathbf{b})$

return $\mathbf{x}''$

Source : RN-50	Target model									
Attack	VGG-16	RN-18	DN-121	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
MI-TI	3.2	6.6	8.4	0.3	0.4	2.4	0.1	0.0	0.0	2.3
ATTA-MI-TI	6.6	10.8	14.7	0.9	0.8	3.8	0.2	0.0	0.0	14.1
DI-MI-TI	62.2	54.9	71.4	10.5	9.0	28.5	4.5	0.0	0.0	2.7
RDI-MI-TI	67.8	73.4	82.9	32.4	24.6	44.5	17.4	0.0	0.0	2.3
ODI-MI-TI	**76.8**	**77.0**	**86.8**	**67.4**	**55.4**	**66.8**	**48.0**	**0.7**	**1.7**	6.0
Source : VGG-16	Target model									
Attack	RN-18	RN-50	DN-121	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
MI-TI	0.8	1.0	1.3	0.0	0.2	0.9	0.0	0.0	0.0	5.3
ATTA-MI-TI	1.6	2.6	3.5	0.2	0.2	1.7	0.0	0.0	0.0	21.0
DI-MI-TI	7.6	11.2	12.7	0.6	2.3	6.3	0.2	0.0	0.0	6.1
RDI-MI-TI	28.7	31.5	35.9	6.6	9.5	18.0	3.7	0.0	0.0	5.4
ODI-MI-TI	**60.8**	**64.3**	**71.1**	**37.0**	**38.0**	**47.0**	**21.1**	0.0	0.0	9.0
Source : DN-121	Target model									
Attack	VGG-16	RN-18	RN-50	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
MI-TI	4.1	5.5	6.5	0.3	0.6	2.2	0.5	0.0	0.0	2.5
ATTA-MI-TI	6.9	8.9	10.6	1.4	1.6	3.6	0.8	0.0	0.0	18.7
DI-MI-TI	38.3	30.1	43.6	7.1	7.7	13.7	4.5	0.0	0.0	2.8
RDI-MI-TI	41.9	45.3	55.9	21.0	19.1	21.8	12.9	0.0	0.0	2.5
ODI-MI-TI	**64.5**	**63.4**	**71.6**	**53.5**	**46.4**	**44.2**	**38.3**	**0.4**	**0.7**	6.2
Source : Inc-v3	Target model									
Attack	VGG-16	RN-18	RN-50	DN-121	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2	Computation time per image (sec)
MI-TI	0.3	0.0	0.1	0.1	0.1	0.1	0.1	0.1	0.0	1.9
ATTA-MI-TI	0.3	0.3	0.2	0.4	0.5	0.4	0.1	0.0	0.0	15.2
DI-MI-TI	4.2	2.2	3.6	5.4	4.3	2.4	3.6	0.0	0.0	2.2
RDI-MI-TI	3.2	4.4	4.4	6.9	8.3	3.0	5.5	0.0	0.0	1.9
ODI-MI-TI	**15.7**	**14.7**	**17.4**	**30.4**	**32.1**	**14.1**	**26.9**	**0.3**	**0.6**	5.5

Table 1. Targeted attack success rates (%) against nine black-box target models with the four source models. For each attack, we also reported the average computation time to generate an adversarial example. We added experimental results of MI-TI and ATTA-MI-TI.

Source : VGG-16	Target model								
Object	RN-18	RN-50	DN-121	Inc-v3	Inc-v4	Mob-v2	IR-v2	Adv-Inc-v3	Ens-adv-IR-v2
One ball	21.0	25.2	37.0	12.4	11.8	16.1	5.1	0.0	0.0
Two balls	40.1	33.9	47.1	36.6	25.0	25.0	19.4	0.8	1.0
Three balls	**45.5**	**37.7**	**50.6**	**36.8**	**26.9**	**29.0**	**21.6**	0.5	0.5
Four balls	32.4	25.3	35.5	25.5	17.6	16.7	14.7	**1.4**	**2.5**

Table 2. Targeted attack success rates (%) against nine black-box target models according to the source objects. We consider the different numbers of ball objects.

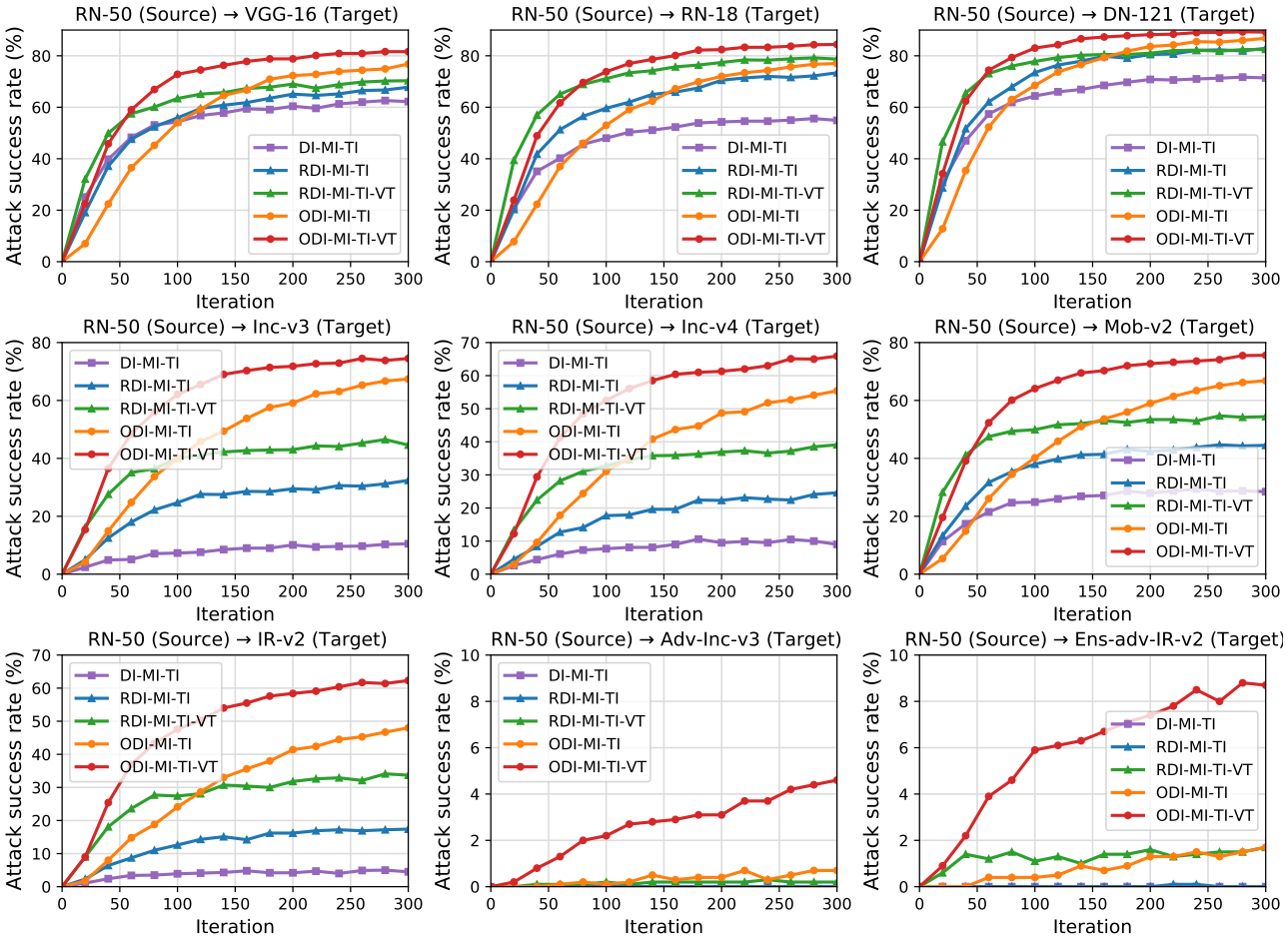

Figure 1. Targeted attack success rates (%) according to the number of iterations. The source model is RN-50.

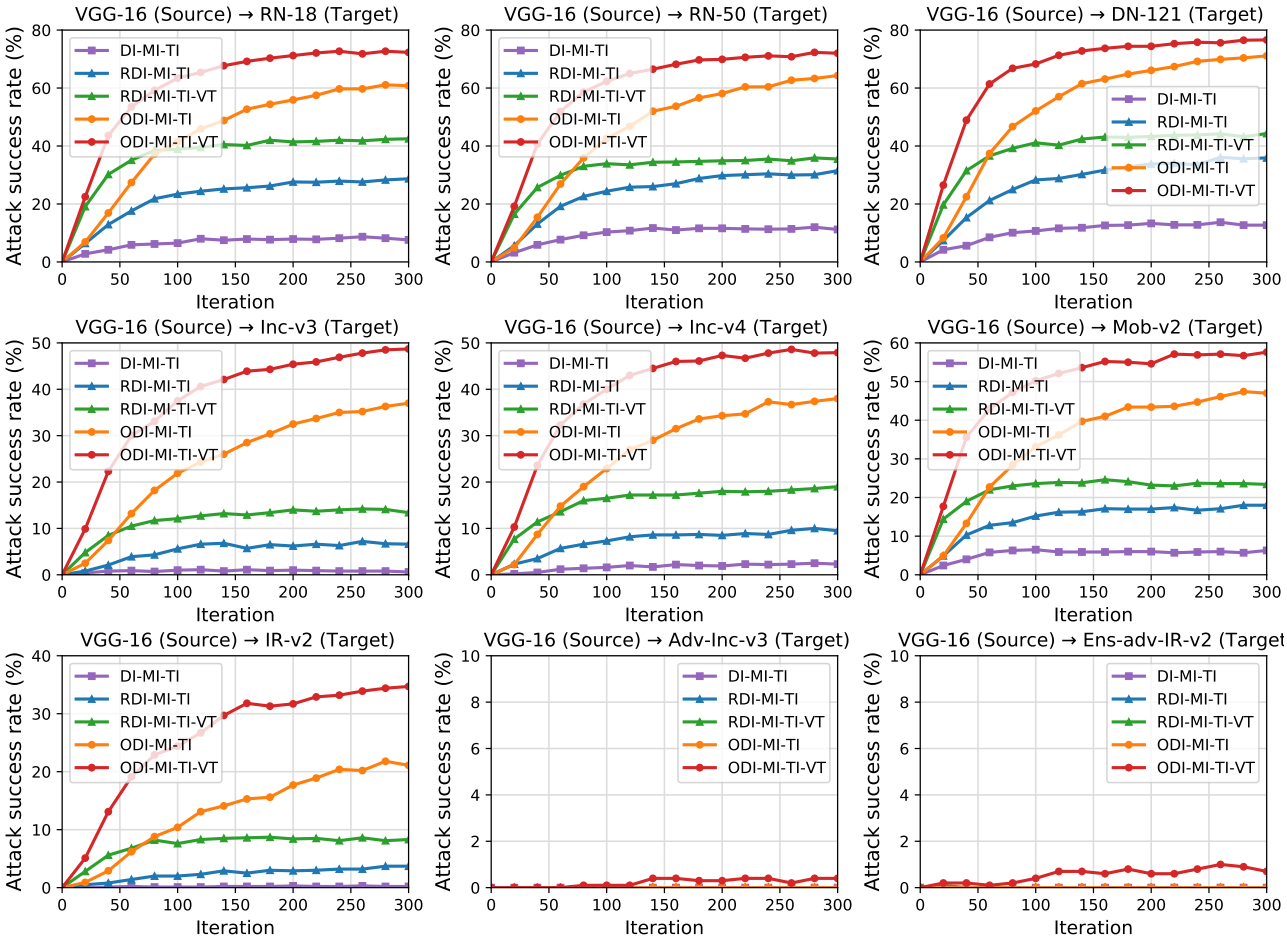

Figure 2. Targeted attack success rates (%) according to the number of iterations. The source model is VGG-16.

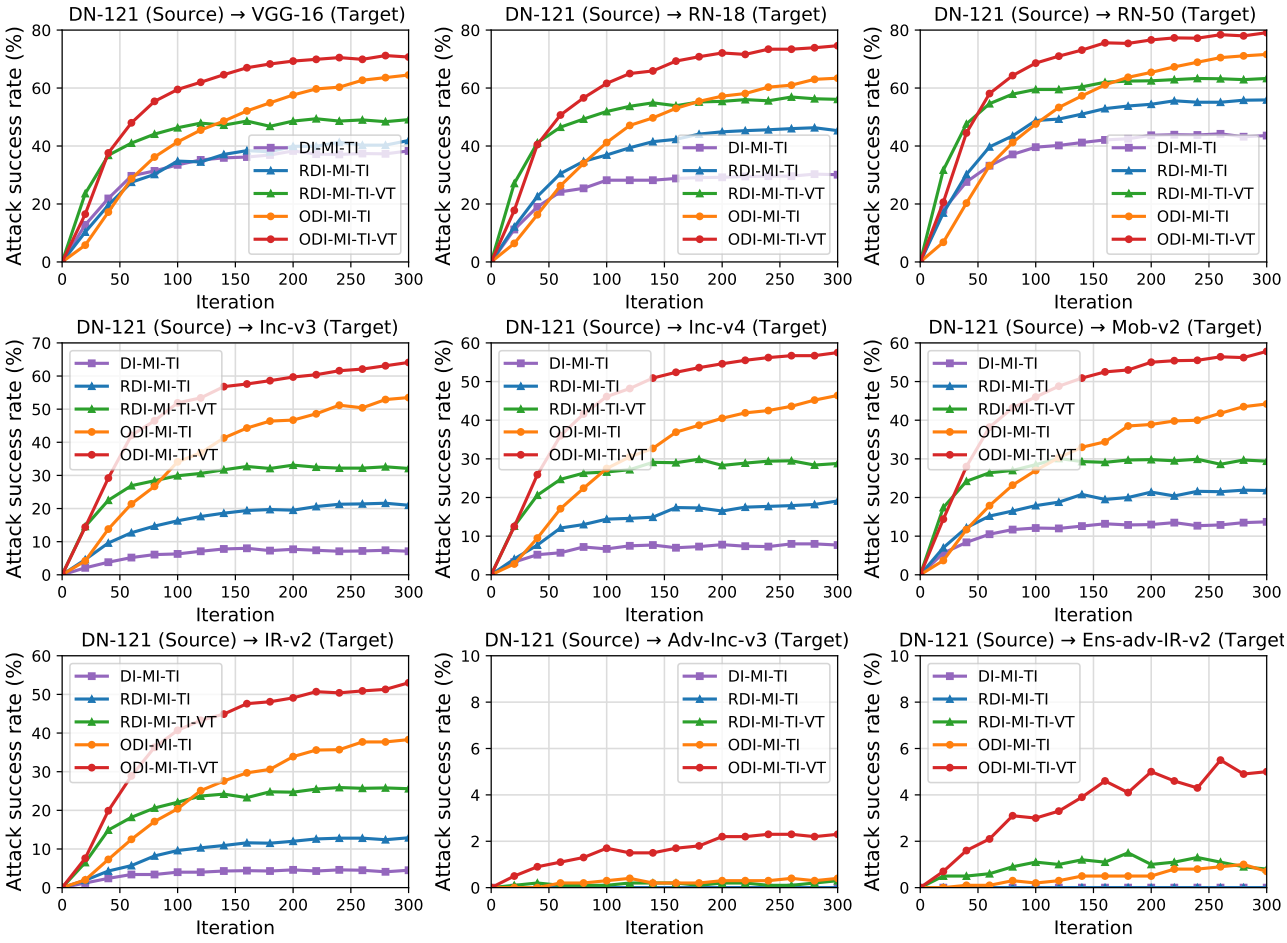

Figure 3. Targeted attack success rates (%) according to the number of iterations. The source model is DN-121.

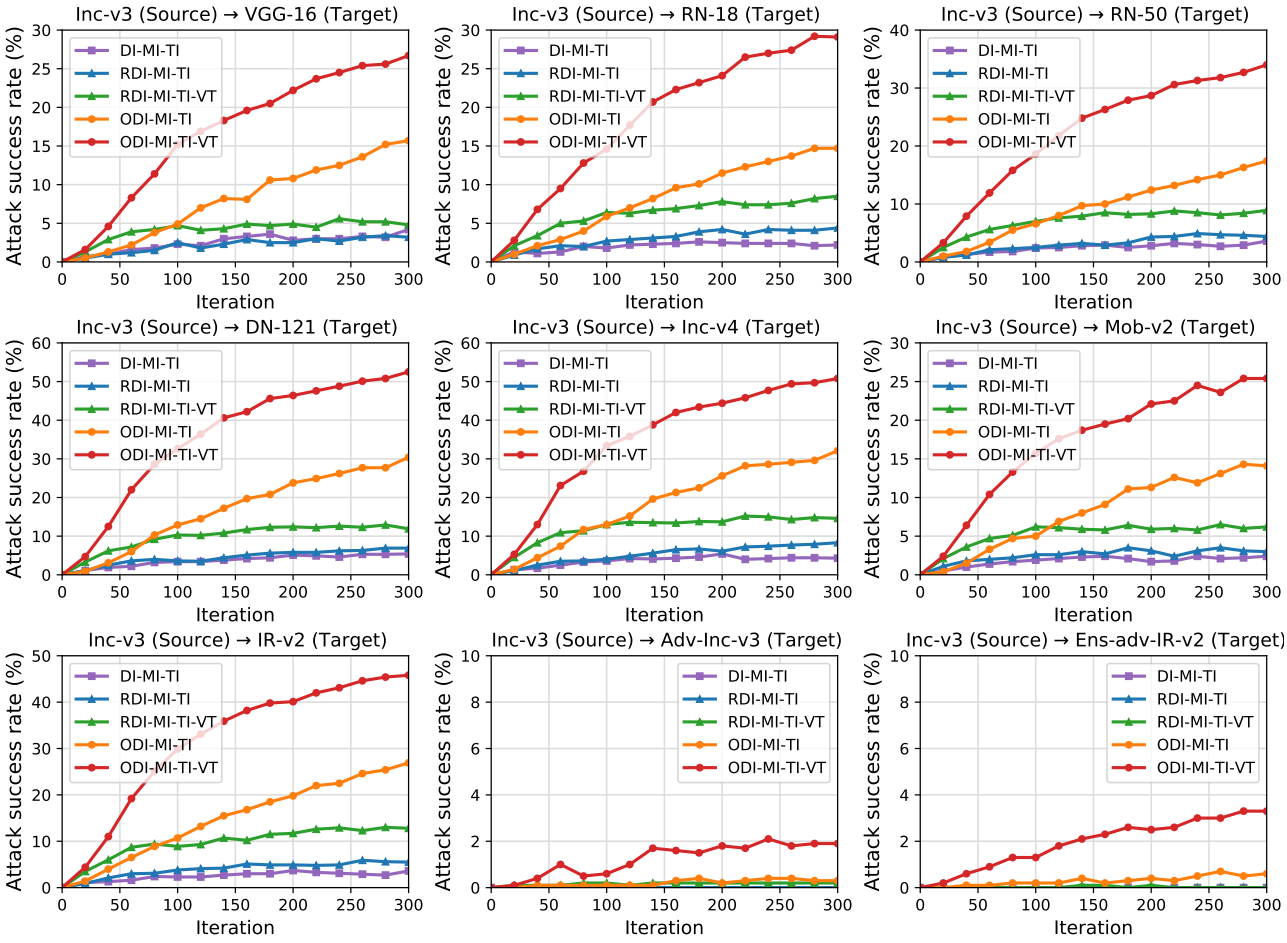

Figure 4. Targeted attack success rates (%) according to the number of iterations. The source model is Inc-v3.