

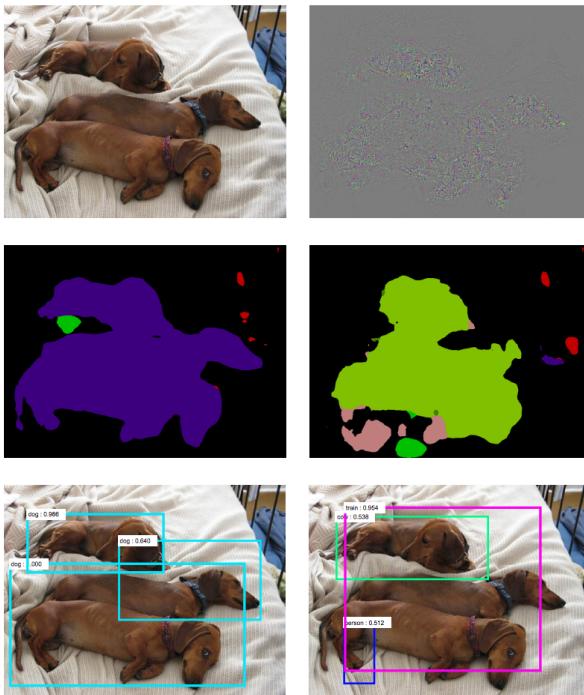
Evasion Attacks Against Various Machine Learning Models

Adversarial examples for semantic segmentation and object detection

- Generating adv is a critical step for evaluating and improving robustness of learning models.
- So far we introduced adv against classifiers
- What about other learning tasks?

Adversarial examples for semantic segmentation and object detection

- Both segmentation and detection are based on classifying multiple targets on an image
- Dense adversary generation (DAG)



Adversarial examples for semantic segmentation and object detection

Problem statement

Untargeted attack

$$\forall n, \arg \max_c f_c(X + r, t_n) \neq l_n$$

Perturbation targets Ground truth

Targeted attack

$$L(\mathbf{X}, \mathcal{T}, \mathcal{L}, \mathcal{L}') = \sum_{n=1}^N [f_{l_n}(\mathbf{X}, t_n) - f_{l'_n}(\mathbf{X}, t_n)]$$

Algorithm 1: Dense Adversary Generation (DAG)

Input : input image \mathbf{X} ;
the classifier $f(\cdot, \cdot) \in \mathbb{R}^C$;
the target set $\mathcal{T} = \{t_1, t_2, \dots, t_N\}$;
the original label set $\mathcal{L} = \{l_1, l_2, \dots, l_N\}$;
the adversarial label set $\mathcal{L}' = \{l'_1, l'_2, \dots, l'_N\}$;
the maximal iterations M_0 ;

Output: the adversarial perturbation \mathbf{r} ;

1 $\mathbf{X}_0 \leftarrow \mathbf{X}$, $\mathbf{r} \leftarrow 0$, $m \leftarrow 0$, $\mathcal{T}_0 \leftarrow \mathcal{T}$;

2 **while** $m < M_0$ **and** $\mathcal{T}_m \neq \emptyset$ **do**

3 $\mathcal{T}_m = \{t_n \mid \arg \max_c \{f_c(\mathbf{X}_m, t_n)\} = l_n\}$;

4 $\mathbf{r}_m \leftarrow$
 $\sum_{t_n \in \mathcal{T}_m} [\nabla_{\mathbf{X}_m} f_{l'_n}(\mathbf{X}_m, t_n) - \nabla_{\mathbf{X}_m} f_{l_n}(\mathbf{X}_m, t_n)]$;

5 $\mathbf{r}'_m \leftarrow \frac{\gamma}{\|\mathbf{r}_m\|_\infty} \mathbf{r}_m$;

6 $\mathbf{r} \leftarrow \mathbf{r} + \mathbf{r}'_m$;

7 $\mathbf{X}_{m+1} \leftarrow \mathbf{X}_m + \mathbf{r}'_m$;

8 $m \leftarrow m + 1$;

9 **end**

Return: \mathbf{r}

Transferability analysis

- Cross training transfer
 - Models are trained with different subset of data
- Cross network transfer
 - Models are of different architecture
- Cross task transfer
 - Use the perturbation generated against detection to attack a segmentation network

Adversarial Perturbations from	FR-ZF-07	FR-ZF-0712	FR-VGG-07	FR-VGG-0712	R-FCN-RN50	R-FCN-RN101
None	58.70	61.07	69.14	72.07	76.40	78.06
FR-ZF-07 (r_1)	3.61	22.15	66.01	69.47	74.01	75.87
FR-ZF-0712 (r_2)	13.14	1.95	64.61	68.17	72.29	74.68
FR-VGG-07 (r_3)	56.41	59.31	5.92	48.05	72.84	74.79
FR-VGG-0712 (r_4)	56.09	58.58	31.84	3.36	70.55	72.78
$r_1 + r_3$	3.98	21.63	7.00	44.14	68.89	71.56
$r_1 + r_3$ (permute)	58.30	61.08	68.63	71.82	76.34	77.71
$r_2 + r_4$	13.15	2.13	28.92	4.28	63.93	67.25
$r_2 + r_4$ (permute)	58.51	61.09	68.68	71.78	76.23	77.71

Cross training

Adversarial Perturbations from	FCN-Alex	FCN-Alex*	FCN-VGG	FCN-VGG*	DL-VGG	DL-RN101
None	48.04	48.92	65.49	67.09	70.72	76.11
FCN-Alex (r_5)	3.98	7.94	64.82	66.54	70.18	75.45
FCN-Alex* (r_6)	5.10	3.98	64.60	66.36	69.98	75.52
FCN-VGG (r_7)	46.21	47.38	4.09	16.36	45.16	73.98
FCN-VGG* (r_8)	46.10	47.21	12.72	4.18	46.33	73.76
$r_5 + r_7$	4.83	8.55	4.23	17.59	43.95	73.26
$r_5 + r_7$ (permute)	48.03	48.90	65.47	67.09	70.69	76.04
$r_6 + r_8$	5.52	4.23	13.89	4.98	44.18	73.01
$r_6 + r_8$ (permute)	48.03	48.90	65.47	67.05	70.69	76.05

Cross Network

Takeaways

- Heuristically generate perturbation to move each target towards the adversarial goal
- Transferability exists for adversarial examples for segmentation/detection
- Adding multiple adversarial perturbations often works better than adding a single source of perturbation in terms of transferability

Similar work

- Delving into transferable adversarial examples and black-box attacks
 - Apply ensemble attack to attack multiple models to increase targeted transferability
 - Multi-source perturbation helps?



Ground truth: running shoe

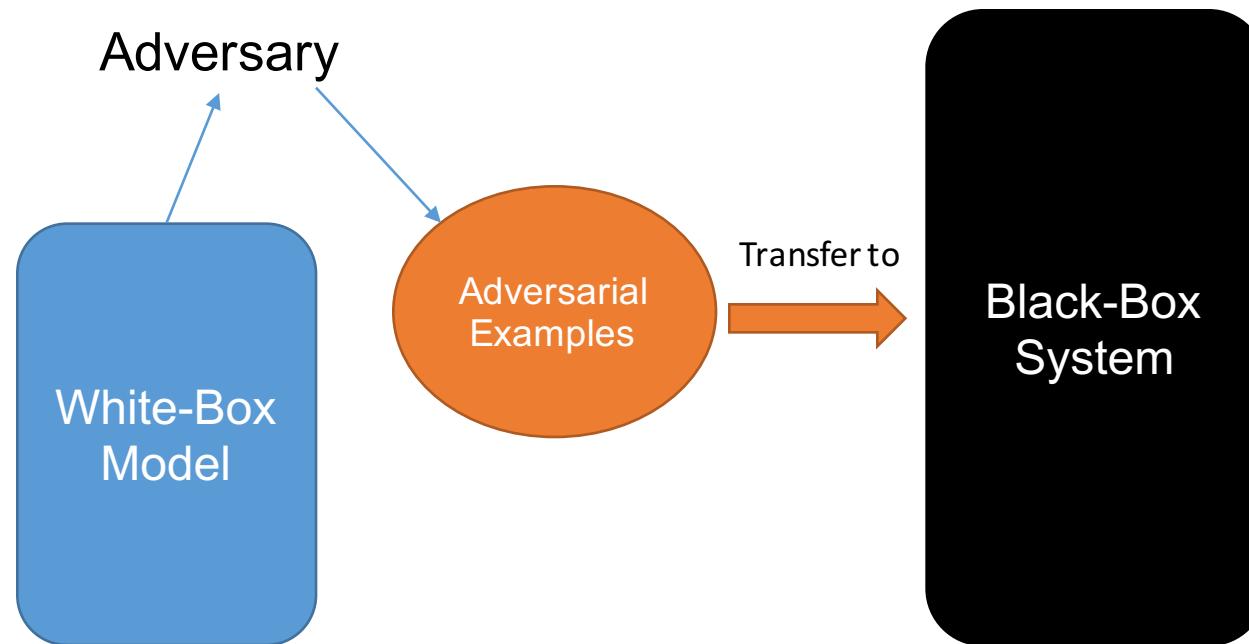
VGG16	Military uniform
ResNet50	Jigsaw puzzle
ResNet101	Motor scooter
ResNet152	Mask
GoogLeNet	Chainsaw

Targeted Adversarial Example's Transferability Among Two Models is Poor!

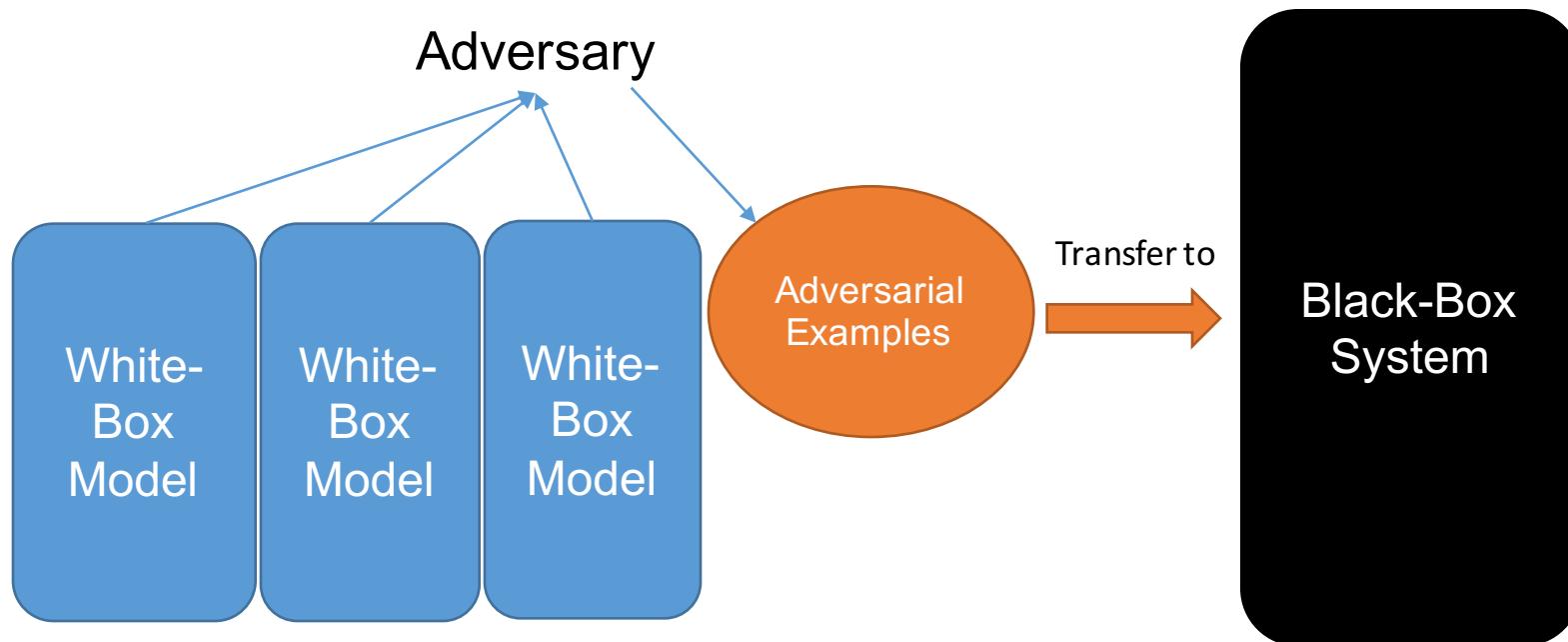
	ResNet152	ResNet101	ResNet50	VGG16	GoogLeNet	Incept-v3
ResNet152	100%	2%	1%	1%	1%	0%
ResNet101	3%	100%	3%	2%	1%	1%
ResNet50	4%	2%	100%	1%	1%	0%
VGG16	2%	1%	2%	100%	1%	0%
GoogLeNet	1%	1%	0%	1%	100%	0%
Incept-v3	0%	0%	0%	0%	0%	100%

Only 2% of the adversarial images generated for VGG16 (row) can be predicted as the targeted label by ResNet50 (column)

Black-box Attacks Based On Transferability

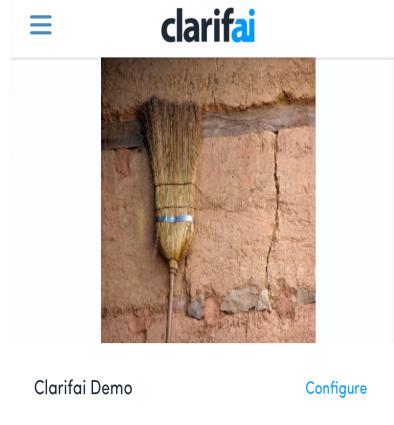


Ensemble Targeted Black-box Attacks Based On Transferability



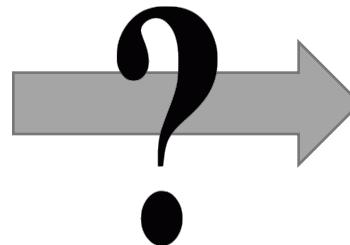
Clarifai.com

Ground truth from ImageNet: broom



The image shows a screenshot of the Clarifai web interface. At the top, there is a navigation bar with three horizontal lines and the word "clarifai". Below the navigation bar, there is a large image of a broom leaning against a textured wall. Underneath the image, the text "Clarifai Demo" and "Configure" are visible. At the bottom of the interface, the text "GENERAL-V1.3" is followed by three rows of circular buttons. The first row contains "wall", "dirty", "old", and "no person". The second row contains "architecture", "stone", "building", and "dust". The third row contains "rope", "rustic", "brick", "ancient", and "soil".

jacamar



Adversarial Example on Clarifai.com

- Ground truth: **broom**
- Target label: **jacamar**

Clarifai Demo [Configure](#)

GENERAL-V1.3



bird nature desktop color art tree
pattern bright feather painting texture
design decoration flora no person
beautiful leaf garden old illustration

Similar work

- Physical Adversarial Examples for Object Detectors

$$J_d(x, y) = \max_{s \in S^2, b \in B} P(s, b, y, f_\theta(x))$$

Cell in YOLO Bounding box

Difference: instead of ensemble over models, here it ensembles over object regions

Houdini: Fooling Deep Structured Prediction Models

- Other deterministic objective function for attacking different learning models?
- Houdini: tailored for the final performance measure
 - Speech recognition
 - Pose estimation
 - Semantic segmentation

Houdini: Fooling Deep Structured Prediction Models

- Optimization based method

$$\tilde{x} = \underset{\tilde{x}: \|\tilde{x}-x\|_p \leq \epsilon}{\operatorname{argmax}} \ell(y_\theta(\tilde{x}), y) \quad f_2(x') = (\max_{i \neq t}(F(x')_i) - F(x')_t)^+$$

- Houdini

$$\bar{\ell}_H(\theta, x, y) = \mathbb{P}_{\gamma \sim \mathcal{N}(0, 1)} \left[g_\theta(x, y) - g_\theta(x, \hat{y}) < \gamma \right] \cdot \ell(\hat{y}, y)$$



Stochastic margin
Confidence of the model Task loss



original semantic segmentation framework



adversarial attack



compromised semantic segmentation framework



(a) initial prediction



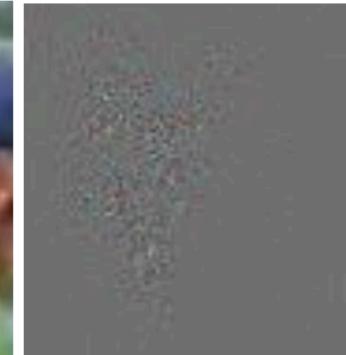
(b) adversarial prediction



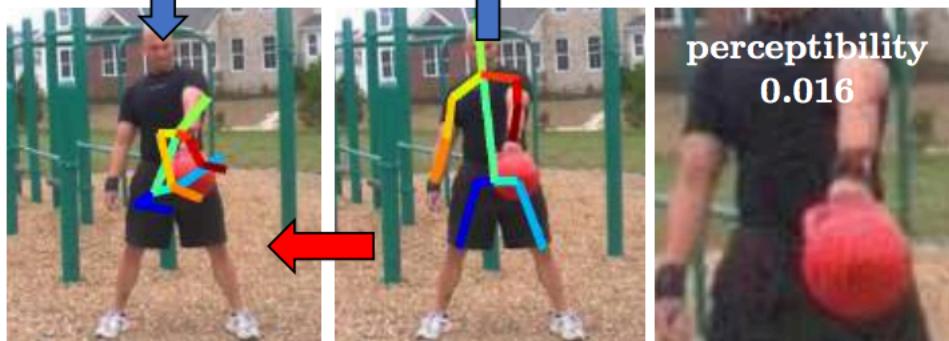
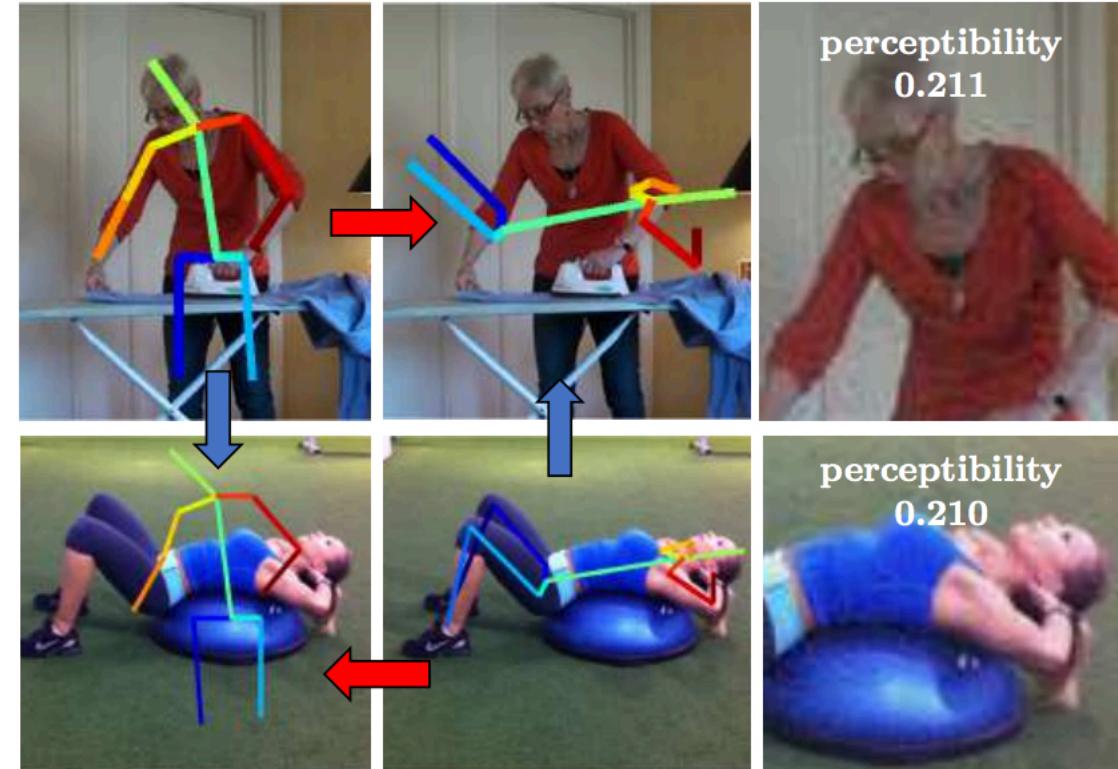
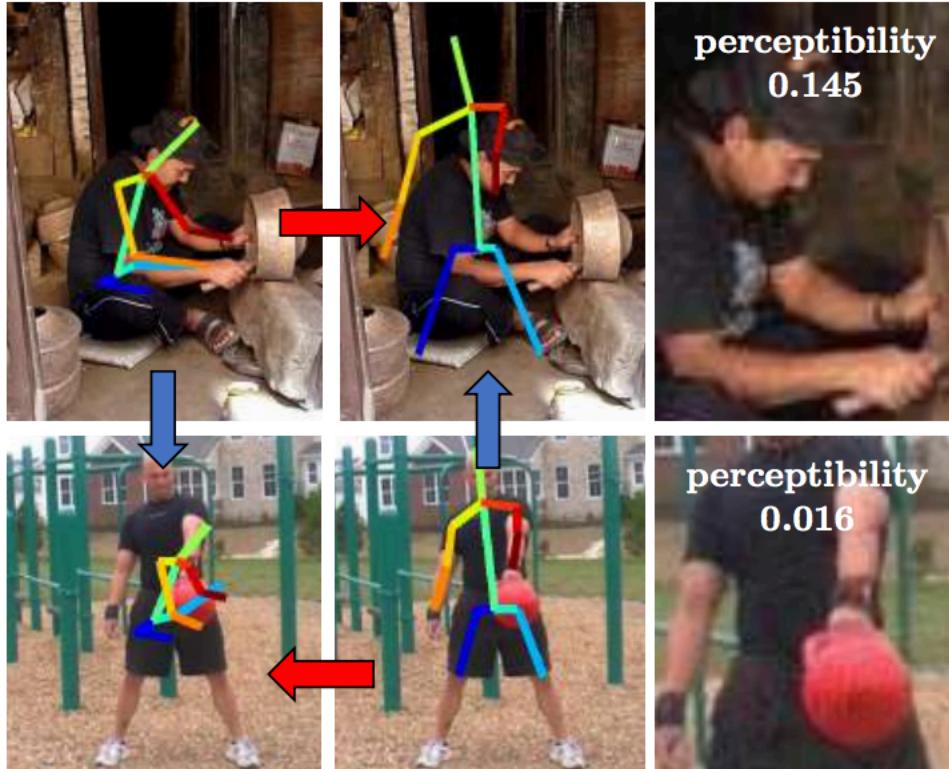
(c) source image



(d) perturbed image



(e) noise



	$\epsilon = 0.3$		$\epsilon = 0.2$		$\epsilon = 0.1$		$\epsilon = 0.05$	
	WER	CER	WER	CER	WER	CER	WER	CER
CTC	68	9.3	51	6.9	29.8	4	20	2.5
Houdini	96.1	12	85.4	9.2	66.5	6.5	46.5	4.5

Groundtruth Transcription:

The fact that a man can recite a poem does not show he remembers any previous occasion on which he has recited it or read it.

G-Voice transcription of the original example:

The fact that a man can **decide** a poem does not show he remembers any previous occasion on which he has **work cited** or read it.

G-Voice transcription of the adversarial example:

The fact that **I can rest I'm just not sure that you heard there is** any previous occasion **I am at he has your side** it or read it.

Groundtruth Transcription:

Her bearing was graceful and animated she led her son by the hand and before her walked two maids with wax lights and silver candlesticks.

G-Voice transcription of the original example:

The bearing was graceful **an** animated she **let** her son by the hand and before he walks two maids with wax lights and silver candlesticks.

G-Voice transcription of the adversarial example:

Mary was grateful then admitted she let her son before the walks to Mays would like slice furnace filter count six.

Takeaways

- By adding margin based constraint together with the task loss, the attack can be generated against a range of tasks with high confidence
- Targeted attacks seem to be more challenging when dealing with speech recognition systems than when we consider artificial visual systems such as pose estimators or semantic segmentation systems
- Adversarial audios also transfer among models

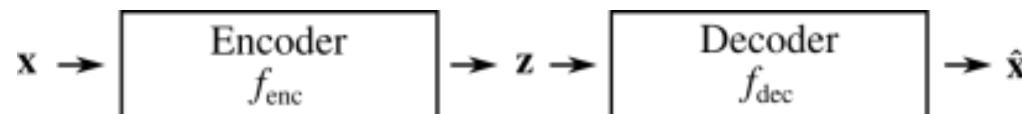
Adversarial Examples for Generative Models

- Idea: Create adversarial inputs that can control the latent space of a generative model.
- Generate based on adversarial target

Adversarial Examples for Generative Models

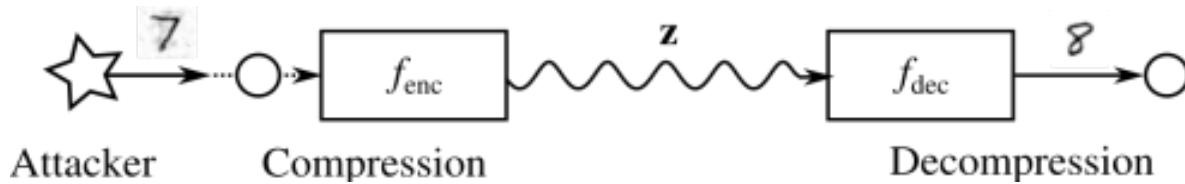
- **Generative Models.**

- An **encoder** maps a high-dimensional input into lower-dimensional latent representation.
- A **decoder** maps the latent representation back to a high-dimensional reconstruction.
- A **latent space** is an internal representation of the data.



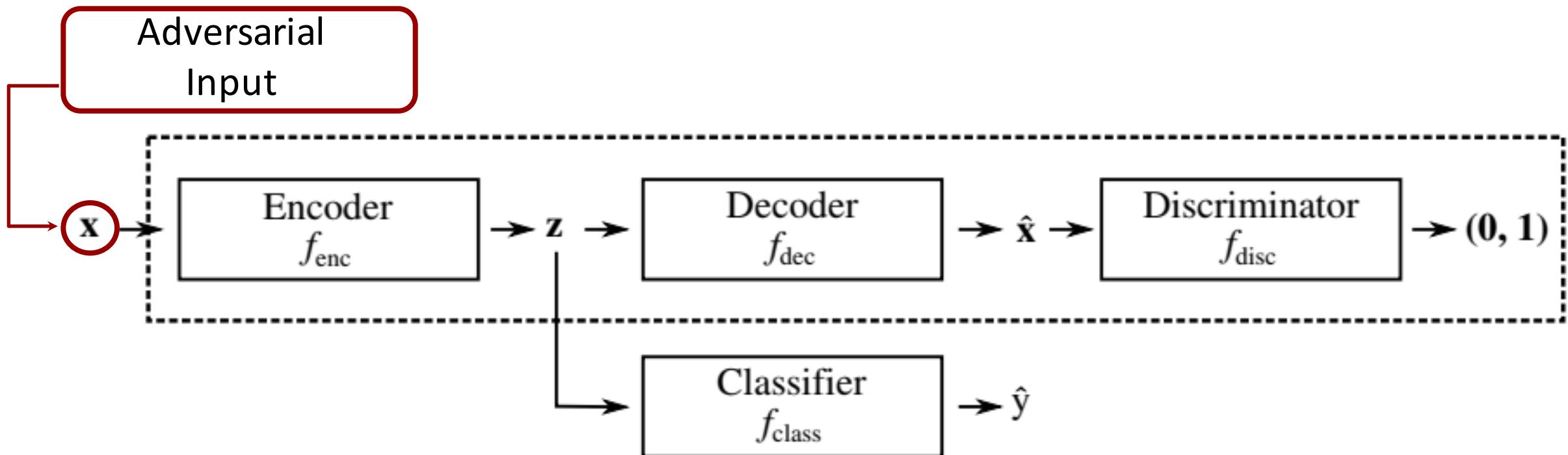
Adversarial Examples for Generative Models

- An example attack scenario:
 - Generative model used as a compression scheme

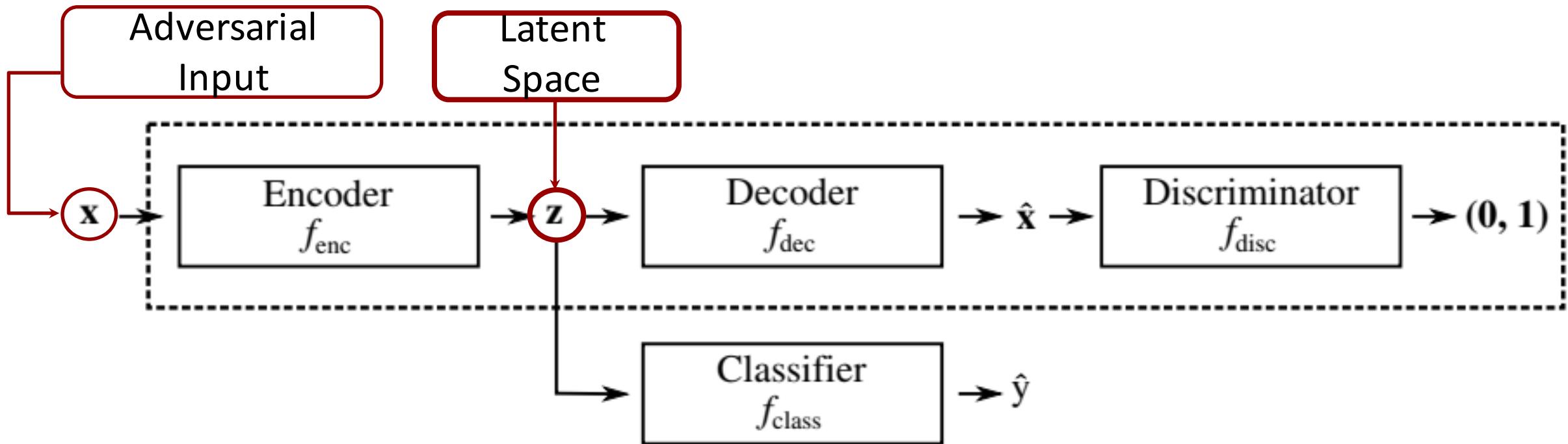


- Attacker's goal: for the decompressor to reconstruct a different image from the one that the compressor sees.

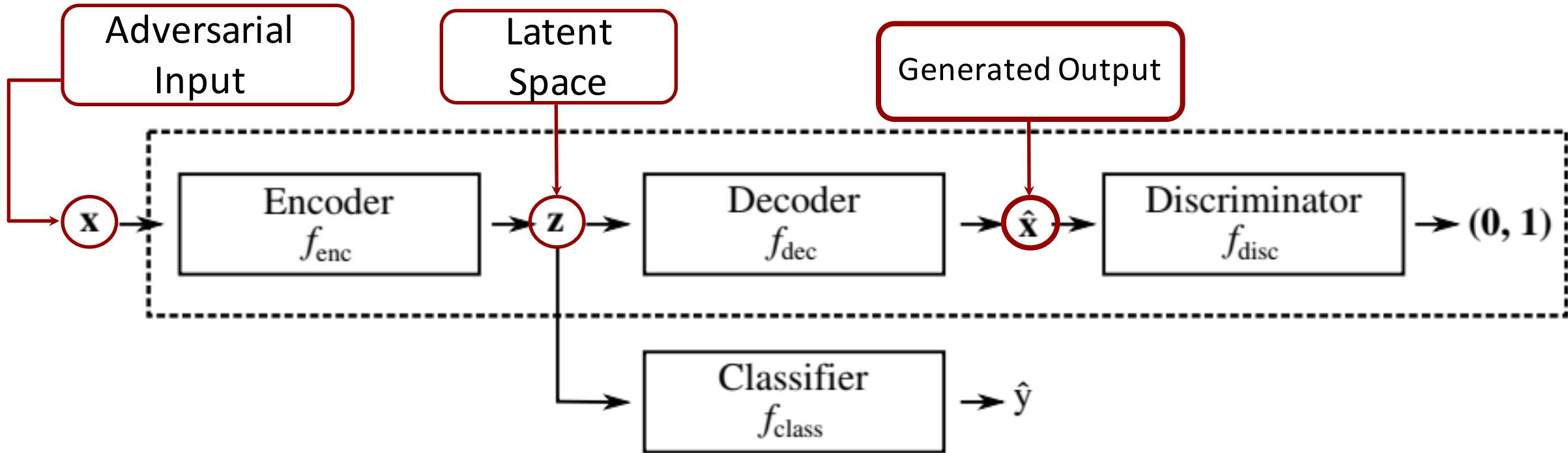
Adversarial Examples for Generative Models



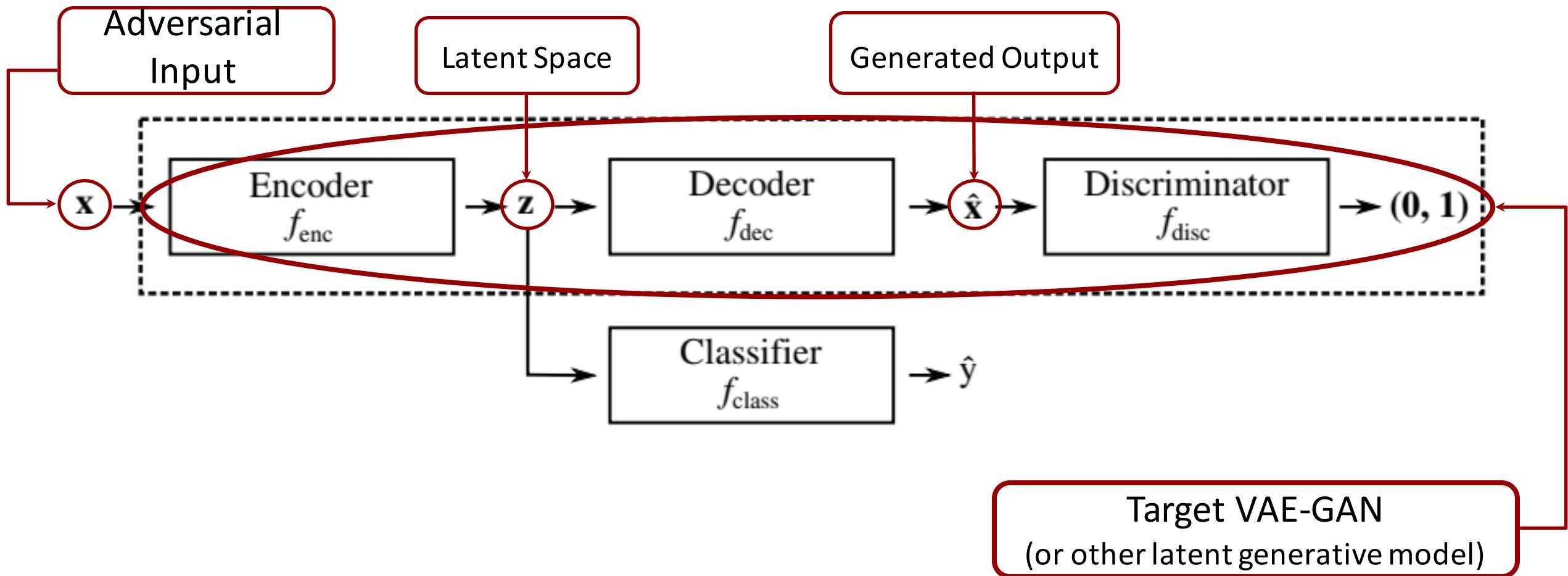
Adversarial Examples for Generative Models



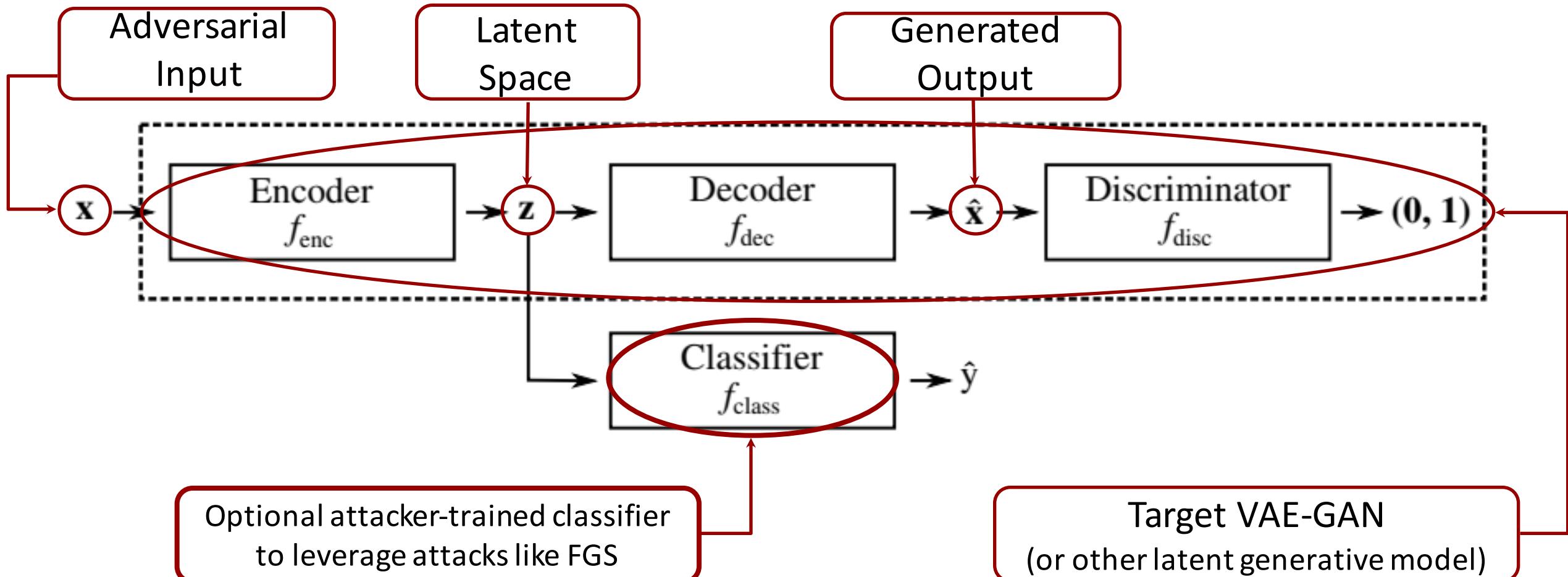
Adversarial Examples for Generative Models



Adversarial Examples for Generative Models

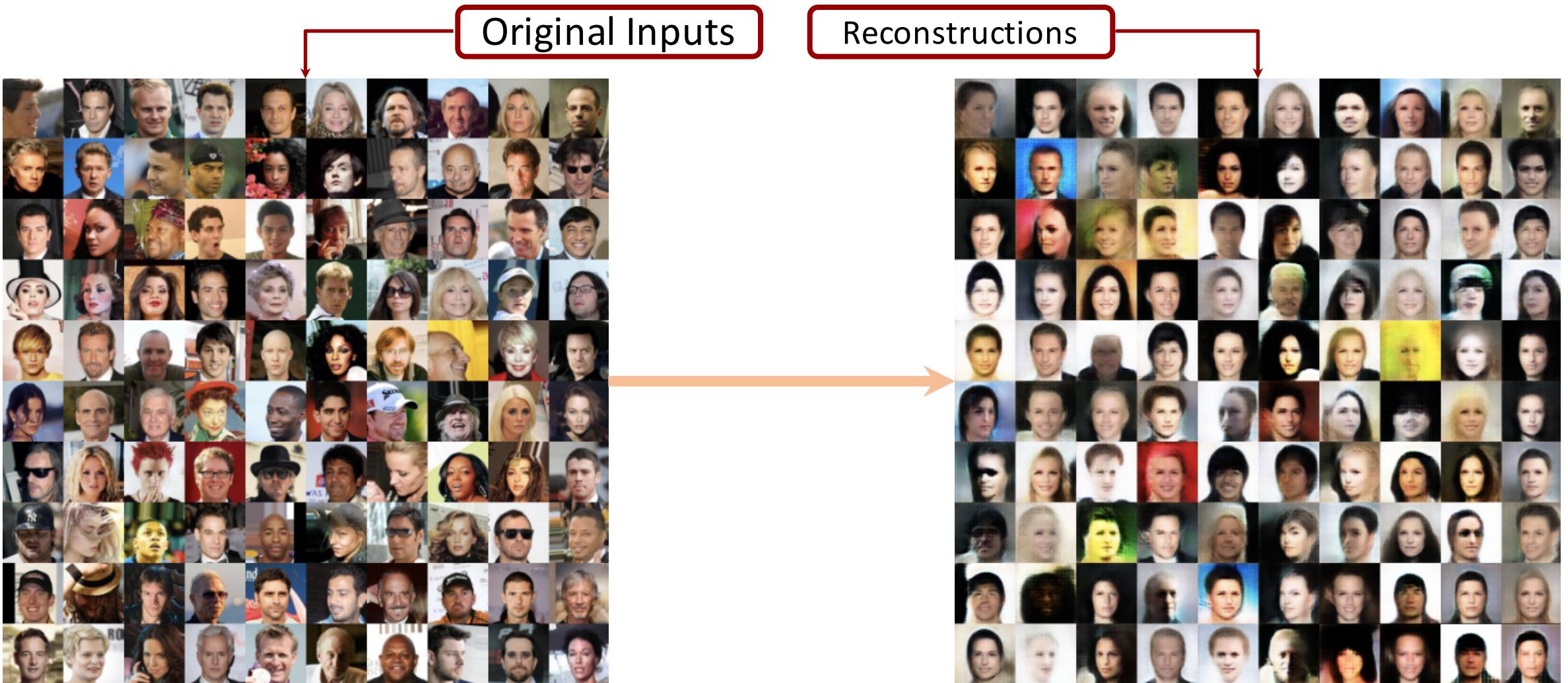


Adversarial Examples for Generative Models

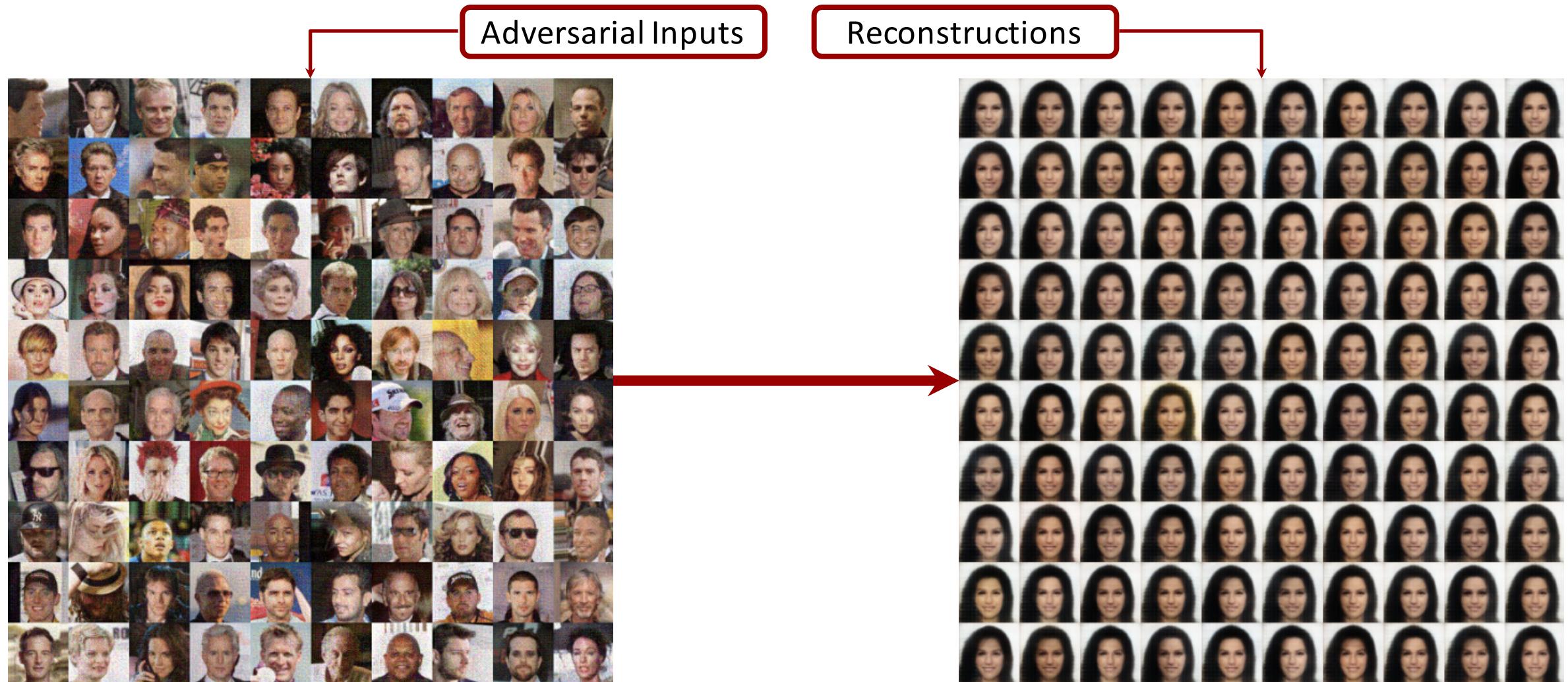


$$\arg \min_{x^*} L(x, x^*) \quad s.t. Oracle(G_{targ}(x^*)) = y^t$$

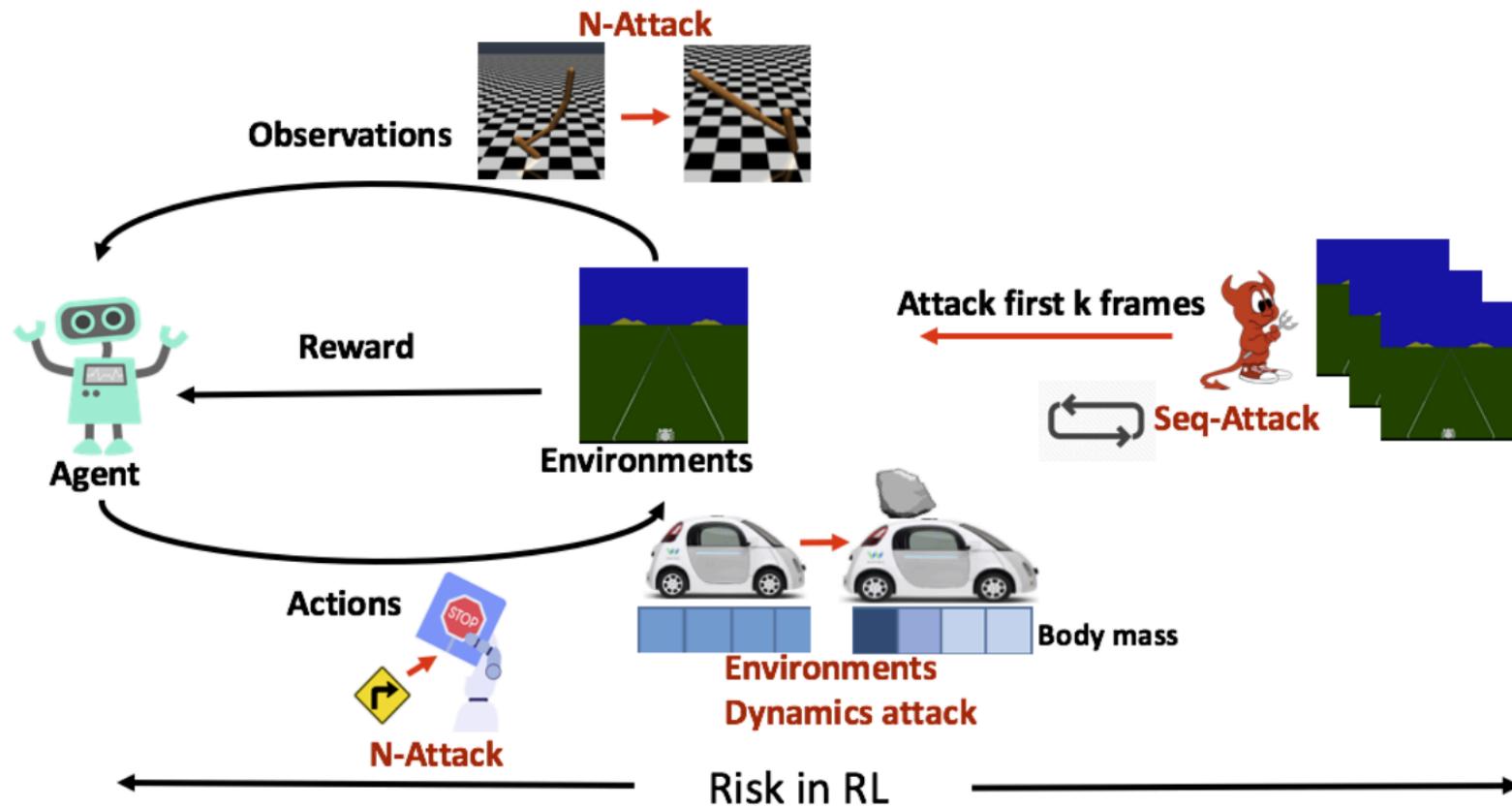
Adversarial Examples for Generative Models



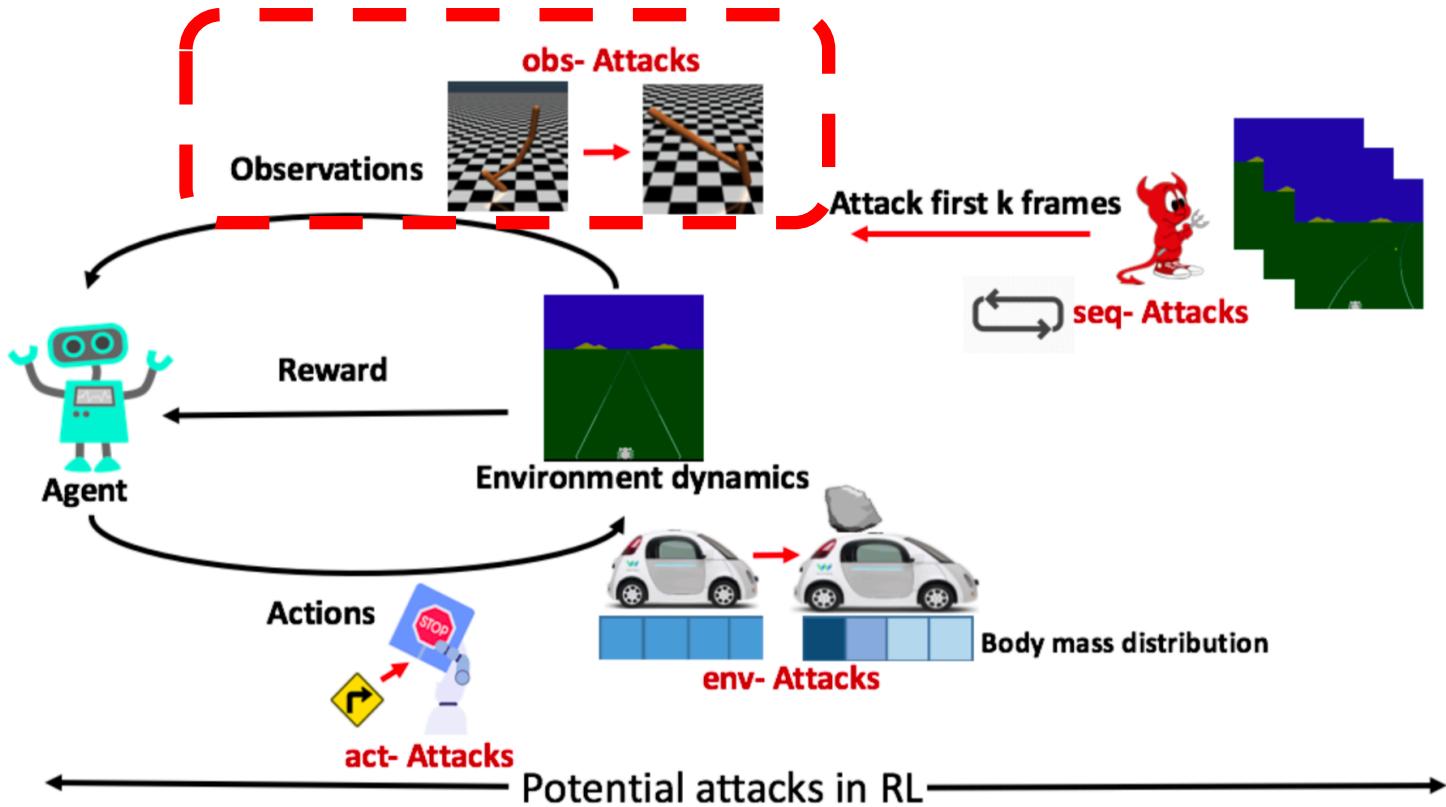
Adversarial Examples for Generative Models



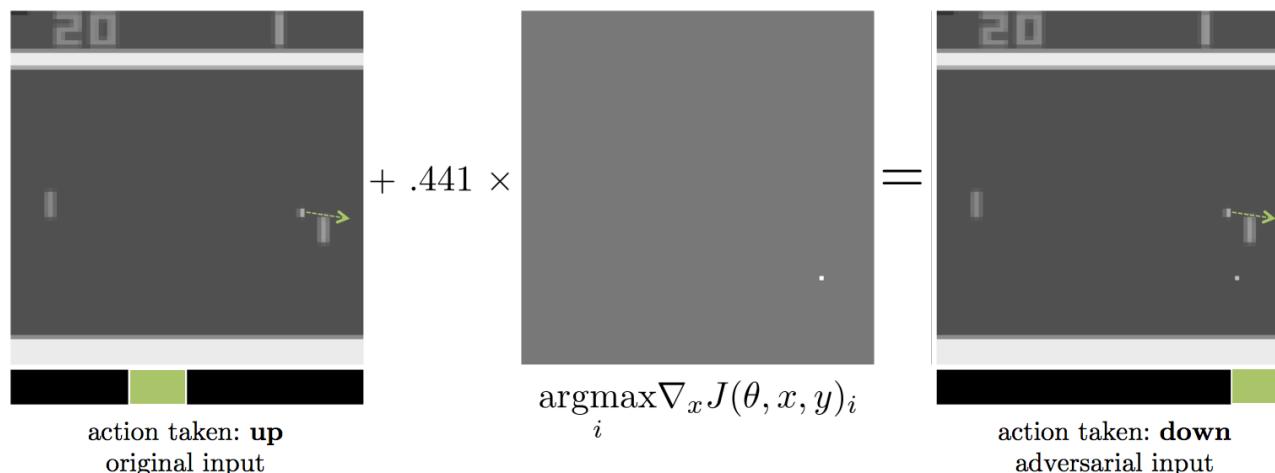
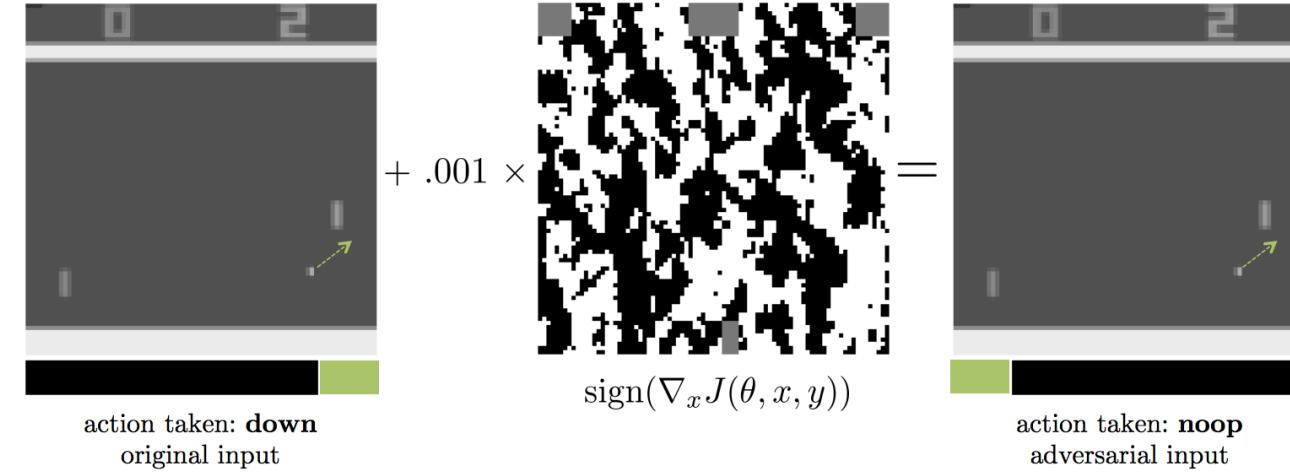
Attacking Deep Reinforcement Learning



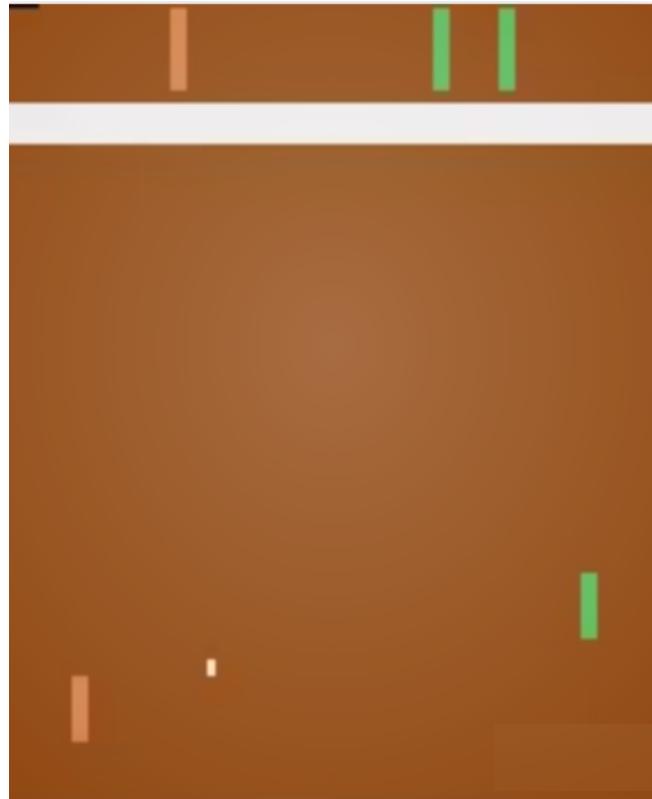
Attacking Deep Reinforcement Learning



Adversarial Attacks on Neural Network Policies



A3C: A Deep Policy on Pong



Reinforcement learning algorithms:

- Actor – **policy network** to predict the action based on each frame
- Critics – **value function** to predict the value of each frame, and the action is chosen to maximize the expected value
- Actor-critics (A3C) – combine value function into the policy network to make prediction

Agent in Action: attack the policy network

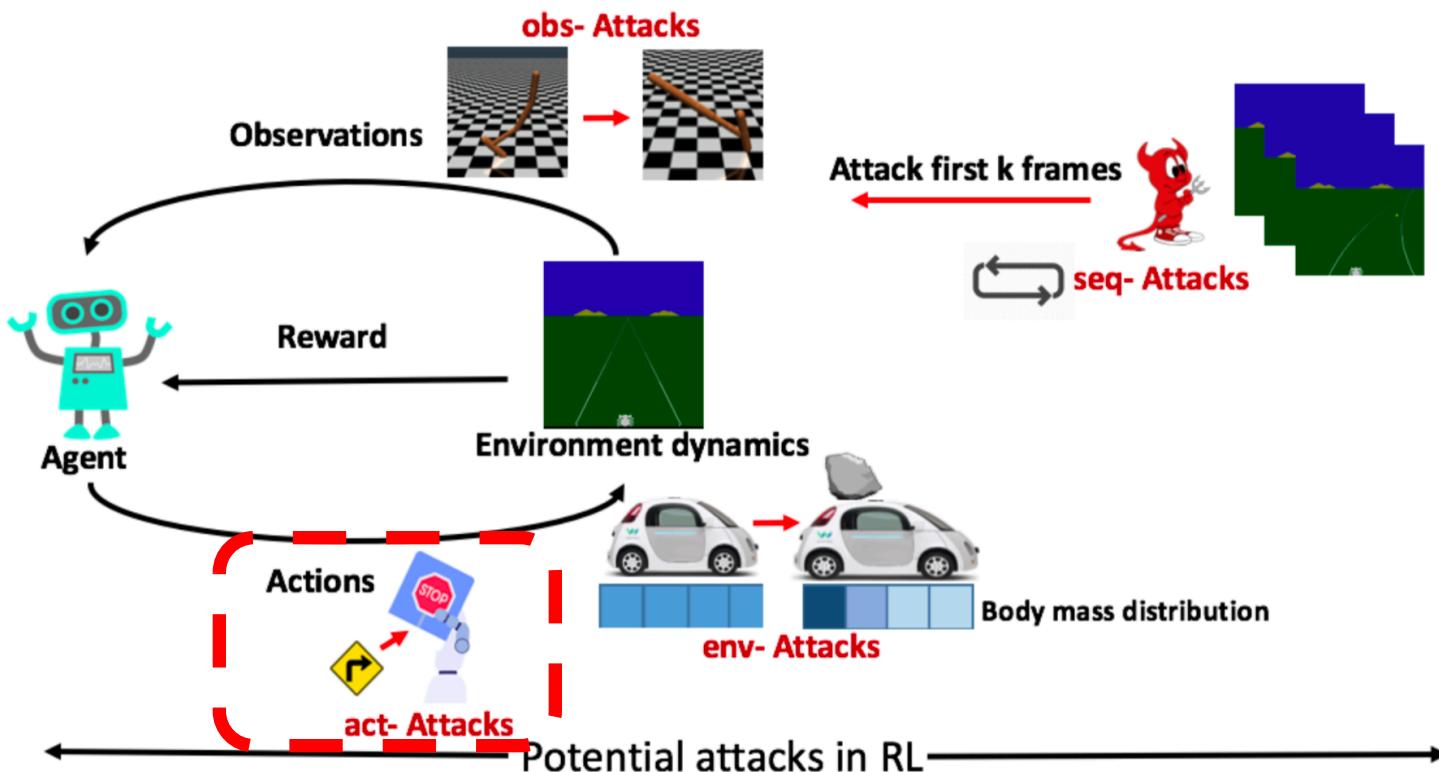


Original Frames

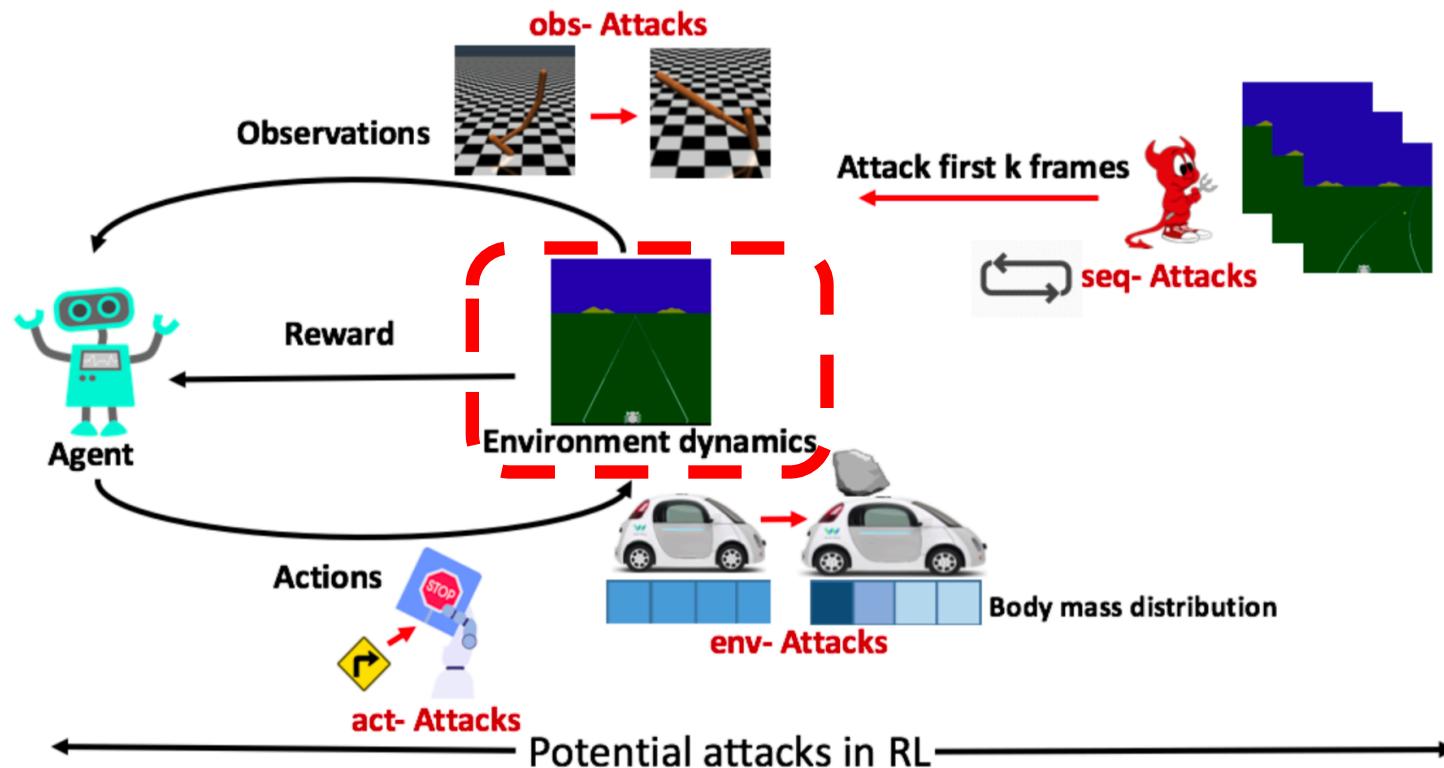


Adversarial perturbation
injected into every frame

Attacking Deep Reinforcement Learning



Attacking Deep Reinforcement Learning



Attacks on dynamic environments



Normal environment



Adversarial environment

