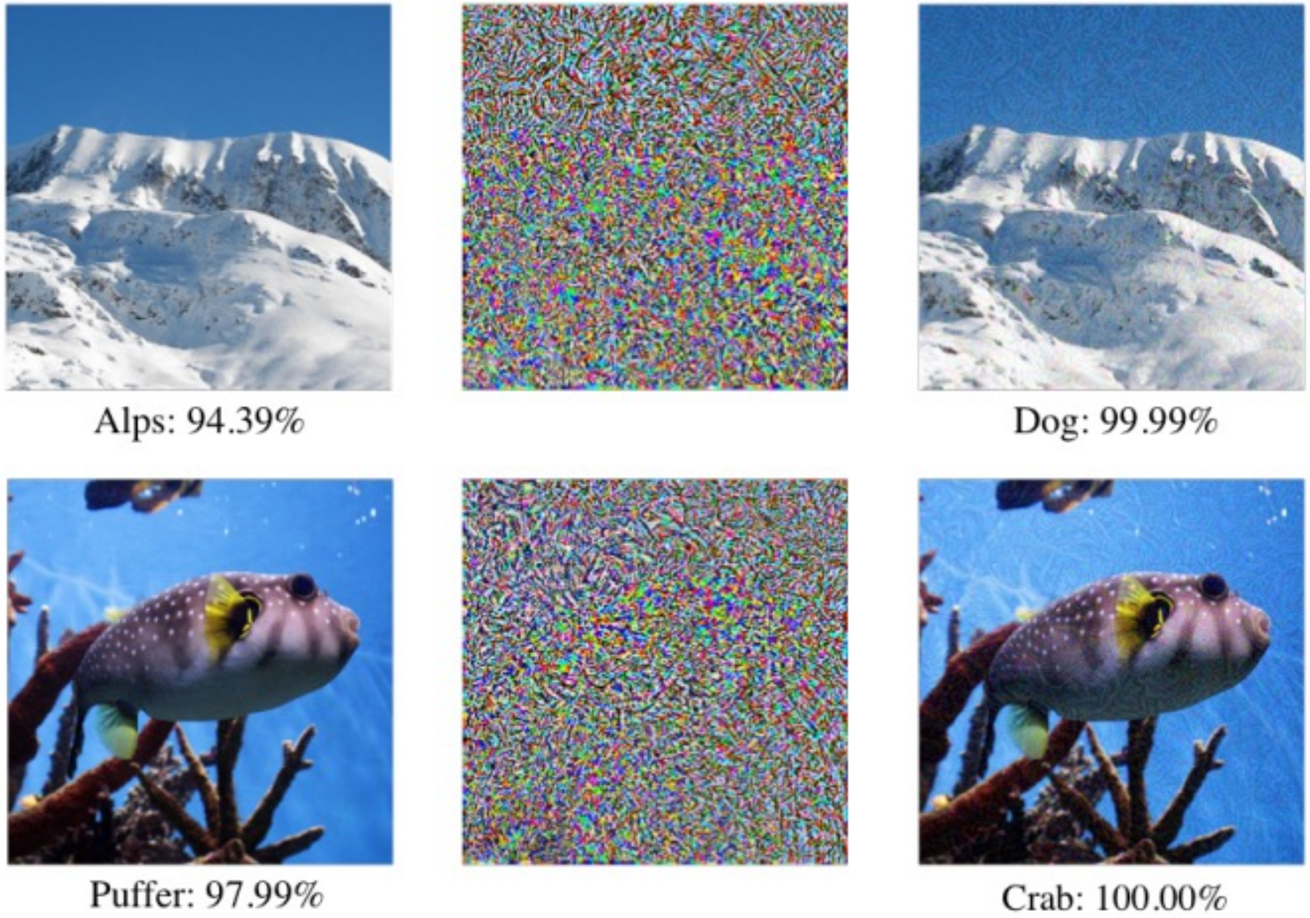
On Improving and Evaluating Adversarial Robustness

Tianyu Pang

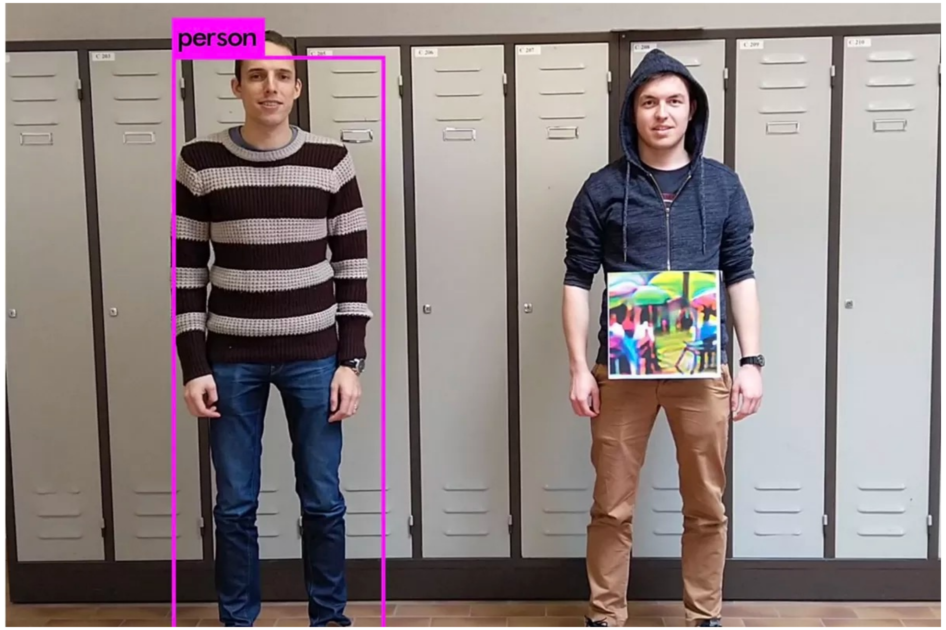
Sea AI Lab, Singapore

Adversarial vulnerability



[Dong et al. CVPR 2018]

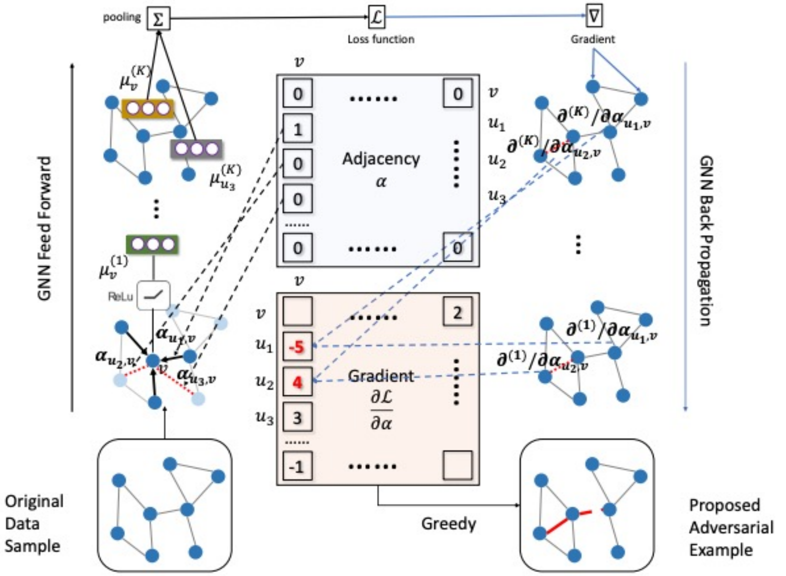
Adversarial examples in physical world



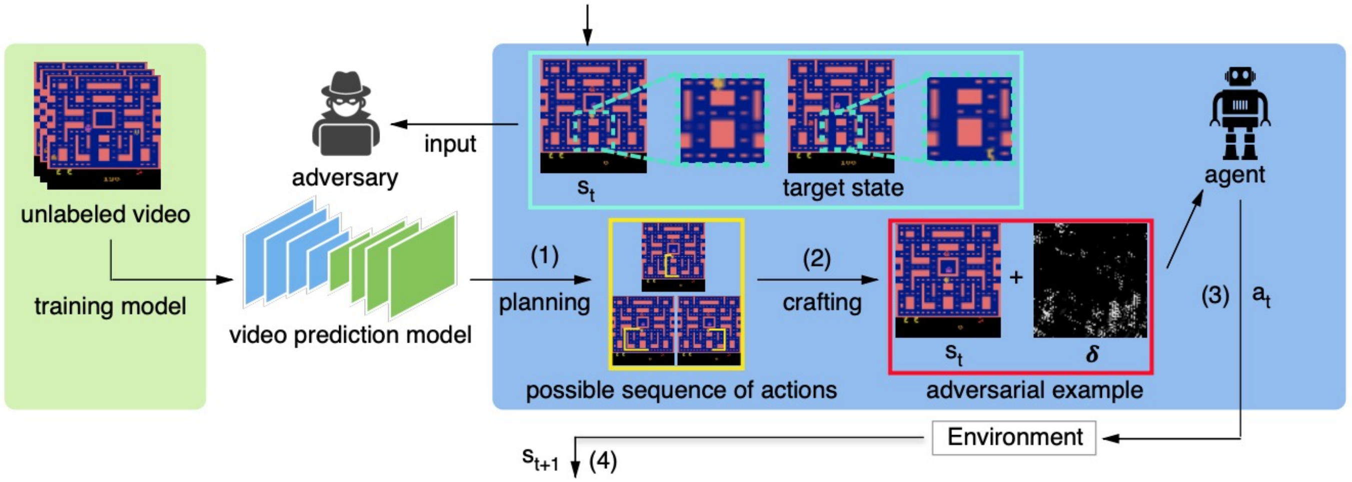
Not only in computer vision

Movie Review (Positive (POS) ↔ Negative (NEG))	
Original (Label: NEG)	The characters, cast in impossibly **contrived situations**, are **totally** estranged from reality.
Attack (Label: POS)	The characters, cast in impossibly **engineered circumstances**, are **fully** estranged from reality.
Original (Label: POS)	It cuts to the **knot** of what it actually means to face your **scares**, and to ride the **overwhelming** **metaphorical wave** that life wherever it takes you.
Attack (Label: NEG)	It cuts to the **core** of what it actually means to face your **fears**, and to ride the **big** **metaphorical wave** that life wherever it takes you.
SNLI (Entailment (ENT), Neutral (NEU), Contradiction (CON))	
Premise	Two small boys in blue soccer uniforms use a wooden set of steps to wash their hands.
Original (Label: CON)	The boys are in band **uniforms**.
Adversary (Label: ENT)	The boys are in band **garment**.
Premise	A child with wet hair is holding a butterfly decorated beach ball.
Original (Label: NEU)	The **child** is at the **beach**.
Adversary (Label: ENT)	The **youngster** is at the **shore**.

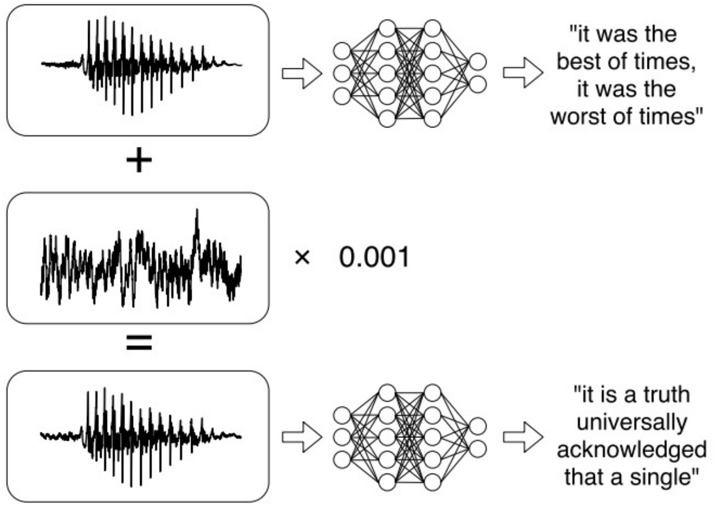
NLP (Jin et al. AAAI 2020)



Graph (Dai et al. ICML 2018)

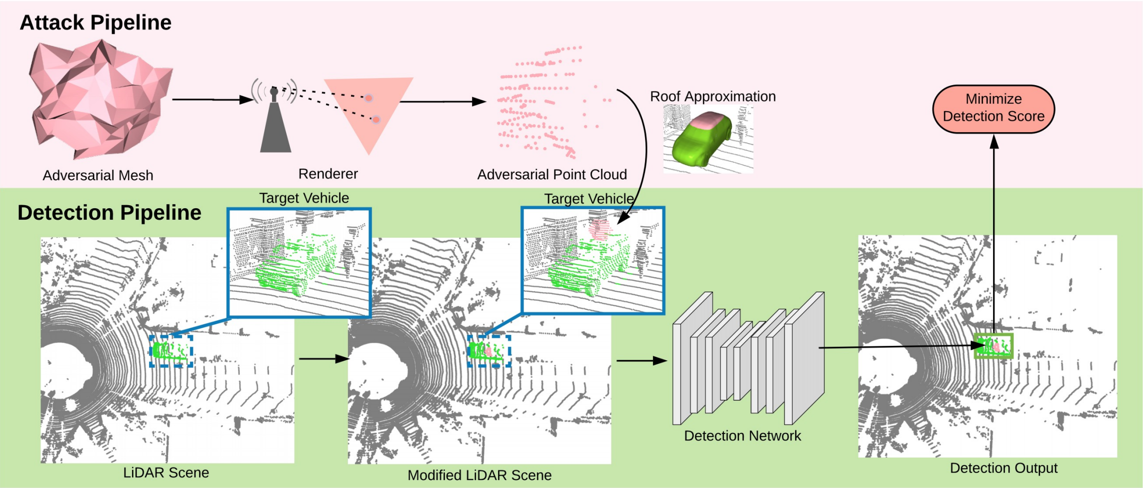


Reinforcement Learning (Lin et al. IJCAI 2017)

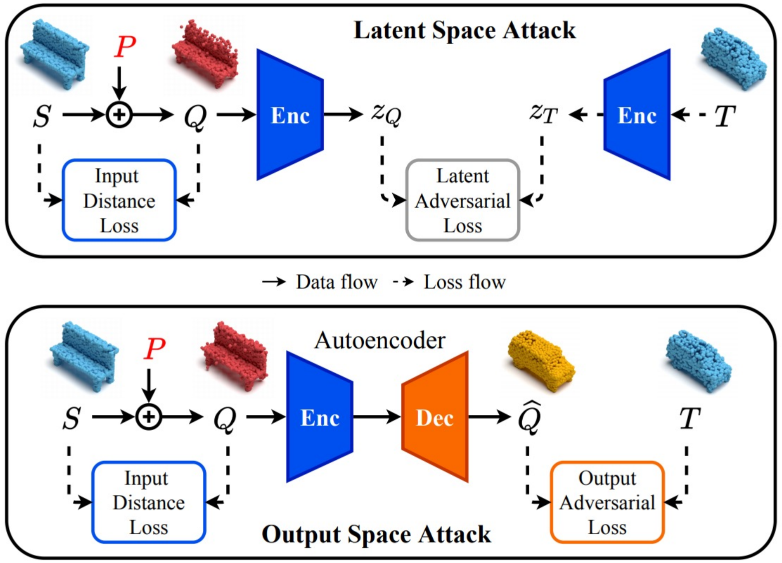


Audio (Carlini and Wagner. S&P 2018)

Not only in computer vision



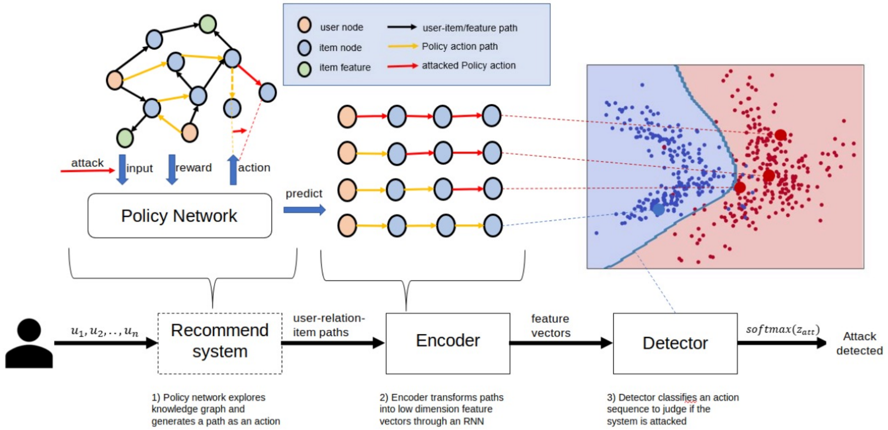
LiDAR (Tu et al. CVPR 2020)



3D Point Cloud (Lang et al. 2020)

Class	Representative Example
VC	**OD:** Given a string a, what is the length of a. **OO:** `(strlen a)` **AD:** Given a string b, what is the length of b. **AO:** `(strlen a)`
RR	**OD:** Given a number a, compute the product of **all** the numbers from 1 to a. **OO:** `(invoke1 (lambda1 (if ( ≤ arg1 1 )1(*( self( -arg1 1 )) arg1 ))) a)` **AD:** Given a number a, compute the product of the numbers from 1 to a. **AO:** `( * a 1 )`
SR	**OD:** consider an array of numbers, what is reverse of elements in the given array that are odd **OO:** `(reverse ( filter a ( lambda1 ( == ( % arg1 2 )1))))` **AD:** consider an array of numbers, what equals reverse of elements in the given array that are odd **AO:** `(reduce ( filter a ( lambda1 ( == ( % arg1 2 )1))))`

Code Generation (Anand et al. 2021)



Recommender System (Cao et al. SIGIR 2020)

Trade-off between robustness and accuracy

Empirically:

Standard training
clean accuracy 95%
robust accuracy 0%

Adversarial training
clean accuracy 85%
robust accuracy 50%

Where the trade-off stems from?

Theoretically:

Exists in some simple cases

[Zhang et al. ICML 2019; Tsipras et al. ICLR 2019]

What is an **accurate** model?

An **accurate** model refers to the one with **low standard error**:

$$\mathbf{R}_{\text{Standard}} = \mathbb{E}_{p_d(x)}\left[\text{KL}\left(p_d(y|x) \middle\| p_\theta(y|x)\right)\right]$$

data distribution (p_d) **model distribution** (p_θ)

Optimal solution: $p_{\theta^*}(y|x) = p_d(y|x)$

What is a **robust** model?

A **robust** model refers to the one with **low robust error**:

$$\mathbf{R}_{\text{Madry}} = \mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \text{KL}\left(p_d(y|x) \| p_\theta(y|x') \right) \right]$$

Optimal solution: $p_{\theta^*}(y|x) \neq p_d(y|x)$

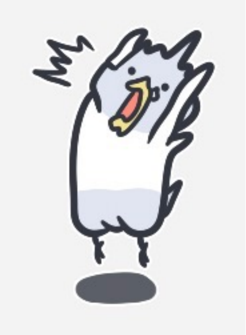
[Madry et al. ICLR 2018]

Trade-off naturally comes!

An optimally **accurate** model is **NOT** an optimally **robust** model

⬇ paradox

$p_d(y|x)$ is not an optimally **robust** model w.r.t. itself???!!!



Did we properly define robustness?

$$\mathbf{R}_{\text{Madry}} = \mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \mathrm{KL}\left(p_d(y|x) \| p_\theta(y|x')\right) \right]$$

⬆ differentiable surrogate

0-1 robust error: $\mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x)\right) \right]$

$\mathcal{Y}_\theta(x) = \mathrm{argmax}_y \; p_\theta(y|x)$

hard label of model distribution (i.e., predicted label)

$\mathcal{Y}_d(x) = \mathrm{argmax}_y \; p_d(y|x)$

hard label of data distribution (i.e., true label)

Did we properly define robustness?

0-1 robust error: $\mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x)\right)\right]$

true label is invariant in $B(x)$ ⬆

Self-consistent 0-1 robust error: $\mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x')\right)\right]$

- no assumption on $p_d(y|x)$
- allows for flexible $B(x)$

Did we properly define robustness?

$$\mathbf{R}_{\text{Madry}} = \mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \text{KL}\left(p_d(y|x) \middle\| p_\theta(y|x') \right) \right]$$

⬆ differentiable surrogate ($p_d(y|x)$ is invariant in $B(x)$)

Unreasonable (overcorrection towards smoothness)

$$\mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x) \right) \right]$$

⬆ true label is invariant in $B(x)$

Reasonable

$$\mathbb{E}_{p_d(x)} \left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x') \right) \right]$$

Self-COnsistent Robust Error (SCORE)

$$\mathbf{R}_{\text{SCORE}}(\theta) = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \text{KL}\left(p_d(y|x') \| p_\theta(y|x')\right)\right]$$

⬆ differentiable surrogate

$$\mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathbf{1}\left(\mathcal{Y}_\theta(x') \neq \mathcal{Y}_d(x')\right)\right]$$

$\mathbf{R}_{\text{Madry}}(\theta)$ **invariance** $\Longrightarrow$ $\mathbf{R}_{\text{SCORE}}(\theta)$ **equivariance**

Self-COnsistent Robust Error (SCORE)

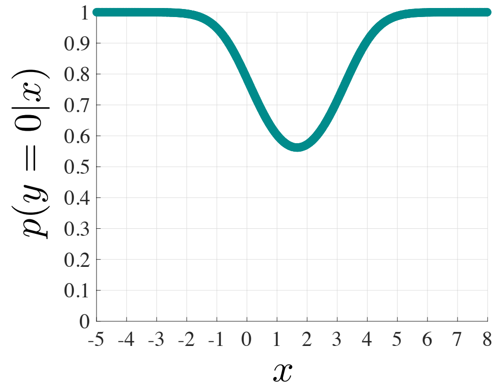
$$\mathbf{R}_{\mathrm{SCORE}}(\theta) = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathrm{KL}\left(p_d(y|x') \| p_\theta(y|x')\right)\right]$$

- Optimal solution: $p_{\theta^*}(y|x) = p_d(y|x)$

  (**self-consistency**, i.e., $p_d(y|x)$ is the optimally robust model w.r.t. itself under supervised learning framework)

- Keep the paradigm of robust optimization

Toy demo (self-consistency)

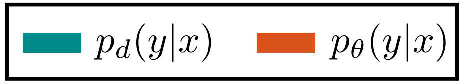


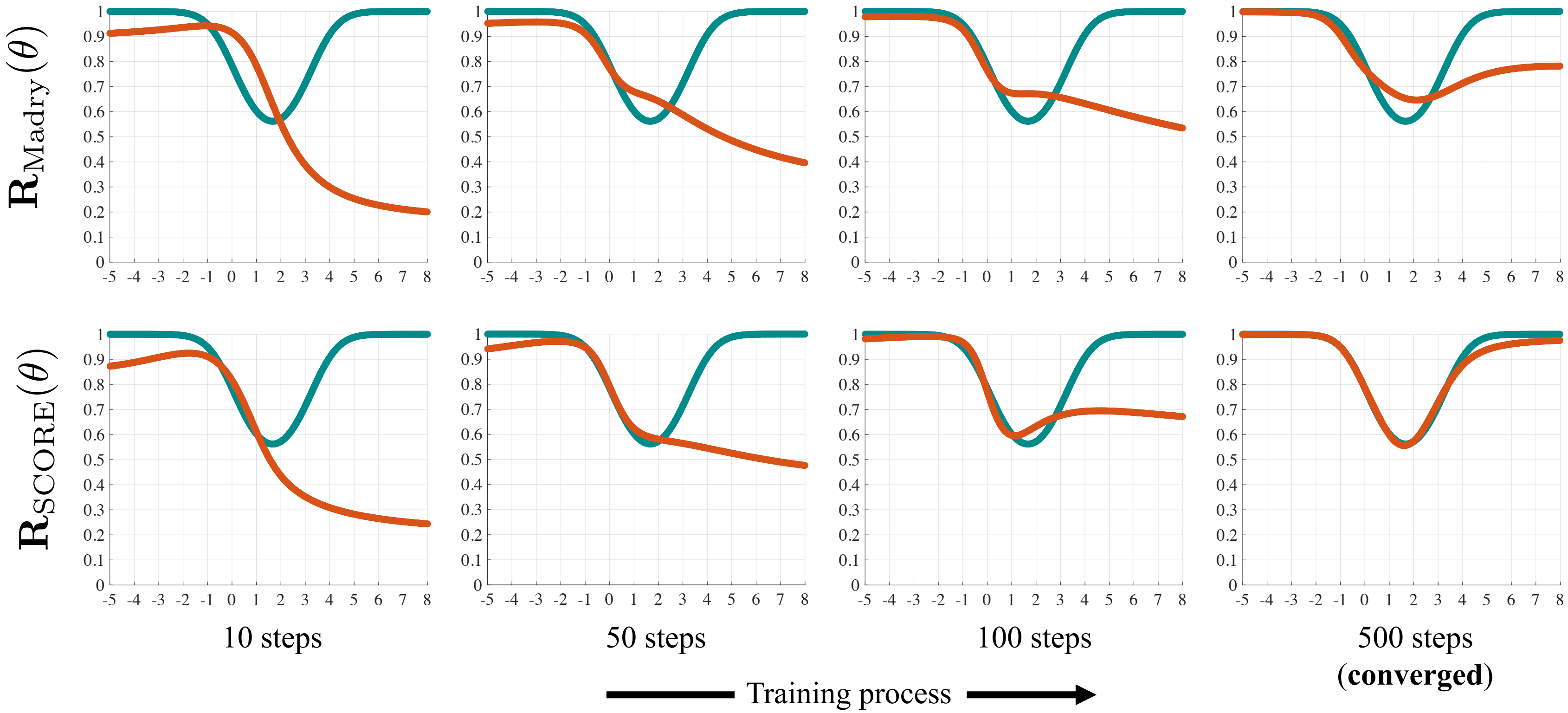
Construction of $p_d(x, y)$**:**

$p_d(y=0) = \frac{5}{6}, p_d(y=1) = \frac{1}{6};$

$p_d(x|y=0) \sim \mathcal{N}(-1, 4);$

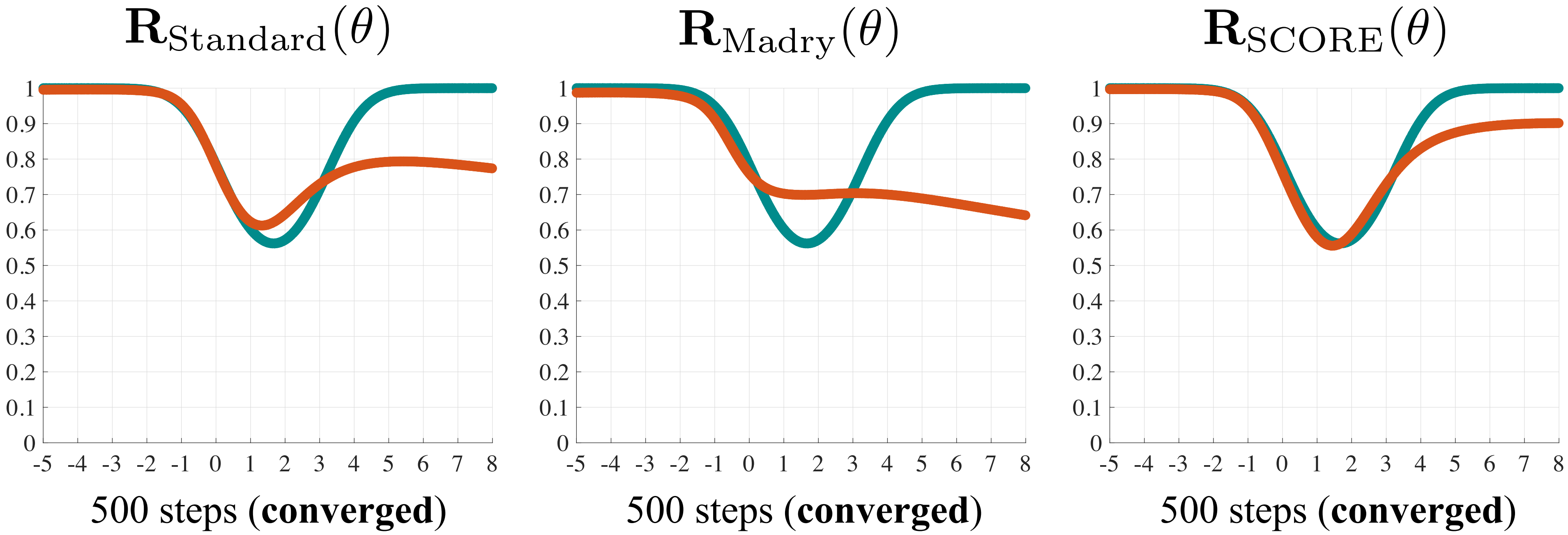
$p_d(x|y=1) \sim \mathcal{N}(1, 1).$





60,000 training pairs, mimics the expectation form

Toy demo (robust optimization)



6 training pairs, mimics the finite-sample form

Standard error has the same optimal solution as SCORE, but does not enjoy robust optimization in finite-sample cases

In practice, how to optimize SCORE?

Directly applying first-order optimizers requires:

$$
\begin{aligned}
&\nabla_x \mathrm{KL}\left(p_d(y|x) \| p_\theta(y|x)\right) \\
=&\mathbb{E}_{p_d(y|x)}\Big[-\underbrace{\nabla_x \log p_\theta(y|x)}_{\textbf{model gradient}} + \left(\log \frac{p_d(y|x)}{p_\theta(y|x)}\right) \cdot \underbrace{\nabla_x \log p_d(y|x)}_{\textbf{data gradient}}\Big]
\end{aligned}
$$

- Initial experiments using **score matching** are of high variance
- More advanced score matching like **[Chao et al. ICLR 2022]** could be explored

Goodbye KL divergence!

Substitute KL divergence with any **distance metric** $\mathcal{D}$

does not satisfy

- Symmetry: $\mathcal{D}(A\|B) = \mathcal{D}(B\|A)$
- Triangle inequality: $\mathcal{D}(A\|C) \leq \mathcal{D}(A\|B) + \mathcal{D}(B\|C)$

Typical distance metrics include $\|A - B\|_p$

Goodbye KL divergence!

Substitute KL divergence with any **distance metric** $\mathcal{D}$

$$\mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathcal{D}\left(p_d(y|x) \middle\| p_\theta(y|x')\right)\right];$$

$$\mathbf{R}^{\mathcal{D}}_{\text{SCORE}}(\theta) = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathcal{D}\left(p_d(y|x') \middle\| p_\theta(y|x')\right)\right]$$

Upper and lower bounds for SCORE

Theorem 1:

$$|\mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) - C^{\mathcal{D}}| \leq \mathbf{R}^{\mathcal{D}}_{\text{SCORE}}(\theta) \leq \mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) + C^{\mathcal{D}},$$

$$\text{where } \underline{C^{\mathcal{D}}} = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathcal{D}\left(p_d(y|x) \| p_d(y|x')\right)\right]$$

intrinsic property of data distribution, indicates the (Madry) robust error of $p_d(y|x)$ itself

Upper and lower bounds for SCORE

Theorem 1:

$$|\mathbf{R}^{\mathcal{D}}_{\mathrm{Madry}}(\theta) - C^{\mathcal{D}}| \leq \mathbf{R}^{\mathcal{D}}_{\mathrm{SCORE}}(\theta) \leq \mathbf{R}^{\mathcal{D}}_{\mathrm{Madry}}(\theta) + C^{\mathcal{D}},$$

$$\text{where } C^{\mathcal{D}} = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathcal{D}\left(p_d(y|x) \| p_d(y|x')\right)\right]$$

- **Upper bound:** minimizing SCORE without estimating $\nabla_x \log p_d(y|x)$
- **Lower bound:** indicates the overfitting phenomenon

Upper and lower bounds for SCORE

Theorem 1:

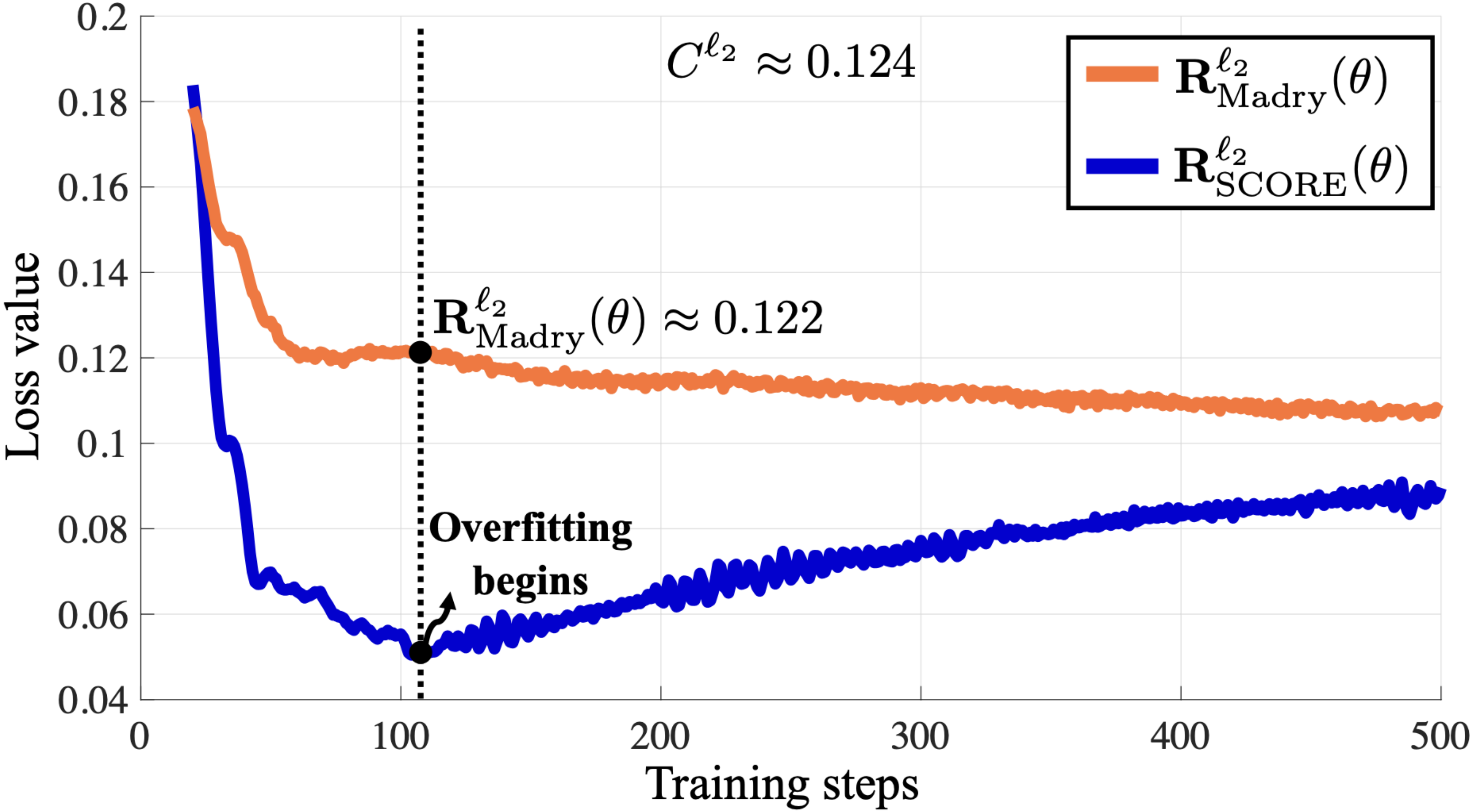
$$|\mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) - C^{\mathcal{D}}| \leq \mathbf{R}^{\mathcal{D}}_{\text{SCORE}}(\theta) \leq \mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) + C^{\mathcal{D}},$$

$$\text{where } C^{\mathcal{D}} = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathcal{D}\left(p_d(y|x) \| p_d(y|x')\right)\right]$$

- **Upper bound:** minimizing SCORE without estimating $\nabla_x \log p_d(y|x)$
- **Lower bound:** indicates the overfitting phenomenon

Upper and lower bounds for SCORE

$\mathcal{D}$ is ℓ_2-distance : $\|A - B\|_2$



Extending to composite function of distance

Theorem 2:

$$|\mathbf{R}^{\mathcal{D}}_{\mathrm{SCORE}}(\theta) - C^{\mathcal{D}}| \leq \phi^{-1}\left(\mathbf{R}^{\phi\circ\mathcal{D}}_{\mathrm{Madry}}(\theta)\right)$$

$\phi(\cdot)$ is a **monotonically increasing convex** function, e.g., square function

Examples of $\phi \circ \mathcal{D}$ include **squared error (SE)** and **JS-divergence**

Composite function of distance empirically works better

Loss	Alias	$l.r. = 0.1$		$l.r. = 0.05$		$l.r. = 0.01$	
		Clean	PGD	Clean	PGD	Clean	PGD
$\|P - Q\|_2$	ℓ_2-dis.	75.91	52.16	77.98	52.74	78.45	51.13
$\|P - Q\|_1$	ℓ_1-dis.	58.51	43.87	64.88	46.77	70.02	47.76
$\|P - Q\|_\infty$	ℓ_∞-dis.	58.34	43.71	59.75	45.02	65.65	46.36
$\sqrt{\mathrm{JS}(P\|Q)}$	JS-dis.	53.06	40.08	55.27	41.86	68.50	46.49
$\mathrm{JS}(P\|Q)$	JS-div.	79.41	51.75	81.27	51.85	80.12	49.10
$\mathrm{KL}(P\|Q)$	KL-div.	82.74	53.02	83.21	51.52	82.65	47.45
$\|P - Q\|_1^2$	-	79.87	50.96	81.49	52.00	81.26	47.51
$\|P - Q\|_2^2$	SE	80.59	54.63	83.38	54.01	81.43	51.13

PGD-AT and TRADES are equivalent (under $\mathcal{D}$)

Theorem 3: For $\beta \geq 1$

$$\mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta) \leq \mathbf{R}^{\mathcal{D}}_{\text{TRADES}}(\theta;\beta) \leq (1+2\beta) \cdot \mathbf{R}^{\mathcal{D}}_{\text{Madry}}(\theta)$$

- Similar as the equivalence among ℓ_p-norms
- Induce the same topology of loss landscapes in parameter space [Conrad 2018]

Back to KL divergence with new insights

A bridge between **KL divergence** and **distance metrics**:
Pinsker's inequality

$$\frac{1}{2}\|P - Q\|_1^2 \leq \mathrm{KL}(P||Q)$$

[Csiszar and Korner 2011]

Back to KL divergence with new insights

Corollary 1:

$$|\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) - C^{\ell_1}| \leq \sqrt{2 \cdot \underline{\mathbf{R}_{\mathrm{Madry}}(\theta)}}$$

original **KL-based** robust error

Explaining overfitting and early-stopping

minimized in previous work

$$|\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) - C^{\ell_1}| \leq \sqrt{2 \cdot \boxed{\mathbf{R}_{\mathrm{Madry}}(\theta)}}$$

Explaining overfitting and early-stopping

$$|\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) - C^{\ell_1}| \leq \sqrt{2 \cdot \mathbf{R}_{\mathrm{Madry}}(\theta)}$$

$$\Downarrow \quad \boxed{\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) = 0}$$

condition for optimal solution

Explaining overfitting and early-stopping

$$|\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) - C^{\ell_1}| \leq \sqrt{2 \cdot \mathbf{R}_{\mathrm{Madry}}(\theta)}$$

$$\Downarrow \quad \mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) = 0$$

indicates early-stopping

$$C^{\ell_1} \leq \sqrt{2 \cdot \mathbf{R}_{\mathrm{Madry}}(\theta)} \Longrightarrow \boxed{\mathbf{R}_{\mathrm{Madry}}(\theta) \geq \frac{\left(C^{\ell_1}\right)^2}{2}}$$

Explaining overfitting and early-stopping

$$\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) = 0 \Rightarrow p_\theta(y|x) = p_d(y|x)$$

Explaining overfitting and early-stopping

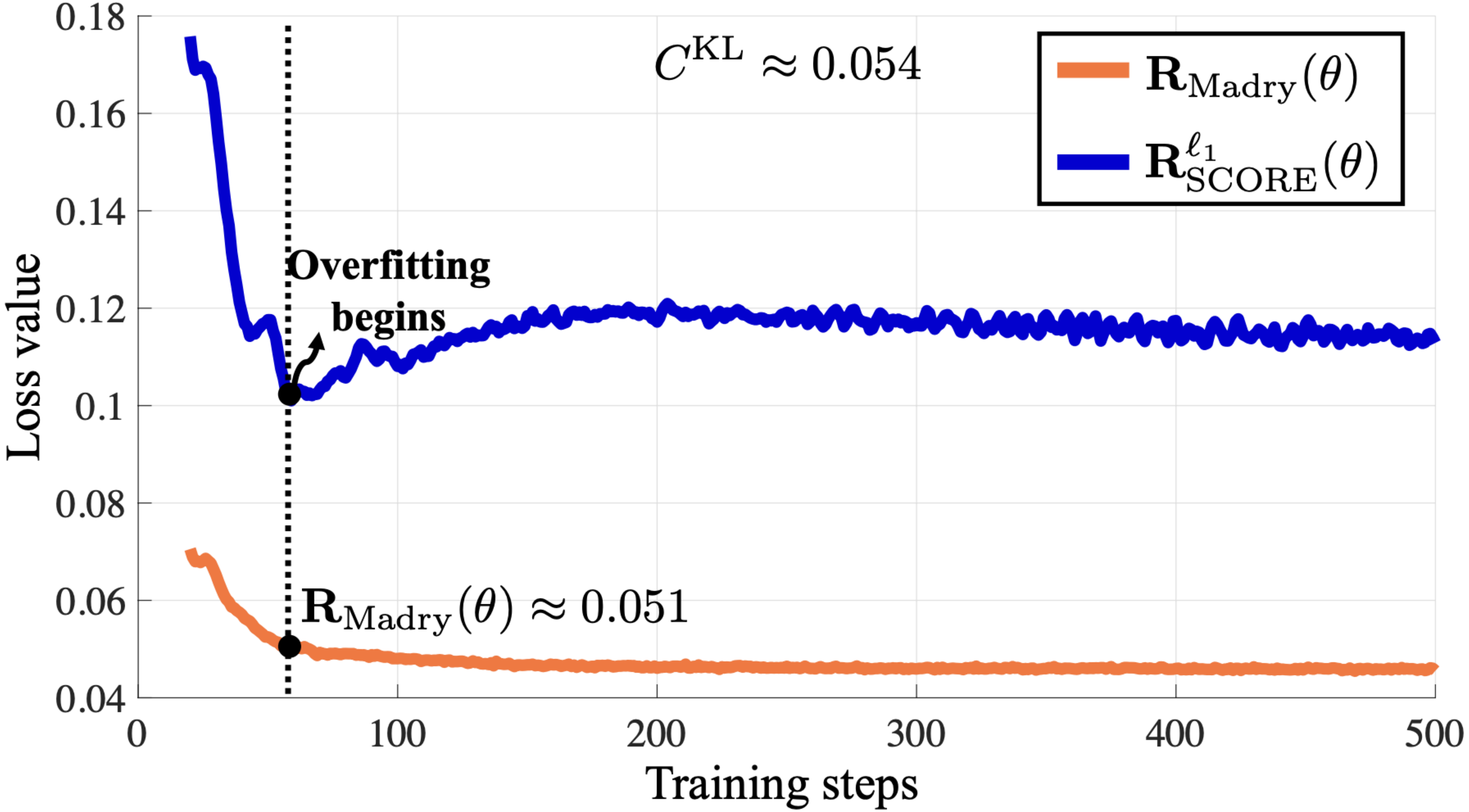
$$\mathbf{R}^{\ell_1}_{\mathrm{SCORE}}(\theta) = 0 \Rightarrow p_\theta(y|x) = p_d(y|x)$$

$$\Downarrow$$

$$\mathbf{R}_{\mathrm{Madry}}(\theta) = C^{\mathrm{KL}} \geq \frac{(C^{\ell_1})^2}{2}$$

$$\text{where } C^{\mathrm{KL}} = \mathbb{E}_{p_d(x)}\left[\max_{x' \in B(x)} \mathrm{KL}\left(p_d(y|x) \| p_d(y|x')\right)\right]$$

Explaining overfitting and early-stopping



Explaining semantic gradients (for adversarial training)

Theorem 4: (under mild condition)

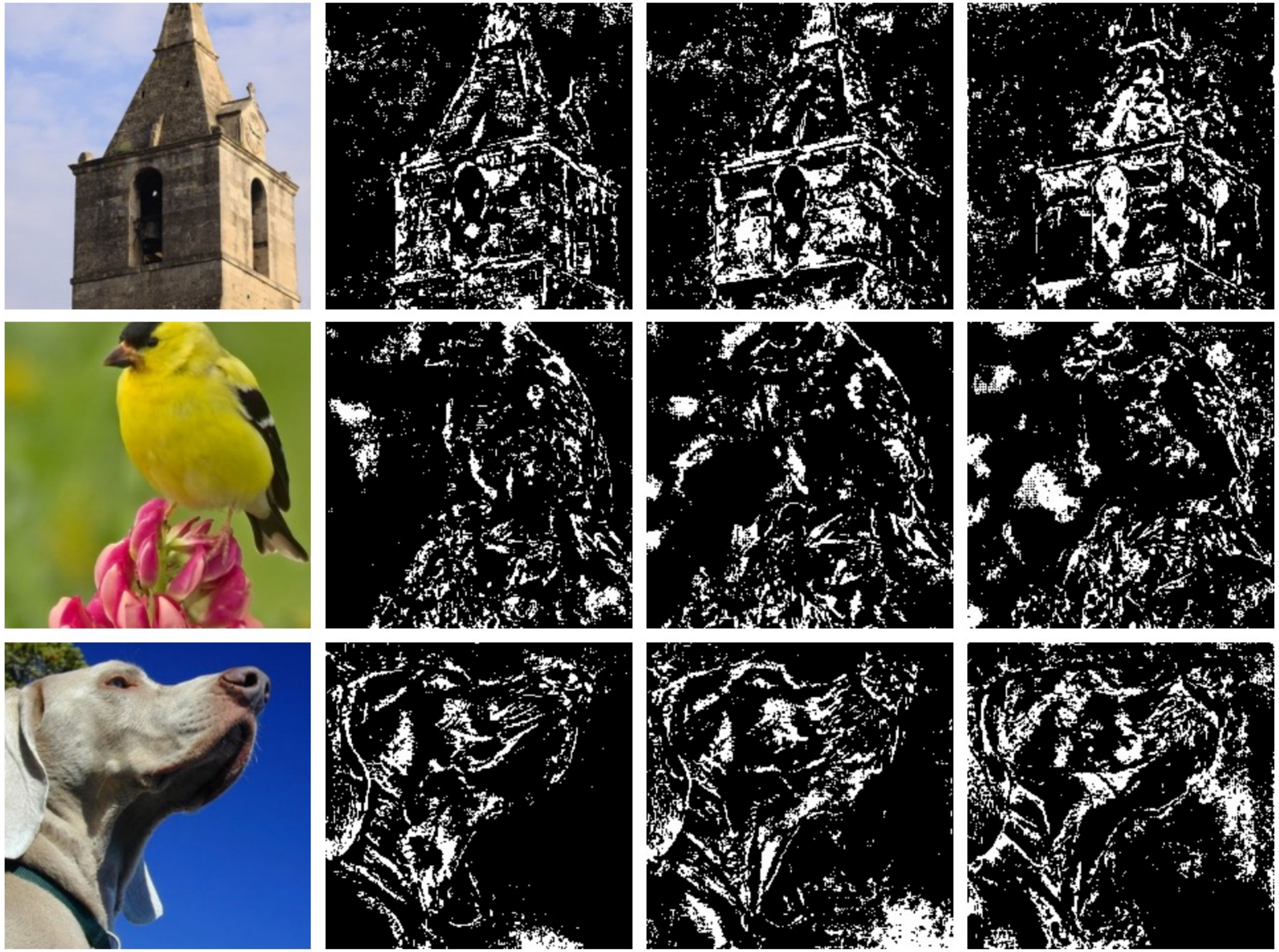
$$\mathbf{R}^{\ell_1}_{\text{SCORE}}(\theta) = \mathbf{R}^{\ell_1}_{\text{Standard}}(\theta) +$$

$$2\epsilon \cdot \mathbb{E}_{p_d(x)}\left[\left\| \nabla_x p_d(\mathcal{Y}_d(x)|x) - \nabla_x p_\theta(\mathcal{Y}_d(x)|x) \right\|_q \right] + o(\epsilon)$$

alignment between **model gradient** and **data gradient**

where $\mathcal{Y}_d(x) = \text{argmax}_y p_d(y|x)$

Explaining semantic gradients (for adversarial training)



Clean images $\nabla_x \log p_\theta(\mathcal{Y}_d(x)|x)$ $\nabla_x \log p_\theta(x, \mathcal{Y}_d(x))$ $-\nabla_x \log p_\theta(x)$

Empirical performance

Table 2. Classification accuracy (%) on clean images and under AutoAttack (ℓ_∞, $\epsilon = 8/255$). Here we use ResNet-18 trained by PGD-AT or TRADES on CIFAR-10, using KL divergence or squared error (SE) as the loss function. Clipping loss is executed at every training step, compatible with early-stopping. We average the results over five runs and report the mean $\pm$ standard deviation.

Method	Loss	Clip	Clean	AutoAttack
PGD-AT	KL div.	-	82.46 ± 0.41	48.39 ± 0.14
	SE	✗	82.13 ± 0.14	49.41 ± 0.27
	SE	✓	**82.80** $\pm$ **0.16**	**49.63** $\pm$ **0.17**
TRADES	KL div.	-	81.47 ± 0.12	49.14 ± 0.16
	SE	✗	83.50 ± 0.05	49.44 ± 0.35
	SE	✓	**83.75** $\pm$ **0.14**	**49.57** $\pm$ **0.28**

Table 3. Classification accuracy (%) on clean images and under AutoAttack (ℓ_∞, $\epsilon = 8/255$). The model is WRN-28-10 (SiLU), following the training pipeline in Rebuffi et al. (2021) and using 1M DDPM generated data. KL divergence is substituted with the SE function in TRADES, and no clipping loss is executed.

Dataset	β	Clean	AutoAttack
CIFAR-10	6	86.64 ± 0.13	60.78 ± 0.16
	5	87.19 ± 0.20	61.05 ± 0.11
	4	87.89 ± 0.19	61.11 ± 0.27
	3	88.60 ± 0.13	60.89 ± 0.09
	2	89.28 ± 0.15	60.13 ± 0.21
CIFAR-100	4	61.94 ± 0.13	31.21 ± 0.12
	3	63.12 ± 0.37	31.01 ± 0.09

Table 4. Classification accuracy (%) on clean images and under AutoAttack. The results of our methods are in **bold**, and no clipping loss is executed. Here ‡ means *no CutMix applied*, following Rade and Moosavi-Dezfooli (2021). We use a batch size of 512 and train for 400 epochs due to limited resources, while a larger batch size of 1024 and training for 800 epochs are expected to achieve better performance.

Dataset	Method	Architecture	DDPM	Batch	Epoch	Clean	AutoAttack
CIFAR-10 ($\ell_\infty, \epsilon = 8/255$)	Rice et al. (2020)	WRN-34-20	✗	128	200	85.34	53.42
	Zhang et al. (2020)	WRN-34-10	✗	128	120	84.52	53.51
	Pang et al. (2021)	WRN-34-20	✗	128	110	86.43	54.39
	Wu et al. (2020)	WRN-34-10	✗	128	200	85.36	56.17
	Gowal et al. (2020)	WRN-70-16	✗	512	200	85.29	57.14
	Rebuffi et al. (2021)‡	WRN-28-10	1M	1024	800	85.97	60.73
	+ **Ours** (KL → SE, $\beta = 3$)	WRN-28-10	1M	512	400	**88.61**	**61.04**
	+ **Ours** (KL → SE, $\beta = 4$)	WRN-28-10	1M	512	400	**88.10**	**61.51**
	Rebuffi et al. (2021)‡	WRN-70-16	1M	1024	800	86.94	63.58
	+ **Ours** (KL → SE, $\beta = 3$)	WRN-70-16	1M	512	400	**89.01**	**63.35**
	+ **Ours** (KL → SE, $\beta = 4$)	WRN-70-16	1M	512	400	**88.57**	**63.74**
	Gowal et al. (2021)	WRN-70-16	100M	1024	2000	88.74	66.10
CIFAR-10 ($\ell_2, \epsilon = 128/255$)	Wu et al. (2020)	WRN-34-10	✗	128	200	88.51	73.66
	Gowal et al. (2020)	WRN-70-16	✗	512	200	90.90	74.50
	Rebuffi et al. (2021)‡	WRN-28-10	1M	1024	800	90.24	77.37
	+ **Ours** (KL → SE, $\beta = 3$)	WRN-28-10	1M	512	400	**91.52**	**77.89**
	+ **Ours** (KL → SE, $\beta = 4$)	WRN-28-10	1M	512	400	**90.83**	**78.10**
CIFAR-100 ($\ell_\infty, \epsilon = 8/255$)	Wu et al. (2020)	WRN-34-10	✗	128	200	60.38	28.86
	Gowal et al. (2020)	WRN-70-16	✗	512	200	60.86	30.03
	Rebuffi et al. (2021)‡	WRN-28-10	1M	1024	800	59.18	30.81
	+ **Ours** (KL → SE, $\beta = 3$)	WRN-28-10	1M	512	400	**63.66**	**31.08**
	+ **Ours** (KL → SE, $\beta = 4$)	WRN-28-10	1M	512	400	**62.08**	**31.40**
	Rebuffi et al. (2021)‡	WRN-70-16	1M	1024	800	60.46	33.49
	+ **Ours** (KL → SE, $\beta = 3$)	WRN-70-16	1M	512	400	**65.56**	**33.05**
	+ **Ours** (KL → SE, $\beta = 4$)	WRN-70-16	1M	512	400	**63.99**	**33.65**

Robustness and Accuracy Could Be Reconcilable by (Proper) Definition

Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, Shuicheng Yan

ICML 2022

Better Diffusion Models Further Improve Adversarial Training

Zekai Wang*, Tianyu Pang*, Chao Du, Min Lin, Weiwei Liu, Shuicheng Yan

ICML 2023

On Evaluating Adversarial Robustness of Large Vision-Language Models

Yunqing Zhao*, Tianyu Pang*, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, Min Lin

NeurIPS 2023

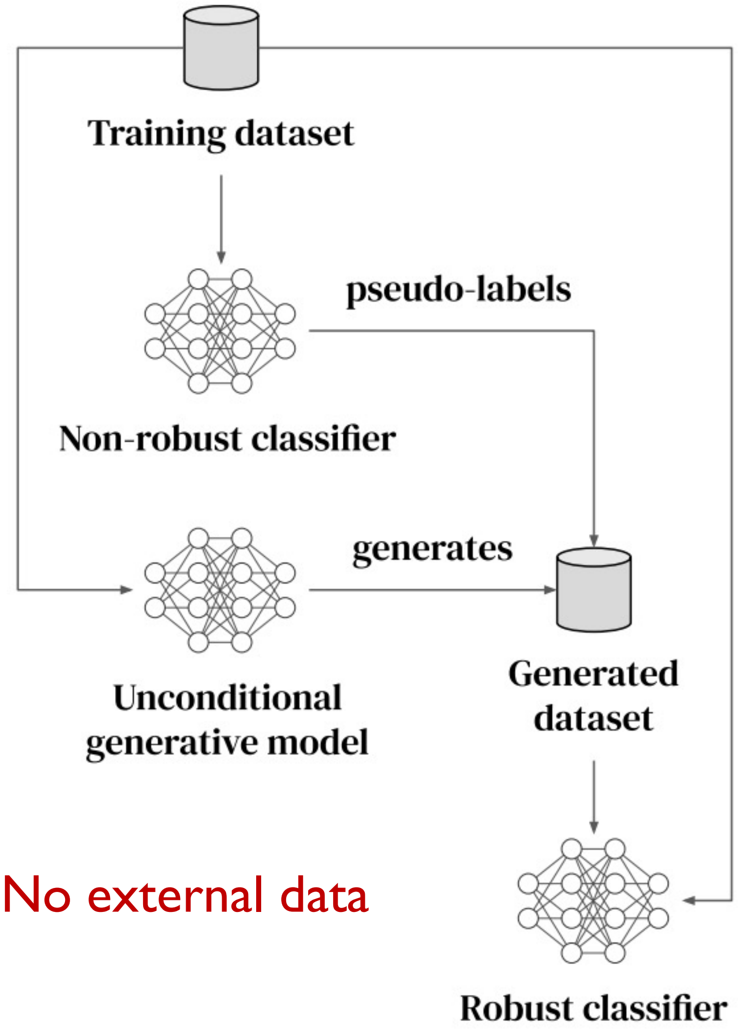
Wait! Why does empirical trade-off still exit?

SCORE makes sure that there is no trade-off for the **optimal solution**, so the remain challenge leaves to **more efficient learning processes.**

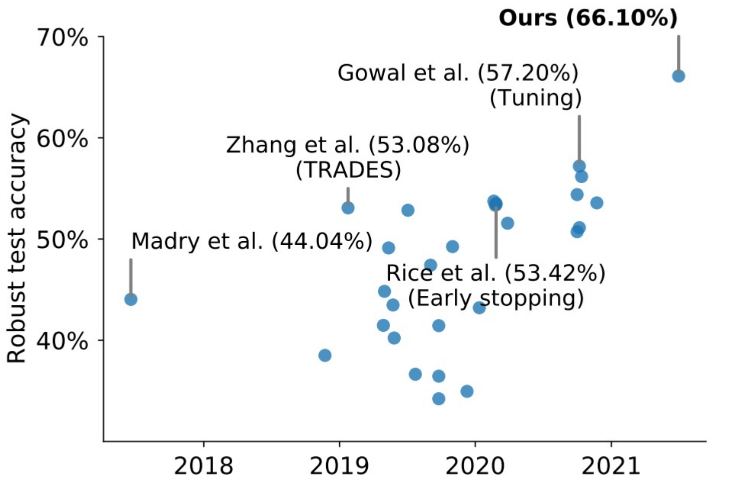
- Beyond MLE (KL divergence), resorting to more advanced score matching methods (Fisher divergence) to train SCORE

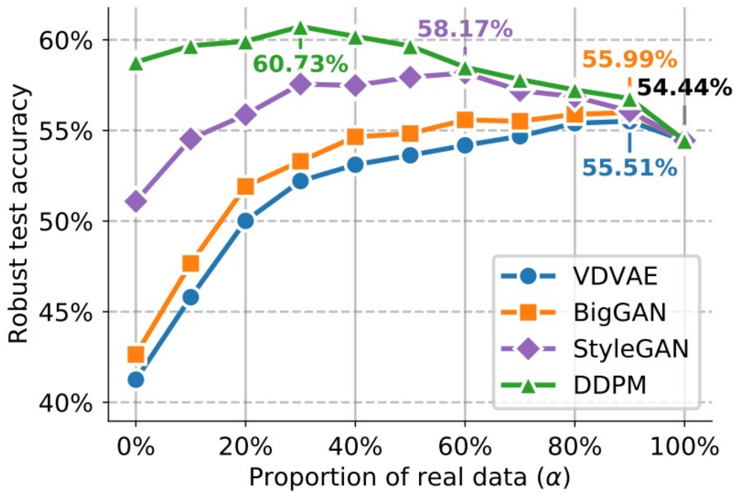
- Extra data; robust architectures; training tricks

Diffusion Models for Adversarial Robustness



- No external data





Dominate for two years!

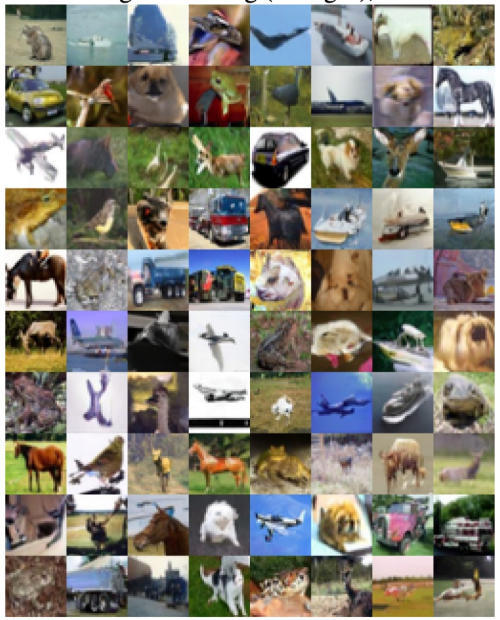
ROBUSTBENCH
A standardized benchmark for adversarial robustness

[Rebuffi et al., NeurIPS 2021; Gowal et al., NeurIPS 2021]

Does Lower FID lead to Better Downstream Performance?

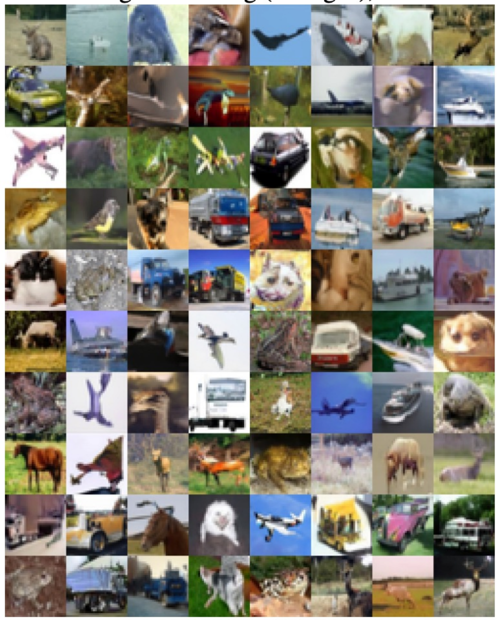
	CIFAR-10 [29] at 32×32				FFHQ [27] 64×64		AFHQv2 [7] 64×64	
	Conditional		Unconditional		Unconditional		Unconditional	
Training configuration	VP	VE	VP	VE	VP	VE	VP	VE
A Baseline [49] (* pre-trained)	2.48	3.11	3.01*	3.77*	3.39	25.95	2.58	18.52
B + Adjust hyperparameters	2.18	2.48	2.51	2.94	3.13	22.53	2.43	23.12
C + Redistribute capacity	2.08	2.52	2.31	2.83	2.78	41.62	2.54	15.04
D + Our preconditioning	2.09	2.64	2.29	3.10	2.94	3.39	2.79	3.81
E + Our loss function	1.88	1.86	2.05	1.99	2.60	2.81	2.29	2.28
F + Non-leaky augmentation	**1.79**	**1.79**	**1.97**	**1.98**	**2.39**	**2.53**	**1.96**	**2.16**
NFE	35	35	35	35	79	79	79	79

Original training (config A), VP



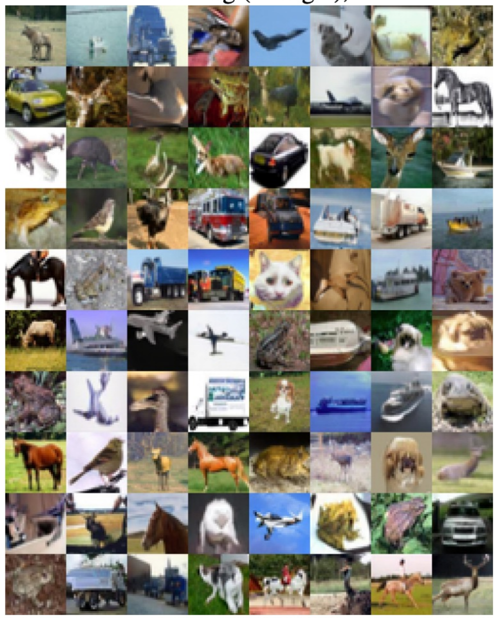
FID 3.01 NFE 35

Original training (config A), VE



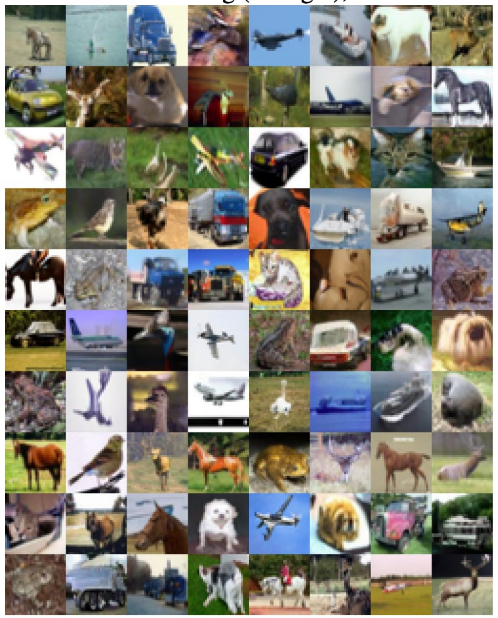
FID 3.77 NFE 35

Our training (config F), VP



FID **1.97** NFE 35

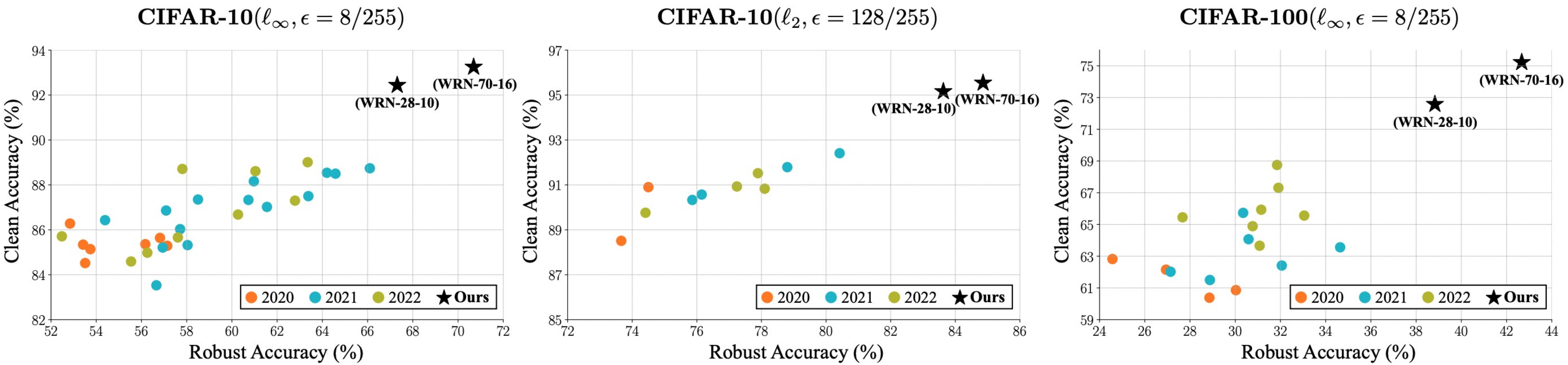
Our training (config F), VE



FID 1.98 NFE 35

[Karras et al., NeurIPS 2022]

Yes! Better Diffusion Models are Indeed Better



- New state-of-the-art!

ROBUSTBENCH

A standardized benchmark for adversarial robustness

Yes! Better Diffusion Models are Indeed Better

Table 1. A brief summary comparison of test accuracy (%) between our models and existing Rank #1 models, *with* (✓) and *without* (✗) external datasets, as listed in RobustBench (Croce et al., 2021).

Dataset	Method	External	Clean	AA
CIFAR-10 (ℓ_∞, $\epsilon = 8/255$)	Rank #1	✗	88.74	66.11
		✓	92.23	66.58
	Ours	✗	**93.25**	**70.69**
CIFAR-10 (ℓ_2, $\epsilon = 128/255$)	Rank #1	✗	92.41	80.42
		✓	**95.74**	82.32
	Ours	✗	95.54	**84.86**
CIFAR-100 (ℓ_∞, $\epsilon = 8/255$)	Rank #1	✗	63.56	34.64
		✓	69.15	36.88
	Ours	✗	**75.22**	**42.67**

- Even beat previous SOTA that using external datasets
- No extra training time (only extra cost for generating data)

Yes! Better Diffusion Models are Indeed Better

- Alleviate overfitting in adversarial training

Generated	Best epoch	Clean			PGD-40			AA		
		Best	Last	Diff	Best	Last	Diff	Best	Last	Diff
✗	91	84.55	82.59	−1.96	55.66	46.47	−9.19	54.37	45.29	−9.08
50K	171	86.15	85.47	−0.68	56.96	50.02	−6.94	55.71	48.85	−6.86
100K	274	88.20	87.47	−0.73	59.85	54.95	−4.90	58.85	53.42	−5.43
200K	365	89.71	89.48	−0.23	61.69	60.32	−1.37	59.91	59.11	−0.80
500K	395	90.76	90.58	−0.18	63.85	63.69	−0.16	62.76	62.77	+0.01
1M	394	91.13	90.89	−0.24	64.67	64.50	−0.17	63.35	63.50	+0.15
5M	395	91.15	90.93	−0.22	64.88	64.88	0	64.05	64.05	0
10M	396	**91.25**	**91.18**	−0.07	65.03	64.96	−0.07	64.19	64.28	+0.09
20M	399	91.17	91.07	−0.10	65.21	65.13	−0.08	64.27	64.16	−0.11
50M	395	91.24	91.15	−0.09	**65.35**	**65.23**	−0.12	**64.53**	**64.51**	−0.02

Yes! Better Diffusion Models are Indeed Better

	Step	FID ↓	Clean	PGD-40	AA
Class-cond.	5	35.54	88.92	57.33	57.78
	10	2.477	90.96	66.21	62.81
	15	1.848	91.05	64.56	63.24
	20	**1.824**	**91.12**	**64.61**	**63.35**
	25	1.843	91.07	64.59	63.31
	30	1.861	91.10	64.51	63.25
	35	1.874	91.01	64.55	63.13
	40	1.883	91.03	64.44	63.03
Uncond.	5	37.78	88.00	56.92	57.19
	10	2.637	89.40	62.88	61.92
	15	1.998	89.36	63.47	62.31
	20	**1.963**	**89.76**	**63.66**	**62.45**
	25	1.977	89.61	63.63	62.40
	30	1.992	89.52	63.51	62.33
	35	2.003	89.39	63.56	62.37
	40	2.011	89.44	63.30	62.24

- Conditional > Unconditional
- Lower FID is better

Yes! Better Diffusion Models are Indeed Better

Table 6. Test accuracy (%) with different **augmentation methods** under the (ℓ_∞, $\epsilon = 8/255$) threat model on CIFAR-10, using WRN-28-10 and 1M EDM generated data.

Methed	Clean	PGD-40	AA
Common	91.12	**64.61**	**63.35**
Cutout	**91.25**	64.54	63.30
CutMix	91.08	64.34	62.81
AutoAugment	91.23	64.07	62.86
RandAugment	91.14	64.39	63.12
IDBH	91.08	64.41	63.24

- Data augmentation seems ineffective

Robustness and Accuracy Could Be Reconcilable by (Proper) Definition

Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, Shuicheng Yan

ICML 2022

Better Diffusion Models Further Improve Adversarial Training

Zekai Wang*, Tianyu Pang*, Chao Du, Min Lin, Weiwei Liu, Shuicheng Yan

ICML 2023

On Evaluating Adversarial Robustness of Large Vision-Language Models

Yunqing Zhao*, Tianyu Pang*, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, Min Lin

NeurIPS 2023

Large vision-language models (Large VLMs)

Backgrounds: Emerging Large VLMs are powerful in response generation with visual input

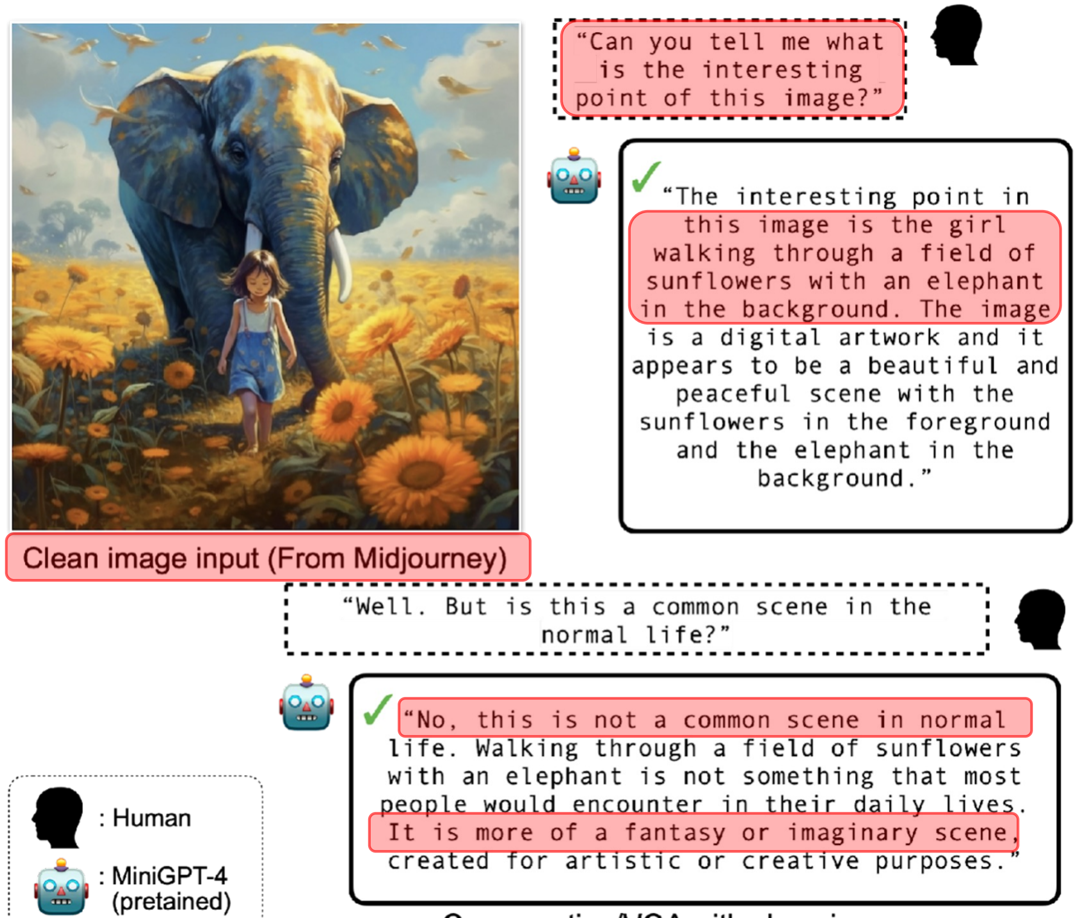
ChatGPT 11. 2022	**GPT4** 03. 2023
A Chatbot that provides a detailed response	A more advanced system that producing safer and more useful responses.

[Closed-Sourced]

BLIP-2 01. 2023	**LLaVA** 04. 2023	**Mini-GPT4** 04. 2023
Conditional text generation given an image and an optional text prompt.	General-purpose visual and language understanding	General-purpose visual and language understanding

[Open-Sourced]

Example: MiniGPT-4



Conversation/VQA with clean image x_{cle}

Everything Looks Great:

1. Good Visual and language understanding
2. Reasonable and detailed response
3. Running on a **single** GPU
4. Wide application scenarios

…

Zhu et al., Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv 2023

Large vision-language models (Large VLMs)

Questions:

- **When Large VLMs are deployed in practice:**
  Responsible answer generation in companies, Gov., or commercial usage

- **Consequently, we ask:**
  What if the generated responses are wrong? It may raise serious concerns

We research the "worst case" of these large VLMs:

Can we let these VLMs generate "targeted response"?

Matching image-text features (MF-it)

An intuitive method:

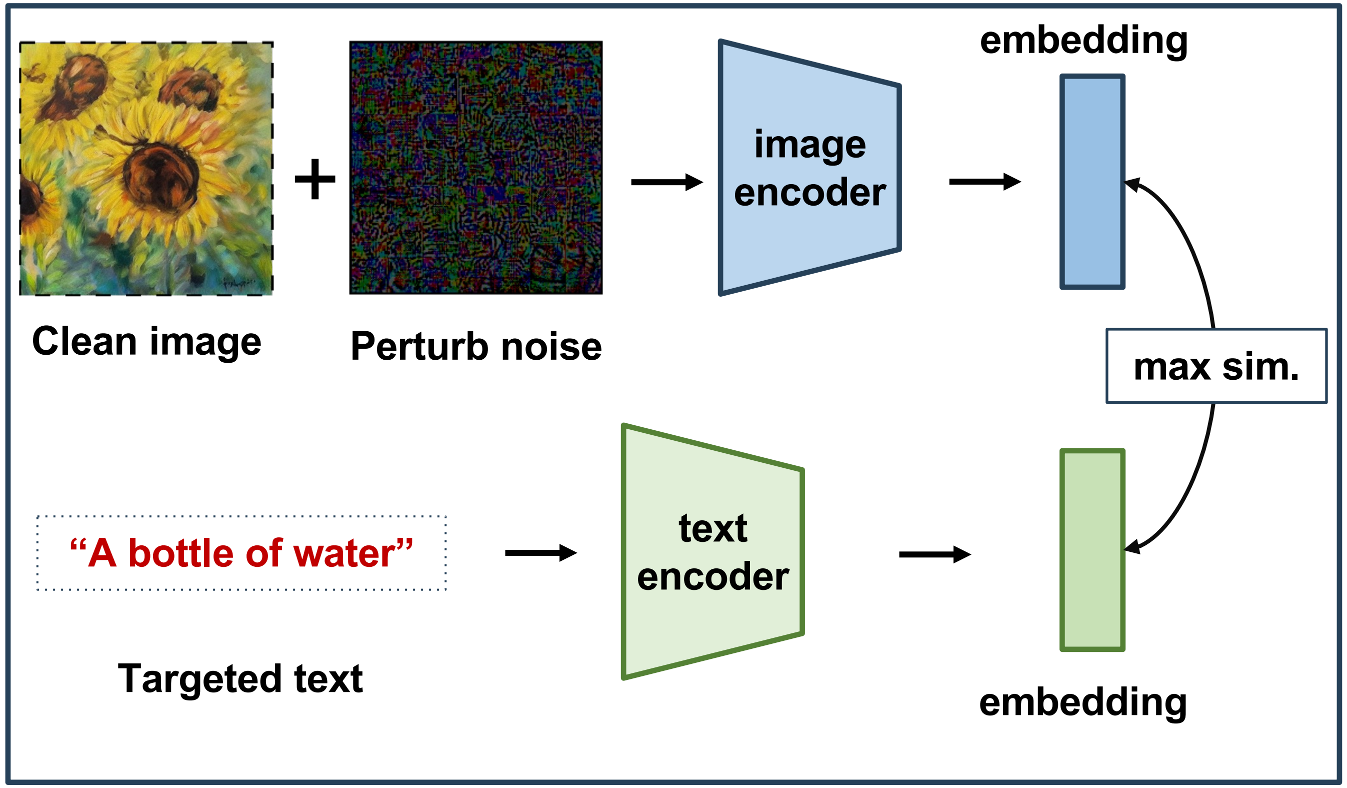
$$\underset{\|\boldsymbol{x}_{\text{cle}}-\boldsymbol{x}_{\text{adv}}\|_p\le\epsilon}{\arg\max}\; f_\phi(\boldsymbol{x}_{\text{adv}})^\top g_\psi(\boldsymbol{c}_{\text{tar}})$$

f_ϕ : image encoder

g_ψ : text encoder

Surrogate models

■ white-box



Matching the features via an **image encoder** and a **text encoder**

Matching image-image features (MF-ii)

Match target image features via an **image encoder** and a **text-to-image model**:

$$\arg\max_{\|\boldsymbol{x}_{\text{cle}}-\boldsymbol{x}_{\text{adv}}\|_p\le\epsilon} f_\phi(\boldsymbol{x}_{\text{adv}})^\top f_\phi(h_\xi(\boldsymbol{c}_{\text{tar}}))$$

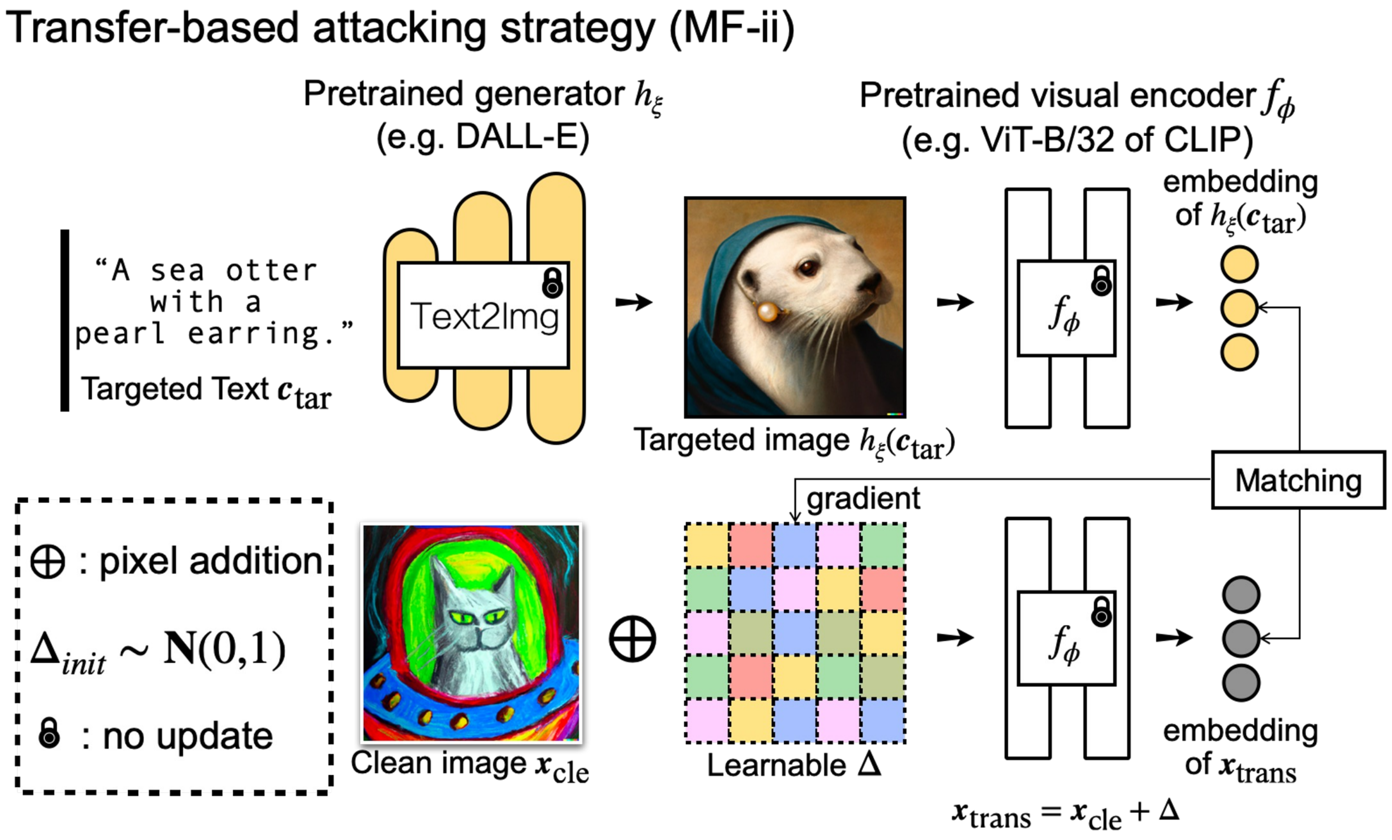
f_ϕ : image encoder

h_ξ : text2img model

Surrogate models

■ white-box

■ black-box



Matching text-text features (MF-tt)

Matching the features via a **text encoder**:

$$\underset{\|\boldsymbol{x}_{\text{cle}}-\boldsymbol{x}_{\text{adv}}\|_p\leq\epsilon}{\arg\max}\ g_\psi(p_\theta(\boldsymbol{x}_{\text{adv}};\boldsymbol{c}_{\text{in}}))^\top g_\psi(\boldsymbol{c}_{\text{tar}})$$

g_ψ : text encoder

Surrogate model

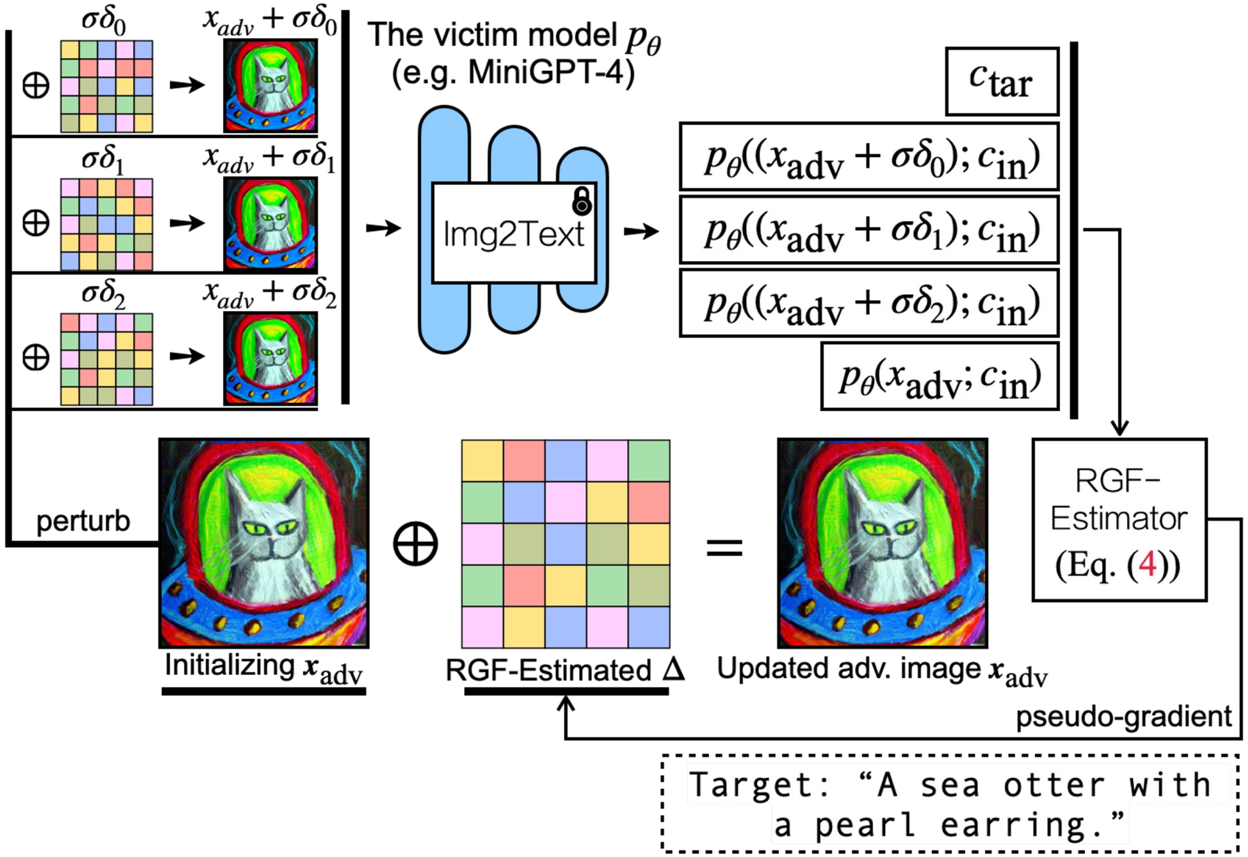
p_θ :image-2-text model

Target model

■ white-box

■ black-box

Query-based attacking strategy (MF-tt)



Matching text-text features (MF-tt)

Matching the features via a **text encoder** (**black-box setting**):

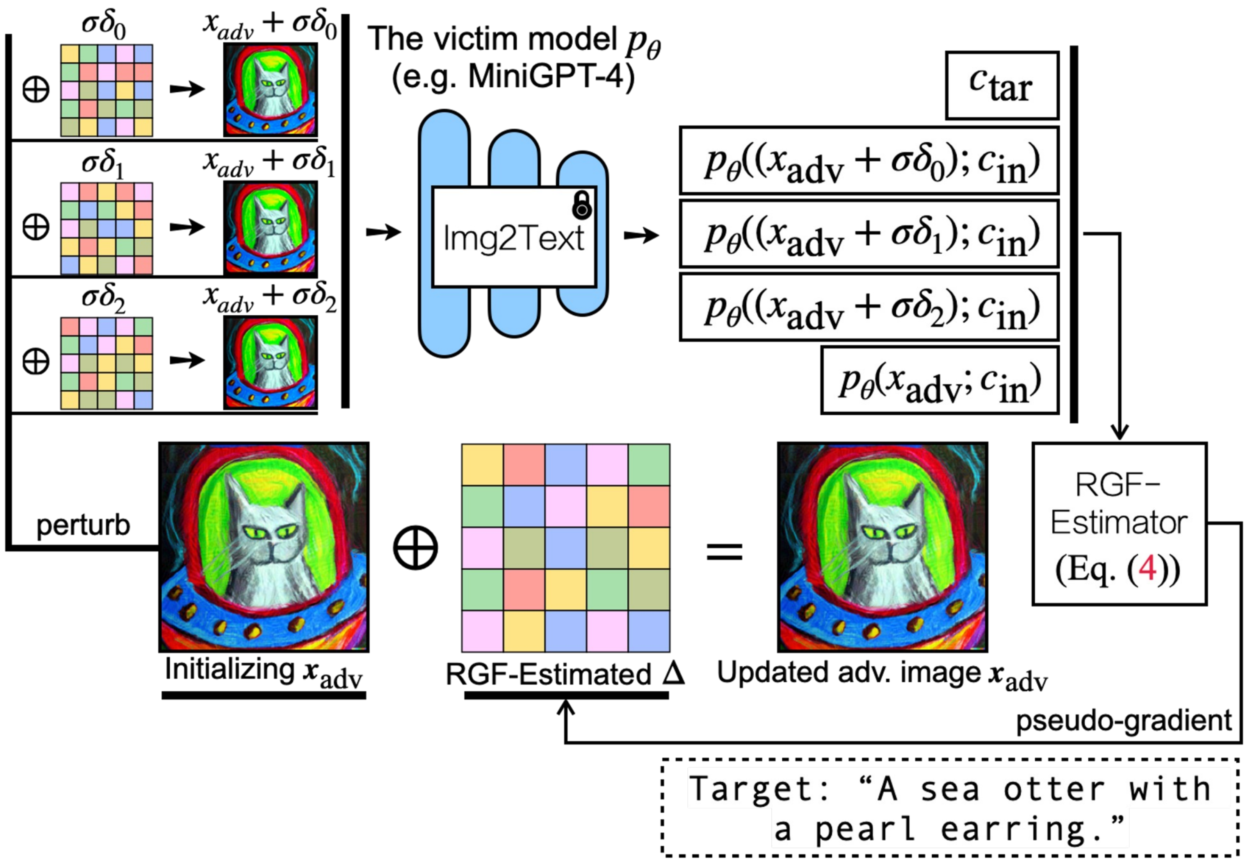
$$\underset{\|\boldsymbol{x}_{\text{cle}}-\boldsymbol{x}_{\text{adv}}\|_p\leq\epsilon}{\arg\max}\; g_\psi(p_\theta(\boldsymbol{x}_{\text{adv}};\boldsymbol{c}_{\text{in}}))^\top g_\psi(\boldsymbol{c}_{\text{tar}})$$

Gradient estimation: (Eq. (4))

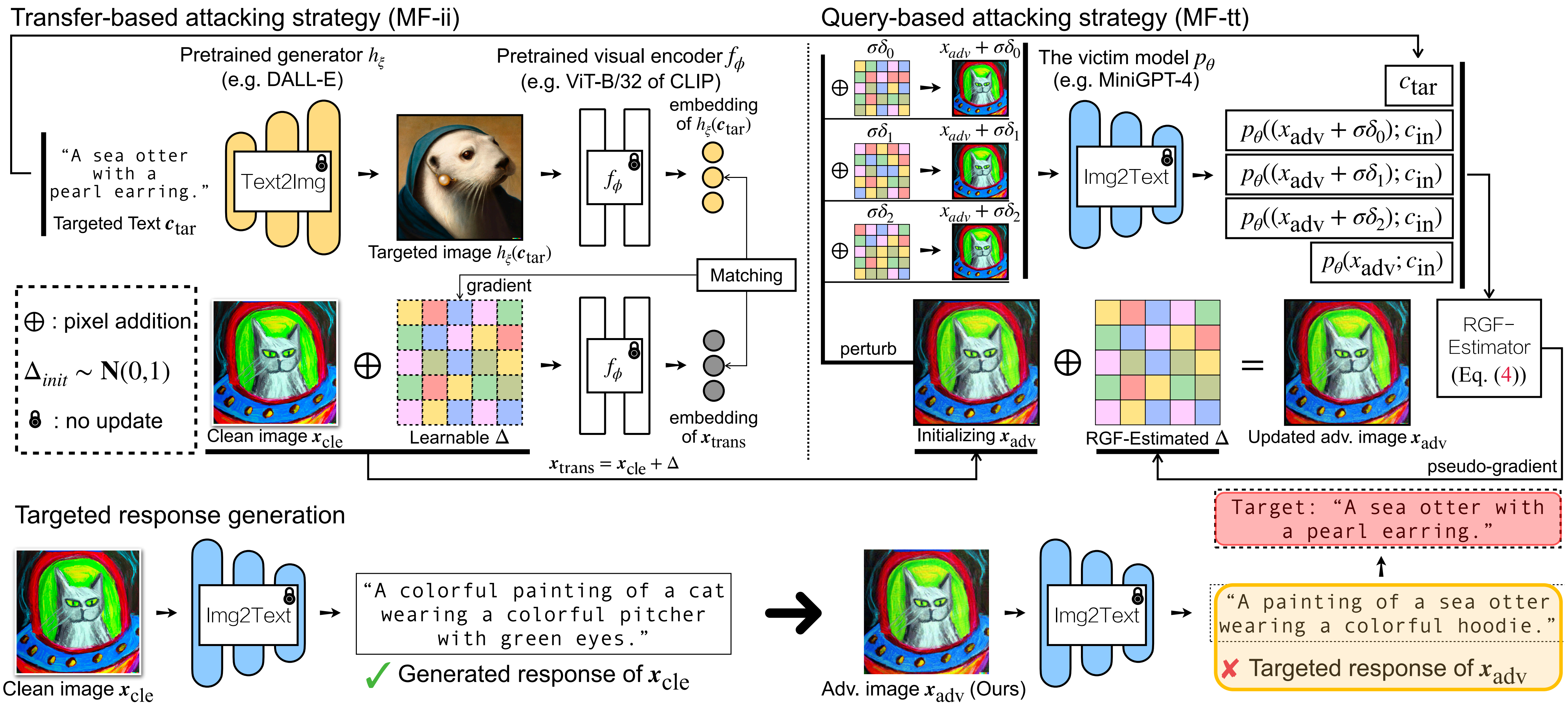
$$\nabla_{\boldsymbol{x}_{\text{adv}}} g_\psi(p_\theta(\boldsymbol{x}_{\text{adv}};\boldsymbol{c}_{\text{in}}))^\top g_\psi(\boldsymbol{c}_{\text{tar}}) \approx \frac{1}{N\sigma}\sum_{n=1}^{N}\left[g_\psi(p_\theta(\boldsymbol{x}_{\text{adv}}+\sigma\boldsymbol{\delta}_n;\boldsymbol{c}_{\text{in}}))^\top g_\psi(\boldsymbol{c}_{\text{tar}}) - g_\psi(p_\theta(\boldsymbol{x}_{\text{adv}};\boldsymbol{c}_{\text{in}}))^\top g_\psi(\boldsymbol{c}_{\text{tar}})\right]\cdot\boldsymbol{\delta}_n$$

RGF-Estimator

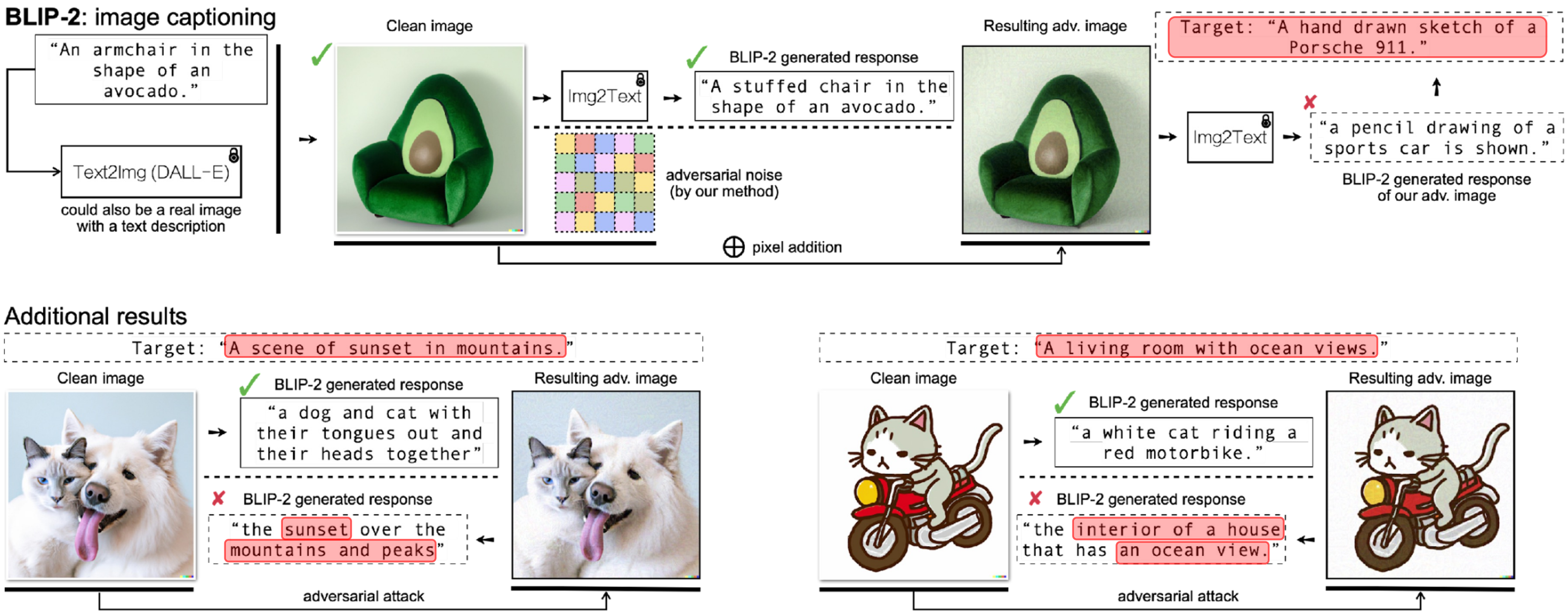
Query-based attacking strategy (MF-tt)



MF-ii + MF-tt (Ours)

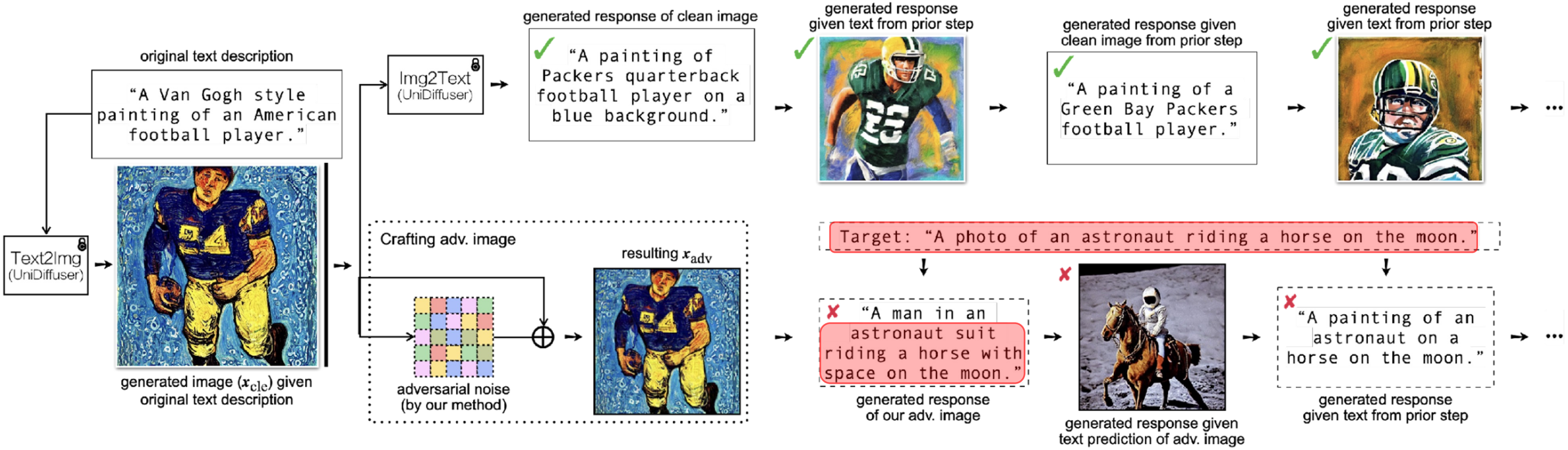


Evading BLIP-2



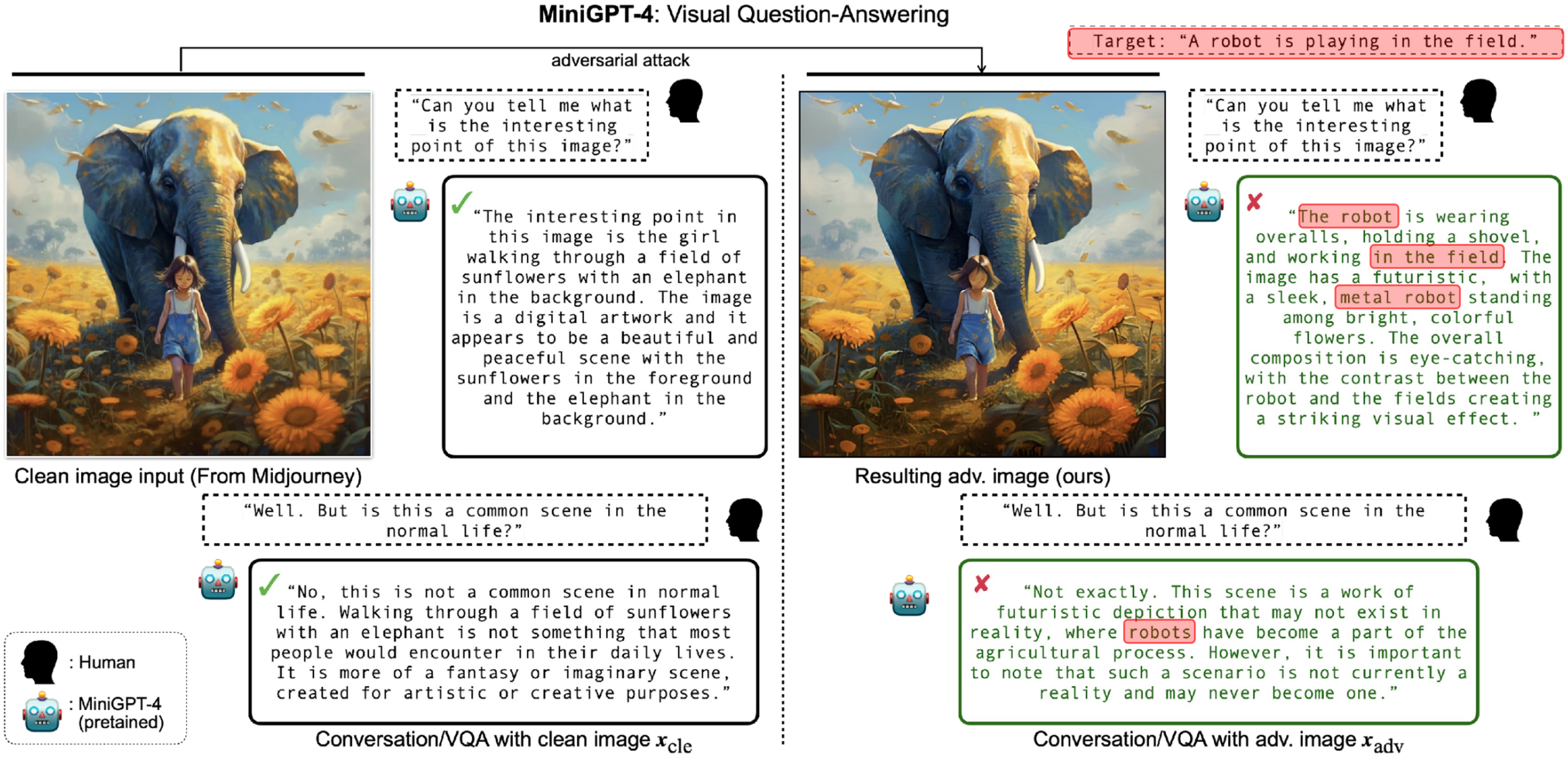
Li et al., Blip-2: Bootstrapping languageimage pre-training with frozen image encoders and large language models. arXiv 2023.

Evading UniDiffuser



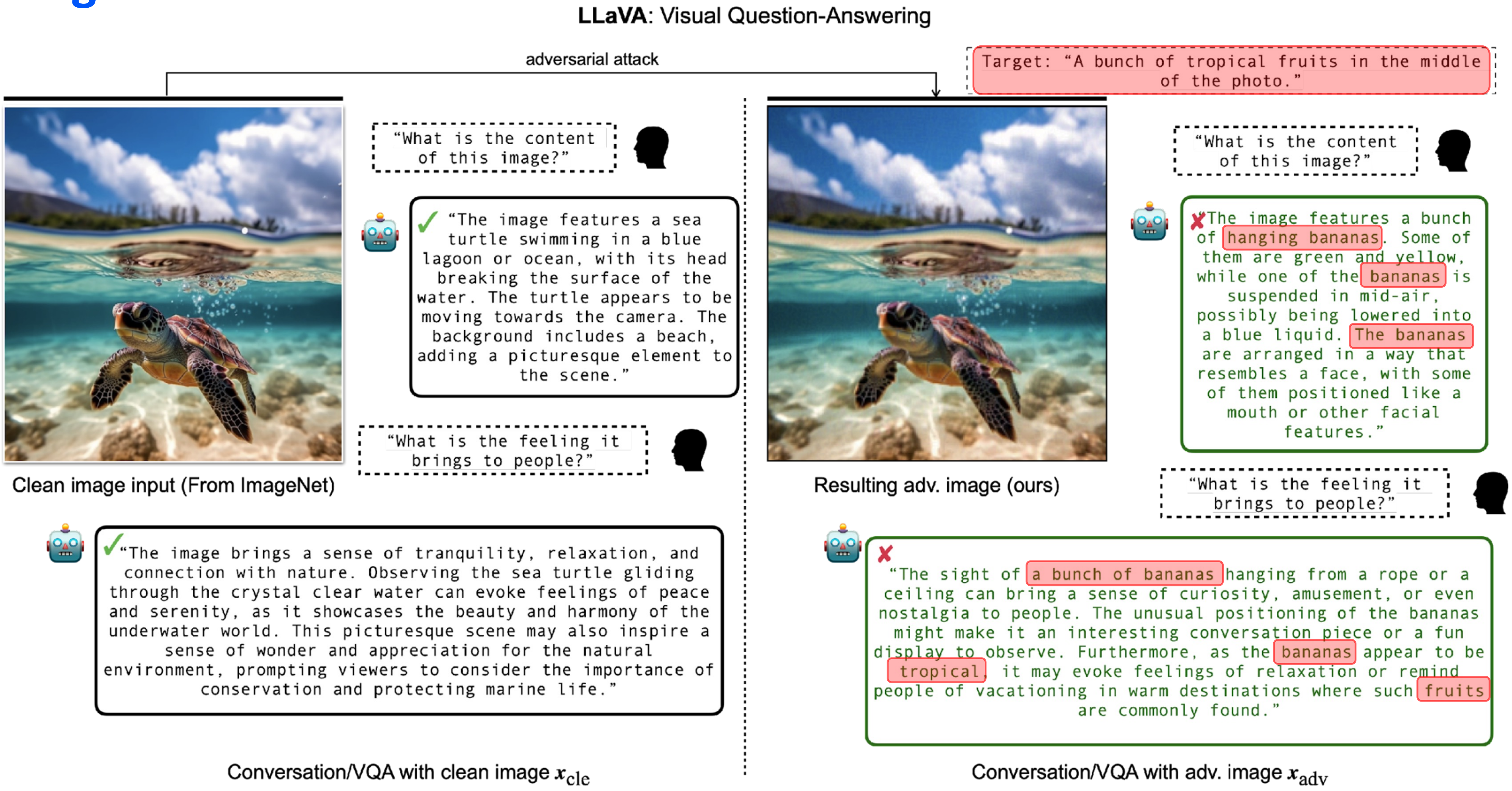
Bao et al., One Transformer Fits All Distributions in Multi-Modal Diffusion. arXiv 2023.

Evading MiniGPT-4



Zhu et al., Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv 2023.

Evading LLaVA



Liu et al., Visual instruction tuning. arXiv 2023.

Quantitative evaluation
(CLIP score between text and image features)

Performance: Matching image-text features (MF-it)

White-box attacks against surrogate models

Model	Clean image		Adversarial image		Time to obtain a single $\boldsymbol{x}_{\text{adv}}$	
	$\boldsymbol{x}_{\text{cle}}$	$h_{\xi}(\boldsymbol{c}_{\text{tar}})$	MF-ii	MF-it	MF-ii	MF-it
CLIP (RN50) [62]	0.094	0.261	0.239	**0.576**	0.543	0.532
CLIP (ViT-B/32) [62]	0.142	0.313	0.302	**0.570**	0.592	0.588
BLIP (ViT) [39]	0.138	0.286	0.277	**0.679**	0.641	0.634
BLIP-2 (ViT) [40]	0.037	0.302	0.294	**0.502**	0.855	0.852
ALBEF (ViT) [38]	0.063	0.098	0.091	**0.451**	0.750	0.749

Good performance in white-box setting

Quantitative evaluation (CLIP text score ↑)

Black-box attacks against victim models.

MF-it is not that transferrable in black-box setting;

VLM model	Attacking method	Text encoder (pretrained) for evaluation						Other info.	
		RN50	RN101	ViT-B/16	ViT-B/32	ViT-L/14	Ensemble	# Param.	Res.
BLIP [41]	Clean image	0.472	0.456	0.479	0.499	0.344	0.450	224M	384
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.766	0.753	0.774	0.786	0.696	0.755		
	MF-ii + MF-tt	**0.855**	**0.841**	**0.861**	**0.868**	**0.803**	**0.846**		
UniDiffuser [5]	Clean image	0.417	0.415	0.429	0.446	0.305	0.402	1.4B	224
	MF-it	0.655	0.639	0.678	0.698	0.611	0.656		
	MF-ii	0.709	0.695	0.721	0.733	0.637	0.700		
	MF-ii + MF-tt	**0.754**	**0.736**	**0.761**	**0.777**	**0.689**	**0.743**		
Img2Prompt [30]	Clean image	0.487	0.464	0.493	0.515	0.350	0.461	1.7B	384
	MF-it	0.499	0.472	0.501	0.525	0.355	0.470		
	MF-ii	0.502	0.479	0.505	0.529	0.366	0.476		
	MF-ii + MF-tt	**0.803**	**0.783**	**0.809**	**0.828**	**0.733**	**0.791**		
BLIP-2 [42]	Clean image	0.473	0.454	0.483	0.503	0.349	0.452	3.7B	224
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.562	0.541	0.571	0.592	0.449	0.543		
	MF-ii + MF-tt	**0.656**	**0.633**	**0.665**	**0.681**	**0.555**	**0.638**		
LLaVA [46]	Clean image	0.383	0.436	0.402	0.437	0.281	0.388	13.3B	224
	MF-it	0.389	0.441	0.417	0.452	0.288	0.397		
	MF-ii	0.396	0.440	0.421	0.450	0.292	0.400		
	MF-ii + MF-tt	**0.548**	**0.559**	**0.563**	**0.590**	**0.448**	**0.542**		
MiniGPT-4 [109]	Clean image	0.422	0.431	0.436	0.470	0.326	0.417	14.1B	224
	MF-it	0.472	0.450	0.461	0.484	0.349	0.443		
	MF-ii	0.525	0.541	0.542	0.572	0.430	0.522		
	MF-ii + MF-tt	**0.633**	**0.611**	**0.631**	**0.668**	**0.528**	**0.614**		

Quantitative evaluation (CLIP text score ↑)

Black-box attacks against victim models.

MF-it is not that transferrable in black-box setting;

MF-ii is better, but the performance is limited by the targeted images;

VLM model	Attacking method	Text encoder (pretrained) for evaluation						Other info.	
		RN50	RN101	ViT-B/16	ViT-B/32	ViT-L/14	Ensemble	# Param.	Res.
BLIP [41]	Clean image	0.472	0.456	0.479	0.499	0.344	0.450	224M	384
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.766	0.753	0.774	0.786	0.696	0.755		
	MF-ii + MF-tt	**0.855**	**0.841**	**0.861**	**0.868**	**0.803**	**0.846**		
UniDiffuser [5]	Clean image	0.417	0.415	0.429	0.446	0.305	0.402	1.4B	224
	MF-it	0.655	0.639	0.678	0.698	0.611	0.656		
	MF-ii	0.709	0.695	0.721	0.733	0.637	0.700		
	MF-ii + MF-tt	**0.754**	**0.736**	**0.761**	**0.777**	**0.689**	**0.743**		
Img2Prompt [30]	Clean image	0.487	0.464	0.493	0.515	0.350	0.461	1.7B	384
	MF-it	0.499	0.472	0.501	0.525	0.355	0.470		
	MF-ii	0.502	0.479	0.505	0.529	0.366	0.476		
	MF-ii + MF-tt	**0.803**	**0.783**	**0.809**	**0.828**	**0.733**	**0.791**		
BLIP-2 [42]	Clean image	0.473	0.454	0.483	0.503	0.349	0.452	3.7B	224
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.562	0.541	0.571	0.592	0.449	0.543		
	MF-ii + MF-tt	**0.656**	**0.633**	**0.665**	**0.681**	**0.555**	**0.638**		
LLaVA [46]	Clean image	0.383	0.436	0.402	0.437	0.281	0.388	13.3B	224
	MF-it	0.389	0.441	0.417	0.452	0.288	0.397		
	MF-ii	0.396	0.440	0.421	0.450	0.292	0.400		
	MF-ii + MF-tt	**0.548**	**0.559**	**0.563**	**0.590**	**0.448**	**0.542**		
MiniGPT-4 [109]	Clean image	0.422	0.431	0.436	0.470	0.326	0.417	14.1B	224
	MF-it	0.472	0.450	0.461	0.484	0.349	0.443		
	MF-ii	0.525	0.541	0.542	0.572	0.430	0.522		
	MF-ii + MF-tt	**0.633**	**0.611**	**0.631**	**0.668**	**0.528**	**0.614**		

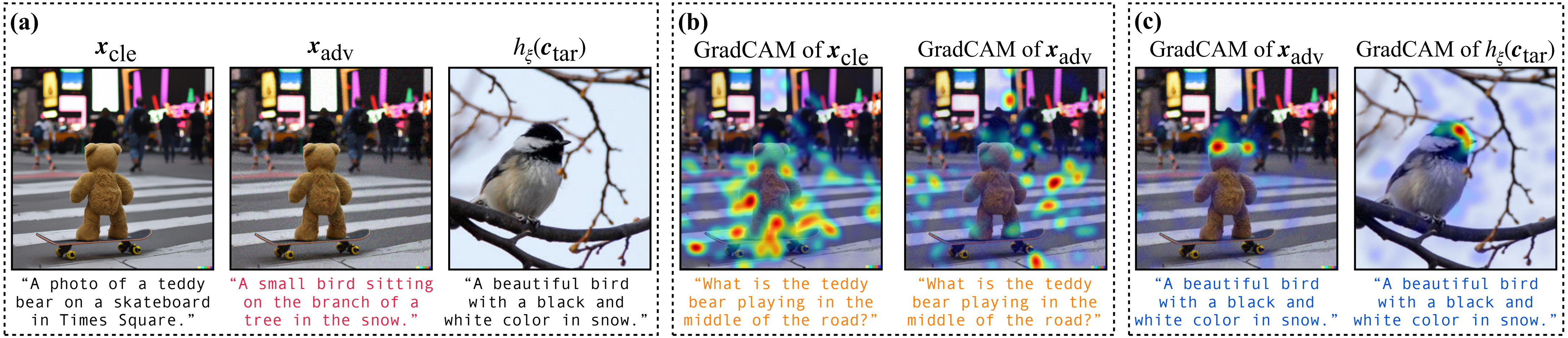
Quantitative evaluation (CLIP text score ↑)

Black-box attacks against victim models.

MF-it is not that transferrable in black-box setting;
MF-ii is better, but the performance is limited by the targeted images;
MF-ii + MF-tt achieves better performance

VLM model	Attacking method	Text encoder (pretrained) for evaluation						Other info.	
		RN50	RN101	ViT-B/16	ViT-B/32	ViT-L/14	Ensemble	# Param.	Res.
BLIP [41]	Clean image	0.472	0.456	0.479	0.499	0.344	0.450	224M	384
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.766	0.753	0.774	0.786	0.696	0.755		
	MF-ii + MF-tt	**0.855**	**0.841**	**0.861**	**0.868**	**0.803**	**0.846**		
UniDiffuser [5]	Clean image	0.417	0.415	0.429	0.446	0.305	0.402	1.4B	224
	MF-it	0.655	0.639	0.678	0.698	0.611	0.656		
	MF-ii	0.709	0.695	0.721	0.733	0.637	0.700		
	MF-ii + MF-tt	**0.754**	**0.736**	**0.761**	**0.777**	**0.689**	**0.743**		
Img2Prompt [30]	Clean image	0.487	0.464	0.493	0.515	0.350	0.461	1.7B	384
	MF-it	0.499	0.472	0.501	0.525	0.355	0.470		
	MF-ii	0.502	0.479	0.505	0.529	0.366	0.476		
	MF-ii + MF-tt	**0.803**	**0.783**	**0.809**	**0.828**	**0.733**	**0.791**		
BLIP-2 [42]	Clean image	0.473	0.454	0.483	0.503	0.349	0.452	3.7B	224
	MF-it	0.492	0.474	0.520	0.546	0.384	0.483		
	MF-ii	0.562	0.541	0.571	0.592	0.449	0.543		
	MF-ii + MF-tt	**0.656**	**0.633**	**0.665**	**0.681**	**0.555**	**0.638**		
LLaVA [46]	Clean image	0.383	0.436	0.402	0.437	0.281	0.388	13.3B	224
	MF-it	0.389	0.441	0.417	0.452	0.288	0.397		
	MF-ii	0.396	0.440	0.421	0.450	0.292	0.400		
	MF-ii + MF-tt	**0.548**	**0.559**	**0.563**	**0.590**	**0.448**	**0.542**		
MiniGPT-4 [109]	Clean image	0.422	0.431	0.436	0.470	0.326	0.417	14.1B	224
	MF-it	0.472	0.450	0.461	0.484	0.349	0.443		
	MF-ii	0.525	0.541	0.542	0.572	0.430	0.522		
	MF-ii + MF-tt	**0.633**	**0.611**	**0.631**	**0.668**	**0.528**	**0.614**		

Visual interpretation via GradCAM Analysis

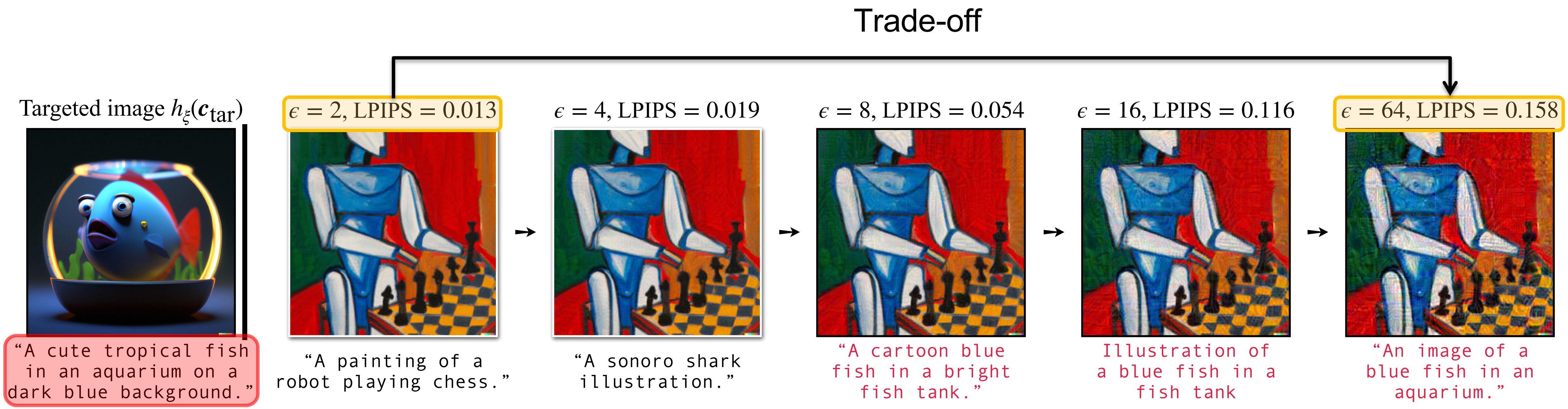


(a): Craft an adv image given a target string and a target image

(b): GradCAM shows good correspondence to the query text over clean images, but not for adv images.

(c): For adv image, we obtain similar GradCAM results as the target image.

Trade-off between image quality and perturbation budget

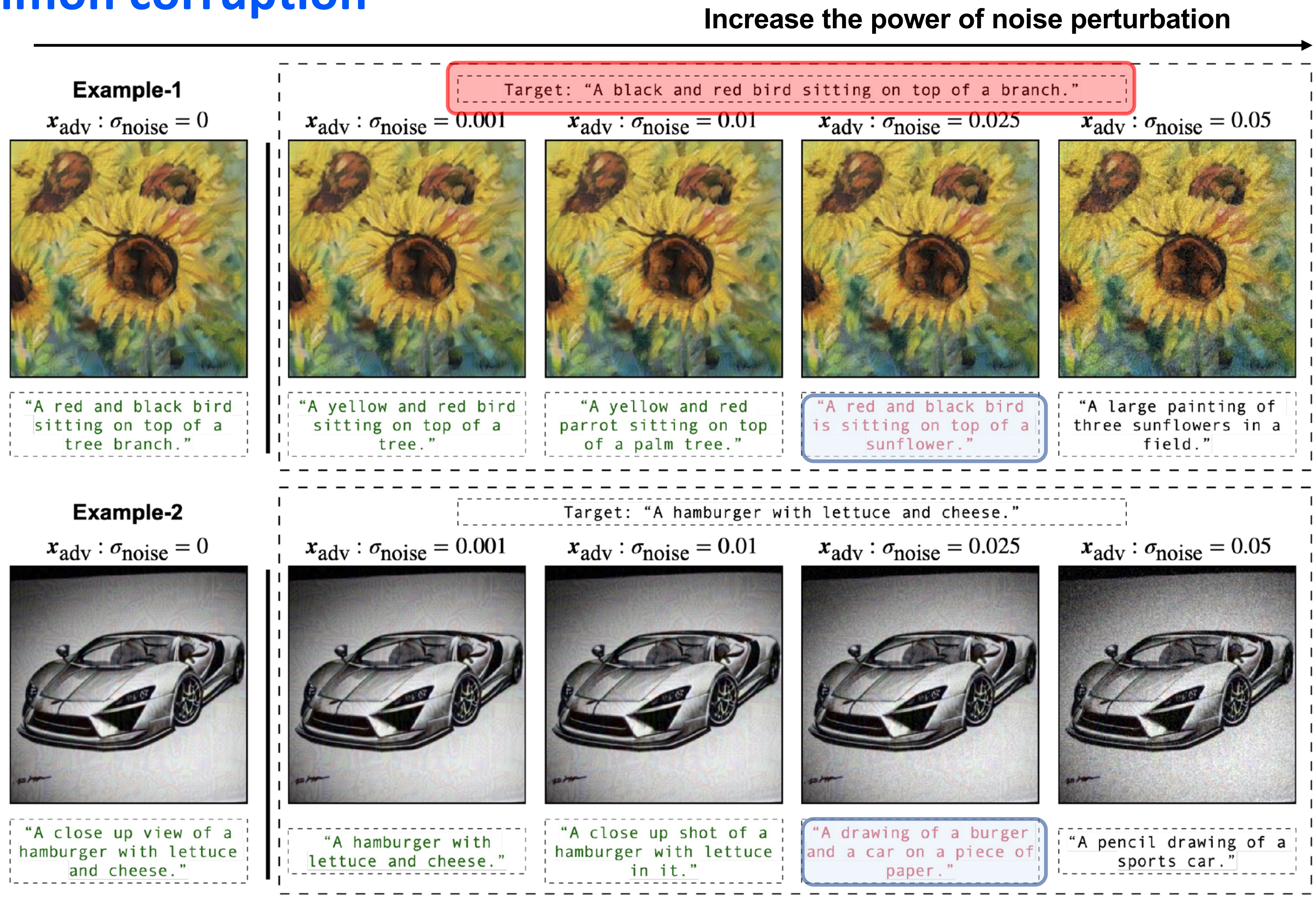


- LPIPS indicates perceptual similarity to the clean image.
- Lower means better quality

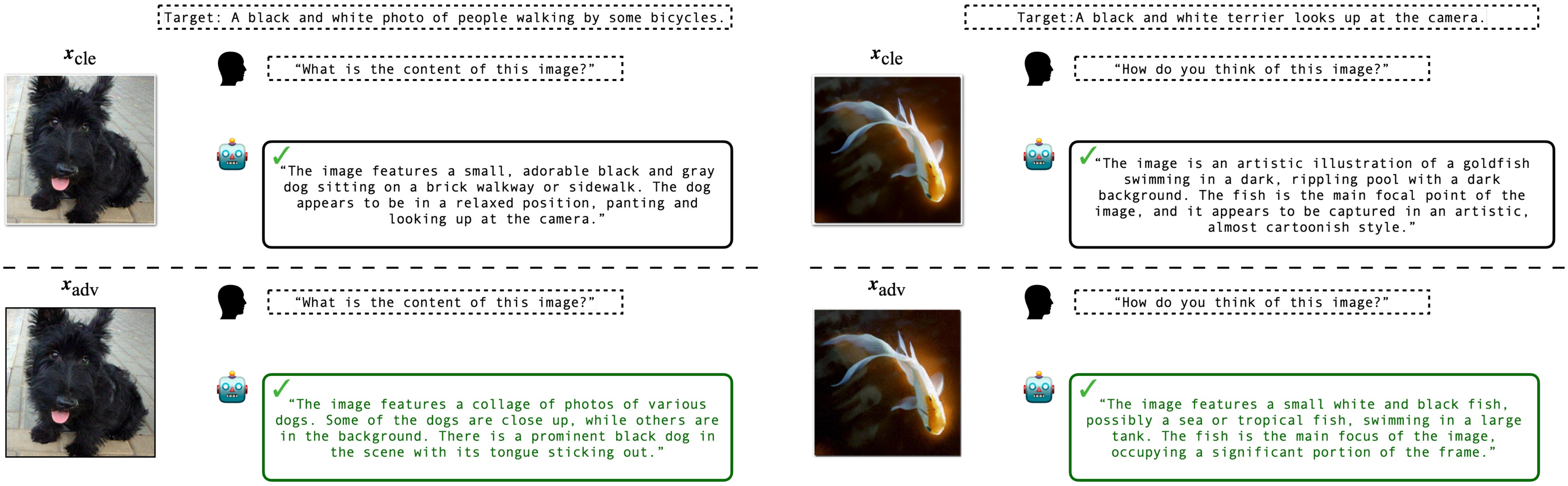
Sensitivity to common corruption

Sensitivity of adversarial examples to Gaussian noises.

Learnt noise perturbation gradually becomes invalid.



Failure cases



Two failure cases, where the correct response is generated over adv images.

Robustness and Accuracy Could Be Reconcilable by (Proper) Definition

Tianyu Pang, Min Lin, Xiao Yang, Jun Zhu, Shuicheng Yan

ICML 2022

Better Diffusion Models Further Improve Adversarial Training

Zekai Wang*, Tianyu Pang*, Chao Du, Min Lin, Weiwei Liu, Shuicheng Yan

ICML 2023

On Evaluating Adversarial Robustness of Large Vision-Language Models

Yunqing Zhao*, Tianyu Pang*, Chao Du, Xiao Yang, Chongxuan Li, Ngai-Man Cheung, Min Lin

NeurIPS 2023

Joint work with

Yunqing Zhao

Zekai Wang

Chao Du

Min Lin

Thanks!

Contact: tianyupang3@gmail.com

Homepage: https://p2333.github.io/

Welcome for collaboration on

Trustworthy AI & **Generative Models**