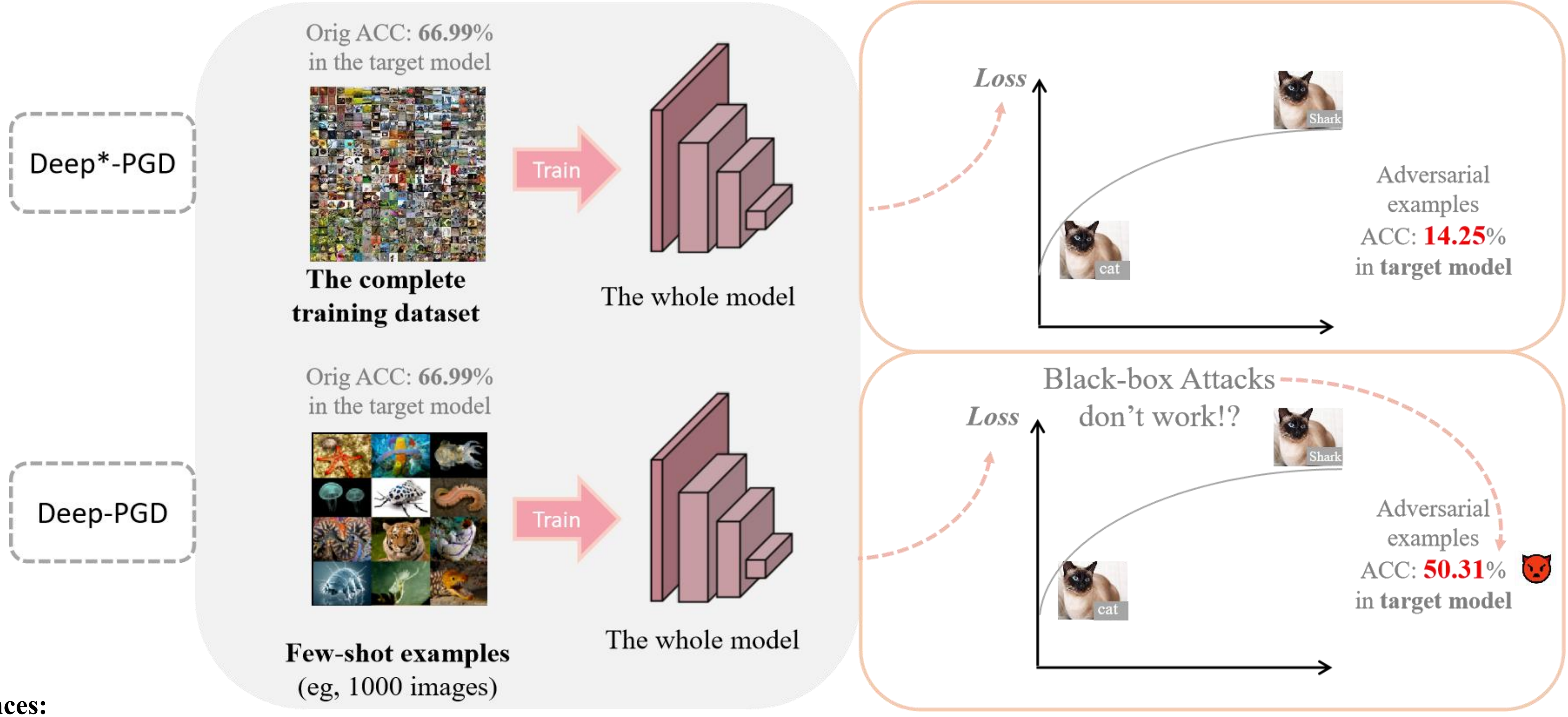
Towards Lightweight Black-Box Attacks Against Deep Neural Networks

Chenghao Sun[1] **Yonggang Zhang**[2] **Wan Chaoqun**[3] **Qizhou Wang**[2]
Ya Li[4] **Tongliang Liu**[5] **Bo Han**[2] **Xinmei Tian**[1*]
[1]University of Science and Technology of China [2]Hong Kong Baptist University
[3]Alibaba Cloud Computing Ltd [4]iFlytek Research
[5]The University of Sydney

Motivation

Background:

It was usually considered **infeasible** to mount effective black-box attacks with **a few test samples** (eg 1000 images) because adversaries can not train a surrogate model well with limited data.**[1]**



References:

[1] Q. Li, Y. Guo, and H. Chen. Practical no-box adversarial attacks against dnns. In NeurIPS , 2020.

Motivation

Background:

It was usually considered **infeasible** to mount effective black-box attacks with **a few test samples** (eg 1000 images) because adversaries can not train a surrogate model well with limited data.**[1]**

Existing research:

(1) Adversarial examples can be generated by perturbing representations at shallow layers of DNNs.**[1]**

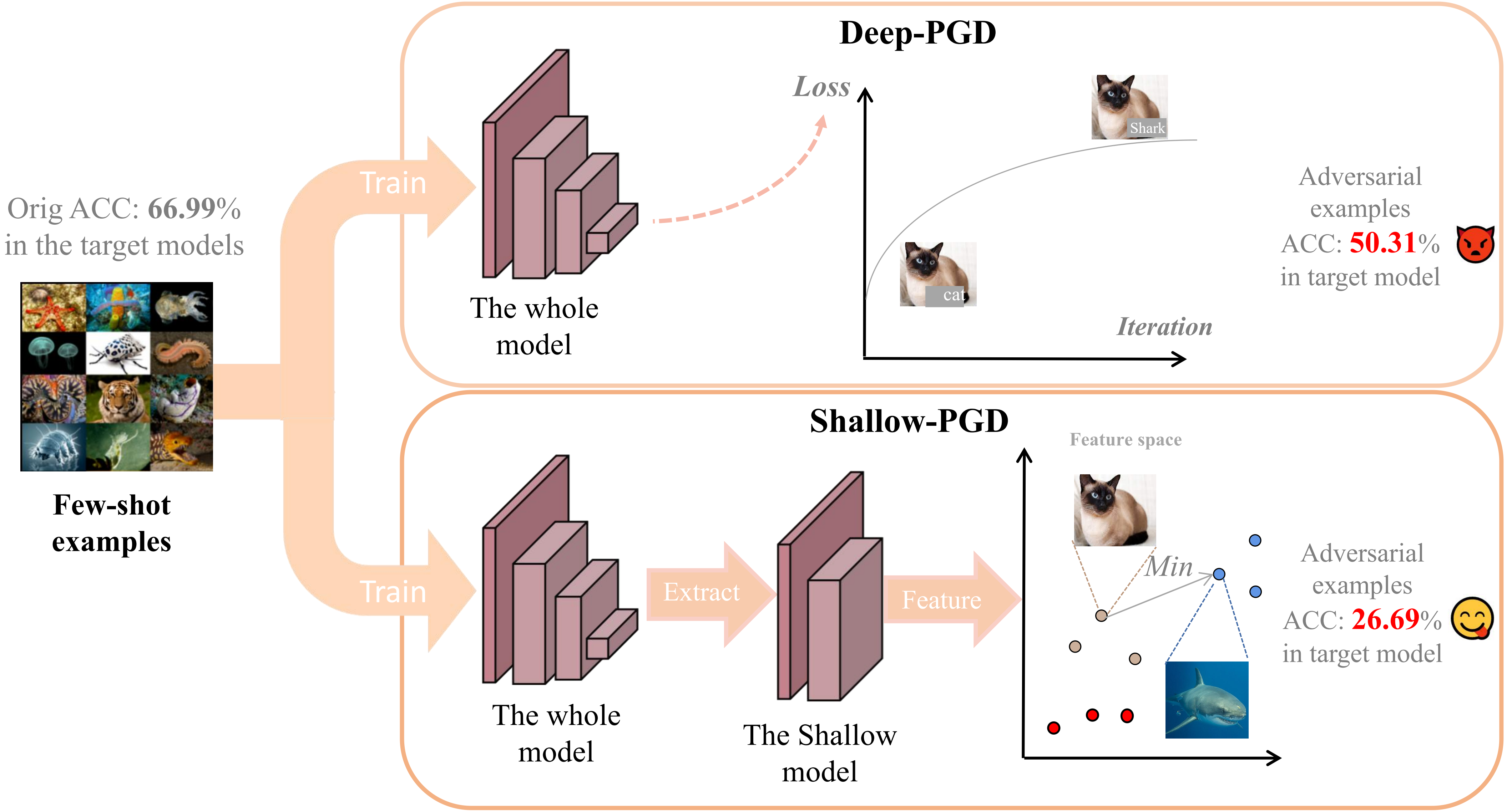
(2) Regarding the representation of shallow layers, there do not exist critical differences between those models learned from a few data and that of the whole training data.**[2]**

References:

[1] Q. Li, Y. Guo, and H. Chen. Practical no-box adversarial attacks against dnns. In NeurIPS , 2020.

[2] N. Inkawhich, W. Wen, H. H. Li, and Y. Chen. Feature space perturbations yield more transferable adversarial examples. In CVPR, 2019.

Lightweight Black-box Attack



Method

Error Transformer：

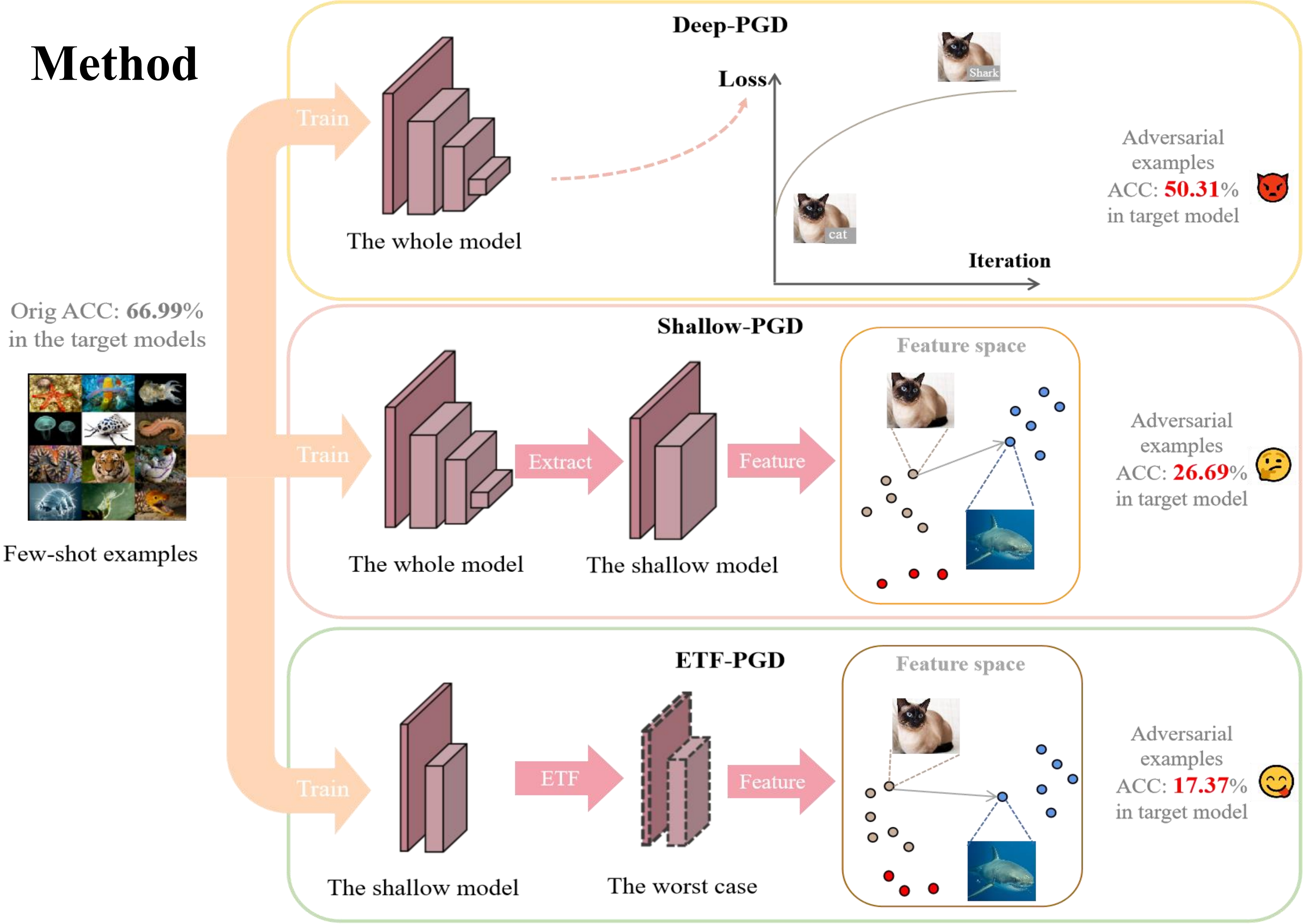
To further improve the attack performance, we propose Error TransFormer (ETF) to alleviate adverse impact caused by approximation error:

$$\varphi(x; \{w^1 + w^1 A\} \cup \{w \backslash w^1\}) = g((w^1 + w^1 A)x; w \backslash w^1) = g(w^1(x + Ax); w \backslash w^1) = \varphi(x + Ax; w)$$

$$x_{adv} = \arg \min_{\|x' - x\|_p \le \epsilon} \max_{\|\Delta_s\|_p \le \tau,\ \|\Delta_g\|_p \le \tau} d(\varphi(x_g + \Delta_g; w), \varphi(x' + \Delta_s; w)),$$

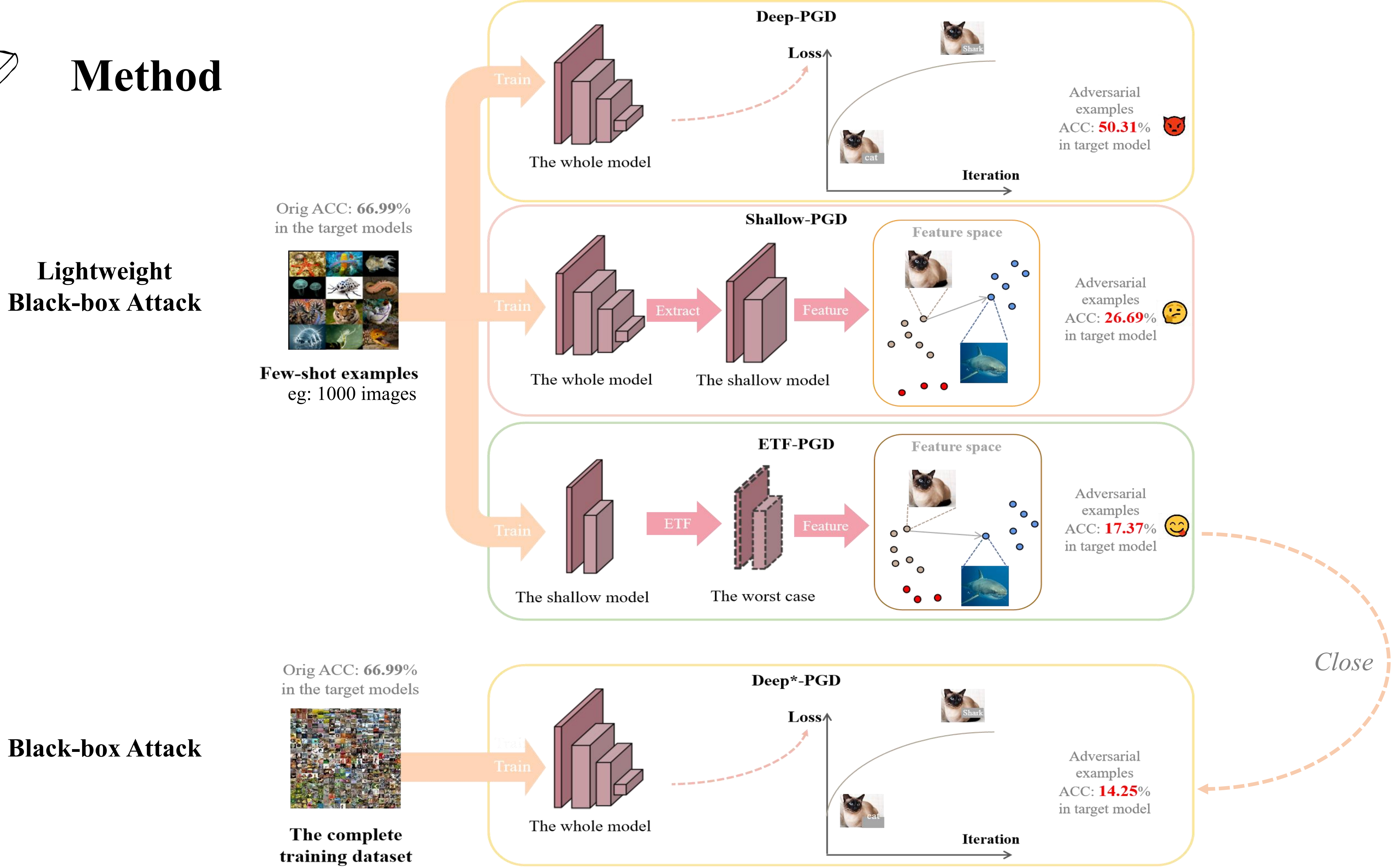
φ:	*The shallow layers of lightweight sorrogate model*
x:	*The images*
x_g:	*The guide images*
$\varphi(x, w)$:	*The feature of shallow layers*
g:	*the function parameterized with $w \backslash w^1$ used for processing the first layer's outputs*
w:	*The parameters of model φ*
w^1:	*The first layer parameters of model φ*
$w \backslash w^1$:	*The parameters of model φ with the first layer*
$w^1 A$:	*The approximation error between the lightweight model and the target model*

Method



Orig ACC: 66.99%
in the target models
Few-shot examples
Train
Train
Train
Deep-PGD
The whole model
Loss
Shark
cat
Iteration
Adversarial
examples
ACC: 50.31%
in target model
Shallow-PGD
The whole model
Extract
The shallow model
Feature
Feature space
Adversarial
examples
ACC: 26.69%
in target model
ETF-PGD
The shallow model
ETF
The worst case
Feature
Feature space
Adversarial
examples
ACC: 17.37%
in target model

Method


Lightweight
Black-box Attack
Orig ACC: 66.99%
in the target models
Few-shot examples
eg: 1000 images
Train
Deep-PGD
The whole model
Loss
Iteration
Shark
cat
Adversarial
examples
ACC: 50.31%
in target model
Shallow-PGD
Extract
Feature
The whole model
The shallow model
Feature space
Adversarial
examples
ACC: 26.69%
in target model
ETF-PGD
ETF
Feature
The shallow model
The worst case
Feature space
Adversarial
examples
ACC: 17.37%
in target model
Close
Black-box Attack
Orig ACC: 66.99%
in the target models
The complete
training dataset
Train
Deep*-PGD
The whole model
Loss
Iteration
Shark
cat
Adversarial
examples
ACC: 14.25%
in target model

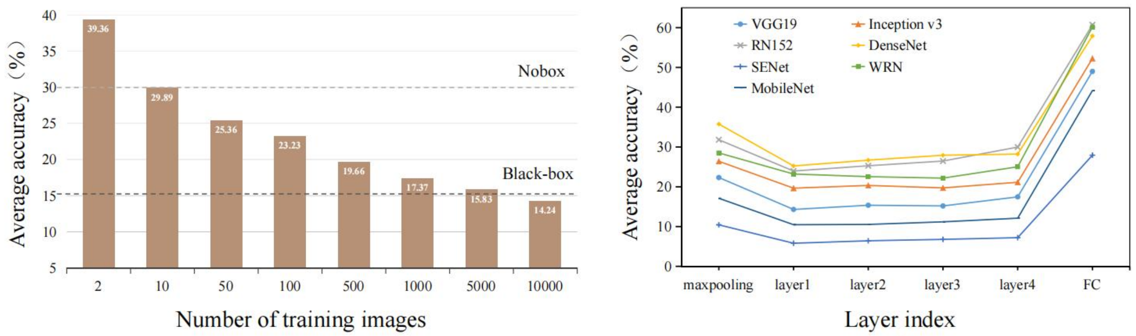
Experiment

Table 1: The accuracy (%) of 7 normally trained target models evaluated on 1000 adversarial examples generated by lightweight black-box attacks or existing black-box attacks, under $\epsilon \leq 0.1$. Shallow-(PGD, MI, DI, TI) means applying PGD, MI, DI and TI to the shallow layers of the model. Deep-(PGD, MI, DI and TI) means applying PGD, MI, DI and TI to the model's output. EFT-(PGD, MI, DI and TI) means applying ETF combined with PGD, MI, DI or TI to the shallow layers. (The lower, the better)

Model	VGG19 [51]	Inception v3[53]	RN152 [21]	DenseNet [23]	SENet [22]	WRN [59]	MobileNet v3[49]	Average
Clean	67.43	64.36	74.21	73.34	51.28	73.22	65.06	66.99
Autoattack[9]	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Deep-PGD	49.01±0.23	52.26±0.25	60.71±0.74	57.92±0.37	27.94±0.18	60.18±0.64	44.20±0.63	50.31±0.52
Deep-MI	38.92±0.43	42.37±0.37	49.53±0.49	49.06±0.89	19.44±0.75	49.11±0.82	33.46±0.80	40.69±0.96
Deep-DI	43.34±0.40	43.13±0.52	53.78±0.38	55.41±0.53	23.53±0.52	51.77±0.48	38.14±0.74	44.15±0.60
Deep-TI	49.46±0.52	49.64±0.27	58.89±0.71	58.75±0.30	26.19±0.16	56.31±0.58	44.02±0.46	49.03±0.51
Shallow-PGD	22.93±0.33	31.07±0.58	34.71±0.67	36.20±0.87	13.08±0.36	32.16±0.66	16.65±0.54	26.69±0.49
Shallow-MI	22.62±0.25	30.83±0.48	34.05±0.27	35.74±0.76	12.31±0.41	29.98±0.65	17.72±0.31	26.17±0.56
Shallow-DI	22.14±0.39	29.78±0.17	35.51±0.33	35.79±0.61	8.99±0.42	30.61±0.88	16.88±0.47	25.67±0.55
Shallow-TI	21.82±0.45	28.54±0.34	34.78±0.15	34.71±0.39	7.96±0.48	30.14±0.85	15.77±0.51	24.81±0.37
ETF-PGD	**14.11**±0.24	20.22±0.29	24.20±0.34	24.74±0.37	6.96±0.44	**20.73**±0.28	**10.66**±0.31	**17.37**±0.35
ETF-MI	15.32±0.52	19.97±0.28	26.25±0.14	28.10±0.65	7.02±0.43	22.21±0.66	12.23±0.32	18.72±0.45
ETF-DI	14.77±0.35	20.63±0.32	23.71±0.83	25.70±0.51	7.23±0.37	20.22±0.64	11.53±0.50	17.68±0.47
ETF-TI	15.45±0.37	**18.03**±0.34	**22.63**±0.45	**24.20**±0.68	**6.94**±0.41	21.53±0.25	12.88±0.34	17.38±0.71
Deep*-PGD	12.43±0.51	28.15±0.43	16.54±0.49	12.61±0.22	7.09±0.32	13.33±0.54	9.64±0.28	14.25±0.37
Deep*-MI	11.77±0.75	25.14±0.56	18.10±0.64	13.72±0.34	4.26±0.35	14.61±0.37	8.30±0.37	13.70±0.68
Deep*-DI	7.61±0.41	18.17±0.45	8.23±0.33	9.90±0.57	6.66±0.34	9.72±0.42	7.91±0.46	9.74±0.55
Deep*-TI	9.55±0.48	23.48±0.86	13.51±0.46	10.63±0.64	6.46±0.26	10.92±0.61	9.55±0.35	12.01±0.43

Deep* refers to the attacks mounted in the model trained on the large-scale training data.

Experiment



(a) (b)

Figure 1: (a) How the lightweight attack performance of our approach varies with the number of images used for training the surrogate model. (b) The influence of low-level feature extraction at different layers of ResNet-18 on lightweight black-box attack performance. (The lower, the better)

Table 3: The performance of different attacks on the adversarial trained ResNet-50 [13]. Therein, ϵ refers to the constraint ℓ_∞ in adversarial examples for adversarial training. The accuracy (%) is evaluated on 1000 adversarial examples. $\epsilon = 0.1$ (the lower the better). White-box refers to Auto-Attack [9].

Adv_model	Clean	ETF ours	Black-box [40]	No-box [34]	White-box [9]
$\epsilon = 0/255$	69.43	16.97	**8.20**	24.53	0.00
$\epsilon = 4/255$	55.62	**29.13**	48.11	39.62	0.00
$\epsilon = 8/255$	41.68	**26.14**	38.24	35.87	0.48

Table 4: Model accuracy (%) under lightweight black-box attacks under challenging scenarios, where supervision information or the in-distribution data are unavailable, named Unsupervised and OOD.

Model	VGG19 [51]	Inception v3[53]	RN152 [21]	DenseNet [23]	SENet [22]	WRN [59]	MobileNet v3[49]	Average
Clean	67.43	64.36	74.21	73.34	51.28	73.22	65.06	66.99
Supervised	14.11	20.22	**24.20**	24.74	6.96	**20.73**	10.66	17.37
Unsupervised	15.54	**19.16**	26.27	23.75	7.66	22.79	11.43	18.08
OOD	**6.13**	21.72	25.44	**21.89**	**5.02**	24.33	**7.16**	**15.96**

Experiment

(a) ETF (b) Deep (c) Deep*

Figure 2: Adversarial examples crafted by: a) ETF, b) Deep, and c) Deep* attacks.

Acknowledgement

Thanks for your listening!