西安交通大學
XI'AN JIAOTONG UNIVERSITY

2024 4th International Symposium on Artificial Intelligence and Big Data (AIBDF 2024)

Security Analysis of Machine Learning Lifecycle

Zhengyu Zhao (赵正宇)

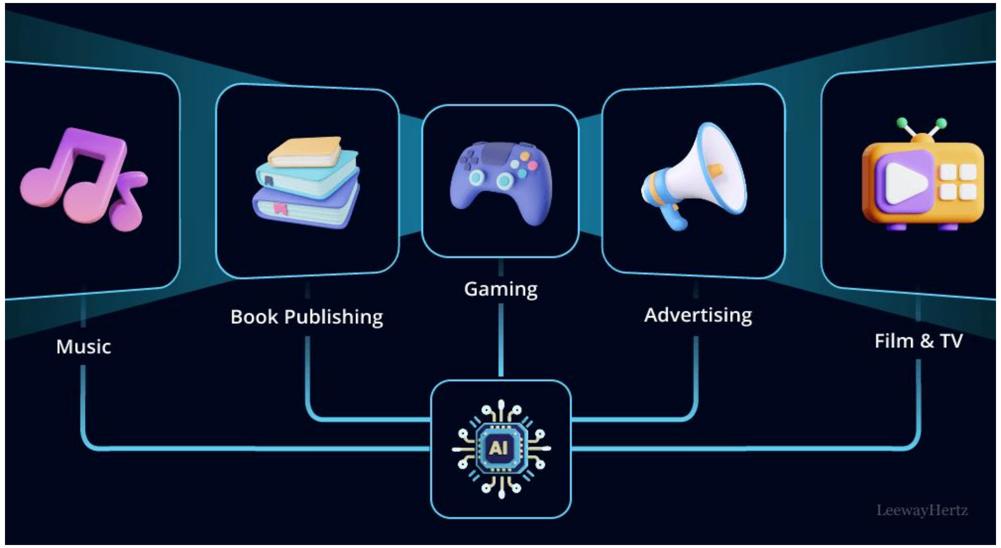
Xi'an Jiaotong University (西安交通大学)

2024/12/28

智能网络与网络安全教育部重点实验室
Ministry of Education Key Lab For Intelligent Networks and Network Security

信息物理融合系统教育部工程研究中心
Ministry of Education Engineering Research Center for Cyber-Physical Systems

Success of AI



(Own and Derived) Problems of AI

Individual ⇨ Society ⇨ Country

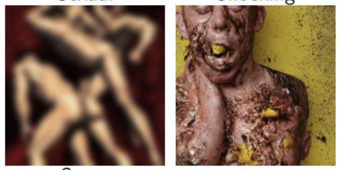
Social engineering

Deepfakes about wars

Sensitive data leakage

AIGC NSFW content

Sexual Shocking



2016

2017

2018

2019

2020

2021

2022

2023

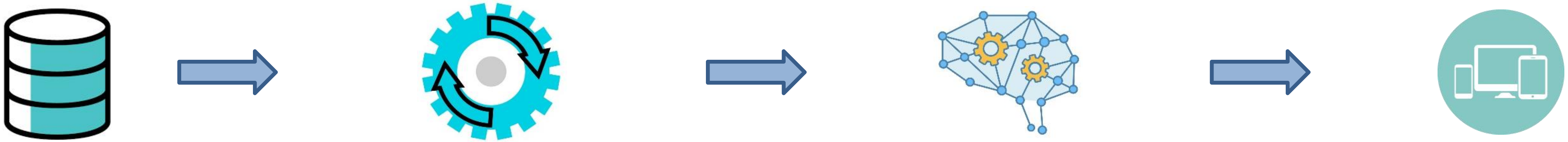
Deepfakes about politics

Autonomous driving accidents

Deepfakes about COVID-19

ChatGPT for telecom fraud

Our Group: Security Analysis of ML Lifecycle



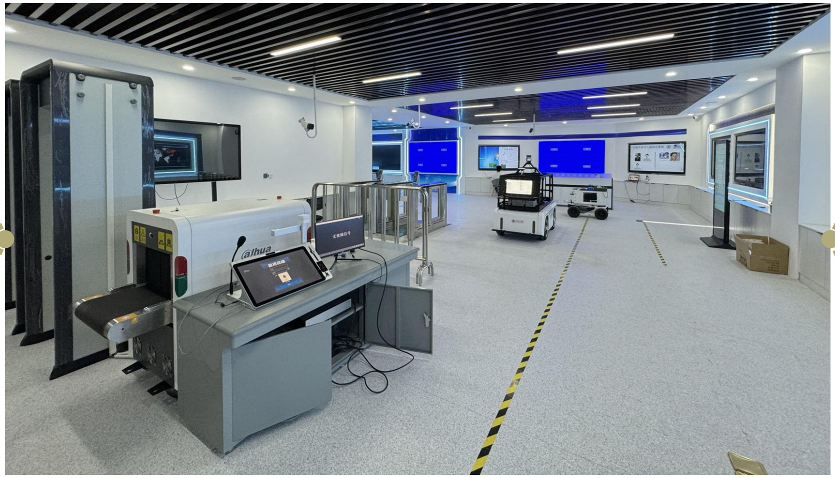
Data | Train/Develop | Test/Deploy | Application

Poison&Deepfake	Failures&Bias	OOD&Adv. Example	Auto-driving&More
NDSS'21, ICML'23 ICLR'23, EMNLP'23 ACL'24, NeurIPS'24 ACL'24, NAACL'24	ICSE'21, CCS'22 USENIX'22, NDSS'22 NeurIPS'22, FSE'23 ISSTA'23, ISSTA'24	USENIX'19, CVPR'20 NeurIPS'21, USENIX'23 TIFS'23, TIFS'24 FSE'24, AAAI 2025	ICML'24, CVPR'24 AAAI'24, TIFS'24

Our Group: Real-world Application Scenarios

Identity Authentication

Autonomous Driving

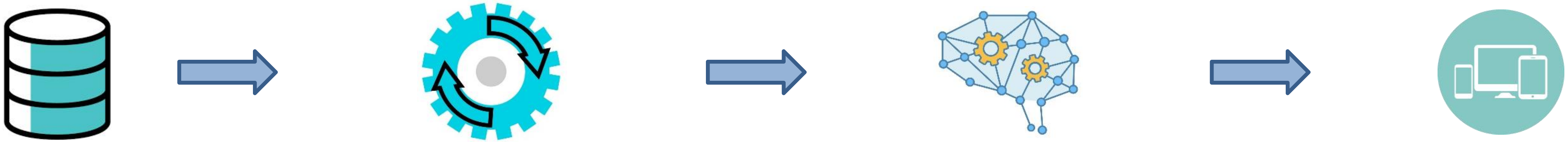


AISEC Lab

Behavior Analysis

Smart Finance

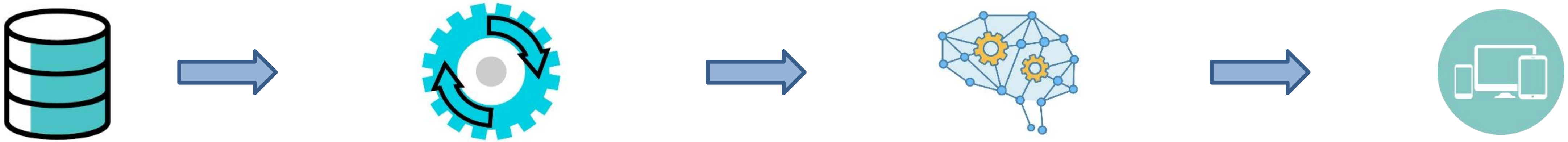
Security Analysis of ML Lifecycle



Data | Train/Develop | Test/Deploy | Application

Poison&Deepfake	Failures&Bias	OOD&Adv. Example	Auto-driving&More
NDSS’21, ICML’23 ICLR’23, EMNLP’23 ACL’24, NeurIPS’24 ACL’24, NAACL’24	ICSE’21, CCS’22 USENIX’22, NDSS’22 NeurIPS’22, FSE’23 ISSTA’23, ISSTA’24	USENIX’19, CVPR’20 NeurIPS’21, USENIX’23 TIFS’23, TIFS’24 FSE’24, AAAI 2025	ICML’24, CVPR’24 AAAI’24, TIFS’24

Security Analysis of ML Lifecycle: Four Studies



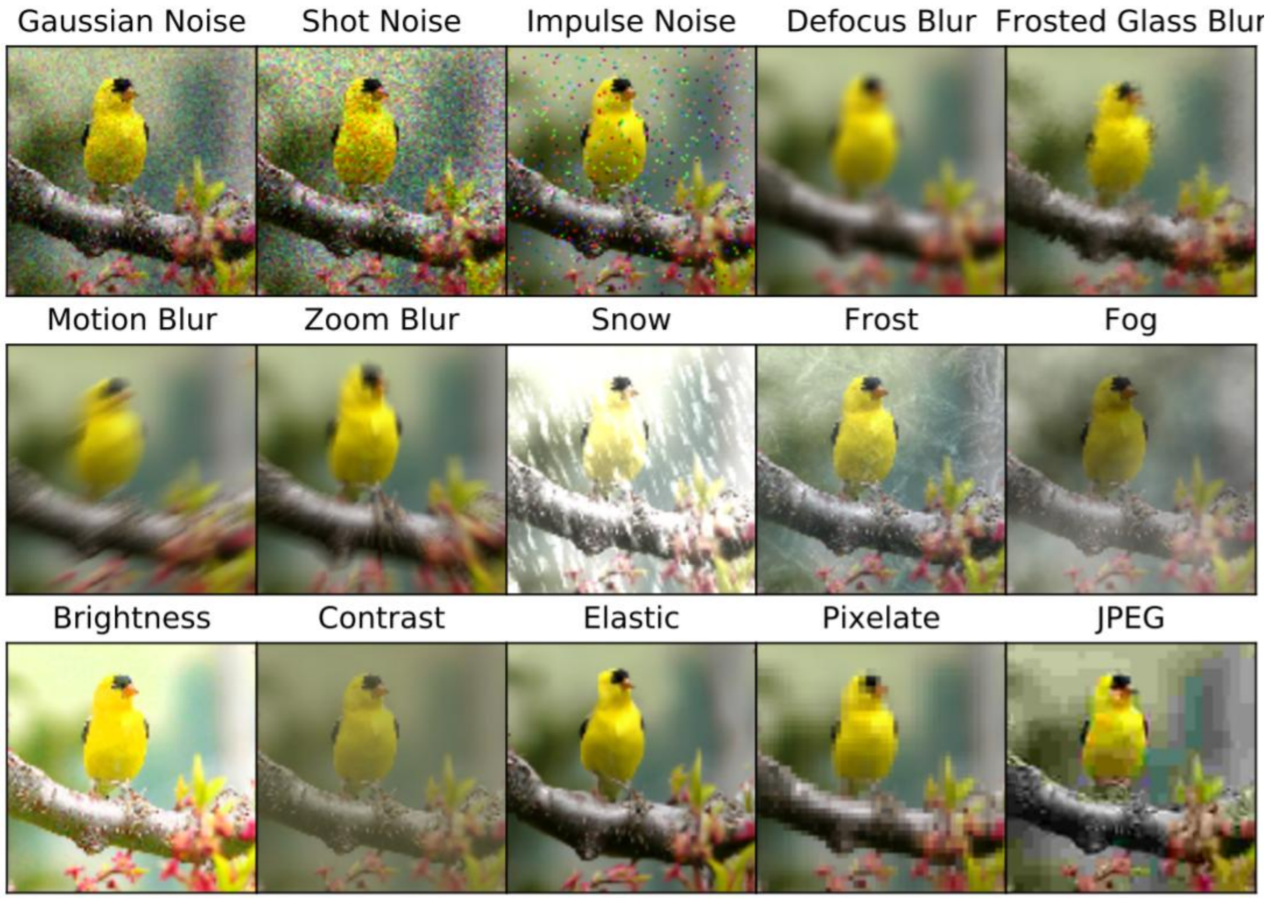
Data → Train/Develop → Test/Deploy → Application

Data	Train/Develop	Test/Deploy	Application
Poison&Deepfake	**Failures**&Bias	OOD&**Adv. Example**	**Auto-driving**&More
NDSS'21, **ICML'23** ICLR'23, EMNLP'23 ACL'24, NeurIPS'24 ACL'24, NAACL'24	**ICSE'21**, CCS'22 USENIX'22, NDSS'22 NeurIPS'22, FSE'23 ISSTA'23, ISSTA'24	USENIX'19, CVPR'20 **NeurIPS'21**, USENIX'23 TIFS'23, TIFS'24 FSE'24, AAAI 2025	ICML'24, **CVPR'24** AAAI'24, TIFS'24
③	④	①	②

① Transferable Targeted Adversarial Examples (NeurIPS'21)

① Transferable Targeted Adversarial Examples (NeurIPS'21)

Noisy Examples → **Adversarial Examples**

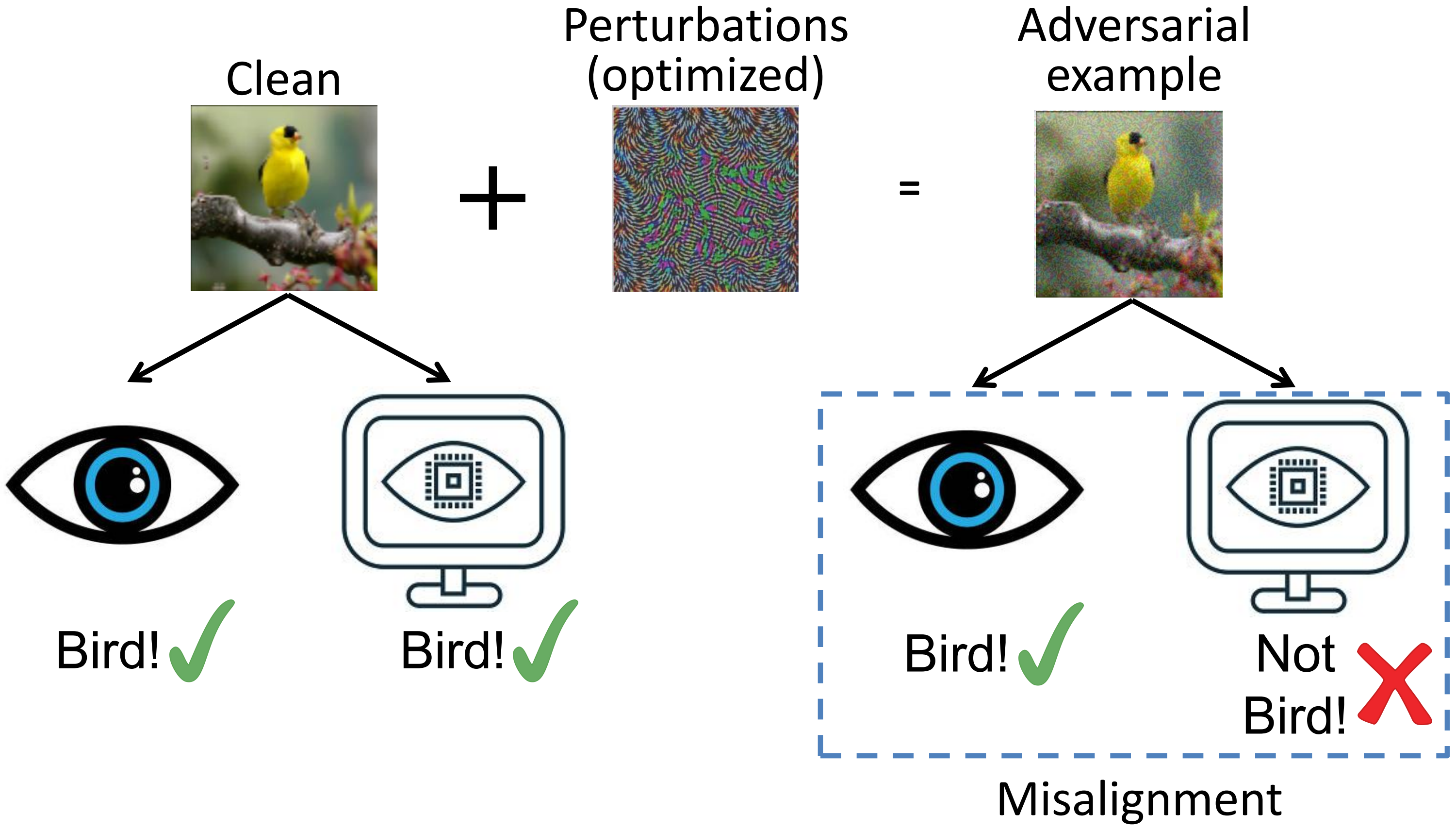


Common

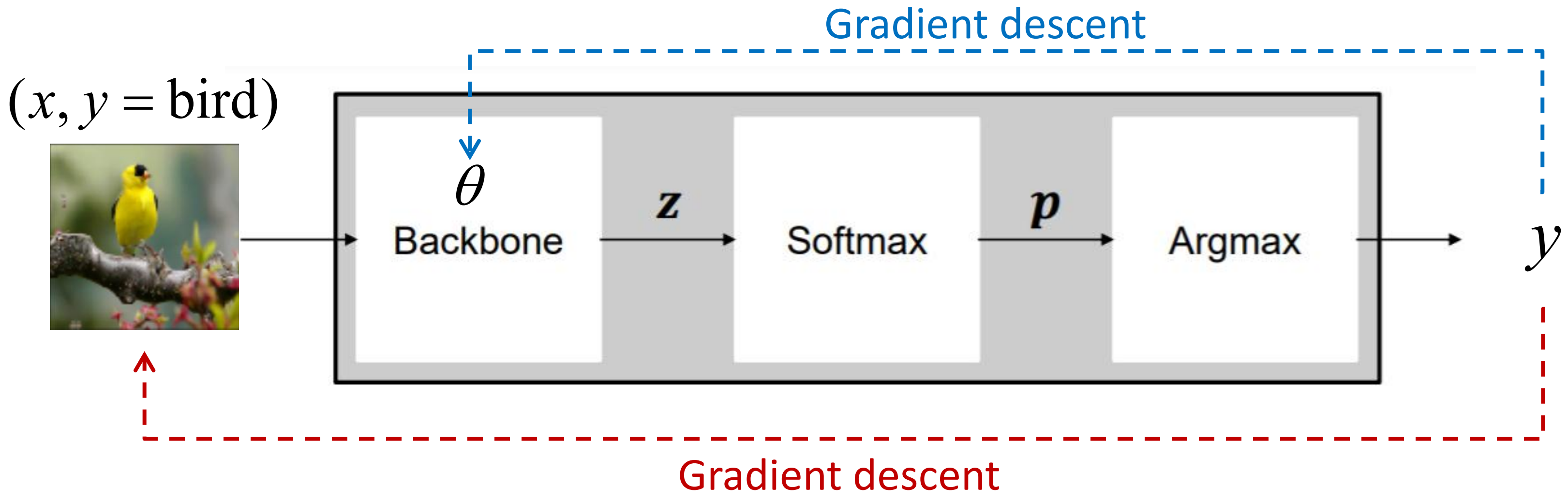
→

Intentional
(optimized)

① Transferable Targeted Adversarial Examples (NeurIPS'21)



① Transferable Targeted Adversarial Examples (NeurIPS'21)



Model training

$$\boxed{\theta' = \arg\min_{\theta} d(y, y_{\text{bird}})}$$

Adversarial example

$$\boxed{x' = \arg\max_{x} d(y, y_{\text{bird}})}$$

$$d(y, y_{\text{bird}}) = -y_{\text{bird}} \log(p)$$

① Transferable Targeted Adversarial Examples (NeurIPS'21)

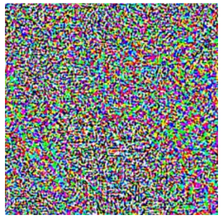
Loss function: $x' = \arg\max_{x} d(y, y_{\text{bird}})$

s.t. $\Big\|$  $-$  $\Big\|_p \le \varepsilon$

x' $\quad$ x

L_2-norm:

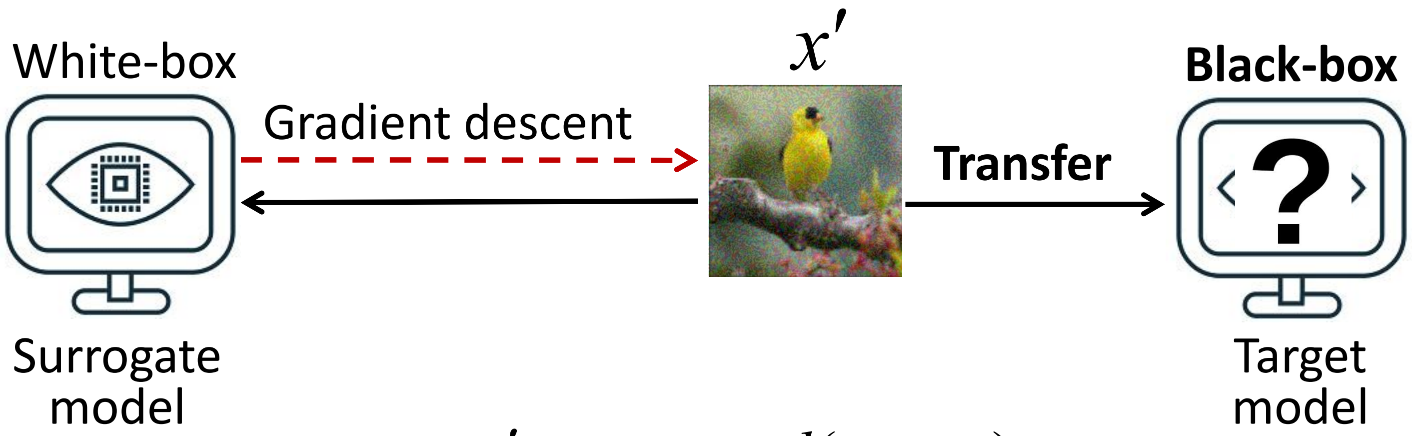
$d = \Delta x_1{}^2 + \Delta x_2{}^2 + \ldots$; total value



L_∞-norm:

$d = \max(\Delta x_1, \Delta x_2, \ldots)$; max value

① Transferable Targeted Adversarial Examples (NeurIPS'21)



$$x' = \arg\max_{x} d(y, y_{\text{bird}})$$

Non-targeted Attack

$$x' = \arg\min_{x} d(y, y_{\text{tar}})$$

Targeted Attack

Targeted adversarial examples achieve **little** transferability.

Previous work[1-6]

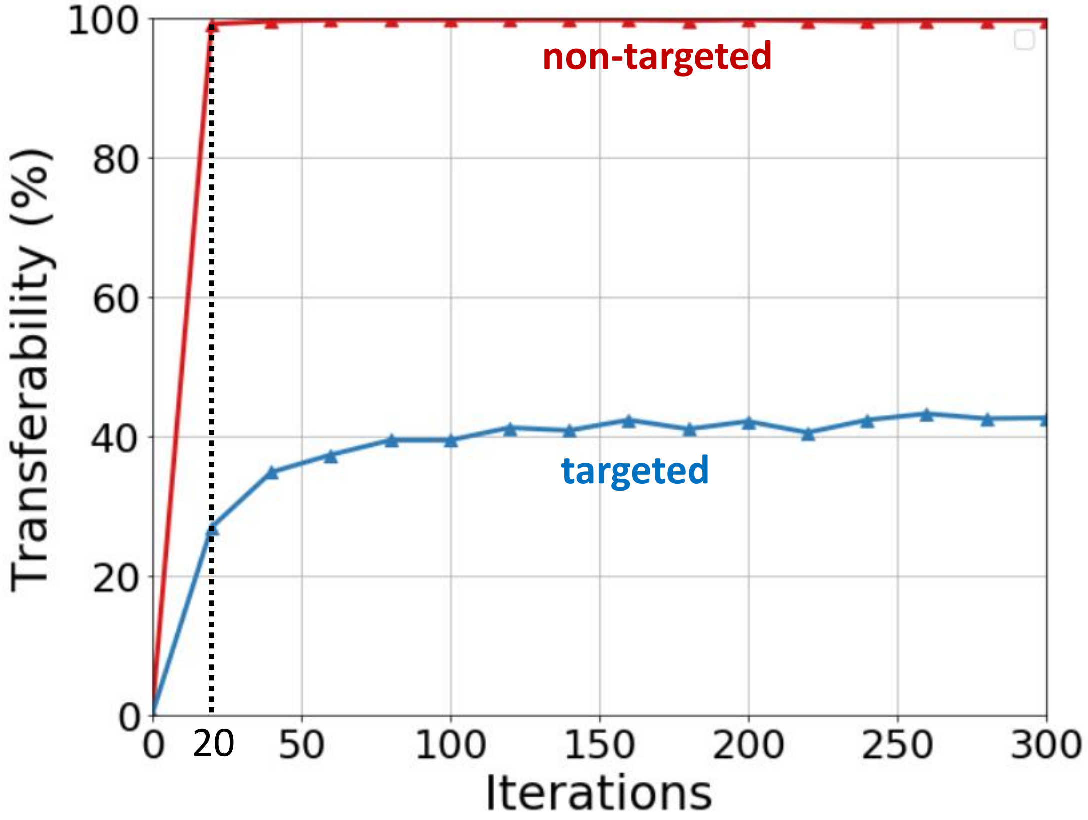
Targeted adversarial examples achieve **Strong** transferability.

Ours

[1] **Delving into transferable adversarial examples and black-box attacks.** Liu et al. ICLR 2017.
[2] **Boosting Adversarial Attacks with Momentum.** Dong et al. CVPR 2018.
[3] **Feature space perturbations yield more transferable adversarial examples.** Inkawhich et al. CVPR 2019.
[4] **Transferable perturbations of deep feature distributions.** Inkawhich et al. ICLR 2020.
[5] **Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability.** Inkawhich et al. NeurIPS 2020.
[6] **On generating transferable targeted perturbations.** Naseer et al. ICCV 2021.

Insight 1: More Iterations

ResNet50→DenseNet121 (MTDI-FGSM)



converge after
100 iterations?

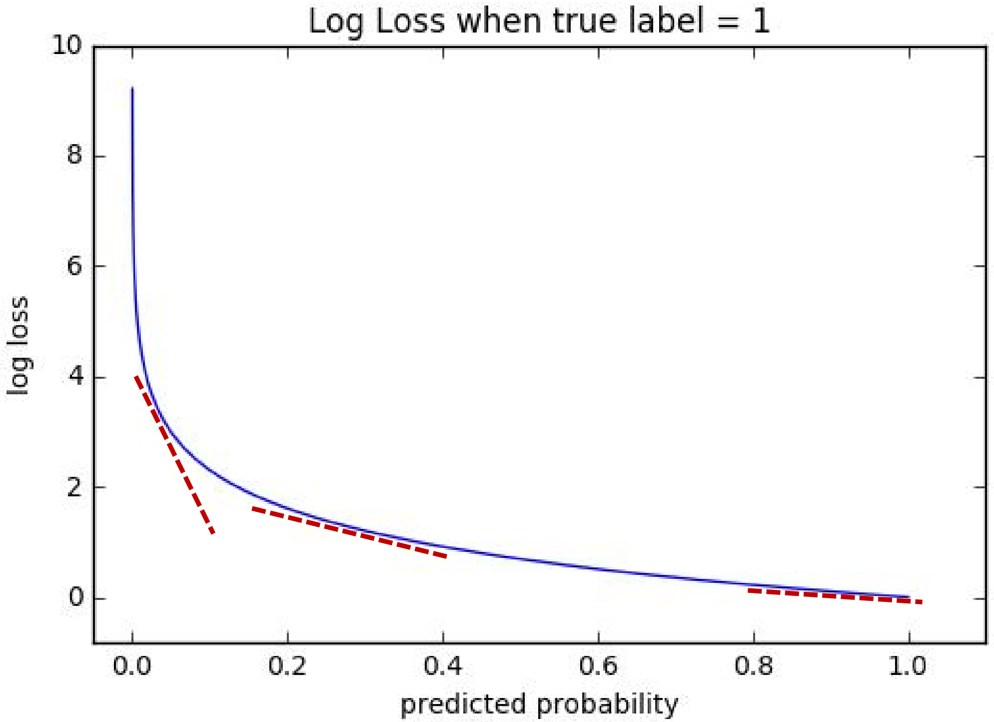
<20 iterations in existing work:

- fail to converge
- fine to use many iterations

Insight 2: Better Loss

$$L_{CE} = -1 \cdot \log(p_t) = -\log(\frac{e^{z_t}}{\sum e^{z_j}}) = -z_t + \cancel{\log(\sum e^{z_j})},$$

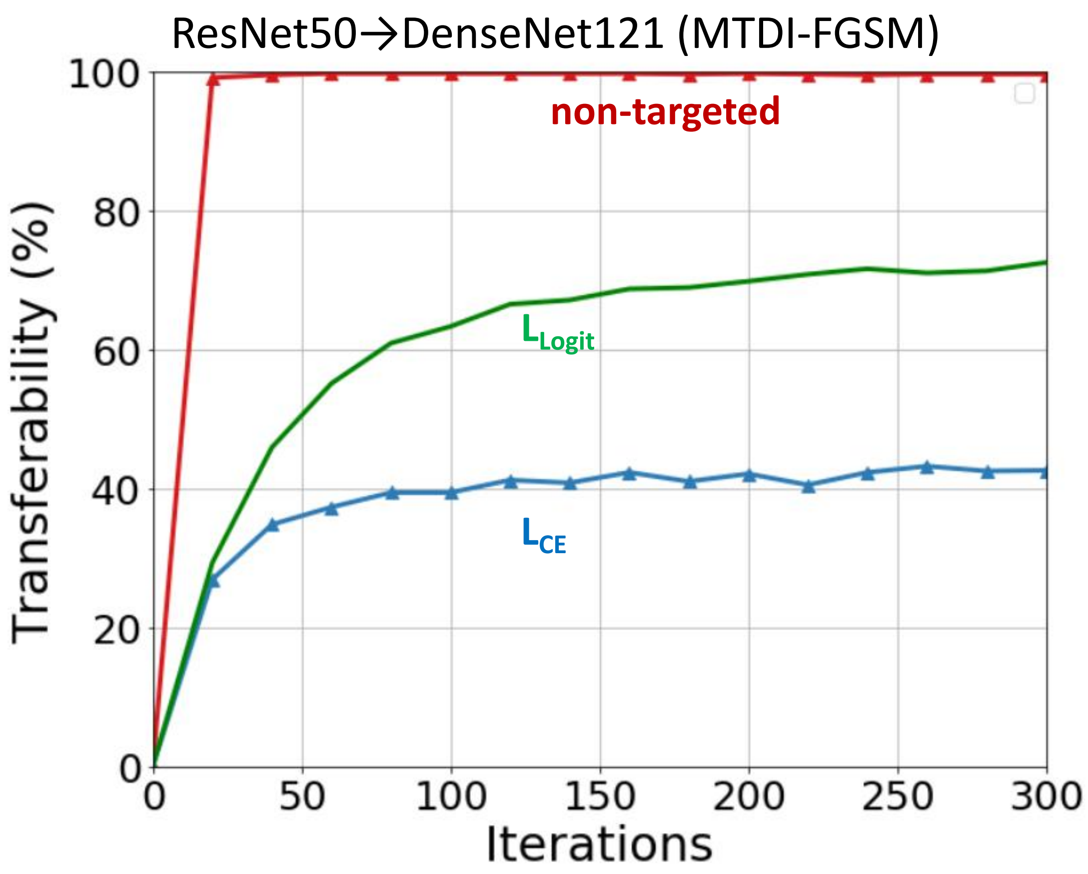
$$\frac{\partial L_{CE}}{\partial z_t} = -1 + \frac{\partial \log(\sum e^{z_j})}{\partial e^{z_t}} \cdot \frac{\partial e^{z_t}}{\partial z_t} = -1 + \frac{e^{z_t}}{\sum e^{z_j}} = \underline{-1 + p_t}.$$



Cross-Entropy Loss (L_{CE}) causes **vanishing gradient** problem

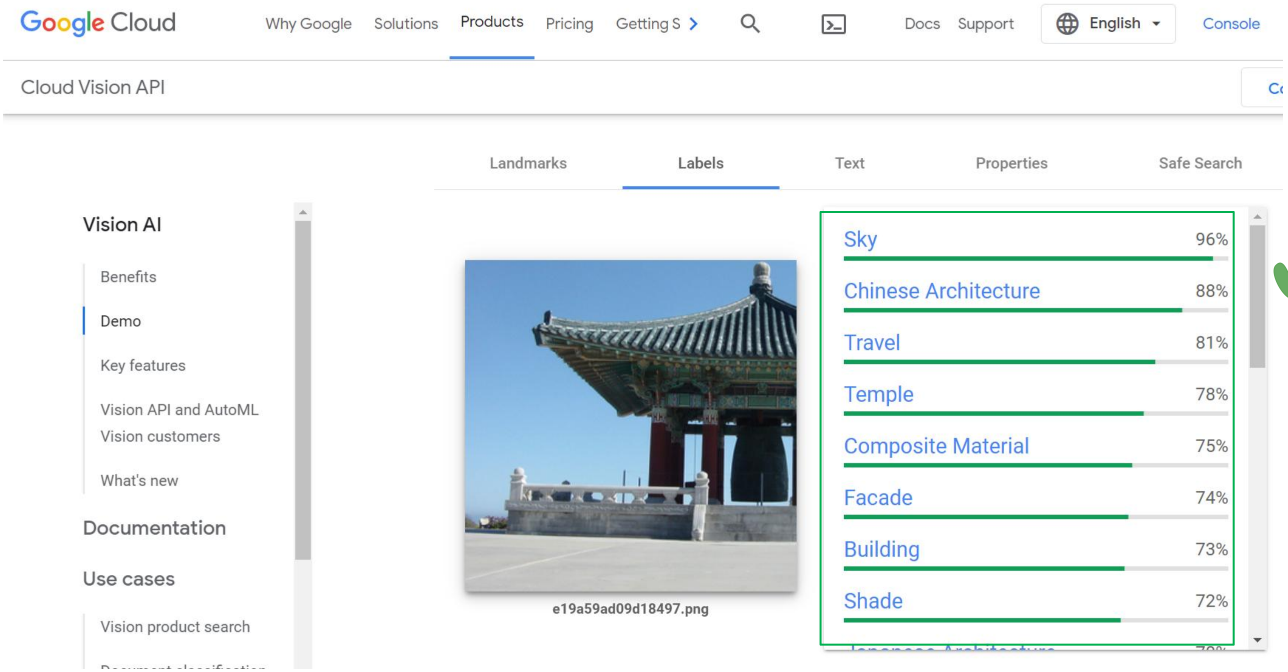
$$L_{Logit} = -z_t, \ \frac{\partial L_{Logit}}{\partial z_t} = -1.$$

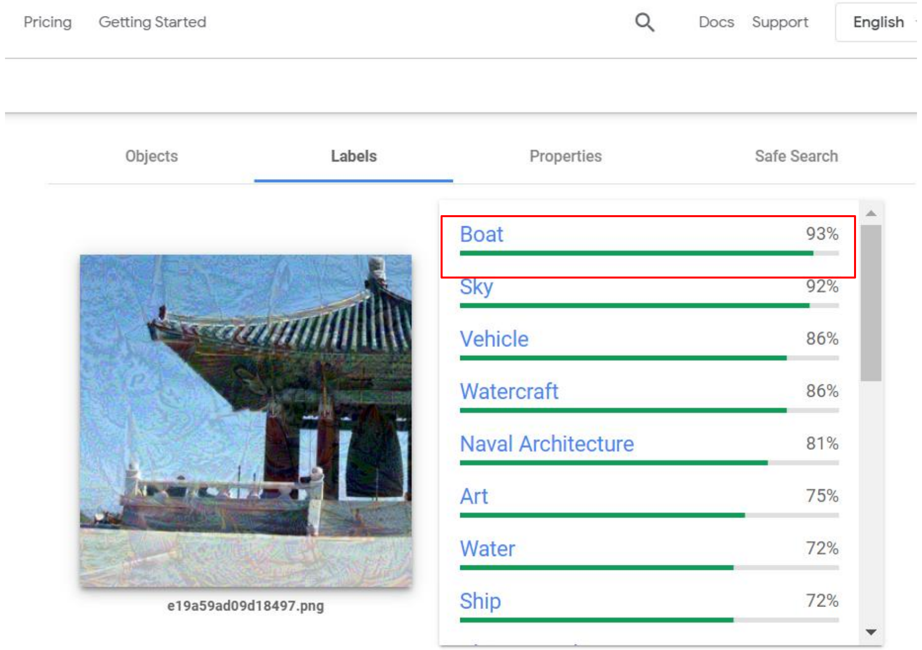
Insight 2: Better Loss



Attacking Google Vision API

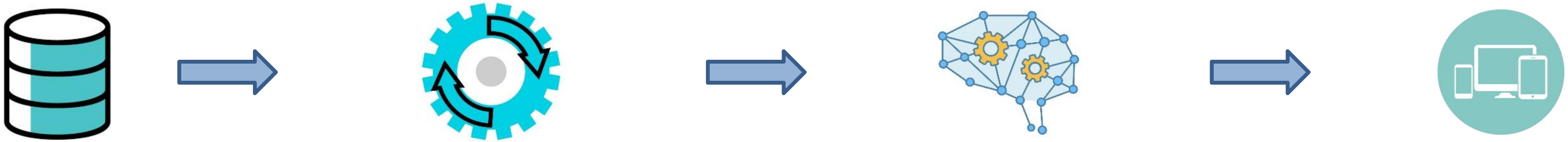
Services	Evaluation	Ori	CE	Po+Trip	Logit
Object localization	targeted	0	9.00	8.50	**19.25**
Label detection	targeted	0	4.50	2.25	**6.25**





y_t = "yawl" (a type of boat)

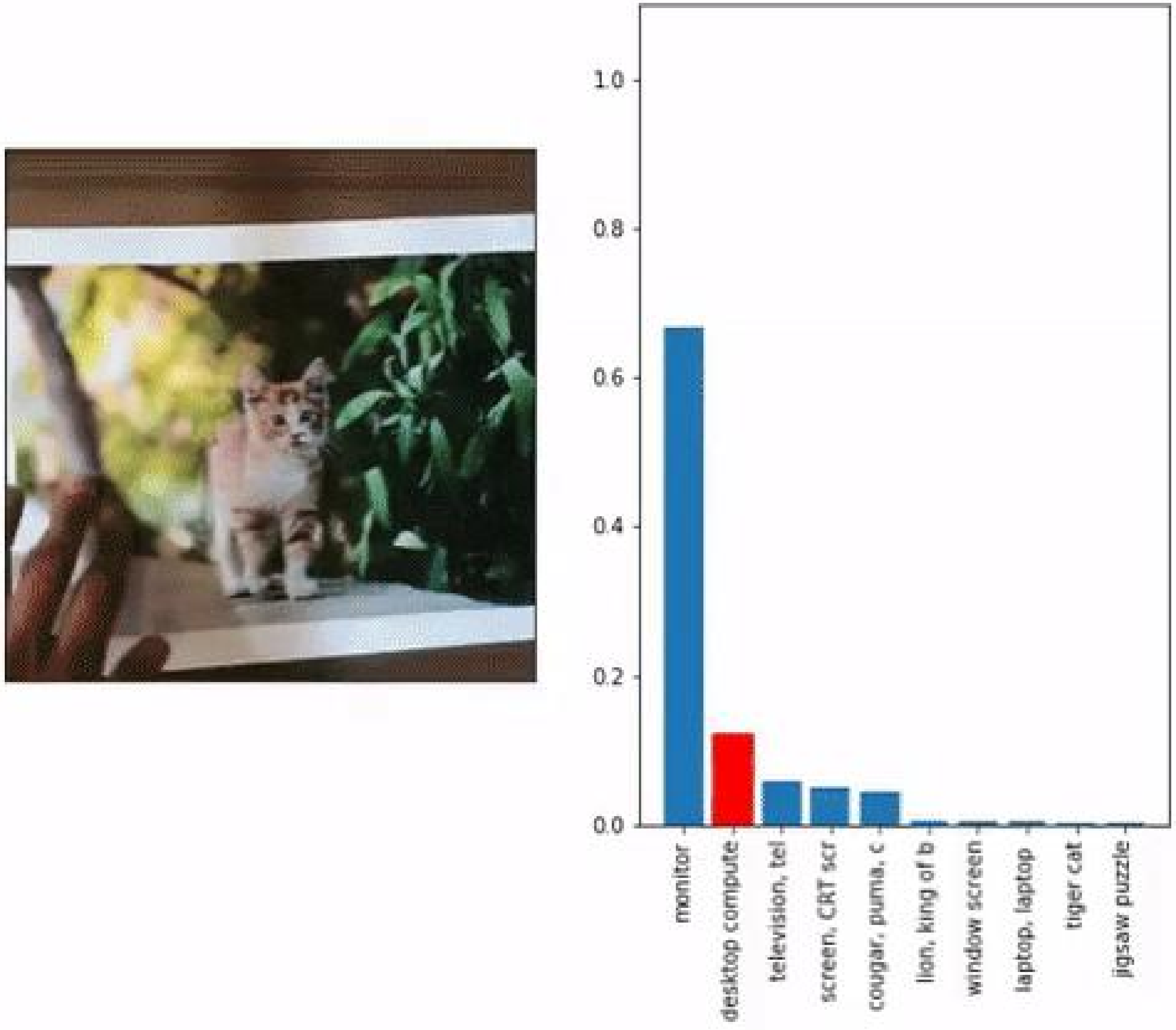
Security Analysis of ML Lifecycle: Four Studies



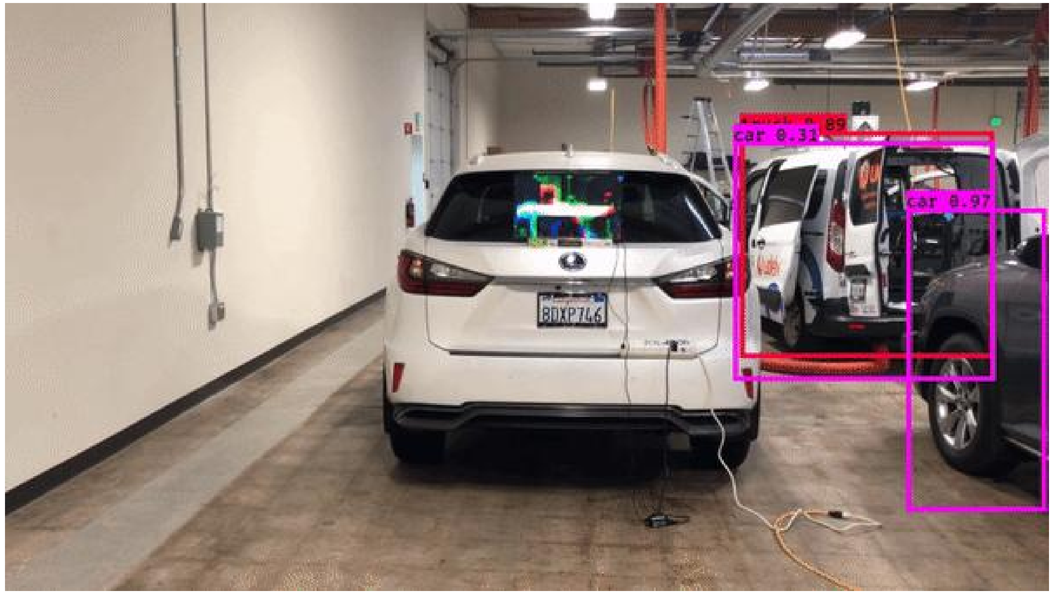
Data | Train/Develop | Test/Deploy | Application

Poison&Deepfake	Failures&Bias	OOD&Adv. Example	Auto-driving&More
NDSS'21, **ICML'23** ICLR'23, EMNLP'23 ACL'24, NeurIPS'24 ACL'24, NAACL'24	**ICSE'21**, CCS'22 USENIX'22, NDSS'22 NeurIPS'22, FSE'23 ISSTA'23, ISSTA'24	USENIX'19, CVPR'20 **NeurIPS'21**, USENIX'23 TIFS'23, TIFS'24 FSE'24, AAAI 2025	ICML'24, **CVPR'24** AAAI'24, TIFS'24
③	④	①	②

② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

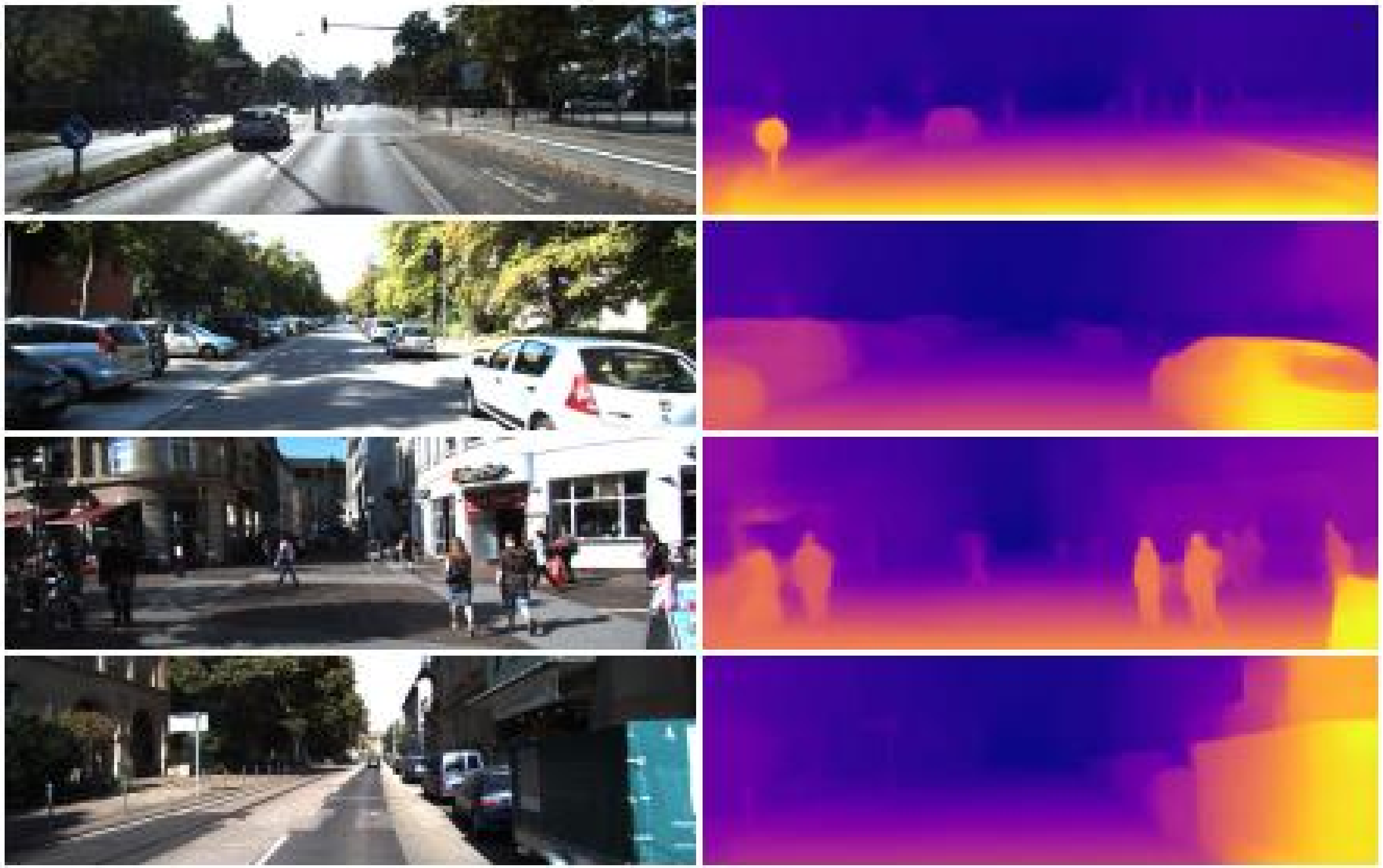


② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)



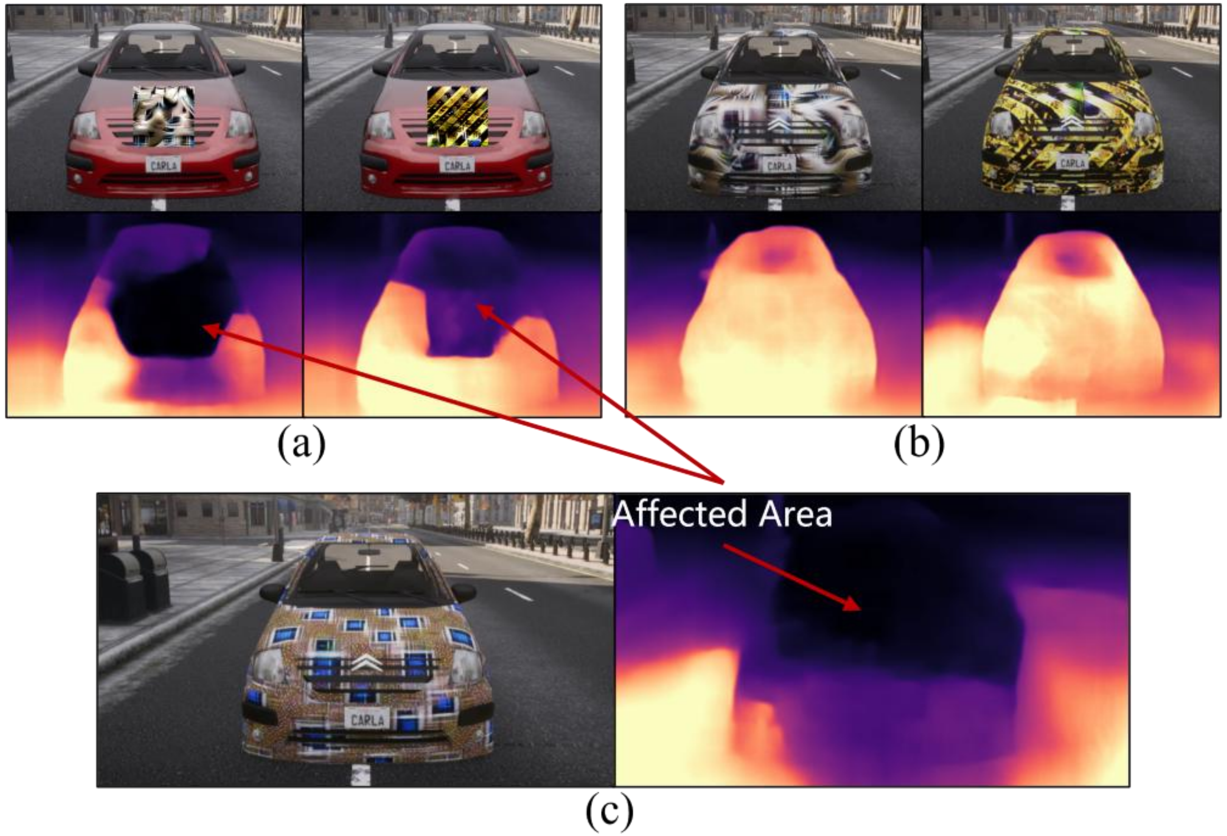
② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- **Monocular Depth Estimation (MDE):** Estimate the depth (distance to the camera)



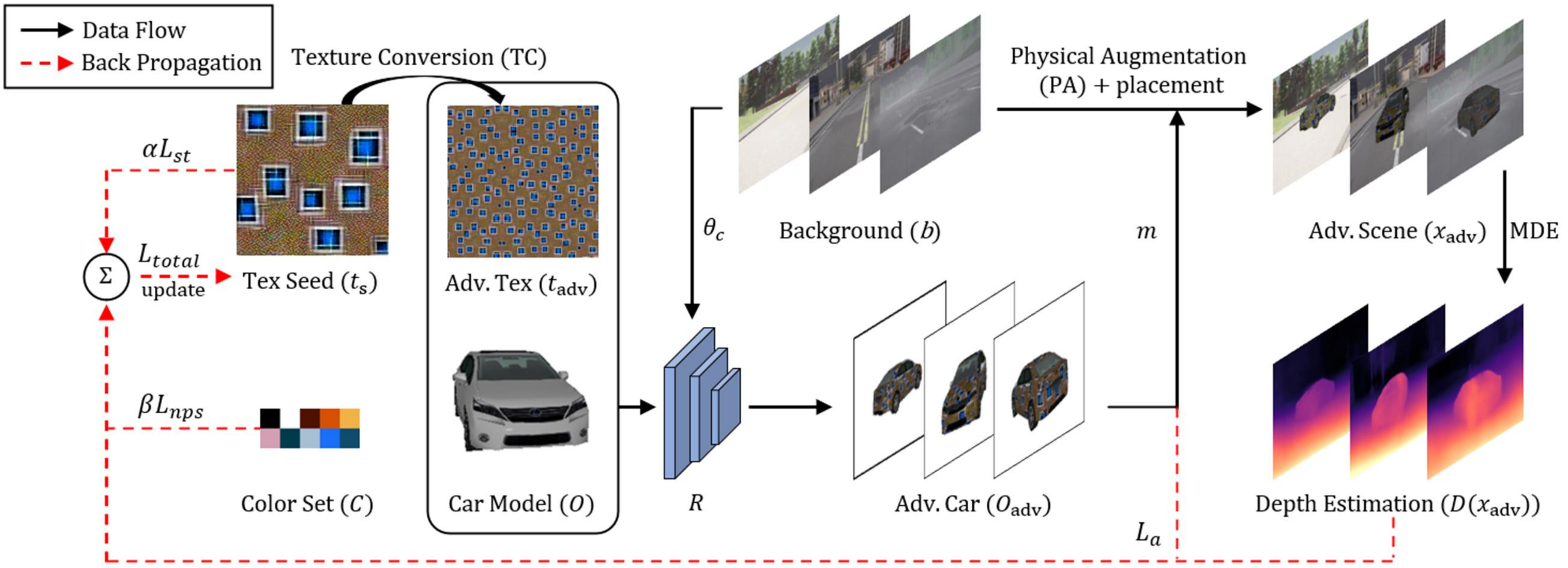
② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- **Drawbacks of existing attacks (a) and (b):**
  - Only Affect a small and localized area
  - Fail at different conditions (e.g., angles, weathers, objects)



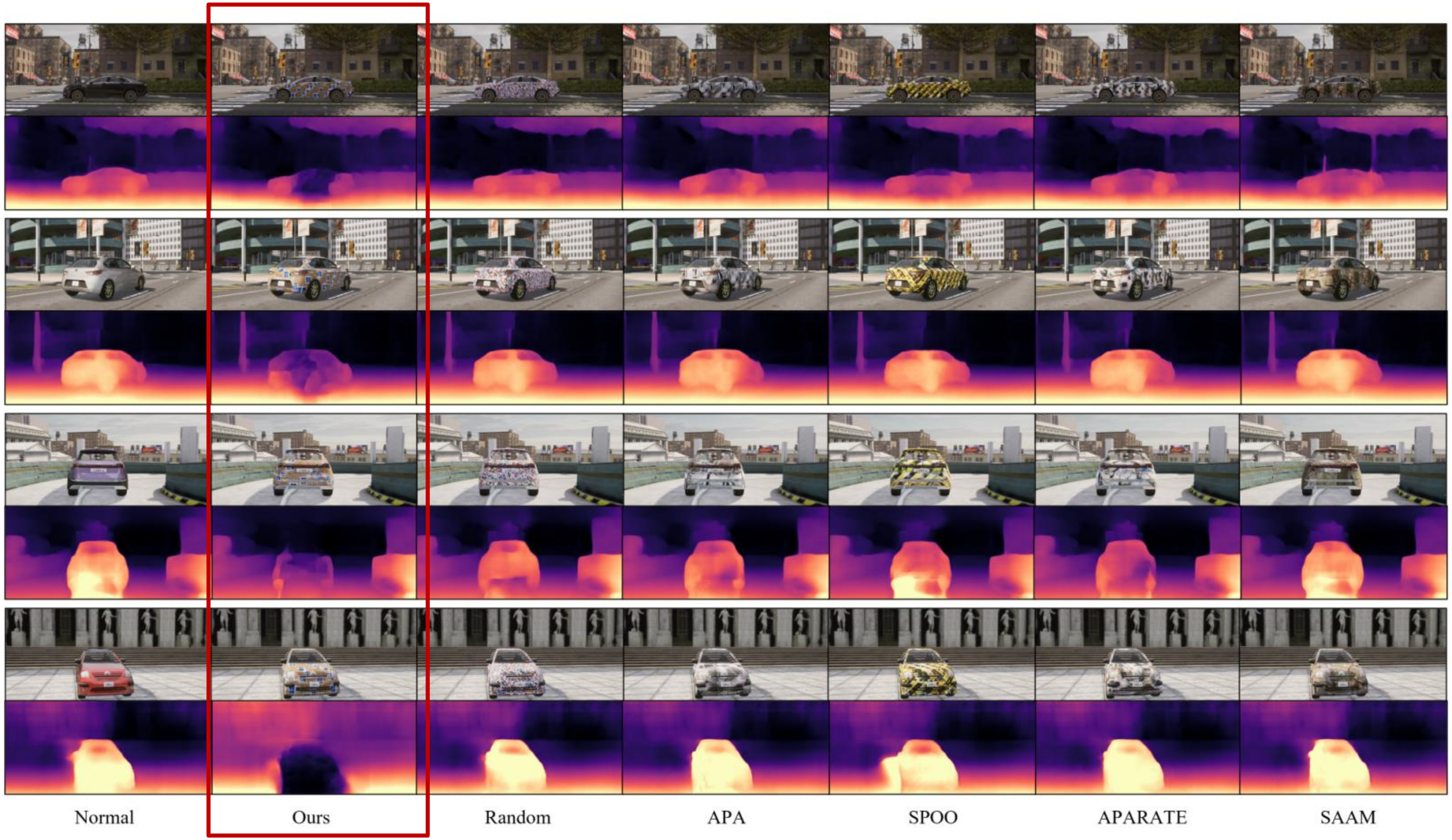
② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- We propose **3D²Fool** to generate robust 3D adversarial textures



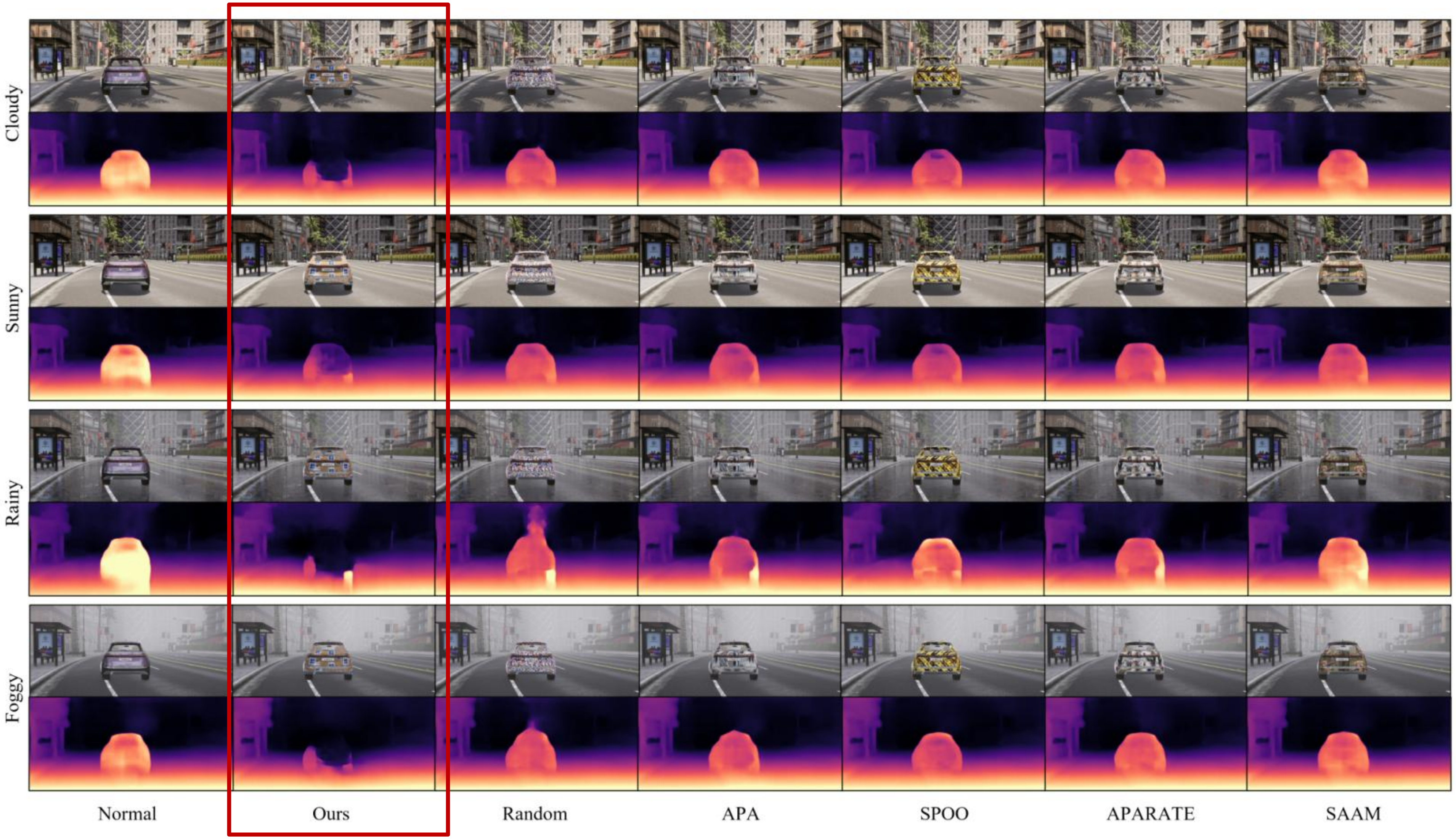
② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- Comparisons at various **angles**



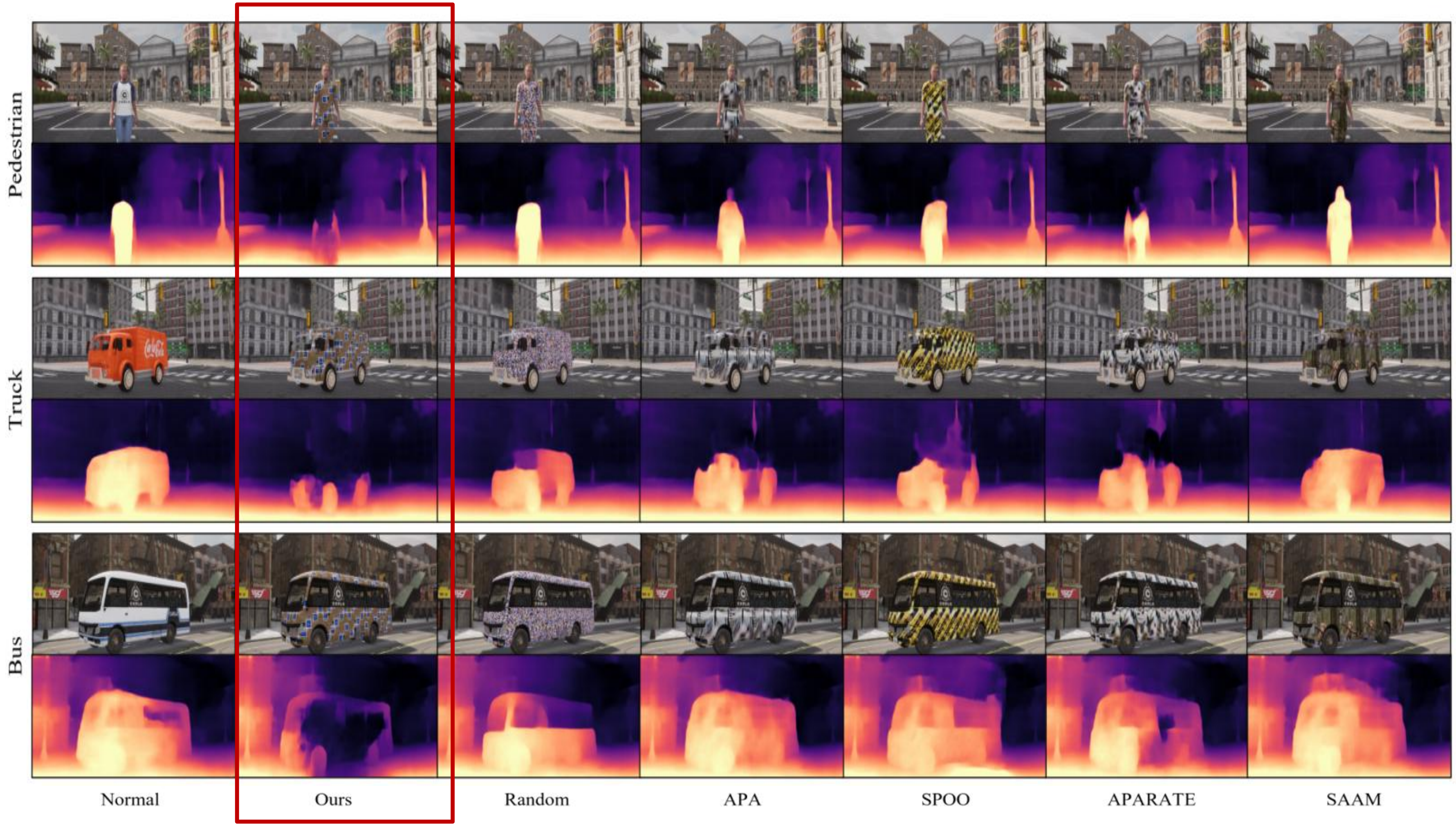
② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- ☐ Comparisons at various **weathers**

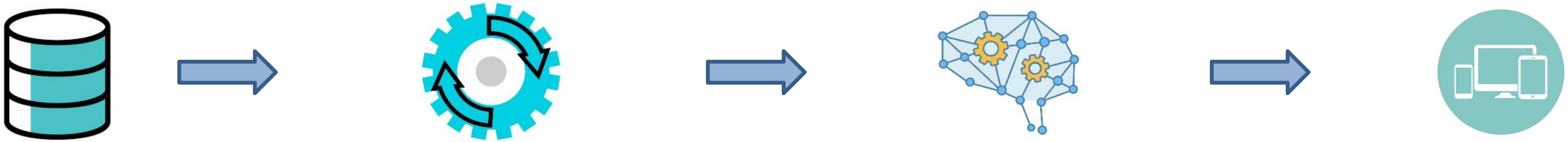


② Physical 3D Adversarial Examples in Auto-Driving (CVPR'24)

- Comparisons at various **objects**



Security Analysis of ML Lifecycle: Four Studies

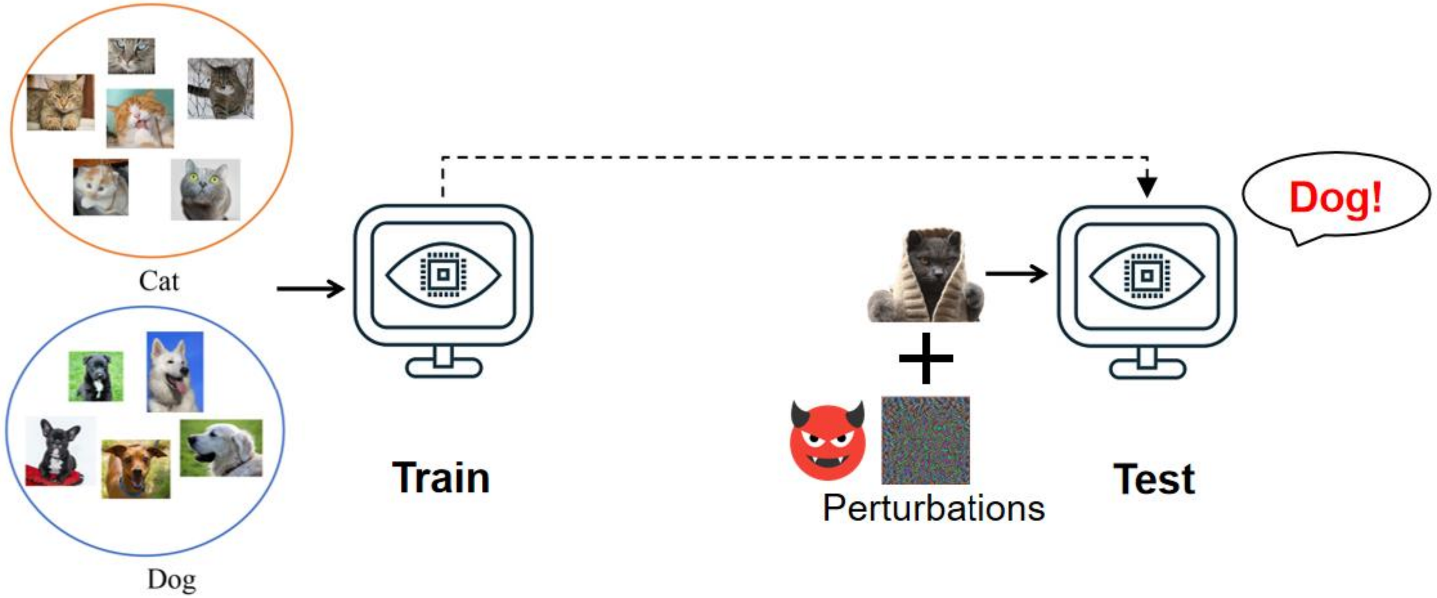


Data | **Train/Develop** | **Test/Deploy** | **Application**

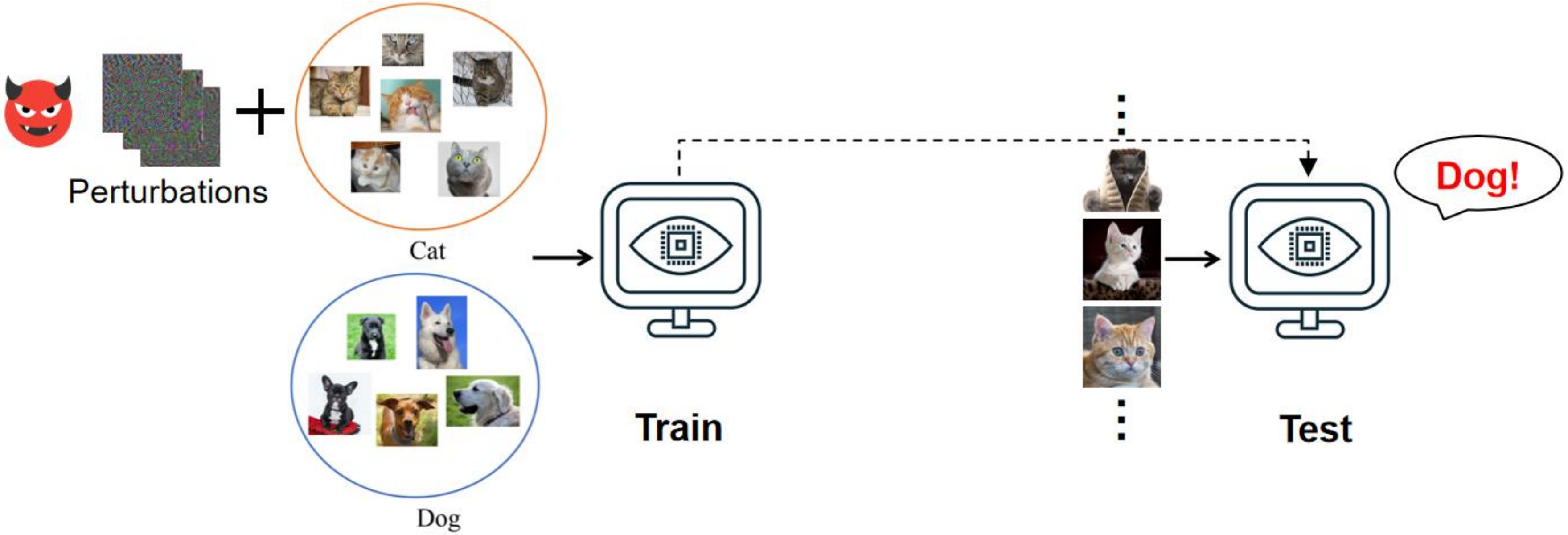
Poison&Deepfake	Failures&Bias	OOD&Adv. Example	Auto-driving&More
NDSS'21, **ICML'23** ICLR'23, EMNLP'23 ACL'24, NeurIPS'24 ACL'24, NAACL'24	**ICSE'21**, CCS'22 USENIX'22, NDSS'22 NeurIPS'22, FSE'23 ISSTA'23, ISSTA'24	USENIX'19, CVPR'20 **NeurIPS'21**, USENIX'23 TIFS'23, TIFS'24 FSE'24, AAAI 2025	ICML'24, **CVPR'24** AAAI'24, TIFS'24
③	④	①	②

③ Defense against Poisons via Image Pre-processing (ICML'23)

(Testing-time)
Adversarial Examples

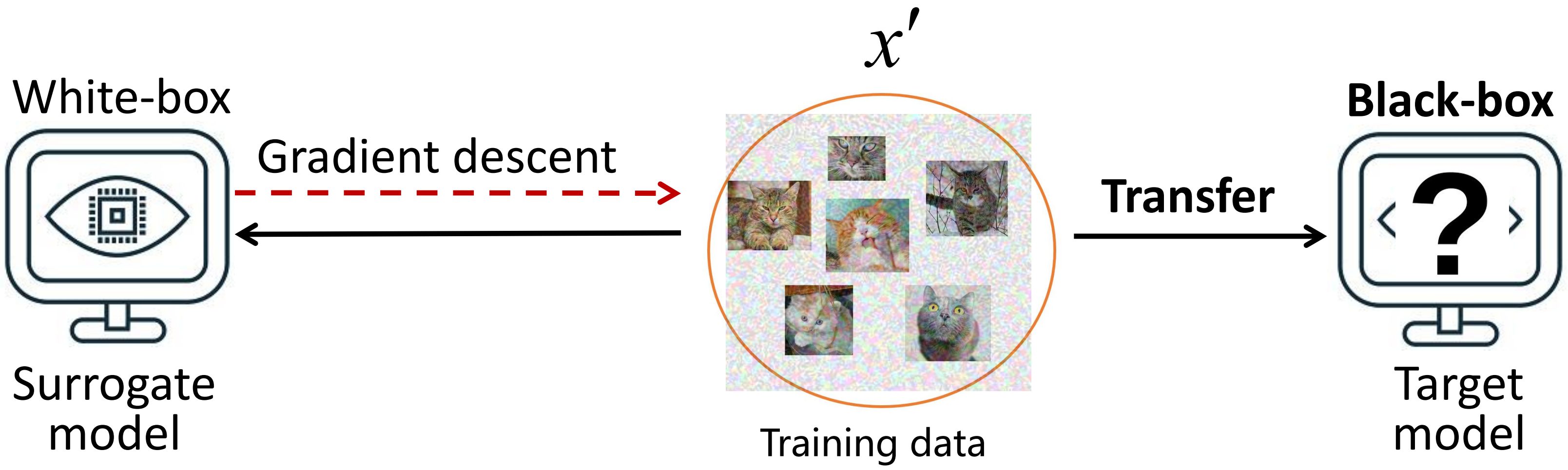


(Training-time)
Data Poisons

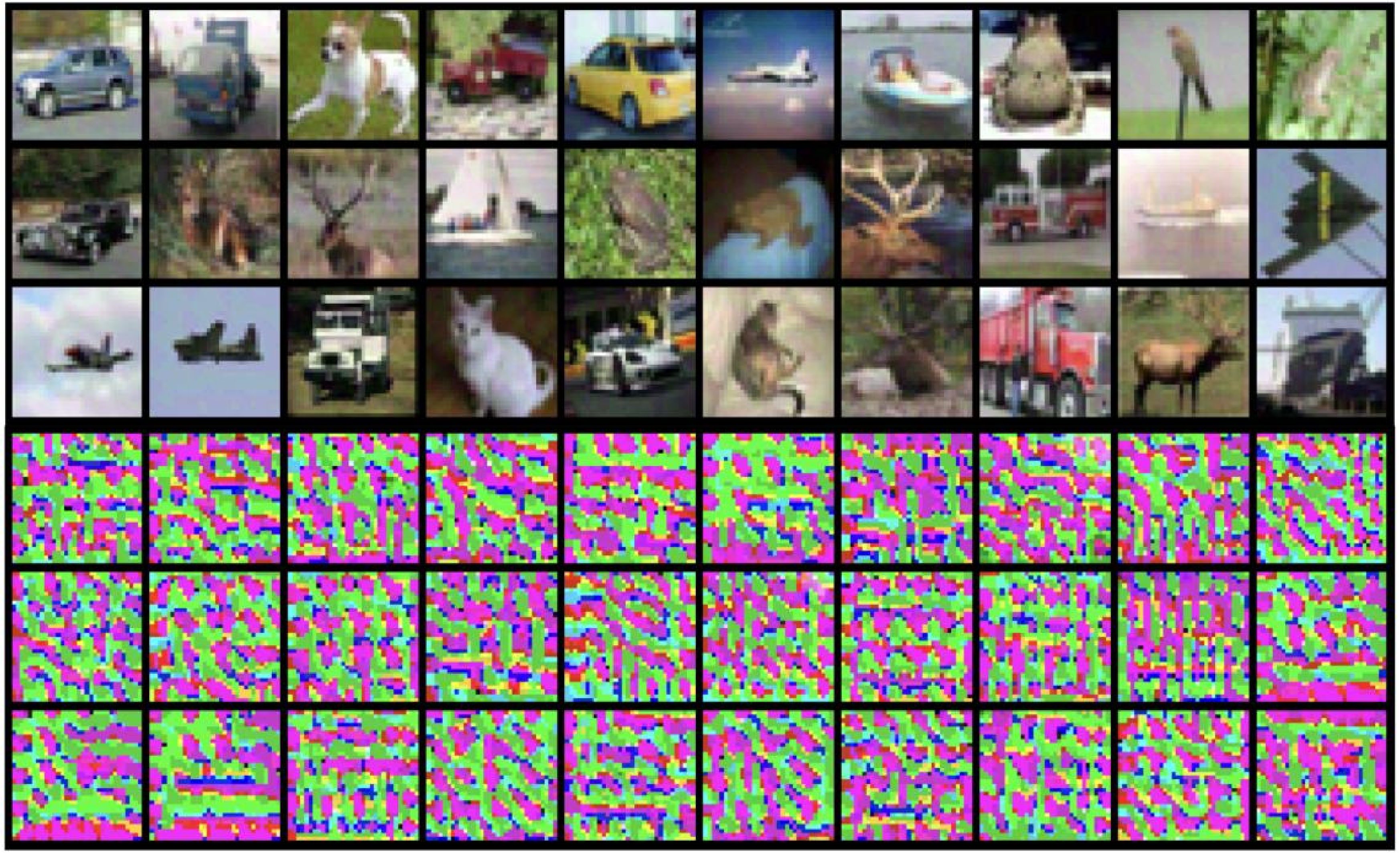


③ Defense against Poisons via Image Pre-processing (ICML'23)

When the poisoning attack happens,
a (fully-trained) target model hasn't existed yet.

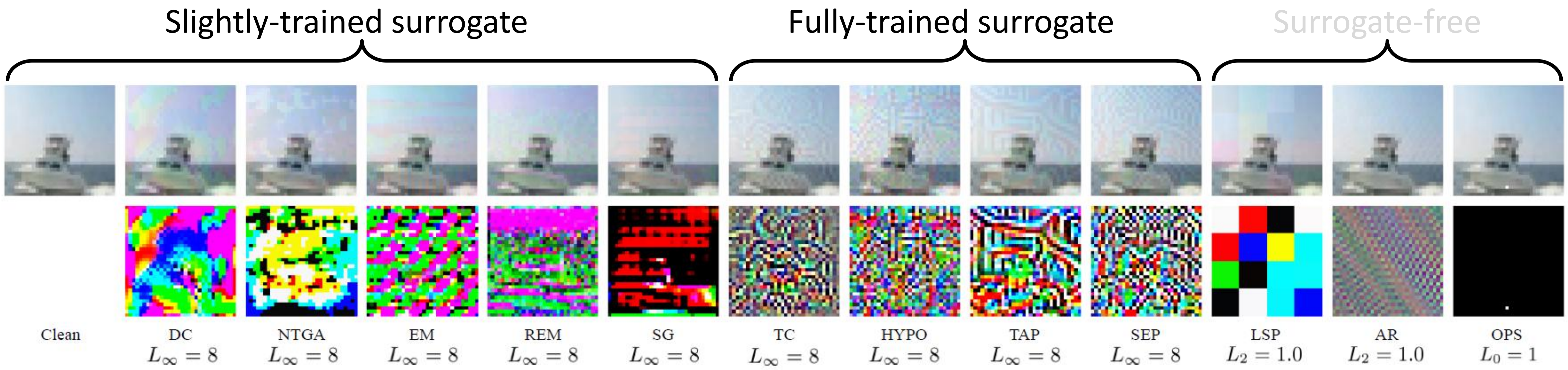


③ Defense against Poisons via Image Pre-processing (ICML'23)

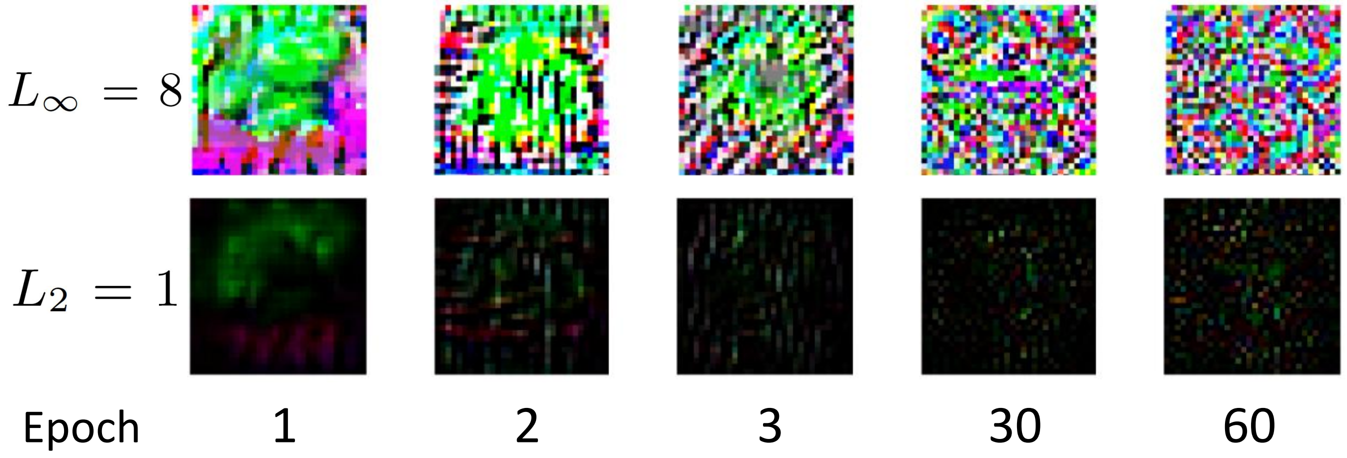


Unlearnable Examples: Making Personal Data Unexploitable. Huang et al. ICLR 2021

③ Defense against Poisons via Image Pre-processing (ICML’23)



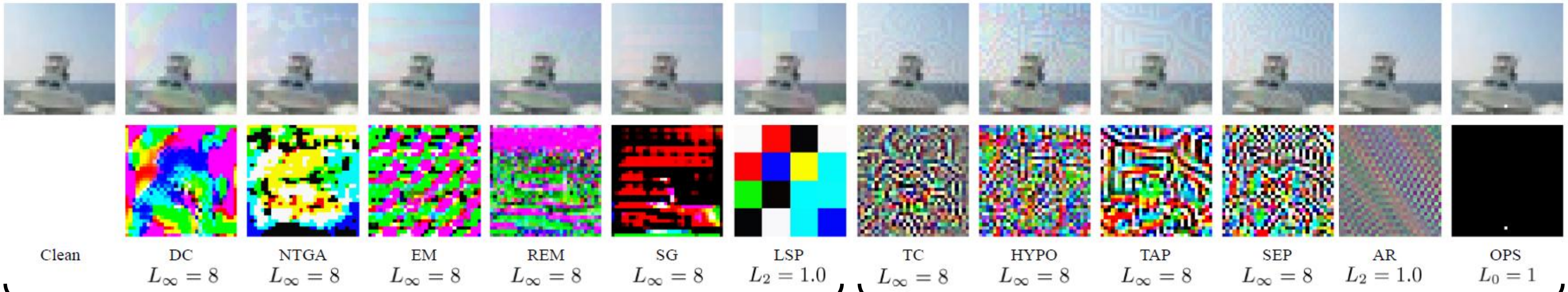
Frequency principle:
Deep neural networks often learn from low to high frequencies during training[1,2,3].



[1] **On the Spectral Bias of Neural Networks.** Rahaman et al. ICML 2019
[2] **Training Behavior of Deep Neural Network in Frequency Domain.** Xu et al. ICONIP 2019
[3] **Theory of the Frequency Principle for General Deep Neural Networks.** Luo et al. CSIAM Trans. Appl. Math. 2021

③ Defense against Poisons via Image Pre-processing (ICML'23)

Low-frequency High-frequency



Color shortcuts Texture shortcuts

Grayscale-based defense:

JPEG-based defense:

③ Defense against Poisons via Image Pre-processing (ICML'23)

ours SOTA

Norm	Poisons/Countermeasures	w/o	Gray	JPEG	Gray+JPEG	AT
	Clean (no poison)	94.68	92.41	85.38	83.79	84.99
$L_\infty = 8$	DC (Feng et al., 2019)	16.30	**93.07**	81.84	83.09	78.00
	NTGA (Yuan & Wu, 2021)	42.46	**74.32**	69.49	69.86	70.05
	EM (Huang et al., 2021)	21.05	**93.01**	81.50	83.06	84.80
	REM (Fu et al., 2021)	25.44	**92.84**	82.28	83.00	82.99
	SG (van Vlijmen et al., 2022)	33.05	**86.42**	79.49	79.21	76.38
	TC (Shen et al., 2019)	88.70	79.75	85.29	82.43	84.55
	HYPO (Tao et al., 2021)	71.54	61.86	**85.45**	82.94	84.91
	TAP (Fowl et al., 2021b)	8.17	9.11	**83.87**	81.94	83.31
	SEP (Chen et al., 2023)	3.85	3.57	**84.37**	82.18	84.12
$L_2 = 1.0$	LSP (Yu et al., 2022)	19.07	82.47	**83.01**	79.05	84.59
	AR (Sandoval-Segura et al., 2022)	13.28	34.04	**85.15**	82.81	83.17
$L_0 = 1$	OPS (Wu et al., 2023)	36.55	42.44	82.53	79.10	14.41

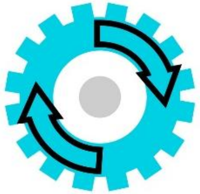
effective++

efficient++

Assumption: Attacks **do not** know our defense, i.e., no adaptive attacks.

Security Analysis of ML Lifecycle: Four Studies

Data



Train/Develop

Test/Deploy

Application

Poison&Deepfake

NDSS'21, **ICML'23**
ICLR'23, EMNLP'23
ACL'24, NeurIPS'24
ACL'24, NAACL'24

③

Failures&Bias

ICSE'21, CCS'22
USENIX'22, NDSS'22
NeurIPS'22, FSE'23
ISSTA'23, ISSTA'24

④

OOD&**Adv. Example**

USENIX'19, CVPR'20
NeurIPS'21, USENIX'23
TIFS'23, TIFS'24
FSE'24, AAAI 2025

①

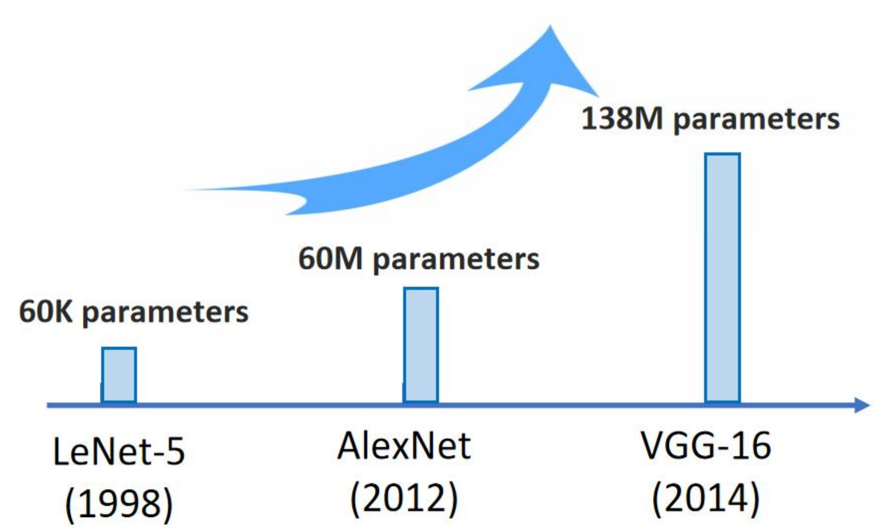
Auto-driving&More

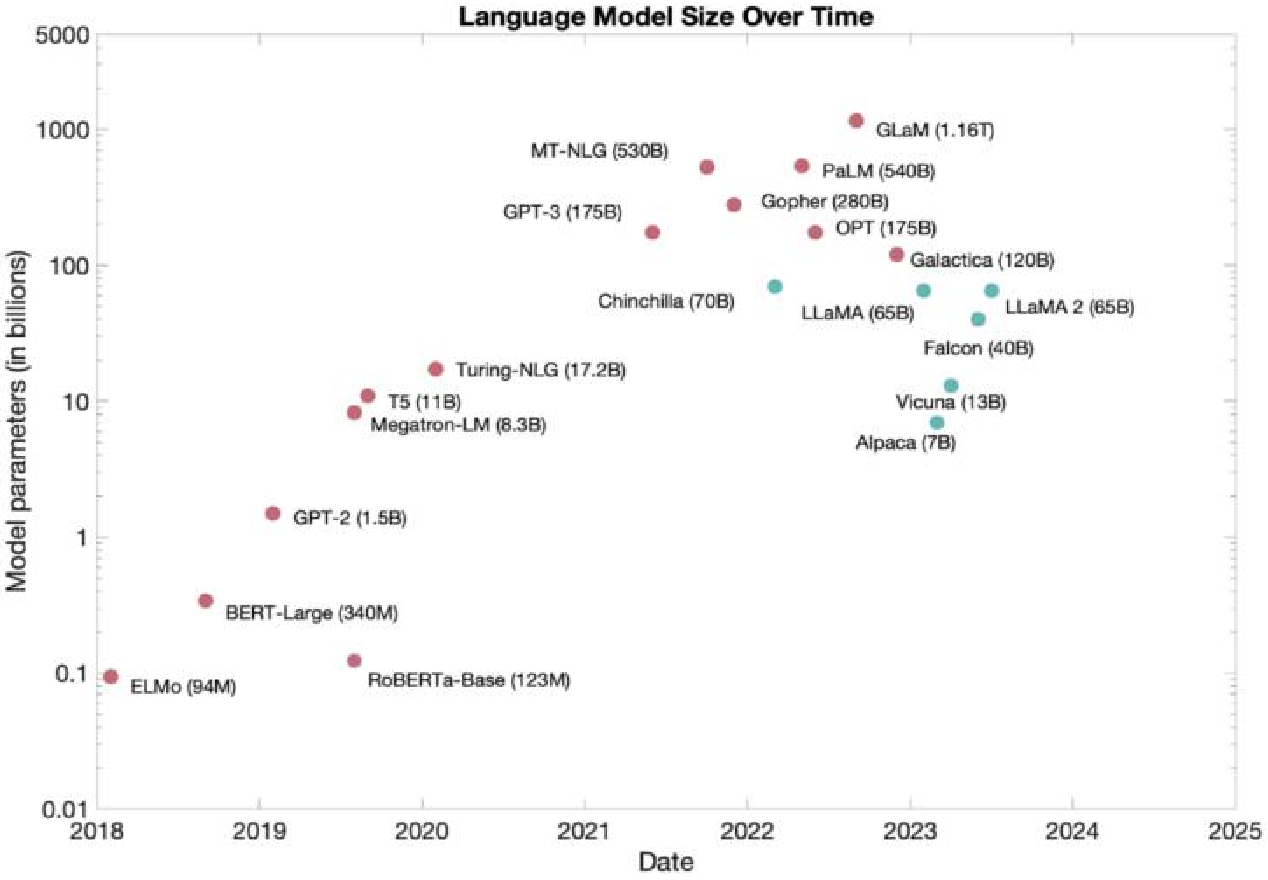
ICML'24, **CVPR'24**
AAAI'24, TIFS'24

②

④ Automatic Training Problem Detection&Repair (ICSE'21)

Model training gets heavier and heavier…

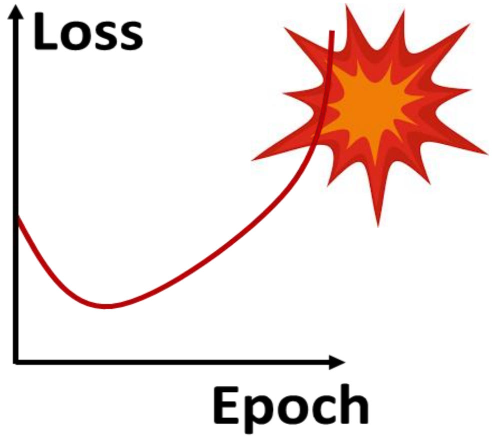


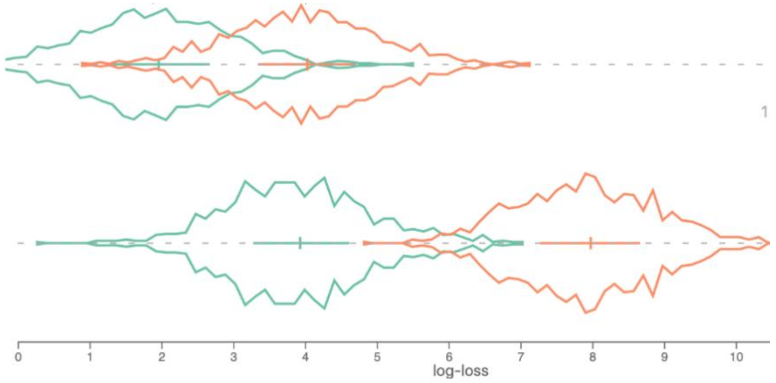


④ Automatic Training Problem Detection&Repair (ICSE'21)

Model training may fail sometimes...

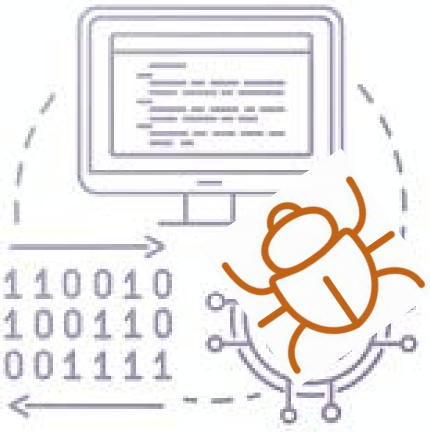
Gradient Explosion



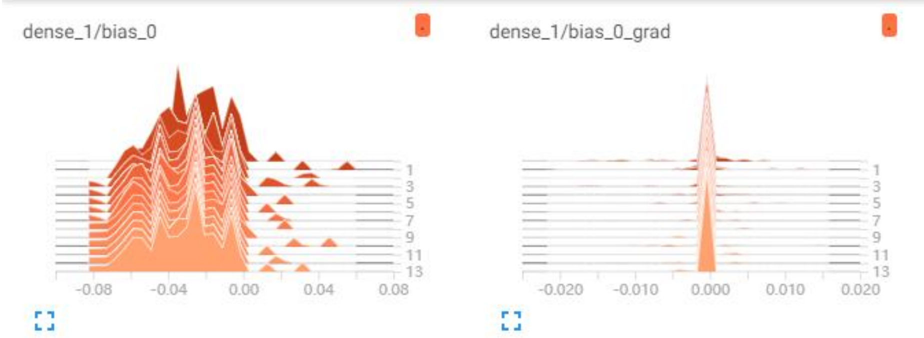


Uber Manifold

Code bugs



- Only **Visualize**
- **Manual** detection
- **Manual** repair



TensorBoard

④ Automatic Training Problem Detection&Repair (ICSE'21)

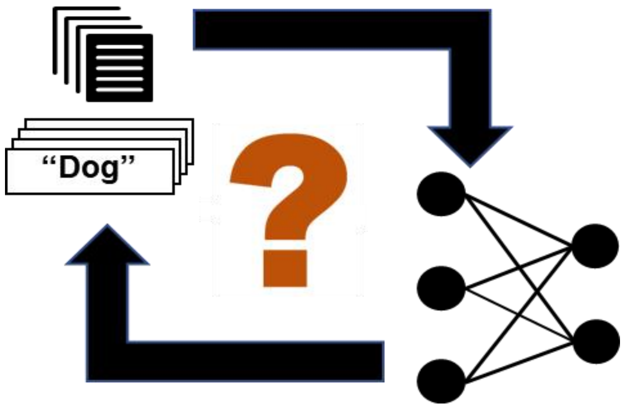
Designing goals

- To **detect** the training bugs in **real time**
  - What is the symptoms of problems?
- To **repair** the buggy model **automatically**
  - Which is the suitable solution?

④ Automatic Training Problem Detection&Repair (ICSE'21)

Being real-time and automatic is necessary because…

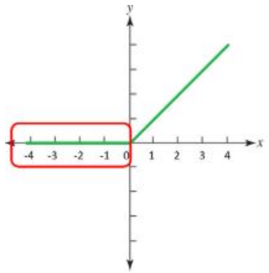
Random Initializer
Shuffled Data
…



Randomness

- **20** layers, **410K** parameters
- **ReLU** activation, **glorot_uniform** initializer, **Adam** optimizer
- **MNIST** dataset, **50** epoch, **100** repeated runs

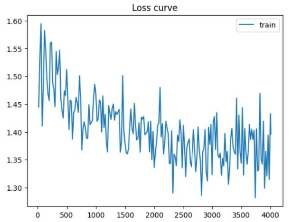
Dying ReLU



Dying ReLU Not Happened:
20 runs Avg ACC: 85.34%

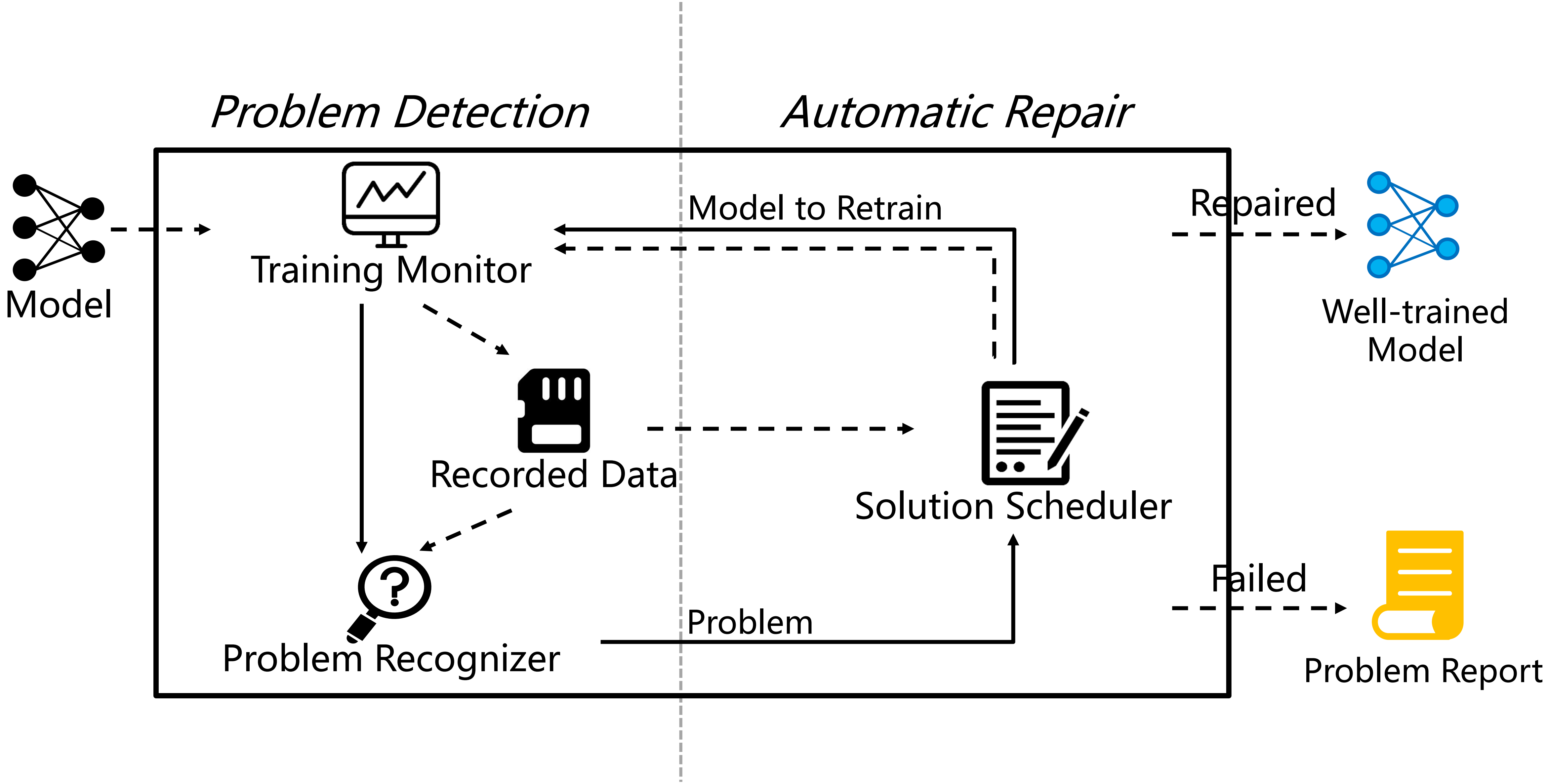
Dying ReLU Happened:
80 runs Avg ACC: 11.35%

Oscillating Loss



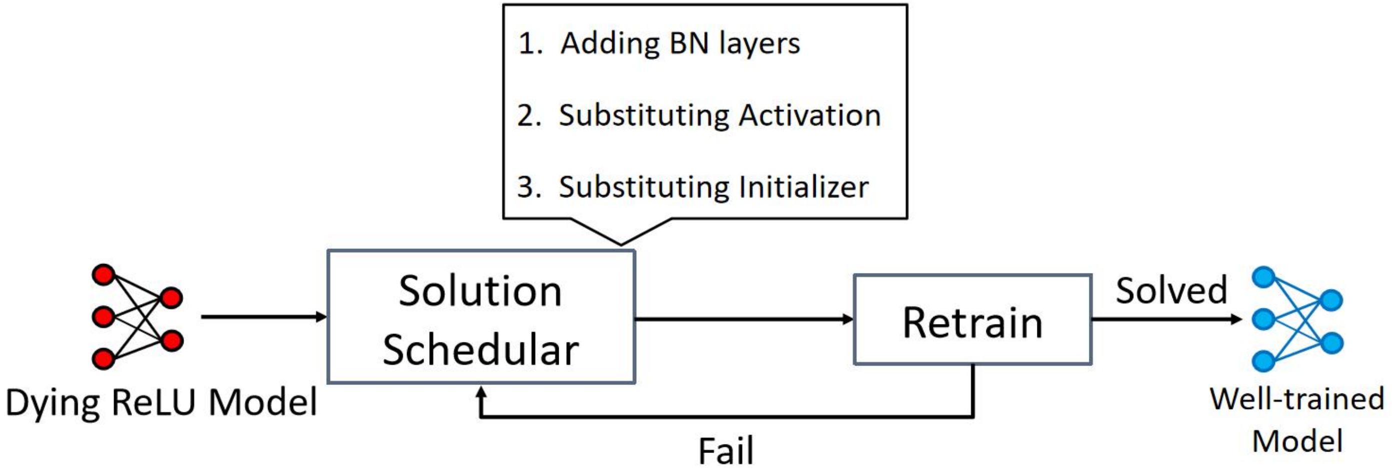
0~9:	50 runs	Avg Acc: 90.36%
10~19:	9 runs	Avg Acc: 89.82%
20~29:	8 runs	Avg Acc: 86.89%
30~49:	4 runs	Avg Acc: 85.99%
No	29 runs	Avg Acc: 90.47%

④ Automatic Training Problem Detection&Repair (ICSE'21)

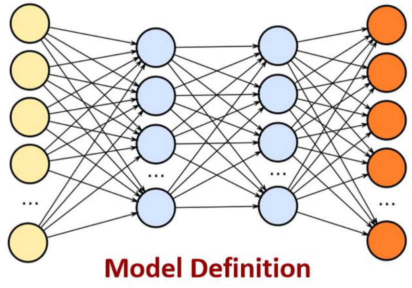


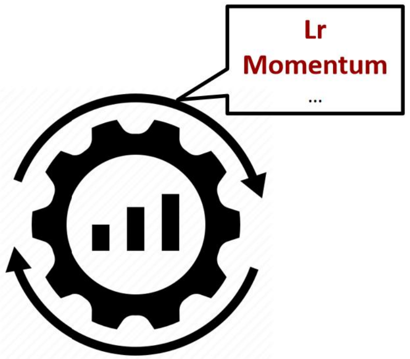
④ Automatic Training Problem Detection&Repair (ICSE'21)

- Analyze recorded data for 5 problems
  - Vanishing Gradient & Exploding Gradient
  - Dying ReLU
  - Oscillating Loss
  - Slow Convergence

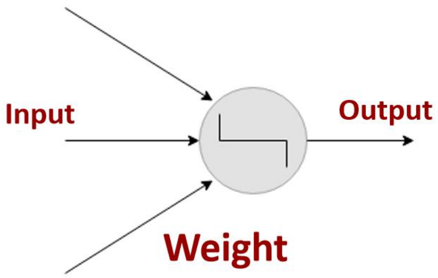


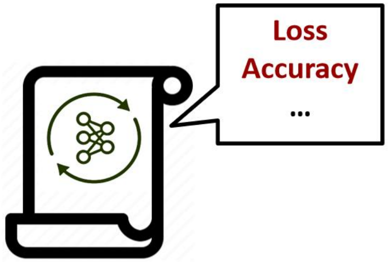
- Static Data





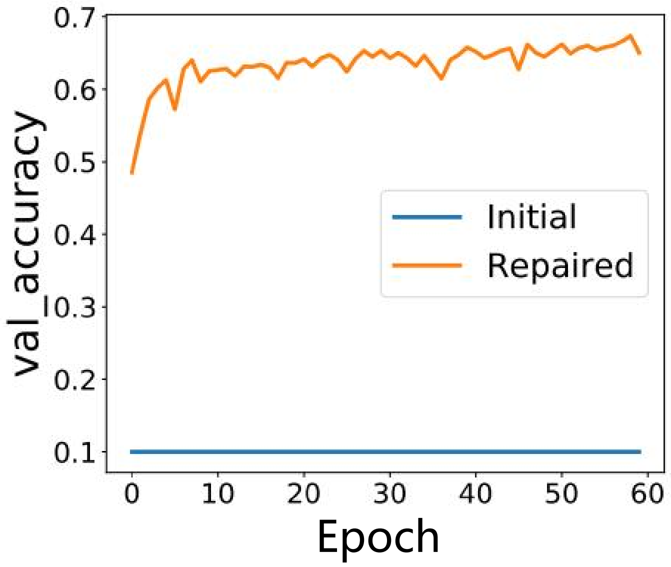
- Runtime Data



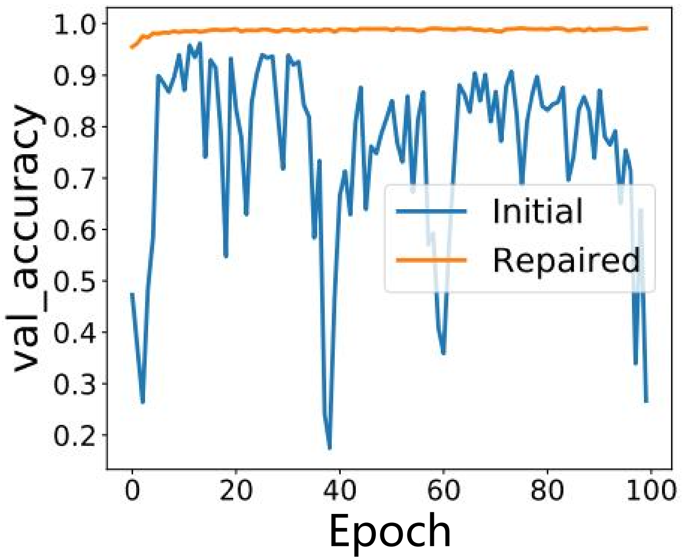


④ Automatic Training Problem Detection&Repair (ICSE'21)

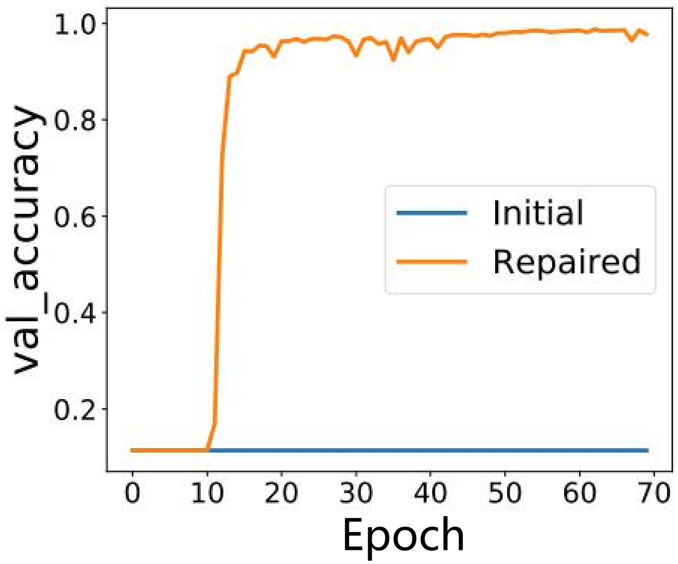
- Detect **316** problems in **262**/495 buggy models on 6 datasets.
- Repair **309** problems with a ratio of **97.78%**.
- Improve average model accuracy by **47.08%.**



Vanishing Gradient Case on CIFAR-10 dataset



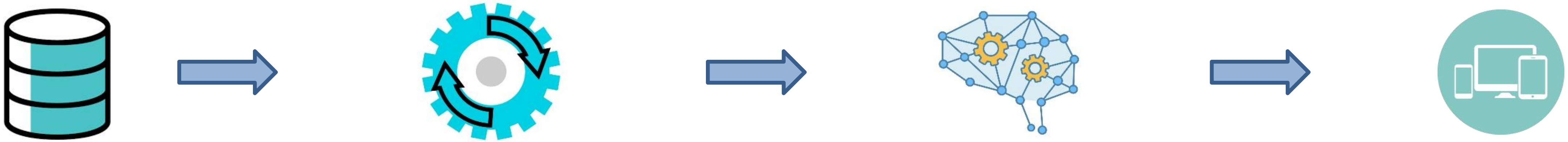
Oscillating Loss Case on MNIST dataset



Dying ReLU Case on MNIST dataset

1.19x more training time on buggy models, **1%** more training on normal models
1% more memory overhead, **1%** more overhead in automatically searching solutions

Security Analysis of ML Lifecycle: Four Studies



Data → Train/Develop → Test/Deploy → Application

Poison&Deepfake	Failures&Bias	OOD&Adv. Example	Auto-driving&More
NDSS'21, **ICML'23** ICLR'23, EMNLP'23 ACL'24, NeurIPS'24 ACL'24, NAACL'24	**ICSE'21**, CCS'22 USENIX'22, NDSS'22 NeurIPS'22, FSE'23 ISSTA'23, ISSTA'24	USENIX'19, CVPR'20 **NeurIPS'21**, USENIX'23 TIFS'23, TIFS'24 FSE'24, AAAI 2025	ICML'24, **CVPR'24** AAAI'24, TIFS'24
③	④	①	②

Awesome-LM-SSP (1300+ items)

Large Model Safety, Security, and Privacy

Stars 1k

A1. Jailbreak

- [2024/11] In-Context Experience Replay Facilitates Safety Red-Teaming of Text-to-Image Diffusion Models Diffusion
- [2024/11] "Moralized" Multi-Step Jailbreak Prompts: Black-Box Testing of Guardrails in Large Language Models for Verbal Attacks LLM
- [2024/11] Preventing Jailbreak Prompts as Malicious Tools for Cybercriminals: A Cyber Defense Perspective LLM
- [2024/11] GASP: Efficient Black-Box Generation of Adversarial Suffixes for Jailbreaking LLMs LLM
- [2024/11] Rapid Response: Mitigating LLM Jailbreaks with a Few Examples LLM Defense
- [2024/11] JailbreakLens: Interpreting Jailbreak Mechanism in the Lens of Representation and Circuit LLM
- [2024/11] SoK: Unifying Cybersecurity and Cybersafety of Multimodal Foundation Models with an Information Theory Approach VLM Survey
- [2024/11] The VLLM Safety Paradox: Dual Ease in Jailbreak Attack and Defense VLM
- [2024/11] SequentialBreak: Large Language Models Can be Fooled by Embedding Jailbreak Prompts into Sequential Prompt Chains LLM
- [2024/11] MRJ-Agent: An Effective Jailbreak Agent for Multi-Round Dialogue LLM Agent
- [2024/11] What Features in Prompts Jailbreak LLMs? Investigating the Mechanisms Behind Attacks LLM
- [2024/11] SQL Injection Jailbreak: a structural disaster of large language models LLM
- [2024/10] Transferable Ensemble Black-box Jailbreak Attacks on Large Language Models LLM
- [2024/10] Effective and Efficient Adversarial Detection for Vision-Language Models via A Single Vector VLM
- [2024/10] RobustKV: Defending Large Language Models against Jailbreak Attacks via KV Eviction LLM Defense
- [2024/10] You Know What I'm Saying: Jailbreak Attack via Implicit Reference LLM
- [2024/10] Adversarial Attacks on Large Language Models Using Regularized Relaxation LLM
- [2024/10] SafeBench: A Safety Evaluation Framework for Multimodal Large Language Models VLM Benchmark
- [2024/10] AdvWeb: Controllable Black-box Attacks on VLM-powered Web Agents VLM Agent
- [2024/10] Feint and Attack: Attention-Based Strategies for Jailbreaking and Protecting LLMs LLM
- [2024/10] Faster-GCG: Efficient Discrete Optimization Jailbreak Attacks against Aligned Large Language Models LLM
- [2024/10] Jailbreaking and Mitigation of Vulnerabilities in Large Language Models LLM
- [2024/10] Refusal-Trained LLMs Are Easily Jailbroken As Browser Agents LLM
- [2024/10] SoK: Prompt Hacking of Large Language Models LLM
- [2024/10] Derail Yourself: Multi-turn LLM Jailbreak Attack through Self-discovered Clues LLM
- [2024/10] Deciphering the Chaos: Enhancing Jailbreak Attacks via Adversarial Prompt Translation LLM
- [2024/10] BlackDAN: A Black-Box Multi-Objective Approach for Effective and Contextual Jailbreaking of Large Language Models LLM
- [2024/10] RePD: Defending Jailbreak Attack through a Retrieval-based Prompt Decomposition Process LLM Defense
- [2024/10] AutoDAN-Turbo: A Lifelong Agent for Strategy Self-Exploration to Jailbreak LLMs LLM
- [2024/10] Root Defence Strategies: Ensuring Safety of LLM at the Decoding Level LLM Defense
- [2024/10] Chain-of-Jailbreak Attack for Image Generation Models via Editing Step by Step Diffusion
- [2024/10] Functional Homotopy: Smoothing Discrete Optimization via Continuous Parameters for LLM Jailbreak Attacks LLM
- [2024/10] Harnessing Task Overload for Scalable Jailbreak Attacks on Large Language Models LLM
- [2024/10] FlipAttack: Jailbreak LLMs via Flipping LLM
- [2024/10] Jailbreak Antidote: Runtime Safety-Utility Balance via Sparse Representation Adjustment in Large Language Models LLM
- [2024/10] VLMGuard: Defending VLMs against Malicious Prompts via Unlabeled Data VLM Defense
- [2024/10] Adversarial Suffixes May Be Features Too! LLM
- [2024/09] Multimodal Pragmatic Jailbreak on Text-to-image Models Diffusion
- [2024/09] Read Over the Lines: Attacking LLMs and Toxicity Detection Systems with ASCII Art to Mask Profanity LLM
- [2024/09] RED QUEEN: Safeguarding Large Language Models against Concealed Multi-Turn Jailbreaking LLM
- [2024/09] MoJE: Mixture of Jailbreak Experts, Naive Tabular Classifiers as Guard for Prompt Attacks LLM Defense
- [2024/09] PathSeeker: Exploring LLM Security Vulnerabilities with a Reinforcement Learning-Based Jailbreak Approach LLM
- [2024/09] Effective and Evasive Fuzz Testing-Driven Jailbreaking Attacks against LLMs LLM
- [2024/09] AdaPPA: Adaptive Position Pre-Fill Jailbreak Attack Approach Targeting LLMs LLM
- [2024/09] Unleashing Worms and Extracting Data: Escalating the Outcome of Attacks against RAG-based Inference in Scale and

C2. Copyright

- [2024/11] SoK: Watermarking for AI-Generated Content LLM SoK
- [2024/11] CDI: Copyrighted Data Identification in Diffusion Models Diffusion
- [2024/11] CopyrightMeter: Revisiting Copyright Protection in Text-to-image Models Diffusion
- [2024/11] WaterPark: A Robustness Assessment of Language Model Watermarking LLM
- [2024/11] One Prompt to Verify Your Models: Black-Box Text-to-Image Models Verification via Non-Transferable Adversarial Attacks Diffusion
- [2024/11] Debiasing Watermarks for Large Language Models via Maximal Coupling LLM
- [2024/11] CLUE-MARK: Watermarking Diffusion Models using CLWE Diffusion
- [2024/11] SoK: On the Role and Future of AIGC Watermarking in the Era of Gen-AI LLM
- [2024/11] Conceptwm: A Diffusion Model Watermark for Concept Protection Diffusion
- [2024/11] LLM App Squatting and Cloning LLM
- [2024/11] InvisMark: Invisible and Robust Watermarking for AI-generated Image Provenance LLM
- [2024/11] Watermarking Language Models through Language Models LLM
- [2024/11] Revisiting the Robustness of Watermarking to Paraphrasing Attacks LLM
- [2024/11] ROBIN: Robust and Invisible Watermarks for Diffusion Models with Adversarial Optimization Diffusion
- [2024/10] Embedding Watermarks in Diffusion Process for Model Intellectual Property Protection Diffusion
- [2024/10] Shallow Diffuse: Robust and Invisible Watermarking through Low-Dimensional Subspaces in Diffusion Models Diffusion
- [2024/10] Inevitable Trade-off between Watermark Strength and Speculative Sampling Efficiency for Language Models LLM
- [2024/10] Watermarking Large Language Models and the Generated Content: Opportunities and Challenges LLM
- [2024/10] Robust Watermarking Using Generative Priors Against Image Editing: From Benchmarking to Advances Diffusion
- [2024/10] Provably Robust Watermarks for Open-Source Language Models LLM
- [2024/10] REEF: Representation Encoding Fingerprints for Large Language Models LLM
- [2024/10] CoreGuard: Safeguarding Foundational Capabilities of LLMs Against Model Stealing in Edge Deployment LLM
- [2024/10] NSmark: Null Space Based Black-box Watermarking Defense Framework for Pre-trained Language Models LLM
- [2024/10] UTF:Undertrained Tokens as Fingerprints A Novel Approach to LLM Identification LLM
- [2024/10] FreqMark: Frequency-Based Watermark for Sentence-Level Detection of LLM-Generated Text LLM
- [2024/10] MergePrint: Robust Fingerprinting against Merging Large Language Models LLM
- [2024/10] An undetectable watermark for generative image models Diffusion
- [2024/10] WAPITI: A Watermark for Finetuned Open-Source LLMs LLM
- [2024/10] Signal Watermark on Large Language Models LLM
- [2024/10] Ward: Provable RAG Dataset Inference via LLM Watermarks LLM RAG
- [2024/10] Universally Optimal Watermarking Schemes for LLMs: from Theory to Practice LLM
- [2024/10] Can Watermarked LLMs be Identified by Users via Crafted Prompts? LLM
- [2024/10] A Watermark for Black-Box Language Models LLM
- [2024/10] Optimizing Adaptive Attacks against Content Watermarks for Language Models LLM
- [2024/10] Discovering Clues of Spoofed LM Watermarks LLM
- [2024/09] Dormant: Defending against Pose-driven Human Image Animation Diffusion
- [2024/09] A Certified Robust Watermark For Large Language Models LLM
- [2024/09] Multi-Designated Detector Watermarking for Language Models LLM
- [2024/09] Measuring Copyright Risks of Large Language Model via Partial Information Probing LLM
- [2024/09] Towards Effective User Attribution for Latent Diffusion Models via Watermark-Informed Blending Diffusion
- [2024/09] PersonaMark: Personalized LLM watermarking for model protection and user attribution LLM
- [2024/09] FP-VEC: Fingerprinting Large Language Models via Efficient Vector Addition LLM
- [2024/08] Watermarking Techniques for Large Language Models: A Survey LLM Survey
- [2024/08] MCGMark: An Encodable and Robust Online Watermark for LLM-Generated Malicious Code LLM codeGen

清華大學 TSINGHUA UNIVERSITY 1911

香港科大(广州) HKUST(GZ)

西安交通大學 XI'AN JIAOTONG UNIVERSITY 1896

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY