Learning Data-Free Universal Adversarial Perturbation with Hybrid Priors and Gradient-Guided Sharpness Regularization

Zhi Tan[1], Jiazheng Cui[1], Wenwen Zhang[1], Yiran Liu[2], Guodong Liu[3], Chris Ding[4], Di Ming[1,5]✉

[1] School of Computer Science and Engineering, Chongqing University of Technology, China
[2] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China
[3] Eli Lilly & Company, USA
[4] School of Data Science, The Chinese University of Hong Kong, Shenzhen, China
[5] Chongqing Key Laboratory of Trusted Human-Machine Interaction and Security/Safety for Intelligent Connected Vehicles, China
{tanzhi0815, 12307040316, 0zwwzww}@stu.cqut.edu.cn, liuyiran@bupt.edu.cn
guodong.liu.e@pitt.edu, chrisding@cuhk.edu.cn, diming@cqut.edu.cn

GitHub Project

Introduction

➢ Background

❖ Sample-Specific Adversarial Attack

❑ A perturbation is optimized for a single input $\boldsymbol{x}$ to maximize the model's classification loss:

$$\max_{\boldsymbol{\delta}_{\boldsymbol{x}}} \mathcal{L}\big(f(\boldsymbol{x}+\boldsymbol{\delta}_{\boldsymbol{x}}), y\big) \quad \text{s.t. } \|\boldsymbol{\delta}_{\boldsymbol{x}}\|_\infty \le \epsilon,$$

❖ Universal Adversarial Perturbation

❑ Universal attacks seek a single image-agnostic perturbation $\boldsymbol{\delta}$ that misleads the target model on most inputs by maximizing expected classification loss over the input distribution $\mathcal{X}$:

$$\max_{\boldsymbol{\delta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{X}}\Big[\mathcal{L}\big(f(\boldsymbol{x}+\boldsymbol{\delta}), y\big)\Big] \quad \text{s.t. } \|\boldsymbol{\delta}\|_\infty \le \epsilon,$$

❖ Data-Free Universal Adversarial Perturbation

❑ Generate UAPs without real training images or labels by using synthetic priors. A representative activation-maximization objective is:

$$\max_{\boldsymbol{\delta}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim p}\left[\sum_{i=1}^{L}\phi\Big(\mathcal{A}_i^f(\boldsymbol{x}+\boldsymbol{\delta})\Big)\right] \quad \text{s.t. } \|\boldsymbol{\delta}\|_\infty \le \epsilon,$$

where $\boldsymbol{x}\sim p$ is a synthetic input drawn from a predefined prior. $\mathcal{A}_i^f$ denotes the response of the i-th layer of model f, $\phi(\cdot)$ measures activation magnitude, and L is the total number of layers.

❖ HPSR-UAP

❑ **Goal**: Improve the effectiveness and transferability of data-free UAPs by jointly addressing prior bias and unstable optimization.

❑ **Contributions**:

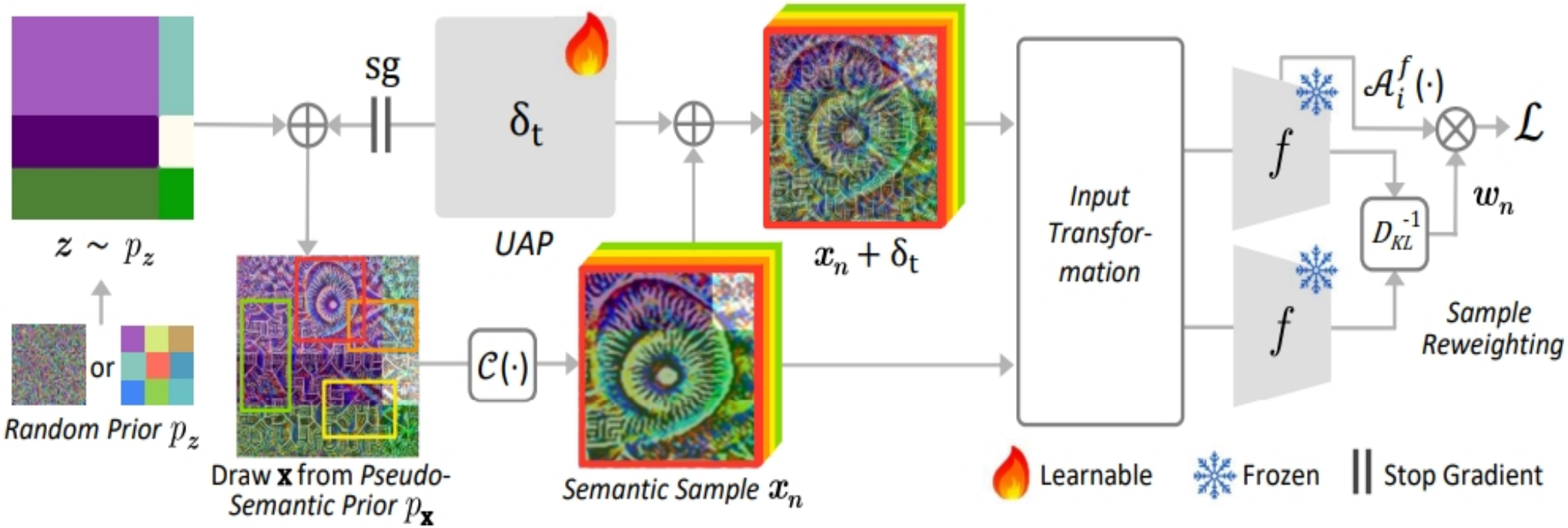
- ❑ Adaptive Hybrid-Prior Reweighting: Combine heterogeneous synthetic priors and dynamically balance their contributions.
- ❑ Gradient-Guided Sharpness Regularization: Penalize local loss variations to favor flatter and more transferable solutions.

Introduction

➢ Background

❖ PSP-UAP[1]-Data-free Universal Adversarial Perturbation with Pseudo-semantic Prior.

❑ PSP-UAP extracts pseudo-semantic priors from the perturbation itself and combines input transformations with hard-sample reweighting to improve the effectiveness and cross-model transferability of data-free universal adversarial perturbations.



Single-prior pseudo-sample generation → Limited coverage of multi-scale and semantic vulnerabilities → Insufficient structural diversity

Convergence to sharp loss regions → Sensitivity to model and input variations → Limited cross-model transferability

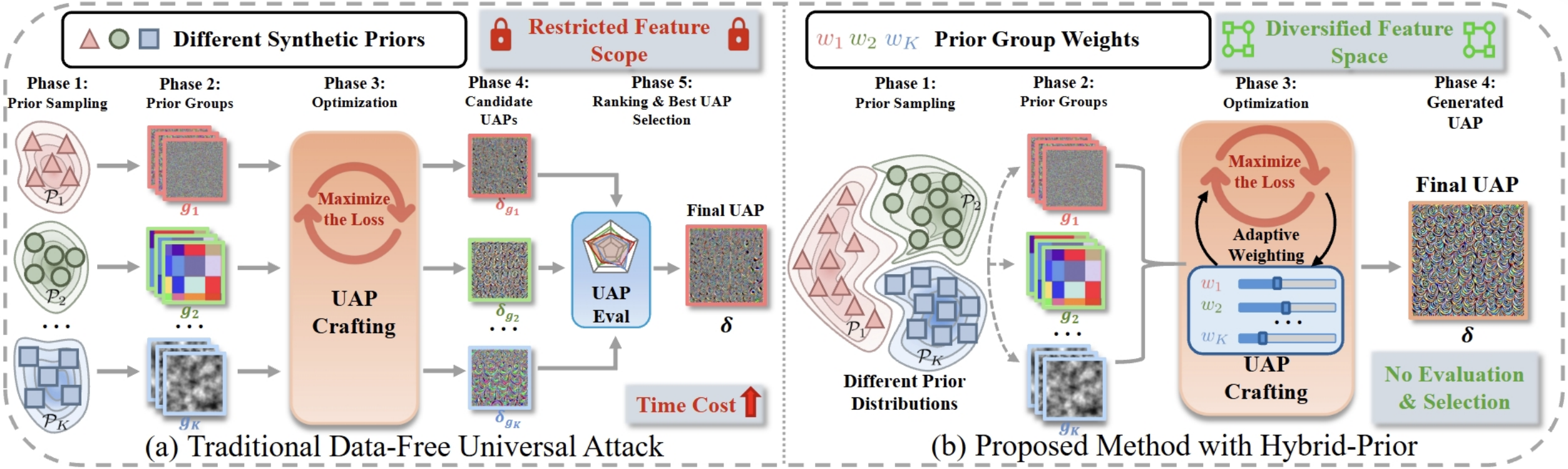
[1] Lee C, Song Y, Son J. Data-free Universal Adversarial Perturbation with Pseudo-semantic Prior[C]//Proceedings of the Computer Vision and Pattern Recognition Conference. 2025: 13907-13916.

Methodology

➢ Single-Prior vs. Hybrid-Prior Data-Free UAPs

❖ Comparison of Single- and Hybrid-Prior Designs.

❑ Conventional data-free UAPs rely on a single prior, limiting feature diversity and producing irregular perturbations. Our hybrid-prior strategy combines multiple priors to explore richer features and generate more structured, repetitive perturbations.



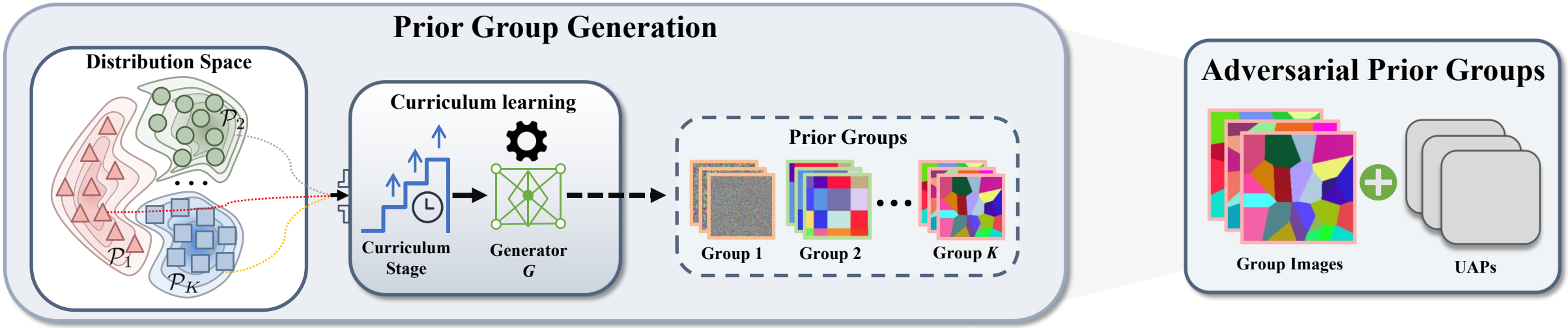
Methodology

➢ From Single-Prior to Hybrid-Prior Training.

❖ Joint Integration of Heterogeneous Priors for Broader Feature-Space Exploration

❑ We jointly optimize UAPs over heterogeneous priors, such as Gaussian, jigsaw, and Perlin, to diversify synthetic inputs and broaden feature-space exploration beyond a single manually selected prior. Let $\{\mathcal{P}_k\}_{k=1}^{K}$ denote prior groups, each containing samples from a distinct generative distribution p_k. We define their joint objective as:

$$\max_{\boldsymbol{\delta}} \mathcal{L}_{\text{act}}(\boldsymbol{\delta}, \{\mathcal{P}_k\}_{k=1}^{K}) = \sum_{k=1}^{K} w_k \sum_{\boldsymbol{x}\in\mathcal{P}_k} \sum_{i=1}^{L} \log \left\| \mathcal{A}_i^f(\boldsymbol{x}+\boldsymbol{\delta}) \right\|_2 \quad \text{s.t. } \|\boldsymbol{\delta}\|_\infty \le \epsilon,$$



Methodology

➢ Discrepancy-Driven Reweighting of Hybrid Priors at Medium Difficulty

❖ Discrepancy Measure for Each Prior Group

❑ For each prior group $\mathcal{P}_k$, we define the discrepancy $D^{(k)}$ between clean and adversarial predictions using a measure $\mathcal{D}$, such as KL divergence or cosine similarity:

$$D^{(k)} = \mathcal{D}(\tilde{\boldsymbol{Z}}_k, \boldsymbol{Z}_k), \quad \boldsymbol{Z}_k = f(\boldsymbol{x}), \quad \tilde{\boldsymbol{Z}}_k = f(\boldsymbol{x} + \boldsymbol{\delta}_t).$$

❖ Problem Formulation

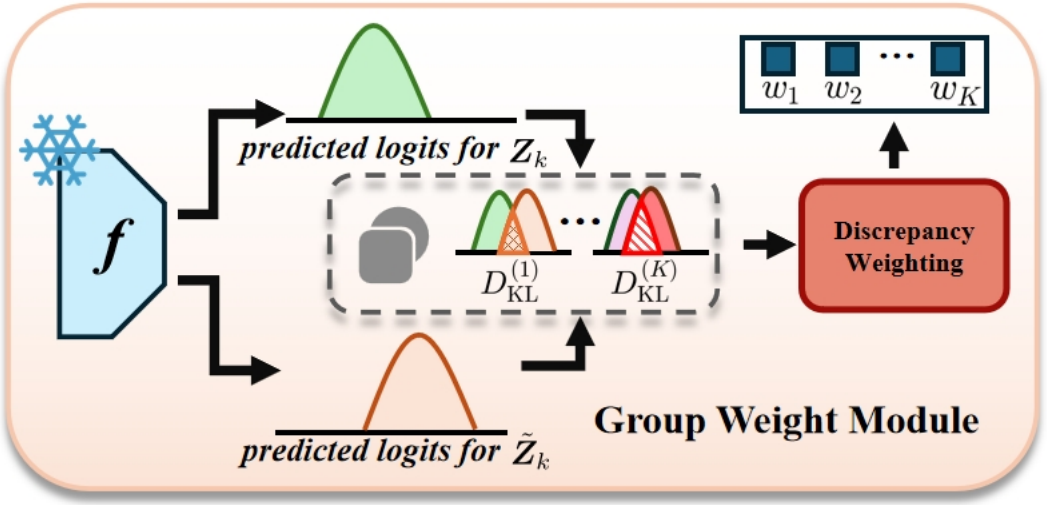
❑ Given a reference difficulty D_{ref} , Prior-group weights are obtained by minimizing:

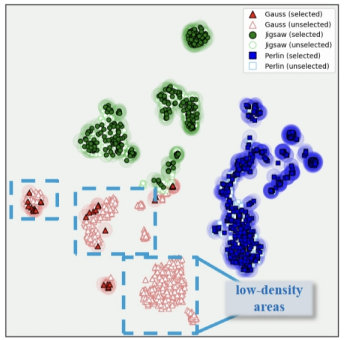
$$\min_{\boldsymbol{w}} \sum_{k=1}^{K} w_k \log(|D^{(k)} - D_{\text{ref}}| + \kappa) + \frac{1}{\alpha} \sum_{k=1}^{K} w_k \log w_k$$

$$\text{s.t. } w_k \geq 0,\ k = 1, \cdots, K, \sum_{k=1}^{K} w_k = 1,$$

❑ The optimization yields the following closed-form weights:

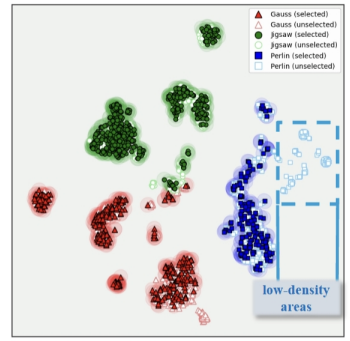
$$w_k = \left(|D^{(k)} - D_{\text{ref}}| + \kappa\right)^{-\alpha} \Big/ \sum_{j=1}^{K} \left(|D^{(j)} - D_{\text{ref}}| + \kappa\right)^{-\alpha}.$$





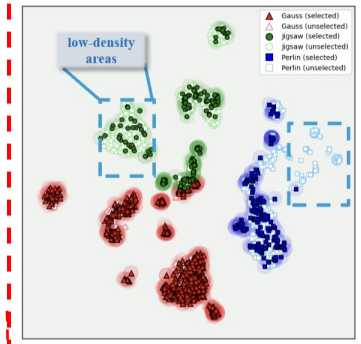
(a) Easy-difficulty distribution

$D_{\text{ref}} = \max_k D_k$



(b) Medium-difficulty distribution

$D_{\text{ref}} = 1/K \sum_{k=1}^{K} D_k$



(c) Hard-difficulty distribution

$D_{\text{ref}} = \min_k D_k$

Methodology

➢ Gradient-Guided Neighborhood Sampling.

❖ Random Noise Sampling

❑ A neighboring perturbation is sampled by adding uniform noise to the current UAP:

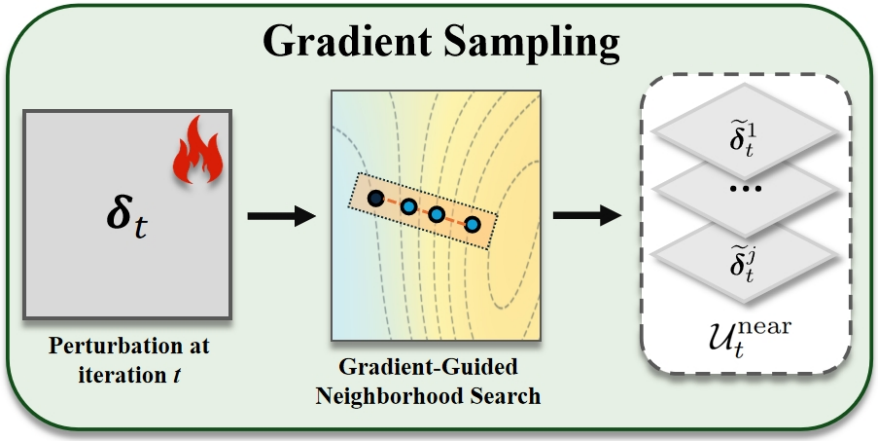
$$\hat{\boldsymbol{\delta}}_t = \boldsymbol{\delta}_t + \boldsymbol{\nu}, \qquad \boldsymbol{\nu} \sim \mathcal{B}(-\zeta, \zeta).$$

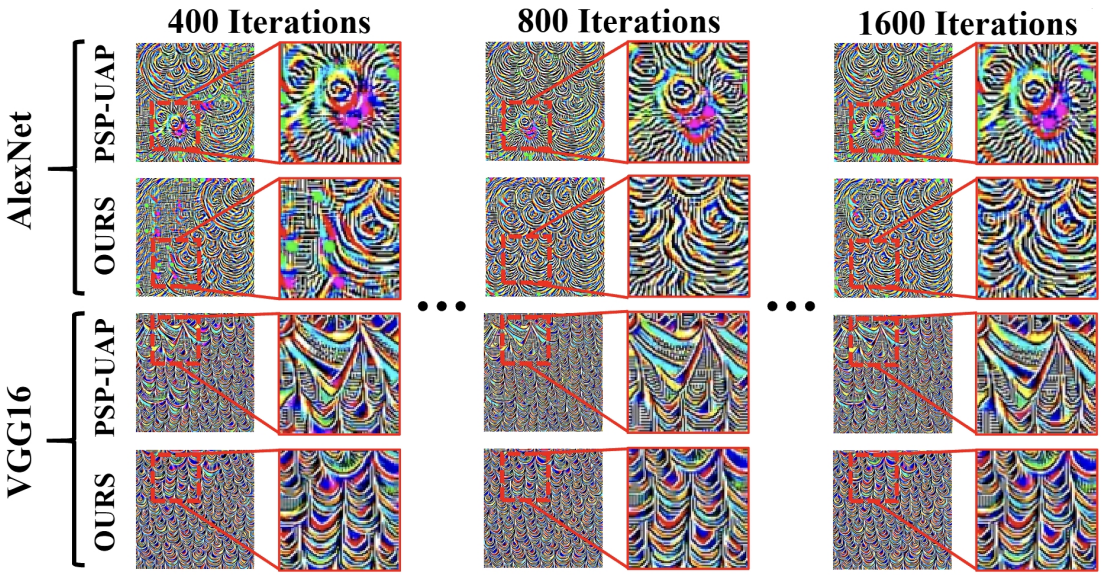
❖ Gradient-Guided Neighborhood Sampling

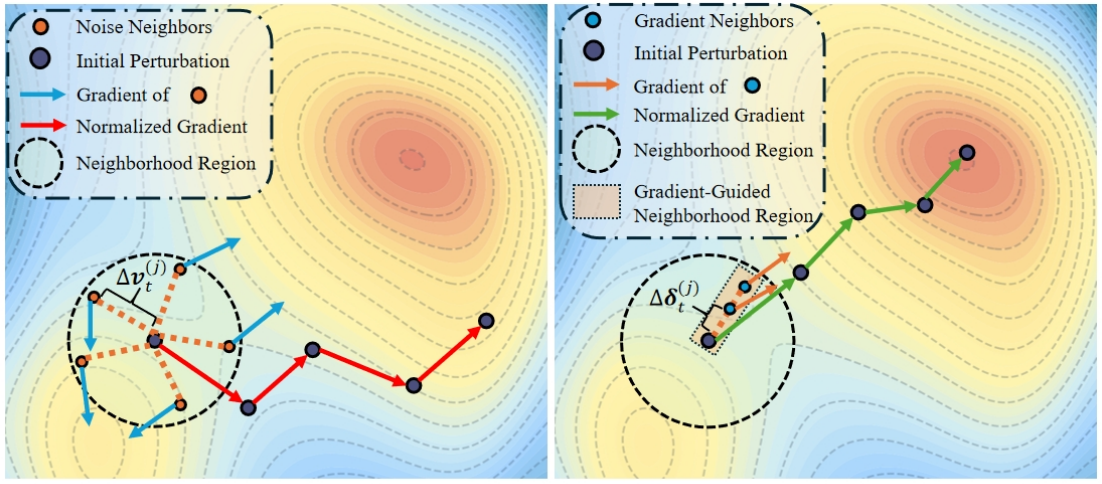
❑ Neighborhood perturbations follow the previous attack direction with varying step sizes, preserving directional consistency while exploring attack-sensitive regions.

$$\mathcal{U}_t^{\text{near}} = \{\boldsymbol{\delta}_t + \Delta\boldsymbol{\delta}_t^{(j)}\}_{j=1}^{M}, \quad \Delta\boldsymbol{\delta}_t^{(j)} = \gamma_j \cdot \text{sign}\big(\nabla\mathcal{L}_{\text{prev}}\big),$$

$$\nabla\mathcal{L}_{\text{prev}} = \nabla_{\boldsymbol{\delta}_{t-1}}\mathcal{L}_{\text{total}}(\boldsymbol{\delta}_{t-1})$$







(a) Random Sampling (b) Gradient-Guided Sampling

Methodology

➢ Gradient-Guided Neighborhood Sampling.

❖ Neighborhood Loss and Sharpness Regularization

❑ The neighborhood loss approximates the local loss landscape by averaging activation losses over $\mathcal{U}_t^{\text{near}}$ with recomputed prior-group weights:

$$\mathcal{L}_{\text{act}}^{\text{near}}(\mathcal{U}_t^{\text{near}}, \{\mathcal{P}_k\}_{k=1}^K) = \frac{1}{|\mathcal{U}_t^{\text{near}}|} \sum_{\widetilde{\boldsymbol{\delta}}_t \in \mathcal{U}_t^{\text{near}}} \left(\sum_{k=1}^K w_k^{\text{near}} \sum_{\boldsymbol{x} \in \mathcal{P}_k} \sum_{i=1}^L \log \left\| \mathcal{A}_i^f(\boldsymbol{x} + \widetilde{\boldsymbol{\delta}}_t) \right\|_2 \right)$$

❖ Overall Training Objective

❑ The final objective combines activation maximization with a sharpness penalty on the discrepancy between current and neighborhood losses:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{act}}\left(\boldsymbol{\delta}, \{\mathcal{P}_k\}_{k=1}^K\right) - \lambda \left\| \mathcal{L}_{\text{act}}\left(\boldsymbol{\delta}, \{\mathcal{P}_k\}_{k=1}^K\right) - \mathcal{L}_{\text{act}}^{\text{near}}(\mathcal{U}_t^{\text{near}}, \{\mathcal{P}_k\}_{k=1}^K) \right\|_2$$

Algorithm 1 HPSR-UAP

Input: surrogate model f; prior distributions p_k; learning rate η; batch size S; max iterations T; perturbation bound ϵ; sharpness step γ; sharpness weight λ; convergence threshold $F_{\max}$; validation interval H; saturation threshold r.
Output: Universal adversarial perturbation $\boldsymbol{\delta}$

Initialize $\boldsymbol{\delta}_0 \sim \mathcal{B}(-\epsilon, \epsilon)$, $\nabla\mathcal{L}_{\text{prev}} \leftarrow \mathbf{0}$, $t \leftarrow 0$, $F \leftarrow 0$
while $t < T$ **and** $F < F_{\max}$ **do**
  $t \leftarrow t + 1$
  Generate neighborhood $\mathcal{U}_t^{\text{near}}$ via Eq. 7
  Generate K prior groups $\{\mathcal{P}_k\}_{k=1}^K$ by sampling from diverse prior distributions p_k
  Compute group weights w_k, w_k^{near} via Eq. 6
  Compute $\mathcal{L}_{\text{act}}$ and $\mathcal{L}_{\text{act}}^{\text{near}}$ via Eqs. 4 & 8
  Compute total loss $\mathcal{L}_{\text{total}}$ via Eq. 9
  Update $\boldsymbol{\delta}_{t+1} \leftarrow \boldsymbol{\delta}_t + \eta \cdot \nabla_{\boldsymbol{\delta}_t} \mathcal{L}_{\text{total}}$
  Clip $\boldsymbol{\delta}_{t+1} \leftarrow \text{clip}(\boldsymbol{\delta}_{t+1}, -\epsilon, \epsilon)$
  Save $\nabla\mathcal{L}_{\text{total}}$ as $\nabla\mathcal{L}_{\text{prev}}$
  Compute saturation rate $\hat{r}$ and adjust $\boldsymbol{\delta}_{t+1}$ if $\hat{r} > r$
  if $t \bmod H = 0$ **then**
    Evaluate fooling rate F_R and update F if not improved
  end if
end while
return $\boldsymbol{\delta}$

Experiment

CHONGQING UNIVERSITY OF TECHNOLOGY

NEURAL INFORMATION PROCESSING SYSTEMS

➢ Evaluating the Transferability using CNN Surrogate Models

- ❖ **Models:**
  - ❑ AlexNet, VGG16, VGG19, ResNet152, and GoogleNet.
- ❖ **Comparative Methods:**
  - ❑ FFF、AAA、GD-UAP、 PD-UAP、Cosine-UAP、AT-UAP-U、TRM-UAP、PSP-UAP.

Table 1: White-box fooling rates (%) of different universal adversarial attack methods on ImageNet. The best result in each column is highlighted in **bold red**.

Attack	AlexNet	VGG16	VGG19	ResNet152	GoogleNet	Avg.
FFF	80.92	47.10	43.62	–	56.44	–
AAA	89.04	71.59	72.84	60.72	75.28	73.89
GD-UAP	90.47	92.32	90.43	30.09	70.78	74.82
PD-UAP	–	70.69	64.98	46.39	67.12	–
Cosine-UAP	91.07	89.48	86.81	65.35	87.57	84.05
AT-UAP-U	**96.66**	94.50	92.85	73.15	82.60	87.95
TRM-UAP	91.92	94.76	92.03	62.65	76.68	83.61
PSP-UAP	91.84	95.18	94.14	80.45	83.48	89.02
HPSR-UAP(w/GD)	95.47	95.16	89.89	38.30	79.07	79.58
HPSR-UAP(w/TRM)	94.45	94.16	91.27	69.75	82.13	86.35
HPSR-PSP(w/PSP)	93.82	**96.51**	**94.47**	**85.88**	**84.67**	**91.07**



Experiment

CHONGQING UNIVERSITY OF TECHNOLOGY

NEURAL INFORMATION PROCESSING SYSTEMS

➢ Evaluating the Transferability using CNN Surrogate Models

❖ Models:

❑ AlexNet, VGG16, VGG19, ResNet152, and GoogleNet.

❖ Comparative Methods:

❑ FFF、AAA、GD-UAP、 PD-UAP、Cosine-UAP、AT-UAP-U、TRM-UAP、PSP-UAP.

❖ Baseline Methods:

❑ GD-UAP、TRM-UAP、PSP-UAP.

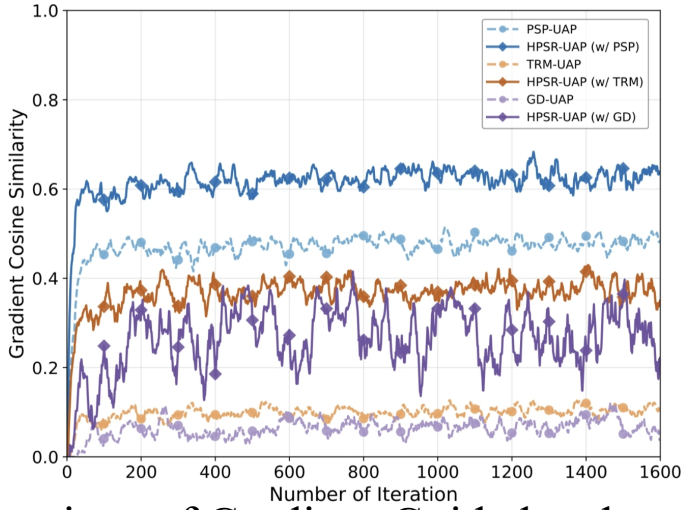
Table 2: Black-box fooling rate comparison between HPSR-integrated baseline methods (GD-UAP, TRM-UAP, PSP-UAP) and existing data-free UAPs. The best results are highlighted in bold white. Darker colors indicate a higher fooling rate. **Avg.bb** reports the average fooling rate over black-box models only, excluding the white-box model.

Model	Attack	AlexNet	VGG16	VGG19	ResNet152	GoogleNet	Avg.	Avg.bb
AlexNet	GD-UAP	90.47±1.03	58.47±1.78	55.76±2.74	31.71±4.04	49.19±5.84	57.12	48.78
	TRM-UAP	91.92±0.81	68.47±1.56	65.83±1.65	41.98±2.72	56.39±3.19	64.92	58.17
	PSP-UAP	91.84±0.45	73.92±1.56	71.73±1.53	47.58±1.14	63.52±1.84	69.72	64.19
	HPSR-UAP (w/GD)	**95.47±0.39**	64.95±0.90	64.30±0.85	41.47±1.52	60.93±1.33	65.42	57.91
	HPSR-UAP (w/TRM)	94.45±0.71	68.73±1.09	66.94±0.87	43.09±0.92	60.72±0.82	66.79	59.87
	HPSR-UAP (w/PSP)	93.82±0.25	**80.11±0.58**	**75.93±0.65**	**52.53±0.79**	**68.51±0.81**	**74.18**	**69.27**
VGG16	GD-UAP	44.59±0.40	92.32±0.63	86.77±0.86	46.37±1.71	40.05±1.52	62.02	54.45
	TRM-UAP	46.99±0.39	94.76±0.20	90.30±0.26	58.54±0.61	50.00±0.45	68.12	61.46
	PSP-UAP	47.69±0.52	95.18±0.31	91.02±0.62	62.97±2.56	53.31±1.70	70.03	63.75
	HPSR-UAP (w/GD)	47.44±0.40	95.16±0.21	91.37±0.16	67.34±0.62	57.46±0.91	71.75	65.90
	HPSR-UAP (w/TRM)	48.71±0.34	94.16±0.26	89.95±0.29	65.23±0.98	56.46±0.41	70.90	65.09
	HPSR-UAP (w/PSP)	**51.19±0.92**	**96.51±0.20**	**92.69±0.45**	**72.26±1.02**	**64.71±0.53**	**75.47**	**70.21**
VGG19	GD-UAP	45.48±0.53	88.89±0.63	90.43±0.54	45.16±1.32	44.22±1.22	62.84	55.94
	TRM-UAP	46.56±0.78	90.68±0.21	92.03±0.21	49.48±0.78	48.26±0.71	65.40	58.75
	PSP-UAP	46.07±0.73	93.59±0.31	94.14±0.26	64.63±1.38	56.59±0.88	71.00	65.22
	HPSR-UAP (w/GD)	**50.60±0.44**	89.40±0.29	89.89±0.47	50.78±0.71	54.65±0.68	67.06	61.36
	HPSR-UAP (w/TRM)	49.94±0.54	89.72±0.71	91.27±0.68	53.60±0.60	56.41±0.71	68.19	62.42
	HPSR-UAP (w/PSP)	49.80±0.43	93.44±0.17	**94.47±0.16**	62.23±0.36	**61.33±0.83**	**72.25**	**66.70**
ResNet152	GD-UAP	44.27±1.51	39.12±1.82	38.03±1.78	30.09±3.89	28.29±1.62	35.96	37.43
	TRM-UAP	45.17±0.65	72.53±1.26	69.85±1.20	62.65±1.36	52.83±1.58	60.61	60.10
	PSP-UAP	55.89±1.49	84.34±0.86	84.86±0.90	80.45±1.64	70.52±1.09	75.21	73.90
	HPSR-UAP (w/GD)	43.14±0.84	39.38±1.19	39.70±1.69	38.30±5.41	31.61±2.24	38.43	38.46
	HPSR-UAP (w/TRM)	55.74±0.68	79.17±0.34	76.34±0.40	69.75±0.57	58.22±0.53	67.84	67.37
	HPSR-UAP (w/PSP)	**59.20±0.85**	**87.92±0.84**	**86.36±0.96**	**85.88±1.61**	**72.51±0.95**	**78.37**	**76.50**
GoogleNet	GD-UAP	58.16±0.81	67.09±1.16	66.21±1.11	43.42±1.14	70.78±0.88	61.13	58.72
	TRM-UAP	57.22±1.13	73.93±0.68	73.61±0.85	52.28±0.79	76.68±0.85	66.74	64.26
	PSP-UAP	63.81±0.62	79.11±0.41	79.84±0.44	57.93±0.54	83.48±0.43	72.83	70.17
	HPSR-UAP (w/GD)	58.49±0.59	71.39±0.86	71.18±1.05	49.50±1.14	79.07±0.68	65.93	62.64
	HPSR-UAP (w/TRM)	60.47±0.51	76.74±0.52	77.55±0.53	55.83±0.46	82.13±0.34	70.54	67.65
	HPSR-UAP (w/PSP)	**65.41±0.44**	**80.41±0.23**	**80.69±0.33**	**59.90±0.59**	**84.67±0.49**	**74.22**	**71.60**

Experiment

➢ Ablation and Extension Studies on HPSR-UAP

❑ Gradient direction consistency across training iterations



❑ Effect of the number of priors.

Attack	Method	AlexNet	VGG16	VGG19	ResNet152	GoogleNet	Avg
AlexNet	HP-UAP (w/GD)	**95.50**	64.32	63.35	41.09	60.13	64.88
	HP-UAP (w/TRM)	94.10	65.62	64.41	41.23	59.29	64.93
	HP-UAP (w/PSP)	93.99	**76.46**	73.34	51.18	67.43	72.48
	HMP-UAP (w/GD)	95.38	63.80	63.44	40.74	59.91	64.65
	HMP-UAP (w/TRM)	92.87	65.87	63.77	37.65	54.09	62.85
	HMP-UAP (w/PSP)	95.30	76.23	**73.79**	**53.47**	**69.34**	**73.63**
VGG16	HP-UAP (w/GD)	48.27	94.85	90.70	65.41	55.81	71.01
	HP-UAP (w/TRM)	48.12	94.12	89.83	63.95	53.88	69.98
	HP-UAP (w/PSP)	49.76	**96.33**	**92.47**	70.71	62.90	74.43
	HMP-UAP (w/GD)	48.50	94.43	90.65	64.87	56.09	70.91
	HMP-UAP (w/TRM)	48.95	94.20	90.07	63.84	55.71	70.55
	HMP-UAP (w/PSP)	**51.35**	96.13	92.24	**71.43**	**64.75**	**75.18**
ResNet152	HP-UAP (w/GD)	41.89	40.72	39.34	35.87	28.25	37.21
	HP-UAP (w/TRM)	53.02	78.81	76.28	70.46	56.83	67.08
	HP-UAP (w/PSP)	58.18	87.03	84.95	82.94	70.65	76.75
	HMP-UAP (w/GD)	41.00	40.68	40.63	44.49	32.44	39.85
	HMP-UAP (w/TRM)	52.03	78.82	76.24	70.51	57.80	67.08
	HMP-UAP (w/PSP)	**59.37**	**88.53**	**87.52**	**85.87**	**74.47**	**79.15**

❑ Comparison of Gradient-Guided and Gaussian UAP Neighborhood

Model	Method	Cosine ↑	LPIPS ↓
AlexNet	Gradient-guided	**0.985020**	**0.009152**
	Gaussian (avg)	0.973994	0.014440
VGG16	Gradient-guided	**0.983898**	**0.006135**
	Gaussian (avg)	0.972892	0.008134
VGG19	Gradient-guided	**0.983005**	**0.007803**
	Gaussian (avg)	0.971954	0.009975
ResNet152	Gradient-guided	**0.985020**	**0.009152**
	Gaussian (avg)	0.974003	0.014421

❑ Runtime and GPU Memory Usage of Baseline Methods and HPSR-UAP

Model	GD-UAP / HPSR-GD	TRM-UAP / HPSR-TRM	PSP-UAP / HPSR-PSP	Test Time
AlexNet	202.85 / 266.84	151.74 / 709.66	142.72 / 552.16	190.45
VGG16	174.44 / 402.05	330.97 / 638.83	192.98 / 1574.55	396.72
VGG19	163.21 / 537.99	126.63 / 933.23	1111.54 / 2431.12	437.84
ResNet152	178.46 / 631.78	498.34 / 2474.23	1070.48 / 2284.79	295.23
GoogleNet	179.00 / 537.55	706.51 / 1649.01	332.13 / 1203.52	205.61

Model	GD-UAP / HPSR-GD	TRM-UAP / HPSR-TRM	PSP-UAP / HPSR-PSP
AlexNet	3361 / 3625	3945 / 4301	3385 / 4178
VGG16	4887 / 6029	7501 / 10197	11235 / 22236
VGG19	5003 / 6358	7067 / 10615	12475 / 23525
ResNet152	4137 / 5189	6405 / 8745	11587 / 22757
GoogleNet	2639 / 3797	4019 / 4157	8507 / 15771



重慶理工大學

CHONGQING UNIVERSITY OF TECHNOLOGY



Thank you for listening!