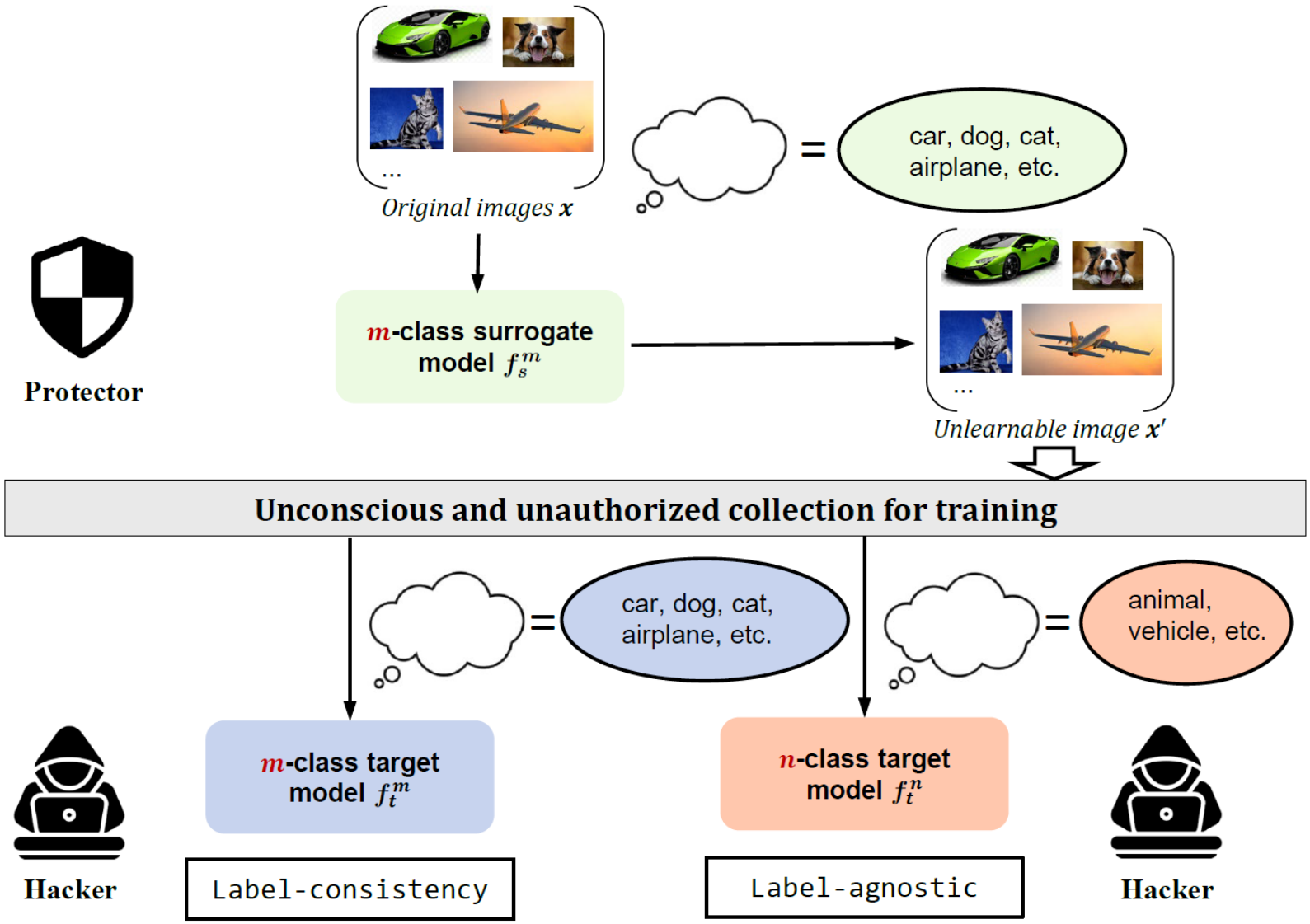
TUE-AM-380

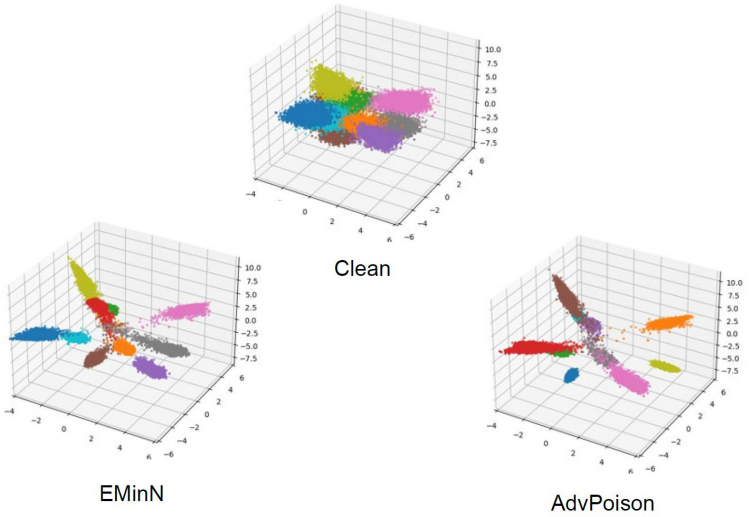
Unlearnable Clusters: Towards Label-agnostic Unlearnable Examples

Jiaming Zhang, Xingjun Ma, Qi Yi, Jitao Sang, Yu-Gang Jiang, Yaowei Wang, Changsheng Xu

label-consistency vs. label-agnostic



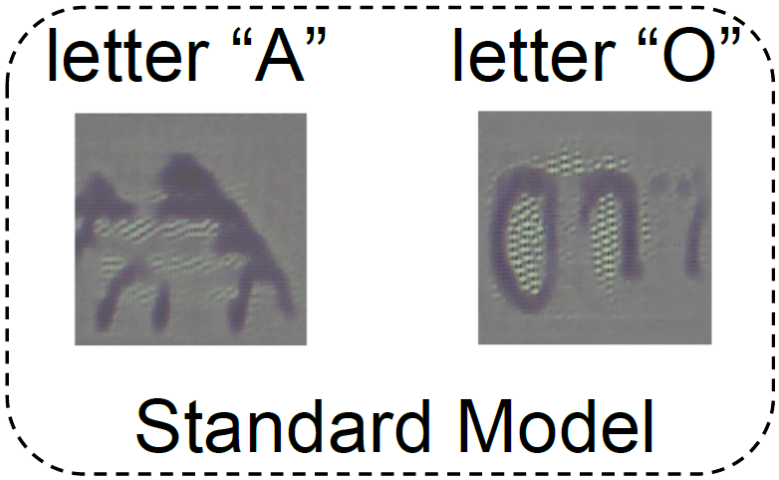
The mechanism of existing UEs



- Error-minimizing noise reduces the training loss of the model to zero to make the model think "there is no more information to learn"[1].
- Adversarial Poisoning uses the concept of non-robust features [2] to make the model to learn the wrong non-robust features [3] .

[1] Unlearnable examples: Making personal data unexploitable, ICLR 2021.
[2] Adversarial Examples Are Not Bugs, They Are Features, NeurIPS 2019.
[3] Adversarial examples make strong poisons, NeurIPS 2021.

Universal Adversarial Perturbation

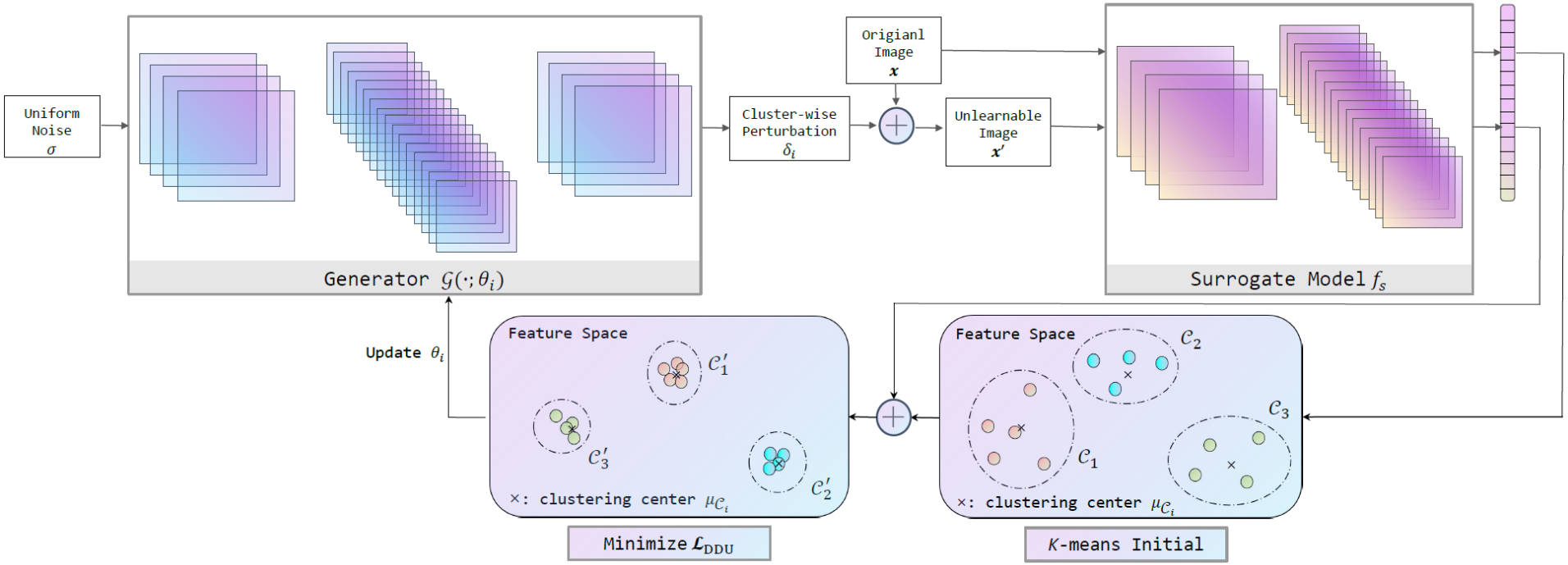


- Universal Adversarial Perturbation (UAP) is a class-wise perturbation that fools the model after being applied to any image [1].
- It can both "overwrite" the original semantic features in the image and work "independently" [2].

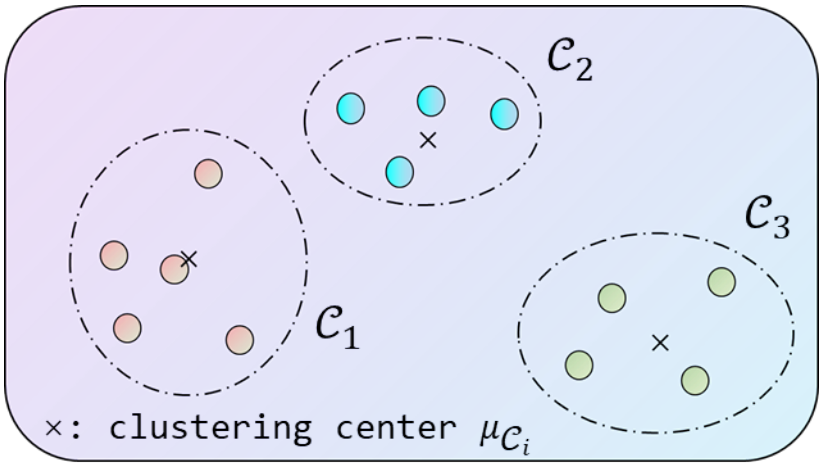
[1] Universal adversarial perturbations, CVPR2017.
[2] ImageNet Pre-training Also Transfers Non-robustness, AAAI2023.

Unlearnable Clusters (UCs)

Achieving breaking uniformity and discrepancy without relying on label information (classification layer).



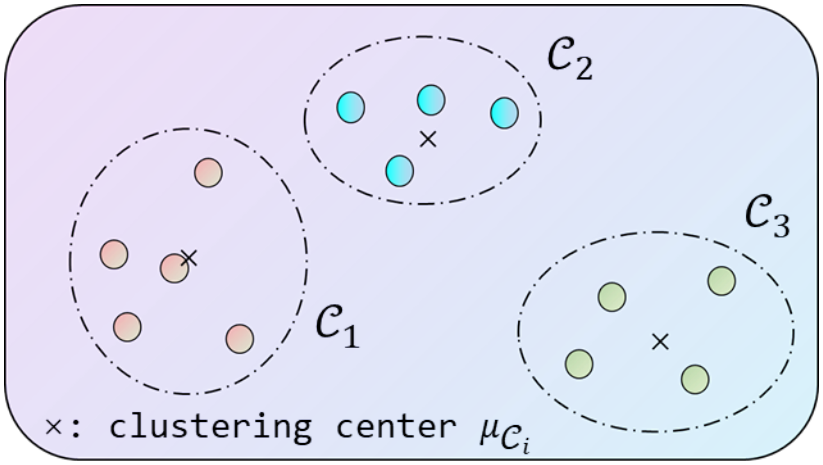
K-means initial

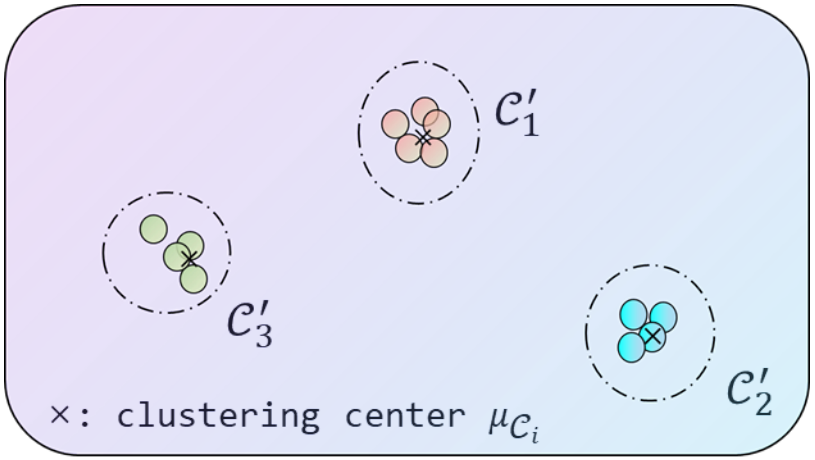


K-means Initial

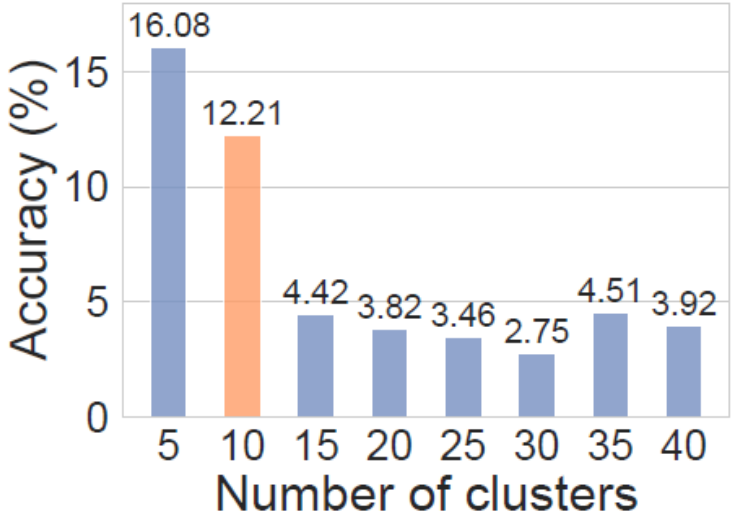
- Given surrogate model f_s, the clean dataset D_c is fed to extract the representation matrix $E = [e_1, \dots, e_k]$, and k。is the number of class.
- K-means is then applied on the representation matrix to detect p number of clusters $C = \{C_1, \dots, C_p\}$, where $\mu_c = \{\mu_{C_1}, \dots, \mu_{C_p}\}$。

Disrupting Discrepancy and Uniformity



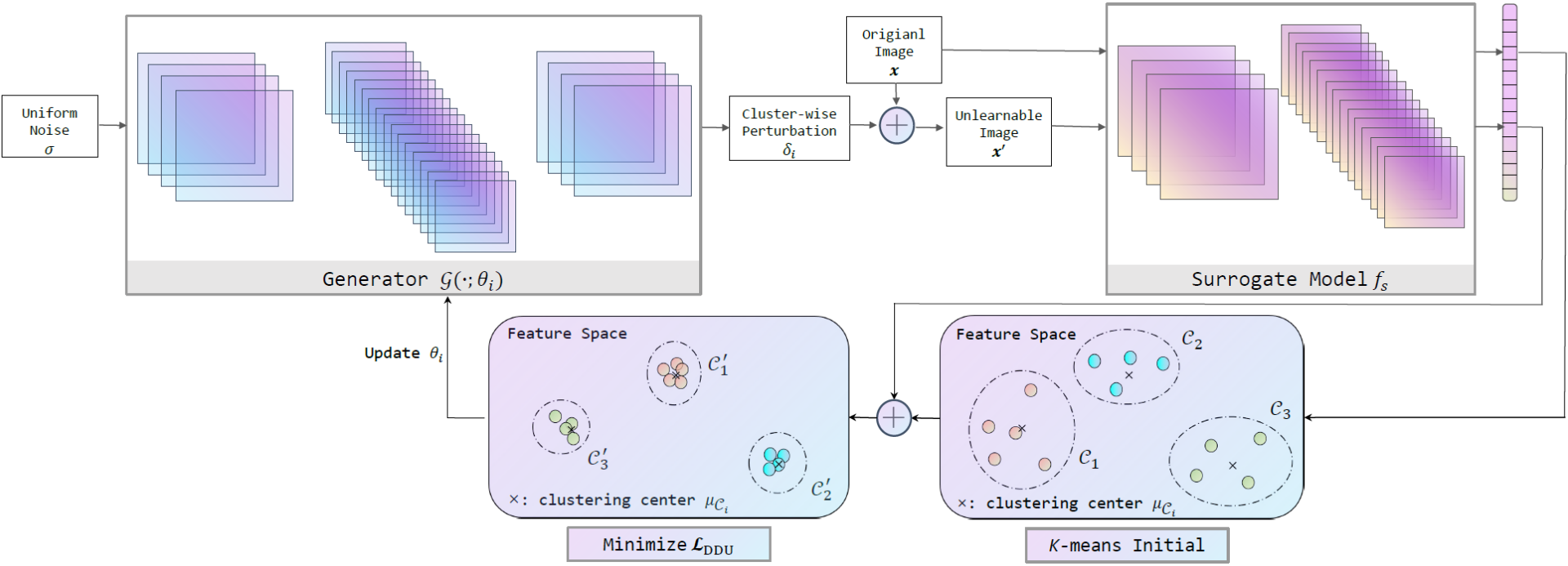


- The clusters are first "aggregated" and then "spun".
- Each cluster has a fixed cluster-wise noise, which means that p noise is generated.



(a) Effect of p on UC

Methodology



$$\theta_i = \arg\min_{\theta_i} \mathcal{L}_{\text{DDU}}(\mathcal{C}_i, g(\mu_{\mathcal{C}_i}), \theta_i)$$

$$= \arg\min_{\theta_i} \sum_{\boldsymbol{x}_{ij} \in \mathcal{C}_i} d(f_s(\boldsymbol{x}_{ij} + \mathcal{G}(\sigma;\theta_i)), g(\mu_{\mathcal{C}_i})),$$

Surrogate model

- The choice of surrogate models is an understudied problem in both unlearnable examples and adversarial examples.
- Representation capability and data coverage capability.
- This paper also explores the use of CLIP as a surrogate model.

Experiments

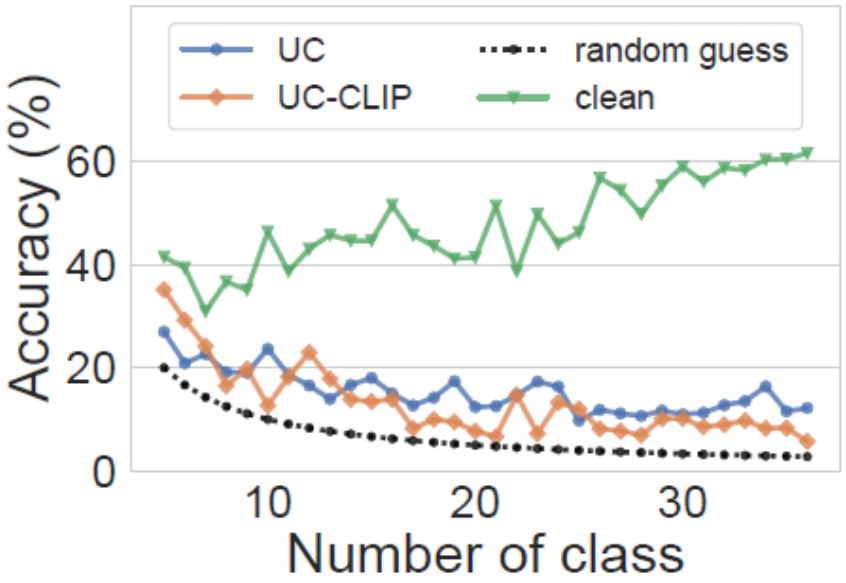
Table 1. The test accuracy (%) of different target models trained on the unlearnable datasets generated by our UC/UC-CLIP and the 5 baseline methods, under the label-agnostic setting. The top-2 best results are highlighted in **bold**.

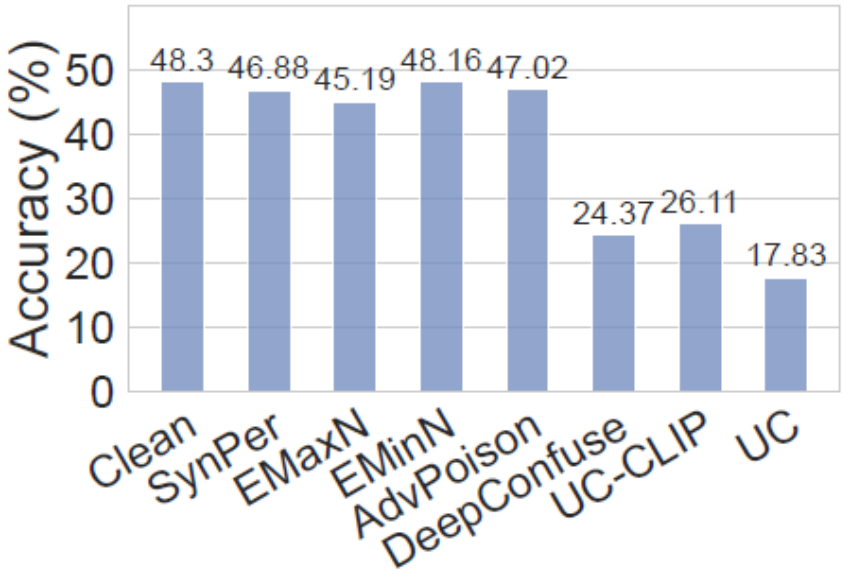
Methods	ResNet-18						EfficientNet-B1						RegNetX-1.6GF					
	Pets	Cars	Flowers	Food	SUN397	ImageNet*	Pets	Cars	Flowers	Food	SUN397	ImageNet*	Pets	Cars	Flowers	Food	SUN397	ImageNet*
Clean	62.31	67.18	67.18	78.97	43.08	77.76	48.68	72.33	52.46	80.29	42.84	78.04	44.86	63.84	52.69	84.02	43.27	80.78
SynPer	52.60	53.50	52.74	74.80	38.26	74.69	28.02	58.34	42.93	74.99	35.92	72.94	34.51	45.54	47.16	77.65	37.78	60.38
EMaxN	54.70	52.95	51.70	73.77	37.57	73.82	33.71	55.64	42.66	74.40	37.30	73.72	34.26	43.40	46.25	78.76	37.82	76.72
EMinN	52.96	54.43	50.58	75.47	38.48	74.20	36.88	54.23	44.06	75.54	37.20	72.20	37.04	39.67	47.34	79.43	36.82	74.86
AdvPoison	50.86	51.91	50.64	75.07	38.51	73.76	37.99	50.08	**41.65**	74.88	36.44	72.54	34.29	46.06	47.41	78.64	36.42	76.32
DeepConfuse	53.72	51.11	50.94	73.13	34.41	55.12	35.54	47.15	43.28	72.91	35.22	45.74	33.71	41.15	46.01	77.26	33.52	49.88
UC (Ours)	**12.21**	**33.57**	**35.55**	**55.29**	**20.38**	**54.80**	**17.06**	**13.92**	42.28	**53.45**	**22.97**	**32.30**	**4.28**	**29.46**	**33.79**	**64.48**	**22.28**	**56.10**
UC-CLIP (Ours)	**4.69**	**4.74**	**10.07**	**19.07**	**3.89**	**39.78**	**6.49**	**15.33**	**14.13**	**17.44**	**12.95**	**31.82**	**3.87**	**4.18**	**8.12**	**26.76**	**6.04**	**41.66**

Table 2. The test accuracy (%) of models trained by `Azure` and `PaddlePaddle` platforms on unlearnable Cars dataset crafted by different methods. The training configuration on the platform was set to "fastest training".

Methods	`Azure`	`PaddlePaddle`
Clean	48.45	83.74
SynPer	42.38	47.59
EMaxN	42.83	42.99
EMinN	44.06	44.40
AdvPoison	43.97	43.38
DeepConfuse	39.47	41.88
UC (RN50)	**36.40**	**30.96**
UC-CLIP (RN50)	**26.97**	**25.79**
UC-CLIP (ViTB32)	**22.47**	**11.49**

Experiments





(a) Different labelings (b) Unsupervised exploitation

Figure 4. (a) The accuracy of ResNet-18 target models trained on the unlearnable Pets dataset but with its labels were re-labeled by the hacker into 5 to 35 classes. (b) Comparison of our approach with the baselines on Pets dataset against ResNet-18 target model trained via self-supervised SimCLR.

Experiments

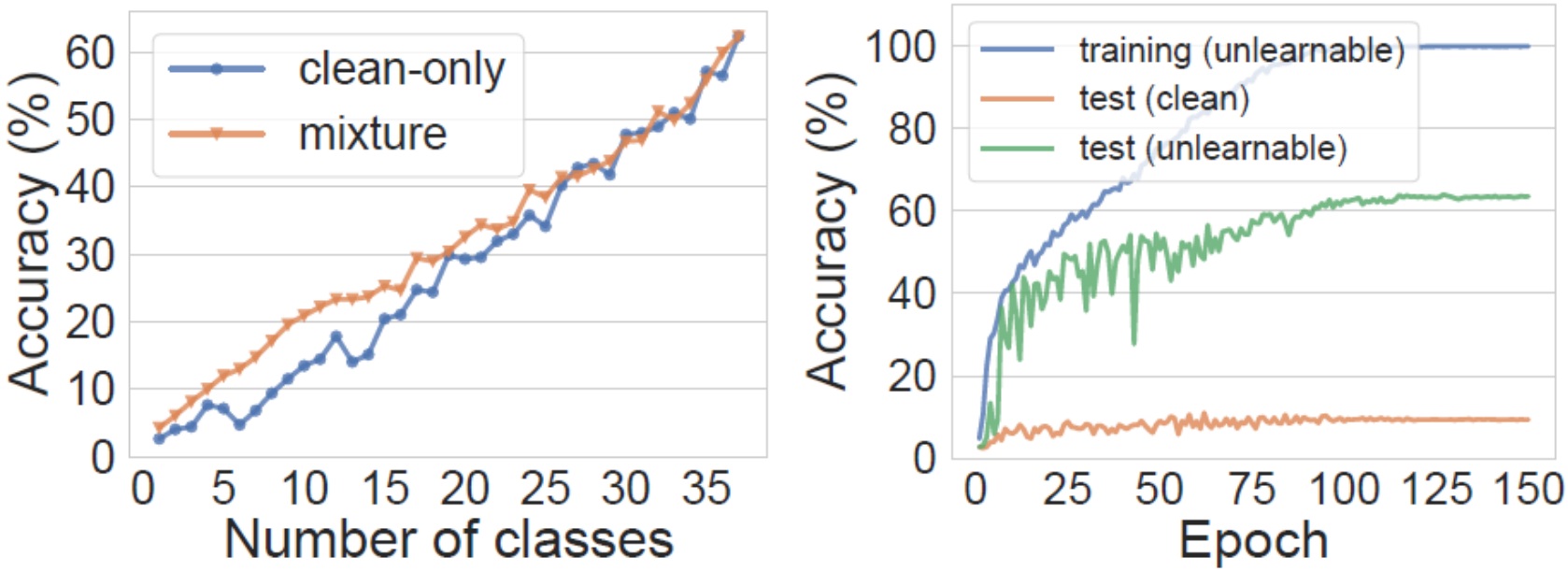
Table 3. The test accuracy (%) of ResNet-18 trained using different defenses against our methods on Pets dataset.

METHODS	NO DEFENSE	MIXUP	GAUSSIAN	CUTMIX	CUTOUT
UC	12.21	14.34	24.26	14.50	12.35
UC-CLIP	4.69	11.96	18.59	6.21	12.29

Table 1. Test accuracy (%) of AT against different protections.

	Clean	SynPer	EMaxN	EMinN	AdvPoison	DeepConfuse	UC	UC-CLIP
ρ=1	58.80	45.88	45.25	43.62	43.48	45.20	22.23	14.04
ρ=2	58.01	48.08	44.08	45.22	42.06	42.94	22.58	16.84

Experiments



(a) Mixture vs. Clean-only (b) Accuracy trends

Figure 6. (a) The test accuracy (%) of ResNet-18 trained on unlearnable-clean mixed vs. clean-only data; and (b) the accuracy trends on clean vs. unlearnable examples. The unlearnable examples are crafted using our UC method on Pets dataset.

Contact me: jiamingzhang@bjtu.edu.cn