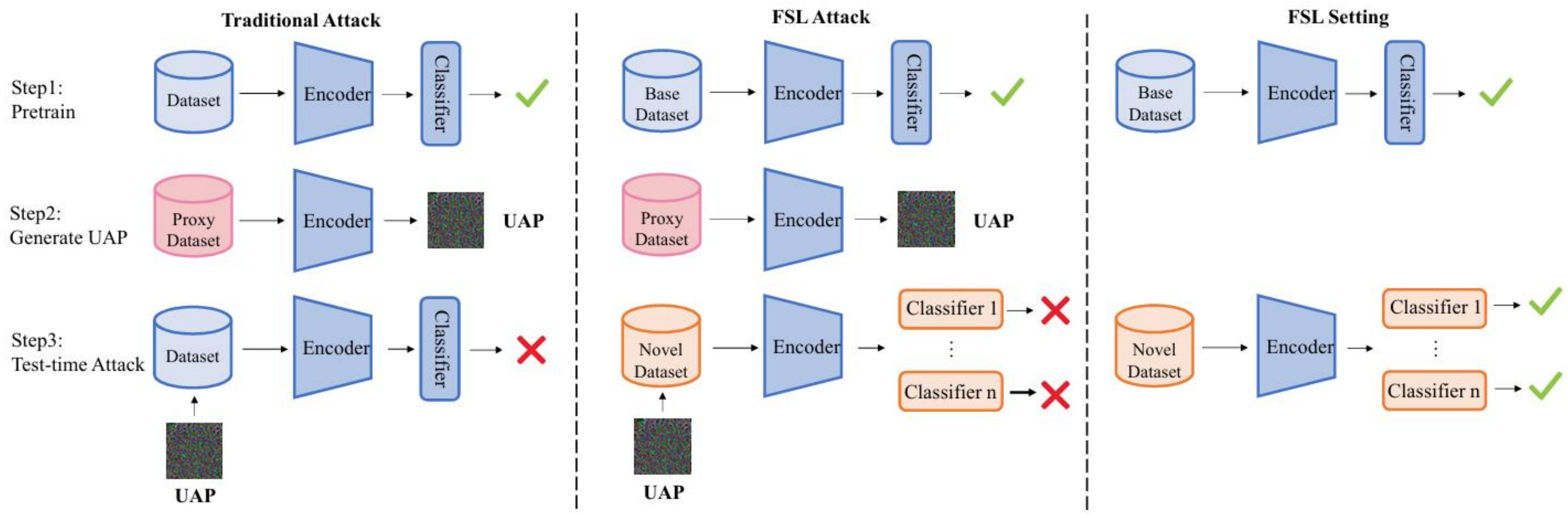
Generate Universal Adversarial Perturbations for Few-Shot Learning

Yiman Hu , YixiongZou, RuixuanLi and Yuhua Li
School of Computer Science and Technology, Huazhong University of Science and Technology
{imane, yixiongz, rxli, idcliyuhua}@hust.edu.cn

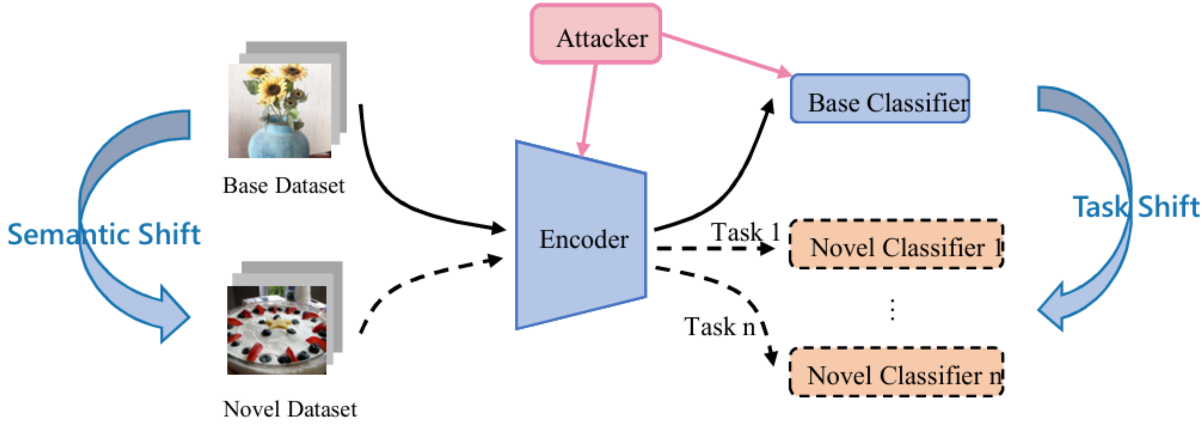
Adversarial Attacks on Few-Shot Tasks

- Setting
  - Few-shot learning
  - Adversarial attacks
- Task
  - Attacking the downstream few-shot tasks without foreseeing them
- Key
  - Generalize the attack ability to downstream tasks



Challenges

- Task Shift and Semantic Shift
  - The two shifts existed in few-shot learning
- Contribution
  - Analyze the presence and impact of the two shifts
  - Build an attack framework and gradually fill up the two shifts to improve attack performance
  - Propose a new standard for studying UAP in FSL scenarios, significantly advancing state-of-the-art methods.



Preliminaries

- Threat Model
  - An attacker aims to create a Universal Adversarial Perturbation (UAP) to attack a pre-trained model and degrade the performance of downstream few-shot tasks.
  - The attacker can not achieve the pre-training and downstream data
- Generate a UAP
  - Train the generator g_θ

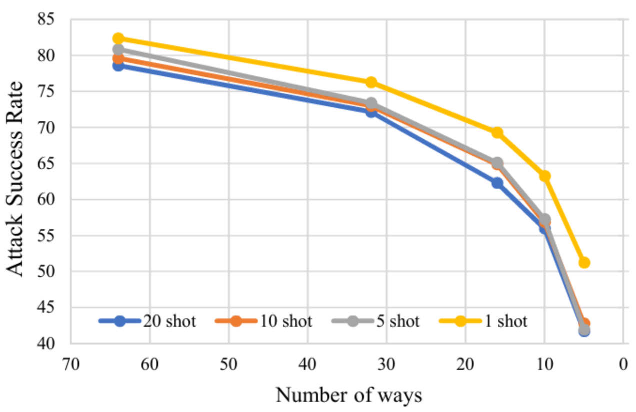
$$L = -H(f(x + g_\theta(z)), y)$$

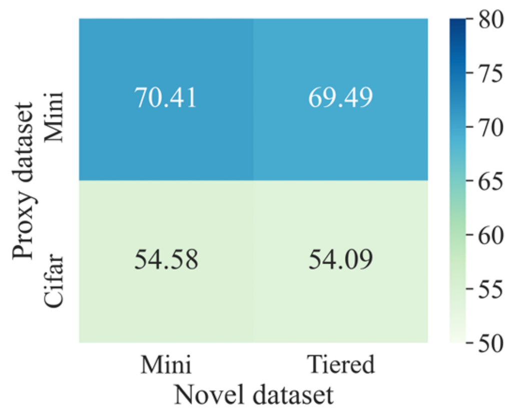
  - Apply the perturbation

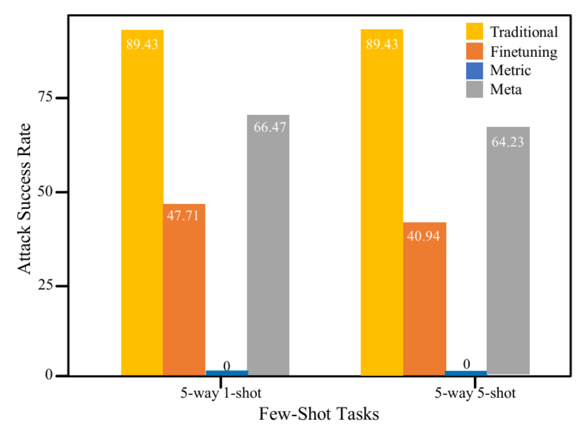
$$x^{adv} = x + g_\theta(z) \;\; s.t. \; \|g_\theta(z)\|_p \leq \epsilon$$

Analysis of the Challenges

- Existence
  - The Attack Success Rate (ASR) decreases when downstream tasks differ from pre-training tasks
  - The Attack Success Rate (ASR) decreases when downstream datasets differ from pre-training datasets
- Impact
  - The two shifts hinder the attack's transferability

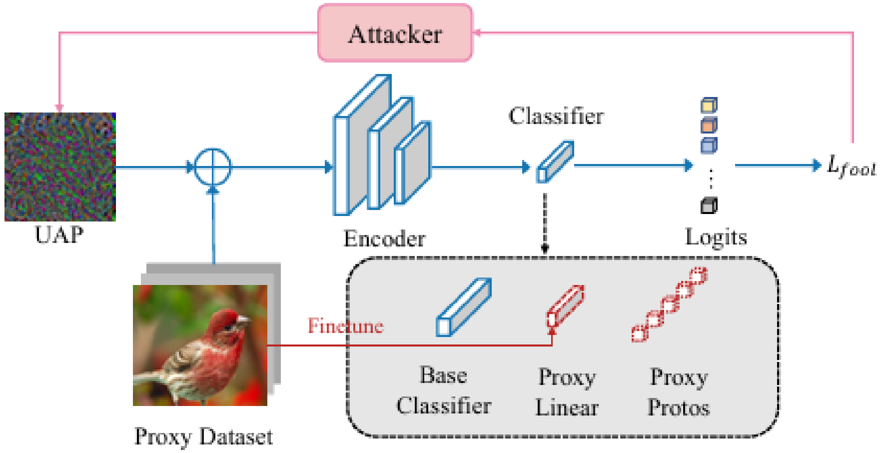






Few-Shot Attacking FrameWork

- Fill up the task shift
  - Align the upstream and downstream tasks during the generation of the UAP
    - Sample 5-way 1-shot tasks on the proxy dataset
    - Generate the UAP based on the proxy tasks
- Fill up the semantic shift
  - Leverage the encoder's generalizability to better transfer
    - Abandon the linear classifier to avoid the influence of the proxy dataset's supervision
    - Construct proxy prototypes to maximize the use of the encoder



(a) An illustration of the attacking framework.

Methods	TS	SS	ASR 1-shot	ASR 5-shot
Base classifier	✗	✗	65.07±0.36	61.73±0.30
Proxy linear	✓	✗	70.87±0.36	68.87±0.27
Base linear	✓	✓	76.75±0.32	74.67±0.21
Proxy protos	✓	✓	**80.04±0.30**	**77.69±0.21**

(b) An illustration of different ASRs.

Experiments

State-of-the-art performance

Table 3: Comparison of different attack methods on ASR for 5-way 1-shot tasks.

Victim	Method	Mark	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	UAN	SPW-18	52.27±0.33	47.68±0.42	43.64±0.26	-	-	-
	GAP	CVPR-18	47.71±0.31	49.40±0.35	66.47±0.32	-	-	-
	AdvEncoder	ICCV-23	76.68±0.31	57.37±0.38	68.51±0.31	59.10±0.29	66.63±0.34	72.85±0.31
	FSAFW	Ours	**81.56**±0.29	**58.94**±0.43	**77.84**±0.28	**70.34**±0.29	**69.03**±0.36	**73.31**±0.32
Tiered	UAN	SPW-18	40.34±0.31	33.00±0.27	51.91±0.28	-	-	-
	GAP	CVPR-18	49.72±0.33	58.23±0.28	61.19±0.30	-	-	-
	AdvEncoder	ICCV-23	75.99±0.32	62.16±0.29	53.82±0.30	71.01±0.31	60.23±0.33	68.86±0.32
	FSAFW	Ours	**76.03**±0.32	**62.56**±0.29	**68.26**±0.28	**76.49**±0.27	**76.73**±0.32	**78.47**±0.34

Table 4: Comparison of different attack methods on ASR for 5-way 5-shot tasks.

Victim	Method	Mark	Baseline	Baseline++	ANIL-1	R2D2-1	ProtoNet	DN4
Mini	UAN	SPW-18	46.09±0.30	45.42±0.30	42.23±0.26	-	-	-
	GAP	CVPR-18	40.94±0.28	45.71±0.29	64.23±0.29	-	-	-
	AdvEncoder	ICCV-23	74.19±0.23	55.12±0.28	67.34±0.26	59.47±0.25	67.76±0.26	74.72±0.21
	FSAFW	Ours	**79.00**±0.18	**63.41**±0.27	**78.31**±0.21	**70.42**±0.22	**67.96**±0.28	**74.88**±0.22
Tiered	UAN	SPW-18	32.97±0.29	23.45±0.23	51.75±0.27	-	-	-
	GAP	CVPR-18	44.90±0.30	52.40±0.28	59.52±0.28	-	-	-
	AdvEncoder	ICCV-23	75.03±0.23	58.23±0.28	53.25±0.28	70.93±0.25	60.40±0.30	69.55±0.22
	FSAFW	Ours	**75.09**±0.21	**59.40**±0.30	**68.96**±0.23	**76.94**±0.19	**78.33**±0.17	**75.91**±0.21

Ablation studies on different shapes of proxy tasks

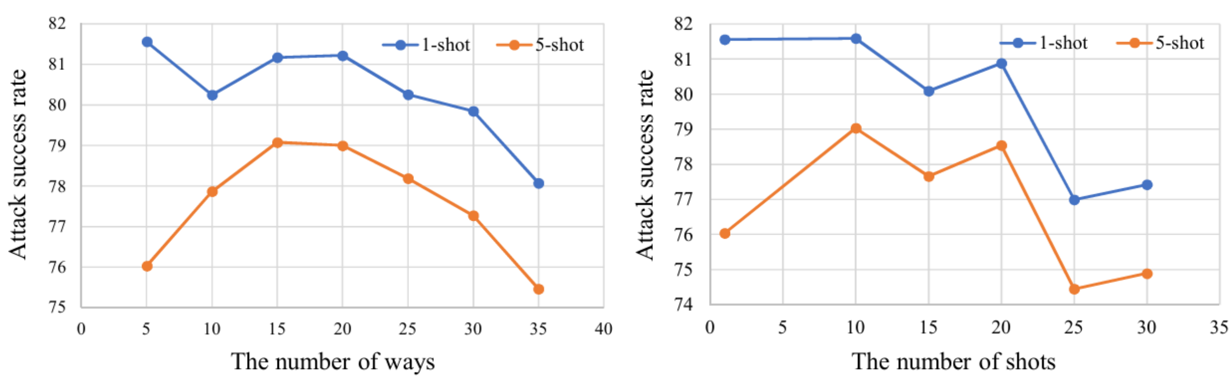


Figure 6: An illustration of the ASR that different forms of proxy tasks bring.

Thanks!