ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. «ΚΥΒΕΡΝΟΑΣΦΑΛΕΙΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ»

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

A Unified Taxonomy of Adversarial Attacks and Defenses in Few-Shot Learning

Άλεξης Λεονταρίδης

ΑΜ: mte24020

Επιβλέπων Καθηγητής : Χρήστος Ξενάκης, Καθηγητής

ΠΕΙΡΑΙΑΣ

Ιούνιος, 2026

ΠΕΡΙΛΗΨΗ:

Η Μάθηση με Λίγα Δείγματα (Few-Shot Learning, FSL) επιτρέπει σε ένα μοντέλο να αναγνωρίζει νέες κλάσεις από ελάχιστα μόνο επισημασμένα παραδείγματα ανά κλάση, αντί για τα μεγάλα σύνολα δεδομένων που χρειάζονται τα παραδοσιακά μοντέλα βαθιάς μάθησης (deep learning). Χρησιμοποιείται ήδη σε τομείς όπως η ιατρική απεικόνιση, η αναγνώριση προσώπου, η ταξινόμηση σπάνιων ειδών και η κυβερνοασφάλεια, όπου η συλλογή πολλών δειγμάτων είναι δύσκολη ή και αδύνατη. Το πρόβλημα είναι ότι τα ίδια στοιχεία που καθιστούν χρήσιμη την FSL, δηλαδή το μικρό σύνολο υποστήριξης (support set), η επεισοδιακή εκπαίδευση (episodic training) και η ταχεία προσαρμογή σε νέες εργασίες, δημιουργούν παράλληλα αδύναμα σημεία που δεν υπάρχουν στα συμβατικά μοντέλα βαθιάς μάθησης. Η ανταγωνιστική μηχανική μάθηση (adversarial machine learning) έχει μελετηθεί εκτενώς, και η ευρωστία των μοντέλων FSL αρχίζει επίσης να προσελκύει περισσότερο ενδιαφέρον· ωστόσο, η υπάρχουσα βιβλιογραφία είναι διάσπαρτη και ανοργάνωτη και δεν υπάρχει ακόμη μια ενιαία ταξινομία αφιερωμένη ειδικά στις επιθέσεις και τις άμυνες στο πλαίσιο της μάθησης με λίγα δείγματα.

Η παρούσα διπλωματική εργασία επιχειρεί να καλύψει αυτό το κενό προτείνοντας μια ενοποιημένη ταξινομία ανταγωνιστικών επιθέσεων και αμυνών για μοντέλα FSL στην όραση υπολογιστών (computer vision). Η ανάλυση βασίζεται σε ένα μοντέλο απειλής τεσσάρων διαστάσεων: την επιφάνεια επίθεσης, τη γνώση που διαθέτει ο αντίπαλος, τον στόχο που επιδιώκει και τη χρονική στιγμή της επίθεσης. Με βάση αυτές τις τέσσερις διαστάσεις, οι επιθέσεις κατηγοριοποιούνται σε έξι οικογένειες (επιθέσεις διαφυγής στο σύνολο ερωτημάτων, δηλητηρίαση του συνόλου υποστήριξης, δηλητηρίαση καθαρής ετικέτας, επιθέσεις κερκόπορτας/backdoor, καθολικές ανταγωνιστικές διαταραχές και επιθέσεις στον χώρο ενσωμάτωσης), ενώ οι άμυνες ομαδοποιούνται σε τέσσερις κατηγορίες (ανταγωνιστική εκπαίδευση, πιστοποιημένες άμυνες, μέθοδοι βασισμένες στην ανίχνευση και άμυνες ειδικά για επιθέσεις backdoor).

Η ταξινομία αναδεικνύει ορισμένα μοτίβα. Σχεδόν κάθε επίθεση προϋποθέτει πρόσβαση τύπου white-box, στοχεύει στην ακεραιότητα του μοντέλου και πραγματοποιείται κατά τον χρόνο εξαγωγής συμπερασμάτων (inference time), ενώ οι επιθέσεις τύπου gray-box και αυστηρού black-box, καθώς και η δηλητηρίαση κατά τη μετα-εκπαίδευση (meta-training), σχεδόν δεν καλύπτονται στη βιβλιογραφία. Οι άμυνες είναι επίσης ανομοιόμορφες: ελάχιστες προστατεύουν το σύνολο υποστήριξης, σχεδόν καμία δεν αντιμετωπίζει τις χειραγωγήσεις στον χώρο ενσωμάτωσης, και δεν υπάρχει ουσιαστική άμυνα έναντι των backdoor επιθέσεων που είναι ειδικές για την FSL. Για να συνδεθεί η ανάλυση με την πράξη, υλοποιείται μια απόδειξη ιδέας (proof-of-concept) στο CIFAR-FS και miniImageNet με Prototypical Networks με δύο backbone, Conv4 και ResNet-12, αξιολογώντας το μοντέλο απέναντι στις επιθέσεις FGSM, Adversarial Support Poisoning (ASP) και FAMF, με και χωρίς Adversarial Querying. Τα αποτελέσματα επιβεβαιώνουν ότι η αδυναμία στα FSL εξαρτάται σε μεγάλο βαθμό τόσο από την οικογένεια της επίθεσης

όσο και από το backbone, και ότι μια άμυνα σχεδιασμένη για την κλασική βαθιά μάθηση δεν μεταφέρεται καθαρά στο πλαίσιο της μάθησης με λίγα δείγματα.

ABSTRACT:

Few-Shot Learning (FSL) allows a model to recognize new classes from only a few labeled examples per class, instead of the large datasets that traditional deep learning models need. It is already used in sensitive areas like medical imaging, face recognition, rare-species classification and cybersecurity, where collecting many samples is difficult or even impossible. The problem is that the same things that make FSL useful, the small support set, the episodic training and the fast adaptation to new tasks, also create weak points that conventional deep learning models do not have to deal with. Adversarial machine learning has been studied a lot, and FSL robustness is also starting to get more attention, but the existing work is spread out and there is still no single taxonomy built specifically for attacks and defenses in the few-shot case.

This thesis tries to cover that missing part by building a unified taxonomy of adversarial attacks and defenses for FSL models in computer vision. Everything is mapped onto a four-dimensional threat model: the attack surface, what the adversary knows, what the adversary wants to achieve, and when the attack takes place. Using these four dimensions, we sort the attacks into six families (query-set evasion, support-set poisoning, clean-label poisoning, backdoor attacks, universal adversarial perturbations and embedding-space attacks) and the defenses into four categories (adversarial training, certified defenses, detection-based methods and backdoor-specific defenses).

The taxonomy makes a few patterns clear. Almost every attack assumes white-box access, targets the model's integrity and happens at inference time, while gray-box and strict black-box attacks, together with poisoning during meta-training, are barely covered in the literature. The defenses are uneven as well: there is little that protects the support set, almost nothing for embedding-space manipulations, and no real defense against FSL-specific backdoors. To connect the analysis to practice, we run a proof-of-concept on CIFAR-FS and miniImageNet with Prototypical Networks under a Conv4 and a ResNet-12 backbone, testing the model against FGSM, Adversarial Support Poisoning (ASP) and FAMF, with and without Adversarial Querying. The results confirm that robustness in FSL depends a lot on both the attack family and the backbone, and that a defense built for standard deep learning does not transfer cleanly to the few-shot setting.

Table of Contents

LIST OF TABLES:

LIST OF FIGURES:

1. INTRODUCTION:

1.1 Motivation:

Few-Shot Learning models have become a significantly important field of research in recent years. These models can classify data using only a handful of labeled examples, often as few as five per class, instead of needing large datasets that traditional deep learning models require. They work by using a small support set of labeled examples and then classifying new data in the query set based on the few examples from the support set, in a way that is similar to human mind in some cases. Few-Shot Learning (FSL) models are already being used in many critical fields such as medical imaging for rare diseases where hospitals only have a small number of labeled samples, rare species classification where only a few images of an animal may exist, robotics, face recognition, and zero-day threat detection. Given how many critical applications depend on these models, protecting them from security threats is not just important, it is necessary.

There has been extensive research on the weaknesses of deep learning models against adversarial perturbations. However, transferring the defenses designed for traditional deep learning models into few-shot learning is not the best practice nor mandatory that it will bring the expected results. FSL models work in fundamentally different ways, through episodic training, learning from very few examples, and meta-learning, introducing new attack surfaces that do not exist in standard models. Applying existing defenses blindly could create security flaws. For example, if an attacker poisons the support set in a medical diagnosis system, it could lead to false classification of a disease, potentially changing the entire class prototype. In a face recognition system, a poisoned support set could allow an unauthorized person to access restricted areas or deny access to the right person. In a threat detection system, adversarial manipulation could cause critical security threats to go unnoticed. A meta-training attack is even more dangerous, because it could affect every future task the model encounters. Not having defenses for the vulnerabilities that exist could lead to misclassification and do serious damage in terms of security integrity and the critical systems that might rely on them.

Despite the growing interest from researchers, scientists, and engineers who are constantly proposing new attacks to expose vulnerabilities and new defenses to increase model robustness, there appears to be a gap in the literature when it comes to a unified taxonomy that systematically explains the FSL pipeline, the unique vulnerabilities it introduces, and the ways to defend against them. This thesis aims to address this gap.

1.2 Contributions:

This thesis focuses specifically on FSL models in computer vision. Adversarial attacks and defenses in NLP or multimodal few-shot settings are outside the scope of this work.

Right now, the existing literature in FSL models is scattered and unorganized. This thesis aims to provide a structured review of the material existing on the field. This work is categorized in the literature of FSL, with emphasis on adversarial attacks and the robust challenges of the models. Specifically, this work detects and analyzes attack surfaces for FSL, such as attacks related to episodic-training, support set, query set and meta-learning. Based on the analysis there will be proposed a unified taxonomy that covers attacks in query set, manipulation of support set, threats that are oriented on the training of the model, backdoor attacks. At the same time, a unified taxonomy for defense and mitigation mechanism will be proposed with defenses like episodic training, support set filtering methods and train techniques aiming to make the model more robust to adversarial attacks.

Lastly, this thesis will provide a discussion of the existing literature, highlighting the main limitations of the approaches, the field fragmentation, open research topics for a better understanding of the security threats and robustness of FSL models. There is also a proof-of-concept experimental evaluation on CIFAR-FS and miniImageNet using ProtoNet model under two different backbones, Conv4 and ResNet-12 showing the different results across three different attack families (FGSM, ASP, FAMF) and how robust the model becomes after being trained with AQ.

2. RELATED WORK:

This study lies between two active research areas: FSL and adversarial attacks on deep learning models. Although a growing number of studies examine aspects of this topic, no work has provided a unified taxonomy that systematically covers both attacks and defenses specifically for FSL models.

2.1 Surveys on Adversarial Attacks on Deep Learning:

The research interest for adversarial attacks on Deep Learning models can be traced back to Szegedy et al. (2013) in their paper “Intriguing Properties of Neural Networks”, where they discovered that deep neural networks are vulnerable to adversarial perturbations. This finding triggered rapid growth in literature and the publication of several surveys [1].

One of the first more in-depth surveys of adversarial attacks in Deep learning models, especially in the field of Computer Vision, was published by Akhtar & Mian (2018) in their paper “Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey”. Their study organizes the attacks in three categories based on the adversary’s knowledge (white-box vs black-box), the target of the attack (targeted vs non-targeted) and the perturbation measure. The authors also categorize the defenses in three categories: the modified training/input, the modified networks, and the usage of network add-ons. However, this survey only focuses on Computer Vision and does not examine execution frameworks besides the conventional supervised models [2].

A follow-up, Survey II (Akhtar & Mian, 2021), extends their earlier review with the advances appearing after 2018, covering more recent attacks and defenses across the same three defense categories, but it remains within standard computer-vision deep learning and does not address the few-shot setting [81].

A Taxonomy was suggested by Yuan et al. (2017) in their work "Adversarial Examples: Attacks and Defenses for Deep learning". This paper taxonomizes the adversarial attacks in three dimensions. The first dimension is a threat model framework subdivided into: adversarial falsification, adversary's knowledge, adversarial specificity, and attack frequency. The second dimension is perturbation covering its scope (universal, individual) and its limitation. The third dimension is benchmark referring to the datasets and the architecture of the models. Compared to Akhtar & Mian, it extends beyond computer vision, covering applications of reinforcement learning, natural language processing and malware detection. However, the threat model focuses exclusively on the testing / inference phase. Attacks that occur during the training time are not included in that paper [3].

A more comprehensive survey is provided by Chakraborty et al. (2018) in "A survey on Adversarial Attacks and Defenses". This paper organizes adversarial attacks based on three criteria: attack surface (evasion attacks, poisoning attacks, exploratory attacks), adversarial capabilities and adversarial goals. It distinguishes between white-box and black-box scenarios, with the black-box being subdivided into non-adaptive, adaptive and strict. When it comes to defenses it examines adversarial training, defensive distillation, feature squeezing and Defense-GAN, and concludes that none of the above is providing a general solution for all types of adversarial attacks [4].

Similar taxonomy efforts have recently appeared for Large Language Models too. A recent SoK by Stylianou et al. (2026) builds a unified taxonomy of 20 threats against Large Language Models, splitting them into training-time, inference-time and system-level, which is very similar lifecycle logic to the one used in this thesis, only applied to LLMs and not in FSL [82]. On the defense side there is also work like Adaptive DeSeTra, a Transformer-based detector that first sorts incoming prompts into benign, jailbreak or prompt injection before they reach the model, and then checks the response to see if the attack actually worked [83]. These are still not about FSL, but they show that organizing attacks and defenses around the model lifecycle is an active direction, while a taxonomy built specifically for few-shot is missing.

These surveys are detailed, but they all target large-data deep learning, not few-shot models. So in the next section we turn to surveys focused on FSL.

2.2 Surveys on Few-Shot Learning models:

At the same time with adversarial attacks surveys, there is also an extensive survey literature that focuses on FSL models. These surveys organize FSL approaches into categories, analyze their architecture and evaluate their performance using benchmark

datasets. However, security and robustness issues are not usually the main focus of their analysis.

Wang et al. (2020) in "Generalizing from a Few Examples: A Survey on Few-Shot Learning" provide one of the most widely cited general FSL surveys, formalizing the field through empirical risk minimization and organizing methods by how they exploit prior knowledge into three categories: data, model and algorithm. The survey discusses future directions across problem setups, techniques, applications and theory, but adversarial robustness is not raised as a concern in any of them, like the other FSL surveys it does not address security [38].

Gharoun et al. (2023) in “Meta-learning approaches for few-shot learning: A survey of recent advances” presented a survey focused specifically on methods of meta-learning for FSL. That study classifies the methods into three basic categories: metric-based, memory-based and learning-based, with them being also divided into initialization learning, parameter learning and optimizer learning. The study provides a detailed overview of architecture such as Prototypical Networks, Matching Networks, Relation Networks, MAML, along with their variations, providing comparative performance tables for benchmark datasets such as Omniglot, miniImageNet and CUB-200-2011. The survey’s mainly focus is on the performance of the models under normal conditions, mentioning briefly that the meta-learning models are not robust in adversarial examples [5].

The previously discussed survey is particularly useful to understand the basic categories of FSL models. However, it is not sufficient for a systematic mapping of the attacks and defenses that concern their security.

In a broader context, Song et al. (2022) suggested a taxonomy of FSL methods based on the challenges they face. The survey distinguishes four broad categories based on prior knowledge: data augmentation, transfer learning, meta-learning and multimodal learning. In addition, the survey lists the applications of FSL in computer vision, including image classification, object detection, semantic segmentation and instance segmentation. However, it does not address robustness or security issues at all. This absence is particularly significant, as this is a broad and influential review of the field, which highlights that security has not yet been integrated as a key taxonomy axis in FSL [7].

Tsoumplekas et al. (2024) published a more comprehensive review, expanding previous taxonomies by integrating more recent approaches such as Neural Processes, hybrid methods (semi-supervised FSL, self-supervised FSL, unsupervised FSL), as well as the recent trends on the field: (cross-domain FSL, in-context learning in LLMs), foundation models, federated few-shot learning and continual learning. The survey also introduces the conversation for Green AI, claiming that FSL can reduce the energy footprint of Artificial Intelligence. Even though this survey is very broad, adversarial robustness is

being treated just as a future work and is not systematically analyzed, reinforcing the view that this research gap remains open [8].

These FSL surveys focus on performance rather than security, so the next section moves to studies that directly examine attacks and defenses in FSL.

2.3 Adversarial Attacks and Defenses in Few-Shot Learning:

Besides the existing surveys on the field, an increasing number of studies specifically examine adversarial robustness on FSL Models. These studies can be broadly organized into four main directions: attacks on query samples, attacks or manipulation of the support set, defense-oriented training approaches, and, more recently, backdoor attacks. However, the studies remain scattered, and none provides a unified framework of adversarial attacks and defenses in FSL.

One of the most systematic studies that was posted on FSL vulnerabilities in adversarial examples was the study of Goldblum et al., "Adversarial Robust Few-Shot Learning: A Meta-Learning Approach" (2020). They prove that naturally trained meta-learners are not robust to attacks like PGD. They suggest an algorithm called Adversarial Querying, that will include adversarial training in the stage of query during fine-tuning. Their analysis mainly focuses on query-time adversarial perturbations instead of support-set manipulation. Their contribution is particularly important because it shows that adversarial vulnerability is a fundamental issue in FSL, making this study a key reference point for the analysis of adversarial robustness in FSL models [6].

Like Goldblum et al. previously, Oldewage et al., in their study "Attacking Few-Shot Classifiers with Adversarial Support Poisoning" (2021), switch the focus from query set to support set. Their study shows that even undetectable perturbations in the support set can drop drastically the accuracy of the Few-Shot classifier, in some cases even below random guessing. The study has a major role because it proves that the support set plays a dual role in FSL: it supports adaptation but also creates a unique attack surface. Contrary to conventional supervised settings, in FSL the support set directly affects the model adaptation process in every new task, which makes it a really important threat for FSL [10].

In continuation of the above type of surveys, Tan et al. (2021) proposed a relatively simple but effective defensive approach against adversarial support sampled for few-shot classifiers. Particularly, this study introduces an attack-agnostic detection method based on self-similarity and filtering of the support set, aiming the detection of adversarial support set. The contribution of this study is extremely important, because it proves that support set is not only a new attack surface, but also a new area for adversarial defense in FSL [11].

Looking at the defense side, Li et al., in their study "Defensive Few-Shot Learning" (2019), address the problem of Defense in Few-Shot Learning models and support that the

existing adversarial defenses cannot be applied to the FSL models. Main reason for this is that because the commonly assumed sample-level distribution between the training and test sets cannot be met in the few-shot setting. They propose an episode-based adversarial training method, in which the model is trained with perturbed few-shot episodes and not isolated samples. They enforce consistency between clean and adversarial episodes. Like the above study, this one has also a major role in FSL because it points out that the existing defenses cannot be transferred to FSL and proposes a defense framework that is better suited to FSL. However, this study's focus is on defense and does not classify attack types. Despite this limitation, the study is particularly important because it moves the discussion of defense from the individual sample level to the episode level, which is more compatible with the nature of FSL [12].

More recently, Dong et al. (2024), proposed a new approach for adversarial robust FSL, based on the idea that the robust similarity learning and the class concept learning are complementary. Their method unifies those models through co-distillation parameter mechanism, aiming to improve robustness against diverse attack strengths. This study shows that the research for robust FSL has improved beyond the initial meta-learning-based defenses and continues to develop defense frameworks that are more complex and diverse [13].

In this next study, Liu et al. "Does Few-shot Learning Suffer from Backdoor Attacks?" (2024), the authors are investigating the backdoor attacks on FSL. They point out that existing backdoor methods do not work in Few-Shot and they propose a new method called FLBA: Few-Shot Learning Backdoor Attack. First, they evaluate few known backdoor methods like BadNet, Blended and Refool. They find that these methods either cause the model to overfit benign features or trigger features or lose stealthiness because it's easier to detect the poisoning in the small dataset. To address the problems mentioned, the authors propose FLBA, which is a four-stage backdoor attack - a white-box attack - that poisons the support set that is used for fine-tuning. FLBA achieves higher attack success rates than the existing backdoor attacks, it works even in 1-shot learning and is resistant to fine-tuning defenses. With this, this study expands the threat landscape of FSL, showing that the attack surface is not limited to adversarial perturbations, but also includes persistent and targeted forms of manipulation, like backdoor attacks [14].

From the previous studies, it can be concluded that FSL models are also vulnerable to security threats, which manifest themselves through attack surfaces related to the support set, query set, and episodic adaptation. However, every study only analyzes one type of attack or defense without having an organized taxonomy that includes both and is categorizing them thoroughly.

2.4 Research Gap and Contribution:

Based on the above, to the best of my knowledge, there is a gap in literature. The existing taxonomies of adversarial attacks and defenses only cover the classic deep learning models without taking into consideration the specific characteristics of the FSL models. The existing surveys for FSL models are very detailed in terms of architecture and performance but do not include the security threats the models have. There is an increasing number of studies that are about adversarial attacks and defenses in FSL models from support set poisoning attacks and query attacks to backdoor attacks and defense frameworks. Every study that is published has its focus on either attack or defense without having a unified taxonomy that includes both cases.

The aim of this thesis is to fill this research gap that exists in the literature by proposing a unified taxonomy from adversarial attacks and defenses specifically to FSL models, by taking into consideration the attack surfaces made by the episodic training, the support set and meta-learning. This will contribute for future scientists to evaluate where they should turn their focus on.

The table below maps all the surveys discussed above against the dimensions this thesis covers, to make the gap easier to see.

Survey / Work	Primary focus	Attacks	Defenses	FSL-specific surfaces	Unified attack + defense taxonomy for FSL
Akhtar & Mian (2018) [2]	Adversarial DL (CV)	✓	✓	✗	✗
Akhtar & Mian — Survey II (2021) [81]	Adversarial DL (CV)	✓	✓	✗	✗
Yuan et al. (2017) [3]	Adversarial DL	✓	✓	✗	✗
Chakraborty et al. (2018) [4]	Adversarial DL	✓	✓	✗	✗
Stylianou et al. (2026) [82]	Adversarial LLM threats	✓	✓	✗	✗
Wang et al. (2020) [38]	FSL	✗	✗	✗	✗
Gharoun et al. (2023) [5]	FSL (meta-learning)	✗	✗	✗	✗
Song et al. (2022) [7]	FSL	✗	✗	✗	✗
Tsoumplekas et al. (2024) [8]	FSL	✗	✗	✗	✗
This thesis	FSL (CV)	✓	✓	✓	✓

Table 1: Comparison of existing surveys against this work.

As shown, the surveys split into two groups. The first group is the adversarial-ML surveys (Akhtar & Mian, Yuan et al., Chakraborty et al.). These cover attacks and defenses in detail, but they are made for standard deep learning and don't deal with the support set, the episodic training or the meta-training as attack surfaces. This is true even for Survey II (2021), the most recent one that reviews more than 3,000 papers, which only mentions few-shot once and just as a single defense. The second group is the FSL surveys (Wang et al., Gharoun et al., Song et al., Tsoumplekas et al.). These analyze the FSL models and their performance well, but they treat adversarial robustness either as a quick mention or as future work, never as a main part of their taxonomy. No survey sits in the middle of the two groups, which is exactly the spot this thesis is trying to fill. The table also includes a recent LLM threat taxonomy (Stylianou et al., 2026), which follows the same lifecycle-based logic but is built for Large Language Models and does not touch the FSL attack surfaces either. This shows that even the newest taxonomy work in adjacent model families leaves the few-shot case uncovered.

3. FEW-SHOT LEARNING MODELS (FSL):

According to Duan et al. (2021), FSL is an important area in machine learning (ML), in which the focus is on the use of very few labeled examples to train the model. Compared to traditional deep learning methods, which typically require large datasets for effective training, FSL models are designed to adapt quickly to novel classes using only one (one-shot) or a few examples per class [15].

FSL is frequently described through N-way-K-shot setting/problem. Specifically, N-way-K-shot is a setting which is used to train models to recognize and classify new concepts, classes or patterns, using very few labeled examples (IBM, n.d.) [80]. Usually, the N denotes the number of classes, and the K denotes the number of examples for each class (Yang et al., 2022) [16]. The reference set of the task is the support set, which provides reference information for the prediction stage of the task. The query set contains unseen examples that the model must classify based on the information given by the support set. In a classical way, N-way-K-shot task represents support set with N categories and K examples, then, the support set contains a total of N x K labeled samples (Song et al., 2022) [7]. This support–query structure is a fundamental characteristic of Few-Shot Learning, since classification is performed on the basis of very limited task-specific information rather than large-scale training data.

Figure 1: Few-Shot Learning model concept [71]

FSL models are trained through episodes, which are organized into mini batches. Each episode functions as a standalone task consisting of a support set and a query set. Episodic training is widely used in FSL because it mimics the evaluation setting, where the model must classify new samples based on a small support set. In this way, each episode allows the model to learn how to solve few-shot tasks rather than simply process individual samples. In many FSL approaches, meta-learning further supports this process by enabling the model to adapt more rapidly to new tasks using previously acquired knowledge (Foukanelis, 2025). [79] [80]

FSL methods are divided into three basic categories. The first category includes the metric-based methods, in which the model compares examples based on similarity or distance. A characteristic example of the below, is Prototypical Networks (Snell et al., 2017) [17] which calculate a prototype per class as the average of the embeddings and classify new examples based on their distance from each prototype. A similar logic is followed by Matching Networks (Vinyals et al., 2016) [18], which are using an attention mechanism to compare queries with examples of the support set.

The second category refers to optimization-based methods. The most well-known method in that category is the Model-Agnostic Meta-Learning (MAML), introduced by Finn et al. (2017) [19]. This algorithm trains a model so that it is in a parameter initialization state from which it can quickly adapt to new tasks with minimal gradient updates. The algorithm follows a double training loop: the inner loop performs rapid adaptation per task while the outer loop updates the meta-learner globally.

The third category refers to model-based methods. These methods use specialized architecture (usually neural networks or external storage memory) to store and recall rapid information from past tasks. Prime example is Memory-Augmented Neural Network (MANN) by Santoro et al. (2016) [20], which extends Neural Network architectures with external memory, allowing the model to store and retrieve examples from new classes with high efficiency. A similar logic is adopted by SNAIL (Mishra et al., 2018) [21], that integrates temporal convolutional layers and attention mechanisms to

handle task sequences, while achieving competitive performance in few-shot learning problems.

The growth of the FSL Models was largely driven by applications where the collections of larger amounts of data were very costly or impossible in examples such as medical diagnosis, detecting intrusions in computer networks and identifying rare entities. However, the same feature that makes those models particularly useful (adaptation from few examples, small datasets) is also what makes them vulnerable to attacks, as the attacker will need much less data to significantly affect the model's behavior in attacks like poisoning the support set.

3.1 Analysis of Models:

Contrary to Conventional Supervised Learning, where the model is typically trained on a large amount of data and on classes that are available during training, in FSL the model is required to generalize to new classes based on a very limited number of examples. In FSL, the prediction is not solely based on a model that has learned a stable set of classes but is organized around the support set and the query set, which means that it is organized in a task-based structure. This makes FSL a distinct learning setting and therefore more demanding. As previously mentioned, there are three main categories in FSL: metric-based, optimization-based and model-based. The following section compares the three main FSL categories.

3.1.1 Metric-Based Methods:

As mentioned in section 3.1, one of the three categories is metric-based. In this category, the examples from the same class should be close together in the embedding space, while the ones from different classes should be far from each other. The model's effectiveness depends on how the data is embedded, how each class is represented and how the distance between them is counted. One of the models used in metric-based approaches is Siamese Networks.

Siamese Networks architecture consists of two input samples. The encoders (often referred to as the model's backbone) are processing the samples which are identical and use the exact same parameters. As a result, they produce a vector representation of the input. Then the encodings are compared based on similarity, because if the inputs are similar, their embeddings will also be similar. Finally, the distance between the vectors is calculated, so similar inputs will have embeddings close together and different inputs will have embeddings far away. [22]

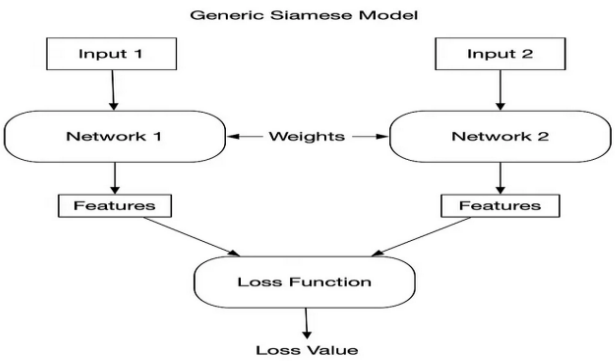

Figure 2: Generic Siamese Model [23]

Prototypical networks are one of the foundational models and the most common one for metric-based FSL. They introduce the idea of class prototypes and classification by distance. More specifically, Prototypical Networks learn an embedding space in which every class has a prototype computed by the embedded support samples in the class, named the average embedding. It classifies the query sample by comparing to which prototype in the learned metric-space is the closest one. Compared to similarity functions in Siamese networks, in which the comparison is pairwise, the prototype networks rely on distance to the classes prototype. [17]

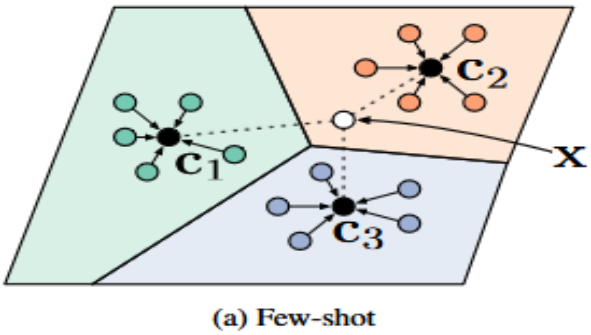

Figure 3: Prototypical Networks in the few-shot scenario [17]

Matching Networks is another metric-based model. It starts with making the inputs embeddings, like every other metric-based model. Then it compares the cosine similarity of the query sample to every sample in the support set. Next, the attention mechanism weighs every sample in the support set. The model then sums the attention weights for the support samples of each class and predicts the class with the highest total weight. [18]

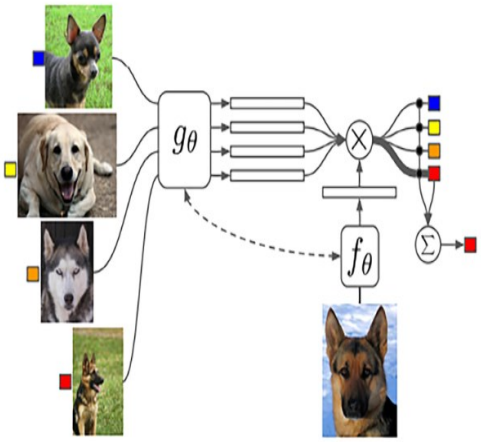

Figure 4: Matching Networks architecture [18]

Relation networks while being a metric-based model, are different compared to the previous models. They follow a unique path to decide where to classify new data. They first make the inputs to feature maps in the first part of the embedding module and concatenate both inputs. Then comes the relation module in which, based on many layers of non-linear activation function (ReLU), they compare the two inputs. The model is trained with episodes and based on the relation score it classifies the input into the right class. [26]

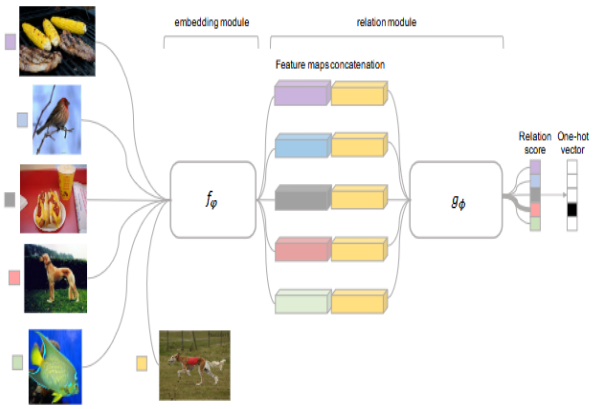

Figure 5: Relation Network architecture for a 5-way 1-shot problem with one query example. [26]

Finally, there are the **Graph Neural Networks (GNN).** In this approach, every input, both labeled and unlabeled, is defined by a node and is on a fully connected graph. Each node is connected with the other by the edges that are trainable similarity functions, and not with fixed mathematical formulas like Euclidean or cosine distance. The GNN contains multiple layers, where in each layer is using a non-linear approach. GNN determines how nodes share information (adjacency matrix), based on their similarities. Nodes with high similarities are given higher connection weights and are getting volumed up, while the ones with few similarities are getting volumed down in every layer. Graph neural networks are working very similarly with Relation networks, but they are more evolved

and more efficient according to Victor Garcia et al. (2018) by using significantly fewer parameters [28].

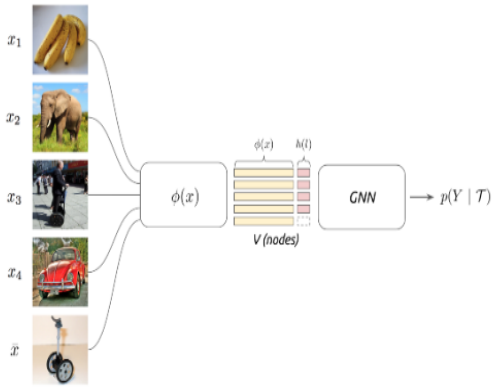

Figure 6: Graph Neural Network architecture for few-shot classification [28]

A summary of the above models and their way of comparing with the query set can be found on the table below:

Model	**Comparison**
Siamese Networks	Pair-to-Pair
Prototypical Networks	Query-Prototype
Matching Networks	Query-support set
Relation Networks	Feature concatenation
Graph Neural Networks	Label propagation

Table 2: Comparison Strategies in Metric-Based Few-Shot Models.

3.1.2 Optimization-based Methods:

This category refers to the models that the meta-learning procedure happens within their optimization process and not by comparison like in metric based. Instead of comparing distances, the Optimization based models can adapt quickly to any new task with only few gradient updates. Those models are categorized as follows: initialization-based, update rule-based and latent-based.

The **initialization-based** models' representative is Model Agnostic Meta Learning model. The way the model works is that it has two optimization loops in which the inner loop adapts to tasks with gradient descent on the support set and the outer loop updates the initialization parameters based on model's performance (loss) on the query sets across all tasks in the mini batch [19].

Update rule-based methods shift the focus from finding a good starting point like initialization-based models, to learning to optimize the algorithm itself. The main meta-learner used in this category is LSTM and functions as an optimize. This meta-learner is trained to observe the gradients of a learner network and output the parameter updates needed for each specific task [31].

Finally, the **Latent-based** methods. In this approach an Encoder is analyzing and compressing the task's support set into a smaller latent code. The model does all the necessary optimizations inside the latent space, instead of messing with thousands of parameters and weights like MAML does. Then, a decoder takes the optimized code and translates it back into the actual weights. With that, the numerous amounts of parameter changes are minimized and as a result, the model is able to handle new tasks in a more efficient way [32].

Category	Idea	Mechanism
Initialization-based (MAML)	Finding a good starting point	Finds weights, and adapts in gradient steps
Update Rule-based (LSTM)	Learning the optimizer	Predicts weight updates instead of gradient steps
Latent Based (LEO)	Learning in a latent space	Optimizes latent code, then decoder translates to weights.

Table 3: Comparison of optimization-based methods.

3.1.3 Model-Based Methods:

Model-based methods are designed to learn instantly new tasks without the need to change the "weights" with gradient updates like the optimization-based methods. Their main method is that the knowledge for a new class is not saved in the network but on an internal state of the model or in an external memory.

The Memory-Augmented Networks use a dedicated memory matrix, or external memory (similar to RAM), that stays separate from model's parameters, unlike traditional models that store all the knowledge in their static "weights". This allows the network to store new information about classes dynamically and retrieve later without changing the underlying weights. In FSL models, it will store the X number of new classes in external memory and find the best match later. This works much faster than other models [8], [20], [34].

Fast weight methods are a bit different than the ones mentioned before, mainly because they are not based on neither external memory (MANN), nor gradient descent (MAML), but with weight generation. What this means is that the meta-learner instead of classifying extracts task-specific features and calculating the values of the parameters needed. Those are called fast weights and are calculated in just one step. The calculated values then pass into the main network to adapt to the new task. This is like having a network within your network [35].

Generative model-based paradigms are using generative architecture such as VAE (Variational Autoencoders) and GAN (Generative Adversarial Networks) as a data augmentation strategy. Those models are not relying on the support set which is limited, instead they design additional training examples or features. So, by using the known classes, they generate new examples of unseen classes, and they convert the model

from a few-shot learning problems into a more standard learning scenario with bigger sets of data. More importantly this is a way of mitigating risks in Few-Shot [36], [37].

Graph-based approaches, as previously said in the Metric-Based section (3.1.1), also function as Model-Based architectures. They treat the support and query sets as a connected network, where the architecture itself facilitates label propagation through the graph's structure, eliminating the need for traditional optimization [28].

Although metric-based, optimization-based and model-based few-shot learning methods differ in how they adapt to new tasks, they all rely on deep neural network components. Metric-based approaches use a neural encoder (backbone) to map inputs to embeddings before applying a distance-based classifier, optimization-based methods adapt the parameters of a neural network through gradient descent on the support set, and model-based methods employ neural architectures or generators together with memory modules or weight generators. As a result, most FSL models have the same vulnerability to gradient-based adversarial attacks as standard deep neural networks, which allows classical adversarial-example methods such as FGSM and PGD to be directly applied in few-shot [38], [39], [40].

3.2 Threat Model:

As FSL models gain popularity because of their ability to function with very limited labeled data, the importance of understanding their unique vulnerabilities has increased. It is crucial to understand the unique attack surfaces of FSL models, as well as their vulnerability and robustness against adversarial examples and attacks. It is known that Standard deep learning classifiers are vulnerable to adversarial attacks, but FSL introduce additional attack surfaces through their support set, query set, episodic training, and task-adaptation process. Therefore, it is necessary to define the adversarial setting in FSL-specific terms. In this section, we describe the main dimensions of the threat model for FSL, focusing on the attack surfaces, the adversary's capabilities and knowledge, and the possible goals of the attack, in order to clarify how threats in FSL differ from those in conventional supervised learning. As previously mentioned, studies have shown that it is possible to craft adversarial examples that seem the same as legitimate examples and human or algorithms can't see the difference. The way this happens is by taking a legitimate unperturbed input and adding a small amount of noise [39].

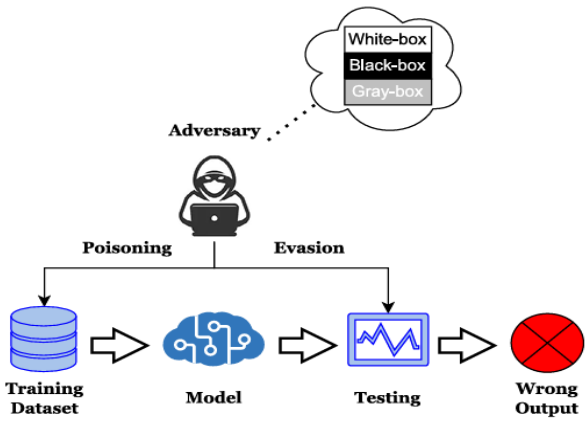

Figure 7: Adversary attacking AI models. [41]

3.2.1 Attack surfaces:

There are three main attack surfaces in FSL models, first being support set then comes query set and lastly meta-training. Depending on what the adversaries' goal and access is, targets might be one or more of the above components.

In Few-Shot, the support set is what contains the few labeled examples that will be used to adapt the model to the new task needed, considered the training-data for that specific episode.

Support set attacks refer to when the attacker manipulates the few labeled examples of the task to cause the model to misclassify the query input. Commonly seen those attacks include adversarial perturbations on existing support inputs without changing the inputs label, injecting malicious crafted samples within the support set, with the same goal of causing the models accuracy on predictions to drop on the specific query set. If we look at this from another point of view by a conventional supervised learning model, we easily can understand that the support set in FSL is much more sensitive due to their low number of examples per class [10].

The query set, that is also the next attack surface, contains the unlabeled examples that the model has to classify after "training" with the support set. The attacks in this case happen not during the training process nor by modifying support set. They happen during the test-time in each episode by applying adversarial perturbations to query inputs either per-example or universal. The result of this attack is that it significantly reduces the accuracy in the few shot while leaving the support clean [6].

The last attack surface is meta-training, in which the adversary manipulates the base dataset or poisons the backbone during the model's initial training phase. If successful, such attacks can corrupt the feature representations learned by the backbone, affecting all future tasks the model encounters. Unlike support-set and query-set attacks, which are limited to a single episode, meta-training attacks would persist across every task at meta-test time. Yang et al. (2024) proposed an in-process defense against backdoor

attacks in few-shot classification and evaluated it in a poisoned meta-training setting, where a patch-based trigger is injected into the pre-training data [42].

3.2.2 Adversary Knowledge:

This section refers to knowledge the adversary has focusing in white-box, gray-box and black-box and access to the target system and then explaining how these apply on FSL.

White-box attacks are when the adversary has complete knowledge of the model that has been used, including its parameters, how it was trained, and access to gradients with respect to inputs. This allows the attacker to apply gradient-based methods (for example FGSM or PGD-style updates) to craft small adversarial perturbations that reliably cause misclassification. Because the adversary can optimize the perturbations directly, they can craft highly effective inputs that confuse the model [39], [40].

Black-box attacks are when the adversary has no access to the internal details as in white-box attacks. The adversary can only attack by observing and interacting with the input–output interface. Without knowledge of any part of the model, the attacker must rely on query-based methods (for example, estimating gradients from repeated queries to the model) or on transfer attacks from surrogate models [43], [44].

Gray-Box lies in between the two previous attack types; in this case the adversary has partial knowledge of what is being used in the model. They might know what model type or backbone architecture is being used but not the exact parameters or weights. Gray-box attacks usually follow a black-box style of interaction but are more realistic in real-world scenarios than a white-box attack that assumes knowledge of everything.

This is how the general “box” terminology is used in standard supervised models. FSL follows the same idea, but the notion of what counts as the “model” is broader and introduces some unique aspects. It's worth clarifying that the adversary's knowledge (white-box, gray-box, black-box) and the attack surface (query set, support set, meta-training) are two independent dimensions: the knowledge level is about how much the attacker knows, while the surface is about what they target. A white-box attacker, for example, has full access to gradients and parameters and can use it to attack either the query set, as in evasion attacks, or the support set, as in support-set poisoning, where perturbing even one of the few support examples can drop the model's accuracy significantly [10].

Gray-box attacks in FSL are the most realistic scenario due to their way of “act”. As said in previous paragraph, the attacker has knowledge on the model’s backbone architecture but doesn’t have access in specific parameters or weights. It is common to FSL to use pre-trained backbones, an adversary can use the publicly available datasets to train a “copy model” with the same architecture. By understanding the model’s vulnerabilities, the attacker can craft samples to fool the model. Because the backbone is what the model will use to adapt to any new few-shot task, these types of attacks are

compromising the model's ability to learn new classes, and the attacker didn't need access to information like support set or query set [38], [40].

Unlike standard black-box attacks that often rely on thousands of queries to estimate the gradients, many practical black-box attacks in deep learning rely instead on transfer attacks from surrogate models, trained on auxiliary data. Recent work outside FSL, such as CEMA by Wang et al. (2025), shows that this can even be done in a very query-efficient way for multi-task NLP models, by training a "deep-level" substitute model on clustered input–output representations and then transferring adversarial examples to a black-box victim. However, this approach is developed for multi-task text models, and to the best of my knowledge, there is currently no published work that adapts such deep-level surrogate strategies or any other query-efficient black-box method specifically to FSL models [44], [45].

3.2.3 Goals:

Besides the knowledge of the adversaries and access level in the target system, adversaries differ in their goals as well. Following common adversarial Machine Learning taxonomies suggested by NIST. The different goals an adversary will aim are related to integrity, availability and confidentiality [46].

Integrity attack's goal is to make the model give incorrect predictions without making the system stop working. In FSL this is either a targeted attack where specific samples are forced to be misclassified, or untargeted attacks, where the goal of the adversary is to reduce the overall accuracy on few-shot episodes. Support-set poisoning and adversarial querying methods are typically integrity attacks, adversarial support poisoning can drop the accuracy significantly by perturbing a small percentage of the support examples [10].

Availability attacks aim to make the system unavailable or completely unreliable. In FSL, availability attacks would mean the model cannot process episodes as a whole. The accuracy dropping even close to random guessing, will be classified as an integrity attack in this taxonomy and not availability because the model is still running and giving responses, which are wrong. While some of the attacks that will be discussed in chapter 4 are claimed to be availability attacks, this thesis will follow the NIST definition and classify the attacks causing big accuracy drops as integrity and not as availability since the model is operating [46].

Confidentiality attacks are privacy-oriented attacks, and they aim to extract sensitive information about the training data, the model and the architecture. While this is an important part of the goals of the adversary, this thesis won't mention any attacks from this category due to the lack of material found online.

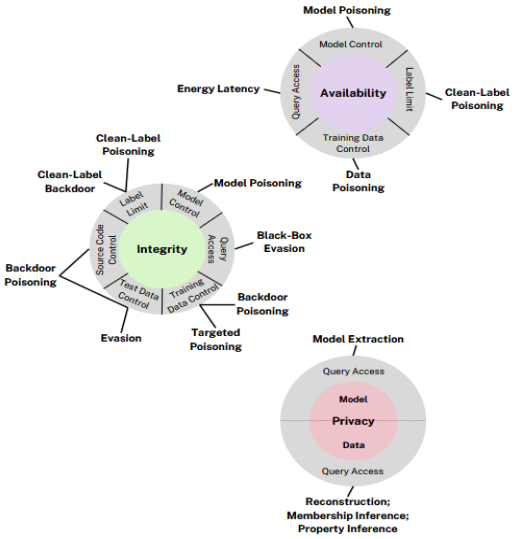

Figure 1. Taxonomy of attacks on PredAI systems

Figure 8: Taxonomy of attacks on PredAI systems. [46]

3.2.4 Timing:

The final dimension of the threat model, in which the following taxonomy will categorize the attacks into, is the timing of the attack. The two categories are meta-training (training-time), which is the time that the model parameters are still being optimized, or meta-test (inference-time) which is the time that model is trying to make predictions (test-time).

In **training time**, the adversary is manipulating the data the FSL model used to learn or adapt from. Clean-label poisoning would be a good example of training-time threat in FSL, the adversary would craft samples that carry the correct label but are optimized to corrupt the backbones feature representations. Backdoor poisoning for standard deep learning is another example for that, given that the adversary can insert a small number of trigger samples in the training data and can "confuse" the backbone to associate the trigger with a label chosen by the attacker himself, which will lead into misclassifying in the meta-test time. But as seen in chapter 4, FSL backdoor attacks are focusing on inference time at support set, making them a different threat [48], [49].

In **inference-time**, this type of attack happens after the model has been trained and it occurs when the adversary is trying to manipulate the support set or query set of the new episodes. If the adversary manages to add a trigger into the few support images in the support set, it could be enough to force triggering queries that would classify into the adversaries desired class, which would make this a successful backdoor attack during inference-time. Query sets are also vulnerable since the adversary can construct universal or task-specific adversarial perturbations, leading to degrading significantly the accuracy of the model while keeping the support set clean labeled [6].

In the broader backdoor literature on standard deep learning, there are attacks that affect both phases: the adversary poisons the training data to plant a backdoor trigger,

which is later activated at inference time when the trigger appears in the input. In FSL, however, the backdoor attacks discussed in this thesis focus on poisoning the support set at inference time rather than manipulating the meta-training data directly, which makes them a different type of threat [49].

The taxonomy will follow the four dimensions: attack surface, attacker knowledge, adversarial goals and timing of the attack, to categorize the attacks in the following chapters, to allow us to systematically compare existing published work and to identify under explored areas.

4. ADVERSARIAL ATTACKS:

In this chapter, we focus on adversarial attacks that target FSL models. We organize these attacks using the four dimensions of the threat model we defined in Chapter 3: knowledge, goal, surface and timing. After researching the literature in Google Scholar and other sources we will group the attacks into six families depending on the mechanism they use to attack: query-set evasion attacks, support-set poisoning, clean-label poisoning of the training data, backdoor and trojan attacks, universal adversarial perturbations, embedding space attacks. We will describe representative attacks and explain how they are successful and provide a unified table that will place all the attacks based on the four dimensions.

4.1 Evasion Attacks (query-set perturbations):

The first attack family will be evasion attacks. This type of attack family is the most studied type of adversarial attacks in deep learning models. Evasion in this context means that the adversary does not interfere with training data nor the model's parameters but perturbs the input at test time. In this family, the query set is being attacked, while the support set as well as, the meta-training data remain clean, and the attacker adds a small perturbation to the query image of a specific episode. We will treat those attacks as query-only attack surface that leaves adaptation data unchanged. These attacks fall under white-box attacks due to the knowledge of the attacker in gradients of episodic training. The goal is to drop accuracy during the meta-test, and the timing is at inference time, when the training is complete.

The first adversarial attack is Fast Gradient Sign Method (FGSM), that was introduced by Goodfellow, Shlens & Szegedy at (2015) in their paper "Explaining and Harnessing Adversarial Examples". After having an input x with its label, y the models loss function J and the perturbation budget ε the adversarial example is computed as follows:

$$x\_adv = x + \varepsilon \cdot sign(\nabla\_x J(\theta, x, y))$$

The gradient of the loss with respect to the input tells the attacker which direction each pixel must shift to maximize the error. Taking only the sign (direction) rather than the full gradient means every pixel is pushed by exactly ε in whichever direction increases the loss the most, making this attack as effective as allowed within the perturbation range. FGSM is a one-pass attack, which is why it is a computationally cheap one. In FSL, the attack is applied to query set images right before they pass the backbone to get classified. Even with small ε values, it is enough to cause misclassification without touching the support set [39], [6].

The second attack is another very common one, named Projected Gradient Descent (PGD), and introduced by Madry et al. (2017) in "Towards Deep Learning Models Resistant to Adversarial Attacks". FGSM that was mentioned before is a single step attack to maximize the model's error, however, this one extends it into a multiple step attack. Instead of a large gradient step, PGD takes many small steps and rechecks after every step to make sure its perturbation stays in the allowed range ε.

$$x^{t+1} = \Pi_{x+S}(x^t + a \cdot sign\left(\nabla_x L(\theta, x^t, y)\right))$$

Where α is the step size and Π_{x+S} is the projection operator (to make sure after each step that the next gradient step is not too far from the original input). Because PGD is a multi-step attack with many repeats, it finds stronger perturbations than the FGSM in the same range. Because of how strong this attack is, it became a standard attack in benchmarks that evaluate FSL models robustness. In FSL, PGD is applied directly to query sets to measure how much accuracy drops under attack. Goldblum et al. (2020) demonstrated that naturally trained meta-learners achieve 0% accuracy under 20-step PGD attacks on 5-shot Mini-ImageNet, with ProtoNet (70,23% clean accuracy), R2-D2 (73,02% clean accuracy), and MetaOptNet (78,12% clean accuracy) all completely failing under attack. Dong et al. (2024) also used it as their primary evaluation attack, proving how widely PGD is used as attack for testing FSL defenses [40], [6], [13].

The next core and well-studied adversarial attack we are including in the taxonomy will be introduced by Carlini & Wagner (2017) in their work "Towards Evaluating the Robustness of Neural Networks" C&W. This attack follows a different logic than the previous ones. Instead of using gradient signs to maximize the loss, this attack tries to find the smallest perturbation δ so the model misclassifies x + δ. The attack optimizes:

$$\text{minimize } ||\delta||_p + c \cdot f(\chi + \delta)$$

where f is the attack loss function that becomes negative when the attack is succeeded and c is found by binary search and must be greater than 0. This attack can be targeted

at forcing the model to classify in a specific class that is chosen by the attacker, or untargeted where the goal is to cause misclassification. C&W comes in three variants depending on which norm is being minimized. L_0 which minimizes the numbers of pixels changes, L_2 which minimizes the total perturbation size and L∞ that minimizes the maximum pixel change among all pixels. This makes this attack more flexible compared to FGSM and PGD that are usually on the L∞ norm. C&W was the first attack to bypass defensive distillation, a defense that was considered robust at the time. While C&W is well grounded in standard machine learning it is used as an evaluation attack on FSL models too, to test the robustness along with PGD. Dong et al. (2024) used it to verify their defense suggestion are strong against optimization-based attacks too, along with gradient sign [50], [13].

This attack family concludes with this last attack in this study, which is AutoAttack, that was introduced by Croce & Hein (2020) in their work "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks". Autoattack was proposed because many defenses looked robust just because the attack wasn't fine-tuned enough or just simply not strong enough. AutoAttack includes four attacks with no parameters : APGD-CE which is an adaptive PGD on the cross-entropy, APGD-DLR which is an adaptive PGD that is based on difference of logits ratio loss (instead of softmax), the third attack in AutoAttack is FAB which is a targeted attack that minimizes the distance to the closest decision boundary and last is Square Attack which is a black-box score-based attack that queries the model's output probabilities to guide random square-shaped perturbations until the model misclassifies the query. The reason Autoattack is among evaluation benchmarks is because it has both white-box and black-box attacks so it's easier to find an adversarial example or in cases prove your model is robust. Dong et al. (2024) use it as their benchmark robust evaluation in their suggested defense [51], [13].

Attack	**Attack Surface**	**Adversary Knowledge**	**Adversarial Goal**	**Timing**
FGSM	Query Set	White-box	Untargeted Integrity	Inference-Time
PGD	Query Set	White-box	Untargeted Integrity	Inference-Time
C&W	Query Set	White-box	Untargeted /Targeted Integrity	Inference-Time
AutoAttack	Query Set	White-box/black-box	Untargeted Integrity	Inference-Time

Table 4: Evasion attacks mapped across the four taxonomy dimensions.

From the above table, the evasion attack family is concentrated on the query set as their attack surface, with integrity as their adversarial goal of the model. Many of them are white-box attacks and happen during inference-time after the model has been trained. This corresponds with the fact that those attacks do transfer in FSL models but were not designed for specifically those. They came from standard deep learning vulnerabilities and do not take into consideration the FSL architecture with the support set and the episodic training. The next attack family will exploit exactly that.

4.2 Support Set Poisoning Attacks:

Unlike evasion attacks which target query samples, this attack family manipulates the support set to break the model's adaptation step. Critically, these attacks happen at inference time (meta-test), not training time, and their effects are episodic, not persistent which is how those are distinct from the training-time poisoning attacks discussed in Family 3. The support set is the small set of labeled examples the model uses to adapt and classify query images during each episode. If an attacker manages to poison even a small percentage, such as 20%, they can break the entire episode and cause the model to misclassify every query image. We will cover the work of Oldewage et al. in their two studies that exploit this attack surface in different ways and then map them onto the taxonomy table at the end of the chapter.

The attack we will discuss in this family is presented by Oldewage, Bronskill & Turner (2021) in "Attacking Few-Shot Classifiers with Adversarial Support Poisoning". This was the first paper to define and evaluate a support set poisoning attack in FSL models. The attack works by backpropagating through the meta-learner's adaptation step at meta-test time. The attacker crafts poisoned support images that keep their correct labels but are optimized using the models' own gradients to break the adaptation step. ASP used PGD as the optimization algorithm to craft the poisoned support examples. The optimization the attacker solves is the following:

$$\delta^* = \text{argmax}\, L(f(x^*, g(\chi + \delta, y)), y^*) \qquad ||\delta||\infty \leq \varepsilon$$

x,y = support set inputs and labels (Ds), x*,y* = query set inputs and labels (Dq) , g= meta-learner adaptation function, f= classifier.

PGD also appears in this attack family with the difference that this time it's targeting the support set and not the query set as in the attack family 1. ASP exploits the fact that the support set determines the adaptation of the model, meaning small perturbations to a small percentage of the support samples can drop the accuracy across all query samples in that episode. It's worth noticing that the attack can generalize beyond the query set used to craft the poisoned set. The attack was evaluated across ProtoNet,

MAML and CNAPs. CNAPs showed the highest decrease (as shown in Table A.7 of the original paper), accuracy fell from approximately 70% to 9%, when only 20% of support images were poisoned in some datasets, proving that the support set attack surface is a valid and dangerous concern. An important thing to notice in this attack is that the authors classify ASP attack as an availability one, due to making the model useless for the episode, however we are following the NIST definition explained in section 3.2.3 [10].

The following year the same authors published a newer paper "Adversarial Attacks are a Surprisingly Strong Baseline for Poisoning Few-Shot Meta-Learners", that is an extended version of ASP to meta-learners such as simple CNAPs and studies its transferability. The result was more dramatic with accuracy dropping to some cases from 78,5% to 0% under the same attack features, white-box, with only 20% poisoned of the support set. The interesting finding comes on their test of ASP transferability which was not better than an average swap attack (pretty similar to evasion attack but by adding the perturbed image in the support set instead of the query set). While ASP is a very strong attack on the white-box scenario, it lacks on the black-box one, making it rather strong, but unrealistic in real world scenarios [52].

This attack "MetaAttacker" came from Xu,Li,Liu & Tang (2021) in their paper "Yet Meta Learning Can Adapt Fast, It Can Also Break Easily". MetaAttacker is a support set poisoning attack that happens during inference-time like the ASP. Because of the small number of examples the model uses to learn, the manipulation of a small sample will cause the model to learn wrong. The attack does not target the query set, but the support set and the adaptation mechanism.

In a clean setting the meta learner f_θ receives the D^T_{train} so the model can adapt like that:

$$\varphi \quad = f_\theta\,(D^T_{train})$$

After the attack the support set will be replaced with adversarial like that:

$$\varphi' = f_\theta\,(D^T_{adv})$$

This attack does not try to misclassify poisoned query samples but instead it targets on the model performing badly on clean set samples and it happens because the model "learned" and adapted with poisoned support samples:

$$\underset{D^{\mathcal{T}}_{\text{adv}}}{\text{maximize}} \quad \sum_{x,y \in D^{\mathcal{T}}_{\text{train}}} [\mathcal{L}(F(x;\phi'),y)]$$

The attack has two constraints. First is that the attacker must poison a small number of support samples k :

$$\sum_{x^{\text{adv}} \in D^{\mathcal{T}}_{\text{adv}}} \quad \mathbb{1}(x^{\text{adv}} \neq \text{clean}(x^{\text{adv}})) \leq k,$$

And secondly each perturbation must be small:

$$||x^{\text{adv}} - \text{clean}(x^{\text{adv}})|| \leq \epsilon, \quad \forall x^{\text{adv}} \in D^{\mathcal{T}}_{\text{adv}}.$$

Those restrictions exist so the attack remains “stealthy” and undetectable. The authors that performed this attack studied both targeted and non-targeted attacks. In the non-targeted scenario, their aim is to drop accuracy in overall, or how they call it “Denial of service”. Similarly to ASP, Denial of Service will be categorized as an Integrity attack due to the model working even with low accuracy. In the targeted attack the attacker aims to fail to classify the support samples of one specific class.

The attack works by choosing a support sample to poison by the two constraints mentioned above. To avoid being too expensive computationally, they use a greedy search and select the sample that increase the loss by the most and they add the perturbations there. This attack is a white-box attack due to the knowledge the attacker has. They tested the attack in models like MAML, ProtoNet, SNAIL, and in all models, accuracy dropped significantly. MiniImageNet non targeted on MAML dropped from 63,3% to 16,2% with 10 samples and to 48,2% with just 1 sample. Omniglot not targeted on MAML dropped from 99% accuracy to 61,4% with 10 samples. This is another proof of the advance made by fast adaptation leaves an open surface for attacks [53].

We mentioned before how support poisoning is a FSL native attack family in this taxonomy compared to evasion attack family that transferred from the standard deep learning, because of the support set which is unique to FSL. The three studies show how a small percentage of the support set being poisoned could lead to the whole episode collapse and many different FSL architectures are vulnerable to the same attack. After researching, the real finding is that while FSL has such a specific attack surface area, literature is not so extensive. We only found the two main papers mentioned previously ASP and MetaAttacker and the follow up paper in ASP. The fact this attack surface remains underexplored in the literature, lacks more attacks in the area and the finding that this attack is not transferable in black-box scenario in the case of ASP leaves a good gap for future research, having a successful black-box support set poisoning without full model access.

Attack	Attack Surface	Adversary Knowledge	Adversarial Goal	Timing
ASP	Support set	White-Box	Untargeted Integrity	Inference-time
MetaAttacker	Support-set	White-box	Targeted/untargeted integrity	Inference-time

Table 5: Support-set poisoning attacks mapped across the four taxonomy dimensions.

What we can notice compared to table 4, is that the attack surface switches from query set to support set. We notice that timing is still inference-time like in evasion attacks but for quite a different reason. The aim of the first attacks was misclassification, and that was the target, however, in this category the attack aims to the adaptation step.

4.3 Clean-Label Poisoning Attacks:

This attack family refers to clean-label poisoning attacks, where the adversary injects samples during meta-training time (not in inference like families 1 and 2), that even when they carry the correct labels, are crafted to make the model inaccurate at inference time. This is a “stealthier” attack because the labels are not changed and the images appear normal.

Similarly with evasion attacks, this attack family is not a FSL specific attack family but was introduced in standard deep learning by Shafahi et al. (2018), a targeted attack that doesn’t require any control on the labeling of training data. The images carry correct labels but are optimized to cause misclassification on a specific test but do not affect the model’s overall behavior. Later on, Huang et al. (2020) introduced MetaPoison which is an extension of the previous attack that solves a bi-level optimization problem to craft clean label poisons to confuse the deep neural network. Later on, came Geiping et al. (2021) that employed gradient matching to craft poisons and achieved higher attack success rates [48], [54], [55].

If we bring these attacks into FSL context, the adversary would need to target the meta-training data during training (f.e miniImageNet

64 base classes). The poison would be in the base dataset and would be “absorbed” by the backbone. In meta-test time the backbone doesn’t change. The adaptation step updates temporarily the task-specific weights without changing the backbone, meaning the poison will survive across every episode, no matter how clean the support and query set is. Support set poisoning from Family 2 is limited to one episode and requires the attacker to access the user's data at that exact moment. Clean-label poisoning only requires touching the training dataset once and the effect is permanent.

This is not an easy task to do and has two main challenges to address, to bypass that. Meta-training is episodic-training, meaning every episode picks randomly some sample classes and images within those classes each time, so it's not guaranteed a poisoned sample will appear. Even if it's picked, it is compared with another random sample class. Meta-learning models are trained to "learn how to learn" and recognize patterns, so if a poison appears irregularly, it gets filtered out as noise. Another important difference is that in DL you poison the class to affect that specific class in test time. In FSL the model picks unseen classes that it hasn't seen in the training time. The poison would need to warp the feature space by guessing where the unseen class will land later which is extremely difficult. For optimization-based models like MAML this type of attacks become even harder. The poison needs to be strong enough, that the backbone produces identical feature vectors for different classes, because otherwise the adaptation step can technically find a boundary to separate them.

But nothing relevant was found, an attack that implements clean-label poisoning and has as a target an FSL model. This is a direction for future work along with the difficulties that exist in that type. The closest attacks to clean-label poisoning for FSL, like FLBA and the CVAE-encoded backdoor, work by poisoning the episodic support set rather than the base training data, and are therefore placed under the backdoor family (Family 4) and not under clean-label attacks.

4.4 Backdoor attacks:

Backdoor attacks are a well-studied attack type in Deep Learning, first introduced by Chen et al. (2017) in "Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning" and later by Gu et al. (2019) with "BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain", where a trigger gets embedded in a small percentage of the large dataset and caused the model to misclassify any input carrying that trigger. It's important to understand what a trigger is. The attacker during training poisons some samples, by adding the trigger, which could be a texture or a perturbation, and assigning the samples with that trigger with the desired class. During training, the model learns to classify clean images in the correct class and to classify the samples with the trigger in the desired class. The model behaves normally on clean inputs and produces the attacker's chosen class when the trigger is present. In standard deep learning, this manipulation happens at training time. In FSL, as the following papers show, the attack surface shifts to the support set at inference time. It's also important to understand that in clean-label poisoning attacks, corruption is permanently embedded in the model, while backdoor attacks require the trigger as previously mentioned and in support set poisoning it's only in that specific task [49], [56].

FSL treats backdoor attacks in a different way but with a similar idea. In DL, the trigger is injected in a large training dataset. In FSL the trigger switches to being injected in the support set at meta-test time instead of training time, as shown in all three attacks that we will discuss in the following sections. Because the support set contains only a handful of images, poisoning even one or two would be enough to cause the model behavior to be corrupted for the specific trigger while everything else is classified correctly [14].

Kato et al. (2023) in their work "Backdoor poisoning attacks against few-shot classifiers based on meta-learning" introduce a backdoor attack in meta-test time targeting MAML, evaluated on the Omniglot dataset. The attacker's goal is to make a specific query image (with the trigger X'_{tr}) get misclassified into the attackers' desired class Y'_{adv}.

$$\begin{aligned}\hat{Y}'_{\mathrm{tr}} &= f(X'_{\mathrm{tr}}, g^*(\tilde{D}_S, \theta^*)) \\ &= Y'_{\mathrm{adv}}\end{aligned}$$

Where $\tilde{D}_S = (\tilde{\boldsymbol{X}}, \boldsymbol{Y})$ is the support set with perturbed data X, g* is the fine-tuning function and θ* are the meta-trained parameters. The attacker only modifies the support data and the trigger image itself remains clean. To get the poisoned support data the attacker must solve the following:

$$\begin{aligned}&\boldsymbol{\delta}^* = \arg\min_{\boldsymbol{\delta}} \mathcal{L}\left(f\left(X'_{\mathrm{tr}}, g\left((\tilde{\boldsymbol{X}} = \boldsymbol{X} + \boldsymbol{\delta},\ \boldsymbol{Y}), \theta^*\right)\right),\ Y'_{\mathrm{adv}}\right) \\ &\text{subject to } \|\boldsymbol{\delta}^*\|_\infty \leq \epsilon,\end{aligned}$$

This is the main difference between this backdoor attack and ASP. ASP maximizes the loss in the whole query set causing all queries to fail. Here the attacker minimizes the loss only towards the attackers chosen class, every other query sample classifies correctly. The perturbation is generated using PGD with step size γ and L iterations, keeping the perturbation within budget ϵ. This is a white-box attack because it requires meta-trained parameters θ* and the trigger X'_{tr}. The attack achieved 98,2% ASR of at ε = 0,07, while maintaining a backdoor accuracy of 97,92% and a clean accuracy of 99,46%, proving how a small perturbation budget is enough to plant a reliable backdoor without dropping the normal model behavior [57].

While Kato et al. proved that backdoor attacks can occur in FSL, Liu et al. (2024) in "Does Few-shot Learning Suffer from Backdoor Attacks?" generalized this by designing a reusable trigger pattern that works in many architectures and datasets, while remaining stealthy and avoiding overfitting features in FSL (overfits the poisoned and forgets how to classify the clean, overfits the clean and ignores the poisoned).

FLBA addresses both problems in a mechanism of four stages. It starts with generating a trigger t* with embedding deviation, in which x is the clean image, m is the mask that limits the size of the pattern, the benign zb and poisoned zp can be expressed as:

$$\begin{aligned} \mathbf{z_b} &= f_\theta(\mathbf{x}) \\ \mathbf{z_p} &= f_\theta(\mathbf{x} \odot (\mathbf{1} - \mathbf{m}) + \mathbf{m} \odot \mathbf{t}) \end{aligned}$$

The trigger t* is optimized to maximize the distance between benign and poisoned features as following:

$$\mathbf{t}^* = \arg\max_{\mathbf{t}} \sum_{\mathbf{x} \in \mathcal{S}} d(\mathbf{z}_b, \mathbf{z}_p),$$

Where d is the cosine distance. This solves the overfitting problem since the model learns with this maximized gap both clean and trigger features at the same time.

The second stage hides the trigger by generating imperceptible perturbations, instead of attaching the trigger to a specific image. For the target class, attractive perturbations δ_a bring the hidden poisoned features to the trigger features:

$$\begin{aligned} \min_\delta \quad & d(\mathbf{t_h}, \mathbf{t_p}) + \lambda_1 d(\mathbf{t_h}, f_\theta(\mathbf{x_t})), \\ \text{s.t.} \quad & \|\delta_a\|_\infty \leqslant \varepsilon. \end{aligned}$$

For the untargeted classes, repulsive perturbations δ_r push features away from the trigger:

$$\begin{aligned} \max_\delta \quad & d(\mathbf{u_h}, \mathbf{u_p}) - \lambda_2 d(\mathbf{u_h}, f_\theta(\mathbf{x_u})). \\ \text{s.t.} \quad & \|\delta_r\|_\infty \leqslant \varepsilon. \end{aligned}$$

Where λ1 is balance parameter of regularizer and attractive perturbation (bring their features closer to target class) δa, λ2 controls the closeness to its original feature, and repulsive perturbation δr. In the third stage the benign support set is poisoned with the perturbation and produces the hidden poisoned support set. In the last stage the model is being evaluated, clean queries give correct outputs, and any query with the trigger t* produces the target label. The authors evaluated this attack on Baseline++, MAML, ProtoNet on miniImageNet, with ASR 89,1%, 81,2% and 60,1% and benign accuracy 65,5%, 63,6% and 61,6% outperforming previous backdoor methods that were designed for DL like BadNet, Blended, Refool, WaNet,HTBA which failed in FSL. [14]

The more recent paper “Leveraging CVAE Encoding for Backdoor Attacks in Few-Shot Learning with Prototypical Networks” by Sana et al. (2025), extend the backdoor family by addressing some limitations that exist in previous works. First, in a small support set it is easier to spot the perturbations on pixel or feature space, because of the low amount. Secondly and mainly to Prototypical Networks small, localized pixel-space changes get diluted when averaged into the class prototype, making it harder for the prototype to steer towards the targeted class while keeping everything else the same. This also explains why FLBA achieved its lower ASR on ProtoNet at 60,1% compared to 89,1% on Baseline++. They solve both of these problems by removing the trigger from pixel-space and encoding it inside the latent representation of a Conditional Variational Autoencoder as shown below.

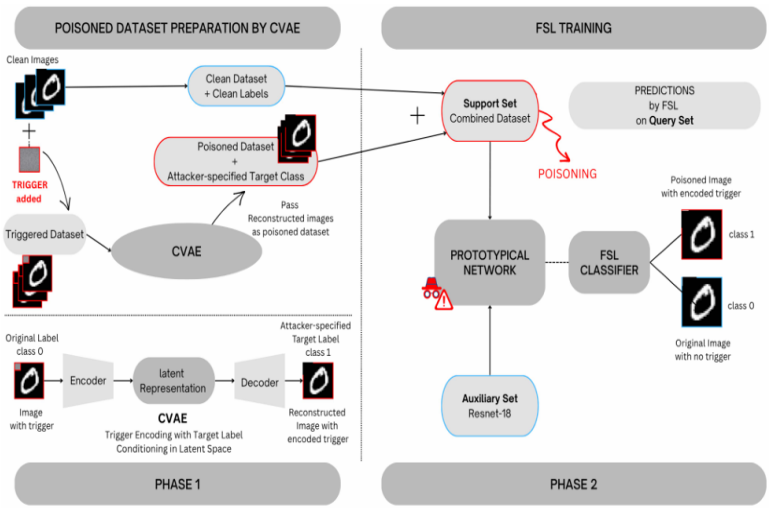

Figure 9: The framework of launching CVAE attack on prototype-based FSL. [58]

The attacker trains a CVAE. A visible trigger pattern T is concatenated to each clean image to produce a triggered input:

$$\mathrm{X}_{\text{poisoned}} = X_{\text{clean}} \oplus T$$

The encoder takes this triggered image together with a one-hot target label $Y_{one-hot}$ and produces a latent vector z using the reparameterization trick:

$$\mathrm{z} = \mu_x + \sigma_x \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

where μ and σ are the mean and standard deviation of the latent distribution. This latent vector then passes through an attention mechanism that splits z into query, key and value components. The attention weights $\alpha = softmax(query \cdot key^T)$ highlight the dimensions of z most relevant to the trigger. The attended vector $z' = \alpha \cdot value$ is passed to the decoder, which reconstructs the image conditioned on both z' and the target label:

$$X_{\text{reconstructed}} = P_\theta\left(z', Y_{\text{one-hot}}\right)$$

The original trigger T is not visible in the reconstruction. The CVAE is trained with a combined loss:

$$L_{\text{total}} = L_{\text{recon}} + \beta L_{\text{KL}} + \lambda L_{\text{TV}}$$

where L_{recon} is binary cross-entropy between the original and reconstructed image, L_{KL} regularizes the latent space to stay close to a standard Gaussian distribution, keeping poisoned and clean encodings statistically similar, and L_{TV} smooths the output image to avoid visible artifacts from the encoding. The reconstructed poisoned images are then combined with clean images to form the support set, labeled as the target class. This is how the prototype shift happens, the reconstructed images embed near the target class because the decoder was conditioned on the target label, so when ProtoNet averages the support embeddings into a prototype, the prototype drifts toward the target. At inference time, any query with the original trigger classifies as the attacker's chosen class while everything else classifies correctly.

The attack was evaluated on MNIST (10-way, 50-shot), Omniglot (26-way, 19-shot) and miniImageNet (100-way, 50-shot). On MNIST it achieves 100% ASR and 93% BA. On Omniglot 100% ASR and 84,62% BA. On miniImageNet 99,97% ASR and 56,75% BA

against a clean baseline of 68,20%. In the directly comparable 5-way 5-shot miniImageNet setting FLBA reaches 60,1% ASR while this attack reaches 99,89%, a difference of nearly 40 percentage points. The attack was also evaluated against six standard backdoor detection and defense mechanisms, bypassing all of them. The backdoor has no existence at the pixel level or as a detectable pattern in embedding space, which is why standard detection methods cannot defend against it.

We categorize this attack, following Sana et al., as a practical black-box clean-label backdoor: the adversary has no access to model parameters or gradients but knows the ProtoNet architecture and can inject poisoned support samples during the task-specific adaptation phase. In our taxonomy this places it between strict query-only black-box attacks and full white-box attacks, since it still assumes model awareness and control over the support set. The attack surface is therefore the support set and the timing is inference time, and structurally it is related to Family 6 because both operate in the embedding space rather than directly in pixel space, although FAMF perturbs the query feature vector while this attack shifts the support-set prototypes [58].

All three papers discussed in this attack family share the same categories in the taxonomy: the attack surface is the support set, and the timing is inference-time. Kato et al. and FLBA are white-box attacks, because the adversary needs access to the meta-trained parameters to optimize the perturbations, while the CVAE-encoded attack by Sana et al. follows a practical black-box setting, where the attacker has no access to gradients or parameters but still knows the ProtoNet architecture and can poison the support set during the adaptation phase. In all three cases the adversary goal is integrity, because the attacker aims to force specific outputs while the model is otherwise operating normally on clean inputs. The important difference between backdoor attacks and the poisoning attacks in Family 2 is that in a backdoor attack the adversary wants a stealthy misclassification of a chosen trigger, while in the discussed poisoning attacks the attacker breaks the adaptation step and ends up destroying the whole episode. This is another family in which strict query-only black-box attacks remain an unexplored area, since all three existing attacks either require white-box access to the model or, in the case of Sana et al., at least model awareness and control over the support set to craft the perturbations. Even though this type of backdoor attacks remains undefended in the current taxonomy, there is a defense mentioned in Chapter 5 that instead targets training-time backbone backdoors [57], [58], [14].

4.5 Universal Adversarial Perturbations:

Where the methods like FGSM, PGD, AutoAttack and C&W create perturbations to fool the model on a single image, the universal perturbations are able to fool the model with high probability, as introduced by Moosavi-Dezfooli et al. (2017) in “Universal adversarial

perturbations". That is what differs this attack family from the evasion attack family. While evasion crafts a unique perturbation for every input image, UAPs are input-agnostic, a single perturbation is generated once and can be applied to any image and cause misclassification. This makes this attack a more realistic real-world scenario because the attacker wouldn't need to create a unique perturbation for every new image query [59].

However, UAPs are typically a close-set to standard DL, where the training and testing tasks share the same classification model. FSL makes the classic methods ineffective for two reasons. First is the task shift. FSL tasks in meta-test are different from the ones in pre-training, so a UAP optimized for one task does not mean it generalizes to all tasks. The second is semantic shift. The visual categories at meta-test time are different from the ones UAP was crafted on, meaning the UAP was never exposed in the features it needs to fool.

Hu et al. (2024) in their work "Generate Universal Adversarial Perturbations for Few-Shot Learning" proposed FSAFW (Few-shot Attacking Framework). That solves both of the shift problems, the attacker has access on the pretrained encoder $f_e(\cdot)$ and uses a proxy dataset Dp to generate the UAP without prior knowledge to pre-training dataset. The adversarial image is crafted like this: $x_{adv} = x + \delta$, $\|\delta\|_\infty \leq \epsilon$., where δ is the perturbation and is trained by a generator $g_\theta(\cdot)$ and takes a random vector z as input: $\delta = g_\theta(z)$

To get past the shifts, starting with task shift FSAFW creates proxy tasks that mirror the shape of the few-shot tasks in meta-test time. For every proxy task, a linear classifier $f_{pc}(\cdot)$ is trained on the support set Sp and then the generator is optimized using the fooling loss:

$$L_{fool} = -\mathcal{H}(f_{pc}(f_e(x + g_\theta(z))), y),$$

(x,y) are the data from Qp, and H is the cross-entropy.

To address the semantic shift, FSAFW replaces the proxy linear classifier with class prototypes computed directly from the support set, removing the reliance on proxy labels and allowing the generator to better approximate the novel class distributions it will face at meta-test time:

$$c_k = \frac{1}{N_k} \sum_{y=k} f_e(x).$$

The fooling loss is then computed based on the distance between the perturbed query features and the class prototypes, using smoothed labels to avoid reliance on proxy labels:

$$L_{fool} = -\log \mathcal{H}(f_{proto}(f_e(x + g(z))), \hat{y}).$$

The proposed attack was evaluated on FSL models in Baseline, Baseline++, ANIL, R2D2, ProtoNet, DN4 on miniImageNet and TieredImageNet. For 5-way-1 shots tasks the ASR were 81,56% on Baseline, 77,84% on ANIL, 69,03% on ProtoNet and 73,31% on DN4. For 5-way 5-shot tasks the results were consistent, with ASR of 79,00% on Baseline and 78,31% on ANIL. The attack surface is query set, the attack happens in inference-time, the knowledge is white-box given, that the attacker must have access in the pre-trained encoder. Black-box UAP generation for FSL would be a realistic and very dangerous attack that remains underexplored, and similarly with clean-label attacks so does the question if UAP can persist enough through the adaptation step when the support changes in every new task [60].

4.6 Embedding Space Attacks:

Every previous family discussed was targeting either the query set or the support set by adding perturbations directly to the images. The adversary pollutes the images and depends on those to cause or disrupt misclassification. Metric-based FSL models do not classify by decision boundaries. They classify by comparing the feature vectors in an embedding space, with the similarity score between query set and support set prototypes is classifying. FAMF, the only attack in this category, comes with a different approach. Instead of perturbating the image itself, to attack the feature vectors.

Kim et al. in their work “FAMF: Robust Feature-Level Adversarial Attack

on Metric-Based Few-Shot Learning Models” propose FAMF. Instead of aiming at perturbing images, they perturb the feature vectors that have been produced by the embedding function f (the backbone that maps the input to a feature vector). To specify the difference between the two types, this is the standard image adversarial attack:

$$x^{adv} = x + \arg\max_{\delta} \mathcal{L}(\theta, x + \delta, y) \quad s.t.\ \|\delta\| \leq \epsilon,$$

The perturbation is added to the image and has to survive the entire process before it can affect the similarity function. FAMF skips that part and targets directly the output of the embedding function:

$$f(x)^{adv} = f(x) + \arg\max_{\delta} \mathcal{L}\big(\theta, f(x) + \delta, y\big) \quad s.t.\ \|\delta\| \leq \epsilon.$$

Where L is the loss function that is calculated with the similarity scores of query set and support set feature vectors. The goal of the attacker is to decrease the similarity between the vectors of the same class and increase the similarity between those of different classes. The way FAMF works is that instead of updating the image, it updates the feature vector. It starts with a feature vector that is clean, Q= f(Xq) and runs N iterations. In every repetition it counts the similarity between the current adversarial vector Qadv and the support features, it calculates how much the loss is increased and takes a small step in that direction similarly with what PGD does.

$$g \leftarrow \nabla_{Q_{\text{adv}}} L$$
$$Q_{\text{adv}} \leftarrow Q_{\text{adv}} + \alpha\, \text{sign}(g)$$

In each step the result is clipped to make sure the perturbation stays within limits ε from the original vector.

$$Q_{\text{adv}} \leftarrow Q + \text{clip}\big(Q_{\text{adv}} - Q,\ -\epsilon,\ \epsilon\big)$$

The suggested attack was evaluated by authors on metric-based models like ProtoNet, FEAT, DN4, ADM on both miniImageNet and Omniglot under N-way K-shot configurations and was compared with FGSM, C&W and PGD with PGD being far more efficient due to the way that it is designed from the other two. Despite PGD performing good, FAMF outperforms in every model in both datasets. The gap between them depends on the dataset. ProtoNet reached 96,89% ASR under PGD compared to FAMF 100% on miniImageNet. On Omniglot the difference is easier to notice with PGD dropping to 36,72% ASR on ProtoNet while FAMF remaining above 90%.

Omniglot:

TABLE 3. Comparison results with the existing attack methods using omniglot dataset.

Model	Original	FGSM		C&W		PGD		FAMF	
	Acc	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
ProtoNet	94.51	85.39	9.16	74.98	20.32	59.48	36.72	6.03	93.58
FEAT	75.92	59.93	21.06	22.92	69.80	54.11	28.73	0.00	100
DN4	99.12	96.24	2.90	97.33	1.80	87.64	11.58	42.73	56.89
ADM	98.93	24.81	73.46	67.04	28.29	0.00	100	0.00	100

MiniImageNet:

TABLE 4. Comparison results with the existing attack methods using mini-ImageNet.

Model	Original	FGSM		C&W		PGD		FAMF	
	Acc	Acc	Asr	Acc	Asr	Acc	Asr	Acc	Asr
ProtoNet	50.41	8.86	81.55	33.50	30.21	1.49	96.89	0.00	100
FEAT	33.87	13.51	60.10	23.19	31.54	10.61	68.68	0.00	100
DN4	59.58	12.46	77.33	49.32	10.33	0.31	99.43	0.12	99.78
ADM	62.33	5.56	91.08	56.46	9.42	0.01	99.97	0.03	99.95

This attack was also evaluated against Adversarial Querying defense, which is a FSL specific defense and trains the model by exposing it to adversarial query during meta-training, which we will discuss further in chapter 5. AQ provides very good results in PGD and ADM by reaching accuracy scores ~90%. While FAMF accuracy remains 0% or relatively close to it in all four models that it was tested. The defense is not working because it focuses on training the model to produce the correct vectors with images with perturbations. FAMF is attacking past that stage. It waits until the backbone has produced the feature vector and then corrupts it before the similarity function. AQ is defending a different part of the procedure. This is why at the time of writing, adversarial training based FSL defenses are unprepared for feature space attacks, though Chapter 5 will discuss a defense that operates at the feature level and may address this gap [61],[6].

This attack is another query set attack at the feature level this time, happens during inference-time, the attacker has white-box knowledge, and the goal is untargeted integrity. It's the only attack in this taxonomy that is not aimed directly at images. FAMF is limited when it comes to transferability, specifically by the authors, due to the need of full knowledge in the target model. FAMF in a black-box scenario is an open direction to move to.

4.7 Taxonomy Summary:

The following table will map all previously discussed attacks across the four taxonomy dimensions we split them into in Chapter 3. Each entry is mapped consistently with its attack surface, adversary knowledge, adversarial goal and timing

Paper	Family	Attack Surface	Adversary Knowledge	Adversarial Goal	Timing
FGSM (Goodfellow et al., 2015)	Evasion	Query Set	White-box	Untargeted Integrity	Inference-time
PGD (Madry et al., 2017)	Evasion	Query Set	White-box	Untargeted Integrity	Inference-time
C&W (Carlini & Wagner, 2017)	Evasion	Query Set	White-box	Targeted/Untargeted Integrity	Inference-time
AutoAttack (Croce & Hein, 2020)	Evasion	Query Set	White-box/Black-box	Untargeted Integrity	Inference-time
ASP (Oldewage et al., 2021)	Support Set Poisoning	Support Set	White-box	Untargeted Integrity	Inference-time
ASP v2 (Oldewage et al., 2022)	Support Set Poisoning	Support Set	White-box	Untargeted Integrity	Inference-time
MetaAttacker (Xu et al., 2021)	Support Set Poisoning	Support Set	White-box	Targeted/Untargeted Integrity	Inference-time
Kato et al. (2023)	Backdoor	Support Set	White-box	Targeted Integrity	Inference-time
FLBA / Liu et al. (2024)	Backdoor	Support Set	White-box	Targeted Integrity	Inference-time
Sana et al. (2025)	Backdoor	Support Set	Black-box	Targeted Integrity	Inference-time
FSAFW / Hu et al. (2024)	UAPs	Query Set	White-box	Untargeted Integrity	Inference-time
FAMF / Kim et al. (2026)	Embedding Space	Feature Space	White-box	Untargeted Integrity	Inference-time

Table 6: All attacks mapped across the four taxonomy dimensions.

This chapter organized the adversarial attacks in six families. Some of them were transferred directly from standard deep learning without any changes, others were designed for FSL and their unique surfaces. One of the families, clean-label poisoning at the meta-training level, remains unexplored, and no confirmed paper demonstrates it in FSL. The related FLBA and CVAE-encoded attacks operate on the episodic support set and are therefore classified under the backdoor family, not as meta-training clean-label poisoning.

With the above table, there are some observations to be made. First of all, every attack that is categorized is an integrity attack, because the adversary wants the model to produce wrong outputs not to make the model unresponsive. No confidentiality or availability attack was found, which means that the attacks with that goal are not considered or a harder target given how FSL models work. All attacks happen during inference time which is also another finding. For families 2,4 and 6 this is expected due to their target surfaces being available only in meta-testing time. But training-time threats remain a gap. Family 3 could be an attack family that will cover this gap, but there hasn't been any published work until now that successfully demonstrated an attack.

Another finding is that there were almost no black-box attacks found in the literature, apart from the Square Attack component of AutoAttack and the CVAE Encoding backdoor, which we classify as practical black-box because the attacker has no access to gradients or parameters but can poison the support set. Everything else is a white-box attack, which means the attacker has full access to gradients, architecture and parameters, or at least an exact surrogate. No strict query-only black-box attacks, where the adversary can only send queries without touching the support set, were observed. Gray-box attacks, which are a more realistic scenario in real-world threats, were also not observed. Chapter 5 will examine what defenses currently exist in FSL models and how those can be used in the attacks that have been discussed

4.8 Attack Families and the Adaptation Step:

Another way to notice the differences between the different attack families is their relationship with the adaptation step: Family 1 and 5 ignore it, 2 and 4 exploit it, Family 3 has to survive it, and Family 6 comes after the adaptation step. Family 1 and 5 perturb the query directly and leave the support set clean. Family 2 and 4 poison the support set so the model either adapts wrong or learns a trigger. Family 3 puts the poison in the meta-training data, so it has to survive the episodic training to still work in meta-testing time. Family 6 attacks the feature vector once the backbone has already run. So, the adaptation step is not just what makes FSL flexible, but also where most of its attack surfaces come from.

5. ADVERSARIAL DEFENSES:

In chapter 4 we discussed and presented some of the vulnerabilities of FSL models in adversarial attacks, targeting the query set, the support set and the feature space. This chapter will present the defenses that exist in FSL models in specific adversarial attacks. As in the attack section, some of the defenses were not designed for FSL since the beginning, but were adopted by standard deep learning, while others were specifically designed for FSL. After researching, the existing defenses can split into four categories: adversarial training, certified defenses and detection-based defenses and backdoor specific defense. Adversarial training is the most explored one by far and 7 of the defenses we will discuss will be part of that category to build robustness in the model in meta-training time. Certified defenses are not just showing that the model is robust but prove that the model will not misclassify as long as the perturbation is within an allowed size. Detection-based defenses are working in a different way. Instead of trying to make the model more robust, those are trying to detect adversarial inputs before they reach it. In standard deep learning these two directions have also been compared directly. Ziras et al. (2025) tested adversarial training against detection-based methods on intrusion detection data and found adversarial training to be more effective and consistent

defense against evasion attacks [84]. Eleven total defenses will be discussed. Many of the attack families with their relative attacks remain undefended. Support set, universal adversarial perturbations, embedding space and backdoor in which FLBA showed how the typical backdoor defenses are not effective on the attacks.

5.1 Adversarial Training Defense:

Adversarial training is among the most known defense categories against adversarial attacks, in standard deep learning as well as in FSL. The idea on how these defenses work is self-explanatory from the title. The model is not being trained only with clean examples, but instead adversarial examples are added in the meta-training process. This way the model learns to handle perturbations and becomes more robust to adversarial attacks. In standard DL, clean training samples are being replaced by adversarial ones. In FSL, adversarial examples are generated and injected in meta-training episodes and the model learns how to handle those attacks at the same time when it learns how to generalize on tasks. The defenses in this category are a somehow historical overview on how the defenses evolved in the past few years. Some methods in this category focus on injecting adversarial examples in the query set, while others focus on building robust feature representations that are resistant to perturbations.

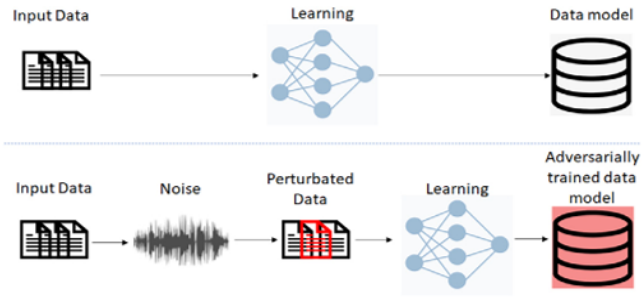

Fig. 4. Traditional VS adversarial training.

Figure 10: Traditional VS adversarial training. [73]

Yin et al. (2018) introduced ADML (Adversarial Meta-Learner), which was one of the first attempts for adversarial robustness in meta-learning. ADML is a model-agnostic meta-learning algorithm that includes both clean and adversarial samples in the meta-training process. The way ADML works is by training the model to work in an adversarial way. In the inner loop the model updates towards the adversarial sample and evaluates on clean ones and at the same time it does the opposite. This method can be applied in any model that is being trained by gradient descent. The writers also argue that simply mixing adversarial and clean in the training, like MAML-AD, doesn't work efficiently. They call this "arm wrestling" between inner loop and outer loop, forcing the model to learn both clean and adversarial examples.

$$\min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta} - \alpha_1 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{adv_i})}, \mathbf{D}'_{c_i})$$

$$\min_{\boldsymbol{\theta}} \sum_{\mathcal{T}_i \sim \mathcal{T}} \mathcal{L}_i(f_{\boldsymbol{\theta} - \alpha_2 \nabla_{\boldsymbol{\theta}} \mathcal{L}_i(f_{\boldsymbol{\theta}}, \mathbf{D}_{c_i})}, \mathbf{D}'_{adv_i})$$

The two formulas work in the opposite way. The first one is using adversarial examples and is evaluated on clean ones and the second updates using clean examples, evaluating on adversarial. The model learns to handle both. The adversarial examples have been generated using FGSM and the method is transferable to any model that is trained by gradient descent.

The defense method was tested in MAML in a 5-way 5 -shot scenario and with ε=2. MAML accuracy drops from 61,45% to 36,65% under attack. While ADML only drops from 59,38% to 57,03% when the support set is clean, but the query set has adversarial samples, the defense was tested on miniImageNet. They used FGSM to generate the adversarial samples which is a single step attack [62].

Li et al. (2019) introduced the Defensive Few-Shot Learning (DFSL) framework, which identified two FSL challenges that previous adversarial training like ADML did not address. In standard DL, model trains on some classes and gets tested on the same ones. FSL trains on some classes and contain different ones in meta-testing time that are unseen. This means that defenses which the model learned on base classes on meta-training time, do not work instantly in new classes in meta-testing time, because the model never saw the classes there. As a result, the adversarial patterns it learned to handle might look completely different on the new classes images. Secondly, standard DL defense methods use tight regularization that work fine with large datasets, but in FSL the test set has a different distribution from the training set and with the few samples, the tight criteria make it more difficult for the model to generalize.

To address the two problems, DFSL builds on episodic adversarial training with adversarial examples but extends it by adding distribution consistency criteria in each episode. The objective of the episodic-based training is:

$$\Gamma = \arg\min_{\varphi,\theta} \sum_{i=1}^{n} \sum_{\boldsymbol{x}\in\mathcal{Q}_i} \left(\mathcal{L}^{\mathrm{CE}}(\boldsymbol{x}, \mathcal{S}_i, y) + \max_{\boldsymbol{x}^{\mathrm{adv}}=\boldsymbol{x}+\boldsymbol{\delta}} \mathcal{L}^{\mathrm{CE}}(\boldsymbol{x}^{\mathrm{adv}}, \mathcal{S}_i, y) \right)$$

In every episode, the model minimizes the loss on both clean and adversarial query examples. To evaluate this framework, DFSL adopts five DL defenses to work in the FSL episodic training process (AT, VAT, ALP, ATDA, TRADES). AT which is the most common defense, by mixing clean and adversarial examples during training. VAT that uses KL divergence to minimize the difference between clean and adversarial examples. ALP which enforces similar logit representations between clean and adversarial examples. ATDA which minimizes the distribution gap between clean and adversarial examples using domain adaptation (a technique that treats clean and adversarial examples as two different domains and trains the model to bridge the gap between them). TRADES which enforce that the model's predictions on adversarial and clean examples are similar through a regularized surrogate loss. While those five methods were adapted from standard DL and brought into FSL, DFSL also introduces three distribution consistency measures, Kullback-Leibler divergence-based distribution measure (KLD), Task-conditioned distribution measure (TCD), Symmetric Kullback-Leibler divergence measure (SKL). The idea in these measures is that when the model processes a clean image and its adversarial version, it will produce two different feature vectors. Those three try, as much as possible, to make those features vectors similar within each episode, so the classifier sees the same thing if it was being attacked or not. What makes DFSL special is that it can generalize, it separates the embedding module and the classifier module, so every FSL method like ProtoNet, RelationNet or DN4 can be integrated into it by replacing the classifier module. That is why they say this is a framework instead of a defense method, because it can make many FSL methods robust without redesigning the defense.

The defense was evaluated on miniImageNet under 5-way 5-shot at ε=0,01 without any defense DN4 drops to 8,65% adversarial accuracy. DFSL-DN4-AT recovers this to 54,97% while maintaining 71,54% clean accuracy. The best variant, DFSL-DN4-SKL, reaches 62,68% adversarial accuracy while maintaining 71,68% clean accuracy. Under PGD attack on miniImageNet DFSL-DN4-TCD reaches 58,83% accuracy while maintaining 68,43% clean accuracy. [63]

Goldblum et al. (2020) introduced Adversarial Querying (AQ), which is a big contribution in the field, and has been mentioned through this thesis in chapters 2 and 4. AQ is an algorithmic agnostic which doesn't require specific architecture to work. Where the model is MAML, ProtoNet, R2-D2, as long as it updates its parameters through gradient descent, AQ can be applied without changes to the architecture. The way it works is by

injecting PGD created query images into every episode instead of clean query images. The model is trained to minimize the loss of the injected adversarial queries after adapting to the clean support set.

$$\min_{\theta} \mathbb{E}_{S,(\mathbf{x},y)} \left[\max_{\|\delta\|_p < \epsilon} \mathcal{L}(F_{A(\theta,S)}, \mathbf{x} + \delta, y) \right]$$

Where S is the support set, A(θ,S) the fine-tuning algorithm, θ the model parameters. The objective finds a parameter initialization that, when fine-tuned on the support set, minimizes the expected query loss against an attacker.

AQ only attacks the query data and not the support data. The authors point out that attacking support sets during training doesn't improve robustness but increases computational cost.

The defense was evaluated on ProtoNet, R2-D2 and MetaOpNet on minImageNet and CIFAR-FS. Trained naturally ProtoNet achieves 70,23% clean accuracy but drops to 0,00% under PGD attack. ProtoNet AQ recovers robust accuracy to 27,99% while maintaining 52,04% clean accuracy on 5-shot miniImageNet. The best result is R2-D2 AQ which achieves 31,52% robust accuracy while maintaining 57,87% clean accuracy. AQ became the base for adversarial defenses on FSL.

AQ defends against query set evasion. But in chapter 4 under FAMF we did show the results of the authors on AQ having almost no effect upon that feature space attack, AQ makes the backbone more robust, while FAMF operates after that step [6].

Wang et al. (2021) proposed a new defense to make MAML models more robust. MAML is working in two stages, the inner and the outer loop as mentioned in chapter 2. The inner loop fine-tunes the support set for each task, while the outer loop updates the meta-learner based on the fine-tuned support set. To acquire a more effective defense someone would guess that both loops need to be more robust.

The authors Wang et al. disagree and prove that making more robust the outer loop (meta-updater) is enough. Because if the point the model starts from is already robust, the inner loop fine-tuning will preserve that robustness instead of destroying it**.** They confirm this by checking the neuron activation maps, R-MAML_out neurons pick up object outlines and high-level features, standard MAML neurons do not, and this stays the same even after fine-tuning. They call this R-MAML_out. To control how much

adversarial regularization is applied at each stage, they introduce two hyperparameters γ_out and γ_in :

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{N}\sum_{i=1}^{N}[\ell_i'(\mathbf{w}_i';\mathcal{D}_i') + \gamma_{\text{out}}\mathcal{R}_i(\mathbf{w}_i';\mathcal{D}_i')] \\ \text{subject to} \quad & \mathbf{w}_i' = \arg\min_{\mathbf{w}_i}[\ell_i(\mathbf{w}_i;\mathcal{D}_i,\mathbf{w}) + \gamma_{\text{in}}\mathcal{R}_i(\mathbf{w}_i;\mathcal{D}_i)],\ \forall i \in [N] \end{aligned}$$

The choice of γ_in and γ_out plays a role in determining that variant. If both are greater than zero, we have R-MAML_both, which is making both stages more robust. If γ_in=0 and γ_out>0 it gives R-MAML_out, which is the paper's main finding, adversarial regularization in meta-update stage while meta-test remains standard.

Adversarial Querying which was discussed before is a special case of R-MAML_out in which γ_out is set to infinity, the meta-update is overridden by the adversarial training regularization. That is why AQ gets strong robust accuracy but pays a large clean accuracy cost. R-MAML_out, keeps both objectives active with a finite γ_out. On miniImageNet 1-shot 5-way, R-MAML_out reaches 40,9% clean accuracy and 22,9% robust accuracy compared to AQ's 29,6% and 24,9%. The robust accuracy is 2% lower but the clean accuracy difference is 11%.

The paper also introduces two more computational efficient versions of R-MAML. R-MAML_out-FGSM in which the PGD is being replaced with FGSM during the meta-update and trains faster while improving robustness to 23,04%. Unlike FGSM, which takes a single deterministic step every time, PGD starts from a random point within the ε-ball and refines the perturbation over multiple steps, making it stronger but more expensive. R-MAML_out-ANIL keeps the backbone fixed at meta-test and adapts only the classifier (ANIL = Almost no inner loop, simplified version of MAML that doesn't fine tune the backbone), which is the cheapest one computationally but also has the weakest clean accuracy by having 37,46% clean accuracy and 22,7% robust accuracy. R-MAML_out-TRADES is another extension of R-MAML_out with unlabeled data augmentation on top of TRADES regularization, mining additional unlabeled data from ImageNet to match the miniImageNet distribution. It reaches 37,1% clean accuracy and 25,51% robust accuracy. It is worth noting that the unlabeled data augmentation is dataset dependent. For miniImageNet they mine from ImageNet, for CIFAR-FS from STL-10, so this approach requires also a suitable unlabeled source for every dataset.

The last version of R-MAML_out is the R-MAML_out-CL which is also the strongest one. It extends the meta-update objective with contrastive learning loss over both clean and adversarial "views" of the query data:

$$\begin{aligned} \underset{\mathbf{w}}{\text{minimize}} \quad & \frac{1}{N}\sum_{i=1}^{N}\left[\ell_i'(\mathbf{w}_i';\mathcal{D}_i') + \gamma_{\text{out}}\mathcal{R}_i(\mathbf{w}_i';\mathcal{D}_i') + \gamma_{\text{CL}}\ell_{\text{CL}}(\mathbf{w}_{\text{c},i}';p_i^{+}\cup p_i^{\text{adv}})\right] \\ \text{subject to} \quad & \mathbf{w}_i' = \arg\min_{\mathbf{w}_i}\ell_i(\mathbf{w}_i;\mathcal{D}_i,\mathbf{w}),\ \forall i \in [N], \end{aligned}$$

The three terms are the clean classification loss, the robust regularizer and the contrastive loss. The contrastive loss pulls together features of the same sample under different augmentations and pushes apart features of different samples, with adversarial examples counted as additional views. This makes the backbone learn features that are both class-aware and robust. R-MAML_out-CL was evaluated on miniImageNet, CIFAR-FS and Omniglot. On miniImageNet it reaches 38,6% clean accuracy and 26,81% robust accuracy, a 9% clean accuracy and ~2% robust accuracy improvement over AQ, with consistent results across CIFAR-FS and Omniglot.

R-MAML is a MAML-specific robust method designed for gradient-based meta-learners, and is not directly applicable to metric-based methods like ProtoNet or Matching Networks. Like AQ, it doesn't cover attacks that occur in the support set, backdoor attacks or feature space attacks. R-MAML trains the backbone to be robust in input perturbations, and as shown in section 4.6, this does not protect against attacks that operate after the backbone has already run like FAMF [64], [6].

Dong et al. (2022) proposed a different approach compared to the previous ones. Instead of focusing on meta-learning-based approaches like ADML, AQ and R-MAML, which rely on episodic task sampling during training, they train a single robust embedding model on the full training set and reuse it for few-shot tasks. The authors argue that training in many few-shot tasks causes the model to overfit and drop its robustness on the new unseen classes that it tries to classify. They instead show that a robust embedding trained on the full training set without any task sampling transfers well in FSL and they compare it with existing FSL defenses. They also propose three more modules to further push the robustness of the model.

The baseline is standard adversarial training where clean images are replaced with PGD generated adversarial examples in the cross-entropy loss and is defined:

$$\mathcal{L}_{bl} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \left[-\log \frac{\exp(f_\theta\,(\mathbf{x}+\delta)_y)}{\sum_{i=0}^{N_{train}} \exp\left(f_\theta\,(\mathbf{x}+\delta)_i\right)} \right]$$

They add three extra mechanisms. The first one is adversarial-aware module in which they add an auxiliary classifier that learns to tell the clean and adversarial features. By learning to tell them apart the model picks up on feature-level differences that standard adversarial training ignores. The auxiliary predicts if the input is clean and outputs 0 or adversarial 1.

$$\mathcal{L}_{aa} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[-\log \frac{\exp(f_{\phi,\omega}(\mathbf{x})_0)}{\sum_{i=0}^{1}\exp\left(f_{\phi,\omega}(\mathbf{x})_i\right)} - \log \frac{\exp(f_{\phi,\omega}(\mathbf{x}+\delta)_1)}{\sum_{i=0}^{1}\exp\left(f_{\phi,\omega}(\mathbf{x}+\delta)_i\right)}\right]$$

By adding this on top of the adversarial training, the model learns the feature level differences better than with adversarial training alone.

The second mechanism they suggest is adversarial-reweighted training. Not all adversarial examples are equally dangerous to the model, so this mechanism assigns different weights depending on how much the adversarial example disrupts the model. Previous defenses used gradient ascent steps to measure this, but the authors say that relying only on that gives a rough estimation of how adversarial the example is. They instead propose a combination of minimum steps needed to fool the model with the loss variation during adversary generation within the current batch, which yields a more precise weight per example. Examples that strongly disrupt the model, get higher weight:

$$w_\theta(\mathbf{x}) = \alpha\frac{1+\tanh\left(4-10t(\mathbf{x})/T\right)}{2} + \beta\frac{v}{\max\limits_{V_i\in V}(V_i)} \quad ($$

where t(x) is the least number of gradient ascent steps needed to change the prediction and v is the loss variation value. The reweighted training loss becomes:

$$\mathcal{L}_{ar} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\left[-w_\theta(\mathbf{x})\log\frac{\exp(f_\theta(\mathbf{x}+\delta)_y)}{\sum_{i=0}^{N_{train}}\exp\left(f_\theta(\mathbf{x}+\delta)_i\right)}\right] \quad ($$

The third mechanism is the feature purifier, which is the most unique part of this defense because it is the only component that operates at inference time. While every other defense method discussed so far in this adversarial training section operates only at training time, this mechanism is an inference-time feature purification module. It takes the feature vectors that were produced by the backbone either clean or adversarial and pulls the adversarial towards the clean feature distribution using mean squared error loss:

$$\mathcal{L}_{fp} = \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}}\Big[\mathcal{L}_{MSE}\left(f_{\phi,p}\left(\mathbf{x}\right), f_{\phi}\left(\mathbf{x}\right)\right) + \mathcal{L}_{MSE}\left(f_{\phi,p}\left(\mathbf{x}+\delta\right), f_{\phi}\left(\mathbf{x}\right)\right)\Big]$$

The overall training loss combining adversarial awareness loss, adversarial reweighted loss and the feature purification loss can be calculated:

$$\mathcal{L} = \mathcal{L}_{ar} + \lambda_1\mathcal{L}_{aa} + \lambda_2\mathcal{L}_{fp}$$

The defense was evaluated on miniImageNet and CIFAR-FS against FGSM, PGD and C&W with three backbones. On miniImageNet 1-shot with ResNet12, it reaches 45,81% clean accuracy and 35,18% PGD robust accuracy, compared to R-MAML's 37,52% and 27,46% and AQ's 41,89% and 20,53%. They also made a table of how much each component contributed to the robustness of the model. It shows that every component plays a role. The baseline alone reaches 40,59% percent clean and 29,13% percent under PGD attack and adding all three modules brings it to 45,81% clean and 35,18% [65].

Like every other defense in this adversarial training section, this is a defense method for query-set evasion attacks. While support set poisoning and backdoor attacks are out of scope in this section. The third component, the feature purifier, operates at feature level in inference time. While they are not testing this attack space in this paper, it could be a possible defense for FAMF.

Nayak et al. (2022) proposed another unique defense that works in fundamentally different way than any previous defense mentioned. The other defenses so far ADML, DFSL, AQ, R-MAML and the one Dong et al. proposed, made the models more robust by training with adversarial examples. Nayak et al. on the other hand claim that it has higher cost computationally to generate adversarial examples in training phase and propose a different direction which was inspired by human cognition. In image processing, low-frequency represents slow changes across pixels, like smooth areas, large shapes, and colors of objects, while high-frequency represents fast changes, sharp edges. Adversarial perturbations are small rapid pixel changes which is why they appear in the high-frequency part of the image. Humans rely on low-frequency information the images give to recognize objects, while deep models usually rely on high-frequency information. If the model is making decisions based on the low-frequency information, adversarial

perturbations, that are high-frequency, corrupt part of the image that the model has learned to ignore.

Low-frequency information is obtained from the Fourier transform, which converts an image from pixels to a frequency representation. By only keeping the frequency components within a certain radius r and converting back to pixel representations, the low frequency of the image obtained:

$$xl_{ir} = FT^{-1}(zl_{ir}) \quad ; \quad zl_{ir} = z_i \circ F^{mask_{lr}}$$

where F_{lr}^{mask} is a circular mask that keeps only frequencies within radius r from the center of the frequency representation

The defense works in two stages: training and finetuning

In training, A teacher network T is trained on the base classes by minimizing cross-entropy loss. A student network S with the same architecture is then trained on low-frequency versions of the same images, using cosine similarity to match the teachers' feature responses on the original images:

$$\theta_s^* = \min_{\theta_s}(L_{ce} - L_{cs});\ L_{cs} = \frac{1}{b}\sum_{i=1}^{b} \frac{S(xl_i^B)^T T(x_i^B)}{\left\|S(xl_i^B)\right\| \left\|T(x_i^B)\right\|} \quad (3$$

The teacher network provides rich class-aware features that guide the student network to be as class-aware on low-frequency images as the teacher is on originals. If this step didn't exist it would lead to performing poorly because low-frequency images contain less information. It's crucial to select the right radius r because if it is too high the adversarial contamination survives (perturbation still in the image), if it's too low the images become too blurry to tell classes apart. To solve this they use Progressive Learning, which starts at a high radius and gradually shifts to lower radius as the model becomes accurate enough at each level automating the radius selection in a data-driven way.

In fine-tuning stage, the pre-trained student network is fine-tuned on the novel class support set. Cosine similarity is applied to the query set and its low-frequency versions to improve discriminability on the new classes

$$L_{cs} = 1/n_q \sum_{i=1}^{n_q} \frac{S(xl_{ir}^{N_q})^T S(x_i^{N_q})}{\left\| S(xl_{ir}^{N_q}) \right\| \left\| S(x_i^{N_q}) \right\|}$$

At the evaluation they don't classify on a single radius, instead the model produces logits at multiple radius and combines through a weighted combination using the weight distribution learned during training:

$$Pred = \text{argmax}(\sum_{r=r_{min}}^{r_{max}} (logit_r \cdot \bar{w}_r^{r^{range}}))$$

The defense was evaluated on CIFAR-FS and miniImageNet against PGD and AutoAttack. On CIFAR-FS 1-shot without any defense the accuracy was 68,4% and it dropped to 0,01% under PGD. With this method it had similar clean accuracy with 65,03% and under PGD 60,55% and 62,05% under AutoAttack. On miniImageNet 1-shot it reaches 54,05% clean and 53,63% PGD compared to AQ-R2D2's 37,91% and 20,59%. Training time is approximately 5 times faster than AQ methods.

This is another query set evasion defense method, and it works because the model is trained to rely on low-frequency information which is why high-frequency perturbations have no effect. An attack targeting low-frequency information would possibly bypass this defense method [66].

The next defense we are going to discuss came from Dong et al. (2024), it's called RESISTANCE. It works in a different way than any previous method discussed and by the way it works it is also more robust. Previous defenses treated separately similarity learning and class concept learning. Similarity learning is the episodic way that AQ and R-MAML used to compare support and query features across tasks, class concept is the approach Dong et al. (2022) that trains the full dataset with class labels. The two methods achieve different results when it comes to robustness. RESISTANCE uses the fact these two are completing each other to create a stronger defense.

The defense trains two models at the same time with the same backbone architecture but different classification heads, each operating on a different label space. The first model is a similarity-based meta-learner like AQ using episodic prototypical training with adversarial samples:

$$\mathcal{L}_s(\mathcal{Q}, \mathbf{M}) = \frac{1}{|\mathcal{Q}|} \sum_{(\mathbf{x}, y_\mathbf{x}) \in \mathcal{Q}} \left[\mathcal{L}_{\text{CE}}(\mathbf{p}_\mathbf{x}^\mathbf{M}, \mathbf{y}_\mathbf{x}) + \lambda \max_{\|\boldsymbol{\delta}_s\|_\infty < \epsilon} \mathcal{L}_{\text{KL}}\left(\mathbf{p}_\mathbf{x}^\mathbf{M} \| \mathbf{p}_{\mathbf{x}+\boldsymbol{\delta}_s}^\mathbf{M}\right) \right]$$

The second model is a standard adversarially trained classifier on the base classes similar to the defense Dong et al. (2022) proposed:

$$\mathcal{L}_c(\mathcal{B}, \mathbf{W}) = \qquad ($$
$$\frac{1}{|\mathcal{B}|} \sum_{(\mathbf{x}, y_\mathbf{x}) \in \mathcal{B}} \left[\mathcal{L}_{\mathrm{CE}}(\mathbf{p}_\mathbf{x}^\mathbf{W}, \mathbf{y}'_\mathbf{x}) + \lambda \max_{\|\boldsymbol{\delta}_c\|_\infty < \epsilon} \mathcal{L}_{\mathrm{KL}}\left(\mathbf{p}_\mathbf{x}^\mathbf{W} \| \mathbf{p}_{\mathbf{x}+\boldsymbol{\delta}_c}^\mathbf{W}\right) \right]$$

There is also a third copy of the backbone called the unified embedding. After each training iteration, the weights of both previous models are averaged into this unified backbone using Exponential Moving Average, and every few iterations those averaged weights are copied back into both branches:

$$\boldsymbol{\theta}_u := \beta \boldsymbol{\theta}_u + (1-\beta)\left[\gamma \boldsymbol{\theta}_s + (1-\gamma)\boldsymbol{\theta}_c\right].$$

Where β is the decay rate, γ balances the impact of each branch so the unified embedding absorbs robustness by both branches at the same time.

Instead of initializing PGD from random noise like previous methods, they compute one shared perturbation per class before each episode. This perturbation is designed to push all images of that class away from their class prototype in all three backbones at the same time. They call this cross-branch class-wise Global Adversarial Initialization Perturbations (GAIP). Because both branches start from the same adversarial base, the adversarial examples they generate are more effective than starting from random points.

The branch robustness harmonization module measures per episode which branch is currently less robust, meaning whose predictions change more between clean and adversarial inputs. That branch gets a larger learning rate so it can catch up:

$$\eta'_s = \eta_s \left[1 - \tanh\left(\tau \max\left(0, \log\left(\kappa_s\right)\right)\right)\right]$$

where ηs is the original learning rate of the similarity branch, η's is the corrected one, κs is the relative robustness score between the two branches and τ controls how aggressively the learning rate is adjusted.

At test time the two heads are discarded and only the unified backbone is used as the embedding, with a simple classifier built from the support set of each episode.

RESISTANCE was evaluated on miniImageNet, CIFAR-FS and FC100 against PGD, C&W and AutoAttack. On miniImageNet 1-shot with ResNet-12 it reaches 50,28% clean accuracy and 33,71% AutoAttack robust accuracy, compared to Dong et al. (2022) at 45,81% clean and 32,61% AutoAttack. On CIFAR-FS 5-shot with ResNet-12 it reaches 74,83% clean and 58,76% AutoAttack, a 9% improvement in clean accuracy and 7,8% in AutoAttack robust accuracy over the previous state of the art. Results are consistent across FC100 and both Conv-4 and ResNet-12 backbones [13], [65].

Like all adversarial training defenses, RESISTANCE covers query-set evasion attacks only.

After discussing all seven adversarial defenses in this category, it becomes clear that all of them only cover against query-set evasion attacks (FGSM, PGD,C&W, AutoAttack) and none addresses the issue with support set attacks nor backdoor attacks or feature space attacks. Many methods have been tried across the different defenses such as robustness from episodic adversarial training in ADML and DFSL, embedding approach from Dong et al. (2022) to a combination of similarity and class concept learning in RESISTANCE, but all aim to defend against query set attacks.

One evaluation issue worth noting is also that comparing these defenses is not straightforward. Most defenses report clean accuracy and robust accuracy separately, but a method with higher clean accuracy and lower robust accuracy cannot simply be ranked above one with the opposite profile. DFSL defense introduced also the Fβ score which combines both into a metric where β is the parameter that controls the weight to robust accuracy over clean accuracy. This is a good evaluation metric for future FSL robustness work because it provides a fairer way to compare defenses. [63]

5.2 Certified Defenses:

Certified Defenses is a completely different approach than adversarial training. Instead of training the model to make it more robust in adversarial attacks, those defenses prove mathematically that the model's prediction cannot change while the perturbation stays within a specific radius. This is not proven with experimental observations like the previous methods but instead with mathematical certainty. The mechanism used in both papers in this section is by randomized smoothing (Cohen et al. 2019), by adding a Gaussian noise to the input many times to take the average of the results to later compute the certified radius. Certified methods usually get lower accuracy in both clean and robust and the radius is small. Which means that the certified defense will work only for weak perturbations, but within that range they achieve high robustness [67].

The first theoretical robustness guarantee for FSL came from Pautov et al. (2022). Standard randomized smoothing works for classifiers that output class probabilities, but metric-based FSL models output embeddings and classify by distance to prototypes. This makes standard randomized smoothing not transferable to FSL, which is the gap this paper addresses.

They adapted randomized smoothing to work in embedding space. Instead of classifying the inputs embedding it runs through the backbone n amount of times with different Gaussian noise added to the input and then average the result embeddings. This averaged embedding is the smoothed embedding g(x) :

$$\mathbb{E}_{\varepsilon \sim \mathcal{N}(0,\sigma^2 I)} f(x+\varepsilon)$$

From the smoothed embedding they calculate two things. How stable the embedding is (Lipschitz L) which tells you how much the maximum amount g(x) can change in one input image. They prove:

$$L = \sqrt{\frac{2}{\pi\sigma^2}}.$$

So the more the noise σ the more stable the embedding becomes. They also calculate how far the smoothed embedding sits from the nearest decision boundary between the two closest prototypes c_1 and c_2 in the embedding space. This is called adversarial embedding risk γ:

$$\gamma = \|\Delta\|_2 = \frac{\|c_2 - g(x)\|_2^2 - \|c_1 - g(x)\|_2^2}{2\|c_2 - c_1\|_2^2},$$

These two values are what give the certified radius r = γ/L. If the perturbation of the attacker is smaller than that r, the model won't change its prediction, it is mathematically impossible. If the model can't decide which prototype is closer, it skips classification instead of guessing.

This defense was evaluated on Cub-200-2011, CIFAR-FS and miniImageNet with Protonet and Conv4 backbone in 1-shot and 5-shot, as most of the defenses described. The writers reported the results in curves, this time on different perturbations and noise levels. Higher σ increases robustness but drops clean accuracy.

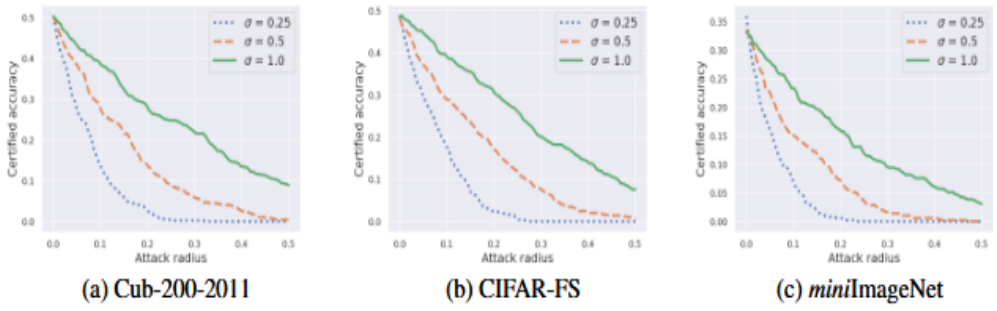

Figure 11: Dependency of certified accuracy on attack radius ε for different σ, 1-shot case, n = 1000. [74]

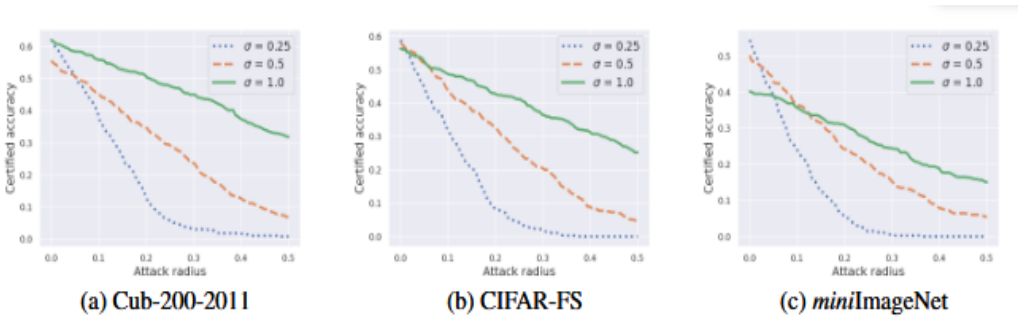

Figure 12: Dependency of certified accuracy on attack radius ε for different σ, 5-shot case, n = 1000. [74]

As said, this method provides a provable guarantee instead of empirical and tested robustness. This guarantee only applies to L2 bounded perturbations on query images and doesn't cover support set attacks, backdoor attacks or feature space attacks [68].

FCert (Wang et al. 2024) is the only defense until now that addresses a gap left by all previous defenses, support set poisoning. It doesn't change the model to be more robust like previously. It directly changes how classification happens in testing time.

The two observations the authors based this idea on are that: first, a foundation models feature vector for a test input will be closer to support samples of the correct class than to those of the other classes. Secondly, if the poisoned data are T samples of a K total, the clean support samples would be K-T. FCert computes a robust distance by sorting the distances between the test input and each support sample, removing the K' largest and smallest, and averaging what remains.

On every new test input Fcert uses a foundation model like CLIP or DINOv2 as a feature extractor and computes the distance between the test and every support sample in each class. The difference with what ProtoNet would do is that instead of averaging all of them it sorts them and dismisses the K' largest and K' smallest and then averages what's left. K' is a hyperparameter that controls how many distances are trimmed from each end:

$$\underline{R}_p^c(T^c) = \frac{\sum_{i=K'+1-T^c}^{K-K'-T^c} d_i^c}{K - 2 \cdot K'}.$$

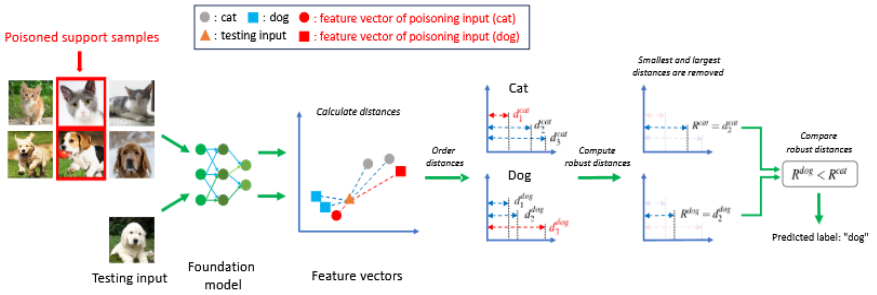

Figure 2: Overview of FCert under Individual Attack. We have three support samples for each of the two classes (i.e., 2-way-3-shot classification). An attacker could poison one support sample for each class, where the feature vectors with red color are for poisoned support samples. $d_1^{cat}, d_2^{cat}, d_3^{cat}$ (or $d_1^{dog}, d_2^{dog}, d_3^{dog}$) are distances between the feature vectors of three support samples whose labels are "cat" (or "dog") and the testing input, which are used to compute two robust distances R^{cat} and R^{dog}. Our FCert still predicts the correct label "dog" for the testing input under two poisoned support samples.

Figure 13: Overview of FCert under Individual Attack. [69]

The class with the smallest robust distance is where the input will be classified. That's because the attacker will try to misclassify by either pulling poisoned support set close to test input or by pushing it too far away, which are both edges that get removed after the sorting. The certified poisoning size T* depends on K', if the attacker poisons more samples than K', poisoned samples survive the removal and the guarantee breaks.

They evaluated this on CUB200-2011, CIFAR-FS and tieredImageNet with CLIP and DINOv2. And as shown in the figures it maintains higher certified accuracy from Bagging,

DPA and k-NN as the poisoning size T increases, while achieving high clean accuracy like ProtoNet without any attack.

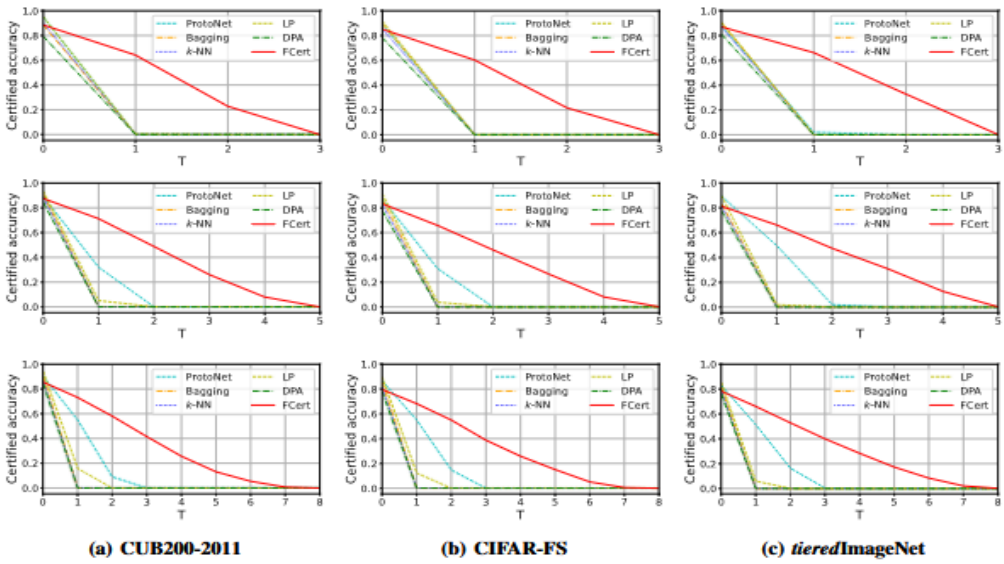

Figure 14: Comparing the certified accuracy of FCert with existing provable defenses (or empirical accuracy of existing few-shot learning methods) for C-way-K-shot few-shot classification with CLIP. The attack type is individual attack.K = 5, C = 5 (first row); K = 10, C = 10 (second row); K = 15, C = 15 (third row). T is poisoning size. [69]

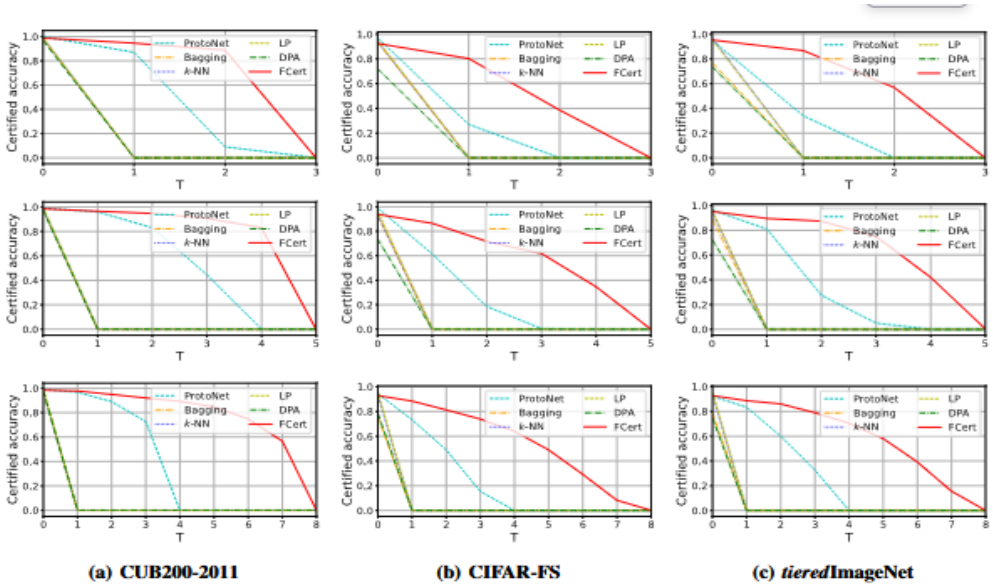

Figure 15: Comparing the certified accuracy of FCert with existing provable defenses (or empirical accuracy of existing few-shot learning methods) for C-way-K-shot few-shot classification with DINOv2. The attack type is individual attack. K = 5, C = 5 (first row); K = 10, C = 10 (second row); K = 15, C = 15 (third row). T is poisoning size. [69]

This defense is the first in this thesis that fills the gap between support set poisoning attacks and model robustness with a provable guarantee. It uses a foundation model as a feature extractor, so it can't be used in FSL in which the backbone is trained from the beginning. Also, they assume that the foundation model is trusted and that if an attacker can corrupt pre-training data of CLIP or DINOv2 that guarantee breaks [69].

A newer 2025 defense, LeFCert, builds on FCert but instead of using only the visual embeddings, it also uses the text embeddings from foundation models and blends the two together. It still gives a certified guarantee against poisoning through a trimmed-mean prototype, and it's the first certified defense that works on multimodal few-shot models [70].

Both defenses in this category differ from category A, not by the different component they cover but by the mechanism they work. This category provides a guaranteed robustness. Pautov et al. prove that no adversarial perturbation can change the model's prediction within a certified radius. FCert proves that support set poisoning below T* won't change the prediction of the model. One is query set defense the other support set.

5.3 Detection-Based Defense:

This category is different from both adversarial training and certified defenses. It doesn't make the model more robust nor provide mathematical guarantees, instead they focus on identifying if the input to the model has been manipulated in the first place. If the defense detects adversarial samples, it raises an alert similarly to an Intrusion Detection System. While detection-based defenses don't really improve models accuracy under attack, they are more flexible and attack-agnostic, which means those can detect attacks they were not trained against.

Tan et al. (2021) proposed this detection-based defense against adversarial support samples in FSL. They proposed a method using self-similarity in which adversarial support samples are crafted to be pushed in a different direction in feature space while remaining consistent among themselves, so adversarial samples are classifying correctly among each other but misclassify clean query images of the same class. When a filtering function is applied to remove the adversarial perturbations the consistency breaks, clean images survive filtering without much change. This difference in behavior before and after filtering is how the defense work to detect adversarial support sets.

The defense mechanism is based on three components. First, the support set of a class is randomly split into an auxiliary support set and an auxiliary query set. Second, a filtering function r(·) is applied to the auxiliary support set to remove potential adversarial perturbations. Third, the adversarial score is computed as the L1 norm difference between the logits of the auxiliary query before and after filtering:

$$U_{adv} = \| h(r(S^c_{aux}), Q^c_{aux}) - h(S^c_{aux}, Q^c_{aux}) \|_1,$$

The support set is flagged as adversarial if the adversarial score goes above a threshold.

The authors explored four filtering functions: a Feature-space Preserving Autoencoder (FPA), median filtering (FeatS), Total Variation Minimisation (TVM), and Bit Reduction

(BitR). FPA reconstructs images while preserving their feature space representation. The other three are simple image processing filters with different mechanisms.

The defense was evaluated on miniImageNet and CUB with RelationNet, Prototypical Network and Cross-Attention Network under PGD and CW-SGD attacks of varying strengths. Results are reported as AUROC scores. FPA achieves near-perfect detection with AUROC scores of 0,997-0,999 for PGD attacks across all models and datasets. Performance drops slightly when only one support sample is attacked rather than all of them, which is expected since a single adversarial sample is harder to detect in a small support set.

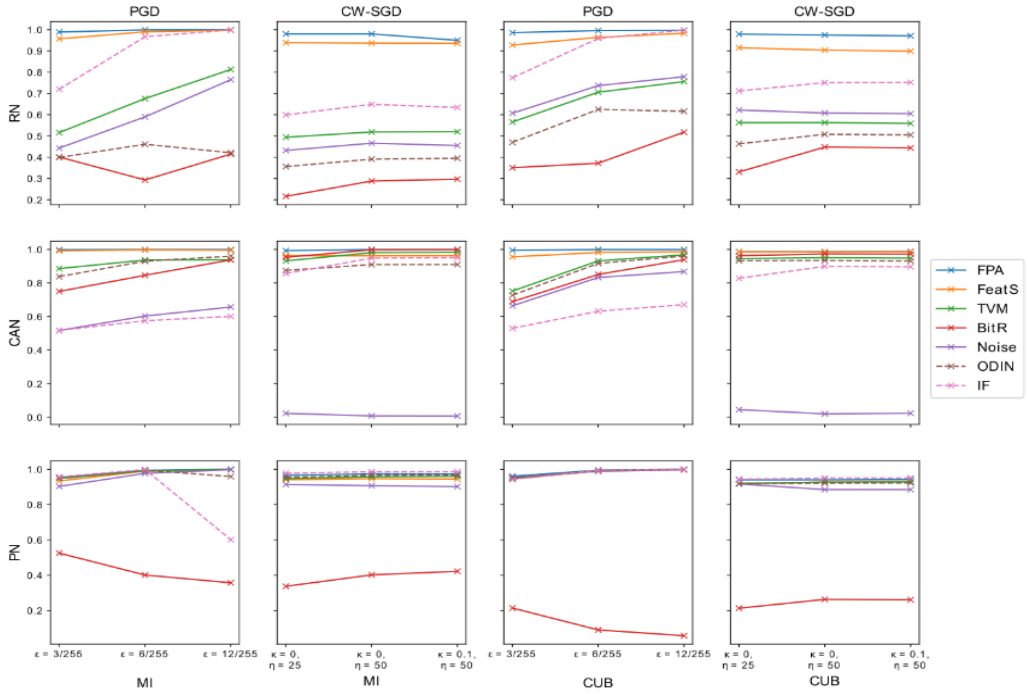

Figure 16: Area Under the Receiver Operator Characteristic (AUROC) scores for the various filter functions (normal distributed noise, median filtering from Feature Squeezing (FeatS), Total Variation Minimisation (TVM), bit reduction @ r = 6 (BitR), FPA; denoted by solid lines), and our baselines (ODIN and IF; denoted by dashed lines) across our experiment settings, for RN, CAN and PN models. Attack strength increases from left to right within each plot. For PGD attacks, it is determined by the parameter while for CW-SGD, it is determined by the parameters κ and η. Higher is better. [11]

This detection-based defense covers adversarial perturbations on support sample poisoning which was earlier described in Family 2 in chapter 4. It doesn't provide a mathematical guarantee like FCert but this being attack-agnostic is an advantage because it can detect attacks it wasn't trained against [11].

5.4 Backdoor Specific Defense:

The previous defense categories all address the vulnerabilities of the query set or support set against evasion and poisoning attacks. None of the defenses presented were specifically designed to counter backdoor attacks in FSL. This section presents the only defense found in the literature that directly targets backdoor vulnerabilities in few-shot

classification, though as will be noted, it addresses a different backdoor mechanism than the inference-time support set injection described in Family 4.

Yang et al. (2024) in their work “Adapting Few-Shot Classification via In-Process Defense”, proposed a defense that works during forward propagation which is why they call it an in-process defense. The idea is trigger-backdoor mismatch. Instead of detecting or removing the trigger, the defense generates task-level representations from the support set that cause the model to produce feature embeddings that no longer align with the backdoor pattern embedded in the backbone.

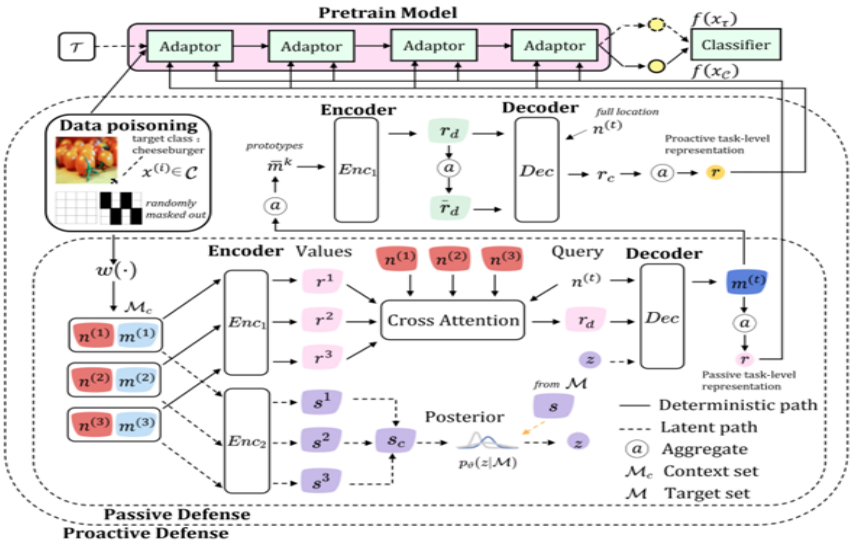

Figure 17: Architecture of the proposed defense framework (proactive and passive) against data poisoning attacks. [42]

To achieve this, they use latent neural processes to learn the task distribution from the support set. The model has two pathways. The first one is deterministic and uses cross-attention to capture local feature relationships from the support set. The second one is latent and samples from the task distribution to capture the bigger picture. Both pathways are combined into a task-level representation that guides how the pre-trained model adapts to the current task. Because this representation comes from sampling the task distribution rather than from the poisoned backbone directly, the backdoor pattern in the backbone no longer gets activated.

They propose two versions of this defense. The passive version just conditions the representations through the neural process without touching the input. The proactive version goes further by also training the neural process with adversarial samples, making it harder to fool across different trigger configurations.

The defense was evaluated on Meta-Dataset against Simple CNAPs and TSA using a visible patch trigger based on BadNets. The passive defense increases in-distribution accuracy by at least 47,4% on backdoored Simple CNAPs compared to the undefended baseline. The proactive defense recovers accuracy from below 25,1% to above 73,4%. Across varying-way 5-shot and 5-way 1-shot settings the overall accuracy against backdoor attacks improves by approximately 30%, and the difference between attacked and unattacked conditions stays within 1%.

While this defense is a backdoor defense, it won't make the model more robust in backdoor attacks, like the ones we described in Chapter 4. The difference is that adversarial attacks in the attack taxonomy kept the backbone clean and injected the backdoor trigger into the support set at inference time. This defense wouldn't work because there is no backdoor pattern in the backbone to mismatch. Similar attacks like the ones in chapter 4 remain undefended in the current taxonomy [42].

Across all four defense categories mentioned in this chapter, evasion attack family and support set poisoning attack family have confirmed defense coverage that is widely spread out. With evasion attacks being addressed by all seven adversarial training defenses and Pautov et al., while support set is addressed by FCert and Tan et al. For the other attack families there wasn't anything related found in terms of robustness. In discussion chapter we will analyze these gaps and talk about the current state of FSL security.

6. EXPERIMENTAL PROOF-OF-CONCEPT:

6.1 Overview:

In this chapter, we present experimental proof of concept demonstrating that the taxonomy's family distinctions are meaningful in practice. We trained a ProtoNet model with a Conv4 backbone and evaluated it under three attacks from different taxonomy families: Fast Gradient Sign Method (Family 1), Adversarial Support Poisoning (Family 2), and FAMF (Family 6). We then trained a second model using the Adversarial Querying defense and re-evaluated all three attacks under both 5-way 1-shot and 5-way 5-shot configurations. The results confirm that different attack families bring much different accuracy which was expected. To validate what was already understood from the literature review itself, we also used ResNet-12 as a backbone and repeated the same process as we did for Conv4. We ran the whole evaluation on two datasets, CIFAR-FS and miniImageNet, to check that the findings hold on more than one benchmark.

6.2 Experimental Setup:

All the experiments were run on Google Colab, on a single NVIDIA A100 40GB GPU, using PyTorch 2.11.0 and Python 3.12.13. No external few-shot library was used, so the ProtoNet models, the episode sampling, the three attacks and the Adversarial Querying defense were all written from scratch in PyTorch. A fixed random seed (42) was set for PyTorch, NumPy and Python's random generator, so the same episodes and the same training can be reproduced if the code runs again. The image normalization is done inside the model and not on the dataset, so the attacks work directly on the clean images in the [0,1] pixel range, which keeps the ε budgets meaningful as real pixel changes.

The experiments were run on two datasets, CIFAR-FS and miniImageNet, both standard few-shot benchmarks. CIFAR-FS is derived from CIFAR-100, split into 64 training and 20 test classes. miniImageNet is an ImageNet-derived benchmark using the standard Ravi & Larochelle split of 64 training, 16 validation and 20 test classes, with 600 images per class. In both datasets all images were resized to 84x84 pixels and the same training and evaluation protocol was applied. We used as target model Prototypical Networks and evaluated two backbone architectures: Conv4 and ResNet-12. Conv4 consists of four convolutional blocks each containing a convolutional layer, batch normalization, ReLU activation and max pooling, with approximately 113,000 parameters. ResNet-12 consists of four residual blocks with skip connections, with approximately 12.4 million parameters. Both backbones were trained from scratch without pre-training to ensure a fair comparison.

The models were trained from scratch without pre-training, initialized with random weights and learned feature vectors by episodic training on the base classes. The training lasted 30,000 episodes under a 5-way 1-shot configuration using Adam optimizer, an adaptive optimization algorithm that adjusts the learning rate individually for each parameter during training, with a starting learning rate of 0.001 decayed by a factor of 0.5 every 10,000 episodes. In each episode, 5 classes are randomly sampled from the 64 training classes, with 1 support image and 15 query images per class. Support images are encoded by the backbone and averaged per class to form prototypes, and queries are classified by nearest prototype using Euclidean distance.

For each backbone, we also trained a second model using the Adversarial Querying defense following the same architecture and training setup. Instead, we replaced clean query images at every episode with adversarial examples generated by PGD with 5 steps, with $\varepsilon = 8/255$ and step size $\alpha = 2/255$ as Goldblum et al. (2020) did in their proposal. This

defense is the reason we evaluated FGSM instead of PGD, to avoid having the same attack evaluated twice and offer more comprehensive results.

Three attacks were implemented, each representing a different taxonomy family as defined in Chapter 4. FGSM (Section 4.1) was implemented with ε=8/255. ASP (Section 4.2) was implemented with ε=8/255, step size α=2/255 and 20 PGD steps, poisoning one support image per class, following Oldewage et al. (2021). FAMF (Section 4.6) was implemented with ε=0.5 and 20 iterations applied directly to query feature vectors after the backbone, following Kim et al. (2026). Results are reported in terms of classification accuracy and Attack Success Rate, calculated as ASR = (1 − ACC_attack / ACC_clean) × 100, where higher ASR indicates a more effective attack. All evaluations were conducted over 600 episodes in 5-way 1-shot and 5-way 5-shot configurations. This allows us to compare how attack success changes when only one support example per class is available (1-shot) versus when five support examples per class are available (5-shot).

6.3 Results:

Classification accuracy (ACC) is calculated as the percentage of query images correctly classified across all episodes. Attack Success Rate (ASR) is calculated as ASR = (1 − ACC_attack / ACC_clean) × 100, where ACC_attack is the accuracy under attack and ACC_clean is the clean accuracy of the same model. A higher ASR indicates a more effective attack. Random guessing on a 5-way classification task corresponds to roughly 20% accuracy (random chance), with the corresponding ASR depending on each model's clean accuracy.

The following Table 7 is the results we got after running all adversarial attacks in the cleanly trained model and after training it with adversarial querying:

	Clean Acc	FGSM Acc	FGSM ASR	ASP Acc	ASP ASR	FAMF Acc	FAMF ASR
Clean ProtoNet 1-shot	56,2%	3,6%	93,6%	19,2%	65,8%	0,0%	100,0%
Clean ProtoNet 5-shot	73,7%	6,0%	91,9%	36,1%	51,0%	0,0%	100,0%
AQ ProtoNet 1-shot	38,2%	27,9%	27,0%	32,1%	16,0%	0,0%	100,0%
AQ ProtoNet 5-shot	50,5%	37,4%	26,0%	48,9%	3,2%	0,0%	100,0%

Table 7: Conv4 backbone, CIFAR-FS.

As shown in the table above, all three attacks did drop the accuracy significantly in the clean ProtoNet model across both 1-shot and 5-shot configurations. With FGSM

achieving an ASR of 93,6% in 1-shot and 91,9% in 5-shot while ASP achieves 65,8% ASR in 1-shot and 51,0% in 5-shot. FAMF achieved 100,0% ASR in all scenarios.

Once we trained the model with adversarial querying the ASR of FGSM dropped to 27,0% in 1-shot and 26,0% in 5-shot, which is a proof of how good results this defense brings in evasion attacks. ASP ASR drops from 65,8% to 16,0% in 1-shot and from 51,0% to 3,2% in 5-shot which indicates that this defense is partial working in support set poisoning attacks too. FAMF however remains at 100,0% even after the adversarial querying confirming that AQ provides no protection against Family 6 embedding space attacks, consistent with the analysis in Section 4.6. [6], [39], [10], [61], [17].

Table 8 shows the results for the ResNet-12 backbone under the same evaluation:

	Clean Acc	FGSM Acc	FGSM ASR	ASP Acc	ASP ASR	FAMF Acc	FAMF ASR
Clean ProtoNet 1-shot	59,7%	15,2%	74,5%	19,3%	67,7%	0,4%	99,3%
Clean ProtoNet 5-shot	75,1%	18,7%	75,1%	33,1%	55,9%	1,5%	98,0%
AQ ProtoNet 1-shot	44,3%	25,2%	43,1%	31,1%	29,8%	0,1%	99,8%
AQ ProtoNet 5-shot	59,3%	37,8%	36,2%	56,2%	5,2%	0,3%	99,5%

Table 8: ResNet-12 backbone, CIFAR-FS.

For the ResNet-12 backbone, the patterns are similar but slightly stronger in favor of AQ. On the clean ProtoNet model, all three attacks significantly reduce accuracy, with FGSM and ASP achieving high ASR in both 1-shot and 5-shot and FAMF again reaching almost 100% ASR. After training with Adversarial Querying, FGSM ASR drops substantially in both 1-shot and 5-shot, and ASP ASR becomes small, especially in the 5-shot case where multiple clean support examples are available. However, FAMF remains near 100% ASR under all conditions, confirming that AQ does not protect against embedding-space (Family 6) attacks.

To prove that these findings hold in more than a single dataset, we repeated the full process on miniImageNet using the same exact attacks, perturbation budgets and training length. Tables 9 and 10 show the results for the Conv4 and ResNet-12 backbones.

	Clean Acc	FGSM Acc	FGSM ASR	ASP Acc	ASP ASR	FAMF Acc	FAMF ASR
Clean ProtoNet 1-shot	49,7%	1,7%	96,6%	20,0%	59,8%	0,0%	100,0%
Clean ProtoNet 5-shot	68,5%	1,3%	98,1%	35,1%	48,8%	0,1%	99,9%
AQ ProtoNet 1-shot	33,2%	21,0%	36,7%	25,4%	23,5%	0,0%	100,0%
AQ ProtoNet 5-shot	54,8%	30,1%	45,1%	50,7%	7,5%	0,0%	100,0%

Table 9: Conv4 backbone, miniImageNet.

	Clean Acc	FGSM Acc	FGSM ASR	ASP Acc	ASP ASR	FAMF Acc	FAMF ASR
Clean ProtoNet 1-shot	51,6%	1,5%	97,1%	18,5%	64,1%	0,0%	100,0%
Clean ProtoNet 5-shot	67,6%	1,4%	97,9%	26,0%	61,5%	3,3%	95,1%
AQ ProtoNet 1-shot	38,0%	20,5%	46,1%	27,7%	27,1%	0,0%	100,0%
AQ ProtoNet 5-shot	64,7%	27,2%	58,0%	59,1%	8,7%	0,0%	100,0%

Table 10: ResNet-12 backbone, miniImageNet.

The same patterns appear on miniImageNet. On both backbones and in both 1-shot and 5-shot, all three attacks drop the accuracy of the clean model a lot. FGSM reaches very high ASR (96–98%), ASP works moderately well, and FAMF brings the accuracy down almost to zero. After we train with Adversarial Querying, the ASR of FGSM drops a lot and the ASR of ASP also goes down, but FAMF stays at or near 100% ASR. This confirms again that AQ does not protect against Family 6 embedding space attacks.

There are two differences from the CIFAR-FS results. First, the clean accuracy is lower on miniImageNet (for example, 49,7% instead of 56,2% for Conv4 1-shot), because miniImageNet is a harder dataset made of natural images. Second, FGSM is stronger on miniImageNet, reaching 96–98% ASR on all clean models, compared to the 74–94% range on CIFAR-FS. This probably happens because the model learns less clear decision

boundaries on the harder data, so a single-step attack like FGSM can cross them more easily. Even with these differences, the main conclusions are the same on both datasets. This shows that how well a defense works depends on the attack family, not on the dataset.

It is also useful to compare these results with the numbers reported in the original papers, to check that the implementation behaves as it should. Goldblum et al. (2020), who proposed Adversarial Querying, report a robust accuracy of around 28% for ProtoNet on 5-shot miniImageNet after AQ training, starting from a much higher clean accuracy. The AQ model we made follows the same pattern, on 5- shot miniImageNet the FGSM accuracy of the AQ model recovers to 27,2% (was 1,4% on the clean model), while the clean accuracy drops 67,6% to 64,7% which is the tradeoff they describe. For FAMF, Kim et al. (2026) report that the attack keeps an ASR close to 100% even against an AQ-trained model, because it attacks the feature vector after the backbone. The results we got match this, with FAMF staying almost at 100% in all tests, on both backbones and both datasets, before and after the AQ (for example 100,0% on Conv4 miniImageNer and 99,5% on ResNet-12 CIFAR-FS, even after the AQ training). The fact that our results match both papers, the partial success of AQ on FGSM and its total failure on FAMF, is a good sign that the implementation is correct and that the differences between the attack families are real and not just on paper.

Another point worth explaining is why the two backbones react so differently to FGSM. On the clean models, FGSM is clearly stronger against Conv4 than against ResNet-12: on CIFAR-FS it reaches a 93,6% ASR on Conv4 but only 74,5% on ResNet-12. A likely reason is the size and design of the two networks. Conv4 is a small network with around 113,000 parameters, while ResNet-12 has around 12.4 million and also uses skip connections. This gives ResNet-12 the capacity to learn stronger and more stable features, so a single-step attack like FGSM, which only takes one gradient step, has a harder time pushing a query across the decision boundary. This does not mean ResNet-12 is safe, since its ASR is still high, but it shows that the same attack does not affect every backbone in the same way, and that the choice of backbone is part of the robustness picture in FSL. This is not something specific to FSL. Zarras et al. (2025) tested FGSM, PGD, DeepFool and C&W on five different architectures in standard deep learning and also found that the same attack does not affect every architecture in the same way [85].

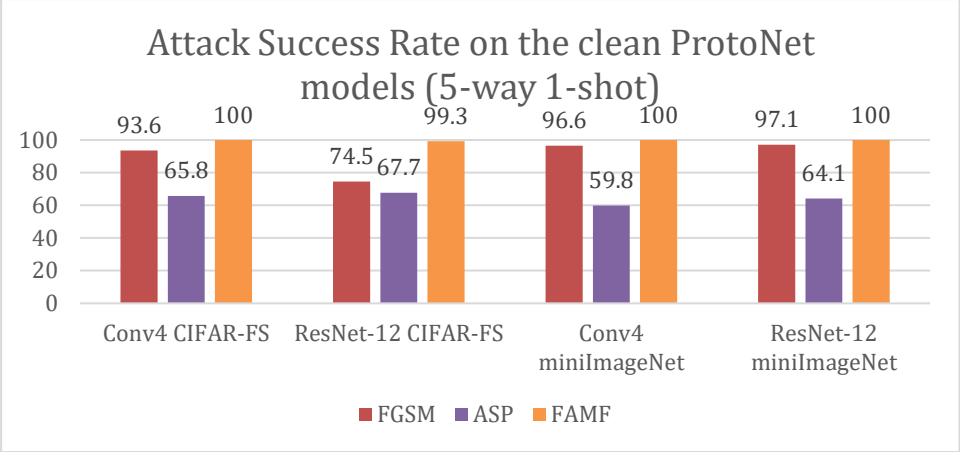

Table 11: Attack Success Rate of the three attacks on the clean ProtoNet models across all four configurations (5-way 1-shot).

6.4 Limitations:

FGSM and ASP use an 8/255 budget in pixel space, while FAMF uses an ε=0,5 budget in feature space. These two are not directly comparable, because a perturbation on the feature vector is not a visible change on the image, and how strong it depends on the scale of the embedding and not on the pixels. This is why the ASR of FAMF should not be read directly against the pixel-space attacks in the same table. FAMF here is more of a qualitative result. As we explained in Section 4.6, FAMF attacks past the backbone stage, it waits until the backbone has produced the feature vector and corrupts it after that, so AQ cannot reach it because AQ only makes the backbone robust against input perturbations. This is why FAMF stays at almost 100% ASR even on the AQ trained model, and it confirms, no matter the exact budget, that a defense built for the input stage does not protect the feature stage.

This chapter evaluated three attacks on clean models and against AQ across two backbone architectures and two datasets, however this is not an exhaustive evaluation. Only FGSM, ASP and FAMF were evaluated while different attacks within the same taxonomy could lead to different results. Both backbones were trained from scratch without pre-training, which limits absolute accuracy compared to state-of-the-art results in the literature where pre-trained backbones typically achieve 70-80% clean accuracy on similar benchmarks. An exhaustive benchmark covering all attack and defense families organized in this taxonomy, as well as evaluations using pre-trained backbones, remain directions for future work.

7. DISCUSSION:

The previous chapters presented a unified taxonomy of adversarial attacks and defenses for Few-Shot Learning models, we organized them across six attack families and four defense categories. We will synthesize what we already explained among previous

chapters from the individual papers and methods to a broader picture of the current state in these models. The goal of this taxonomy is to reveal the gaps and act like a map on where future research should be directed.

Before we go into the gaps, we first put all twelve attacks from Chapter 4 into one table. Table 6 mapped the attacks only across the four taxonomy dimensions, but here we also add the dataset each attack was tested on, the target model and the application domain. This way the table shows not only how each attack is classified, but also where and on what model it was actually tested.

Paper	Family	Attack Surface	Knowledge	Goal	Timing	Dataset(s)	Target model(s)	Application domain
FGSM (Goodfellow et al., 2015)	Evasion	Query Set	White-box	Untargeted Integrity	Inference	miniImageNet, CIFAR-FS	ProtoNet (any gradient-based FSL)	Generic image classification
PGD (Madry et al., 2017)	Evasion	Query Set	White-box	Untargeted Integrity	Inference	miniImageNet	ProtoNet, R2-D2, MetaOptNet	Generic image classification
C&W (Carlini & Wagner, 2017)	Evasion	Query Set	White-box	Targeted Integrity / Untargeted Integrity	Inference	miniImageNet, CIFAR-FS	ProtoNet (metric-based)	Generic image classification
AutoAttack (Croce & Hein, 2020)	Evasion	Query Set	White-box/Black-box	Untargeted Integrity	Inference	miniImageNet, CIFAR-FS	ProtoNet, R2-D2	Generic image classification
ASP (Oldewage et al., 2021)	Support-set poisoning	Support Set	White-box	Untargeted Integrity	Inference	miniImageNet, CIFAR	ProtoNet, MAML, CNAPs	Generic image classification
ASP v2 (Oldewage et al., 2022)	Support-set poisoning	Support Set	White-box	Untargeted Integrity	Inference	miniImageNet, CIFAR	Simple CNAPs	Generic image classification
MetaAttacker (Xu et al., 2021)	Support-set poisoning	Support Set	White-box	Targeted Integrity / Untargeted Integrity	Inference	miniImageNet, Omniglo	MAML, ProtoNet, SNAIL	Generic image classification
Kato et al. (2023)	Backdoor	Support Set	White-box	Targeted Integrity	Inference	Omniglot	MAML	Generic image classification
FLBA / Liu et al. (2024)	Backdoor	Support Set	White-box	Targeted Integrity	Inference	miniImageNet	Baseline++, MAML, ProtoNet	Generic image classification
Sana et al. (2025)	Backdoor	Support Set	Black-box	Targeted Integrity	Inference	MNIST, Omniglot, miniImageNet	ProtoNet	Generic image classification
FSAFW / Hu et al. (2024)	UAPs	Query Set	White-box	Untargeted Integrity	Inference	miniImageNet, TieredImageNet	Baseline, Baseline++, ANIL, R2D2, ProtoNet, DN4	Generic image classification
FAMF / Kim et al. (2026)	Embedding Space	Feature Space	White-box	Untargeted Integrity	Inference	miniImageNet, Omniglot	ProtoNet, FEAT, DN4, ADM	Generic image classification

Table 12:Master overview of all attacks across taxonomy dimensions.

The patterns in this table are the same ones we discuss in the rest of this chapter. The knowledge, timing and goal columns point to the gaps we analyze below, but the dataset and domain columns show one more gap on their own. Every attack is tested only on generic benchmarks like miniImageNet, Omniglot and CIFAR-FS, and the application domain is generic image classification in all twelve cases. Even though we mention in the introduction that medical imaging, face recognition and threat detection are the fields

that make FSL security important, none of the attacks in the literature is actually tested on such a real domain.

One of the most important findings of this taxonomy is that there aren't any black-box and gray-box attacks in the FSL adversarial literature in computer vision besides Square Attack within AutoAttack and the CVAE encoding backdoor by Sana et al., which we categorize as practical black-box following the authors, given the attacker has no access to gradients or parameters but has knowledge of the ProtoNet architecture and control over the support set, leaving strict query only black-box backdoors for FSL underexplored. This is due to the episodic training of FSL in which support set drives adaptation, so the attacker would need access to gradients to create effective perturbations. Another reason is because of the structure of FSL, where the models encounter completely new classes at meta-test time, making it harder for perturbations crafted on a surrogate model to remain effective on the target. Authors including Oldewage et al. and Kim et al. consider black-box scenario attacks as a future direction of the field and still an open one. Gray-box attacks are also absent as much as black-box, which was a surprising finding given that they are the most realistic threat, because an attacker knowing the backbone architecture and not the exact parameters, is much more common than having full access in the model. The existing taxonomy is therefore heavily focused towards a white-box adversary knowledge, which rarely reflects real-world scenarios. Black-box and gray-box attack development for FSL should be prioritized in future work as mentioned among many of the writers of the above attacks.

Another important finding that this taxonomy revealed is that all attacks happen at inference-time. While in theory all attack families could adapt in training-time, the two attack families in which training-time attacks, are more dangerous and best fit the category are Family 3 (Clean-Label Poisoning) and backdoor attacks. Clean-label poisoning corrupts the backbone permanently by injecting crafted images into the training data. The episodic training of FSL is acting like a robust method against that because the crafted images could possibly not be picked, this is theoretically the reason why there isn’t any demonstrated attack in the literature about this attack family. Backdoor attacks in training-time got to bypass the same mechanism as Clean-label poisoning. Even though Yang et al. (2024) proposed a defense against training-time backdoor attacks in FSL, there wasn’t any attack that was evaluated in FSL model to prove this is transferable from standard deep learning. On the other hand, inference-time backdoor attacks were confirmed by Kato et al., Liu et al. and Sana et al. without any defense addressing them which creates an unusual asymmetry on the current attacks and defenses.

Every attack and defense discussed in this thesis was evaluated in FSL specific scenarios by the authors that proposed the related attack and defense. What also becomes clear is the asymmetry in defenses against attacks. Most of the defenses found in the literature were proposed against evasion attacks that transferred from DL, while attacks that

require the most adaptation for FSL, like inference-time backdoor attacks or UAPs and embedding space attacks, remain completely defenseless while support set poisoning attacks have some coverage through Tan et al. and FCert but they don't provide a complete solution into making the model robust against those attacks.

Table 13 summarizes this coverage across the six attack families and makes the asymmetry clear:

Attack Family	Defense coverage	Defenses that address it
1. Evasion (query set)	Covered	The seven adversarial-training defenses (AQ, R-MAML, DFSL, RESISTANCE...) + Pautov et al.
2. Support-set poisoning	Partial	Tan et al. (detection/filtering), FCert (certified)
3. Clean-label poisoning	None	- (no demonstrated FSL attack either)
4. Backdoor (inference-time)	None	- (Yang et al. defends *training-time* backbone backdoors, not the inference-time support-set backdoors that actually exist)
5.Universal perturbations (UAP)	None	-
6. Embedding-space	None	-

Table 13: Defense coverage across the six attack families.

Adversarial querying, which is the most widely used defense in this taxonomy and most widely used in FSL literature, has a significant risk of dropping clean accuracy. This was confirmed by the experimental proof of concept we did in chapter 6, in which the clean accuracy dropped from 56,2% to 38,2% in 1-shot setting on CIFAR-FS, and a similar drop on miniImageNet (49,7% to 33,2%), which can cause critical problems in real life scenarios such as medical imaging for rare disease detection. Other defenses like R-MAML, DFSL and RESISTANCE tried to fix this but the tradeoff between clean and robust accuracy remains an open problem in FSL. The Fβ score introduced by Li et al. (2019), which combines both clean and robust accuracy into a metric with tunable parameter β, is an interesting generalization for future work to compare across FSL defense proposals.

In general, the taxonomy reveals that FSL adversarial security threats are still not well researched, which is expected given it's relatively new. Evasion attacks seem to be the most defended attack family, which is also expected due to them being transferred from standard DL, but there are still significant gaps for threat models in real life scenarios.

8. CONCLUSION:

This thesis goal was to provide a complete understanding of the current state of FSL adversarial security by mapping the existing adversarial attacks and defenses in computer vision and proposing a unified taxonomy for both. We explained the FSL pipeline along with the models used to classify the queries (e.g ProtoNet, MAML), we split the attacks into six attack families and the defenses into four. We also made a proof-of-concept to evaluate few adversarial attacks on CIFAR-FS and miniImageNet using ProtoNet under a Conv4 and a ResNet-12 backbone.

The taxonomy revealed three main gaps, however it's not exhaustive. Majority of the attacks are white-box attacks, while black-box and gray-box are less common even though they are the most realistic threat in a real life scenario. All the attacks happen at inference-time, since the episodic sampling of FSL blocks training-time clean-label or backdoor attacks. There is also an asymmetry between attacks and defenses, as noticed most are evasion attacks transferred from standard deep learning, while embedding space attacks, universal perturbations and inference-time backdoor attacks have no defense at all.

The experiment we made, supported the finding that defenses made for one attack family don't transfer to another. Adversarial Querying defended well against evasion attack FGSM, partly on ASP and had no impact on FAMF which maintained a close to 100% attack success rate while it also dropped clean accuracy from 56,2% to 38,2% in 1-shot on CIFAR-FS, with a similar drop on miniImageNet (49,7% to 33,2%).

The same gaps is where future work in the field should focus on, developing black-box and gray-box attacks for FSL, testing whether training-time clean-label attacks and backdoor attacks can survive the episodic sampling, defending against families that currently have no defense and running an exhaustive benchmark across the whole taxonomy with pre-trained backbones, possibly by using the Fβ score as a common metric.

Lastly, this thesis is meant to act as a map for FSL adversarial security and where it should go.

REFERENCES:

1) Szegedy, Christian, et al. “Intriguing Properties of Neural Networks.” ArXiv.org, 19 Feb. 2013, arxiv.org/abs/1312.6199.

2) Akhtar, Naveed, and Ajmal Mian. “Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey.” ArXiv (Cornell University), 2 Jan. 2018, https://doi.org/10.48550/arxiv.1801.00553.

3) Yuan, Xiaoyong, et al. “Adversarial Examples: Attacks and Defenses for Deep Learning.” ArXiv (Cornell University), 19 Dec. 2017, https://doi.org/10.48550/arxiv.1712.07107.

4) Chakraborty, Anirban, et al. “A Survey on Adversarial Attacks and Defences.” CAAI Transactions on Intelligence Technology, vol. 6, no. 1, Mar. 2018, pp. 25–45, https://doi.org/10.1049/cit2.12028.

5) Gharoun, Hassan, et al. “Meta-Learning Approaches for Few-Shot Learning: A Survey of Recent Advances.” ArXiv.org, 2023, arxiv.org/abs/2303.07502.

6) Goldblum, Micah, et al. “Adversarially Robust Few-Shot Learning: A Meta-Learning Approach.” ArXiv.org, 2020, arxiv.org/abs/1910.00982. Accessed 4 June 2026.

7) Song, Yisheng, et al. “A Comprehensive Survey of Few-Shot Learning: Evolution, Applications, Challenges, and Opportunities.” ArXiv (Cornell University), 13 May 2022, https://doi.org/10.48550/arxiv.2205.06743.

8) Tsoumplekas, Georgios, et al. “Toward Green and Human-like Artificial Intelligence: A Complete Survey on Contemporary Few-Shot Learning Approaches.” Arxiv.org, 2024, arxiv.org/html/2402.03017v2.

9) Song, Junhao, et al. Meta-Learning and Few-Shot Learning: A Comprehensive Survey of Algorithms, Theory, and Applications. 8 Jan. 2026, https://doi.org/10.36227/techrxiv.176784520.02393968/v1. Accessed 11 Apr. 2026.

10) Elre Talea Oldewage, et al. “Attacking Few-Shot Classifiers with Adversarial Support Poisoning.” Openreview.net, 2021, openreview.net/forum?id=SBjXIq5Zk8. Accessed 4 June 2026.

11) Xiang Yi Marcus Tan et al. “Towards a Conceptually Simple Defensive Approach for Few-Shot Classifiers against Adversarial Support Samples.” ArXiv.org, 24 Oct. 2021, arXiv.org/abs/2110.12357. Accessed 4 June 2026.

12) Li, Wenbin, et al. “Defensive Few-Shot Learning.” ArXiv.org, 16 Nov. 2019, arXiv.org/abs/1911.06968. Accessed 4 June 2026.

13) Dong, Junhao, et al. “Adversarially Robust Few-Shot Learning via Parameter Co-Distillation of Similarity and Class Concept Learners.” 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16 June 2024, pp. 28535–28544, https://doi.org/10.1109/cvpr52733.2024.02696. Accessed 4 June 2026.

14) Liu, Xinwei, et al. “Does Few-Shot Learning Suffer from Backdoor Attacks?” ArXiv.org, 31 Dec. 2024, arXiv.org/abs/2401.01377.

15) Duan, Ruixue, et al. “A Survey of Few-Shot Learning: An Effective Method for Intrusion Detection.” Security and Communication Networks, vol. 2021, no. 1, 31 Oct. 2021, pp. 1–10, https://doi.org/10.1155/2021/4259629.

16) Yang, Zhanyuan, et al. “Few-Shot Classification with Contrastive Learning.” ArXiv.org, 17 Sept. 2022, arXiv.org/abs/2209.08224.

17) Snell, Jake, et al. “Prototypical Networks for Few-Shot Learning.” ArXiv.org, 15 Mar. 2017, arXiv.org/abs/1703.05175.

18) Vinyals, Oriol, et al. “Matching Networks for One Shot Learning.” ArXiv:1606.04080 [Cs, Stat], 29 Dec. 2016, arxiv.org/abs/1606.04080, https://doi.org/10.48550/arXiv.1606.04080.

19) Finn, Chelsea, et al. “Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks.” ArXiv.org, 18 July 2017, arXiv.org/abs/1703.03400v3.

20) Santoro, Adam, et al. “One-Shot Learning with Memory-Augmented Neural Networks.” ArXiv.org, 19 May 2016, arXiv.org/abs/1605.06065.

21) Mishra, Nikhil, et al. “A Simple Neural Attentive Meta-Learner.” ArXiv.org, 25 Feb. 2018, arXiv.org/abs/1707.03141.

22) Koch, Gregory, et al. Siamese Neural Networks for One-Shot Image Recognition. 2015.

23) Figure 2: Nag, Rinki. “A Comprehensive Guide to Siamese Neural Networks.” Medium, 19 Nov. 2022, medium.com/@rinkinag24/a-comprehensive-guide-to-siamese-neural-networks-3358658c0513.

24) Figure 3: Snell, Jake, et al. “Prototypical Networks for Few-Shot Learning.” ArXiv.org, 15 Mar. 2017, arXiv.org/abs/1703.05175.

25) Figure 4: Vinyals, Oriol, et al. “Matching Networks for One Shot Learning.” ArXiv:1606.04080 [Cs, Stat], 29 Dec. 2016, arxiv.org/abs/1606.04080, https://doi.org/10.48550/arXiv.1606.04080.

26) Sung, Flood, et al. “Learning to Compare: Relation Network for Few-Shot Learning.” ArXiv.org, 16 Nov. 2018, arXiv.org/abs/1711.06025.

27) Figure 5: Sung, Flood, et al. “Learning to Compare: Relation Network for Few-Shot Learning.” ArXiv.org, 16 Nov. 2018, arXiv.org/abs/1711.06025.

28) Garcia, Victor, and Joan Bruna. Few-Shot Learning with Graph Neural Networks. 20 Feb. 2018, https://doi.org/10.48550/arxiv.1711.04043.

29) Figure 6: Garcia, Victor, and Joan Bruna. Few-Shot Learning with Graph Neural Networks. 20 Feb. 2018, https://doi.org/10.48550/arxiv.1711.04043.

30) Table 1: Comparison based on query within metric-based models

31) Ravi, Sachin, and Hugo Larochelle. OPTIMIZATION as a MODEL for FEW-SHOT LEARNING. 2017.

32) Rusu, Andrei A, et al. “Meta-Learning with Latent Embedding Optimization.” ArXiv.org, 26 Mar. 2019, arXiv.org/abs/1807.05960. Accessed 4 June 2026.

33) Table 2: comparison of optimization-based methods

34) Karunaratne, Geethan, et al. “Robust High-Dimensional Memory-Augmented Neural Networks.” Nature Communications, vol. 12, no. 1, 29 Apr. 2021, https://doi.org/10.1038/s41467-021-22364-0.

35) Munkhdalai, Tsendsuren, and Hong Yu. “Meta Networks.” ArXiv.org, 8 June 2017, arXiv.org/abs/1703.00837. Accessed 4 June 2026.

36) Wang, Yu-Xiong, et al. “Low-Shot Learning from Imaginary Data.” ArXiv.org, 3 Apr. 2018, arXiv.org/abs/1801.05401. Accessed 4 June 2026.

37) Sergey Bartunov, and Dmitry Vetrov. “Few-Shot Generative Modelling with Generative Matching Networks.” PMLR, vol. 84, Mar. 2018, pp. 670–678, proceedings.mlr.press/v84/bartunov18a.html. Accessed 4 June 2026.

38) Wang, Yaqing, et al. “Generalizing from a Few Examples.” ACM Computing Surveys, vol. 53, no. 3, 5 July 2020, pp. 1–34, https://doi.org/10.1145/3386252.

39) Goodfellow, Ian, et al. “Explaining and Harnessing Adversarial Examples.” ArXiv, 20 Mar. 2015, https://doi.org/10.48550/arxiv.1412.6572.

40) Madry, Aleksander, et al. Towards Deep Learning Models Resistant to Adversarial Attacks. June 2017, https://doi.org/10.48550/arxiv.1706.06083.

41) Figure 7: Bountakas et al. (2023) Defense strategies for Adversarial Machine Learning: A survey

42) Yang, Xi, et al. “Adapting Few-Shot Classification via In-Process Defense.” IEEE Transactions on Image Processing, vol. 33, 17 Sept. 2024, pp. 5232–5245, https://doi.org/10.1109/tip.2024.3458858. Accessed 4 June 2026.

43) Brendel, Wieland, et al. “Decision-Based Adversarial Attacks: Reliable Attacks against Black-Box Machine Learning Models.” ArXiv (Cornell University), 16 Feb. 2018, https://doi.org/10.48550/arxiv.1712.04248.

44) Papernot, Nicolas, et al. “Practical Black-Box Attacks against Machine Learning.” Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security - ASIA CCS ’17, Apr. 2017, pp. 506–519, arxiv.org/pdf/1602.02697.pdf, https://doi.org/10.1145/3052973.3053009.

45) Wang, Wenqiang, et al. “Multi-Task Adversarial Attacks against Black-Box Model with Few-Shot Queries.” Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), July 2025, pp. 13994–14014,

aclanthology.org/2025.acl-long.684/, https://doi.org/10.18653/v1/2025.acl-long.684. Accessed 4 June 2026.

46) Vassilev, A., Oprea, A., Fordyce, A., Anderson, H., Davies, X. and Hamin, M. (2025), Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations, NIST Trustworthy and Responsible AI, National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.AI.100-2e2025, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=959735 (Accessed June 4, 2026)

47) Figure 8: Vassilev, A. , Oprea, A. , Fordyce, A. , Anderson, H. , Davies, X. and Hamin, M. (2025), Adversarial Machine Learning: A Taxonomy and Terminology of Attacks and Mitigations, NIST Trustworthy and Responsible AI, National Institute of Standards and Technology, Gaithersburg, MD, [online], https://doi.org/10.6028/NIST.AI.100-2e2025, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=959735 (Accessed June 4, 2026)

48) Shafahi, Ali, et al. “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks.” ArXiv.org, 10 Nov. 2018, arXiv.org/abs/1804.00792.

49) Chen, Xinyun, et al. “Targeted Backdoor Attacks on Deep Learning Systems Using Data Poisoning.” ArXiv (Cornell University), 15 Dec. 2017, https://doi.org/10.48550/arxiv.1712.05526.

50) Carlini, Nicholas, and David Wagner. Towards Evaluating the Robustness of Neural Networks. 22 Mar. 2017.

51) Croce, Francesco, and Matthias Hein. “Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks.” ArXiv.org, 4 Aug. 2020, arXiv.org/abs/2003.01690.

52) Oldewage, Elre T., et al. “Adversarial Attacks Are a Surprisingly Strong Baseline for Poisoning Few-Shot Meta-Learners.” ArXiv, Nov. 2022, arxiv.org/pdf/2211.12990. Accessed 4 June 2026.

53) Xu, Han, et al. “Yet Meta Learning Can Adapt Fast, It Can Also Break Easily.” ArXiv.org, 2 Sept. 2021, arxiv.org/abs/2009.01672. Accessed 4 June 2026.

54) Huang, W. Ronny, et al. “MetaPoison: Practical General-Purpose Clean-Label Data Poisoning.” ArXiv.org, 20 Feb. 2021, arXiv.org/abs/2004.00225v2.

55) Geiping, Jonas, et al. “Witches’ Brew: Industrial Scale Data Poisoning via Gradient Matching.” ArXiv:2009.02276 [Cs], 10 May 2021, arxiv.org/abs/2009.02276, https://doi.org/10.48550/arXiv.2009.02276.

56) Gu, Tianyu, et al. “BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain.” ArXiv (Cornell University), 11 Mar. 2019, https://doi.org/10.48550/arxiv.1708.06733.

57) Kato, Ganma, et al. “Backdoor Poisoning Attacks against Few-Shot Classifiers Based on Meta-Learning.” Nonlinear Theory and Its Applications, IEICE, vol. 14, no. 2, 2023, pp. 491–499, https://doi.org/10.1587/nolta.14.491. Accessed 4 June 2026.

58) Sana, et al. “Leveraging CVAE Encoding for Backdoor Attacks in Few-Shot Learning with Prototypical Networks.” IEEE Transactions on Dependable and Secure Computing, vol. 23, no. 1, Sept. 2025, pp. 432–448, https://doi.org/10.1109/tdsc.2025.3606799. Accessed 4 June 2026.

59) Moosavi-Dezfooli, Seyed-Mohsen, et al. “Universal Adversarial Perturbations.” ArXiv (Cornell University), 9 Mar. 2017, https://doi.org/10.48550/arxiv.1610.08401.

60) Hu, X., Han, L. and Wei, W., 2024. Generate Universal Adversarial Perturbations for Few-Shot Learning. In: Advances in Neural Information Processing Systems (NeurIPS 2024). Vancouver, Canada, 10-15 December 2024.

61) Kim, G.-N., Lee, H.-J., Jeong, I.-W., Shin, J.-M. and Choi, S.-H., 2026. FAMF: Robust Feature-Level Adversarial Attack on Metric-Based Few-Shot Learning Models. IEEE Access, 14, pp.1-15.

62) Yin, Chengxiang, et al. “Adversarial Meta-Learning.” ArXiv.org, June 2018, arXiv.org/abs/1806.03316. Accessed 4 June 2026.

63) Li, Wenbin, et al. “Defensive Few-Shot Learning.” ArXiv.org, 16 Nov. 2019, arXiv.org/abs/1911.06968. Accessed 4 June 2026.

64) Wang, Ren, et al. “On Fast Adversarial Robustness Adaptation in Model-Agnostic Meta-Learning.” ArXiv.org, 20 Feb. 2021, arXiv.org/abs/2102.10454. Accessed 4 June 2026.

65) Dong, J., Wang, Y., Lai, J. and Xie, X., 2022. Improving adversarially robust few-shot image classification with generalizable representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9025-9034.

66) Nayak, Gaurav Kumar, et al. “Robust Few-Shot Learning without Using Any Adversarial Samples.” ArXiv.org, 3 Nov. 2022, arXiv.org/abs/2211.01598. Accessed 4 June 2026.

67) Cohen, Jeremy M., et al. “Certified Adversarial Robustness via Randomized Smoothing.” ArXiv:1902.02918 [Cs, Stat], 15 June 2019, arxiv.org/abs/1902.02918, https://doi.org/10.48550/arXiv.1902.02918.

68) Pautov, Mikhail, et al. “Smoothed Embeddings for Certified Few-Shot Learning.” ArXiv.org, 16 Sept. 2022, arXiv.org/abs/2202.01186. Accessed 4 June 2026.

69) Wang, Yanting, et al. “FCert: Certifiably Robust Few-Shot Classification in the Era of Foundation Models.” ArXiv.org, 12 Apr. 2024, arXiv.org/abs/2404.08631. Accessed 4 June 2026.

70) Lai, Yuni, et al. “Provably Robust Adaptation for Language-Empowered Foundation Models.” ArXiv.org, 9 Oct. 2025, arXiv.org/abs/2510.08659. Accessed 4 June 2026.

71) Figure 1: BotPenguin (2026). Few-shot Learning: Applications, Challenges & Techniques. [online] Botpenguin.com. Available at: https://botpenguin.com/glossary/few-shot-learning [Accessed 4 Jun. 2026].

72) Figure 9: Sana, et al. “Leveraging CVAE Encoding for Backdoor Attacks in Few-Shot Learning with Prototypical Networks.” IEEE Transactions on Dependable and Secure Computing, vol. 23, no. 1, Sept. 2025, pp. 432–448, https://doi.org/10.1109/tdsc.2025.3606799. Accessed 4 June 2026.

73) Figure 10: Bountakas et al. (2023) Defense strategies for Adversarial Machine Learning: A survey

74) Figures 11 & 12: Pautov, Mikhail, et al. “Smoothed Embeddings for Certified Few-Shot Learning.” ArXiv.org, 16 Sept. 2022, arXiv.org/abs/2202.01186. Accessed 4 June 2026.

75) Figure 13: Wang, Yanting, et al. “FCert: Certifiably Robust Few-Shot Classification in the Era of Foundation Models.” ArXiv.org, 12 Apr. 2024, arXiv.org/abs/2404.08631. Accessed 4 June 2026.

76) Figures 14 & 15: Wang, Yanting, et al. “FCert: Certifiably Robust Few-Shot Classification in the Era of Foundation Models.” ArXiv.org, 12 Apr. 2024, arXiv.org/abs/2404.08631. Accessed 4 June 2026.

77) Figure 16: Xiang, Yi, et al. “Towards a Conceptually Simple Defensive Approach for Few-Shot Classifiers against Adversarial Support Samples.” ArXiv.org, 24 Oct. 2021, arXiv.org/abs/2110.12357. Accessed 4 June 2026.

78) Figure 17: Yang, Xi, et al. “Adapting Few-Shot Classification via In-Process Defense.” IEEE Transactions on Image Processing, vol. 33, 17 Sept. 2024, pp. 5232–5245, https://doi.org/10.1109/tip.2024.3458858. Accessed 4 June 2026.

79) Foukanelis, C.G. (2025). Few shot learning : an overview of methods, applications and benchmarks. *Unipi.gr*. [online] doi:https://dione.lib.unipi.gr/xmlui/handle/unipi/18719.

80) IBM (n.d.). *Few Shot Learning*. [online] Ibm.com. Available at: https://www.ibm.com/think/topics/few-shot-learning [Accessed 7 Jun. 2026].

81) Akhtar, Naveed, et al. "Threat of Adversarial Attacks on Deep Learning in Computer Vision: Survey II." ArXiv (Cornell University), 1 Aug. 2021, https://doi.org/10.48550/arXiv.2108.00401.

82) Stylianou, I., Bountakas, P., Zarras, A., Farao, A., Bolgouras, V. and Xenakis, C. (2026). SoK: A taxonomy of LLM threats. Computer Science Review, 62, p.101013. doi:10.1016/j.cosrev.2026.101013.

83) Giapantzis, K., Bountakas, P., Zarras, A., Farao, A., Bolgouras, V. and Xenakis, C. (2026). Adaptive DeSeTra: Adaptive deformable-span self-attention transformer for LLM security. Information Sciences, 754, p.123670. doi:10.1016/j.ins.2026.123670.

84) Ziras, G., Farao, A., Zarras, A. and Xenakis, C. (2025). From vulnerability to resilience: Adversarial training and real-time detection for AI security. Array, [online] 28, p.100546. doi:10.1016/j.array.2025.100546.

85) Zarras, A., Kollarou, A., Farao, A., Bountakas, P. and Xenakis, C. (2025). Testing the limits: Exploring adversarial techniques in AI models. PeerJ Computer Science, 11, p.e3330. doi:10.7717/peerj-cs.3330.