CHAPTER 17

Adversarial robustness in meta-learning and contrastive learning

A popular trend in machine learning is an extension of the methodology of single-task learning to fast adaptation or general representation learning, such as meta-learning, which quickly adapts the current metamodel to solve a new task in few-shot scenarios, and contrastive learning, which obtains general representations using self-supervision for efficient finetuning on specific tasks. In this chapter, we take a close look at these two machine learning paradigms and study their adversarial robustness in the downstream tasks, based on the results of (Wang et al., 2021b) and (Fan et al., 2021).

17.1 Fast adversarial robustness adaptation in model-agnostic meta-learning

Meta-learning, which can offer fast generalization adaptation to unseen tasks (Thrun and Pratt, 2012; Novak and Gowin, 1984), has widely been studied from model- and metric-based methods (Santoro et al., 2016; Munkhdalai and Yu, 2017; Koch et al., 2015; Snell et al., 2017) to optimization-based methods (Ravi and Larochelle, 2016; Finn et al., 2017; Nichol et al., 2018). In particular, model-agnostic meta-learning (MAML) (Finn et al., 2017) is one of the most intriguing bilevel optimization-based meta-learning methods designed for fast-adapted few-shot learning; that is, the learnt metamodel can rapidly be generalized to unforeseen tasks with only a small amount of data. It has successfully been applied to use cases such as object detection (Wang et al., 2020a), medical image analysis (Maicas et al., 2018), and language modeling (Huang et al., 2018).

Tackling the problem of adversarial robustness in MAML is more challenging than that of the standard model training, since MAML contains a bileveled learning procedure in which the meta-update step (outer loop) optimizes a task-agnostic initialization of model parameters, whereas the fine-tuning step (inner loop) learns a task-specific model instantiation updated from the common initialization. Thus it remains elusive *when*

https://doi.org/10.1016/B978-0-12-824020-5.00028-4

(namely, at which learning stage) and *how* robust regularization should be promoted to strike a graceful balance between generalization/robustness and computation efficiency. Note that neither the standard MAML (Finn et al., 2017) nor the standard robust training (Madry et al., 2018; Zhang et al., 2019b) is as easy as normal training. Besides the algorithmic design in robust MAML, it is also important to draw in-depth explanation and analysis on *why* adversarial robustness can efficiently be gained in MAML. Wang et al. (2021b) study the problem of adversarial robustness in MAML (Yin et al., 2018; Goldblum et al., 2019) and make affirmative answers to the above questions on *when*, *how*, and *why*.

MAML framework. MAML attempts to learn an initialization of model parameters (namely, a metamodel) so that a new few-shot task can quickly and easily be tackled by fine-tuning this metamodel over a small amount of labeled data. The characteristic signature of MAML is its *bilevel* learning procedure, where the fine-tuning stage forms a task-specific *inner loop*, whereas the metamodel is updated at the *outer loop* by minimizing the validation error of fine-tuned models over cumulative tasks. Formally, consider N few-shot learning tasks $\{\mathcal{T}_i\}_{i=1}^N$, each of which has a fine-tuning data set $\mathcal{D}_i$ and a validation set $\mathcal{D}'_i$, where $\mathcal{D}_i$ is used in the *fine-tuning* stage, and $\mathcal{D}'_i$ is used in the *meta-update* stage. Here the superscript ($'$) is preserved to indicate operations/parameters at the meta-update stage. MAML is then formulated as the following bilevel optimization problem (Finn et al., 2017):

$$\begin{array}{ll} \text{minimize}_{\mathbf{w}} & \frac{1}{N}\sum_{i=1}^N \ell'_i(\mathbf{w}'_i; \mathcal{D}'_i) \\ \text{subject to} & \mathbf{w}'_i = \underset{\mathbf{w}_i}{\operatorname{argmin}}\, \ell_i(\mathbf{w}_i; \mathcal{D}_i, \mathbf{w}) \ \forall i \in [N], \end{array} \tag{17.1}$$

where $\mathbf{w}$ denotes the metamodel to be designed, $\mathbf{w}'_i$ is the $\mathcal{T}_i$-specific fine-tuned model, $\ell'_i(\mathbf{w}'_i; \mathcal{D}'_i)$ represents the validation error using the fine-tuned model, $\ell_i(\mathbf{w}_i; \mathcal{D}_i, \mathbf{w})$ denotes the training error when fine-tuning the task-specific model parameters $\mathbf{w}_i$ using the task-agnostic initialization $\mathbf{w}$, and for ease of notation, $[K]$ represents the integer set $\{1, 2, \ldots, K\}$. In (17.1) the objective function and the constraint correspond to the meta-update e and fine-tuning stages, respectively. The bilevel optimization problem is challenging because each constraint calls an inner optimization oracle, which is typically instantiated into a K-step gradient descent (GD) based solver:

$$\mathbf{w}_i^{(k)} = \mathbf{w}_i^{(k-1)} - \alpha \nabla_{\mathbf{w}_i} \ell_i(\mathbf{w}_i^{(k-1)}; \mathcal{D}_i, \mathbf{w}), \ \ k \in [K], \ \text{with } \mathbf{w}_i^{(0)} = \mathbf{w}.$$

We note that even with the above-simplified fine-tuning step, updating the metamodel $\mathbf{w}$ still requires the second-order derivatives of the objective function of (17.1) with respect to $\mathbf{w}$.

Recall that the min-max optimization-based adversarial training (AT) is known as one of the most powerful defense methods to obtain a robust model against adversarial attacks (see Chapter 12). We summarize AT and its variants through the following robustness-regularized optimization problem:

$$\underset{\mathbf{w}}{\text{minimize}} \quad \lambda \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}\left[\ell(\mathbf{w};\mathbf{x},y)\right] + \underbrace{\mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}}[\underset{\|\boldsymbol{\delta}\|_\infty\le\epsilon}{\text{maximize}}\, g(\mathbf{w};\mathbf{x}+\boldsymbol{\delta},y)]}_{\mathcal{R}(\mathbf{w};\mathcal{D})}, \tag{17.2}$$

where $\ell(\mathbf{w};\mathbf{x},y)$ denotes the prediction loss evaluated at the point $\mathbf{x}$ with label y, $\lambda \ge 0$ is a regularization parameter, $\boldsymbol{\delta}$ denotes the input perturbation variable within the ℓ_∞-norm ball of radius ϵ, g represents the robust loss evaluated at the model $\mathbf{w}$ at the perturbed example $\mathbf{x}+\boldsymbol{\delta}$ given the true label y, and for ease of notation, $\mathcal{R}(\mathbf{w};\mathcal{D})$ denotes the robust regularization function for model $\mathbf{w}$ under the data set $\mathcal{D}$. In the rest of this section, we consider two specifications of $\mathcal{R}$: (a) *AT regularization* (Madry et al., 2018), where we set $g = \ell$ and $\lambda = 0$; and (b) *TRADES regularization* (Zhang et al., 2019b), where we define g as the cross-entropy between the distribution of prediction probabilities at the perturbed example $(\mathbf{x}+\boldsymbol{\delta})$ and that at the original sample $\mathbf{x}$.

Robustness-promoting MAML. Integrating MAML with AT is a natural solution to enhance adversarial robustness of a metamodel in few-shot learning. However, this seemingly simple scheme is in fact far from trivial, and there exist three critical roadblocks as elaborated below.

First, it remains elusive at which stage (fine-tuning or meta-update) robustness can most effectively be gained for MAML. Based on (17.1) and (17.2), we can cast this problem as a unified optimization problem that augments the MAML loss with robust regularization under two degrees of freedom characterized by two hyper-parameters $\gamma_{\text{out}} \ge 0$ and $\gamma_{\text{in}} \ge 0$:

$$\begin{array}{ll} \underset{\mathbf{w}}{\text{minimize}} & \frac{1}{N}\sum_{i=1}^{N}[\ell_i'(\mathbf{w}_i';\mathcal{D}_i') + \gamma_{\text{out}}\mathcal{R}_i(\mathbf{w}_i';\mathcal{D}_i')] \\ \text{subject to} & \mathbf{w}_i' = \underset{\mathbf{w}_i}{\text{argmin}}[\ell_i(\mathbf{w}_i;\mathcal{D}_i,\mathbf{w}) + \gamma_{\text{in}}\mathcal{R}_i(\mathbf{w}_i;\mathcal{D}_i)]\ \forall i \in [N]. \end{array} \tag{17.3}$$

Here $\mathcal{R}_i$ denotes the task-specific robustness regularizer, and the choice of $(\gamma_{\text{in}}, \gamma_{\text{out}})$ determines the specific scenario of robustness-promoting

MAML. Clearly, the direct application is to set $\gamma_{\text{in}} > 0$ and $\gamma_{\text{out}} > 0$, that is, both fine-tuning and meta-update steps would be carried out using robust training, which calls additional loops to generate adversarial examples. Thus this would make computation most intensive. Spurred by that, we ask: *Is it possible to achieve a robust metamodel by incorporating robust regularization into only either meta-update or fine-tuning step (corresponding to $\gamma_{\text{in}} = 0$ or $\gamma_{\text{out}} = 0$)?*

Second, both MAML in (17.1) and AT in (17.2) are challenging bilevel optimization problems, which need to call inner optimization routines for fine-tuning and attack generation, respectively. Thus we ask whether or not the computationally light alternatives of inner solvers, e.g., partial fine-tuning (Raghu et al., 2019) and fast attack generation (Wong et al., 2020a), can promise adversarial robustness in few-shot learning.

Third, it has been shown that adversarial robustness can benefit from semisupervised learning by leveraging (unlabeled) data augmentation (Carmon et al., 2019; Stanforth et al., 2019). Spurred by that, we further ask: *Is it possible to generalize robustness-promoting MAML to the setup of semisupervised learning for improved accuracy-robustness tradeoff?*

When and how to incorporate robust regularization in MAML?

Based on (17.3), we focus on two robustness-promoting meta-training protocols proposed by Wang et al. (2021b): (a) R-MAML$_{\text{both}}$, where robustness regularization is applied to *both* fine-tuning and meta-update steps with $\gamma_{\text{in}}, \gamma_{\text{out}} > 0$, and (b) R-MAML$_{\text{out}}$, where robust regularization applied to *meta-update only*, i.e., $\gamma_{\text{in}} = 0$ and $\gamma_{\text{out}} > 0$. Compared to R-MAML$_{\text{both}}$, R-MAML$_{\text{out}}$ is more user-friendly since it allows the use of *standard* fine-tuning over the learnt robust metamodel when tackling unseen few-shot test tasks (known as meta-testing). In what follows, we will show that even if R-MAML$_{\text{out}}$ does not use robust regularization in fine-tuning, it is sufficient to warrant the transferability of robustness of the metamodel to downstream fine-tuning tasks.

All you need is robust meta-update during meta-training. To study this claim, we solve problem (17.3) using R-MAML$_{\text{both}}$ and R-MAML$_{\text{out}}$ in the 5-way 1-shot learning setup, where one data sample at each of five randomly selected MiniImagenet classes (Ravi and Larochelle, 2016) constructs a learning task. Throughout this section, we specify $\mathcal{R}_i$ in (17.3) as the AT regularization, which calls a 10-step projected gradient descent (PGD) attack generation method with $\epsilon = 2/255$ in its inner maximization subroutine given by (17.2).

We find that the metamodel acquired by R-MAML$_{\text{out}}$ yields *nearly the same robust accuracy* (RA) as R-MAML$_{\text{both}}$ against various PGD attacks generated at the testing phase using different perturbation sizes $\epsilon = \{0, 2, \ldots, 10\}/255$ as shown in Fig. 17.1. Unless specified otherwise, we evaluate the performance of the meta-learning schemes over 2400 random unseen 5-way 1-shot test tasks. We also note that RA under $\epsilon = 0$ becomes the standard accuracy (SA) evaluated using benign (unperturbed) test examples. It is clear from Fig. 17.1 that both R-MAML$_{\text{out}}$ and R-MAML$_{\text{both}}$ can yield significantly better RA than MAML with slightly worse SA. It is also expected that RA decreases as the attack power ϵ increases.

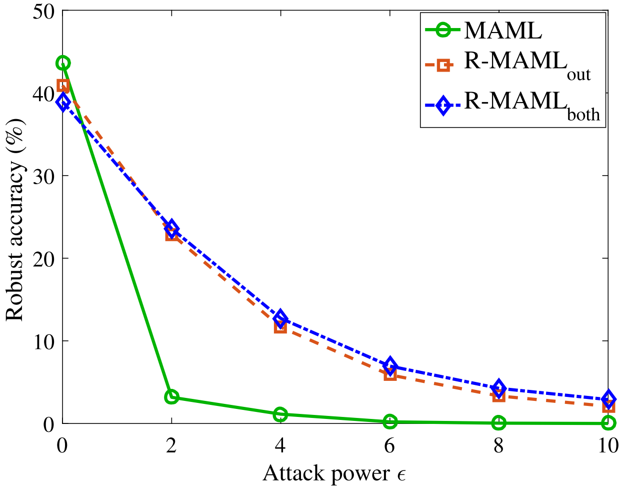

Figure 17.1 RA of metamodels trained by standard MAML, R-MAML$_{\text{both}}$, and R-MAML$_{\text{out}}$ versus PGD attacks of different perturbation sizes during meta-testing. Results show that robustness-regularized meta-update with standard fine-tuning (namely, R-MAML$_{\text{out}}$) has already been effective in promotion of robustness.

Robust meta-update provides robustness adaptation without additional adversarial fine-tuning at meta-testing. Meta-testing includes only the fine-tuning stage. Therefore we need to explore if standard fine-tuning is sufficient to maintain the robustness. Suppose that R-MAML$_{\text{out}}$ is adopted as the meta-training method to solve problem (17.3); we then ask if robustness-regularized meta-testing strategy can improve the robustness of fine-tuned model at downstream tasks. Surprisingly, we find that making an additional effort to adversarially fine-tune the metamodel (trained by R-MAML$_{\text{out}}$) during testing does *not* provide an obvious robustness improvement over the standard fine-tuning scheme during testing (Table 17.1). This consis-

Table 17.1 Comparison of different strategies in meta-testing on R-MAML$_{\text{out}}$: (a) standard fine-tuning (S-FT), (b) adversarial fine-tuning (A-FT).

	S-FT	**A-FT**
SA	40.9%	39.6%
RA	22.9%	23.5%

tently implies that robust meta-update (R-MAML$_{\text{out}}$) is sufficient to render intrinsic robustness in its learnt metamodel regardless of fine-tuning strategies used at meta-testing.

Summary: In Wang et al. (2021b), as inspired by Wong et al. (2020a), they find that the application of the fast sign gradient method (FGSM) to R-MAML$_{\text{out}}$ provides the most graceful tradeoff between the computation cost and the standard and robust accuracies. Moreover, they find that with the help of unlabeled data, R-MAML$_{\text{out}}$-TRADES improves the accuracy-robustness tradeoff over its supervised counterpart R-MAML$_{\text{out}}$ using either AT or TRADES regularization. Finally, they propose a general but efficient robustness-regularized meta-learning framework, which allows the use of unlabeled data augmentation, fast (one-step) adversarial example generation during meta-updating and partial model training during fine-tuning (only fine-tuning the classifier's head).

17.2 Adversarial robustness preservation for contrastive learning: from pretraining to finetuning

Contrastive learning (CL) can learn generalizable feature representations and achieve state-of-the-art performance of downstream tasks by fine-tuning a *linear* classifier on top of it. Early approaches for unsupervised representation learning leverages handcrafted tasks, like prediction rotation (Gidaris et al., 2018), solving the Jigsaw puzzle (Noroozi and Favaro, 2016; Carlucci et al., 2019), and geometry prediction (Gan et al., 2018) and Selfie (Trinh et al., 2019). Recently, contrastive learning (CL) (Chen et al., 2018e; Wang and Isola, 2020; Chen et al., 2020e; van den Oord et al., 2018; He et al., 2020; Chen et al., 2020d) and its variants (Grill et al., 2020; Tian et al., 2020; Purushwalkam and Gupta, 2020; Chen and He, 2020) have demonstrated superior abilities in learning generalizable features in an unsupervised manner. The main idea behind CL is to selfcreate positive samples of the same image from aggressive viewpoints and then acquire data representations by maximizing agreement between positives while contrasting with negatives. However, as adversarial robustness becomes vital in image classification, it remains unclear whether or not CL is able to preserve robustness to downstream tasks. The main challenge is that in the "self-supervised pretraining + supervised finetuning" paradigm, adversarial robustness is easily forgotten due to a learning task mismatch from pretraining to finetuning. We call such challenge "cross-task robustness transferability". To address the above problem, Fan et al. (2021) show that: (i) the design of contrastive views matters: high-frequency components of images are beneficial to improving model robustness; and (ii) augmenting CL with pseudo-supervision stimulus (e.g., resorting to feature clustering) helps preserve robustness without forgetting. They further propose AdvCL, a novel adversarial contrastive pretraining framework, to enhance cross-task robustness transferability without loss of model accuracy and finetuning efficiency. We will introduce this framework in this section.

Fan et al. (2021) focus on the study of accomplishing robustness enhancement using CL without losing its finetuning efficiency, e.g., via a standard linear finetuner. Some relevant works such as (Jiang et al., 2020; Kim et al., 2020) integrate adversarial training with CL. However, the achieved adversarial robustness at downstream tasks largely relies on the use of advanced finetuning techniques, either adversarial full finetuning (Jiang et al., 2020) or adversarial linear finetuning (Kim et al., 2020). Fan et al. (2021) find that self-supervised learning (including the state-of-the-art CL) suffers a new robustness challenge called "cross-task robustness transferability", which was largely overlooked in the previous work; that is, there exists a task mismatch from pretraining to finetuning (e.g., from CL to supervised classification) so that adversarial robustness is not able to transfer across tasks even if pretraining datasets and finetuning datasets are drawn from the same distribution. Different from supervised/semi-supervised learning, this is a characteristic behavior of self-supervision when being adapted to robust learning. As shown in Fig. 17.2, their AdvCLwork advances CL in the adversarial context, and the proposed method outperforms all state-of-the-art baseline methods, leading to a substantial improvement in both robust accuracy and standard accuracy using either the lightweight standard linear finetuning or end-to-end adversarial full finetuning.

In what follows, we elaborate on the *formulation of SimCLR* (Chen et al., 2018e), one of the most commonly used CL frameworks, which this paper focuses on. To be concrete, let $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ denote an *unlabeled*

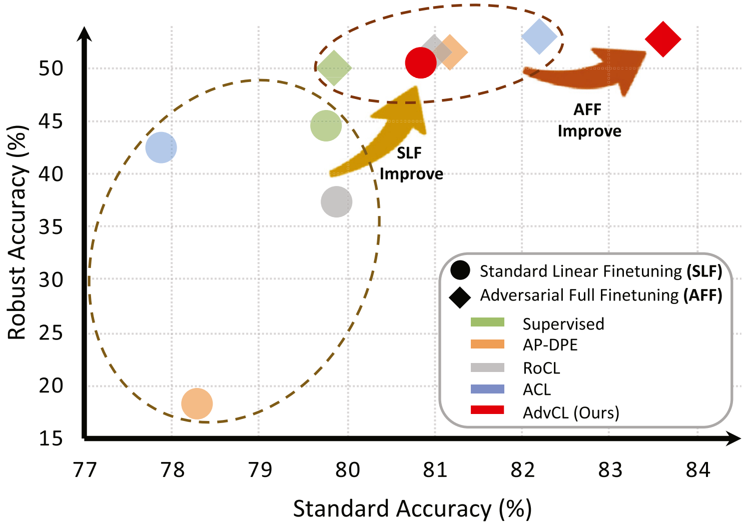

Figure 17.2 Summary of performance for various robust pretraining methods on CIFAR-10. The covered baseline methods include AP-DPE (Chen et al., 2020c), RoCL (Kim et al., 2020), ACL (Jiang et al., 2020), and supervised adversarial training (AT) (Madry et al., 2018). Upper-right indicates better performance with respect to standard accuracy and robust accuracy (under PGD attack with 20 steps and $8/255$ ℓ_∞-norm perturbation strength). Different colors represent different pretraining methods, and different shapes represent different finetuning settings. Circles indicate *Standard Linear Finetuning* (SLF), and Diamonds indicates *Adversarial Full Finetuning* (AFF). The method proposed by Fan et al. (2021) (AdvCL, red circle/diamond (dark gray in print version)) has the best performance across finetuning settings.

source dataset; SimCLR offers a learned *feature encoder* f_θ to generate expressive deep representations of the data. To train $f_{\boldsymbol{\theta}}$, each input $x \in \mathcal{X}$ will be transformed into two *views* $(\tau_1(x), \tau_2(x))$ and label them as a positive pair. Here transformation operations τ_1 and τ_2 are randomly sampled from a predefined transformation set $\mathcal{T}$, which includes, e.g., random cropping and resizing, color jittering, rotation, and cutout. The positive pair is then fed in the feature encoder f_θ with a projection head g to acquire projected features, i.e., $z_i = g \circ f_\theta(\tau_i(x))$ for $j \in \{1, 2\}$. *NT-Xent loss* (i.e., the normalized temperature-scaled cross-entropy loss) is then applied to optimizing f_θ, where the distance of projected positive features (z_1, z_2) is minimized for each input x. SimCLR follows the "*self-supervised pretraining + supervised finetuning*" paradigm, that is, once f_θ is trained, a downstream supervised classification task can be handled by just finetuning a linear classifier ϕ over the fixed encoder f_θ, leading to the eventual classification network $\phi \circ f_\theta$.

Robust pretraining + linear finetuning. We aim to develop robustness enhancement solutions by fully exploiting and exploring the power of CL at the pretraining phase, so that the resulting robust feature representations can seamlessly be used to generate robust predictions of downstream tasks using just a lightweight finetuning scheme. With the aid of adversarial training (AT), we formulate the "*robust pretraining + linear finetuning*" problem as follows:

$$\text{Pretraining: } \min_{\theta} \mathbb{E}_{x \in \mathcal{X}} \max_{\|\delta\|_\infty \le \epsilon} \ell_{\text{pre}}(x + \delta, x; \theta), \tag{17.4}$$

$$\text{Finetuning: } \min_{\theta_c} \mathbb{E}_{(x,y) \in \mathcal{D}} \ell_{\text{CE}}(\phi_{\theta_c} \circ f_\theta(x), y), \tag{17.5}$$

where ℓ_{pre} denotes a properly designed robustness- and generalization-aware CL loss given as a function of the adversarial example $(x+\delta)$, original example x, and feature encoder parameters θ. In (17.4), $\phi_{\theta_c} \circ f_\theta$ denotes the classifier by equipping the linear prediction head ϕ_{θ_c} (with parameters θ_c to be designed) on top of the fixed feature encoder f_θ, and ℓ_{CE} denotes the supervised CE loss over the target dataset $\mathcal{D}$. Note that besides the standard linear finetuning (17.5), we can also modify (17.5) using the worst-case CE loss for adversarial linear/full finetuning (Jiang et al., 2020; Kim et al., 2020). We do not consider standard full finetuning since tuning the full network weights with standard cross-entropy loss is not possible for the model to preserve robustness (Chen et al., 2020c).

ADVCL *framework.* The ADVCL framework proposed by Fan et al. (2021) includes two main components, robustness-aware view selection and pseudo-supervision stimulus generation. In particular, they advance the view selection mechanism by taking into account proper frequency-based data transformations that are beneficial to robust representation learning and pretraining generalization ability. Furthermore, they propose to design and integrate proper supervision stimulus into ADVCL to improve the cross-task robustness transferability since robust representations learned from self-supervision may lack the class-discriminative ability required for robust predictions on downstream tasks. An overview of ADVCL is provided in Fig. 17.3. In contrast to standard CL, Fan et al. (2021) propose two additional contrastive views, the adversarial and frequency views.

Multiview CL loss. Prior to defining new views, we first review the NT-Xent loss and its multiview version used in CL. The contrastive loss with

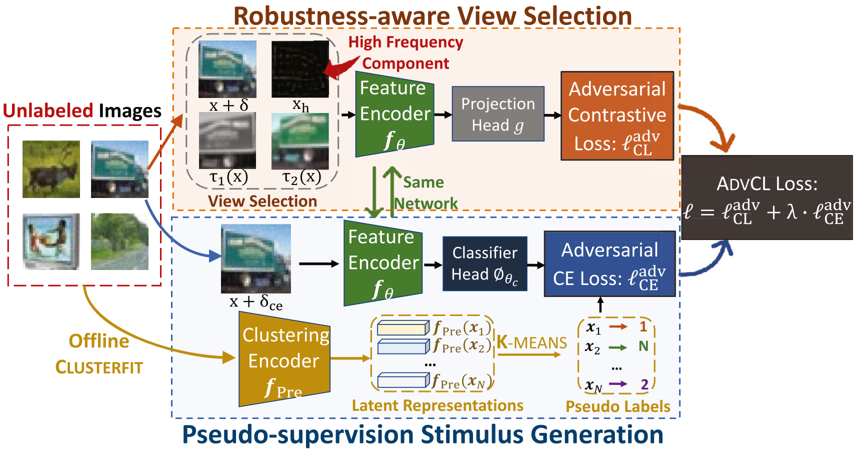
Robustness-aware View Selection
High Frequency Component
Unlabeled Images
$x+\delta$
x_h
$\tau_1(x)$
$\tau_2(x)$
View Selection
Feature Encoder f_θ
Projection Head g
Adversarial Contrastive Loss: $\ell_{\mathrm{CL}}^{\mathrm{adv}}$
Same Network
AdvCL Loss: $\ell = \ell_{\mathrm{CL}}^{\mathrm{adv}} + \lambda \cdot \ell_{\mathrm{CE}}^{\mathrm{adv}}$
$x+\delta_{\mathrm{ce}}$
Feature Encoder f_θ
Classifier Head ϕ_{θ_c}
Adversarial CE Loss: $\ell_{\mathrm{CE}}^{\mathrm{adv}}$
Offline ClusterFit
Clustering Encoder f_{Pre}
$f_{\mathrm{Pre}}(x_1)$
$f_{\mathrm{Pre}}(x_2)$
...
$f_{\mathrm{Pre}}(x_N)$
K-MEANS
$x_1 \to 1$
$x_2 \to$ N
...
$x_N \to 2$
Latent Representations
Pseudo Labels
Pseudo-supervision Stimulus Generation

Figure 17.3 The overall pipeline of AdvCL in (Fan et al., 2021). It mainly has two ingredients, robustness-aware view selection (orange (mid gray in print version) box) and pseudo-supervision stimulus generation (blue (gray in print version)box). The view selection mechanism is advanced by high-frequency components, and the supervision stimulus is created by generating pseudo-labels for each image through ClusterFit. The pseudo-label (in yellow (light gray in print version) color) can be created in an offline manner and will not increase the computation overhead.

respect to a positive pair $(\tau_1(x), \tau_2(x))$ of each (unlabeled) data x is given by

$$\ell_{\mathrm{CL}}(\tau_1(x), \tau_2(x)) = -\sum_{i=1}^{2} \sum_{j \in \mathcal{P}(i)} \log \frac{\exp\left(\mathrm{sim}(z_i, z_j)/t\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\mathrm{sim}(z_i, z_k)/t\right)}, \tag{17.6}$$

where recall that $z_i = g \circ f(\tau_i(x))$ is the projected feature under the ith view, $\mathcal{P}(i)$ is the set of positive views except i (e.g., $\mathcal{P}(i) = \{2\}$ if $i = 1$), $\mathcal{N}(i)$ denotes the set of augmented batch data except the point $\tau_i(x)$, the cardinality of $\mathcal{N}(i)$ is $(2b-1)$ (for a data batch of size b under two views), $\mathrm{sim}(z_{i1}, z_{i2})$ denotes the cosine similarity between representations from two views of the same data, exp denotes exponential function, $\mathrm{sim}(\cdot, \cdot)$ is the cosine similarity between two points, and $t > 0$ is a temperature parameter. The two-view CL objective can be further extend to the *multiview contrastive loss* (Khosla et al., 2020):

$$\ell_{\mathrm{CL}}(\tau_1(x), \tau_2(x), \ldots, \tau_m(x)) = -\sum_{i=1}^{m} \sum_{j \in \mathcal{P}(i)} \log \frac{\exp\left(\mathrm{sim}(z_i, z_j)/t\right)}{\sum_{k \in \mathcal{N}(i)} \exp\left(\mathrm{sim}(z_i, z_k)/t\right)}, \tag{17.7}$$

where $\mathcal{P}(i) = [m]/\{i\}$ denotes the m positive views except i, $[m]$ denotes the integer set $\{1, 2, \ldots, m\}$, and $\mathcal{N}(i)$, with cardinality $(bm - 1)$, denotes the set of m-view augmented b batch samples except the point $\tau_i(x)$.

Contrastive view from adversarial example. The methods proposed by Jiang et al. (2020), Kim et al. (2020), and Gowal et al. (2021) can be explained based on (17.6): an adversarial perturbation $\boldsymbol{\delta}$ with respect to each view of a sample x is generated by maximizing the contrastive loss:

$$\delta_1^*, \delta_2^* = \underset{\|\delta_i\|_\infty \leq \epsilon}{\operatorname{argmax}}\, \ell_{\mathrm{CL}}(\tau_1(x) + \delta_1, \tau_2(x) + \delta_2). \tag{17.8}$$

A solution to problem (17.8) eventually yields a *paired* perturbation view $(\tau_1(x) + \delta_1^*, \tau_2(x) + \delta_2^*)$. However, the definition of adversarial view (17.8) used by Jiang et al. (2020), Kim et al. (2020), and Gowal et al. (2021) may not be proper. First, standard CL commonly uses *aggressive* data transformation that treats small portions of images as positive samples of the full image (Purushwalkam and Gupta, 2020). Despite its benefit to promoting generalization, crafting perturbations over such aggressive data transformations may not be suitable for defending adversarial attacks applied to *full* images in the adversarial context. Thus a new adversarial view built upon x rather than $\tau_i(x)$ is desired. Second, the contrastive loss (17.6) is only restricted to two views of the same data. As will be evident later, the multiview contrastive loss is also needed when taking into account multiple robustness-promoting views. Spurred by above, we define the *adversarial view* over x without modifying the existing data augmentations $(\tau_1(x), \tau_2(x))$. This leads to the following adversarial perturbation generator by maximizing a three-view contrastive loss

$$\delta^* = \underset{\|\delta\| \leq \epsilon}{\operatorname{argmax}}\, \ell_{\mathrm{CL}}(\tau_1(x), \tau_2(x), x + \delta), \tag{17.9}$$

where $x + \delta^*$ is regarded as the third view of x.

Contrastive view from high-frequency component. Next, we use the high-frequency component (HFC) of data as another additional contrastive view. The rationale arises from the facts that 1) learning over HFC of data is a main cause of achieving superior generalization ability (Wang et al., 2020b) and 2) an adversary typically concentrates on HFC when manipulating an example to fool model decision (Wang et al., 2020f). Let $\mathcal{F}$ and $\mathcal{F}^{-1}$ denote the Fourier transformation and its inverse. An input image x can then be decomposed into its HFC x_{h} and low-frequency component (LFC) x_{l}:

$$x_{\mathrm{h}} = \mathcal{F}^{-1}(q_{\mathrm{h}}), \quad x_{\mathrm{l}} = \mathcal{F}^{-1}(q_{\mathrm{l}}), \quad [q_{\mathrm{h}}, q_{\mathrm{l}}] = \mathcal{F}(x). \tag{17.10}$$

In (17.10) the distinction between q_{h} and q_{l} is made by a hard thresholding operation. Let $q(i,j)$ denote the (i,j)th element of $\mathcal{F}(x)$, and let $c = (c_1, c_2)$ denote the centroid of the frequency spectrum. The components q_{l} and q_{h} in (17.10) are then generated by filtering out values according to the distance from c: $q_h(i,j) = \mathbb{1}_{[d((i,j),(c_1,c_2))\geq r]} \cdot q(i,j)$, and $q_l(i,j) = \mathbb{1}_{[d((i,j),(c_1,c_2))\leq r]} \cdot q(i,j)$, where $d(\cdot,\cdot)$ is the Euclidean distance between two spatial coordinates, r is a predefined distance threshold ($r = 8$ in all our experiments), and $\mathbb{1}_{[\cdot]} \in \{0, 1\}$ is an indicator function, which equals to 1 if the condition within $[\cdot]$ is met and 0 otherwise.

Robustness-aware contrastive learning objective. By incorporating the adversarial perturbation δ and disentangling HFC x_{h} from the original data x we obtain a four-view contrastive loss (17.7) defined over $(\tau_1(x), \tau_2(x), x + \delta, x_{\mathrm{h}})$:

$$\ell_{\mathrm{CL}}^{\mathrm{adv}}(\theta; \mathcal{X}) := \mathbb{E}_{x\in\mathcal{X}} \max_{\|\delta\|_\infty \leq \epsilon} \ell_{\mathrm{CL}}(\tau_1(x), \tau_2(x), x+\delta, x_{\mathrm{h}}; \theta), \tag{17.11}$$

where recall that $\mathcal{X}$ denotes the unlabeled dataset, $\epsilon > 0$ is a perturbation tolerance during training, and for clarity, the four-view contrastive loss (17.7) is explicitly expressed as a function of model parameters θ. The eventual learning objective ADVCL will be built upon (17.11).

Supervision stimulus generation: ADVCL *empowered by* CLUSTERFIT. On top of (17.11), we further improve the robustness transferability of learned representations by generating a proper supervision stimulus. The rationale is that robust representation could lack the class-discriminative power required by robust classification as the former is acquired by optimizing an unsupervised contrastive loss, whereas the latter is achieved by a supervised cross-entropy CE loss. However, there is no knowledge about supervised data during pretraining. To improve cross-task robustness transferability without calling for supervision, we take advantage of CLUSTERFIT (Yan et al., 2020), a pseudo-label generation method used in representation learning.

To be more concrete, let f_{pre} denote a pretrained representation network that can generate latent features of unlabeled data. Note that f_{pre} can be set available beforehand and trained over either supervised or unsupervised dataset $\mathcal{D}_{\mathrm{pre}}$, e.g., ImageNet using CL in experiments. Given (normalized) pretrained data representations $\{f_{\mathrm{pre}}(x)\}_{x\in\mathcal{X}}$, CLUSTERFIT uses K*-means clustering* to find K data clusters of $\mathcal{X}$ and maps a *cluster index* c to a *pseudo-label*, resulting in the pseudo-labeled dataset $\{(x, c) \in \hat{\mathcal{X}}\}$. By integrating CLUSTERFIT with (17.11) the eventual training objective of ADVCL is then formed by

$$\min_{\theta} \ \ell_{\mathrm{CL}}^{\mathrm{adv}}(\theta; \mathcal{X}) + \lambda \min_{\theta, \theta_c} \ \underbrace{\mathbb{E}_{(x,c) \in \hat{\mathcal{X}}} \max_{\|\delta_{ce}\|_\infty \le \epsilon} \ell_{\mathrm{CE}}(\phi_{\theta_c} \circ f_\theta(x + \delta_{ce}), c)}_{\text{Pseudo-classification enabled AT regularization}}, \tag{17.12}$$

where $\hat{\mathcal{X}}$ denotes the pseudo-labeled dataset of $\mathcal{X}$, ϕ_{θ_c} denotes a prediction head over f_θ, and $\lambda > 0$ is a regularization parameter that strikes a balance between adversarial contrastive training and pseudo-label stimulated AT. When the number of clusters K is not known a priori, we extend (17.12) to an *ensemble version* over n choices of cluster numbers $\{K_1, \ldots, K_n\}$. Here each cluster number K_i is paired with a unique linear classifier ϕ_i to obtain the supervised prediction $\phi_i \circ f$ (using cluster labels). The ensemble CE loss, given by the average of n individual losses, is then used in (17.12). The experiments by Fan et al. (2021) show that the ensemble version usually leads to better generalization ability.

Empirical comparison. Following Fan et al. (2021), we consider three robustness evaluation metrics: (1) Auto-attack accuracy (**AA**), namely, classification accuracy over adversarially perturbed images via auto-attacks; (2) Robust accuracy (**RA**), namely, classification accuracy over adversarially perturbed images via PGD attacks; and (3) Standard accuracy (**SA**), namely, standard classification accuracy over benign images without perturbations. We use ResNet-18 for the encoder architecture of $f_{\boldsymbol{\theta}}$ in CL. Unless specified otherwise, we use five-step ℓ_∞ projected gradient descent (PGD) with $\epsilon = 8/255$ to generate perturbations during pretraining and use auto-attack and 20-step ℓ_∞ PGD with $\epsilon = 8/255$ to generate perturbations in computing AA and RA at test time. We will compare AdvCL with the CL-based adversarial pretraining *baselines*, ACL (Jiang et al., 2020), RoCL (Kim et al., 2020), (non-CL) self-supervised adversarial learning baseline AP-DPE (Chen et al., 2020c), and the supervised AT baseline (Madry et al., 2018).

Overall performance from pretraining to finetuning (across tasks). In Table 17.2, we evaluate the robustness of a classifier (ResNet-18) finetuned over robust representations learned by different supervised/self-supervised pretraining approaches over CIFAR-10 and CIFAR-100. We focus on two representative finetuning schemes, the simplest standard linear finetuning (SLF) and the end-to-end adversarial full finetuning (AFF). As we can see, the proposed AdvCL method yields a substantial improvement over almost all baseline methods. Moreover, AdvCL simultaneously improves robustness and standard accuracy.

Table 17.2 Cross-task performance of AdvCL (in dark gray color), compared with supervised (in white color) and self-supervised (in light gray color) baselines, in terms of AA, RA, and SA on CIFAR-10 with ResNet-18. The pretrained models are evaluated under the standard linear finetuning (SLF) setting and the adversarial full finetuning (AFF) setting. The top performance is highlighted in **bold**.

Pretraining Method	Finetuning Method	CIFAR-10			CIFAR-100		
		AA (%)	RA (%)	SA (%)	AA (%)	RA (%)	SA (%)
Supervised	Standard linear finetuning **(SLF)**	42.22	44.4	79.77	19.53	23.41	**50.53**
AP-DPE (Chen et al., 2020c)		16.07	18.22	78.30	4.17	6.23	47.91
RoCL (Kim et al., 2020)		28.38	39.54	79.90	8.66	18.79	49.53
ACL (Jiang et al., 2020)		39.13	42.87	77.88	16.33	20.97	47.51
AdvCL (Fan et al., 2021)		**42.57**	**50.45**	**80.85**	**19.78**	**27.67**	48.34
Supervised	Adversarial full finetuning **(AFF)**	46.19	49.89	79.86	21.61	25.86	52.22
AP-DPE (Chen et al., 2020c)		48.13	51.52	81.19	22.53	26.89	55.27
RoCL (Kim et al., 2020)		47.88	51.35	81.01	22.38	27.49	55.10
ACL (Jiang et al., 2020)		49.27	**52.82**	82.19	23.63	**29.38**	56.61
AdvCL (Fan et al., 2021)		**49.77**	52.77	**83.62**	**24.72**	28.73	**56.77**

Table 17.3 Cross-dataset performance of AdvCL (dark gray color), compared with supervised (white color) and self-supervised (light gray) baselines in AA, RA, SA on STL-10 with ResNet-18.

Method	Fine-tuning	CIFAR-10 → STL-10			CIFAR-100 → STL-10		
		AA (%)	RA (%)	SA (%)	AA (%)	RA (%)	SA (%)
Supervised	**SLF**	22.26	30.45	54.70	19.54	23.63	**51.11**
RoCL (Kim et al., 2020)		18.65	28.18	54.56	12.39	21.93	47.86
ACL (Jiang et al., 2020)		25.29	31.80	55.81	**21.75**	26.32	45.91
AdvCL (Fan et al., 2021)		**25.74**	**35.80**	**63.73**	20.86	**30.35**	50.71
Supervised	**AFF**	33.10	36.7	62.78	29.18	32.43	55.85
RoCL (Kim et al., 2020)		29.40	34.65	61.75	27.55	31.38	57.83
ACL (Jiang et al., 2020)		32.50	35.93	62.65	28.68	32.41	57.16
AdvCL (Fan et al., 2021)		**34.70**	**37.78**	**63.52**	**30.51**	**33.70**	**61.56**

Table 17.4 Performance (RA and SA) of AdvCL (in dark gray color) and baseline approaches on CIFAR-10 under different linear finetuning strategies, SLF and adversarial linear finetuning (ALF).

Method	SLF		ALF	
	RA (%)	SA (%)	RA (%)	SA (%)
Supervised	44.40	79.77	46.75	79.06
RoCL (Kim et al., 2020)	39.54	79.90	43.11	77.33
ACL (Jiang et al., 2020)	42.87	77.88	45.40	77.71
AdvCL (Fan et al., 2021)	**50.45**	**80.85**	**52.01**	**79.39**

Robustness transferability across datasets. In Table 17.3, we next evaluate the robustness transferability across different datasets, where $A \to B$ denotes the transferability from pretraining on dataset A to finetuning on another dataset B ($\neq A$) of representations learned by AdvCL. Here the pretraining setup is consistent with Table 17.2. We observe that AdvCL yields better robustness and standard accuracy than almost all baseline approaches under both SLF and AFF finetuning settings. In the case of CIFAR-100 → STL-10, although AdvCL yields 0.89% AA drop compared to ACL (Jiang et al., 2020), it yields a much better SA with 4.8% improvement.

Linear finetuning types. We also study the robustness difference when different linear finetuning strategies: *Standard* linear finetuning (SLF) and

Adversarial linear finetuning (ALF) are applied. Table 17.4 shows the performance of models trained with different pretraining methods. As we can see, our ADVCL achieves the best performance under both linear finetuning settings and outperforms baseline approaches in a large margin. We also note that the performance gap between SLF and ALF induced by our proposal ADVCL is much smaller than in other approaches, and ADVCL with SLF achieves much better performance than baseline approaches with ALF. This indicates that the representations learned by ADVCL is already sufficient to yield satisfactory robustness.