Meta-Learning and Self-Supervised Pretraining for Few-shot Image Translation

by

Ileana Rugina

S.B., Physics and Computer Science and Engineering,
Massachusetts Institute of Technology, 2019

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2021

Author . . . .
Department of Electrical Engineering and Computer Science
May 20, 2021

Certified by . . . .
Marin Soljačić
Professor of Physics
Thesis Supervisor

Accepted by . . . .
Katrina LaCurts
Chair, Master of Engineering Thesis Committee

This page intentionally left blank

Meta-Learning and Self-Supervised Pretraining for Few-shot Image Translation

by

Ileana Rugina

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2021, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Recent advances in machine learning (ML) and deep learning in particular, enabled by hardware advances and big data, have provided impressive results across a wide range of computational problems such as computer vision, natural language, or reinforcement learning. Many of these improvements are however constrained to problems with large-scale curated data-sets which require a lot of human labor to gather. Additionally, these models tend to generalize poorly under both slight distributional shifts and low-data regimes. In recent years, emerging fields such as meta-learning or self-supervised learning have been closing the gap between proof-of-concept results and real-life applications of ML.

We follow this line of work and contribute a novel few-shot multi-task image to image translation problem. We then present several benchmarks for this problem using ideas from both meta-learning and contrastive-learning and improve upon baselines trained using simple supervised learning. Additionally, we contribute to another area of growing interest—applying deep learning to physical problems—and focus our efforts on modeling weather phenomena.

We define an image translation problem between different radar and satellite sensor modalities and leverage spatial and temporal locality to pose it as a multi-task problem. We improve upon naive solutions that ignore this hierarchical dataset structure and demonstrate the effectiveness of meta-learning methods to solving real-world problems. We make our code available here.

Thesis Supervisor: Marin Soljačić
Title: Professor of Physics

This page intentionally left blank

Acknowledgments

First of all, I would like to thank my thesis supervisor Professor Marin Soljačić for his guidance and support throughout the past two years and my mentor Rumen Dangovski for all his help and the countless hours spent working together. I am also indebted to Dr. Preslav Nakov for his mentorship in the past years, as well as all the students in our group who have welcomed and helped me throughout my MEng.

I would also like to thank all the amazing people I have met throughout working on AIA, and in particular those who have supported my work on the Continual and Few-Shot Learning project: Prof. Pulkit Agrawal, Dr. Brian Cheung. Dr. Pooya Khorrami, Capt. John Radovan, Dr. Olga Simek, and Dr. Mark Veillette.

Additionally, I would like to thank Dr. Jason Miller, Prof. Daniel Sanchez, and Dr. Silvina Hanono Wachman for supporting me throughout my 6.004 TA-ship when I started my MEng.

Last but not least, I am very grateful to my friends and family who've supported me throughout the past six years and made my time at MIT as enjoyable as it has been.

Research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

This page intentionally left blank

Contents

List of Figures

List of Tables

This page intentionally left blank

Chapter 1

Introduction

Benchmarks such as ImageNet [9] in computer vision or SQuAD [35] in natural language processing have been pivotal in popularizing deep-learning techniques and demonstrating their power. More recently, works such as ObjectNet [3] in vision have shown impressive performance on these established benchmarks do not translate to good performance in real-world situations, where the datasets might be less structured or more diverse. There is a lot of interest in devising more challenging datasets, both of general interest as well as domain-specific applications, that more closely resemble real-world situations practitioners might encounter when trying to put machine learning models into production.

Growing fields such as self-supervised [30] or multi-task learning [19] reflect these interests and provide promising solutions to the aforementioned issues. However, the problem of model evaluation remains: for example, in few-shot learning model evaluation is currently largely constrained to Omniglot [28, 27] (which has essentially been saturated), Miniimagenet [48] and Metadataset [44]. We address these known limitations in our field by contributing a new computer vision multi-task problem and move away from classification problems towards the field of image-generation by leveraging a weather dataset (that has been recently been introduced to the ML community) to formulate a novel few-shot image-to-image translation problem.

The rest of this thesis is organized as follows:

- **(Chapter 2) Background:** we review training techniques (adversarial and gradient-based meta-learning), pretraining techniques (contrastive learning), model architectures, and the meteorological dataset we will use throughout our work.

- **(Chapter 3) Related Work:** we present recent progress in low-data and/or multi-task settings that are largely orthogonal with our work and could be used either in conjunction with or as an additional benchmark to our contribution. These include data augmentation and regularization techniques in adversarial training, as well as different approaches to multi-task and generative learning.

- **(Chapter 4) Meta-Learning:** we define the few-shot image-to-image translation benchmark we propose and present the settings of all our experiments. We evaluate models trained with MAML or joint optimization on adversarial or reconstruction losses and present empirical differences between all these approaches. In many cases we show MAML provides improvements over the joint training baselines.

- **(Chapter 5) Self-Supervised Pretraining:** We turn our attention to self-supervised pretraining and show it can provide improvements in the most simple scenario above.

- **(Chapter 6) Conclusion and Future Work:** We conclude this thesis and present several experimental and theoretical directions for future work.

Chapter 2

Background

2.1 Training Techniques

We first review a couple of established bilevel gradient-based optimization techniques, model-agnostic meta-learning and adversarial training, and specify their associated loss functions, training algorithms, and general properties.

2.1.1 Model Agnostic Meta Learning (MAML)

Model Agnostic Meta Learning or MAML [12, 16] is a bilevel optimization algorithm that seeks to find a good weight initialization for multi-task problems and bias the optimization procedure towards models that are easy to fine-tune.

Let the model we want to meta-learn be parameterized by θ and assume we have J tasks available ($\mathcal{T}_j$ for $j \in \{1, 2, \dots J\}$), each with a loss function $\mathcal{L}_{\mathcal{T}_j}$. Each task has a limited number of train examples, which we split into N support and M query examples. On each task $\mathcal{T}_j$ we finetune the model weights from θ to task-specific weights $\phi_j(\theta)$ using the N support datapoints x_{j_n}, $j_n \in \{1, 2, \dots N\}$, through a simple gradient descent with early stopping inner-loop optimization procedure. In the case of a single inner gradient step the finetuned weights $\phi_j(\theta)$ are:

$$\phi_j(\theta) = \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_j}(\theta; x_{j_n}) \tag{2.1}$$

We then evaluate the finetuned weights ϕ_j using the M query examples $x_{j_{N+m}}$, $j_{N+m} \in \{N+1, N+2, \ldots N+M\}$, and compute losses $\mathcal{L}_{\mathcal{T}_j}(\phi_j; x_{j_{N+m}})$. In the aforementioned single-step inner loop case this become $\mathcal{L}_{\mathcal{T}_j}(\theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_j}(\theta; x_{j_n}); x_{j_{N+m}})$.

We described the inner loop or adaptation procedure above, which is also how we evaluate models in the MAML framework. For training, we perform the above procedure for several tasks within meta-batches and do an outer-loop optimization where we search for a good common initialization for all train-tasks:

$$\arg\min_\theta \frac{1}{J} \sum_{j \in \{1, \ldots J\}} \mathcal{L}_{\mathcal{T}_j}(\phi_j(\theta); x_{j_{N+m}})$$

using algorithms in the family of stochastic gradient descent (usually Adam [24]).

2.1.2 Adversarial Training

Model: Generative Adversarial Networks (GAN)[14] are deep generative models that learn to sample from an unknown distribution by playing a two-player minimax game with an additional network that learns to discriminate between real train examples and generated samples. In the original formulation the generator and discriminator networks minimize or maximize, respectively, the following objective:

$$\mathcal{L} = \mathbb{E}_z \log\left(1 - D(G(z))\right) + \mathbb{E}_x \log\left(D(x)\right), \tag{2.2}$$

where $z \propto p_z(z)$ is a stochastic noisy variable that the generator uses to output a distribution over samples. Real examples are drawn from an unknown $x \propto p_{\text{data}}(x)$. The transformation G, applied on z, induces a transformed probability distribution in sample space $p_G(x)$. [14] show the optimal discriminator is $D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_G(x)}$, under which the generator's objective functions becomes $2 \cdot JS(p_G || p_{\text{data}}) - 2\log 2$, where $JS(p_G||p_{\text{data}})$ is the Jensen-Shannon divergence. This divergence is minimized when $p_G(x) = p_{\text{data}}(x)$.

Training: GANs are trained by applying stochastic alternate gradient descent-ascent algorithms to a saddle-point problem, which often results in training instabilities, exploding losses, or cycling behavior. In more controlled settings, which generally assume convex-concave problems, the optimization community has proposed alternate algorithms (such as optimistic gradient descent or extra-gradient methods) which provably converge with good rates [33]. For example, [8] trained GANs with optimistic gradient descent methods and improved generated images' quality. Another difficulty in training GANs is the issue of exploding gradients: the gradient signal to G from Eq. 2.2 vanishes when the discriminator classifier can perfectly discern between real and fake samples. Thus, [14] propose making the generator maximize $\mathbb{E}_z \log\left(D(G(z))\right)$ rather than minimizing $\mathbb{E}_z \log\left(1 - D(G(z))\right)$ because the former objective is not-saturating.

Regularization: [1] minimize an approximation to the Wasserstein distance rather than divergence loss functions and introduced Wasserstein GANs (WGANs). They compute Wasserstein distances though the Kantorovich-Rubinstein duality and require the discriminator to be a Lipschitz continuous function. They enforce this condition through a straightforward gradient-clipping procedure, which has since been refined by works such as [17] to use gradient penalties in the loss function. [26] note the GAN objective function does not consistently correspond to divergences such as KL or JS and view GAN training though the lens of regret minimization instead. They explain training instabilities, and mode collapse in particular, by sharp discriminator gradients around real samples and propose a gradient penalty scheme with good results. Similarly, [11] show reducing the two-player zero-sum game between the generator and discriminator networks to a divergence minimization process does not always explain situations where GAN training successfully converges. They also show that controlling the norm of gradients from the discriminator to the generator helps even in situations where doing so is not theoretically justified as it was in [1]. All these work have shown that an important part of improving and stabilizing GAN training is regularizing the discriminator by limiting its rate of change or enforcing it is a Lipschitz continuous function [54]. [32] constrains linear layers' spectral norms by normalizing weight matrices after each update with an approximation to their maximum singular values.

Pix2Pix: [21] extends this body of work to image-to-image translation. The dataset now comes with aligned multiple views of identical scenes $\{x, y\}$ and we wish to learn to generate y from x. In other words, we are modeling a distribution $p_G(y)$ over the target modality y and the generator $G(z, x)$ is now a function of not only a stochastic variable z, but also the input modality x. We also move towards using conditional GANs [31] and let the discriminator network $D(x, y)$ look at the input modality x in order to judge not only if a generated sample y is realistic, but also if it is aligned with the inputs x. The objective in Eq. 2.2 becomes:

$$\mathcal{L}_{cGAN} = \mathbb{E}_{x,z} \log\left(1 - D(x, G(x, z))\right) + \mathbb{E}_{x,y} \log D(x, y) \tag{2.3}$$

An additional loss term for the generator can be added — tasking it to produce images that not only fool the discriminator, but are also close to ground-truth $\mathcal{L}_{l_1} = \mathbb{E}_{x,y,z}||y - G(x, z)||_1$. In practice, image-to-image architectures trained adversarially drop the explicit stochastic component z and incorporate randomness through architectural elements such as dropout [42].

2.2 Self-Supervised Pre-Training

Self-Supervised learning (SSL) pretraining techniques have provided state-of-the-art results in both natural language processing and computer vision problems by training feature extractors on large unlabelled datasets that construct useful representations of the input modalities.

In computer vision several related frameworks have recently been introduced. They all construct representations of the input by clustering together representations of related inputs (and sometimes pushing apart representations of different examples).

The former are called positive pairs and are different views of the same datapoint, obtained using data augmentations the problem domain should be invariant with respect to. The latter consist of negative pairs and are obtained from different train examples. In computer vision common data augmentations include color transformations such as color jitter, geometric transformations such as translations, rotations, or cropping, and context based ones such as jigsaw puzzles [22].

This pretraining strategy is called contrastive learning and has been used in several recent works with slight variations [6]. The common thread is using a Siamese network [25] with two closely related branches to pull together positive pairs while enforcing the network does not collapse to a constant function by pushing apart negative pairs and/or introducing various constraints or asymmetries between the two branches.

One of the most successful models in this line of work is SimCLR [4, 5], which uses both positive and negative pairs. They create positive pairs by stochastically applying various augmentations to any datapoint x to create two different views $\tilde{x_i}$ and $\tilde{x_j}$ of x. An encoder network f constructs representations $h_i = f(\tilde{x_i})$ and $h_j = f(\tilde{x_j})$, and a simple MLP projector g generates the two vectors $z_i = g(h_i)$ and $z_j = g(h_j)$ used for contrastive learning. They construct negative pairs by assuming that all pairs (z_i, z_k) that did not originate from same input x are negative pairs. For a given mini-batch with N inputs x they construct $2N$ vectors z and for each positive pair (i, j) they define a loss function

$$l_{i,j} = -\log \frac{\exp\left(\text{sim}(z_i, z_j)\right)}{\sum_{k=1,k\neq i}^{N} \exp\left(\text{sim}(z_i, z_j)\right)}$$

Note that the above is not symmetrical in z_i and z_j. The loss-function for the whole batch is an average over all $2N$ positive pairs (z_i, z_j) and (z_j, z_i):

$$\mathcal{L} = \frac{1}{2N} \sum_{\text{positive pairs } (i,j)} (l_{i,j} + l_{j,i})$$

2.3 Model Architecture

U-Net architectures [37] are the established models for image-to-image translation problems. This model augments encoder-decoder architectures with skip connections across the informational bottleneck in order to both compress information into a meaningful latent space, as well as enable gradient flow throughout all layers.

[21] leveraged this established model when proposing an adversarial training strategy for image-to-image translation problems. They add the additional patch discriminator network because this allows for implicitly learning loss functions better suited for image evaluation than the simpler L_1 or L_2 norms. Learning just a U-Net with reconstruction losses leads to blurry samples because they bias the model towards predicting the expected value for each pixel rather than outputting realistic images [29].

The U-Net generator consists of convolutional blocks in both the encoder and decoder parts and use batch normalization and ReLU activation functions at each layer. The encoder and decoder also contain downsampling and upsampling operations, respectively. The discriminator uses a PatchGAN convolutional architectures with the same batch normalization and ReLU block structure and looks at 70×70 regions to discern whether or not its input is a realistic sample or not.

2.4 Storm Event Imagery

The Storm Event Imagery (SEVIR) [47] is a dataset curated by MIT Lincoln Labs to democratize research in radar and satellite meteorology. It is a collection of over 10,000 weather events, each of which tracks 5 sensor modalities within 384 km × 384 km patches for 4 hours. The events are uniformly sampled so that there are 49 frames for each 4 hour period, and the 5 channels consist of:

- 1 visible and 2 IR sensors from the GOES-16 advanced baseline [39]
- vertically integrated liquid (VIL) from NEXTRAD
- lightning flashes from GOES-16

Fig. 2-1 shows examples of the two IR and the VIL modalities. From now on we will consistently disregard the visible channel because it often contains no information as visible radiation is easily occluded.

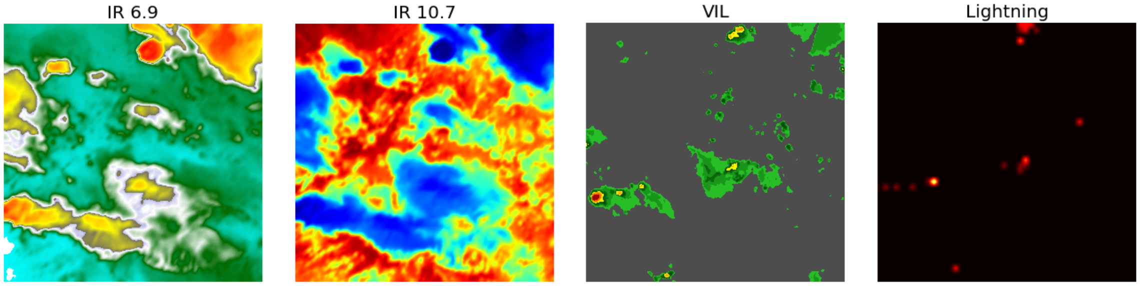

Figure 2-1: **Frame from The Storm Event Imagery (SEVIR) dataset.** We use four of the five available modalities: 2 IR, VIL, and lightning information.

[47] suggested several machine learning problems that can be studied on SEVIR and provided baselines for two of these: nowcasting and synthetic weather radar generation. The former refers to short-term forecasts of either input modality, and the latter to image-to-image translation between different modalities. In practice VIL information is less readily available and they focus on this sensor as the target for both nowcasting and image translation tasks.

In both cases they train U-Net models and experiment with various loss functions. For nowcasting [47] propose a simple mean squared error (MSE) objective, as well as a style and content (SC) loss, and then introduce a patch discriminator for adversarial training with conditional GANs as proposed in [21]. The synthetic radar generation image translation setup follows the same objectives, and additionally experiments with mean absolute error (MAE).

2.4.1 Evaluation

We review common evaluation metrics used in the satellite and radar literature to analyse artificially-generated VIL imagery. They all compare the target and generated samples after binarizing them with an arbitrarily threshold in $[0, 255]$ and looking at counts in the associated confusion matrix. Let H denote the number of true positives, C denote the number of true negatives, M denote the number of false negatives and F the number of false positives. [47] define four evaluation metrics: Critical Succes Index **(CSI)** is equivalent to the intersection over union $\frac{H}{H+M+F}$; Probability of detection **(POD)** is equivalent to recall $\frac{H}{H+M}$; Succes Ratio **(SUCR)** is equivalent to precision $\frac{H}{H+F}$; **BIAS** is defined as $\frac{H+F}{H+M}$.

Chapter 3

Related Work

Most closely related to our work is [7], which is the only prior work we are aware of on the topic of few-shot multi-task image generation. They optimize using Reptile [34], a first-order approximiation to MAML, and evaluate on the MNIST and Omniglot datasets. They also introduce a dataset which presents a very clear delimitation between different tasks and more generally does not exhibit the challenges of modeling real-world phenomena because the examples are icons rather than realistic images.

We focus our attention on [21], the default solution to image-to-image translation problems, which has been extended by works such as [49] to increase photo-realism or [55] to use unpaired image datasets by imposing a cycle-consistency loss function.

3.1 Low Data GAN Training

There has been a lot of interest in training GANs in low-data settings. In this scenario the main challange is that the discriminator network can simply memorize the train-set and quickly reach perfect performance on known examples. In this case training quickly becomes unstable and the generator is not able to create realistic samples. Additionally, the discriminator performs poorly when evaluated on held-out validation or test splits.

3.1.1 Data Augmentation

In classification or regression tasks data augmentation is a technique pivotal to increasing the effective number of train-examples, improving robustness w.r.t. common noisy distortions, and encoding known domain-specific invariant transformations. Applying data augmentation to generative modeling is complicated by the fact that we want to sample from the original data distribution and not include augmentations into our samples. [52] apply augmentations to both real and generated samples and require the transformations to be differentiable in order to backpropagate to the generator with good results using as little as 10% of the available samples.

3.1.2 Consistency Regularization

Consistency Regularization(CR) is a semi-supervised training technique introduced to GANs by [50] as a discriminator regularization method that can be used in conjunction with gradient normalization methods. They have an additional discriminator loss term that encourages this network's predictions to be invariant under arbitrarily transformations applied to real samples. They perform an ablation study where they compare their technique with simply applying data augmentation and show that while the latter prevents the discriminator from over-fitting, it does not lead to an increase in generated image quality, and that CR actually improves the sample quality. They speculate this suggests CR helps the discriminator learn better representations of the data distribution.

[53] extend this work to balanced Consistency Regularization (bCR) and latent Consistency Regularization (zCR) and combine the two into Improved Consistency Regularization (ICR). The former augments CR by applying transformations to the generated samples and adding a regularizer loss term to the generator. zCR modifies CR by applying transformations only to the stochastic latent variable z and regularizing the discriminator to be invariant with respect to this augmentation while encouraging the generated images to be far away in sample space.

3.2 Multi Task Learning

Multi-task learning [51], closely related to meta-learning, is a field traditionally associated to methods that use parameter sharing or task relationships to improve performance of machine learning models. [38] surveys many of these parameter-sharing methods and discerns between soft and hard sharing. Parameter sharing can be formulated as a combinatorial optimization problem where we are interested in assigning multiple tasks to multiple possible components of larger deep models. In computer vision encoder networks are often shared while decoders are task-specific. While solving this type of problem exactly is known to be NP-hard, numerous approximate methods obtain good performance in practice. [43] use early stopping and extrapolate task-relation from pairwise assignment experiments to solve the combinatorial problem under a constrained compute budget. Many of these works consider tasks which have different objectives (e.g. segmentation, edge recognition) while in our case the objective is the same while the input and output samples come from different data distributions on the same domains.

[15] recently introduced the idea of a shared global work-space to deep learning by adapting the Transformer architecture [46]. They replace attention mechanisms with an attention-based two-step process : *i)* write to a shared workspace *ii)* read from shared workspace to inform next-layer's representation. One of the key characteristics of this process is that the write-read process introduces a computational bottleneck. [15] speculate this forces specialization in neural models that would help in multi-task learning. They show improvements in vision object tracking and relational reasoning tasks as well as a multi-agent world modeling problem. This development is of particular interest to us because recent work has also shown transformer architectures are able to obtain good performance in adversarial training scenarios [23].

3.3 Generative Modeling

We considered adversarial generative modeling because these models are currently state-of-the-art in most image generation benchmarks and we build upon prior work from the meteorology community. Recent work has shown a renewed interest in other generative approaches. [45] improve variational autoencoder approaches by discretizing the latent space and learning the prior rather than using a static normal distribution. [41] revisit learning the data distribution with score-matching [20] and perturb data samples with Gaussian noise in order to address two practical concerns: *i)* if the data resides in a lower-dimensional manifold scores are not properly defined everywhere; *ii)* low-density regions make both score estimation and sampling techniques perform poorly. At inference time they propose an annealed Langevin sampling technique and produce high quality samples. [18, 40] introduce probabilistic diffusion models, another latent variable model which imagines a Markov Chain process that sequentially adds noise to data samples until the original distribution is transformed into a simple prior and then learn to reverse this process.

The main advantages of these promising research directions are that training is more stable than in the adversarial setup and there are fewer parameters that require tuning to obtain good performance.

Chapter 4

Meta-Learning

4.1 Benchmark Construction

We leverage the SEVIR [47] dataset to construct a few-shot multi-task image-to-image translation problem where each task corresponds to one event. From the 49 available frames we keep the first N_{support} frames to form the task's support set and the next N_{query} to be the query. Throughout the following experiments we set $N_{\text{support}} = N_{\text{query}} = 10$.

For the sake of this discussion let's assume we have re-scaled all input modalities to the maximum observed resolution 384 × 384 so that we can view all of SEVIR as a simple input tensor $\mathcal{D}_1 \in \mathbb{R}^{N_{\text{event}} \times N_{\text{frames}} \times C \times w \times h}$, where: *i)* $N_{\text{event}} = 11479$; *ii)* $N_{\text{frames}} = N_{\text{support}} + N_{\text{query}}$; *iii)* $C = 4$; *iv)* $w = h = 384$. The four input channels are split into three input modalities $C_{\text{in}} = 3$ and one target $C_{\text{out}} = 1$. For joint training we ignore the hierarchical dataset structure and collapse the first two axis $\mathcal{D}_2 \in \mathbb{R}^{N \times C \times w \times h}$, where $N = N_{\text{event}} \times N_{\text{frames}}$ - the total number of frames.

4.2 Methods

We solve the aforementioned task using either first-order or second-order gradient descent methods on U-Nets trained using either reconstruction or adversarial objectives. Note that in the case when we train GANs using MAML we are searching for a good initialization for multiple related saddle-point problems and still obtain good performance.

Below we present the meta-train loop for adversarial networks, which is a novel contribution of our work. For simplicity, we only present the variant with a single SGD inner-loop adaptation step. We train a U-Net generator G with model weights w_G jointly with an extranous patch discriminator D with model weights w_D using data $\mathcal{D} \in \mathbb{R}^{N_{\text{event}} \times N_{\text{frames}} \times C \times w \times h}$. We use batched alternating gradient descent as our optimization algorithm and consider batches $\mathcal{B} \in \mathbb{R}^{B \times N_{\text{frames}} \times C \times w \times h}$, where B is the meta-batch size. Each of these can be split along the second axis into the support and query sets, and along the third axis into the source (S) and target tensors (T) to create $S^{\text{support}} \in \mathbb{R}^{B \times N_{\text{support}} \times C_{\text{in}} \times w \times h}$, $S^{\text{query}} \in \mathbb{R}^{B \times N_{\text{query}} \times C_{\text{in}} \times w \times h}$, $T^{\text{support}} \in \mathbb{R}^{B \times N_{\text{support}} \times C_{\text{out}} \times w \times h}$, $T^{\text{query}} \in \mathbb{R}^{B \times N_{\text{query}} \times C_{\text{out}} \times w \times h}$. For any of these tensors $X \in \{S^{\text{support}}, S^{\text{query}}, T^{\text{support}}, T^{\text{query}}\}$ we refer to the four-dimensional tensor given by the i^{th} task or event as X_i. We use such four-dimensional tensor quantities to evaluate the generator and discriminator loss functions:

$$\hat{\mathcal{L}}_G(t^{\text{generated}}, t, s; w_G, w_D) = -\log D(s, t^{\text{generated}}) + \lambda ||t^{\text{generated}} - t||_1 \tag{4.1}$$

$$\hat{\mathcal{L}}_D(t^{\text{generated}}, t, s; w_G, w_D) = \frac{1}{2}\left(\log D(s, t^{\text{generated}}) - \log D(s, t)\right), \tag{4.2}$$

where $t^{\text{generated}} = G(s)$ is a generated target sample, t and s are corresponding input and output modalities, $||x||_1$ is the mean absolute error. Note the slight abuse of notation where by $\log D(x, y)$ with $x, y \in \mathbb{R}^{N \times C \times w \times h}$ we mean the average $\frac{1}{N}\sum_{i=1}^{N} \log D(x_i, y_i)$. This formulation also uses the trick mentioned in 2.1.2 of replacing $\max \log(1 - D(G(z)))$ with $\min \log D(G(z))$ to obtain a non-saturating generator objective. We wrote the loss functions above such that both players want to minimize their respective objectives.

For each task in a meta-batch size we evaluate the losses above on the support set frames and adapt to this event using SGD to obtain parameters ϕ. We then evaluate the same losses on the task's query set using finetuned models. We repeat these two steps for each event in the meta-batch and perform a second-order gradient update to the initial parameters to optimize the average loss across all events in the meta-batch. This procedure is schematically summarized in Algorithm 1.

Algorithm 1: One Epoch MAML-Train Loop for U-Net Generator with Adversarial Loss.

```
for meta-train-batch $\mathcal{B} \in \mathbb{R}^{B \times N_{\text{frames}} \times C \times w \times h}$ do
    unpack $\mathcal{B} \in \mathbb{R}^{B \times N_{\text{frames}} \times C \times w \times h}$ along support/query, source/target into:
        $S^{\text{support}}, S^{\text{query}}, T^{\text{support}}, T^{\text{query}}$
    init $l_G^{\text{batch}} = 0$, $l_D^{\text{batch}} = 0$
    for each event $i$ out of $B$ in meta-batch do
        forward pass $T_i^{\text{support; generated}} = G(S_i^{\text{support}})$
        $l_G^{\text{adapt}} = \mathcal{L}_G(T_i^{\text{support; generated}}, T_i^{\text{support}}, S_i^{\text{support}}; w_G, w_D)$ from Eq. 4.1
        $l_D^{\text{adapt}} = \mathcal{L}_D(T_i^{\text{support; generated}}, T_i^{\text{support}}, S_i^{\text{support}}; w_G, w_D)$ from Eq. 4.2
        task-specific parameters $\phi_G \leftarrow w_G - \eta \nabla_{w_G} l_G^{\text{adapt}}$
        task-specific parameters $\phi_D \leftarrow w_D - \eta \nabla_{w_D} l_D^{\text{adapt}}$
        forward pass $T_i^{\text{query; generated}} = G(S_i^{\text{query}})$
        $l_G = \mathcal{L}_G(T_i^{\text{query; generated}}, T_i^{\text{query}}, S_i^{\text{query}}; \phi_G, \phi_D)$ from Eq. 4.1
        $l_D = \mathcal{L}_D(T_i^{\text{query; generated}}, T_i^{\text{query}}, S_i^{\text{query}}; \phi_G, \phi_D)$ from Eq. 4.2
        update rolling sums $l_G^{\text{batch}} += l_G$ and $l_D^{\text{batch}} += l_D$
    end
    backpropagate $2^{\text{nd}}$ order updates $\nabla_{w_G} l_G^{\text{batch}}$ and $\nabla_{w_D} l_D^{\text{batch}}$ to $w_G$ and $w_D$
end
return good initializations $w_G$ and $w_D$ for both generator and discriminator.
```

4.3 Experimental Details

We run experiments on MIT Supercloud [36] using a single 32GB Nvidia Volta V100 GPU. For MAML optimization [2] we use meta-batch sizes of $2-4$ events. For the corresponding joint training baselines we used $N_{\text{support}}+N_{\text{query}}$ frames from each event and comparable number of events to keep comparisons fair. We randomly split all SEVIR events into 9169 train, 1162 validation, and 1148 test tasks.

Joint training baselines and MAML outer loop optimizations are both performed using the Adam optimizer [24] with learning rate 0.0002 and momentum 0.5.

We resize input modalities to all have 192×192 resolution and keep the target at 384×384. The generator's encoder has four convolutional blocks, and the decoder has five. All generator blocks except for the last decoder layer use ReLU activation functions. The very last layer uses linear activation functions to support z-score normalization for all four image modalities.

4.4 Results

We test our multi-task few-shot formulation and demonstrate MAML provides empirical gains by comparing the performance of models trained using either meta-learning algorithms or joint training for both reconstruction and adversarial loss objectives. We find throughout all our experiments that meta-learning minimizes the reconstruction error compared to joint training. On the other hand, achieving better performance on the training objective does not always translate to higher weather evaluation metrics.

4.4.1 Hyper-parameter sweep

We first do a cursory hyper-parameter sweep over λ during adversarial training and η during MAML's inner-loop task adaptation. We experiment with $\lambda \in \{10^2, 10^3, 10^4\}$ and $\eta \in \{10^{-4}, 10^{-5}\}$.

Table 4.1: **Reconstruction MAML and Adversarial Joint - hyperparameter sweep.** Best observed MAE on validation split.

Reconstruction MAML		Adversarial Joint		
$\eta = 10^{-4}$	$\eta = 10^{-4}$	$\lambda = 10^2$	$\lambda = 10^3$	$\lambda = 10^4$
0.35	0.35	0.39	0.40	0.40

We summarize best MAE values on validation split for models trained adversarially with joint optimization or meta-learnt when optimizing the reconstruction loss in table 4.1. We show the same MAE's evolution throughout training in Fig. 4-1 and find marginal differences between different hyperparameter values.

We perform the same analysis for the more complex setting of meta-learning GANs and summarize performance in table 4.2 and figure 4-2. In this case we find training is more brittle and the hyperparameter choice has a higher impact on training dynamics. We find that different models converge at very different speeds: for example, when

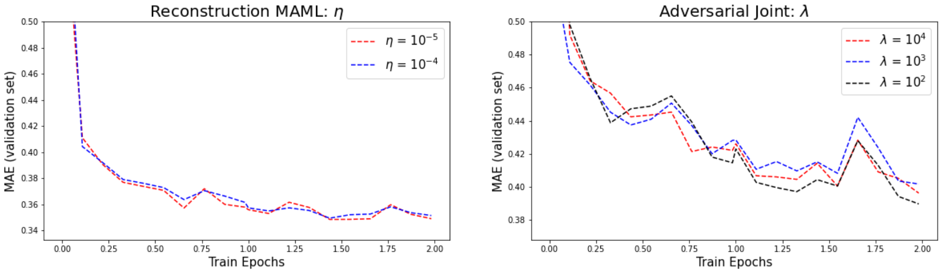

Figure 4-1: **Reconstruction MAML and Adversarial Joint - hyperparameter sweep.** Train curve plots performance on validation-split throughout training. Adversarial settings are slightly more sensitive to the λ hyperparameter than MAML is to η.

Table 4.2: **Adversarial MAML - hyperparameter sweep.** Best observed MAE on validation split.

	$\lambda = 10^2$	$\lambda = 10^3$	$\lambda = 10^4$
$\eta = 10^{-4}$	0.31	0.32	**0.29**
$\eta = 10^{-5}$	0.32	0.32	0.32

using an inner SGD learning rate $\eta = 10^{-4}$, models optimizing loss functions with $\lambda = 10^4$ converge much slower than those with $\lambda = 10^3$. Interestingly, the former actually achieves better performance, which suggests that stabilizing training dynamics is important in meta-learning adversarial networks.

4.4.2 Reconstruction Loss

Fig. 4-3 compares the performance of models trained using either traditional joint training and MAML-based optimization on a held-out validation set. We find that MAML consistently out-performs Joint-Training and is robust with respect to arbitrary hyper-parameters such as the number of inner-steps or the meta-batch size. We restrict our attention to the simplest MAML variant above, which uses meta-batches of two tasks and performs a single adaptation step for each event, and evaluate model performance using weather metrics. We summarize our results in Table 4.3 and find that even though U-Nets trained with MAML achieve better performance on the optimization objective, these improvements do not consistently translate to gains on

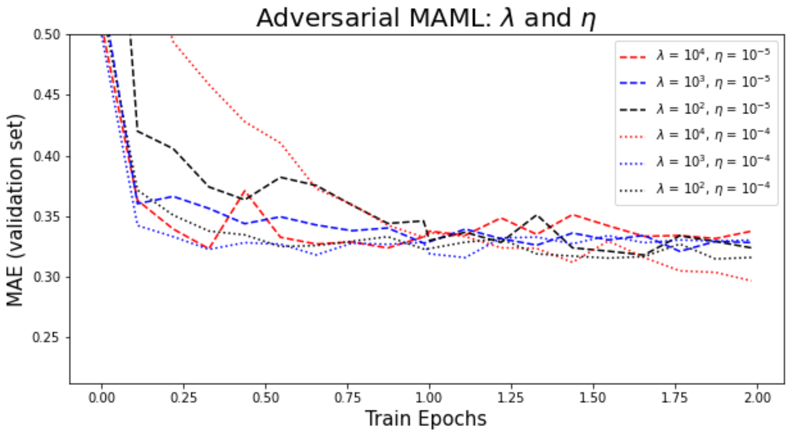

Figure 4-2: **Adversarial MAML - hyperparameter sweep.** Train curve plots performance on validation-split throughout training. The choice of λ and η significantly impact convergence speed and final train objective value.

weather-specific evaluation. In particular, we see that finetuning to specific tasks leads to better precision but worse recall and IOU.

Table 4.3: **Reconstruction loss - evaluation.** Comparison of joint-training and MAML-based optimization with a single inner-adaptation step. Test-set evaluation on meteorological metrics. MAML has better precision but worse recall and IOU.

threshold	74			133		
metric	CSI	POD	SUCR	CSI	POD	SUCR
Joint	**0.21**	**0.23**	0.81	**0.27**	**0.30**	0.79
MAML	0.14	0.14	**0.86**	0.20	0.20	**0.98**

Figure 4-4 shows sample images generated by U-Nets trained with reconstruction loss using either the baseline joint training method or MAML. Limitations given by training with reconstruction loss, such as blurry outputs, remain. The task adaptation mechanism helps in this case to recognize that there are some storm events in the bottom-left corner, although it is not very effective at predicting the correct shape of these low-intensity precipitations on a fine-grained scale.

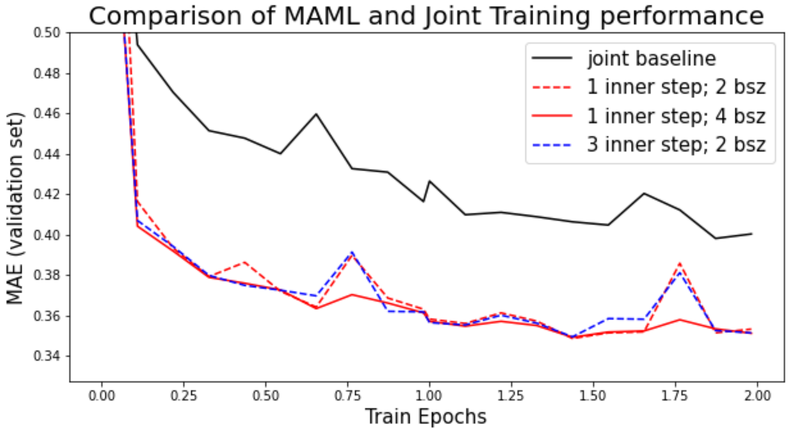

Figure 4-3: **Reconstruction loss - train curve.** MAML outperforms Joint Training. Evaluation done on validation test throughout training. The number of inner steps and meta-batch size do not significantly impact performance.

4.4.3 Adversarial Loss

We train generative adversarial networks using the second-order MAML procedure (on both the generator and discriminator networks, as described in Algorithm 1) and the joint training baseline. We compare the evolution of the reconstruction error throughout training in Fig. 4-5 and notice MAML significantly helps in minimizing the train objective. We used $\lambda = 10^2$ and $\eta = 10^{-4}$ for this MAML curve.

Because the study in Section 4.4.1 suggested both adversarial training methods are more susceptible to hyper-parameters choice, we evaluate on meteorological metrics for all values of λ and η, and summarize our results in Table 4.4 and 4.5 for joint and MAML training, respectively. For joint adversarial training, especially when evaluating with lower thresholds, we see the critical success index is fairly constant as we vary λ while increasing λ leads to lower recall and higher precision. This seems to suggest that placing more weight on the reconstruction loss will lead to predicting fewer high-valued VIL pixels.

For MAML adversarial training we do not identify any clear trend between hyperparameters λ and η and the values of meteorological metrics on the test-split. We believe this shows training is more unstable in this regime: instabilities are further exacerbated when optimizing a bilevel Nash equilibrium problem with gradient de-

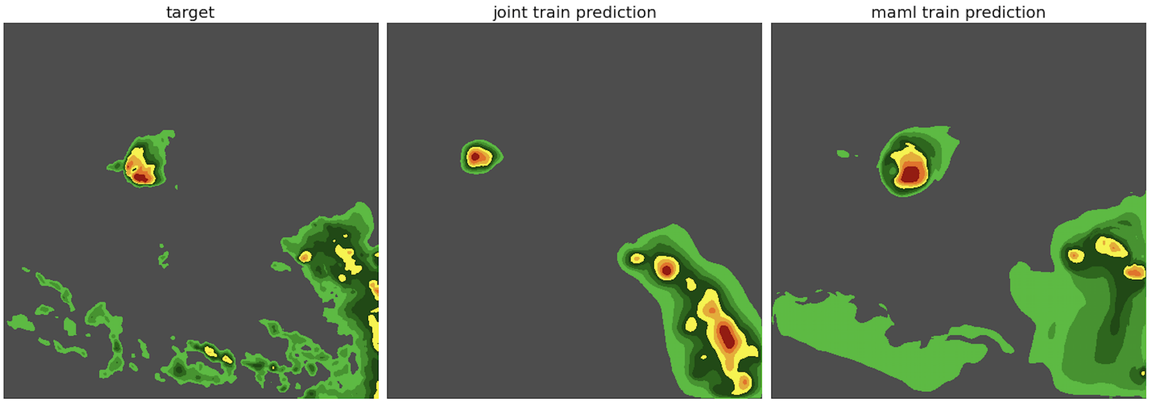

Figure 4-4: **Reconstruction loss - generated samples.** Comparison of joint-training and MAML-based optimization with a single inner-adaptation step. Sample generated VIL frames. Finetuning helps the model predict low-intensity regions. Reconstruction loss makes outputs somewhat blurry.

Table 4.4: **Adversarial Joint - evaluation.** Test-set evaluation on meteorological metrics.

threshold	74			133		
metric	CSI	POD	SUCR	CSI	POD	SUCR
$\lambda = 10^2$	0.29	0.50	0.56	0.27	0.30	0.76
$\lambda = 10^3$	0.29	0.46	0.58	0.29	0.35	0.71
$\lambda = 10^4$	0.29	0.43	0.64	0.29	0.33	0.73

scent, as we did above. A comparison between tables 4.4 and 4.5 shows that, similarly to the case of reconstruction loss, MAML optimization leads to higher precision and lower recall. After visually inspecting the generated samples we find that some of the models seem to exhibit mode collapse where the generated samples are not even realistic, while some of them do resemble the ground-truth. We present examples of samples successfully generated by models trained with MAML on adversarial loss below and note there is a large variance in the fraction of realistic samples across different models. This is not reflected in any of the evaluation metrics: we believe this further underscores that in image generation the correlation between good evaluation performance and high sample quality is rather weak.

Figures 4-6 and 4-7 compare samples generated by models trained on adversarial

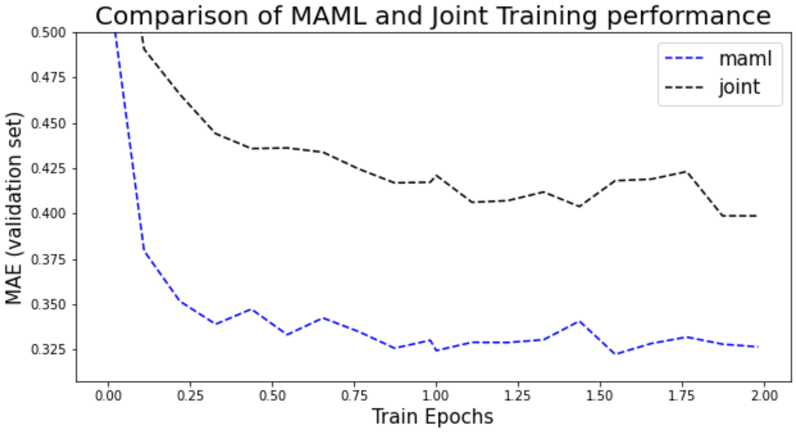

Figure 4-5: **Adversarial loss - train curve.** MAML outperforms Joint Training. Evaluation done on validation test throughout training.

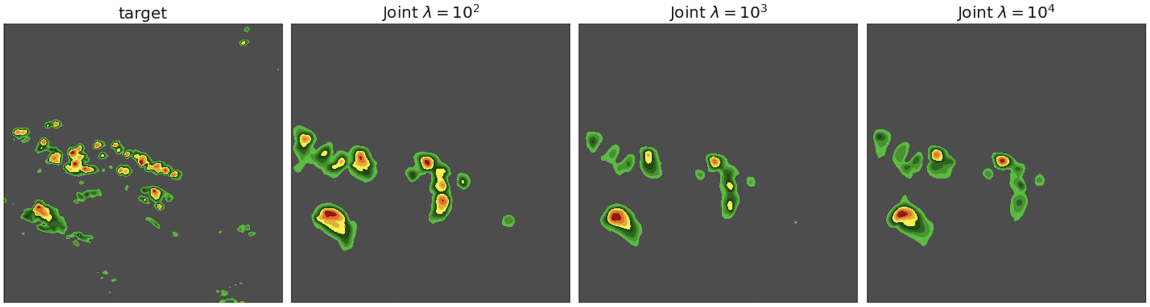

Figure 4-6: **Adversarial Joint - generated samples.** Reconstruction loss biases the model towards sparser predictions.

loss through either joint or MAML-based procedures for different values of λ. The MAML models all used an inner SGD learning rate of 10^{-5}. We see that in this case the intuitions from the reconstruction loss setting are still valid and the task-adaptation inherent to MAML enables it to correctly generate low-intensity VIL data that joint-setting misses out on. We also confirm the aforementioned trend of higher λ values leading to lower VIL values.

Table 4.5: **Adversarial MAML - evaluation.** Test-set evaluation on meteorological metrics. MAML models have higher precision and lower recall and IOU.

threshold		74			133		
metric		CSI	POD	SUCR	CSI	POD	SUCR
$\eta = 10^{-4}$	$\lambda = 10^2$	0.14	0.16	0.93	0.24	0.26	0.90
	$\lambda = 10^3$	0.09	0.09	0.98	0.20	0.20	0.99
	$\lambda = 10^4$	0.13	0.21	0.91	0.21	0.32	0.87
$\eta = 10^{-5}$	$\lambda = 10^2$	0.19	0.23	0.87	0.23	0.27	0.84
	$\lambda = 10^3$	0.17	0.20	0.90	0.25	0.29	0.87
	$\lambda = 10^4$	0.12	0.15	0.93	0.22	0.26	0.91

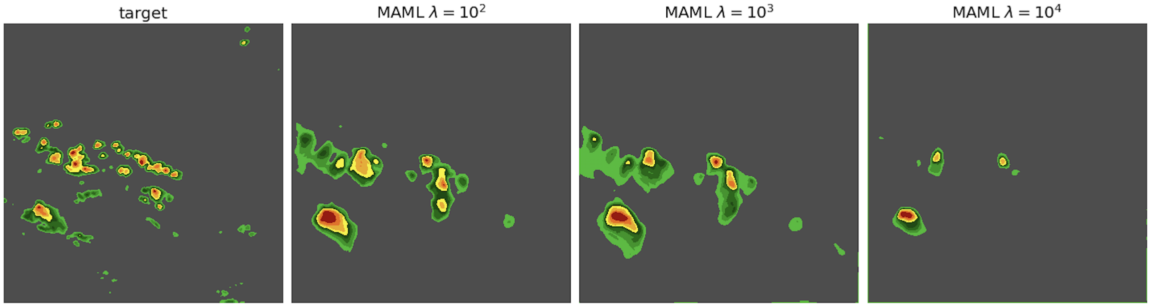

Figure 4-7: **Adversarial MAML - generated samples.** Finetuning helps identify low-intensity VIL regions.

Chapter 5

Self-Supervised Pretraining

5.1 Method

We follow recent work in self-supervised pretraining which applies contrastive learning to convolutional networks before finetuning on classification tasks and improves downstream performance and data efficiency. We ask if these improvements extrapolate to our image-to-image setup. The main distinction between our scenario and those in previous work is that we can initialize only a fraction of our parameters trough contrastive pretraining.

We restrict our attention to the U-Net encoder parameters during the pretraining stage and follow the same network architecture as in Chapter 4. We experiment with both the SimCLR[4] and SimSiam [6] pretraining methods and find the former performs better on downstream tasks.

Data Augmentations. Another difficulty particular to our setup is the problem of choosing data augmentations the input domain is invariant to because weather modalities have different invariances than natural images: for example, the popular color jitter transformation is not applicable here. From the standard augmentations, the only ones we consider are: random resized crops, random horizontal flips, gaussian noise and gaussian blur. We also further exploit the temporal structure of SEVIR to obtain “natural augmentations”.

5.1.1 Natural augmentations

We further consider using the temporal structure of SEVIR for augmentations, as follows. Each even consists of 49 frames, so we anchor frames $[0, 6, 12, 18, 24, 30, 36, 42]$. Then for each anchored frame, we sample an offset uniformly from the interval $[0, I)$, and add it to the index of the frame to obtain a positive pair of frames. We use the 8 positive pairs from 16 events to form a batch of 128 positive pairs. We apply the following transformations to each frame: the upper bound of the interval I for forming positive pairs of the anchors is either 0 or 3; optionally (either with probability 0 or 1) we apply random resized crops using scale (0.2, 1.0); either with probability 0 or 0.5 we apply diagonal gaussian noise with mean 0 and standard deviation 0.1; either with probability 0 or 0.2 we apply gaussian blur with standard deviation sampled from (0.1 and 2.0). The rest of the augmentation arguments follow the default in the Torchvision library[1]. In Figure 5-1 we present a conceptual visualization of the transforms. The base learning rate for contrastive pretraining is 0.015 and we consider a cosine decay scheduler from [6] with 4 warm-up epochs and 40 total epochs.

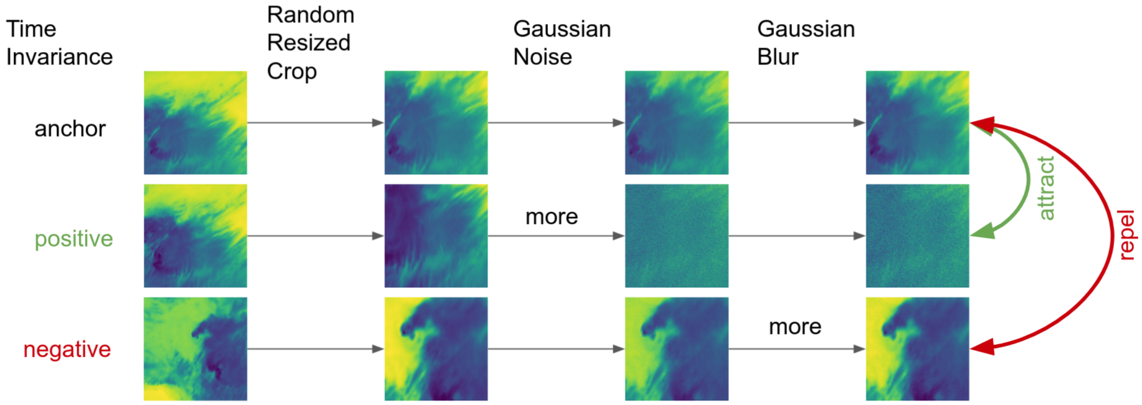

Figure 5-1: **Augmentations for the contrastive learning experiment** The positive pair for the anchor is 2 frames after the anchor. The negative pair is frame 42. Where we indicate "more" is an example of a larger magnitude of the augmentation being applied. We do not vary the magnitude in our experiments in this section, however we present it here as a possibility for future work.

[1] `https://pytorch.org/vision/stable/transforms.html`

Below we report results from preliminary checkpoints evaluated at early epoch stages. We track validation-split MAE throughout training in Fig. 5-2 using U-Net encoders initialized with SimCLR pretraining using either a single type of data augmentation or multiple transformations. In all settings we find marginal yet somewhat consistent gains. Interestingly, we see that pretraining with multiple augmentations leads to both slightly better performance, as well as more diverse training dynamics.

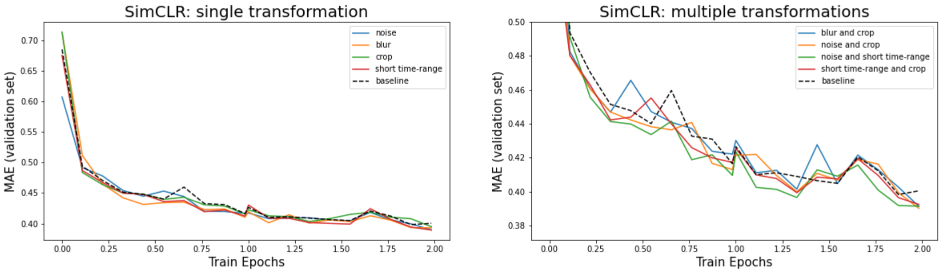

Figure 5-2: **Pretrained encoder - train curve.** Pretraining encoder parameters marginally improves reconstruction loss.

Table 5.1: **Contrastive Pretraining - evaluation.** Test-set evaluation on meteorological metrics. Pretraining significantly improves model precision. Using multiple augmentations leads to better performance.

Augmentation				74			133		
noise	blur	crop	short time	CSI	POD	SUCR	CSI	POD	SUCR
Y	N	N	N	**0.30**	0.48	0.58	0.27	0.29	0.79
N	Y	N	N	0.28	0.48	0.55	0.26	0.27	0.82
N	N	Y	N	0.27	0.37	0.67	0.27	0.30	0.81
N	N	N	Y	0.28	0.47	0.58	0.26	0.27	**0.84**
N	Y	Y	N	0.28	0.43	0.61	0.25	0.27	0.81
Y	N	Y	N	0.29	0.43	0.62	0.27	0.30	0.77
Y	N	N	Y	0.29	**0.51**	0.51	0.28	0.31	0.76
N	N	Y	Y	0.25	0.32	0.74	**0.29**	**0.34**	0.73
N	N	N	N	0.21	0.23	**0.81**	0.27	0.30	0.79

We evaluate on meteorological metrics and summarize our results in table 5.1. We find that even though pretraining had a marginal effect on the reconstruction loss train objective, it often provides important gains on domain-specific evaluation criteria. We highlight the large improvement in CSI for low threshold values, which stems mostly from significant improvements in precision.

We show example samples in figure 5-3 and find that pretraining the U-Net encoder leads to better performance in high-VIL regions.

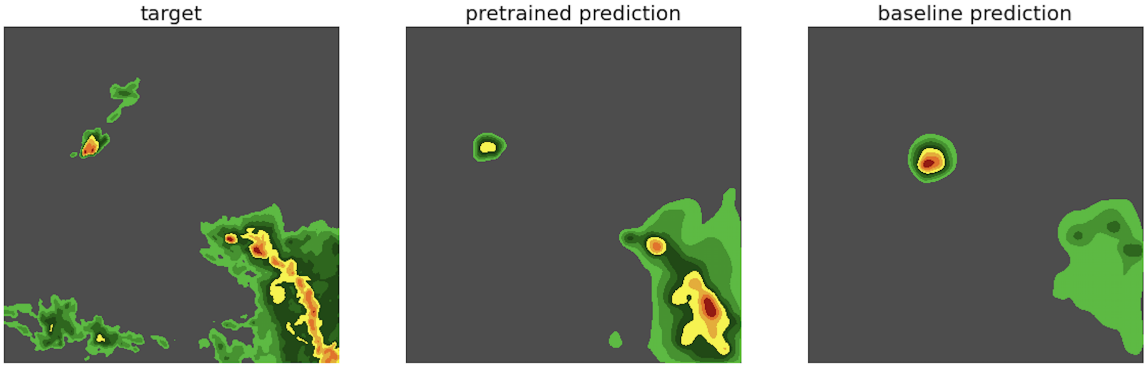

Figure 5-3: **Pretrained encoder - generated samples** Pretrained models better identify the sparse high VIL values.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

We formulate a novel few-shot multi-task image-to-image translation problem leveraging spatio-temporal structure in a large-scale storm event dataset. We provide several benchmarks for this problem and consider two optimization procedures (joint training and gradient-based meta-learning) and two loss functions (reconstruction and adversarial). We train U-Nets in all these regimes and present each model's performance, as well as evaluate on various domain-specific metrics. We discuss the advantages and disadvantages of each of these. In this process we have also explored a training scheduled unexplored until now to the best of our knowledge: meta-learning adversarial GANs with second-order gradient updates. Additionally, we explore pre-training U-Net encoder parameters using various augmentations in both the spatial and temporal domains.

6.2 Future Work

6.2.1 Improving performance and stability

There are numerous tricks for training GANs that have been shown to work well in practice for natural image generation. An interesting research direction would be exploring if these gains extend to our meteorological domain. Two of these techniques are applying spectral normalization to the discriminator network and updating the generator network more often than the discriminator.

We have not fully explored the interplay between adversarial training and MAML's bilevel optimization and believe it would also be very interesting to further develop this aspect of our work. The most immediate next step could be meta-learning just a subset of the networks' parameters.

Another interesting direction would be applying importance sampling or even curriculum learning techniques to the training schedule. An important difference between SEVIR and the natural images datasets we are more accustomed with is that not all events are equally informative: our models can presumably learn much more from complex storms than from frames taken during calm weather where the VIL and lighting frames are very sparse, and the IR imagery has very little variance.

6.2.2 Theoretical guarantees

[10] study both MAML and its first-order approximation in the standard single-agent scenario and prove both of these convergence to a stationary point. They additionally show convergence rates for the second order MAML method. [13] prove a universal theorem variant for second order MAML. In the future it would be interesting to analyse if these results extend to adversarial training and how to combine these formalisms with results that show adversarial training induces the optimal underlying sample distribution, follows universality guarantees, and converges in convex-concave problems when using algorithms such as extra-gradient or optimistic gradient descent.

We have also found that while algorithmic improvements help the optimization procedure reach optima that perform better on the objective function, these gains do not always translate to improvements of evaluation metrics of interest to the meteorology community, and believe it would be worthwhile to formulate loss functions more closely related to our end-goal of generating realistic VIL imagery.

Bibliography

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017.

[2] Sébastien M R Arnold, Praateek Mahajan, Debajyoti Datta, Ian Bunner, and Konstantinos Saitas Zarkias. learn2learn: A library for Meta-Learning research. August 2020.

[3] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[6] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[7] Louis Clouâtre and Marc Demers. FIGR: few-shot image generation with reptile. *CoRR*, abs/1901.02199, 2019.

[8] Constantinos Daskalakis, Andrew Ilyas, Vasilis Syrgkanis, and Haoyang Zeng. Training gans with optimism, 2018.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[10] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. On the convergence theory of gradient-based model-agnostic meta-learning algorithms, 2020.

[11] William Fedus, Mihaela Rosca, Balaji Lakshminarayanan, Andrew M. Dai, Shakir Mohamed, and Ian Goodfellow. Many paths to equilibrium: Gans do not need to decrease a divergence at every step, 2018.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[13] Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm, 2018.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[15] Anirudh Goyal, Aniket Didolkar, Alex Lamb, Kartikeya Badola, Nan Rosemary Ke, Nasim Rahaman, Jonathan Binas, Charles Blundell, Michael Mozer, and Yoshua Bengio. Coordination among neural modules through a shared global workspace, 2021.

[16] Erin Grant, Chelsea Finn, Sergey Levine, Trevor Darrell, and Thomas Griffiths. Recasting gradient-based meta-learning as hierarchical bayes. In *International Conference on Learning Representations*, 2018.

[17] Ishaan Gulrajani, Faruk Ahmed, Martín Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans. *CoRR*, abs/1704.00028, 2017.

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[19] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey, 2020.

[20] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

[21] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[22] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning, 2021.

[23] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two transformers can make one strong gan, 2021.

[24] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[25] Gregory R. Koch. Siamese neural networks for one-shot image recognition. 2015.

[26] Naveen Kodali, Jacob Abernethy, James Hays, and Zsolt Kira. On convergence and stability of gans, 2017.

[27] B. Lake, R. Salakhutdinov, and J. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350:1332 – 1338, 2015.

[28] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. The omniglot challenge: a 3-year progress report, 2019.

[29] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.

[30] Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[31] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

[32] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *CoRR*, abs/1802.05957, 2018.

[33] Aryan Mokhtari, Asuman Ozdaglar, and Sarath Pattathil. Convergence rate of $\mathcal{O}(1/k)$ for optimistic gradient and extra-gradient methods in smooth convex-concave saddle point problems, 2020.

[34] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms, 2018.

[35] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text, 2016.

[36] Albert Reuther, Jeremy Kepner, Chansup Byun, Siddharth Samsi, William Arcand, David Bestor, Bill Bergeron, Vijay Gadepally, Michael Houle, Matthew Hubbell, et al. Interactive supercomputing on 40,000 cores for machine learning and data analysis. In *2018 IEEE High Performance extreme Computing Conference (HPEC)*, pages 1–6. IEEE, 2018.

[37] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv e-prints*, page arXiv:1505.04597, May 2015.

[38] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

[39] Timothy J. Schmit, Paul Griffith, Mathew M. Gunshor, Jaime M. Daniels, Steven J. Goodman, and William J. Lebair. A closer look at the abi on the goes-r series. *Bulletin of the American Meteorological Society*, 98(4):681 – 698, 2017.

[40] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *CoRR*, abs/1503.03585, 2015.

[41] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020.

[42] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[43] Trevor Standley, Amir R. Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning?, 2020.

[44] Eleni Triantafillou, Tyler Zhu, Vincent Dumoulin, Pascal Lamblin, Kelvin Xu, Ross Goroshin, Carles Gelada, Kevin Swersky, Pierre-Antoine Manzagol, and Hugo Larochelle. Meta-dataset: A dataset of datasets for learning to learn from few examples. *CoRR*, abs/1903.03096, 2019.

[45] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning, 2018.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[47] Mark Veillette, Siddharth Samsi, and Chris Mattioli. Sevir : A storm event imagery dataset for deep learning applications in radar and satellite meteorology. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22009–22019. Curran Associates, Inc., 2020.

[48] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning, 2017.

[49] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. *CoRR*, abs/1711.11585, 2017.

[50] Han Zhang, Zizhao Zhang, Augustus Odena, and Honglak Lee. Consistency regularization for generative adversarial networks. *CoRR*, abs/1910.12027, 2019.

[51] Yu Zhang and Qiang Yang. A survey on multi-task learning, 2021.

[52] Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. Differentiable augmentation for data-efficient gan training, 2020.

[53] Zhengli Zhao, Sameer Singh, Honglak Lee, Zizhao Zhang, Augustus Odena, and Han Zhang. Improved consistency regularization for gans, 2020.

[54] Zhiming Zhou, Jiadong Liang, Yuxuan Song, Lantao Yu, Hongwei Wang, Weinan Zhang, Yong Yu, and Zhihua Zhang. Lipschitz generative adversarial nets, 2019.

[55] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv e-prints*, page arXiv:1703.10593, March 2017.