Mitigating Real-World Distribution Shifts in the Fourier Domain

Kiran Krishnamachari *kirank@u.nus.edu*
*Institute for Infocomm Research (I^2R), A*STAR, Singapore*
School of Computing, National University of Singapore, Singapore

See-Kiong Ng *seekiong@nus.edu.sg*
Institute of Data Science, National University of Singapore, Singapore
School of Computing, National University of Singapore, Singapore

Chuan-Sheng Foo *foo_chuan_sheng@i2r.a-star.edu.sg*
*Institute for Infocomm Research (I^2R), A*STAR, Singapore*
*Centre for Frontier AI Research (CFAR), A*STAR, Singapore*

Reviewed on OpenReview: *https://openreview.net/forum?id=lu4oAq55iK*

Abstract

While machine learning systems can be highly accurate in their training environments, their performance in real-world deployments can suffer significantly due to distribution shifts. Real-world distribution shifts involve various input distortions due to noise, weather, device and other variations. Many real-world distribution shifts are not represented in standard domain adaptation datasets and prior empirical work has shown that domain adaptation methods developed using these standard datasets may not generalize well to real-world distribution shifts. Furthermore, motivated by observations of the sensitivity of deep neural networks (DNN) to the spectral statistics of data, which can vary in real-world scenarios, we propose Fourier Moment Matching (FMM), a model-agnostic input transformation that matches the Fourier-amplitude statistics of source to target data using unlabeled samples. We demonstrate through extensive empirical evaluations across time-series, image classification and semantic segmentation tasks that FMM is effective both individually and when combined with a variety of existing methods to overcome real-world distribution shifts.

1 Introduction

DNNs are known to suffer significant drops in accuracy when deployed in real-world environments different from their training distribution. For example, systems trained to classify time-series signals of electroencephalogram (EEG) data undergo distribution shifts between hospitals as well as patients. Acoustic classifiers are susceptible to changes in recording device such as mobile phones as well as changes in background noise. Image classifiers encounter many corruptions in the wild including noise and weather changes that degrade performance. Standard domain adaptation datasets usually comprise tasks with extreme shifts between photos and synthetic target domains such as sketches or between digit datasets. Prior work has shown that domain adaptation methods developed using these datasets may not have the same level of performance in real-world scenarios, highlighting the need for methods effective on a variety of real-world shifts (Sagawa et al., 2022; Xie et al., 2021; Miller et al., 2021; Djolonga et al., 2021; Taori et al., 2020).

Real-world distribution shifts across modalities have often been analysed in the Fourier-domain. For example, prior work has analysed multiple common image corruptions in the Fourier-domain (Yin et al., 2019). Many time-series signals are also amenable to spectral analysis. In audio, frequency-domain analysis is common via the spectrogram. Relatedly, recent work has shown that DNN performance is highly sensitive to shifts

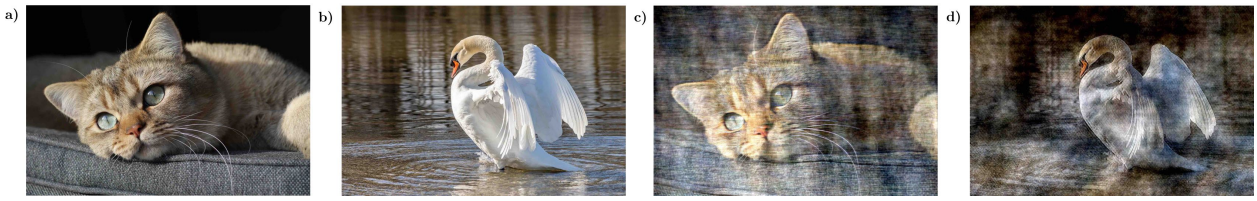

Figure 1: Importance of Fourier phase vs amplitude in natural images. a,b) Examples of natural images. c) Image which has the phase spectrum of image a, and the amplitude spectrum of image b. d) Image which has the phase spectrum of image b, and the amplitude spectrum of image a. Images c and d are perceptually similar to the image the phase spectrum was taken from, which indicates that the phase spectrum is more important for preserving image semantics.

in the Fourier-statistics of data (Jo & Bengio, 2017). Hence, in this paper, we approach the problem of adapting to real-world distribution shifts in the Fourier-domain.

A fundamental result in the Fourier analysis of natural images is that amplitude spectra follow a $1/f$ relationship, where f is frequency (Hyvärinen et al., 2009). This means that amplitude falls off inversely proportional to frequency. Hence, if all natural images have amplitude spectra that approximately have a $1/f$ shape, they cannot carry too much information about the signal or image and thus phase information is key to preserving image semantics. This view can be demonstrated by experiments in which we take the phase spectrum of one natural image and the amplitude spectrum of another image via their respective Fourier transforms. We then compute the inverse Fourier transform of this combination of phase and amplitude spectra to determine which image it turns out to "look" like. Figure 1 demonstrates such an experiment where we can see that the images turn out to resemble the image from which the phase spectrum was taken. Similar results holds for other types of signals e.g. audio [1] (Oppenheim & Lim, 1981). However, DNNs have been shown to be sensitive to spurious shifts in amplitude spectra statistics between train and test data. This motivates our approach to match amplitude statistics while leaving the phase information intact.

We thus propose an efficient model-agnostic input transformation, Fourier Moment Matching (FMM), that matches first and second-order Fourier-amplitude statistics of source data to those of the target data. Our method is also easily applied in different modalities, e.g. time-series and image datasets, using the Fast Fourier Transform (FFT) and does not require any changes to the model architecture. In summary, the main contributions of our work are as follows:

1. We propose a novel and model-agnostic input transformation, **Fourier Moment Matching (FMM)**, which matches the first and second-order Fourier-amplitude statistics of source to target domains using unlabeled data
2. We demonstrate through extensive empirical evaluations on real-world distribution shifts across time-series, image classification and semantic segmentation tasks that FMM is effective as a standalone method and can also improve the performance of existing domain adaptation methods

2 Background

2.1 Unsupervised Domain Adaptation

Unsupervised domain adaptation (UDA) is a commonly studied setting that assumes access to labeled samples from a source domain and unlabeled samples from a target domain where we wish the model to perform well. We consider the standard setting where source and target domains have the same label space $Y = \{1, 2, 3..., K\}$, K being the number of classes. In this work, we focus on real-world time-series and image-based tasks with distribution shifts, as these are relevant to the practical deployment of machine learning models.

[1] https://colab.research.google.com/drive/19FmN-HMIs6_6TLxJnHAqhbMI69nsOVww?usp=sharing

2.1.1 UDA Algorithms

UDA methods commonly perform distribution matching between domains or domain-invariant learning, generally in the embedding space of a deep neural network. CORAL (Sun et al., 2016) and DeepCORAL (Sun & Saenko, 2016) match second-order statistics between domains while HoMM (Chen et al., 2020) matches higher order moments. DDC (Tzeng et al., 2014), DAN (Long et al., 2015) and JAN (Long et al., 2017) utilize maximum mean discrepancy (MMD) to reduce the gap between source and target distributions. MMDA (Rahman et al.) combines CORAL and MMD with conditional entropy minimization, which increases the classifier's confidence on unlabeled data. DANN (Ganin et al., 2016), ADDA (Tzeng et al., 2017) and MCD (Saito et al., 2018) learn domain-invariant features using adversarial learning. CDAN (Long et al., 2018) uses class-conditional adversarial learning to learn aligned features between domains. DIRT-T (Shu et al., 2018) uses virtual adversarial training, a teacher model as well as conditional entropy. MCC (Jin et al., 2020) also uses entropy-regularization to encourage the model to be confident on unlabeled data, but with an instance dependent weight that discourages over-confidence on samples that are likely to be misclassified. MCC further minimizes instance-weighted confusion between classes. Margin Disparity Discrepancy (MDD) (Zhang et al., 2019) is a feature-based domain adaptation method which finds a representation of the input features that minimises the disparity discrepancy (DD) between source and target domains. CoDATS (Wilson et al., 2020), a method for time-series data, uses adversarial training with weak supervision. AdvSKM (Liu & Xue, 2021) uses adversarial spectral kernel matching for non-stationary time-series data. We further consider the following self-training or self-supervision methods benchmarked in the WILDS (Sagawa et al., 2022) framework for real-world distribution shifts: Pseudo-Label (Lee, 2013), FixMatch (Sohn et al., 2020), Noisy Student (Xie et al., 2020) and SwAV (Caron et al., 2020). While many existing methods have been effective on standard UDA datasets, they have been found to be less effective on real-world distribution shifts (Sagawa et al., 2022). Hence, it is highlighted in Sagawa et al. (2022) that new methods specifically designed for real-world distribution shifts are needed.

2.2 Approaches in the Fourier-domain

The Fourier-transform produces an amplitude and a phase spectrum, which together completely represent any signal. The amplitude and phase spectra contain different information about the signal (Oppenheim & Lim, 1981). Generally, changes in the amplitude spectrum alters the signal but does not affect its interpretation while altering the phase spectrum affects the interpretation of the signal. Hence, we propose a phase-preserving transformation for domain adaptation to real-world shifts. Relatedly, many works have used frequency analysis to analyse the generalization performance of models (Zhang, 2019; Vasconcelos et al., 2021; Krishnamachari et al., 2022; Tsuzuku & Sato, 2019; Yin et al., 2019). DNNs have been shown to rely on the frequency statistics of the dataset they are trained on. When tested on data with shifted Fourier-statistics, DNNs suffer significant performance degradation (Jo & Bengio, 2017). Shifts in frequency statistics are common due to various reasons such as noise, distortions and other real-world variations between domains, and are a contributing factor to poor generalization of computer vision models (Yin et al., 2019). A related work in domain adaptive semantic segmentation is Fourier Domain Adaptation (FDA) (Yang & Soatto, 2020), which proposed a transformation that replaces low-frequency amplitudes in source images with those of a randomly chosen target image. FDA also employs entropy minimization and self-training. In this work, we consider *statistics* of all frequencies in the Fourier-domain instead of swapping low-frequencies.

3 Fourier Moment Matching

We now describe Fourier Moment Matching, a domain adaptation method operating in the Fourier domain using unlabeled source and target domain samples. Let $D_s = \{x_s^i\}$ be the set of labelled source domain samples with labels $L_s = \{y_s^i\}$, and $y_s^i \in \{1, 2, ..., C\}$. Let $D_t = \{x_t^i\}$ be unlabelled target domain samples with $x_s^i, x_t^i \in \mathbb{R}^D$. Let $\mathcal{F}$ be the Discrete Fourier Transform (DFT), and $\mathcal{F}^{-1}$ be the inverse DFT. The output of $\mathcal{F}(x)$ is complex-valued and has the same dimensionality of x. For example, if $x \in \mathbb{R}^D$, $\mathcal{F}(x) \in \mathbb{C}^D$.

Let A_s^i, A_t^i be Fourier-amplitudes of source and target samples, respectively, i.e., $A_s^i = ||\mathcal{F}(x_s^i))||$ and $A_t^i = ||\mathcal{F}(x_t^i))||$. Let μ_s, μ_t be mean vectors of A_s^i and A_t^i i.e., $\mu_s = \mathbb{E}[A_s^i]$, $\mu_t = \mathbb{E}[A_t^i]$. C_s, C_t are the sample covariance matrices of A_s and A_t, respectively where $C_s, C_t \in \mathbb{R}^{D \times D}$. Let N denote the sample size of the

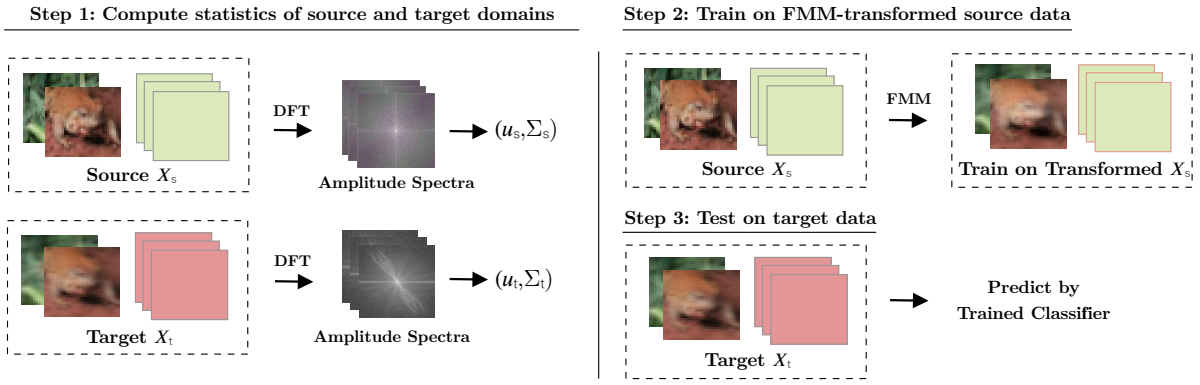

Figure 2: Overview of Fourier Moment Matching to mitigate real-world distribution shifts.

source domain samples and M denote the sample size of target domain samples. FMM matches moments of the source Fourier-amplitude distribution to those of the target domain using a whitening-dewhitening procedure. First, moments of the Fourier-amplitude distributions of unlabeled source and target data are computed individually. These statistics are then used to whiten and de-whiten the source data to match target statistics (Algorithm 1). FMM operates in the input space and does not impose any restrictions on model architecture. We propose two variants of FMM that match either the first-order moment or both first and second-order moments. We describe the two variants of FMM below. The overall FMM algorithm is visualized in Figure 2. We note that although the transformation of source images (Step 2 of Algorithm 1) can be performed *offline* before training, the FMM transformation is efficient enough to be computed *online* as part of the data-augmentation pipeline. Both online and offline methods lead to the same results and can be chosen based on the computational complexity associated with the dataset at hand. We used the online implementation in our experiments below.

3.1 First-Order FMM

The first-order moments of the amplitude distributions of source and target data are μ_s and μ_t, respectively. First-order FMM transforms the mean of the source amplitudes to that of the target domain, i.e. $FMM(A_s^i) = A_s^i - \mu_s + \mu_t$.

3.2 Second-Order FMM

Second-order FMM matches both the first and second-order moments of the amplitude distributions of source and target data. The mean and covariance of the source amplitudes are transformed to match that of the target domain, i.e. $FMM(A_s^i) = (A_s^i - \mu_s) \times C_s^{-1/2} \times C_t^{1/2} + \mu_t$.

3.2.1 Estimating high-dimensional covariance matrices

While *sample covariance* is an unbiased estimator of the population covariance, in the low sample support scenario, i.e., when the input dimensionality D is large compared to the sample size n, estimation error can be high especially when dealing with arbitrary non-Gaussian distributions. Moreover, the resultant sample covariance matrix is singular (non-invertible) since it has rank at most n (when $D > n$), even though the true covariance matrix is invertible. A simple solution is adding the identity matrix scaled by a small value, we used 1×10^{-3}), to the estimated sample covariance matrix. Moreover, second-order FMM requires computing the inverse and matrix square roots of covariance matrices, which are $D \times D$ matrices. The space and time complexity of these operations can grow as $\mathcal{O}(D^3)$, which can be infeasible for high-resolution input on standard machines, e.g. ImageNet-size images. Hence, for high-dimensional input, we use block-diagonal approximations (Appendix Figure 7) of the covariance matrices. The size of the block sub-matrices, b, along the diagonal must be chosen based on available computational capacity and the sample size. A block diagonal

Algorithm 1 Fourier Moment Matching

Input: $D_s = \{x_s^i, y_s^i\}$ are labeled source domain training samples with $i \in \{1, 2, ..., N\}$. $D_t = \{x_t^j\}$ are unlabeled target domain samples with $j \in \{1, 2, ..., M\}$. A model f with initial parameters θ.

Step 1: Compute Fourier-statistics for source and target data (μ_s, C_s, μ_t, C_t)

$\mu_s = \frac{1}{N} \sum_{i=1}^{N} ||\mathcal{F}(x_s^i)||$ and $\mu_t = \frac{1}{M} \sum_{j=1}^{M} ||\mathcal{F}(x_t^j)||$

$C_s = \frac{1}{N-1} \sum_{i=1}^{N} (||\mathcal{F}(x_s^i)|| - \mu_s)(||\mathcal{F}(x_s^i)|| - \mu_s)^T$ and $C_t = \frac{1}{M-1} \sum_{i=1}^{M} (||\mathcal{F}(x_t^j)|| - \mu_t)(||\mathcal{F}(x_t^j)|| - \mu_t)^T$

Step 2: Transform source data to match statistics of target data in Fourier domain

for $i = 1$ **to** N **do**
  $A_s^i = ||\mathcal{F}(x_s^i)||$ {compute DFT-amplitudes}
  $P_s^i = phase(\mathcal{F}(x_s^i))$ {compute DFT-phase}
  if FMM: 1st Order **then**
    $FMM(A_s^i) = A_s^i - \mu_s + \mu_t$
  else if FMM: 2nd Order **then**
    $FMM(A_s^i) = (A_s^i - \mu_s) \times C_s^{-1/2} \times C_t^{1/2} + \mu_t$
  end if
  $x_s^i = \mathcal{F}^{-1}\ (FMM(A_s^i),\ P_s^i)$ {inverse-DFT}
  Store FMM transformed source data x_s^i
end for

Step 3: Standard training on FMM transformed source data

T = training iterations
for $j = 1$ **to** T **do**
  Sample K labeled and FMM-transformed source domain images
  for $i = 1$ **to** K **do**
    Compute classifier loss $\mathcal{L}(x_s^i, y_s^i)$
  end for
  Update classifier $f(\cdot; \theta)$ to minimize loss
end for

Step 4: Standard evaluation on target domain data

for $j = 1$ **to** M **do**
  predict $f(x_t^j; \theta)$
end for

approximation significantly reduces the computational costs of inverting and square-rooting the covariance matrix, which are needed to perform second-order FMM. Computing the inverse or square-root of block-diagonal matrices simplifies to operating on the individual block sub-matrices (Appendix Figure 7b), which significantly reduces memory and computational requirements. In case of very low available computational capacity or sample size relative to the input dimensionality, a diagonal covariance matrix (equivalent to b=1) can be used that excludes off-diagonal covariance terms.

4 Experiments

We first describe experimental settings for benchmarks below (Section 4.1). We then validate that FMM is more effective than existing methods in overcoming shifts in the Fourier-domain using an artificial Fourier-shift (Section 4.3). We then perform unsupervised domain adaptation experiments using classification tasks

containing real-world distribution shifts across time-series (Section 4.4.1), image classification (Section 4.5) and semantic segmentation (Section 4.6). In all experiments, ERM (empirical risk minimization) refers to the non-adapted baseline model trained on labeled source data. "ERM + FMM" (or just "FMM") refer to standard training on source data that have been transformed using FMM. We also combined FMM with existing domain adaptation methods. For example, "DANN + FMM" refers to training using the DANN (domain adversarial neural network) method on source domain samples that have been transformed using FMM. We also include results for "Oracle" models on each task. The "Oracle" model refers to a supervised classifier trained on labeled *target* domain data and uses the same architecture as the domain adaptation methods. It is included only to provide an estimate of the upper-bound performance achievable by domain adaptation methods. As an additional baseline, we benchmarked moment matching in input-space, i.e., matching statistics of source to target data in the pixel-domain for images or raw waveform of time-series and audio data. We observed that FMM could outperform input-space moment matching significantly, demonstrating the advantage of matching statistics in the Fourier-domain (results in Appendix B).

4.1 Experimental settings

4.1.1 Time-series classification

a. Sleep-stage classification: We benchmarked methods on a real-world sleep-stage classification classification task using electroencephalography (EEG) data. We adopted the Sleep-EDF dataset (Goldberger et al., 2000), which contains EEG readings from 20 healthy subjects. The source and target domains in this task refer to EEG readings from different subjects in the Sleep-EDF dataset. Studies have shown that EEG signals vary across subjects due to factors such as subject-fatigue, difference in electrode placement, impedance etc. (Buttfield et al., 2006; Li et al., 2010). This significantly hampers the accuracy of automatic sleep-stage classifiers. Hence, the problem setting is to train on labeled EEG readings of one subject (source domain) and evaluating on EEG readings from another subject (target domain). We selected a single channel (i.e., Fpz-Cz), and 10 different subjects to construct five cross-domain (cross-subject) scenarios as proposed in (Ragab et al., 2023). We used the widely used 1D-CNN as the backbone network for all methods (Ragab et al., 2023) (see Appendix C.2 for details). All methods were trained for 40 epochs with a batch-size of 128. The Adam optimizer with fixed weight-decay (1e-4) and $(\beta_1, \beta_2) = (0.5, 0.99)$ was used to train all models. For each method, learning rate and other hyper-parameters were chosen using an extensive random search including 100 hyperparameter combinations per method and a target validation set (see Appendix C.1 for details). Input samples were 3,000 time steps in length and were consistently pre-processed across methods for fair evaluation. FMM was implemented using the 1-dimensional Fast Fourier Transform (FFT) of the input. When adding FMM to other methods, we added the FMM transformation to their input pre-processing pipelines. We used the AdaTime library for training and evaluation (Ragab et al., 2023); we report results averaged across three random seeds.

b. Acoustic scene classification: We benchmarked UDA methods on real-world distribution shifts between recording devices for an acoustic scene classification task. We used the TAU Urban Audio (Heittola et al., 2020b) dataset as provided in the development set of (Heittola et al., 2020a). The development set contains data from 10 cities and 9 devices: 3 real devices (A, B, C) and 3 simulated devices (S1-S3). The main recording device comprises a Soundman OKM II Klassik/studio A3, electret binaural microphone and a Zoom F8 audio recorder, referred to as device A. The other devices are commonly available customer devices: device B is a Samsung Galaxy S7, device C is iPhone SE. The devices were used to record audio at 10 different acoustic scenes, e.g. airport, park, street-traffic (see Appendix D for details). We used 10 hours of labeled training data from device A as the source domain, while the smaller datasets of the other devices were used as target domains. The audio is provided at 44.1kHz and each sample is 10 seconds long (each input is an array of length 441,000). As is standard, the input was converted to a MelSpectrogram (see Appendix D.1 for details) and fed to a ResNet18 model. FMM was performed on the raw audio input using the 1-dimensional FFT. We report results averaged across all source-target pairs and three runs with different random seeds. We used the same method specific hyper-parameters from the sleep-stage classification task.

4.1.2 Image classification

We benchmarked methods on four image classification tasks involving real-world distribution shifts. We evaluated methods on unsupervised domain adaptation from clean (source) to corrupted (target) images in (CIFAR10→CIFAR10-C and ImageNet→ImageNet-C (Hendrycks & Dietterich, 2019)). We randomly sub-sampled 50 classes in ImageNet-C to create ImageNet50-C in order to reduce computational costs (see Appendix E.1 for details). We used the *Transfer Learning Library* (Junguang Jiang et al., 2020) for benchmarking methods. On CIFAR10, we trained all models for 150 epochs using an initial learning rate (lr) that produced the best target domain validation set performance, selected from {0.1,0.01,0.001}, and lr decayed by a factor of 0.1 every 50 epochs. On ImageNet50, we trained all models for 90 epochs with an initial learning rate that produced the best target domain validation set performance, selected from {0.1,0.01,0.001}, and lr decayed by a factor of 0.1 every 30 epochs). For each method, hyper-parameters were selected on one task and applied to other tasks, requiring the hyper-parameters to generalize across tasks (see Appendix Table 12). This selection strategy is practical and widely adopted by many competitions. We performed the FMM algorithm on the input using 3D-FFT across the three color channels. When combining FMM with other methods, we additionally fine-tuned lr and the trade-off parameters for methods. We used the ResNet50 architecture for training on CIFAR10 and ImageNet50. For benchmarking on iWildCam-WILDS (Beery et al., 2020; Sagawa et al., 2022) we used the *Transfer Learning Library* to train DenseNet121 architectures initialized with ImageNet-pretrained weights and finetuned for 18 epochs. Hyper-parameters including learning rate and trade-off parameters that produced the best target risk were chosen per method (see Appendix Table 13). For benchmarking on Camelyon17-WILDS (Bandi et al., 2018), we used the *WILDS* (Sagawa et al., 2022) library and the provided commands[2]. Following the protocol in the WILDS framework, we used the model with the best validation domain performance, averaged across ten runs with different random seeds. We used the default model architecture and hyper-parameters for each method as used in WILDS.

4.1.3 Semantic segmentation

For semantic segmentation, we used the *Transfer Learning Library* library to train models using ERM, FDA (Yang & Soatto, 2020) and AdvENT (Vu et al., 2019). We used the DeepLabV2-ResNet101 architecture and hyper-parameters found for each method using a target domain validation set (see Appendix Table 14). For training HRDA (Hoyer et al., 2022), we used its official GitHub repository (`https://github.com/lhoyer/hrda`) using the default architecture, hyper-parameters, and averaged results across three runs as done in the HRDA paper.

4.2 Selection of FMM mode

Applying FMM to a task requires selecting the statistics that are matched, i.e. first-order vs second-order FMM. In second-order FMM, we must also select an approximation of the covariance matrix, e.g. diagonal, block-diagonal or the full-covariance matrix. While both the sample mean and covariance are consistent estimators of the true parameters, computing statistics using limited datasets in practice introduces estimation errors. Hence, the FMM mode for a task is treated as a hyper-parameter chosen using the validation set used to choose hyper-parameters of other UDA methods as well. On the high-dimensional ImageNet50-C dataset (D=150,528; N=65,000; M=2,500), it is not possible to perform operations such as inversion on the full covariance matrix due to memory restrictions on standard machines, necessitating the use of block-diagonal approximations. Moreover, we observed that using large block-sizes (b) at small sample sizes (N<20,000) can introduce significant estimation error that deteriorates FMM performance (Figure 3). Hence, the benefit of matching more statistics can be offset by estimation error at small sample sizes. On the other hand, first-order FMM and smaller block-sizes (b) were robust at small sample sizes (we report results using b=50 below). When combining FMM with other methods on ImageNet50-C, we used second-order FMM with a diagonal covariance approximation (b=1). On the lower dimensional CIFAR10-C dataset (D=3,072; N=50,000), second-order FMM using the full-covariance matrix achieved the best accuracy generally (see Appendix Table 15). On the small Sleep-EDG (EEG) dataset (D=3,000; N=14,000; M=6,000), first-order FMM was chosen and on the acoustic scene classification task (D=441,000; N=3,600), we chose second-order

[2] `https://worksheets.codalab.org/worksheets/0xb148346a5e4f4ce9b7cfc35c6dcedd63`

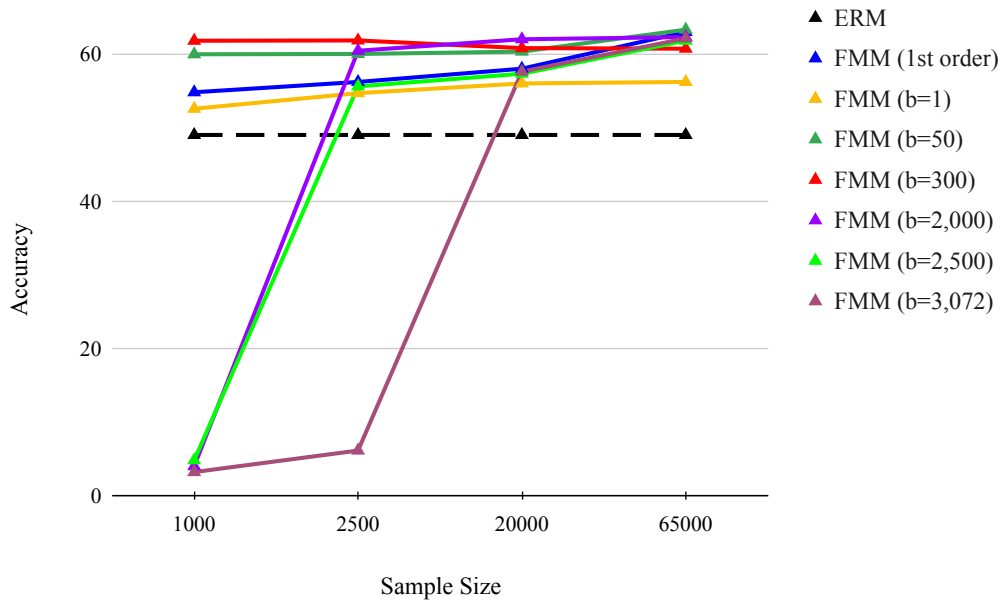

Figure 3: Effect of sample size on different modes of FMM for adaptation to ImageNet50-C (Gaussian Noise). The input-dimensionality ($D = 150,528$) here is large compared to the sample size. b is the block-size in the block-diagonal covariance matrix approximation.

Target	Oracle	ERM	FMM	MCC	MCD	ADDA	CDAN	DANN
$r = 11$	92.2	73.1	**91.9**	58.0	77.3	83.4	84.5	84.4
$r = 7$	88.2	17.6	**87.5**	44.7	63.2	74.5	75.5	75.8
$r = 5$	82.4	14.9	**81.4**	41.1	54.1	64.3	63.3	64.2

Table 1: Domain adaptation performance (accuracy) from clean to Fourier-filtered CIFAR10. Smaller r corresponds to more filtering.

FMM with a diagonal covariance (b=1) approximation. On the high-dimensional iWildCam-WILDs dataset (D=602,112), we used second-order FMM with b=50 for standalone FMM and b=1 when combining FMM with other UDA methods. On the Camelyon17-WILDS dataset, we again used second-order FMM with b=20 for standalone FMM and b=1 in combination with other UDA methods. On the semantic segmentation tasks, where the images have very high-resolution (D=3,145,728) we used second-order FMM with a diagonal covariance approximation (b=1).

A rule of thumb in practice is to first apply first-order FMM, which is expected to contain the least statistical estimation error, as a first approximation. Further optimisation can be then done by using larger block-diagonal approximations of the covariance matrix depending on the available sample size. Another factor that may affect FMM's performance is the *Fourier-sensitivity* (Krishnamachari et al., 2022) of models, which shows that model performance is more sensitive to some frequencies than others. Hence, matching the statistics of frequencies that a model most relies on may affect its performance more than other frequencies. We leave the study of the performance of FMM in relation to the *Fourier-sensitivity* of models for future work.

4.3 Validating FMM on artificial Fourier-shifts

DNNs rely on superficial Fourier-statistics of their training datasets (Jo & Bengio, 2017), which motivates our approach to match frequency statistics using FMM. To validate this, we generated visually similar Fourier-filtered CIFAR10 test images using filtering in frequency space to artificially shift the Fourier-statistics of

images (see Appendix A for examples and details). This operation modified the frequency statistics of the test samples compared to the train samples by setting high-frequencies to zero. A model trained on data with non-zero high-frequencies significantly degraded in performance suffering up to ∼60% absolute drop in accuracy when evaluated on the filtered images (Table 1). This demonstrates that mismatched frequency statistics between training and testing degrades performance. When we applied FMM (second-order), the model was able to adapt to the shifted unlabeled data more effectively than other UDA methods we evaluated and nearly matched Oracle performance. This demonstrates that other UDA methods are not able to completely overcome the distribution shift in the Fourier-domain as effectively.

4.4 Benchmarking on time-series classification

We benchmarked methods on two time-series classification tasks with real distribution shifts.

4.4.1 Sleep-stage classification

We benchmarked methods on the Sleep-EDF dataset (Goldberger et al., 2000) to categorize EEG signals into five stages i.e. Wake (W), Non-Rapid Eye Movement stages (N1, N2, N3), and Rapid Eye Movement (REM). We benchmarked canonical UDA methods that can be applied to time-series tasks, as proposed in (Ragab et al., 2023). Standalone FMM, i.e., ERM + FMM, performed better than all other UDA methods on this task (Table 2). Moreover, when combined with all other methods evaluated, FMM significantly improved their performance. The best performance across methods was achieved by the combination of MMDA and FMM with an accuracy of 77.0% (see Appendix Table 10 for results on each pair of source and target domains). We include results with mean and standard deviation across runs in Appendix J.

4.4.2 Acoustic scene classification

For acoustic scene classification, we used the popular TAU Urban Audio dataset (Heittola et al., 2020b). FMM outperformed other adaptation methods when evaluated as a standalone method, i.e. ERM + FMM. FMM also improved other methods when added to the input as pre-processing (Table 2). The best performance was achieved by the combination of CDAN, an adversarial learning method, and FMM (see Appendix Table 11 for results on each pair of source-target domains). We include results with mean and standard deviation across runs in Appendix J

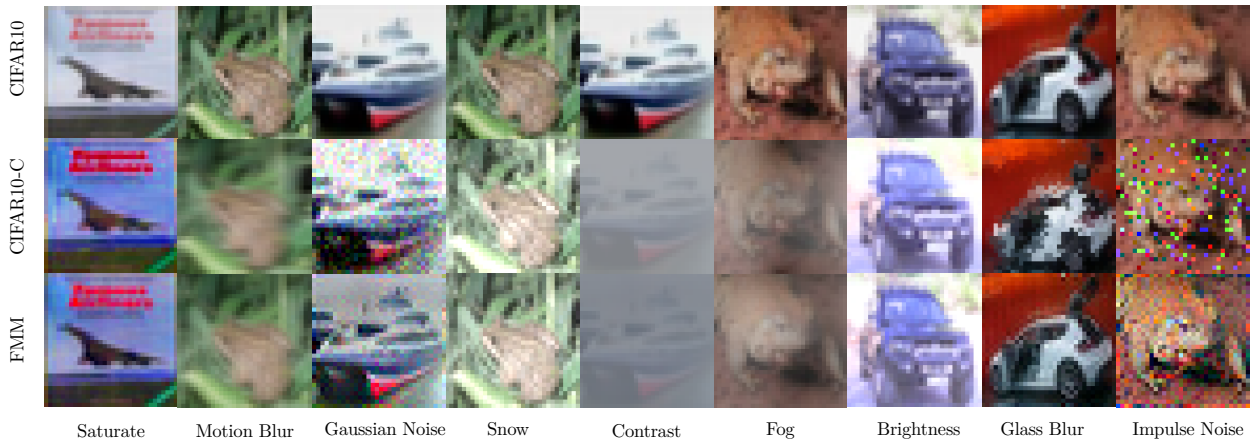

Figure 4: FMM bridges the visual gap between source (CIFAR10) and target (CIFAR10-C) images. CIFAR10 (top-row), CIFAR10-C (middle-row) and corresponding FMM transformed images (bottom-row).

Method	Sleep-EDF				TAU Audio			
	Standalone		Method + FMM		Standalone		Method + FMM	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
ERM	64.5	54.4	**74.1**	**61.7**	24.2	19.1	**30.3**	**27.8**
DANN	70.3	59.2	**72.5**	**61.0**	26.1	25.1	**36.4**	**35.3**
DeepCORAL	69.3	57.7	**74.5**	**63.0**	26.7	24.0	**33.0**	**31.8**
MMDA	72.3	61.4	**77.0***	**65.3***	22.2	20.3	**29.8**	**29.6**
DIRT-T	68.4	58.1	**73.5**	**62.1**	28.5	24.6	**34.4**	**33.0**
CDAN	71.2	59.6	**76.1**	**63.3**	29.9	28.4	**38.9***	**37.9***
HoMM	70.1	58.3	**74.3**	**62.4**	28.7	27.1	**34.6**	**33.4**
CoDATS	67.0	58.5	**72.0**	**60.2**	27.3	25.4	**33.1**	**31.1**
DDC	69.4	57.8	**74.5**	**62.9**	27.1	24.5	**34.1**	**32.8**
AdvSKM	74.1	62.4	**74.7**	**62.8**	28.6	25.6	**33.0**	**31.9**
Oracle	87.5	77.6	87.5	77.6	45.2	44.8	45.2	44.8

Table 2: Performance on Sleep-EDF (EEG) and TAU (audio) datasets. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods. Results were averaged across all source-target domain pairs and three runs with different random seed.

Target	Oracle	ERM	+ FMM	MCC	+ FMM	MCD	+ FMM	ADDA	+ FMM	CDAN	+ FMM	DANN	+ FMM
Contrast	92.3	56.3	**92.8***	19.0	**72.8**	43.9	**66.6**	28.4	**76.0**	29.6	**76.7**	27.3	**76.9**
Elastic Transform	90.0	73.2	**76.1***	68.0	**72.1**	70.5	**66.5**	72.3	**74.8**	73.7	**74.8**	73.9	**74.4**
Pixelate	91.8	41.0	**73.5**	60.3	**73.4**	75.6	**69.4**	77.6	**78.5**	79.0	78.4	79.2	**79.4***
JPEG	88.0	73.5	**77.7**	75.3	72.3	76.8	71.9	79.5	77.7	81.0	78.8	80.8*	80.8*
Defocus Blur	92.1	54.9	**87.9***	47.5	**77.7**	70.0	**72.4**	72.1	**79.1**	73.5	**80.2**	73.8	**80.4**
Glass Blur	88.6	49.5	**64.7**	60.7	**62.0**	64.8	58.7	70.3	67.6	71.2	66.0	71.7*	66.4
Motion Blur	93.0	67.3	**87.6***	50.0	**70.9**	65.9	**67.3**	67.5	**75.2**	68.3	**75.7**	68.7	**76.2**
Zoom Blur	92.5	65.2	**84.0***	49.0	**70.3**	65.9	**71.2**	71.8	**77.2**	71.2	**79.3**	72.1	**79.6**
Snow	92.9	75.8	**87.8***	57.6	**73.8**	69.2	**71.9**	69.2	**76.1**	69.5	**76.5**	70.2	**76.3**
Frost	92.7	65.5	**85.9***	55.0	**72.0**	68.1	**68.4**	69.2	**75.6**	69.0	**75.2**	69.6	**76.7**
Fog	93.1	74.2	**89.6***	23.0	**63.4**	56.5	**63.6**	47.9	**70.1**	47.4	**68.7**	47.2	**69.3**
Brightness	93.7	92.0	**92.4***	65.1	**81.2**	73.5	**79.7**	72.5	**83.2**	73.5	**84.7**	73.8	**84.0**
Gaussian Noise	88.8	31.5	**77.3***	64.6	**68.4**	72.0	65.7	74.7	**74.9**	75.6	72.8	76.3	73.7
Shot Noise	90.0	37.9	**78.1***	65.0	**68.3**	72.2	68.0	76.4	74.7	76.6	74.9	76.9	74.8
Impulse Noise	93.7	33.8	**83.9***	44.2	**59.0**	63.0	58.2	67.6	64.9	66.8	64.6	67.9	65.4

Table 3: Unsupervised adaptation performance (accuracy) from CIFAR10 (clean) to CIFAR10-C dataset for each corruption (severity 5). Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

4.5 Benchmarking on image classification

4.5.1 Adapting to common image corruptions

We benchmarked methods for adaptation from CIFAR10 and ImageNet to images with common corruptions in CIFAR10-C and ImageNet-C, respectively (Hendrycks & Dietterich, 2019). These corruptions represent real-world distribution shifts that image classifiers encounter in the wild such as noise, weather and compression artifacts (Appendix Figure 6). FMM significantly bridged the visual gap between clean and corrupted images (Figure 4) and significantly outperformed benchmarked UDA methods on many corruptions and also improved other methods when combined with them (Table 3). Notably, on some target corruptions, some common UDA methods had worse performance than the unadapted baseline ERM model, i.e, the domain adaptation method actually hurt performance. This is in agreement with observations made in (Sagawa et al., 2022) of the deteriorated performance of common adaptation methods on some real-world distribution shifts. On ImageNet50-C, a significantly higher resolution dataset compared to CIFAR10-C, FMM was both effective standalone, i.e. ERM+FMM, and when combined with other methods (Table 4) across most target

Target	Oracle	ERM	+ FMM	MCC	+ FMM	MCD	+ FMM	ADDA	+ FMM	CDAN	+ FMM	DANN	+ FMM
Contrast	69.8	51.6	**74.0**	14.5	**66.0**	48.7	**53.8**	68.7	**71.6**	61.3	**74.7***	71.6	**72.8**
Elastic transform	71.9	66.3	**71.2**	63.9	62.9	57.0	55.6	71.1	**71.8**	74.0*	73.2	72.6	72.2
Pixelate	76.9	78.4	**80.8**	70.8	**71.3**	65.9	**66.2**	78.8	**79.5**	79.9	**81.3***	78.5	**80.8**
JPEG	77.8	69.8	**73.7**	66.4	**67.5**	61.4	61	74.6	**76.6**	75.4	**76.1**	76.3	**76.8***
Defocus blur	77.0	55.0	**74.7**	49.4	**66.4**	55.1	**58.7**	70.4	**76.6***	69.8	**76.6***	71.0	**74.4**
Glass blur	75.7	59.7	**68.9**	58.2	**67.5**	57.7	**61.4**	73.2	**75.1**	73.8	**76.0**	75.5	**77.2***
Motion blur	77.2	66.2	**77.2***	57.7	**67.4**	56.9	**61.2**	74.9	**76.4**	74.6	**77.2***	75.0	**75.8**
Zoom blur	78.4	63.4	**68.1**	60.8	**66.8**	57.0	**58.0**	71.3	**74.9**	73.3	**75.9***	73.6	**74.4**
Snow	72.8	46.8	**57.7**	31.2	**49.2**	47.0	**47.2**	62.4	**65.8**	56.2	**64.8**	66.0	**68.8***
Frost	70.4	51.8	**62.1**	25.5	**47.3**	46.9	**50.1**	62.4	**67.2**	57.6	**66.1**	62.7	**68.2***
Fog	76.0	51.8	**76.9***	25.3	**55.7**	54.0	**58.5**	67.0	**72.2**	66.3	**71.6**	65.2	**71.2**
Brightness	79.2	77.9	**78.5**	63.6	**68.0**	65.4	**65.5**	77.5	**78.2**	77.2	76.4	79.0*	77.5
Gaussian noise	77.2	49.0	**63.3**	47.7	**49.5**	54.1	52.4	71.9	71.0	70.6	70.6	72.0	**73.6***
Shot noise	76.8	44.1	**54.6**	46.4	**49.2**	54.7	51.6	72.3*	70.4	68.5	**70.2**	71.6	71.0
Impulse noise	70.0	31.8	**54.9**	44.9	44.4	50.4	47.0	67.7	**69.2**	68.2	66.6	70.5*	68.5

Table 4: Unsupervised adaptation performance on ImageNet50-C dataset (severity 2). Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

corruptions. The best performance was often achieved by a combination of FMM and another method (Table 4).

Method	Standalone	Method + FMM
	Test Acc.	Test Acc.
ERM	72.6	**74.9**
MDD	73.5	**75.7**
CDAN	71.2	**73.0**
JAN	68.7	**75.4**
DAN	69.5	**76.4***
DANN	70.1	**72.6**
Oracle	96.5	96.5

Table 5: UDA performance on WILDS-iWildCAM dataset. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

Method	Standalone	Method + FMM
	Test Acc.	Test Acc.
ERM	82.0	**84.1**
CORAL	77.9	**82.0**
DANN	68.4	**70.2**
Pseudo-Label	67.7	**74.7**
FixMatch	71.0	**83.9**
Noisy Student	86.7	86.6
SwAV	91.4*	91.0

Table 6: UDA performance on WILDS-Camelyon17 dataset. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods. Results were averaged across 10 runs with different random seeds.

4.5.2 Animal species classification

We benchmarked methods on domain adaptation between different camera traps in the WILDS-iWildCAM dataset. The task involves classifying animal species in images taken at different camera traps (domains), creating variation in illumination and background. We found that FMM consistently improved the performance of methods (Table 5). The best performance was achieved by the combination of DAN with FMM with an accuracy of 76.4%.

4.5.3 Tumor identification across different hospitals

We further benchmarked methods on domain adaptation between hospitals for a tumor identification task using the WILDS-Camelyon17 dataset. The task is to classify image patches as tumor or normal tissue on data from different hospitals (domains), which can differ in their patient demographics and data acquisition protocols. We benchmarked methods evaluated in WILDS (Sagawa et al., 2022) and found that FMM significantly improved the performance of ERM (+2.1%), CORAL (+4.1%), DANN (+1.8%), FixMatch (+12.9%) and Pseudo-Label (+6.0%). FMM did not improve or worsen the performance of Noisy Student and SwAV on this task.

4.6 Benchmarking on semantic segmentation

We benchmarked methods for domain adaptive semantic segmentation from Cityscapes to FoggyCityscapes (Cordts et al., 2016) and Synthia (Ros et al., 2016) to Cityscapes. We combined FMM with baseline methods and observed improved performance on both datasets (Table 7). Notably, FMM improved the performance of FDA, which replaces only low-frequencies, demonstrating the benefit in matching statistics across all frequencies instead of only low-frequencies as in FDA. FMM also improved the performance of ERM and AdvENT (Vu et al., 2019) across both tasks. On Synthia to Cityscapes, FMM improved the performance of HRDA (Hoyer et al., 2022), the current state-of-the-art method, to achieve a new improved mIoU of 66.2% (across 16 classes) or 72.7% (across 13 classes).

5 Discussion and future work

Here we discuss further analysis of our proposed method and its applicability to different tasks. We observed that there was a correlation between the degree of Fourier-shift between source and target domains and the performance improvement provided by FMM on ImageNet50-C (Appendix F). This suggests that the degree of shift maybe indicative of the performance gain that FMM could provide. We also evaluated FMM on standard domain adaptation tasks such as SVHN → MNIST and Office-Home (Appendix I). As these domain-shifts cannot be captured in the Fourier-amplitude spectrum alone, FMM was less effective here. Hence, FMM maybe better suited for the types of natural distribution shifts that we have considered rather than drastically different domains. We also investigated the effectiveness of matching only low-frequency statistics using FMM. However, we found that matching statistics across all frequencies provided better performance than matching only low-frequency statistics (Appendix H). Finally, we further experimented with test-time adaptation using FMM i.e. matching frequency statistics from target to source domains at test-time rather than from source to target domains at training-time. This approach is suitable when it is not possible to re-train models on the source domain but we wish to adapt models at test-time. We found that test-time FMM could improve performance over ERM in many cases, although not as effectively as train-time FMM. We leave exploration of the test-time adaptation setting to future work (see Appendix G for preliminary results and discussion).

	mIoU	Road	S.walk	Build.	Wall	Fence	Pole	Tr.Light	Sign	Veget.	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	M.bike	Bike
Cityscapes → FoggyCityscapes																				
ERM	51.2	95.3	70.2	64.1	31.9	35.2	30.7	33.3	51.1	42.3	44	32.1	64.4	47	86	64.4	56.4	21.1	43.1	60.8
ERM + FMM	**59.6**	**96.5**	**74.8**	**72.0**	**33.2**	**43.0**	**39.6**	**46.5**	**60.3**	**67.1**	**51.4**	**56.3**	**67.5**	**50.1**	**88.8**	**69.2**	**65.6**	**38.0**	**48.7**	**63.5**
AdvEnt	61.8	96.8	75.1	76.4	46.2	42.6	39.3	43.6	58.9	74.3	50.1	75.9	67.3	51	89.4	70.5	64.7	39.9	47.9	65
AdvEnt + FMM	**64.5***	**97.0**	**76.5**	**82.0**	43.0	**44.8**	**41.3**	**48.2**	**61.5**	**81.4**	**54.3**	**78.9**	**68.6**	**52.0**	**90.3**	**71.7**	**73.6**	**43.6**	**50.8**	**65.1**
FDA (β=0.001)	61.9	96.9	77.2	75.3	46.5	42	39.8	47.1	61	72.7	54.6	63.8	68.4	50.1	90.1	72.8	68	35.5	50.8	64.2
FDA (β=0.001) + FMM	**64.2**	96.9	76.5	77.3	45.0	**45.4**	**40.9**	**48.7**	60.9	**76.1**	**54.8**	**67.6**	**69.1**	**50.9**	**90.3**	72.1	**74.4**	**56.2**	**52.5**	**65.1**
FDA (β=0.05)	64.7	96.9	76.2	82.7	42.6	45.8	42.9	48.1	61.5	82.0	53.5	73.6	68.4	51.1	90.5	73.5	73.4	50.4	50.4	65.5
FDA (β=0.05) + FMM	**65.0**	96.9	**76.5**	82.6	42.2	45.3	43.2	**48.4**	**61.9**	81.8	**53.7**	75.5	**68.7**	**51.5**	90.4	70.2	**74.1**	**55.2**	**51.6**	65.4
Synthia → Cityscapes																				
ERM	40.2	57.0	20.8	75.8	-	-	-	8.7	18.0	75.0	-	83.5	53.3	17.6	53.2	-	22.2	-	12.2	25.3
ERM + FMM	**45.8**	**78.6**	**31.1**	**80.0**	-	-	-	**10.8**	**18.7**	**77.1**	-	83.5	52.2	**18.4**	**69.8**	-	**28.9**	-	**16.5**	**29.3**
AdvEnt	46.1	78.8	34.3	79.8	-	-	-	9.2	14.7	79.5	-	85.1	52.5	20.0	73.9	-	32.5	-	12.1	26.7
AdvEnt + FMM	**47.1**	**80.6**	33.9	**81.0**	-	-	-	**10.2**	**15.3**	**79.6**	-	84.1	48.2	19.8	**79.0**	-	31.5	-	**17.9**	**30.9**
FDA (β=0.001)	44.6	68.9	27.5	75.8	-	-	-	14.0	18.4	77.0	-	82.2	48.9	19.5	77.7	-	29.1	-	9.8	30.6
FDA (β=0.001) + FMM	**45.3**	**78.9**	**34.1**	**77.6**	-	-	-	10.4	17.6	**77.9**	-	**82.7**	47.3	15.8	72.5	-	**33.1**	-	**10.6**	30.1
FDA (β=0.05)	42.4	67.5	26.5	75.3	-	-	-	7.4	15.8	74.6	-	79.4	48.9	17.3	71.5	-	31.9	-	7.5	28.2
FDA (β=0.05) + FMM	**44.1**	**71.2**	**29.0**	**77.3**	-	-	-	**12.7**	**18.6**	**76.4**	-	79.4	44.3	16.9	70.8	-	30.7	-	**15.5**	**30.4**
HRDA	72.4	85.2	47.7	88.8	-	-	-	65.7	60.9	85.3	-	92.9	79.4	52.8	89.0	-	64.7	-	63.9	64.9
HRDA + FMM	**72.7***	**85.3**	**50.5**	87.4	-	-	-	65.2	**63.5**	**85.6**	-	**93.7**	76.5	**54.8**	88.8	-	**65.2**	-	**64.5**	64.7

Table 7: Benchmark for domain adaptive semantic segmentation in Cityscapes to FoggyCityscapes and Synthia to Cityscapes. Results are in bold if adding FMM to the baseline method improves performance. We report mIoU across 13 classes for Synthia to Cityscapes to be consistent with the literature. Results for classes not found or evaluated in the Synthia dataset are replaced with '-'. Results marked with '*' are best across methods.

6 Conclusion

Matching frequency statistics between domains is a simple and efficient approach to mitigate many real-world distribution shifts. Hence, we proposed Fourier Moment Matching, an input transformation that matches Fourier-statistics from source to target domains using unlabeled data. We demonstrated using extensive empirical evaluations across time-series and image tasks that FMM can improve the performance of existing methods and can also be used as a standalone method for many real-world distribution shift scenarios. We hope our work motivates further study on mitigating and analysing real-world distribution shifts using the Fourier domain.

References

Peter Bandi, Oscar Geessink, Quirine Manson, Marcory Van Dijk, Maschenka Balkenhol, Meyke Hermsen, Babak Ehteshami Bejnordi, Byungjae Lee, Kyunghyun Paeng, Aoxiao Zhong, and others. From detection of individual metastases to classification of lymph node status at the patient level: the CAMELYON17 challenge. *IEEE Transactions on Medical Imaging*, 2018.

Sara Beery, Elijah Cole, and Arvi Gjoka. The iWildCam 2020 Competition Dataset. *arXiv preprint arXiv:2004.10340*, 2020.

A. Buttfield, P.W. Ferrez, and Jd.R. Millan. Towards a robust BCI: error potentials and online learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(2):164–168, 2006. doi: 10.1109/TNSRE.2006.875555.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *Advances in Neural Information Processing Systems*, 2020.

Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-sheng Hua. HoMM: Higher-Order Moment Matching for Unsupervised Domain Adaptation. *Proceedings of the AAAI Conference on Artificial Intelligence*, April 2020. doi: 10.1609/aaai.v34i04.5745. URL https://ojs.aaai.org/index.php/AAAI/article/view/5745.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Josip Djolonga, Jessica Yung, Michael Tschannen, Rob Romijnders, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Matthias Minderer, Alexander D'Amour, Dan Moldovan, Sylvain Gelly, Neil Houlsby, Xiaohua Zhai, and Mario Lucic. On Robustness and Transferability of Convolutional Neural Networks. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. ISSN 1533-7928. URL http://jmlr.org/papers/v17/15-239.html.

A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C. K. Peng, and H. E. Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, June 2000.

Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. Acoustic scene classification in DCASE 2020 Challenge: generalization across devices and low complexity solutions. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2020 Workshop (DCASE2020)*, pp. 56–60, 2020a. URL https://arxiv.org/abs/2005.14623.

Toni Heittola, Annamaria Mesaros, and Tuomas Virtanen. TAU Urban Acoustic Scenes 2020 Mobile, Development dataset, February 2020b. URL https://zenodo.org/record/3819968.

Dan Hendrycks and Thomas Dietterich. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJz6tiCqYm.

Lukas Hoyer, Dengxin Dai, and Luc Van Gool. HRDA: Context-Aware High-Resolution Domain-Adaptive Semantic Segmentation. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner (eds.), *Computer Vision – ECCV 2022*. Springer Nature Switzerland, Cham, 2022. URL https://link.springer.com/10.1007/978-3-031-20056-4_22.

Aapo Hyvärinen, Jarmo Hurri, and Patrick O. Hoyer. *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision.* Springer Publishing Company, Incorporated, 1st edition, 2009. ISBN 1-84882-490-4.

Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum Class Confusion for Versatile Domain Adaptation. In *ECCV*, 2020.

Jason Jo and Yoshua Bengio. Measuring the tendency of CNNs to Learn Surface Statistical Regularities. *arXiv:1711.11561 [cs, stat]*, November 2017. URL http://arxiv.org/abs/1711.11561. arXiv: 1711.11561.

Junguang Jiang, Baixu Chen, Bo Fu, and Mingsheng Long. Transfer-Learning-library, 2020. URL https://github.com/thuml/Transfer-Learning-Library. Publication Title: GitHub repository.

Kiran Krishnamachari, See-Kiong Ng, and Chuan-Sheng Foo. Fourier Sensitivity and Regularization of Computer Vision Models. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=VmTYgjYloM.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *ICML Workshop on Challenges in Representation Learning*, 2013.

Yan Li, Hiroyuki Kambara, Yasuharu Koike, and Masashi Sugiyama. Application of Covariate Shift Adaptation Techniques in Brain–Computer Interfaces. *IEEE Transactions on Biomedical Engineering*, 57(6): 1318–1324, 2010. doi: 10.1109/TBME.2009.2039997.

Qiao Liu and Hui Xue. Adversarial Spectral Kernel Matching for Unsupervised Time Series Domain Adaptation. volume 3, pp. 2744–2750, August 2021. doi: 10.24963/ijcai.2021/378. URL https://www.ijcai.org/proceedings/2021/378. ISSN: 1045-0823.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on Machine Learning*, July 2015.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep Transfer Learning with Joint Adaptation Networks. In *Proceedings of the 34th International Conference on Machine Learning*, August 2017.

Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional Adversarial Domain Adaptation. In *Advances in Neural Information Processing Systems*, 2018.

John P. Miller, Rohan Taori, Aditi Raghunathan, Shiori Sagawa, Pang Wei Koh, Vaishaal Shankar, Percy Liang, Yair Carmon, and Ludwig Schmidt. Accuracy on the Line: on the Strong Correlation Between Out-of-Distribution and In-Distribution Generalization. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 7721–7735. PMLR, July 2021. URL https://proceedings.mlr.press/v139/miller21b.html. ISSN: 2640-3498.

A.V. Oppenheim and J.S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981. doi: 10.1109/PROC.1981.12022.

Mohamed Ragab, Emadeldeen Eldele, Wee Ling Tan, Chuan-Sheng Foo, Zhenghua Chen, Min Wu, Chee-Keong Kwoh, and Xiaoli Li. ADATIME: A Benchmarking Suite for Domain Adaptation on Time Series Data. *ACM Trans. Knowl. Discov. Data*, 17(8), May 2023. ISSN 1556-4681. doi: 10.1145/3587937. URL https://doi.org/10.1145/3587937. Place: New York, NY, USA Publisher: Association for Computing Machinery.

Mohammad Mahfujur Rahman, Clinton Fookes, Mahsa Baktashmotlagh, and Sridha Sridharan. On Minimum Discrepancy Estimation for Deep Domain Adaptation. In *Joint IJCAI/ECAI/AAMAS/ICML 2018 Workshop on Domain Adaptation for Visual Understanding.* doi: 10.1007/978-3-030-30671-7_6.

German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Fereshteh Sadeghi and Sergey Levine. CAD2RL: Real Single-Image Flight Without a Single Real Image. In Nancy M. Amato, Siddhartha S. Srinivasa, Nora Ayanian, and Scott Kuindersma (eds.), *Robotics: Science and Systems XIII, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, July 12-16, 2017*, 2017. doi: 10.15607/RSS.2017.XIII.034. URL http://www.roboticsproceedings.org/rss13/p34.html.

Shiori Sagawa, Pang Wei Koh, Tony Lee, Irena Gao, Sang Michael Xie, Kendrick Shen, Ananya Kumar, Weihua Hu, Michihiro Yasunaga, Henrik Marklund, Sara Beery, Etienne David, Ian Stavness, Wei Guo, Jure Leskovec, Kate Saenko, Tatsunori Hashimoto, Sergey Levine, Chelsea Finn, and Percy Liang. Extending the WILDS Benchmark for Unsupervised Adaptation. In *International Conference on Learning Representations (ICLR)*, 2022.

Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum Classifier Discrepancy for Unsupervised Domain Adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018.

Rui Shu, Hung Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T Approach to Unsupervised Domain Adaptation. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=H1q-TM-AW.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/06964dce9addb1c5cb5d6e3d9838f733-Paper.pdf.

Baochen Sun and Kate Saenko. Deep CORAL: Correlation Alignment for Deep Domain Adaptation. In Gang Hua and Hervé Jégou (eds.), *Computer Vision – ECCV 2016 Workshops*, 2016.

Baochen Sun, Jiashi Feng, and Kate Saenko. Return of Frustratingly Easy Domain Adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring Robustness to Natural Distribution Shifts in Image Classification. In *Advances in Neural Information Processing Systems*, 2020.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133.

Yusuke Tsuzuku and Issei Sato. On the Structural Sensitivity of Deep Convolutional Networks to the Directions of Fourier Basis Functions. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 51–60, Long Beach, CA, USA, June 2019. IEEE. ISBN 978-1-72813-293-8. doi: 10.1109/CVPR.2019.00014. URL https://ieeexplore.ieee.org/document/8954086/.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep Domain Confusion: Maximizing for Domain Invariance, December 2014. URL http://arxiv.org/abs/1412.3474. arXiv:1412.3474 [cs].

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial Discriminative Domain Adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Rob Romijnders, Nicolas Le Roux, and Ross Goroshin. Impact of Aliasing on Generalization in Deep Convolutional Networks. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 10509–10518, October 2021. doi: 10.1109/ICCV48922.2021.01036. ISSN: 2380-7504.

Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Perez. ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. URL https://ieeexplore.ieee.org/document/8954439/.

Garrett Wilson, Janardhan Rao Doppa, and Diane J. Cook. Multi-Source Deep Domain Adaptation with Weak Supervision for Time-Series Sensor Data. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2020.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. URL https://ieeexplore.ieee.org/document/9156610/.

Sang Michael Xie, Ananya Kumar, Robbie Jones, Fereshte Khani, Tengyu Ma, and Percy Liang. In-N-Out: Pre-Training and Self-Training using Auxiliary Information for Out-of-Distribution Robustness. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=jznizqvr15J.

Yanchao Yang and Stefano Soatto. FDA: Fourier Domain Adaptation for Semantic Segmentation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. URL https://ieeexplore.ieee.org/document/9157228/.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A Fourier Perspective on Model Robustness in Computer Vision. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2019.

Richard Zhang. Making Convolutional Networks Shift-Invariant Again. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 7324–7334. PMLR, May 2019. URL https://proceedings.mlr.press/v97/zhang19a.html. ISSN: 2640-3498.

Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael Jordan. Bridging Theory and Algorithm for Domain Adaptation. In *International Conference on Machine Learning*, pp. 7404–7413, 2019.

A Fourier-filtering

A.1 Radial Fourier-filtering

The mask radius r determines Fourier components that are preserved with larger radii preserving more components. We use (c_u, c_v) to denote the centre of the mask and $d(\cdot, \cdot)$ to denote Euclidean distance. The mask is applied on the Fourier transform of each image, denoted X, followed by the inverse transform, i.e. $X_{filtered} = \mathcal{F}^{-1}(\mathcal{F}(X) \odot M_r)$, where $\odot$ is the element-wise product. Fourier-filtering is performed on each color channel independently. Formally, the radial mask M_r is:

$$M_r(u, v) := \begin{cases} 1, & \text{if } d((u, v), (c_u, c_v)) \leq r \\ 0, & \text{otherwise} \end{cases}$$

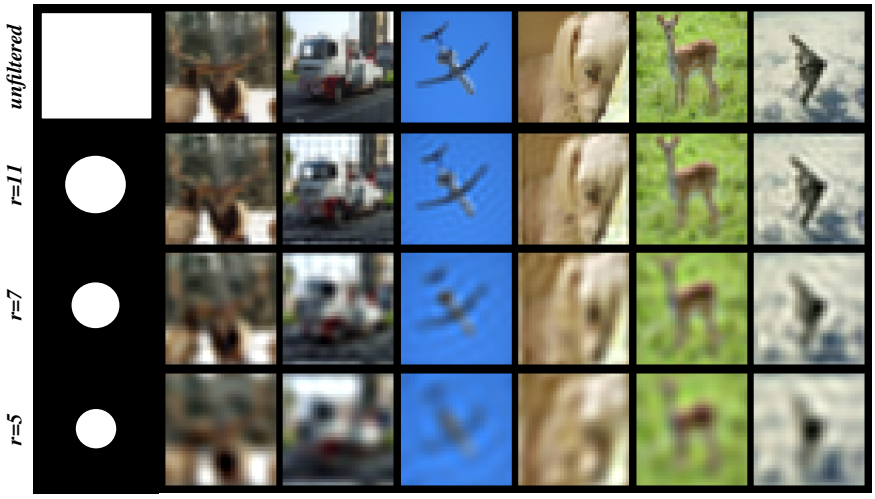

Figure 5: First image in each row is the mask in Fourier space (lowest frequency at centre). White pixels preserve and black pixels set Fourier components to zero. Top row are original CIFAR10 images, other rows are Fourier-filtered with different radial masks.

B Input-space moment matching

We trained models with input-space moment matching (termed IMM) (see Table 8 below) using the same architecture as FMM. While IMM could improve upon the unadapted ERM baseline, the FMM transform resulted in better accuracy, especially on time-series classification and noise distortions.

Method	Sleep-EDF		TAU Audio		CIFAR10 → CIFAR10C			
	Acc.	F1	Acc.	F1	Noise	Blur	Weather	Digital
ERM	64.5	54.4	24.2	19.1	34.4	59.2	76.9	61.0
ERM + IMM	65.4	54.9	23.8	18.2	63.1	77.0	85.2	78.1
ERM + FMM	**74.1**	**61.7**	**30.3**	**27.8**	**79.8**	**81.1**	**88.9**	**80.0**

Table 8: Performance on Sleep-EDF (EEG), TAU (audio) and CIFAR10C. ERM+IMM is a supervised classifier trained on input-space moment matched (IMM) source samples.

C Time-series: Sleep Stage Classification

C.1 Method hyper-parameters

We used hyper-parameters found for each method by Ragab et al. (2023). Hyper-parameters were chosen using an extensive random search involving 100 hyper-parameter combinations. The hyper-parameters that produced the best target risk were chosen in our experiments. The chosen hyper-parameters are listed in Table 9.

C.2 Architecture

As used in Ragab et al. (2023), the network is a 3-block CNN with each block comprising a 1D convolutional layer, followed by a 1D Batch Normalization layer, a ReLU non-linearity and a 1D MaxPooling layer. The convolutional layer in the first block has a kernel size of 25 and stride 1. The implementation can be accessed in the GitHub repository released in Ragab et al. (2023).

Table 9: Hyperparameters for Sleep Stage Classification.

Method	Hyperparameter	Value
ERM	Learning Rate	5e-4
DANN	Learning Rate	5e-4
	Source loss weight	8.30
	Domain loss weight	0.324
Deep CORAL	Learning Rate	5e-4
	Source loss weight	9.39
	CORAL weight	0.19
DDC	Learning Rate	5e-4
	Source loss weight	2.951
	Domain loss weight	8.923
HoMM	Learning Rate	5e-4
	Source loss weight	0.197
	Domain loss weight	1.102
CoDATS	Learning Rate	1e-2
	Source loss weight	9.239
	Domain loss weight	1.342
AdvSKM	Learning Rate	5e-4
	Source loss weight	2.50
	Domain loss weight	2.50
MMDA	Learning Rate	5e-4
	Source loss weight	4.48
	MMD weight	5.951
	CORAL weight	3.36
	Conditional Entropy weight	6.13
CDAN	Learning Rate	1e-3
	Source loss weight	6.803
	Domain loss weight	4.726
	Conditional Entropy weight	1.307
DIRT-T	Learning Rate	5e-3
	Source loss weight	9.183
	Domain loss weight	7.411
	Conditional Entropy weight	2.564
	VAT loss weight	3.583

	mean		0 → 11		12 → 5		16 → 1		7 → 18		9 → 14	
Method	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
ERM	64.5	54.4	54.4	47.0	71.2	50.7	54.4	52.0	62.5	57.7	79.9	64.5
ERM + FMM	**74.1**	**61.7**	70.0	55.0	67.1	51.6	76.2	63.9	75.6	71.2	81.9	66.7
DANN	70.3	59.2	56.3	43.7	73.6	60.1	67.2	58.3	74.2	68.5	80.3	65.5
DANN + FMM	**72.5**	**61.0**	60.2	47.5	71.4	58.6	78.2	63.5	75.2	70.7	77.4	64.3
DeepCORAL	69.3	57.7	54.2	41.9	69.0	53.8	68.7	58.8	74.2	68.5	80.6	65.5
DeepCORAL + FMM	**74.5**	**63.0**	67.7	55.9	68.0	54.3	78.3	65.3	75.8	71.8	82.6	67.7
MMDA	72.3	61.4	52.6	41.9	80.7	65.6	71.4	59.8	77.1	72.3	79.6	67.3
MMDA + FMM	**77.0**	**65.3**	67.2	53.3	78.7	67.2	77.1	64.5	76.6	71.7	85.4	69.6
DIRT-T	68.4	58.1	33.0	29.2	75.7	62.7	74.8	61.3	74.1	68.5	84.7	68.7
DIRT-T + FMM	**73.5**	**62.1**	53.5	41.4	74.7	63.6	80.6	65.2	73.2	67.4	85.7	72.9
CDAN	71.2	59.6	48.5	39.7	80.7	65.9	69.8	59.6	77.1	69.1	79.9	63.9
CDAN + FMM	**76.1**	**63.3**	62.0	49.2	77.1	64.2	80.6	65.1	76.5	70.5	84.3	67.7
HoMM	70.1	58.3	52.3	38.9	71.6	57.3	69.2	58.6	75.0	69.2	82.6	67.5
HoMM + FMM	**74.3**	**62.4**	64.1	50.8	69.9	56.7	78.7	65.2	76.7	72.4	82.0	67.1
CoDATS	67.0	58.5	37.8	33.0	73.0	60.3	69.2	60.3	74.9	68.1	80.2	70.9
CoDATS + FMM	**72.0**	**60.2**	63.1	48.9	66.5	51.3	71.0	62.2	77.5	70.7	81.8	67.7
DDC	69.4	57.8	54.3	42.1	69.0	53.7	68.9	58.9	74.3	68.6	80.7	65.6
DDC + FMM	**74.5**	**62.9**	67.8	55.9	67.8	54.0	78.3	65.0	75.8	71.8	82.6	67.8
AdvSKM	74.1	62.4	68.7	59.1	74.2	58.5	72.9	61.3	72.0	65.9	82.6	67.3
AdvSKM + FMM	**74.7**	**62.8**	67.8	55.1	69.8	56.7	75.7	61.1	78.5	74.1	81.7	66.7
Oracle	87.5	77.6	85.3	68.0	86.5	76.4	89.1	83.6	83.8	78.5	92.6	81.6

Table 10: Domain adaptive acoustic scene classification performance on Sleep-EDF dataset. Results are in bold if FMM improves performance when added to baseline method. Results are averaged over three runs with different random seed.

D Time-series: Acoustic Scene Classification

D.1 Melspectrogram Pre-processing

```
mel_spectrogram = MelSpectrogram(sample_rate=44100, n_fft=2048, hop_length=1024,
n_mels=128, f_min=0.0, f_max=44100/2, mel_scale="htk", norm=None)
```

	mean		A → B		A → C		A → S1		A → S2		A → S3	
Method	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**	**Acc.**	**F1**
ERM	21.7	16.7	26.5	22.2	32.1	27.2	21.4	15.1	20.0	15.4	20.7	15.6
ERM + FMM	**30.4**	**28.0**	36.2	35.6	31.4	29.6	29.9	27.0	27.0	21.9	27.2	25.0
CDAN	28.8	27.1	37.6	36.4	46.9	44.3	22.5	20.4	18.9	17.6	23.4	23.2
CDAN + FMM	**38.4**	**37.3**	41.1	39.8	46.3	44.9	35.2	35.1	32.6	31.5	39.3	38.2
CoDATS	26.2	24.6	30.1	29.3	41.8	39.6	22.6	20.3	19.2	17.0	22.9	20.8
CoDATS + FMM	**30.1**	**27.5**	36.6	36.0	33.5	31.5	31.6	29.7	29.3	27.0	34.3	31.5
DANN	25.8	24.7	35.2	34.1	43.7	41.9	19.5	18.4	14.9	14.8	17.2	16.5
DANN + FMM	**35.1**	**34.0**	41.9	41.6	41.6	39.4	35.3	33.9	29.2	27.7	33.9	34.1
DDC	25.2	22.8	30.8	29.0	38.8	35.8	20.7	17.5	20.9	17.6	24.3	22.5
DDC + FMM	**33.6**	**32.3**	37.9	37.5	39.1	38.2	29.5	28.4	29.4	25.8	34.6	34.2
HoMM	28.1	26.3	34.9	33.9	43.3	40.4	23.9	22.3	18.0	16.9	23.5	22.1
HoMM + FMM	**34.3**	**32.9**	37.8	36.9	40.3	39.2	32.0	32.4	28.9	26.3	33.8	32.4
MMDA	21.9	20.1	27.4	26.0	36.8	32.9	14.9	13.6	13.6	12.2	18.4	16.6
MMDA + FMM	**28.6**	**28.0**	32.4	32.8	44.1	43.3	22.1	22.9	22.5	22.0	28.1	26.9
AdvSKM	26.8	23.8	33.9	32.5	41.5	38.1	22.6	19.1	23.3	20.0	21.6	18.2
AdvSKM + FMM	**33.1**	**32.0**	36.2	36.1	39.7	38.5	26.6	24.6	28.4	26.6	34.3	33.6
DeepCORAL	25.0	22.6	29.9	28.4	42.4	39.1	19.6	16.7	21.1	17.6	20.7	18.1
DeepCORAL + FMM	**32.8**	**31.6**	36.6	36.4	37.4	36.9	30.8	29.0	27.3	24.6	32.9	32.1
DIRT-T	27.7	24.0	38.7	34.7	45.7	40.6	19.6	17.8	14.7	10.4	23.7	19.3
DIRT-T + FMM	**33.3**	**31.8**	40.4	38.6	44.8	43.4	26.8	25.9	29.2	26.9	30.6	30.1
Oracle	45.2	44.8	52.6	51.1	51.4	50.7	40.3	39.4	42.4	42.9	39.4	39.8

Table 11: Domain adaptive acoustic scene classification performance on TAU Urban Audio dataset. Results are in bold if FMM improves performance when added to baseline method. Results are averaged over three runs with different random seed.

E Image Experiments

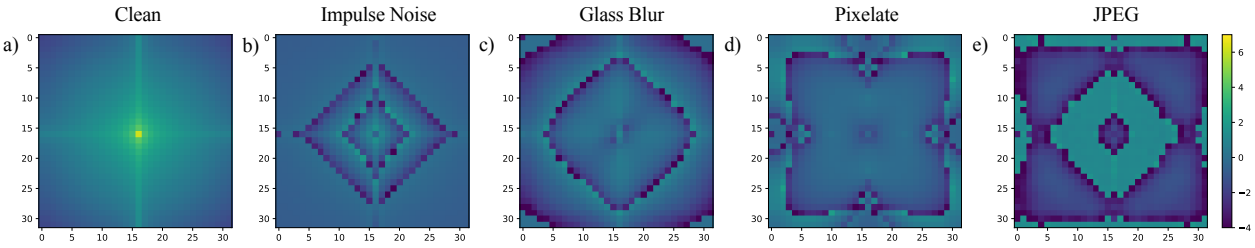

Figure 6: a) Mean (first order moment) of Fourier amplitude-spectrum, i.e., $\mathbb{E}[||\mathcal{F}(x))||_2]$, of clean CIFAR10 images in log-scale. b,c,d,e) Difference between means of the Fourier-amplitude spectra of clean and corrupted images (severity 5) in CIFAR10-C, absolute values are in log-scale.

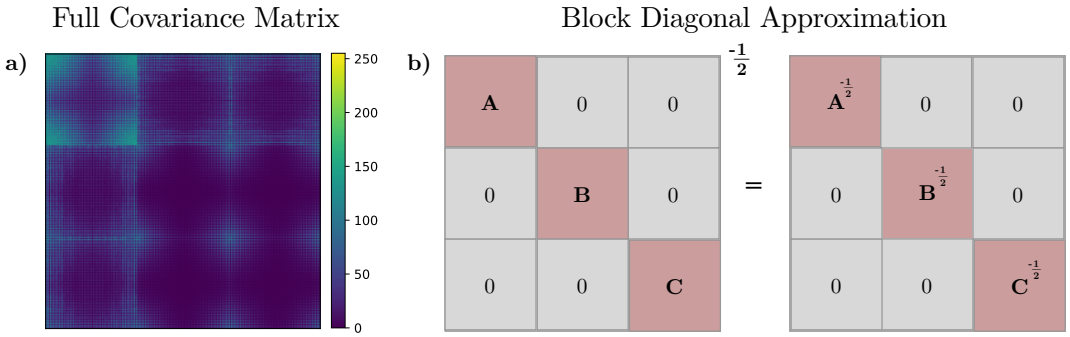

Figure 7: Block diagonal approximation of Fourier amplitude covariance matrix. a) CIFAR10 covariance matrix absolute values (low-frequencies in top-left). The top-left diagonal sub-matrix comprises ∼60% of the total l_1 norm. b) Block diagonal matrices can be inverted or square-rooted by operating on individual diagonal sub-matrices. This reduces peak memory requirements compared to operating on the entire matrix.

E.1 ImageNet50

Randomly chosen ImageNet classes in ImageNet50, which contains 65,000 training images each of size 224×224:

n01440764, n01484850, n01494475, n01531178, n01632777, n01665541, n01687978, n01695060, n01749939, n01775062, n01795545, n01818515, n01820546, n01824575, n01833805, n01914609, n01924916, n01930112, n01950731, n01978455, n01984695, n02007558, n02012849, n02018795, n02037110, n01443537, n01514668, n01514859, n01537544, n01592084, n01608432, n01677366, n01698640, n01728572, n01729977, n01735189, n01740131, n01753488, n01770081, n01773157, n01773549, n01773797, n01774384, n01843383, n01955084, n02018207, n02027492, n02028035, n02058221, n02077923.

Method	Hyperparameter	CIFAR10-C		ImageNet50-C	
		Standalone	Method + FMM	Standalone	Method + FMM
Baseline	Learning rate	1×10^{-1}	1×10^{-1}	1×10^{-1}	1×10^{-1}
	Momentum	0.9	0.9	0.9	0.9
	Batch size	128	128	64	64
	Weight decay	5×10^{-4}	5×10^{-4}	1×10^{-4}	1×10^{-4}
	Epochs	150	150	90	90
	Step-LR (epochs)	50	50	30	30
MCC	Learning Rate	5×10^{-3}	1×10^{-1}	5×10^{-3}	5×10^{-3}
	trade-off (λ)	1.0	0.1	1.0	0.5
	temperature	2.5	2.5	2.5	2.5
MCD	Learning Rate	1×10^{-3}	1×10^{-1}	1×10^{-3}	1×10^{-1}
	trade-off (λ)	1.0	0.1	1.0	0.1
	trade-off-entropy	0.03	0.03	0.03	0.03
ADDA	Learning Rate	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-1}
	trade-off (λ)	0.1	0.1	0.1	0.5
CDAN	Learning Rate	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-1}
	trade-off (λ)	1.0	1.0	1.0	0.5
DANN	Learning Rate	1×10^{-2}	1×10^{-2}	1×10^{-2}	1×10^{-1}
	trade-off (λ)	1.0	1.0	1.0	1.0

Table 12: Hyper-parameters for methods on CIFAR10-C and ImageNet50-C datasets.

Method	Hyperparameter	Standalone	Method + FMM
Baseline	Learning Rate	1.0	1.0
	Epochs	18	18
	Batch size	24	24
DANN	Learning Rate	1.0	1.0
	trade-off (λ)	0.3	0.3
DAN	Learning Rate	0.3	0.3
	trade-off (λ)	0.3	0.3
MDD	Learning Rate	0.3	0.3
	trade-off (λ)	0.3	0.1
CDAN	Learning Rate	0.3	0.3
	trade-off (λ)	0.3	0.3
JAN	Learning Rate	0.3	0.3
	trade-off (λ)	0.3	0.3

Table 13: Hyper-parameters for methods on iWildCAM dataset.

Method	Hyperparameter	Standalone	Method + FMM
Baseline	Learning Rate	2.5×10^{-3}	2.5×10^{-3}
	Weight decay	5×10^{-4}	5×10^{-4}
	Momentum	0.9	0.9
	Epochs	60	60
	Batch size	2	2
AdvENT	trade-off (λ)	0.001	0.001
FDA	trade-off (λ)	0.001	0.001
	ITA (robust entropy)	2.0	2.0

Table 14: Hyper-parameters for semantic segmentation methods.

Target	ERM	+ FMM (b=1)	+ FMM (b=3,072)
Contrast	56.3	86.1	**92.8**
Elastic Transform	73.2	76.5	**76.1**
Pixelate	41.0	58.7	**73.5**
JPEG	73.5	72.5	**77.7**
Defocus Blur	54.9	86.7	**87.9**
Glass Blur	49.5	59.3	**64.7**
Motion Blur	67.3	84.2	**87.6**
Zoom Blur	65.2	79.5	**84.0**
Snow	75.8	84.2	**87.8**
Frost	65.5	79.1	**85.9**
Fog	74.2	86.6	**89.6**
Brightness	92.0	92.3	**92.4**
Gaussian Noise	31.5	60.0	**77.3**
Shot Noise	37.9	63.8	**78.1**
Impulse Noise	33.8	53.7	**83.9**

Table 15: UDA performance (accuracy) on CIFAR10-C for second-order FMM with diagonal (b=1) vs full-covariance (b=3,072) matrix approximations.

F Degree of Fourier-shift and FMM performance improvement on ImageNet50-C

We computed the correlation between the degree of Fourier-shift, as measured by the difference in the first or second statistical moment between source and target domains in the ImageNet-C task, and the relative performance improvement provided by FMM over ERM. While we did not find any significant correlation between performance improvement and shift in the mean of the Fourier-amplitude spectrum between source and target data (Figure 8a, Pearson correlation -0.03), we observed a higher correlation between performance improvement and shift in the covariance structure (Figure 8b, Pearson correlation +0.47). This maybe indicative of the importance of matching the covariance structure on this dataset.

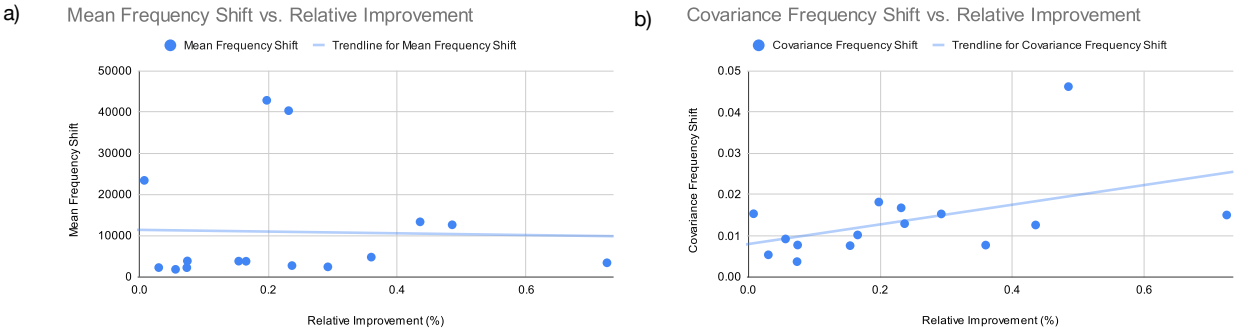

Figure 8: a) Relative performance improvement of FMM over ERM and the norm of the difference between the first moments (mean) of source and target domains in ImageNet50-C. b) Relative performance improvement of FMM over ERM and the norm of the difference between the second moments (covariance) of source and target domains in ImageNet50-C.

G Test-time adaptation using FMM

We further evaluated FMM at test-time i.e. rather than matching source to target statistics, we matched target samples to match source statistics at test time. This setting is called *test-time adaptation* and does not require any modification to the model or training procedure on the source domain. Hence, this approach is beneficial when it is not possible to re-train models on the source domain but we wish to adapt models at test time. We found that test-time FMM could improve performance in many cases, although not as effectively as train-time FMM (Tables 16, 17). This suggests that there is some additional benefit in training the model on the transformed source samples rather than just modifying the target samples at test-time. In the case of adapting to distorted images, one reason for this maybe that it is easier to distort clean images compared to denoising distorted images back to their clean versions at test-time. This could explain why test-time FMM worsens performance in some cases on the CIFAR10-C task (Table 17). We leave further exploration of the test-time adaptation setting for future work.

Method	Sleep-EDF		TAU Audio	
	Acc.	F1	Acc.	F1
ERM	64.5	54.4	24.2	19.1
ERM + FMM (test-time)	71.1	59.1	**33.0**	**30.7**
ERM + FMM (train-time)	**74.1**	**61.7**	30.3	27.8

Table 16: Performance of test-time FMM on Sleep-EDF (EEG) and TAU (audio) datasets. Results were averaged across all source-target domain pairs and three runs with different random seed.

Target	ERM	+ FMM (test-time)	+ FMM (train-time)
Contrast	56.3	76.5	**92.8**
Elastic Transform	73.2	51.4	**76.1**
Pixelate	41.0	52.1	**73.5**
JPEG	73.5	69.0	**77.7**
Defocus Blur	54.9	63.6	**87.9**
Glass Blur	49.5	52.6	**64.7**
Motion Blur	67.3	62.4	**87.6**
Zoom Blur	65.2	73.0	**84.0**
Snow	75.8	29.1	**87.8**
Frost	65.5	74.5	**85.9**
Fog	74.2	82.0	**89.6**
Brightness	92.0	88.0	**92.4**
Gaussian Noise	31.5	62.8	**77.3**
Shot Noise	37.9	64.8	**78.1**
Impulse Noise	33.8	57.1	**83.9**

Table 17: UDA performance (accuracy) on CIFAR10-C for ERM vs FMM (test-time) vs FMM (train-time).

H Matching low-frequency statistics

We restricted FMM to the low-frequencies to investigate the effectiveness of matching statistics of only low-frequencies instead of all frequencies. We found that matching only low-frequency statistics, which is similar to Fourier Domain Adaptation (Yang & Soatto, 2020), was not as effective as matching statistics across all frequencies. Here we varied β, where β refers to the proportion of frequencies used to perform FMM, from the lowest to highest frequencies. Smaller values of β incorporate only low frequencies while higher β values incorporate both low and high frequencies ($\beta = 1$ refers to using all the frequencies as in standard FMM). We observed improved performance as we increased $\beta \rightarrow 1$ (Tables 18, 19).

Method	Sleep-EDF	
	Acc.	F1
ERM	64.5	54.4
ERM + FMM ($\beta = 0.01$)	64.5	54.0
ERM + FMM ($\beta = 0.02$)	64.7	54.1
ERM + FMM ($\beta = 0.2$)	66.2	55.4
ERM + FMM ($\beta = 0.4$)	69.8	58.1
ERM + FMM ($\beta = 0.6$)	69.9	58.5
ERM + FMM ($\beta = 0.8$)	70.0	58.6
ERM + FMM ($\beta = 1.0$)	**74.1**	**61.7**

Table 18: Performance of FMM on Sleep-EDF (EEG) at varying β. Smaller β values uses only lower-frequencies while higher values use lower and higher frequencies. $\beta = 1$ corresponds to standard FMM (uses all frequencies). Results were averaged across all source-target domain pairs and three runs with different random seed.

Method	Gaussian Noise	Frost	Defocus blur	Contrast
ERM	31.5	65.5	54.9	56.3
ERM + FMM ($\beta = 0.2$)	31.6	69.5	56.3	57.0
ERM + FMM ($\beta = 0.4$)	35.7	71.9	61.2	58.9
ERM + FMM ($\beta = 0.6$)	38.9	74.7	75.5	78.6
ERM + FMM ($\beta = 0.8$)	42.9	78.3	80.2	81.2
ERM + FMM ($\beta = 1.0$)	**77.3**	**85.9**	**87.9**	**92.8**

Table 19: Performance of FMM on CIFAR10-C at varying β. Smaller β values uses only lower-frequencies while higher values use lower and higher frequencies. $\beta = 1$ corresponds to standard FMM (uses all frequencies).

I Evaluating FMM on standard domain-shift datasets

FMM was specifically designed for tackling natural variations in real-world applications as many existing domain adaptation methods were shown to be less effective here. Existing domain adaptation methods were largely developed on standard domain-shift datasets e.g. SVHN → MNIST or Office-Home. As these domain-shifts cannot be captured in the Fourier-amplitude spectrum alone, FMM was less effective here (Tables 20, 21). Hence, FMM is suited for natural distribution shifts in real-world applications rather than drastically different domains.

	mean	SVHN → MNIST	MNIST → USPS	USPS → MNIST
ERM	76.9	74.1	82.1	74.6
ERM + FMM	**77.6**	74.4	81.4	76.9
CDAN	95.8	93.8	96	97.7
CDAN + FMM	**96.0**	93.5	96.6	97.8
JAN	87.0	90.3	84.0	86.8
JAN + FMM	**88.8**	90.3	85.1	90.9
DANN	92.5	90.8	91.7	95.2
DANN + FMM	91.0	83.0	93.6	96.5

Table 20: Evaluating FMM on shifts between MNIST, SVHN, USPS.

	mean	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr
ERM	58.4	41.1	65.9	73.7	53.1	60.1	63.3	52.2	36.7	71.8	64.8	42.6	75.2
ERM + FMM	**58.7**	40.0	66.1	73.7	52.5	62.1	64.2	53.2	37.5	73.0	65.2	40.7	76.0
CDAN	68.8	55.2	72.4	77.6	62	69.7	70.9	62.4	54.3	80.5	75.5	61	83.8
CDAN + FMM	68.3	51.2	71.6	77.9	61.2	70.8	71.5	60.7	54.5	80.3	75.0	60.8	84.1
DANN	65.2	53.8	62.6	74.0	55.8	67.3	67.3	55.8	55.1	77.9	71.1	60.7	81.1
DANN + FMM	65.2	52.3	62.2	73.7	55.1	66.7	67.4	59.8	54.6	78.4	70.7	59.7	82
JAN	65.9	50.8	71.9	76.5	60.6	68.3	68.7	60.5	49.6	76.9	71.0	55.9	80.5
JAN + FMM	65.4	49.6	70.6	75.9	59.3	67.7	69.1	60.1	47.9	76.6	71.7	55.2	80.5

Table 21: Evaluating FMM on shifts between Office-Home domains: Ar (Art), Cl (Clipart), Pr (Product), Rw (Real-world).

J Benchmarks

Method	Sleep-EDF				TAU Audio			
	Standalone		Method + FMM		Standalone		Method + FMM	
	Acc.	F1	Acc.	F1	Acc.	F1	Acc.	F1
ERM	64.5 ± 3.8	54.4 ± 3.1	**74.1** ± 3.6	**61.7** ± 3.6	24.2 ± 2.0	19.1 ± 1.8	**30.3** ± 2.2	**27.8** ± 2.0
DANN	70.3 ± 3.9	59.2 ± 3.6	**72.5** ± 2.7	**61.0** ± 3.5	26.1 ± 3.7	25.1 ± 3.8	**36.4** ± 2.4	**35.3** ± 3.0
DeepCORAL	69.3 ± 2.5	57.7 ± 1.7	**74.5** ± 3.1	**63.0** ± 3.9	26.7 ± 2.1	24.0 ± 2.3	**33.0** ± 2.0	**31.8** ± 1.9
MMDA	72.3 ± 4.7	61.4 ± 4.3	**77.0*** ± 2.8	**65.3*** ± 2.5	22.2 ± 4.6	20.3 ± 4.7	**29.8** ± 1.2	**29.6** ± 1.7
DIRT-T	68.4 ± 4.2	58.1 ± 3.5	**73.5** ± 2.7	**62.1** ± 2.8	28.5 ± 3.7	24.6 ± 3.5	**34.4** ± 2.2	**33.0** ± 2.3
CDAN	71.2 ± 6.6	59.6 ± 4.0	**76.1** ± 1.6	**63.3** ± 1.6	29.9 ± 2.7	28.4 ± 2.6	**38.9*** ± 2.6	**37.9*** ± 2.7
HoMM	70.1 ± 2.0	58.3 ± 1.8	**74.3** ± 2.6	**62.4** ± 2.9	28.7 ± 2.3	27.1 ± 2.6	**34.6** ± 3.5	**33.4** ± 3.5
CoDATS	67.0 ± 4.2	58.5 ± 3.6	**72.0** ± 10.4	**60.2** ± 8.8	27.3 ± 4.2	25.4 ± 4.3	**33.1** ± 3.9	**31.1** ± 3.8
DDC	69.4 ± 2.4	57.8 ± 1.7	**74.5** ± 3.1	**62.9** ± 3.9	27.1 ± 3.4	24.5 ± 3.3	**34.1** ± 1.5	**32.8** ± 1.4
AdvSKM	74.1 ± 2.7	62.4 ± 1.6	**74.7** ± 3.3	**62.8** ± 2.2	28.6 ± 2.3	25.6 ± 2.6	**33.0** ± 3.5	**31.9** ± 3.6
Oracle	87.5 ± 1.03	77.6 ± 2.2	87.5 ± 1.03	77.6 ± 2.2	45.2 ± 1.8	44.8 ± 2.0	45.2 ± 1.8	44.8 ± 2.0

Table 22: Performance on Sleep-EDF (EEG) and TAU (audio) datasets. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods. Results were averaged across all source-target domain pairs and three runs with different random seed.

Method	Standalone	Method + FMM
	Test Acc.	Test Acc.
ERM	72.6 ± 1.2	**74.9** ± 1.4
MDD	73.5 ± 1.9	**75.7** ± 1.3
CDAN	71.2 ± 1.7	**73.0** ± 1.5
JAN	68.7 ± 1.6	**75.4** ± 1.2
DAN	69.5 ± 2.0	**76.4*** ± 2.1
DANN	70.1 ± 1.4	**72.6** ± 1.6
Oracle	96.5 ± 1.9	96.5 ± 1.9

Table 23: UDA performance on WILDS-iWildCAM dataset. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

Method	Standalone	Method + FMM
	Test Acc.	Test Acc.
ERM	82.0 ± 7.4	**84.1** ± 5.4
CORAL	77.9 ± 6.6	**82.0** ± 6.5
DANN	68.4 ± 9.2	**70.2** ± 7.5
Pseudo-Label	67.7 ± 8.2	**74.7** ± 8.1
FixMatch	71.0 ± 4.9	**83.9** ± 6.0
Noisy Student	86.7 ± 1.7	86.6 ± 2.0
SwAV	91.4* ± 2.0	91.0 ± 2.5

Table 24: UDA performance on WILDS-Camelyon17 dataset. Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods. Results were averaged across 10 runs with different random seeds.

Target	Oracle	ERM	+ FMM	MCC	+ FMM	MCD	+ FMM	ADDA	+ FMM	CDAN	+ FMM	DANN	+ FMM
Contrast	92.3	56.3 ± 2.4	**92.8*** ± 1.4	19.0 ± 3.4	**72.8** ± 4.2	43.9 ± 4.3	**66.6** ± 4.0	28.4 ± 2.3	**76.0** ± 4.3	29.6 ± 3.4	**76.7** ± 3.1	27.3 ± 4.3	**76.9** ± 5.0
Elastic Transform	90.0	73.2 ± 2.4	**76.1*** ± 2.2	68.0 ± 2.5	**72.1** ± 1.9	70.5 ± 2.0	66.5 ± 2.1	72.3 ± 2.3	**74.8** ± 2.1	73.7 ± 2.1	**74.8** ± 1.9	73.9 ± 1.2	**74.4** ± 1.3
Pixelate	91.8	41.0 ± 2.9	**73.5** ± 3.2	60.3 ± 3.4	**73.4** ± 3.2	75.6 ± 2.1	69.4 ± 2.2	77.6 ± 1.8	**78.5** ± 1.5	79.0 ± 2.1	78.4 ± 2.3	79.2 ± 1.2	**79.4*** ± 1.1
JPEG	88.0	73.5 ± 2.4	**77.7** ± 1.9	75.3 ± 2.1	72.3 ± 1.7	76.8 ± 2.1	71.9 ± 2.2	79.5 ± 1.9	77.7 ± 1.6	81.0 ± 2.2	78.8 ± 3.1	80.8* ± 3.2	80.8* ± 3.3
Defocus Blur	92.1	54.9 ± 3.4	**87.9*** ± 4.2	47.5 ± 4.6	**77.7** ± 4.0	70.0 ± 2.1	**72.4** ± 1.9	72.1 ± 2.3	**79.1** ± 2.9	73.5 ± 2.1	**80.2** ± 2.2	73.8 ± 2.1	**80.4** ± 1.9
Glass Blur	88.6	49.5 ± 2.3	**64.7** ± 3.2	60.7 ± 2.2	**62.0** ± 1.9	64.8 ± 2.1	58.7 ± 1.8	70.3 ± 1.5	67.6 ± 1.2	71.2 ± 1.8	66.0 ± 2.0	71.7* ± 1.3	66.4 ± 1.9
Motion Blur	93.0	67.3 ± 2.2	**87.6*** ± 2.9	50.0 ± 1.9	**70.9** ± 2.1	65.9 ± 1.9	**67.3** ± 1.8	67.5 ± 1.9	**75.2** ± 2.1	68.3 ± 1.9	**75.7** ± 2.1	68.7 ± 2.1	**76.2** ± 2.8
Zoom Blur	92.5	65.2 ± 3.2	**84.0*** ± 3.1	49.0 ± 3.4	**70.3** ± 4.1	65.9 ± 3.2	**71.2** ± 2.1	71.8 ± 1.9	**77.2** ± 1.1	71.2 ± 2.1	**79.3** ± 3.2	72.1 ± 1.8	**79.6** ± 1.9
Snow	92.9	75.8 ± 1.8	**87.8*** ± 2.3	57.6 ± 3.6	**73.8** ± 3.2	69.2 ± 2.8	**71.9** ± 1.7	69.2 ± 3.3	**76.1** ± 2.3	69.5 ± 2.2	**76.5** ± 2.1	70.2 ± 1.8	**76.3** ± 1.4
Frost	92.7	65.5 ± 3.5	**85.9*** ± 2.7	55.0 ± 1.9	**72.0** ± 2.1	68.1 ± 1.8	**68.4** ± 1.7	69.2 ± 1.9	**75.6** ± 2.2	69.0 ± 1.9	**75.2** ± 1.3	69.6 ± 2.3	**76.7** ± 3.2
Fog	93.1	74.2 ± 2.4	**89.6*** ± 1.9	23.0 ± 3.2	**63.4** ± 2.9	56.5 ± 1.8	**63.6** ± 2.3	47.9 ± 1.9	**70.1** ± 2.8	47.4 ± 3.0	**68.7** ± 2.7	47.2 ± 2.1	**69.3** ± 2.2
Brightness	93.7	92.0 ± 1.2	**92.4*** ± 1.3	65.1 ± 2.3	**81.2** ± 2.4	73.5 ± 1.8	**79.7** ± 2.0	72.5 ± 2.1	**83.2** ± 2.8	73.5 ± 3.1	**84.7** ± 3.2	73.8 ± 2.9	**84.0** ± 3.1
Gaussian Noise	88.8	31.5 ± 3.2	**77.3*** ± 2.9	64.6 ± 1.9	**68.4** ± 1.2	72.0 ± 1.5	65.7 ± 1.3	74.7 ± 1.0	**74.9** ± 0.5	75.6 ± 1.3	72.8 ± 0.9	76.3 ± 1.3	73.7 ± 2.1
Shot Noise	90.0	37.9 ± 2.3	**78.1*** ± 2.0	65.0 ± 1.9	**68.3** ± 1.2	72.2 ± 0.5	68.0 ± 0.3	76.4 ± 1.2	74.7 ± 0.9	76.6 ± 0.8	74.9 ± 1.2	76.9 ± 1.2	74.8 ± 2.2
Impulse Noise	93.7	33.8 ± 2.0	**83.9*** ± 2.4	44.2 ± 1.8	**59.0** ± 2.0	63.0 ± 3.1	58.2 ± 2.6	67.6 ± 2.4	64.9 ± 3.1	66.8 ± 2.2	64.6 ± 2.4	67.9 ± 1.9	65.4 ± 1.9

Table 25: Unsupervised adaptation performance (accuracy) from CIFAR10 (clean) to CIFAR10-C dataset for each corruption (severity 5). Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

Target	Oracle	ERM	+ FMM	MCC	+ FMM	MCD	+ FMM	ADDA	+ FMM	CDAN	+ FMM	DANN	+ FMM
Contrast	69.8	51.6 ± 3.4	**74.0** ± 2.4	14.5 ± 3.2	**66.0** ± 2.6	48.7 ± 1.9	**53.8** ± 2.3	68.7 ± 1.8	**71.6** ± 2.1	61.3 ± 3.2	**74.7*** ± 2.8	71.6 ± 0.9	**72.8** ± 1.1
Elastic transform	71.9	66.3 ± 3.1	**71.2** ± 2.3	63.9 ± 1.9	62.9 ± 1.7	57.0 ± 1.5	55.6 ± 2.0	71.1 ± 1.9	**71.8** ± 1.7	74.0* ± 1.4	73.2 ± 2.0	72.6 ± 0.9	72.2 ± 1.9
Pixelate	76.9	78.4 ± 1.2	**80.8** ± 1.8	70.8 ± 1.3	**71.3** ± 1.4	65.9 ± 1.6	**66.2** ± 2.1	78.8 ± 2.2	**79.5** ± 1.0	79.9 ± 1.3	**81.3*** ± 1.4	78.5 ± 0.9	**80.8** ± 1.2
JPEG	77.8	69.8 ± 0.6	**73.7** ± 1.2	66.4 ± 1.1	**67.5** ± 0.8	61.4 ± 1.4	61.0 ± 1.1	74.6 ± 1.2	**76.6** ± 1.3	75.4 ± 0.8	**76.1** ± 1.0	76.3 ± 0.9	**76.8*** ± 0.8
Defocus blur	77.0	55.0 ± 2.3	**74.7** ± 3.1	49.4 ± 1.5	**66.4** ± 3.1	55.1 ± 1.7	**58.7** ± 1.9	70.4 ± 2.5	**76.6*** ± 1.5	69.8 ± 2.9	**76.6*** ± 1.8	71.0 ± 2.9	**74.4** ± 2.0
Glass blur	75.7	59.7 ± 1.9	**68.9** ± 2.8	58.2 ± 1.8	**67.5** ± 2.0	57.7 ± 2.0	**61.4** ± 1.8	73.2 ± 1.2	**75.1** ± 2.0	73.8 ± 1.7	**76.0** ± 1.5	75.5 ± 2.0	**77.2*** ± 1.3
Motion blur	77.2	66.2 ± 2.3	**77.2*** ± 1.8	57.7 ± 2.7	**67.4** ± 3.1	56.9 ± 1.9	**61.2** ± 1.2	74.9 ± 1.9	**76.4** ± 0.8	74.6 ± 1.0	**77.2*** ± 0.5	75.0 ± 0.6	**75.8** ± 0.6
Zoom blur	78.4	63.4 ± 2.2	**68.1** ± 1.2	60.8 ± 1.6	**66.8** ± 2.1	57.0 ± 0.8	**58.0** ± 0.6	71.3 ± 1.0	**74.9** ± 1.6	73.3 ± 1.8	**75.9*** ± 2.0	73.6 ± 1.7	**74.4** ± 2.0
Snow	72.8	46.8 ± 2.1	**57.7** ± 1.8	31.2 ± 1.0	**49.2** ± 1.2	47.0 ± 1.3	**47.2** ± 1.8	62.4 ± 1.8	**65.8** ± 1.9	56.2 ± 1.4	**64.8** ± 2.0	66.0 ± 1.8	**68.8*** ± 1.2
Frost	70.4	51.8 ± 0.9	**62.1** ± 1.2	25.5 ± 0.5	**47.3** ± 1.2	46.9 ± 1.1	**50.1** ± 1.3	62.4 ± 2.1	**67.2** ± 2.0	57.6 ± 3.0	**66.1** ± 3.1	62.7 ± 2.1	**68.2*** ± 2.2
Fog	76.0	51.8 ± 4.0	**76.9*** ± 3.6	25.3 ± 3.6	**55.7** ± 4.3	54.0 ± 2.1	**58.5** ± 1.9	67.0 ± 2.1	**72.2** ± 3.1	66.3 ± 1.9	**71.6** ± 2.5	65.2 ± 0.9	**71.2** ± 2.1
Brightness	79.2	77.9 ± 0.7	**78.5** ± 0.9	63.6 ± 1.0	**68.0** ± 1.2	65.4 ± 0.5	**65.5** ± 0.3	77.5 ± 0.8	**78.2** ± 1.0	77.2 ± 1.2	76.4 ± 2.1	79.0* ± 2.3	77.5 ± 2.0
Gaussian noise	77.2	49.0 ± 2.3	**63.3** ± 3.1	47.7 ± 1.9	**49.5** ± 2.0	54.1 ± 1.9	52.4 ± 2.3	71.9 ± 0.9	71.0 ± 2.4	70.6 ± 2.3	70.6 ± 3.0	72.0 ± 2.1	**73.6*** ± 1.8
Shot noise	76.8	44.1 ± 0.9	**54.6** ± 2.1	46.4 ± 1.2	**49.2** ± 1.3	54.7 ± 4.1	51.6 ± 3.1	72.3* ± 3.2	70.4 ± 2.3	68.5 ± 0.6	**70.2** ± 1.2	71.6 ± 2.1	71.0 ± 2.0
Impulse noise	70.0	31.8 ± 1.1	**54.9** ± 2.3	44.9 ± 1.2	44.4 ± 1.9	50.4 ± 4.1	47.0 ± 2.4	67.7 ± 1.2	**69.2** ± 1.3	68.2 ± 2.4	66.6 ± 1.2	70.5* ± 0.9	68.5 ± 1.3

Table 26: Unsupervised adaptation performance on ImageNet50-C dataset (severity 2). Results are in bold if FMM improves performance when added to baseline method. Results with * are best across all methods.

K Domain Randomization using Fourier-moments

We experimented with domain randomization (DR), which is an orthogonal approach to domain adaptation (DA). Domain randomization aims to achieve improved generalization to real-world scenarios by simulating multiple environments during training by randomizing properties such as lighting conditions, position of objects etc (Tobin et al., 2017; Sadeghi & Levine, 2017). We explored randomization of the Fourier-moments of the source domain rather than matching the statistics of the target domain. We randomized the Fourier-moments of the source data by adding varying degree of random Gaussian noise to the Fourier-moments. This approach exposes the model to different frequency statistics at training and hence, may potentially help improve its robustness in the target domain. However, we did not observe any improvement in performance on the target domain due to this procedure (Table 27), which suggests that merely randomizing frequency statistics is less beneficial than matching target domain statistics to achieve domain adaptation.

Method	Sleep-EDF	
	Acc.	F1
ERM	64.5	54.4
ERM + DR ($\mu = 0; \sigma = 1.0$)	63.7	53.5
ERM + DR ($\mu = 0; \sigma = 2.0$)	63.7	53.5
ERM + DR ($\mu = 0; \sigma = 3.0$)	63.7	53.5
ERM + DR ($\mu = 0; \sigma = 4.0$)	63.7	53.5
ERM + DR ($\mu = 0; \sigma = 5.0$)	63.7	53.5
ERM + DR ($\mu = 0; \sigma = 6.0$)	63.7	53.5
ERM + FMM	**74.1**	**61.7**

Table 27: Performance of domain randomization (DR) and FMM on Sleep-EDF (EEG). Results were averaged across all source-target domain pairs and three runs with different random seed.