## A NOTATIONS

The main notations used in the present study are introduced and defined in Table 6.

Table 6: Notations, where $\mathcal{I} := [0, 1]^{h \times w \times c}$.

| Notation | Meaning | Domain or Signature |
|---|---|---|
| $\boldsymbol{x}$ | Image | $\mathcal{I}$ |
| $y$ | Label (True class) | $\mathbb{N}$ |
| $\epsilon$ | Perturbation strength | $\mathbb{R}$ |
| $L$ | # of classes | $\mathbb{N}$ |
| $D$ | # of features | $\mathbb{N}$ |
| $f(\cdot)$ | Classifier | $\mathcal{I} \to \mathbb{N}$ |
| $\phi(\cdot)$ | Feature extractor | $\mathcal{I} \to \mathbb{R}^D$ |

## B MAX-MAHALANOBIS DISTRIBUTION

The algorithm for producing MMD centers is shown in Alg. 1. Fig. 3 illustrates some examples of produced vectors.

**Algorithm 1** MMD Center Generation Algorithm

**Input:** The constant, $C$; The dimension of vectors, $D$; The number of classes, $L$;
**Output:** The set of center vectors, $\{\boldsymbol{\mu}^{(1)}, ..., \boldsymbol{\mu}^{(L)}\}$;

$\boldsymbol{\mu}^{(1)} \leftarrow \boldsymbol{e}^{(1)}$
$\boldsymbol{\mu}^{(i)} \leftarrow \mathbf{0}^D, i \neq 1$
**for** $i \in \{1, ..., L\}$ **do**
    **for** $j \in \{1, ..., i-1\}$ **do**
        $\boldsymbol{\mu}_j^{(i)} \leftarrow -[1 + (L-1)(\boldsymbol{\mu}^{(i)} \cdot \boldsymbol{\mu}^{(j)})]/[(L-1)\boldsymbol{\mu}_j^{(j)}]$
    **end for**
    $\boldsymbol{\mu}_i^{(i)} \leftarrow \sqrt{1 - \|\boldsymbol{\mu}^{(i)}\|_2^2}$
**end for**
**for** $k \in \{1, ..., L\}$ **do**
    $\boldsymbol{\mu}^{(k)} \leftarrow C \cdot \boldsymbol{\mu}^{(k)}$
**end for**
**return** $\{\boldsymbol{\mu}^{(1)}, ..., \boldsymbol{\mu}^{(L)}\}$

where $\boldsymbol{e}^{(1)}$ and $\mathbf{0}^D$ denote the first unit basis vector and the zero vector in $\mathbb{R}^D$, respectively.

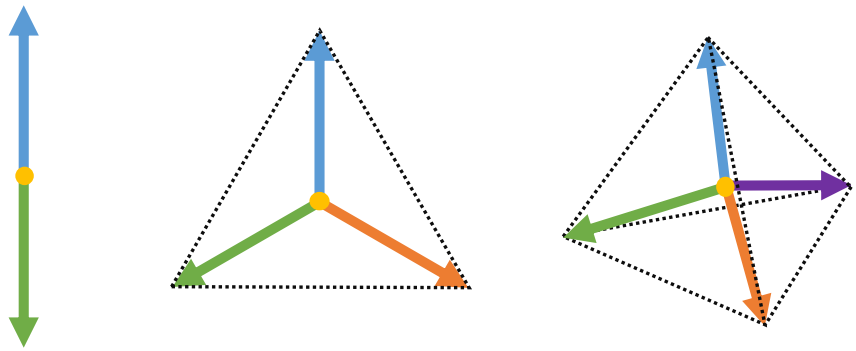

Figure 3: Examples of MMD centers for $L = 2$ (left), $L = 3$ (middle), $L = 4$ (right).

Table 7: Training hyperparameters, where $\eta^{init}$ is the initial learning rate, $\beta$ is the weight of regularization term, and $\epsilon$, # steps, and step size are the parameters of PGD used to find the worst-case examples in training.

| Dataset | Model | $\eta^{init}$ | $\beta$ | $\epsilon$ | # Steps | Step size | Optimizer |
|---|---|---|---|---|---|---|---|
| CIFAR10 | Natural | 0.1 | - | - | - | - | SGD (momentum=0.9) (weight decay=0.0002) |
| | AT | 0.1 | - | 0.031 | - | - | |
| | TRADES | 0.1 | 6.0 | 0.031 | 10 | 0.007 | |
| | MAT | 0.1 | 5.0 | 0.031 | 10 | 0.007 | |
| MNIST | Natural | 0.01 | - | - | - | - | SGD (momentum=0.9) |
| | AT | 0.005 | - | 0.03 | - | - | |
| | TRADES | 0.01 | 3.0 | 0.03 | 40 | 0.01 | |
| | MAT | 0.01 | 3.0 | 0.03 | 40 | 0.01 | |

# C Implementation Details

This section describes the implementation details, including the model structure, training hyperparameters, and attack parameters. By following the details shown in this section, one can reproduce the experiment results presented in this work.

## C.1 Adaption of MAT to Common CNN Architecture

To make fair comparison with existing adversarial defense methods, the model architectures used in this work are the same as the ones in TRADES (Zhang et al., 2019). However, for MAT, the CNN is trained to produce features rather than class scores. As in most adversarial detection works (Feinman et al., 2017; Ma et al., 2018; Lee et al., 2018), the features of traditional CNNs are defined as the output of the penultimate layer. Therefore, when applying MAT to the models used in other works, the last linear layer is removed from the model to match the definition of features. Furthermore, many common models (*e.g.*, wide ResNet (Zagoruyko & Komodakis, 2016)) adopt Rectified Linear Unit (ReLU) as the activation function of the penultimate layer, and hence the output features are all non-negative. However, the MMD centers have negative values, and thus require the features of corresponding samples to also have negative values for minimizing their cosine-distance. As a result, the ReLU function after the last feature layer is also removed for MAT.

## C.2 Training Hyperparameters

The hyperparameters for training are shown in Table 7. The batch size for all models is 128. $\alpha$ for the MAT model with the SO loss set to 0.5. $\beta$ of the TRADES model is equivalent to $1/\lambda$ in Eq. 1. For CIFAR10, the learning rate is divided by 10 at the 75th, 90th, and 100th epochs. For MNIST, the reduction of learning rate happens at the 55th, 75th, and 90th epochs. Most hyperparameters are the same as the settings in Zhang et al. (2019), while $\beta$ for MAT is tuned by experiments.

## C.3 Attack Parameters

The values of the attack parameters are shown in Table 8, which are the same as in Zhang et al. (2019) (PGD) and Lee et al. (2018) (C&W), or the default settings in `foolbox` (`https://foolbox.readthedocs.io/en/v2.3.0/modules/attacks.html`) (DDN).

# D Robustness versus Perturbation Strength

Robust accuracies vs. PGD with different perturbation strengths ($\epsilon$) are shown in Tables 9 and 10.

Table 8: Attack parameters.

| attack | parameters |
|---|---|
| CIFAR10 | |
| PGD | $\epsilon$=0.031, iterations=20, step size=0.03, p=$\infty$ |
| C&W | max iterations=100, learning rate=0.01, initial constant=1.0 |
| DDN | steps=100, gamma=0.05, initial norm=1 |
| MNIST | |
| PGD | $\epsilon$=0.3, iterations=40, step size=0.01, p=$\infty$ |
| C&W | max iterations=100, learning rate=0.01, initial constant=1.0 |
| DDN | steps=100, gamma=0.05, initial norm=1 |

Table 9: Robust accuracies vs. PGD with different perturbation strength ($\epsilon$) for the CIFAR10 dataset.

| Defense | SO | BIBO | $\epsilon = 0.011$ | $\epsilon = 0.021$ | $\epsilon = 0.031$ | $\epsilon = 0.041$ | $\epsilon = 0.051$ |
|---|---|---|---|---|---|---|---|
| Natural | - | - | 02.33% | 00.07% | 00.03% | 00.01% | 00.00% |
| TRADES | - | - | 76.55% | 69.94% | 56.61% | 49.87% | 45.58% |
| **MAT** | X | X | 07.68% | 04.63% | 03.45% | 02.62% | 02.08% |
| | X | O | 74.60% | 63.55% | 53.58% | 47.81% | 44.46% |
| | O | X | **83.22%** | **81.88%** | 79.92% | 77.19% | 73.62% |
| | O | O | 80.57% | 80.55% | **80.55%** | **80.40%** | **80.39%** |

Table 10: Robust accuracies vs. PGD with different perturbation strengths ($\epsilon$) for the MNIST dataset.

| Defense | SO | BIBO | $\epsilon = 0.1$ | $\epsilon = 0.2$ | $\epsilon = 0.3$ | $\epsilon = 0.4$ | $\epsilon = 0.5$ |
|---|---|---|---|---|---|---|---|
| Natural | - | - | 02.28% | 00.59% | 00.19% | 00.06% | 00.00% |
| TRADES | - | - | 98.31% | 96.32% | 94.38% | 75.92% | 20.97% |
| **MAT** | X | X | 98.59% | 87.02% | 61.62% | 33.72% | 16.56% |
| | X | O | **99.30%** | **99.30%** | **99.30%** | **99.25%** | **99.20%** |
| | O | X | 99.04% | 94.44% | 82.06% | 58.66% | 33.60% |
| | O | O | 99.28% | 99.28% | 99.28% | **99.25%** | 99.10% |