Normalized Wasserstein for Mixture Distributions with Applications in Adversarial Learning and Domain Adaptation: Supplementary material

1. Proof of Theorem 1

For NW measure to normalize mode proportions appropriately, we need a good estimate of the number of mode proportions. Theorem 1 provides conditions under which the mode proportions can provable be estimated.

Let $\mathbb{P}_X$ and $\mathbb{P}_Y$ be two mixture distributions whose NW measure we wish to compute. Let $\mathbb{P}_X$ and $\mathbb{P}_Y$ have n_1 and n_2 modes respectively, with r modes overlapping. Let $k^* = n_1 + n_2 - r$. We make the following assumptions

- (A1) If mode i in distribution X and mode j in distribution Y belong to the same mixture component, then their Wasserstein distance is $\leq \epsilon$ i.e., if X_i and Y_j correspond to the same component, $W(\mathbb{P}_{X_i}, \mathbb{P}_{Y_j}) < \epsilon$.
- (A2) The minimum Wasserstein distance between any two modes of one mixture distribution is at least δ i.e., $W(\mathbb{P}_{X_i}, \mathbb{P}_{X_j}) > \delta$ and $W(\mathbb{P}_{Y_i}, \mathbb{P}_{Y_j}) > \delta \ \forall i \neq j$. Also, non-overlapping modes between X and Y are separated by δ i.e., for non-overlapping modes X_i and Y_j, $W(\mathbb{P}_{X_i}, \mathbb{P}_{Y_j}) > \delta$. This ensures that modes are well-separated.
- (A3) We assume that each mode X_i and Y_i have density at least η i.e., $\mathbb{P}_{X_i} \geq \eta \quad \forall i$, $\mathbb{P}_{Y_i} \geq \eta \quad \forall i$. This ensures that every mode proportion is at least η.
- (A4) Each generator $\mathbf{G}_i$ is powerful enough to capture exactly one mode of distribution $\mathbb{P}_X$ or $\mathbb{P}_Y$.

Lemma 1 *$NW(k)$ is a monotonically decreasing function with respect to k.*

This is because in $NW(k+1)$, we add one additional mode compared to $NW(k)$. If we have $\pi^{(1)}$, $\pi^{(2)}$ for this new mode to be 0 and give the same assignements as $NW(k)$ to the rest of the modes, $NW(k+1) = NW(k)$. Since computing $NW(k)$ contains a minimization over mode assignments, the $NW(k+1) \leq NW(k) \forall k$. Hence, it is monotonically decreasing.

Lemma 2 $NW(k^*) \leq \epsilon$

This is because at $k = k^*$, we can make the following mode assignments.

- Assign $n_1 + n_2 - r$ modes of NW to each of $n_1 + n_2 - r$ non-overlapping modes in $\mathbb{P}_X$ and $\mathbb{P}_Y$ with the same mixture .
- Assign the remaining r modes of NW to the overlapping modes of either $\mathbb{P}_X$ or $\mathbb{P}_Y$. WLOG, let us assume we assign them to r overlapping modes of $\mathbb{P}_X$.
- Choose $\pi^{(1)}$ to be same as π for $\mathbb{P}_X$, with 0 to non-overlapping components of $\mathbb{P}_Y$
- Choose $\pi^{(2)}$ to be same as π for $\mathbb{P}_Y$, with 0 to non-overlapping components of $\mathbb{P}_X$

Let us denote $NOv(X)$ to be non-overlapping modes of X, $Ov(X)$ to be overlapping modes of X, $NOv(Y)$ to be non-overlapping modes of Y, and $Ov(Y)$ to be overlapping modes of Y. Then, under the mode assignments given above, $NW(k^*)$ can be evaluated as,

$$
\begin{aligned}
&W_N(\mathbb{P}_X, \mathbb{P}_Y) \\
&:= \min_{\mathbf{G}, \pi^{(1)}, \pi^{(2)}} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G}, \pi^{(1)}}) + W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G}, \pi^{(2)}}). \\
&= \sum_{i \in NOv(X)} \pi_i^X W(\mathbb{P}_{X_i}, \mathbb{P}_{X_i}) + \sum_{i \in Ov(X)} \pi_i^X W(\mathbb{P}_{X_i}, \mathbb{P}_{X_i}) + \\
&\sum_{i \in NOv(Y)} \pi_i^Y W(\mathbb{P}_{Y_i}, \mathbb{P}_{Y_i}) + \sum_{i \in Ov(Y)} \pi_i^Y W(\mathbb{P}_{Y_i}, \mathbb{P}_{X_i}) \\
&= 0 + 0 + 0 + \sum_{i \in Ov(Y)} \pi_i^Y W(\mathbb{P}_{Y_i}, \mathbb{P}_{X_i}) \\
&\leq \epsilon
\end{aligned}
$$

The last step follows from (A1) i.e., overlapping modes are separated by a Wasserstein distance of ϵ.

Lemma 3 $NW(k^* - 1) \geq \frac{\delta}{2}\eta$

By assumption (A2), we know that any two modes have separation of at least δ. In the distribution $\mathbb{P}_X + \mathbb{P}_Y$, there are $n_1 + n_2 - r$ unique cluster centers, each pair of clusters at a Wasserstein distance δ distance apart. In $NW(k^* - 1)$, generators have $n_1 + n_2 - r - 1$ modes, which is 1 less than the number of modes in $\mathbb{P}_X + \mathbb{P}_Y$. Now, let us assume that

$NW(k^*-1) < \frac{\delta}{2}\eta$. Then,

$$W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi^{(1)}}) + W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G},\pi^{(2)}}) < \frac{\delta}{2}\eta$$

Since each mode of $\mathbb{P}_X$ and $\mathbb{P}_Y$ has density at least η (by (A3)), the above condition can be satisfied only if

$$\forall i \in [n_1], \exists j \in [k^*-1] \text{ s.t. } W(\mathbb{P}_{X_i}, \mathbb{P}_{\mathbf{G}_j}) < \frac{\delta}{2} \quad (1)$$

$$\forall i \in [n_2], \exists j \in [k^*-1] \text{ s.t. } W(\mathbb{P}_{Y_i}, \mathbb{P}_{\mathbf{G}_j}) < \frac{\delta}{2} \quad (2)$$

Accounting for r mode overlap between X and Y, there will be n_1+n_2-r unique constraints in Eq. (1) and Eq. (2). Since, $\mathbf{G}$ has only k^*-1 modes, by Pigeonhole principle, there should be at least one pair (i,j) that is matched to the same $\mathbf{G}_j$. WLOG, let us consider both i and j to belong to $\mathbb{P}_X$, although each can either belong to $\mathbb{P}_X$ or $\mathbb{P}_Y$. Then,

$$W(\mathbb{P}_{X_i}, \mathbf{G}_k) < \frac{\delta}{2}$$
$$W(\mathbb{P}_{X_j}, \mathbf{G}_k) < \frac{\delta}{2}$$

Then, by triangle inequality, $W(\mathbb{P}_{X_i}, \mathbb{P}_{X_j}) < \delta$. This contradicts assumption (A2). Hence $NW(k^*-1) \geq \frac{\delta}{2}\eta$

Theorem 1 *Let $\mathbb{P}_X$ and $\mathbb{P}_Y$ be two mixture distributions satisfying (A1)-(A4) with n_1 and n_2 mixture components, respectively, where r of them are overlapping. Let $k^* = n_1 + n_2 - r$. Then, k^* is smallest k for which $NW(k)$ is small ($O(\epsilon)$) and $NW(k) - NW(k-1)$ is relatively large (in the $O(\delta\eta)$)*

Proof: From Lemma 2 and Lemma 1, we know that $NW(k) \leq \epsilon \quad \forall k \geq k^*$. Similarly, from Lemma 3 and Lemma 1, we $NW(k) \geq \frac{\delta}{2}\eta \quad \forall k < k^*$. Hence, k^* is the smallest k for which $NW(k)$ is small ($O(\epsilon)$) and $NW(k) - NW(k-1)$ is relatively large (in the $O(\delta\eta)$). Hence, proved.

2. Properties of Normalized Wasserstein measure

The defined NW measure is not a distance because it does not satisfy the properties of a distance measure.

- In general, $W_N(\mathbb{P}_X, \mathbb{P}_Y) \neq 0$. However, if $\mathbb{P}_X \in \mathbb{P}_{\mathbf{G},\pi}$, $W_N(\mathbb{P}_X, \mathbb{P}_X) = 0$. Moreover, if $\exists \mathbf{G}, \pi$ s.t. $W_N(\mathbb{P}_{\mathbf{G},\pi}, \mathbb{P}_X) < \epsilon$ (i.e., $\mathbb{P}_{\mathbf{G},\pi}$ approximates $\mathbb{P}_X$ within ϵ factor), then $W_N(\mathbb{P}_X, \mathbb{P}_X) \leq 2\epsilon$. This follows from the definition of NW measure. So, when the class of generators are powerful enough, this property is satisfied within 2ϵ approximation
- Normalized Wasserstein measure is symmetric. $W_N(\mathbb{P}_X, \mathbb{P}_Y) = W_N(\mathbb{P}_Y, \mathbb{P}_X)$
- Normalized Wasserstein measure does not satisfy triangle inequality.

3. Optimizing Normalized Wasserstein using duality

NW measure between two distributions $\mathbb{P}_X$ and $\mathbb{P}_Y$ is defined as

$$\min_{\mathbf{G},\pi^{(1)},\pi^{(2)}} W(\mathbb{P}_X, \mathbb{P}_{\mathbf{G},\pi^{(1)}}) + W(\mathbb{P}_Y, \mathbb{P}_{\mathbf{G},\pi^{(2)}})$$

Similar to [1], using the dual of Wasserstein distance, we can write the above optimization as

$$\min_{\mathbf{G},\pi^{(1)},\pi^{(2)}} \Big[\max_{D_1 \in 1-Lip} \mathbb{E}[D_1(X)] - \mathbb{E}[\sum_i \pi_i^{(1)} D_1(G_i(Z))] + \max_{D_2 \in 1-Lip} \mathbb{E}[D_2(Y)] - \mathbb{E}[\sum_i \pi_i^{(2)} D_2(G_i(Z))] \Big] \quad (3)$$

Here, D_1 and D_2 are 1-Lipschitz functions, and $\pi^{(1)}$ and $\pi^{(2)}$ are $k-$dimesional vectors lying in a simplex i.e.,

$$\sum_i \pi_i^{(1)} = 1, \quad \sum_i \pi_i^{(2)} = 1$$

To enforce these constraints, we use the softmax function as follows.

$$\pi^{(1)} = \text{softmax}(\tilde{\pi}^{(1)}), \quad \pi^{(2)} = \text{softmax}(\tilde{\pi}^{(2)})$$

The new variables $\tilde{\pi}^{(1)}$ and $\tilde{\pi}^{(2)}$ become optimization variables. The softmax function ensures that the mixture probabilities $\pi^{(1)}$ and $\pi^{(2)}$ lie in a simplex.

The above equations are optimized using alternating gradient descent given by the following algorithm.

4. Comparative analysis of mixture distributions

In this section, we propose a test using a combination of Wasserstein distance and NW measure to identify if two mixture distributions differ in mode components or mode proportions. Such a test can provide better understanding while comparing mixture distributions. Suppose $\mathbb{P}_X$ and $\mathbb{P}_Y$ are two mixture distributions with the same mixture components but different mode proportions. I.e., $\mathbb{P}_X$ and $\mathbb{P}_Y$ both belong to $\mathcal{P}_{\mathbf{G},k}$. In this case, depending on the difference between $\pi^{(1)}$ and $\pi^{(2)}$, the Wasserstein distance between the two distributions can be arbitrarily large. Thus, using the Wasserstein distance, we can only conclude that the two distributions are different. In some applications,

Algorithm 1 Optimizatizing Normalized Wasserstein

Training iterations = N_{iter}
Critic iterations = N_{critic}
for $t = 1 : N_{iter}$ **do**
  Sample minibatch $\mathbf{x} \sim \mathbb{P}_X$, $\mathbf{y} \sim \mathbb{P}_Y$
  Sample minibatch $\mathbf{z} \sim \mathcal{N}(0, 1)$
  Compute Normalized Wasserstein as

$$NW = \mathbb{E}[D_1(\mathbf{x})] - \mathbb{E}[\sum_i \pi_i^{(1)} D_1(G_i(\mathbf{z}))] + \mathbb{E}[D_2(\mathbf{y})] - \mathbb{E}[\sum_i \pi_i^{(2)} D_2(G_i(\mathbf{z}))]$$

  for $k = 1 : N_{critic}$ **do**
    Maximize NW w.r.t D_1 and D_2
    Minimize NW w.r.t $\tilde{\pi}^{(1)}$ and $\tilde{\pi}^{(2)}$
  end for
  Minimize NW w.r.t G
end for

it can be informative to have a test that determines if two distributions differ only in mode proportions. We propose a test based on a combination of Wasserstein and the NW measure for this task. This procedure is shown in Table. 1. We note that computation of p-values for the proposed test is beyond the scope of this paper.

We demonstrate this test on 2D Mixture of Gaussians. We perform experiments on two settings, each involving two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, which are mixtures of 8 Gaussians:

Setting 1: Both $\mathcal{D}_1$ and $\mathcal{D}_2$ have same mode components, with the i^{th} mode located at $(r\cos(\frac{2\pi i}{8}), r\sin(\frac{2\pi i}{8}))$.

Setting 2: $\mathcal{D}_1$ and $\mathcal{D}_2$ have shifted mode components. The i^{th} mode of $\mathcal{D}_1$ is located at $(r\cos(\frac{2\pi i}{8}), r\sin(\frac{2\pi i}{8}))$, while the i^{th} mode of $\mathcal{D}_2$ is located at $(r\cos(\frac{2\pi i+\pi}{8}), r\sin(\frac{2\pi i+\pi}{8}))$.

In both the settings, the mode fraction of $\mathcal{D}_1$ is $\pi_i = \frac{i+2}{52}$, and that of $\mathcal{D}_2$ is $\pi_i = \frac{11-i}{52}$. We use $2,000$ data points from $\mathcal{D}_1$ and $\mathcal{D}_2$ to compute Wasserstein distance and the NW measure in primal form by solving a linear program. The computed distance values are reported in Table 2. In setting 1, we observe that the Wasserstein distance is large while the NW measure is small. Thus, one can conclude that the two distributions differ only in mode proportions. In setting 2, both Wasserstein and NW measures are large. Thus, in this case, distributions differ in mixture components as well.

5. Additional results

5.1. CIFAR-10

We present the results of training NWGAN on CIFAR-10 dataset. We use WGAN-GP [3] with Resnet-based generator and discriminator models as our baseline method. The proposed NWGAN was trained with $k = 4$ modes using the same network architectures as the baseline. Sample generations produced by each mode of the NWGAN is shown in Figure 1. We observe that each generator model captures distinct variations of the entire dataset, thereby approximately disentangling different modes in input images. For quantitative evaluation, we compute inception scores for the baseline and the proposed NWGAN. The inception score for the baseline model is 7.56, whereas our model achieved an improved score of 7.89.

5.2. Domain adaptation under uniform mode proportions

In this section, we present results on domain adaptation on mode-balanced VISDA dataset – source and target domains contain 3 classes - *aeroplane*, *horse* and *truck* with uniform mode proportion. The results of performing adaptation using NW measure in comparison with classical distance measures are reported in Table 4. We observe that NW measure performs on-par with the compared methods on this dataset. This experiment demonstrates the effectiveness of NW measure on a range of settings – when the source and target datasets are balanced in mode proportions, NW becomes equivalent to Wasserstein distance and minimizing it is no worse than minimizing the classical distance measures. On the other hand, when mode proportions of source and target domains differ, NW measure renormalizes the mode proportions and effectively performs domain adaptation. This illustrates the usefulness of NW measure in domain adaptation problems.

5.3. Adversarial clustering: Quantitative metrics

- Cluster purity: Cluster purity measures the extent to which clusters are consistent i.e., if each cluster constains similar points or not. To compute the cluster purity, the cardinality of the majority class is computed for each cluster, and summed over and divided by the total number of samples.
- ARI - Adjusted Rand Index: The rand index computes the similarity measure between two clusters by considering all pairs of samples, and counting the pairs of samples having the same cluster in the ground-truth and predicted cluster assignments. Adjusted rand index makes sure that ARI score is in the range (0, 1)
- NMI - Normalized Mutual Information: NMI is the normalized version of the mutual information between the predicted and the ground truth cluster assignments.

5.4. Adversarial clustering of CIFAR+CelebA

In this section, we show the results of performing adversarial clustering on a mixture of CIFAR-10 and CelebA datasets. The same dataset presented in Section 3.2 of the

Table 1. Comparative analysis of two mixture distributions

Wasserstein distance	NW measure	Conclusion
High	High	Distributions differ in mode components
High	Low	Distributions have the same components, but differ in mode proportions
Low	Low	Distributions are the same

Figure 1. Sample generations produced by the proposed NWGAN trained on CIFAR-10 with $k = 4$ generator modes.

Table 2. Hypothesis test between two MOG - $\mathcal{D}_1$ and $\mathcal{D}_2$

Setting	Wasserstein Distance	NW measure
Setting 1	1.51	0.06
Setting 2	1.56	0.44

main paper is used in this experiment (i.e) the dataset contains CIFAR-10 and CelebA samples in $1 : 2$ mode proportion. NWGAN was trained with 2 modes - each employing Resnet based generator-discriminator architectures (same architectures and hyper-parameters used in Section 3.2 of main paper). Quantitative evaluation of our approach in comparison with $k - means$ is given in Table 5. We observe that our approach outperforms $k - means$ clustering. However, the clustering quality is poorer that the one obtained on imbalanced MNIST dataset. This is because the samples generated on MNIST dataset had much better quality than the one produced on CIFAR-10. So, as long as the underlying GAN model produces good generations, our adversarial clustering algorithm performs well.

6. Architecture and hyper-parameters

Implementation details including model architectures and hyperparameters are presented in this section:

6.1. Mixture models for Generative Adversarial Networks (GANs)

6.1.1 Mixture of Gaussians

As discussed in Section 3.1 of the main paper, the input dataset is a mixture of 8 Gaussians with varying mode proportion. Normalized Wasserstein GAN was trained with linear generator and non-linear discriminator models using the architectures and hyper-parameters as presented in Table 6. The architecture used for training Vanilla WGAN is provided in Table 7. The same architecture is used for MGAN, however we do not use the $ReLU$ non-linearities in the Generator function (to make the generator affine so that the model is comparable to ours). For WGAN and MGAN, we use the hyper-parameter details as provided in the respective papers – [3] and [4].

6.1.2 CIFAR-10 + CelebA

To train models on CIFAR-10 + CelebA dataset (Section 3.2 of the main paper), we used the Resnet architectures of WGAN-GP [3] with the same hyper-parameter configuration for the generator and the discriminator networks. In Normalized WGAN, the learning rate of mode proportion π was 5 times the learning rate of the discriminator.

6.2. Domain adaptation for mixture distributions

6.2.1 Digit classification

For MNIST→MNIST-M experiments (Section 4.1.1 of the main paper), following [2], a modified Lenet architecture was used for feature network, and a MLP network was used for domain classifier. The architectures and hyper-parameters used in our method are given in Table 8. The same architectures are used for the compared approaches - Source only, DANN and Wasserstein.

Table 3. MNIST → MNIST-M settings

Config	3 modes	5 modes	10 modes
Classes	$\{1, 4, 8\}$	$\{0, 2, 4, 6, 8\}$	$\{0, 1, \ldots 9\}$
Proportion of source samples	$\{0.63, 0.31, 0.06\}$	$\{0.33, 0.26, 0.2, 0.13, 0.06\}$	$\{0.15, 0.15, 0.15, 0.12, 0.12$ $0.11, 0.05, 0.05, 0.05, 0.05\}$
Proportion of target samples	$\{0.06, 0.31, 0.63\}$	$\{0.06, 0.13, 0.2, 0.26, 0.33\}$	$\{0.05, 0.05, 0.05, 0.05, 0.11$ $0.12, 0.12, 0.15, 0.15, 0.15\}$

Table 4. Domain adaptation on mode-balanced datasets: VISDA. Average classification accuracies averaged over 5 runs are reported

Method	Classification accuracy (in %)
Source only	63.24
DANN	84.71
Wasserstein	90.08
Normalized Wasserstein	90.72

Table 5. Performance of clustering algorithms on CIFAR+CelebA dataset

Method	Cluster Purity	NMI	ARI
k-means	0.667	0.038	0.049
Normalized Wasserstein	0.870	0.505	0.547

6.2.2 VISDA

For the experiments on VISDA dataset with three classes (Section 4.1.2 of the main paper), the architectures and hyper-parameters used in our method are given in Table 9. The same architectures are used for the compared approaches: source only, Wasserstein and DANN.

6.2.3 Domain adaptation for Image denoising

The architectures and hyper-parameters used in our method for image denoising experiment (Section 4.2 of the main paper) are presented in Table 10. To perform adaptation using Normalized Wasserstein measure, we need to train the intermediate distributions $\mathbb{P}_{\mathbf{G},\pi^{(1)}}$ and $\mathbb{P}_{\mathbf{G},\pi^{(2)}}$ (as discussed in Section 2, 4.2 of the main paper). We denote the generator and discriminator models corresponding to $\mathbb{P}_{\mathbf{G},\pi^{(1)}}$ and $\mathbb{P}_{\mathbf{G},\pi^{(2)}}$ as Generator (RW) and Discriminator (RW) respectively. In practice, we noticed that the Generator (RW) and Discriminator (RW) models need to be trained for a certain number of iterations first (which we call initial iterations) before performing adaptation. So, for these initial iterations, we set the adaptation parameter λ as 0. Note that the encoder, decoder, generator (RW) and discriminator (RW) models are trained during this phase, but the adaptation is not performed. After these initial iterations, we turn the adaptation term on. The hyperparameters and model architectures are given in Table 10. The same architectures are used for Source only and Wasserstein.

6.3. Adversarial clustering

For adversarial clustering in imbalanced MNIST dataset (Section 5 of the main paper), the architectures and hyper-parameters used are given in Table 11.

6.4. Hypothesis testing

For hypothesis testing experiment (Section 6 of the main paper), the same model architectures and hyper-parameters as the MOG experiment (Table 6) was used.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017. 2

[2] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR. 4

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of Wasserstein GANs. *arXiv preprint arXiv:1704.00028*, 2017. 3, 4

[4] Quan Hoang, Tu Dinh Nguyen, Trung Le, and Dinh Phung. MGAN: training generative adversarial nets with multiple generators. 2018. 4

Table 6. Architectures and hyper-parameters: Mixture of Gaussians with Normalized Wasserstein GAN

Generator	Discriminator
Linear($2 \to 64$)	Linear($2 \to 128$)
Linear($64 \to 64$)	LeakyReLU(0.2)
Linear($64 \to 64$)	Linear($128 \to 128$)
Linear($64 \to 2$)	LeakyReLU(0.2)
	Linear($128 \to 2$)
Hyperparameters	
Discriminator learning rate	0.00005
Generator learning rate	0.00005
π learning rate	0.01
Batch size	1024
Optimizer	RMSProp
Number of critic iters	10
Weight clip	$[-0.003, 0.003]$

Table 7. Architectures: Mixture of Gaussians with vanilla WGAN model

Generator	Discriminator
Linear($2 \to 512$) + ReLU	Linear($2 \to 512$) + ReLU
Linear($512 \to 512$) + ReLU	Linear($512 \to 512$) + ReLU
Linear($512 \to 512$) + ReLU	Linear($512 \to 512$) + ReLU
Linear($512 \to 2$)	Linear($512 \to 2$)

Table 8. Architectures and hyper-parameters: Domain adaptation for MNIST→MNIST-M experiments

Feature network	
Conv($3 \to 32$, 5×5 kernel) + ReLU + MaxPool(2)	
Conv($32 \to 48$, 5×5 kernel) + ReLU + MaxPool(2)	
Domain discriminator	Classifier
Linear($768 \to 100$) + ReLU	Linear($768 \to 100$) + ReLU
Linear($100 \to 1$)	Linear($100 \to 100$) + ReLU
	Linear($100 \to 10$)
Hyperparameters	
Feature network learning rate	0.0002
Discriminator learning rate	0.0002
Classifier learning rate	0.0002
π learning rate	0.0005
Batch size	128
Optimizer	Adam
Number of critic iters	10
Weight clipping value	$[-0.01, 0.01]$
λ	1

Table 9. Architectures and hyper-parameters: Domain adaptation on VISDA dataset

Feature network	
Resnet-18 model pretrained on ImageNet till the penultimate layer	
Domain discriminator	Classifier
Linear($512 \to 512$) + LeakyReLU(0.2) Linear($512 \to 512$) + LeakyReLU(0.2) Linear($512 \to 512$) + LeakyReLU(0.2) Linear($512 \to 1$)	Linear($512 \to 3$)
Hyperparameters	
Feature network learning rate	0.000001
Discriminator learning rate	0.00001
Classifier learning rate	0.00001
π learning rate	0.0001
Batch size	128
Optimizer	Adam
Number of critic iters	10
Weight clipping value	$[-0.01, 0.01]$
λ	1

Table 10. Architectures and hyper-parameters: Domain adaptation for image denoising experiment

Encoder	Decoder
Conv($3 \to 64$, 3×3 kernel) +ReLU + MaxPool(2) Conv($64 \to 128$, 3×3 kernel) +ReLU + MaxPool(2) Conv($128 \to 128$, 3×3 kernel) +ReLU + MaxPool(2) Conv($128 \to 128$, 3×3 kernel) Linear($128 \to 2$)	Linear($2 \to 128$) Conv($128 \to 64$, 3×3 kernel) + ReLU + Upsample(2) Conv($64 \to 64$, 4×4 kernel) + ReLU + Upsample(4) Conv($64 \to 3$, 3×3 kernel)
Domain discriminator	
Linear($2 \to 64$) + ReLU Linear($64 \to 64$) + ReLU Linear($64 \to 1$)	
Generator (RW)	Discriminator (RW)
Linear($2 \to 128$) Linear($128 \to 128$) Linear($128 \to 2$)	Linear($2 \to 128$) + ReLU Linear($128 \to 128$) + ReLU Linear($128 \to 1$)
Hyperparameters	
Encoder learning rate	0.0002
Decoder learning rate	0.0002
Domain Discriminator learning rate	0.0002
Generator (RW) learning rate	0.0002
Discriminator (RW) learning rate	0.0002
π learning rate	0.0005
Batch size	128
Optimizer	Adam
Number of critic iters	5
Initial iters	5000
Weight clipping value	$[-0.01, 0.01]$
λ	1

Table 11. Architectures and hyper-parameters: Mixture models on imbalanced-MNIST3 dataset

Generator	Discriminator
ConvTranspose($100 \to 256$, 4×4 kernel, stride 1)	Spectralnorm(Conv($1 \to 64$, 4×4 kernel, stride 2))
Batchnorm + ReLU	LeakyReLU(0.2)
ConvTranspose($256 \to 128$, 4×4 kernel, stride 2)	Spectralnorm(Conv($64 \to 128$, 4×4 kernel, stride 2))
Batchnorm + ReLU	LeakyReLU(0.2)
ConvTranspose($128 \to 64$, 4×4 kernel, stride 2)	Spectralnorm(Conv($128 \to 256$, 4×4 kernel, stride 2))
Batchnorm + ReLU	LeakyReLU(0.2)
ConvTranspose($64 \to 1$, 4×4 kernel, stride 2)	Spectralnorm(Conv($256 \to 1$, 4×4 kernel, stride 1))
Tanh()	
Hyperparameters	
Discriminator learning rate	0.00005
Generator learning rate	0.0001
π learning rate	0.001
Batch size	64
Optimizer	RMSProp
Number of critic iters	5
Weight clip	$[-0.01, 0.01]$
λ_{reg}	0.01