Supplementary Material: *MEnsA*: Mix-up Ensemble Average for Unsupervised Multi Target Domain Adaptation on 3D Point Clouds

A. PointDA-10 Dataset

The PointDA-10 dataset was proposed for cross-domain 3D objects classification on point clouds [36]. It has been used as a general benchmark for single target domain adaptation (STDA) in literature [1, 19, 36]. It consists of subsets of three widely used point cloud datasets: ShapeNet [4], ScanNet [7] and ModelNet [45]. All three subsets, *i.e*., ModelNet-10 (M), ScanNet-10 (S*) and ShapeNet (S), share ten common categories (*e.g*., chair, table and monitor) across them. We present sample point clouds and statistics of the dataset in Fig 4 and Table 3, respectively.

The dataset is highly class-imbalanced; ModelNet-10 has around 124 Lamp samples while ScanNet-10 and ShapeNet-10 have 161 and 1,620 Lamp samples respectively in the training set. This causes additional difficulty to adapt to the target domains.

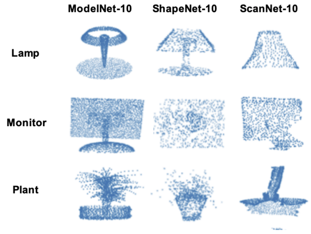

Figure 4. Samples from the PointDA-10 dataset

B. Results in STDA set-up

In Table 1 of the main paper, we compare our method to the methods that are proposed both for STDA setup and for MTDA setup. For the methods for the STDA setup (*e.g*., DANN, MCD, MMD, ADDA and PointDAN [12, 26, 36, 37, 42]), we implement them either from the scratch when there is no publicly available code repositories (*e.g*., DANN, MCD and MMD) or modify the authors' code if there is any (*e.g*., ADDA and PointDAN). For the methods proposed in the MTDA setup (*e.g*., MT-MTDA and AMEAN), we use the authors' implementation.

Here, we validate our implementation of the STDA methods by reproducing the results in their original STDA setup. Specifically, we compare the accuracy of our own implementation to the reported accuracy in the literature in Table 4 (please compare the first and the second row in each block). We observe that our own implementations successfully reproduce the results of the STDA methods in the STDA set-up; in many methods (*e.g*., MMD, DANN and MCD), our implementation improves the accuracy by a noticeable margin (+10.78% in average performance of MMD [26], +11.12% for MCD [37], and +6.93% for DANN [12]). We attribute the improvements on MMD to the choice of kernels and variance values. For the MCD, we use additional data augmentations such as jittering, orientation and *etc*. and they improve the performance by better maximizing the discrepancy between the source and target domains. For the DANN, we attribute the improvement to better selection of scalar multiplier used in reversing the gradients since the aforementioned details were not explicitly mentioned in the paper [19]. Our implementation of PointDAN exhibits comparable performance due to lack of rigorous fine-tuning that the authors might have had employed for selecting the weights of the loss components.

However, we observe a significant decrease in average accuracy for ADDA [42]. In S* → S, our implementation exhibits +10.56% and in M → S* and S* → M, it exhibits small gains. But there is a significant drop for M → S, S → M and S → S*. Note that the ADDA method uses pre-training on source domain data to obtain an initialization for learning a domain adapting classifier. Although the size and the type of the source data for the pre-training affect the domain adaptation performance, it is not well described in the paper [19]. We used ModelNet-10 for pre-training to reproduce the results of ADDA on PointDA-10. However, the performance drop, we believe, is due to the pre-training phase.

We also show the accuracy of our implementation in MTDA setup for easy side-by-side comparison to the STDA setup (please compare the second row to third row in each block). Note that the results in the third rows are the ones we have reported in Table 1 in the main paper. The drop in performance of the methods on moving from STDA to MTDA setup highlights the difficulty of adapting on multiple unlabelled targets using a single source domain. We believe that the difficulty comes from the fact that the model has to adapt to the different target domains in a single phase [30], hence, due to the limited capacity, if it performs well on one target but it does not perform well on the other targets. The performance gain in overall accuracy while reproducing the aforementioned methods partly validates our implementation for MTDA setup to be credible.

C. Ablation Study

In Table 5, we ablate $\mathcal{L}_{adv}$ in Eq. 5, which is a linear combination of a domain confusion loss ($\mathcal{L}_{dc}$), an MMD based discrepancy loss ($\mathcal{L}_{mmd}$) and a mixup loss ($\mathcal{L}_{mix}$), for detailed analysis for the contribution of each module, *i.e*., GRL, MMD, and Domain Mixup module, toward increase in overall accuracy of the proposed method (Fig. 2 of the main paper) in comparison to prior works. Adversarial domain confusion is implemented using the Gradient Reversal Layer (GRL) [12]. The contribution of GRL is

Table 3. Number of samples in PointDA-10 dataset

Dataset		Bathtub	Bed	Bookshelf	Cabinet	Chair	Lamp	Monitor	Plant	Sofa	Table	Total
M	Train	106	515	572	200	889	124	465	240	680	392	4183
	Test	50	100	100	86	100	20	100	100	100	100	856
S	Train	599	167	310	1076	4612	1620	762	158	2198	5876	17378
	Test	85	23	50	126	662	232	112	30	330	842	2492
S*	Train	98	329	464	650	2578	161	210	88	495	1037	6110
	Test	26	85	146	149	801	41	61	25	134	301	1769

Table 4. Quantitative classification accuracy (%) on PointDA-10 dataset in STDA set-up reproduced by us and reported from literature [19]. No adaptation refers to the model trained only by source samples. The results of the respective methods in the MTDA setup are also reported alongside

Src → Tgt	M → S	M →S*	S* → M	S* → S	S → M	S → S*	Average
No adaptation [19]	42.50	22.30	39.90	23.50	34.20	46.90	34.93
↪ Reproduced	45.52	30.79	54.90	31.37	37.26	44.50	40.72
↪ in MTDA setup	35.07	11.75	52.61	29.45	33.65	11.05	28.93
MMD [26]	57.50	27.90	40.70	26.70	47.30	54.80	42.50
↪ Reproduced	59.34	55.70	58.37	53.49	47.92	44.88	53.28
↪ in MTDA setup	57.16	22.68	55.40	28.24	36.77	24.88	37.52
DANN [12]	58.70	29.40	42.30	30.50	48.10	56.70	44.20
↪ Reproduced	50.65	54.27	54.19	52.00	48.11	47.53	51.13
↪ in MTDA setup	55.03	21.64	54.79	37.37	42.54	33.78	40.86
ADDA [42]	61.00	30.50	40.40	29.30	48.90	51.10	43.50
↪ Reproduced	35.64	33.90	40.93	39.86	27.15	32.49	34.88
↪ in MTDA setup	29.39	38.46	46.89	20.79	35.33	24.94	32.63
MCD [37]	62.00	31.00	41.40	31.30	46.80	59.30	45.30
↪ Reproduced	62.27	61.21	54.25	57.59	49.76	53.46	56.42
↪ in MTDA setup	57.56	27.37	54.11	41.71	42.30	22.39	40.94
PointDAN [36]	62.50	31.20	41.50	31.50	46.90	59.30	45.50
↪ Reproduced	57.57	30.63	51.80	58.10	51.68	25.06	45.81
↪ in MTDA setup	30.19	44.26	43.17	14.30	26.44	28.92	31.21

Table 5. **Ablation**. Quantitative classification accuracy (%) on the contribution of each module in $\mathcal{L}_{adv}$ as per Eq. 5 (in main paper) towards the overall pipeline (Fig. 2 in main paper) in MTDA setting. Best results are in **bold** and second best in $\underline{\text{underline}}$

Source Domain	ModelNet (M)		ScanNet (S*)		ShapeNet (S)		
Loss Terms (Eq. 5)	M → S*	M →S	S* → M	S* → S	S→M	S→S*	Average
$\mathcal{L}_{dc}$	34.42	45.08	32.81	13.32	23.55	**38.13**	31.22
$\mathcal{L}_{mmd}$	43.37	36.05	51.87	29.20	$\underline{30.67}$	25.75	36.15
$\mathcal{L}_{mix}$	32.67	43.51	**57.88**	$\underline{33.17}$	30.52	31.59	38.22
$\mathcal{L}_{dc}$ + $\mathcal{L}_{mmd}$	41.05	41.78	42.67	19.83	29.08	$\underline{33.62}$	34.67
$\mathcal{L}_{dc}$ + $\mathcal{L}_{mix}$	35.07	45.19	35.29	16.34	22.59	26.79	30.21
$\mathcal{L}_{mmd}$ + $\mathcal{L}_{mix}$	$\underline{43.47}$	$\underline{53.17}$	55.95	30.04	28.60	30.40	$\underline{40.27}$
$\mathcal{L}_{dc}$ + $\mathcal{L}_{mix}$ + $\mathcal{L}_{mmd}$	**45.31**	**61.36**	$\underline{56.67}$	**46.63**	**37.02**	27.19	**45.70**

Table 6. Class-wise classification accuracy (%) on ModelNet to ScanNet in MTDA setting. 'No adaptation' refers to the model trained only on Source samples and 'Supervised' denotes the model trained with labelled target data.

Method	Bathtub	Bed	Bookshelf	Cabinet	Chair	Lamp	Monitor	Plant	Sofa	Table	Average
No adaptation (Baseline)	40.49	21.95	12.58	6.80	11.11	46.58	51.86	56.00	65.74	46.46	35.96
MMD	55.75	9.75	18.81	0.68	37.54	30.76	46.94	52.00	77.87	75.82	40.59
ADDA	58.71	15.40	23.28	2.68	32.87	50.07	32.95	48.00	61.53	56.6	38.21
DANN	60.42	15.85	24.47	2.72	24.77	12.82	52.03	68.00	65.75	78.42	40.53
MCD	58.72	10.97	27.97	0.68	30.01	12.82	60.33	56.00	82.59	66.06	40.62
AMEAN	58.40	19.05	17.12	7.52	45.17	36.58	54.75	40.00	84.61	72.30	43.55
MTDA-ITA	67.90	11.90	4.11	20.19	21.8	12.19	56.39	45.00	85.38	83.25	40.81
MT-MTDA	59.23	5.88	24.66	4.69	32.08	14.63	66.55	48.00	78.21	72.66	40.66
MEnsA (Ours)	67.11	6.58	6.77	44.89	74.09	46.05	87.92	64.55	50.00	74.47	52.24
Supervised in each domain	91.10	69.51	61.05	89.23	99.67	80.76	91.57	51.37	94.08	81.97	81.03

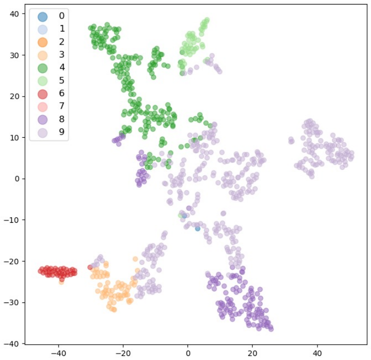

Figure 5. t-SNE embedding with perplexity 25 of the proposed method MEnsA on adapting from ShapeNet to ModelNet. Some classes which have distinct shapes are well clustered together. However, some classes with similar geometric structures such as Lamps and Tables, Beds and Sofas, *etc.,* are closer in the cluster.

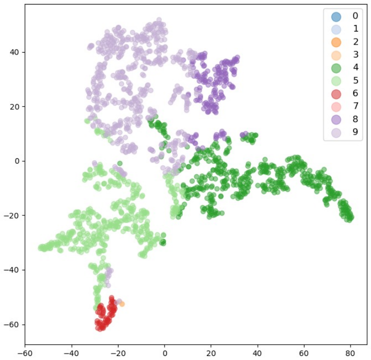

Figure 6. t-SNE embedding with perplexity 25 of the proposed method MEnsA on adapting from ShapeNet to Scannet. Here, sim-to-real adaptation is challenging, and the cluster boundaries are not distinct for objects with similar geometric properties.

measured by $\mathcal{L}_{dc}$. The GRL helps the model build feature representation of the raw input $\mathcal{X}$ that is good to predict the correct object label $\mathcal{Y}$ subject to the domain label of $\mathcal{X}$ to be not easily deduced by it. This promotes domain confusion where the feature extractor (*i.e.*, generator) tries to confuse the domain classifier (*i.e.*, discriminator) by bridging the two distributions closer. The mapping between the source and target domains is learned via MMD loss, *i.e.*, $\mathcal{L}_{mmd}$. $\mathcal{L}_{mix}$ controls the flow of information from the proposed domain mixup module.

It is clearly observed from Table 5 that the mixup module helps in improving the average classification accuracy as well as accuracy over each domain. Please note that the proposed approach performs the best when all the three modules are combined coherently as per Eq. 5.

In addition, we conduct a detailed class-wise accuracy analysis across three domains in the PointDA-10 dataset in Table 6. We observe decent gains by our method in most of classes over the prior works but not significant gains for *Bed*, *Bookshelf* and *Sofa* classes. We believe that the low performance on these classes is due to the fact that the model may neglect the 'scale' information; when different classes share very similar local structures, the model possibly aligns similar structures across these classes (*e.g.*, large columns contained both by *Lamps* and round *Tables*, small legs in *Beds* and *Sofas* or large cuboidal spaces present in *Beds* and *Bookshelves*) and leads to classification confusion.