A Bayesian Approach to OOD Robustness in Image Classification Supplementary

Contents

List of Tables

List of Figures

List of Abbreviations

Avg.	Average
plane	Aeroplane
Occ.-OOD-CV	Occluded-OOD-CV
OOD-CV(L1)	Occluded-OOD-CV (20-40% occlusion)
OOD-CV(L2)	Occluded-OOD-CV (40-60% occlusion)
OOD-CV(L3)	Occluded-OOD-CV (60-80% occlusion)
$\mathcal{A}$	Spatial Coefficient
Λ	von Mises-Fisher (feature distribution) dictionary
$\mathcal{S}$	Source
$\mathcal{R}$	Transitional
$\mathcal{T}$	Target
UGT	Unsupervised Generative Transition
lr	Learning rate
w.d	Weight decay
κ	vMF concentration parameter

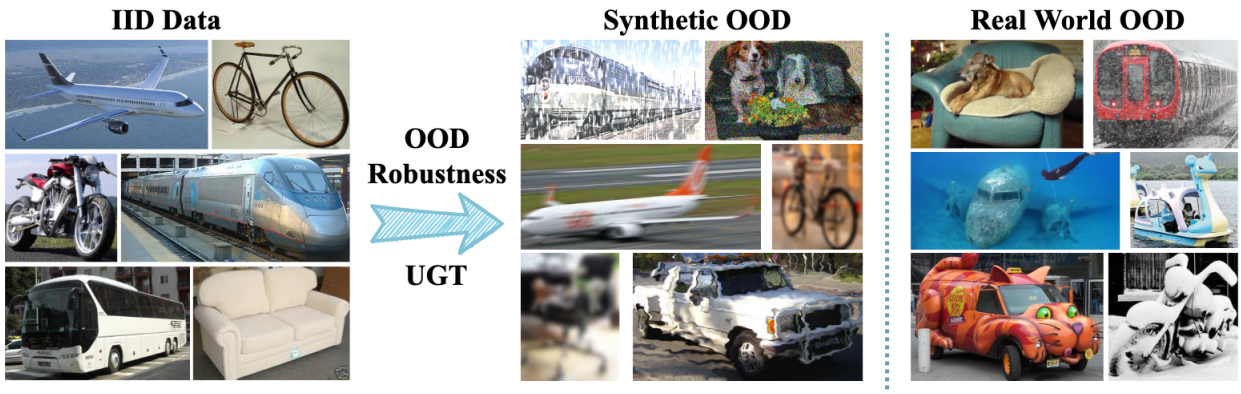

Figure 1. Out of Domain Robustness for Object Classification using Unsupervised Generative Transition (UGT)

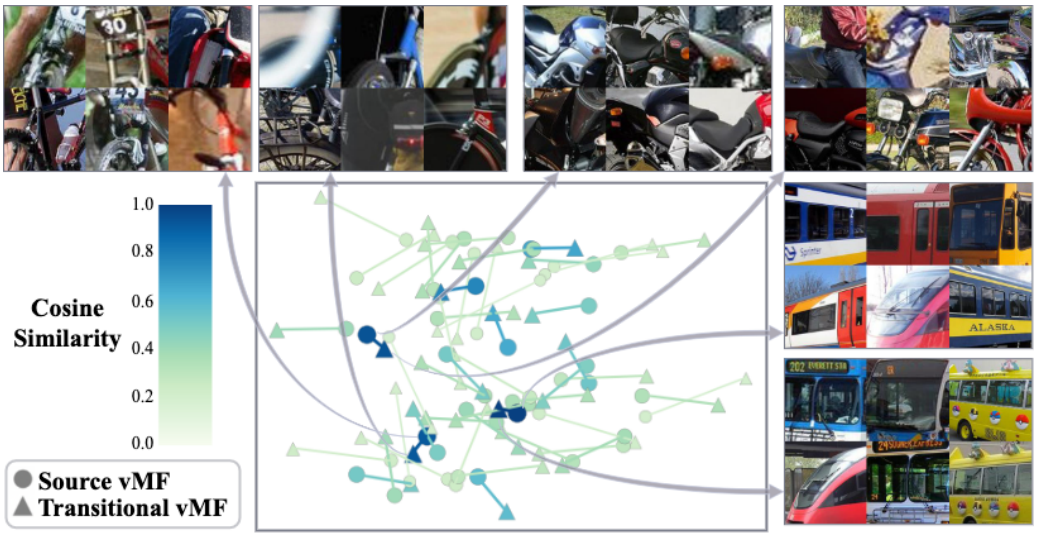

Figure 2. Source and Transitional vMF dictionary Similarity

(a) The cosine similarity between source $\Lambda^{\mathcal{S}}$ and transitional vMF dictionary $\Lambda^{\mathcal{R}}$ vectors (which are represented as circles and triangles respectively) in this conceptual vMF dictionary feature space is represented by the line connecting the circles and pentagons. The image patches from the source and target domains roughly corresponding to these vMF dictionary vectors are shown, which confirm that some similar image parts are represented by similar vMF dictionary vectors in both domains irrespective of the nuisance factor in the target domain. E.g. (lower right) image patches show windows from different vehicles - parts of objects which do not undergo much change when encountering nuisance factors like change in texture, shape and context of the vehicles.

A. FAQs

a. What are the differences/commonalities with Domain Adaptation and Domain Generalization w.r.t our domain robustness setup?
The problem of OOD robustness can be considered a niche subset of the larger unsupervised domain adaptation problem, and is closely related to domain generalisation and transfer learning as well. In unsupervised domain adaptation[21], we often try to mitigate dataset biases between labelled source and unlabelled target data - there are often no constraints on type of domains and biases. The goal of domain generalisation is to learn a model, often from multiple related source domains, such that the resulting model can generalize well to any unseen target domain [25]. Although related to both, our work focuses on an (extended) OOD robustness setup, where our aim is to generalize well to an unlabelled target domain which is parameterized by real world nuisance factors like weather, shape, pose, texture changes and partial occlusion - which often leads to drastic changes to visual scenes and objects in question not found in the source dataset.

b. Difference in results from the original OOD-CV[24] paper.
Differences, if any, in our experimental results may be attributed to different versions of same dataset used. In case of OOD-CV dataset[24], we contacted the authors and used their recommended version of the dataset. Additionally, we try not to use data augmentations and additional data to train our models which may also contribute to the difference. This is done to provide a fairer comparison between different methodologies.

c. What motivates our choice of datasets and models?
Our choice is motivated by the presence of real world nuisances and partial occlusion in the dataset. We chose datasets where either we could create partial occlusion due to presence of object masks or there was already a subset of the data which had partial occlusion. Our choice of datasets is also constrained by the amount of computational resources we use. As can be seen, we conduct a large number of experiments per model/architecture/dataset. For comparative models, we focus on works which have shown to work on previous robustness[15, 17] benchmarks and some well-known UDA works[3, 6, 14, 16, 22]. Secondary to out choice of real world nuisance factors and partial occlusion, we also test our method on Synthetic to Real transfer and synthetic corruptions in order to check its efficacy in these weel known setups.

d. Applied partial occlusion is artificially applied on top of objects in images.
Although the occluded data that we use for evaluation for all our setups does have some naturally occluded images, we use real cropped occluders to ape partial occlusion in the real world. This might not be the best solution however it is the best available since there is a dearth of occlusion data for current vision models. Additionally, earlier works [8, 19] have shown that partial occlusion analysis for many computer vision tasks with such artificially placed occluders is similar to images with real partial occlusion.

e. Can we sample from our generative model?
Our generative model defines a generative model on the level of intermediate neural features and not at the image level. Therefore, it is quite hard to visualize the sampled features.

f. Is dictionary space misalignment a problem?
We empirically show via our SOTA results that dictionary space misalignment is not a big issue for our method. This is likely because the appearance of object parts may be highly variable, while the spatial geometry of the objects remains rather similar.

g. Is the target domain data accessible to models while training?
We follow the setup of OOD-CV[24] paper. For Imagenet-C[5] and UDAParts[13], the target domain data available for finetuning is separate from the evaluation data.

h. Rough intuition of our work.
in this work, we exploit a hitherto unexploited property of CompNets which is that their vMF kernels can be learnt without supervision although learning the generative head requires supervision. This enables us to learn a *transitional vMF kernel dictionary* which contains properties of both the source and target domains. We train the generative model for the CompNets on the annotated source data, using the transitional vMF kernels, and show that this model performs well on the target domain and improves with pseudo-labeling. A crude intuition for this approach is that the vMF kernels capture the visual appearance of parts which can differ between the source and target domain. We observe that some vMF kernels are very similar in both domains and, for example, vehicle windows largely remain unchanged even when the context, texture and shape of the vehicle changes (Figure 2a). This motivates us to learn a *transitional* dictionary which is learnt on the target domain but where the feature clusters are encouraged to be similar to those on the source domain. By contrast the parameters of the generative model capture the spatial structure of the object (e.g., which parts are likely to appear in specific locations), hence is fairly similar in both source and target domain, and be learnt using only supervision on the source domain.

i. Explanations regarding Equation 1-10.
Equation 1 is borrowed directly from earlier works[10]

where $P(m)$ is a uniform prior over the spatial mixture components and is written for completeness of the Bayesian parameterization.

In Equation 2 More mixtures may lead to better results as shown in previous works - we empirically find 4 spatial mixtures to be sufficient for image classification task.

In Equation 4, previous works recommend fixing the occlusion prior apriori $(0.5 - 0.6)$ given this probability can't often be calculated. This prior is therefore largely uninformative, and we think calculating instance dependent occlusion prior is an interesting future work for this class of models.

In Equation 7, the prior probability is initialised using a vMF distribution with final source parameters and transitional features. The parameter update rules in Equation 9 and Equation 10 can be derived from the general MAP estimation equations for a vMF distribution as described in [4] (given for Gaussian Mixture Models, but applicable for vMF mixtures). Note that Equation 9 is an approximation of the derivation which empirically showed better results. The adaptation coefficients ψ simply allow better tuning of adaptation rates between source and transitional data. Given that we use EM algorithm for MAP, we are guaranteed to converge to local minima under minor assumptions.

B. Occluded-OOD-CV

Occluded-OOD-CV is a dataset which is designed to evaluate the robustness of a model to partial occlusion in combination with a nuisance factor, namely, weather, texture, shape, 3D pose and context. In the Occluded-OOD-CV dataset, we simulate the partial occlusion by imposing samples of occluding objects on the primary image objects in OOD-CV. The superimposing occluders are either naturally present in the original OOD-CV test set or are cropped from the MS-COCO dataset[12]. We have 3 levels of foreground occlusions in the Occluded-OOD-CV dataset - L1($20 - 40\%$), L2($40 - 60\%$) and L3($60 - 80\%$). The total number of (test) images for each level is 3820(L1), 3728(L2) and 3606(L3). A model which has been adapted to the OOD-CV nuisance data (in an unsupervised manner in our case) is then tested on the Occluded-OOD-CV data. The model is not shown the Occluded-OOD-CV data at any training or adapting stage. L0(0%) can be used to refer to the normal OOD-CV dataset. Figure 3 shows a few examples from the Occluded-OOD-CV dataset.

Given the dearth of real world occlusion datasets, we place the occluders *artificially* on the image objects to ape real world occlusion. We believe that this a robust analysis of occlusion robustness as earlier works [8, 19] have shown that partial occlusion analysis for many computer vision tasks with such artificially placed occluders is similar to images with real partial occlusion.

C. Experiments

Our experiments are divided into three parts -

1. *Real World Nuisance Factors* - OOD-CV[24] : Evaluation on this dataset is the focus of this work since it allows for real world (individual) nuisance robustness analysis. We can, therefore, test our models' capability to robustly adapt to nuisance-ridden real world data in an unsupervised manner.
2. *Synthetic Corruptions* - Imagenet-C[5] : We also evaluate model's capability in adapting to synthetic corruptions, like, snow, gaussian blur, pixelate, elastic transform, etc, taken from Imagenet-C[5].
3. *Synthetic to Real* - UDAParts[13] (to Pascal3D+[20]) : We do an initial analysis of synthetic to real transfer using images created from UDAParts dataset.

Our choice of datasets is motivated by the fact that we want to evaluate a model's robustness to individual corruptions (real or synthetic) and partial occlusion. Since, our focus is primarily on evaluating model's robustness to unlabelled (real world) nuisance ridden data and occlusion, we choose OOD-CV[24] as our primary evaluation data. We also apply Imagenet-C corruptions on Pascal3D+ objects[20] which allows us to evaluate our models on the compounded problem of synthetic corruptions and partial occlusion given that the dataset has a occluded version (Occluded Pascal) [19].

In order to further evaluate the capabilities of our model, we also include a synthetic to real robustness analysis. For this, we use images rendered from the object models of the UDAParts [13] dataset and then test our model on the same object set of Pascal3D+ dataset. Again, in inclusion to the synthetic data OOD robustness problem, we have partial occlusion added to the evaluation data [19].

C.1. OOD-CV

We evaluate all models on OOD-CV nuisance ridden test data (weather, context, texture, pose, shape) as well as our extended OOD setup which included Occluded OOD-CV with 3 levels of partial occlusions ($L1(0 - 20\%), L2(20 - 40\%), L3(40 - 60\%)$). The unoccluded data can also be referred to as L0.

C.1.1 Training Details

For RPL[15] and BNA[17], we used the official implementation[2]. For CompNets[10] as well, we used the official implementation[1]. For CompNets[10] and our model, to get feature vectors, we used a model backbone trained on the OOD-CV[24] training data. We use VGG16 (with Batch Normalization) and Residual Network (resnet50) for all models as model backbones as most aforementioned works use them. For CompNets[10] and our

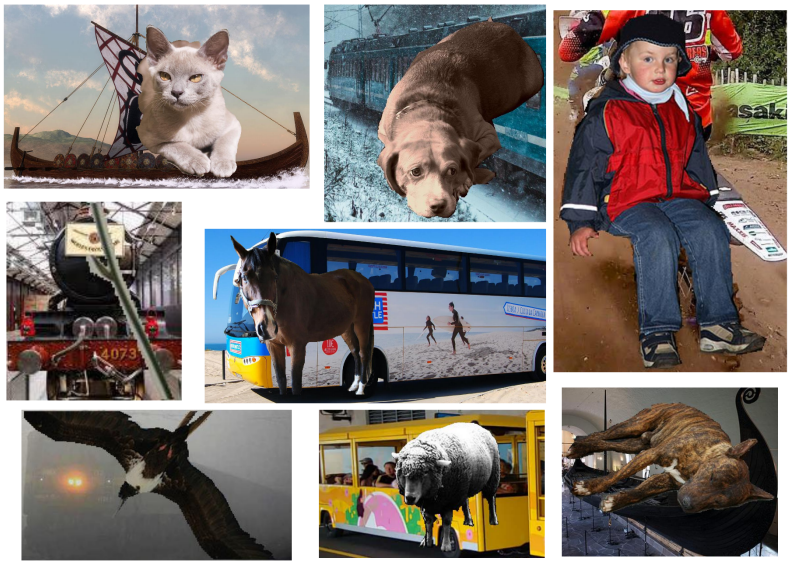

Figure 3. Examples from the Occluded-OOD-CV dataset

Figure 4. Subset of images from OOD-CV dataset activated by the same transitional spatial coefficient for clean source (top) data and nuisance ridden target (bottom) data.

model, we use $\kappa = 30$ (vMF variance) and the length of vMF dictionary is 512, with each vMF dictionary vector being 1x512. The initial VGG16 and Resnet50 backbones are trained using OOD-CV training data with 90-10 training-validation data split, and the same backbones are used for all the methods. These baseline backbones are trained using stochastic gradient descent optimizer with an initial learning rate of .0001 and weight decay of .0005. The batch size is 64 and the learning rate is changed by an exponential learning rate scheduler which decays the learning rate of

Table 1. Category-wise (Top-1) classification results for OOD-CV[24] (Combined Nuisance)

Model	Dataset	Acc.	plane	bicycle	boat	bus	car	chair	table	motorbike	sofa	train
UGT(ours)	L0	.85	.92	.924	.823	.846	.823	.62	.931	.82	.95	.882
UGT(ours)	L1	.6221	.825	.755	.514	.665	.612	.271	.621	.622	.736	.6
UGT(ours)	L2	.5684	.694	.689	.495	.588	.534	.255	.595	.605	.734	.495
UGT(ours)	L3	.5001	.599	.623	.435	.521	.499	.21	.515	.577	.655	.367
Compnet[11]	L0	.72	.782	.803	.867	.72	.759	.364	.849	.847	.812	.729
Compnet[11]	L1	.506	.528	.552	.701	.566	.397	.223	.676	.65	.587	.385
Compnet[11]	L2	.462	.494	.501	.699	.469	.303	.214	.613	.631	.588	.315
Compnet[11]	L3	.415	.491	.451	.598	.394	.289	.156	.537	.604	.444	.278
CDAN[14]	L0	.76	.802	.822	.781	.609	.891	.669	.471	.835	.934	.85
CDAN[14]	L1	.531	.627	.415	.459	.384	.601	.615	.455	.656	.774	.467
CDAN[14]	L2	.42	.507	.313	.412	.232	.461	.604	.253	.483	.758	.341
CDAN[14]	L3	.38	.443	.26	.344	.187	.404	.658	.235	.447	.661	.294
BSP[3]	L0	.753	.795	.8	.713	.677	.883	.65	.673	.77	.916	.802
BSP[3]	L1	.506	.577	.421	.346	.389	.624	.591	.51	.532	.798	.382
BSP[3]	L2	.401	.506	.28	.3	.236	.465	.542	.394	.433	.728	.293
BSP[3]	L3	.351	.431	.268	.217	.197	.38	.604	.294	.393	.682	.242
MDD[22]	L0	.78	.847	.843	.756	.643	.912	.742	.534	.869	.88	.782
MDD[22]	L1	.551	.668	.496	.452	.407	.622	.628	.455	.719	.677	.335
MDD[22]	L2	.469	.561	.354	.423	.218	.497	.639	.309	.583	.633	.243
MDD[22]	L3	.41	.514	.339	.326	.167	.44	.639	.227	.547	.544	.19
MCD[16]	L0	.772	.895	.844	.761	.761	.848	.673	.656	.902	.923	.782
MCD[16]	L1	.556	.755	.484	.414	.47	.432	.506	.531	.719	.716	.377
MCD[16]	L2	.461	.614	.36	.391	.293	.335	.494	.38	.635	.748	.284
MCD[16]	L3	.403	.559	.322	.282	.221	.262	.494	.294	.569	.634	.212
MCC[6]	L0	.785	.83	.81	.697	.725	.865	.405	.373	.738	.413	.75
MCC[6]	L1	.582	.525	.492	.336	.5	.482	.341	.222	.461	.383	.413
MCC[6]	L2	.492	.457	.375	.282	.3	.39	.264	.15	.354	.283	.316
MCC[6]	L3	.434	.415	.326	.214	.278	.327	.235	.126	.314	.209	.28

each parameter group by gamma$=0.95$ every 15 epochs. We use the same optimizer and learning rate schedules for our BNA[17] and RPL[15] experiments. For the generalized cross entropy loss[23], we use the recommended hyperparameter value $q = 0.8$ in all experimental usage. Hyperparameters $\psi_v = \psi_\alpha = 3$ for both CompNets and *UGT*. For finetuning our model, we freeze the model backbone parameters.

For MCC [6], CDAN [14], MCD [16], MDD [22] and BSP [3], we use the Transfer Learning library[7] implementations. We use the recommended hyperparameters for each method. Although, we do not use an Imagenet pretrained backbone for CompNets[10], RPL[15], BNA[17] and our method (UGT) as an Imagenet trained backbone would provide an unfair advantage to the method in the task of OOD robustness for an unlabelled data. However, for MCC [6], CDAN [14], MCD [16], MDD [22] and BSP [3], which are all well known Unsupervised Domain Adaptation methods, the model performance is not good when training from scratch. And therefore, we relax this pretrained backbone constraint for these methods, and show that even though our method is not using an Imagenet backbone, we are still able to perform better than current state-of-the-art OOD robustness[15, 17] and unsupervised domain adaptation methods [3, 6, 14, 16, 22].

C.2. Imagenet-C

Since previous OOD robustness methods [15, 17] primarily use synthetic corruptions [5], we employ the same corruptions and apply them to objects from Pascal3D+ dataset. The objective is to robustly adapt to an unlabelled data of synthetically corrupted images when the training data consists of only clean images. We apply synthetic corruptions to Pascal3D+ test data for our evaluation. Tables 5, 6, 7, 8 show additional classification accuracy results on different corruption target dataset versions.

Training Details We follow the same training methodology as mentioned in Section C.1.1.

Table 2. Category-wise (Top-1) classification results for OOD-CV[24] (Context Nuisance)

Model	Dataset	Acc.	plane	bicycle	boat	bus	car	chair	table*	motorbike	sofa	train
UGT(ours)	L0	.875	.927	.99	.857	1	NA	.733	NA	.432	.935	.789
UGT(ours)	L1	.624	.679	.822	.643	.4	.45	.634	.167	.31	.81	.698
UGT(ours)	L2	.565	.634	.786	.643	.35	.31	.495	.167	.209	.765	.595
UGT(ours)	L3	.511	.554	.713	.537	.2	.29	.433	.167	.139	.7	.579
CDAN[14]	L0	.71	.682	.564	.918	.75	NA	.766	NA	.675	.798	.802
CDAN[14]	L1	.541	.462	.389	.595	.4	.5	.702	.521	.432	.681	.511
CDAN[14]	L2	.436	.435	.23	.523	0	.464	.626	.333	.194	.636	.357
CDAN[14]	L3	.397	.391	.269	.292	0	.178	.742	.183	.194	.56	.342
BSP[3]	L0	.61	.476	.455	.857	.75	NA	.7	NA	.459	.83	.591
BSP[3]	L1	.511	.339	.372	.452	.6	.821	.782	.521	.27	.663	.302
BSP[3]	L2	.419	.237	.256	.38	0	.571	.747	.378	.166	.654	.19
BSP[3]	L3	.385	.228	.304	.219	.2	.5	.762	.3	.111	.51	.236
MCD[16]	L0	.798	.695	.821	.959	.75	NA	.833	NA	.918	.693	.619
MCD[16]	L1	.523	.566	.542	.547	.6	.71	.653	.376	.486	.603	.372
MCD[16]	L2	.426	.445	.35	.5	.08	.46	.595	.272	.277	.6	.309
MCD[16]	L3	.374	.467	.391	.317	.4	.357	.649	.233	.277	.51	.342

Table 3. Category-wise (Top-1) classification results for OOD-CV[24] (Weather Nuisance)

Model	Dataset	Acc.	plane	bicycle	boat	bus	car	chair	motorbike	train
UGT(ours)	L0	.856	.857	.875	.91	.882	.77	1	.843	.892
UGT(ours)	L1	.6	.778	.623	.599	.545	.51	NA	.715	.46
UGT(ours)	L2	.53	.7	.607	.517	.364	.401	NA	.686	.487
UGT(ours)	L3	.465	.558	.534	.479	.438	.37	NA	.67	.322
CDAN[14]	L0	.745	.694	.818	.759	.764	.77	1	.883	.89
CDAN[14]	L1	.476	.538	.491	.46	.272	.45	NA	.713	.367
CDAN[14]	L2	.335	.404	.341	.394	.151	.24	NA	.559	.269
CDAN[14]	L3	.299	.386	.318	.338	.093	.23	NA	.605	.156
BSP[3]	L0	.73	.583	.787	.665	.852	.73	.629	.831	.867
BSP[3]	L1	.391	.383	.516	.322	.272	.52	NA	.593	.261
BSP[3]	L2	.266	.271	.418	.238	.181	.29	NA	.381	.183
BSP[3]	L3	.254	.251	.387	.232	.125	.26	NA	.412	.17
MCD[16]	L0	.81	.791	.852	.763	.877	.49	.925	.841	.849
MCD[16]	L1	.447	.627	.565	.552	.393	.35	NA	.754	.336
MCD[16]	L2	.336	.491	.41	.462	.303	.19	NA	.697	.242
MCD[16]	L3	.286	.429	.405	.401	.156	.2	NA	.655	.189

Table 4. Hyperparameters

Model	Backbone	epochs	lr	Batch	layer
Compnet[10]	**VGG16**	60	0.01	64	pool5
RPL[15]	**VGG16**	60	0.0001	64	-
BNA[17]	**VGG16**	60	0.0001	64	-
UGT (Ours)	**VGG16**	50	0.01	64	pool5
Compnet[10]	**Resnet50**	60	0.01	64	layer4
RPL[15]	**Resnet50**	60	0.0001	64	-
BNA[17]	**Resnet50**	60	0.0001	64	-
UGT (Ours)	**Resnet50**	50	0.01	64	layer4

Figure 5. Example Images with Snow and Elastic corruptions (Severity 4) from Imagenet-C

Table 5. Imagenet-C Snow Corruption (Severity 4) Classification Results

Model	**Backbone**	**L0**	**L1** (20-40%)	**L2** (40-60%)	**L3** (60-80%)
Compnet[10]	**VGG16**	0.529	0.348	0.258	0.21
RPL[15]	**VGG16**	0.854	0.592	0.435	0.408
UGT (Ours)	**VGG16**	**0.885**	**0.742**	**0.634**	**0.523**
RPL[15]	**Resnet50**	0.725	0.589	0.442	0.379
BNA[17]	**Resnet50**	0.74	0.61	0.488	0.406
UGT (Ours)	**Resnet50**	**0.766**	**0.633**	**0.556**	**0.49**

Table 6. Imagenet-C Gaussian Noise Corruption (Severity 4) Classification Results

Model	**Backbone**	**L0**	**L1** (20-40%)	**L2** (40-60%)	**L3** (60-80%)
Compnet[10]	**VGG16**	0.549	0.361	0.221	0.171
RPL[15]	**VGG16**	0.87	0.619	0.468	0.342
UGT (Ours)	**VGG16**	**0.87**	**0.735**	**0.643**	**0.535**

Table 7. Imagenet-C Shot Noise Corruption (Severity 4) Classification Results

Model	**Backbone**	**L0**	**L1** (20-40%)	**L2** (40-60%)	**L3** (60-80%)
Compnet[10]	**VGG16**	0.514	0.285	0.189	0.15
RPL[15]	**VGG16**	0.883	0.63	0.472	0.343
UGT (Ours)	**VGG16**	**0.883**	**0.746**	**0.648**	**0.532**

C.3. UDAParts (Synthetic to Real)

Figure 6 shows an example of the synthetically created images that are used as our training data. They have created using object models and rendered using random 3D pose, texture and background. The purpose of this task is to make the model (which has trained on the synthetic data) robust to the OOD real data, which in this case is Pascal3D+. Additionally, the model needs to be robust to OOD real data which suffers from partial occlusion[19]. We choose around 2000 random images from the randomly generated images in UDAParts for each class of object as our training set.

Training Details We follow the same training methodology as mentioned in Section C.1.1.

Figure 7 shows an example of images which are activated by the same transitional spatial coefficient in the source (UDAParts) and target (Pascal3D+) dataset.

Table 8. Imagenet-C Pixelate Corruption (Severity 4) Classification Results

Model	Backbone	L0	L1 (20-40%)	L2 (40-60%)	L3 (60-80%)
Compnet[10]	**VGG16**	0.852	0.531	0.39	0.304
RPL[15]	**VGG16**	0.946	0.639	0.446	0.346
UGT (Ours)	**VGG16**	0.943	**0.815**	**0.713**	**0.591**
RPL[15]	**Resnet50**	0.948	0.626	0.435	0.33
BNA[17]	**Resnet50**	0.94	0.7	0.514	0.393
UGT (Ours)	**Resnet50**	**0.962**	**0.79**	**0.653**	**0.55**

Table 9. Imagenet-C Glass Blur Noise Corruption (Severity 4) Classification Results

Model	Backbone	L0	L1 (20-40%)	L2 (40-60%)	L3 (60-80%)
Compnet[10]	**VGG16**	0.56	0.35	0.29	0.26
RPL[15]	**VGG16**	0.80	0.49	0.39	0.31
UGT (Ours)	**VGG16**	**0.81**	**0.53**	**0.44**	**0.37**

Figure 6. Examples from UDAParts dataset

C.4. Data Augmentation and Model Backbones

We avoid using different data augmentations to ensure a fair and balanced comparison between all methods. Though, additional data or different kinds of data augmentations do help alleviate the problem, we focus on the methodology aspect and avoid using various kinds of data augmentations or additional data for all methods. Similarly, we keep the model backbones' similar, whenever possible to allow for a fairer comparison.

C.5. Training Cost

Training Costs (time) is similar to the baseline generative model [10] as we are only finetuning a few layers(spatial mixture matrix). Our code isn't optimized yet and our method, sans the finetuning, uses only the CPU. On AMD 5700x CPU and for a training data size of 10k samples,it takes 6 hrs to learn the transitional model and 15 minutes

Figure 7. Subset of images activated by the same transitional spatial coefficient for synthetic source (top) data and real target (bottom) data.

to finetune the model for 10 epochs on a Nvidia 3070.

D. Ablation

As we show in our ablation results in the main draft, each individual improvement of our method, UGT, markedly improves the classification results for unsupervised domain robustness. Here, $\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$ refers to calculating transitional vMF dictionary and then calculating the spatial coefficients using this transitional dictionary and source feature vectors. $\mathcal{A}^{\mathcal{R}'}$ refers to the finetuned transitional spatial coefficient.

In Tables 10, 11, 12, 13 and 14, we can see the improvements in Imagenet-C experiments where different kinds of Imagenet-C corruptions are applied to objects from Pascal3D+ dataset.

We can notice that just learning the transitional parameters is often enough to drastically improve model's performance underlying its importance in our work.

E. Transitional vMF dictionary

The transitional vMF dictionary is learnt by initializing the dictionary vectors with learnt $\Lambda^{\mathcal{S}}$ and regularizing the learning of $\Lambda^{\mathcal{R}}$ by the cosine distance between $\mu^{\mathcal{S}}$ and $\mu^{\mathcal{R}}$. Since, the directional parameter or the vMF variance (σ_k/κ) is constant (and therefore, the normalization term in the denominator is constant), μ is the only parameter being learnt. Figure 2a gives an intuition regarding the source and transitional vMF kernels.

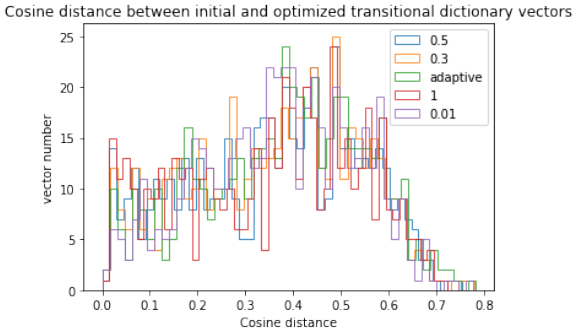

Figure 8. Histogram representing the cosine distance between initial and final transitional dictionary vectors with different values of ψ_k

Also, as we can see in Figure 8, the choice of the adaptation coefficients ψ_k, which represents the hyperparameters used to tune the adaptation between source and transitional μ_k, is not that critical. Figure 8 represents the learning of vMF kernel dictionary for the combined nuisance experiment where the entire OOD-CV test data is considered the target domain. All the normalized ψ_k are initialized to an arbitrary value between 0 and 1 and then step wise increased

Table 10. Ablation analysis for Imagenet-C (Gaussian Noise Corruption) Classification

Model	**Backbone**	**OOD-CV**	**Occ.-OOD-CV** (20-40%)	**Occ.-OOD-CV** (40-60%)	**Occ.-OOD-CV** (60-80%)
Baseline(B)	**VGG16**	0.549	0.361	0.221	0.171
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.765	0.723	0.533	0.423
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	**0.87**	**0.735**	**0.643**	**0.535**

Table 11. Ablation analysis for Imagenet-C (Pixelate Corruption) Classification

Model	**Backbone**	**OOD-CV**	**Occ.-OOD-CV** (20-40%)	**Occ.-OOD-CV** (40-60%)	**Occ.-OOD-CV** (60-80%)
Baseline(B)	**VGG16**	0.852	0.531	0.39	0.304
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.911	0.732	0.597	0.511
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	0.943	**0.815**	**0.713**	**0.591**

Table 12. Ablation analysis for Imagenet-C (Motion Blur Corruption) Classification

Model	**Backbone**	**OOD-CV**	**Occ.-OOD-CV** (20-40%)	**Occ.-OOD-CV** (40-60%)	**Occ.-OOD-CV** (60-80%)
Baseline(B)	**VGG16**	0.639	0.362	0.287	0.241
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.789	0.631	0.492	0.413
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	**0.891**	**0.763**	**0.673**	**0.567**

Table 13. Ablation analysis for Imagenet-C (Shot Noise Corruption) Classification

Model	**Backbone**	**OOD-CV**	**Occ.-OOD-CV** (20-40%)	**Occ.-OOD-CV** (40-60%)	**Occ.-OOD-CV** (60-80%)
Baseline(B)	**VGG16**	0.514	0.285	0.189	0.15
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.752	0.596	0.518	0.418
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	**0.883**	**0.746**	**0.648**	**0.532**

Table 14. Ablation analysis for Imagenet-C (Gaussian Blur Corruption) Classification

Model	**Backbone**	**OOD-CV**	**Occ.-OOD-CV** (20-40%)	**Occ.-OOD-CV** (40-60%)	**Occ.-OOD-CV** (60-80%)
Baseline(B)	**VGG16**	0.732	0.395	0.296	0.241
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.848	0.615	0.5	0.387
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	**0.909**	**0.72**	**0.613**	**0.509**

until likelihood improvement of the particular vMF dictionary vector is below a threshold value($1e-5$). As seen in the figure, different initial values for the adaptation coefficient produce similar results and therefore the usage of the data dependent/adaptive coefficient is optional in our experiments.

Figures 9,10 and 11 show examples of source and their corresponding transitional vMF dictionary vector image patches. The image patches roughly correspond to parts of an image in the data which has high activations for a spe-

Table 15. Ablation analysis for Imagenet-C (Elastic Transformation Corruption) Classification

Model	Backbone	OOD-CV	Occ.-OOD-CV (20-40%)	Occ.-OOD-CV (40-60%)	Occ.-OOD-CV (60-80%)
Baseline(B)	**VGG16**	0.268	0.183	0.157	0.146
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	**VGG16**	0.771	0.637	0.549	0.465
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	**VGG16**	**0.872**	**0.712**	**0.712**	**0.494**

cific vMF dictionary vector. The image patches are from the training and evaluation sets of OOD-CV dataset respectively, which have been curated to filter out training like data in the nuisance factored test set. However, we can still notice that the approximate image patches actually correspond to similar object parts despite drastic changes in 3D pose, shape, texture, weather, etc. For e.g, in Figure 9, we can see patches focusing on vehicle windows, bicycle and motorbike handlebars, etc.

E.1. Finetuning via Pseudo-Labelling

We finetune our transitional spatial coefficients by simple pseudo-labelling on a target data subset. This subset is not used during evaluation - although test time finetuning can be performed to improve the model's performance during inference. We do not follow any sophisticated pseudo-labelling methods and simply use a high threshold value (e.g 0.8) for the final class activations to pseudo-label target domain samples. We can, presumably, achieve higher performance using better and more sophisticated pseudo-labelling and finetuning methods, however, this is not a point of concern for this work.

F. Future Work and Limitations

Figure 12 shows that our adapted image classification model is capable of localizing occluders (like CompNets[10] do in a non-corrupted scenario) even in a nuisance ridden target image. Thus, we think that applications like object detection and segmentation are interesting future applications of our work.

In terms of limitations, our method may not work in case of extreme divergence between the source and target domains which may be be caused by inter-domain noise or large difference in both domain object parts and their spatial orientations. Our model is also bound to the assumptions and weaknesses of the Bayesian, generative model [9] which we build upon, a discussion of which can be found in the relevant works, like [10, 11, 18].

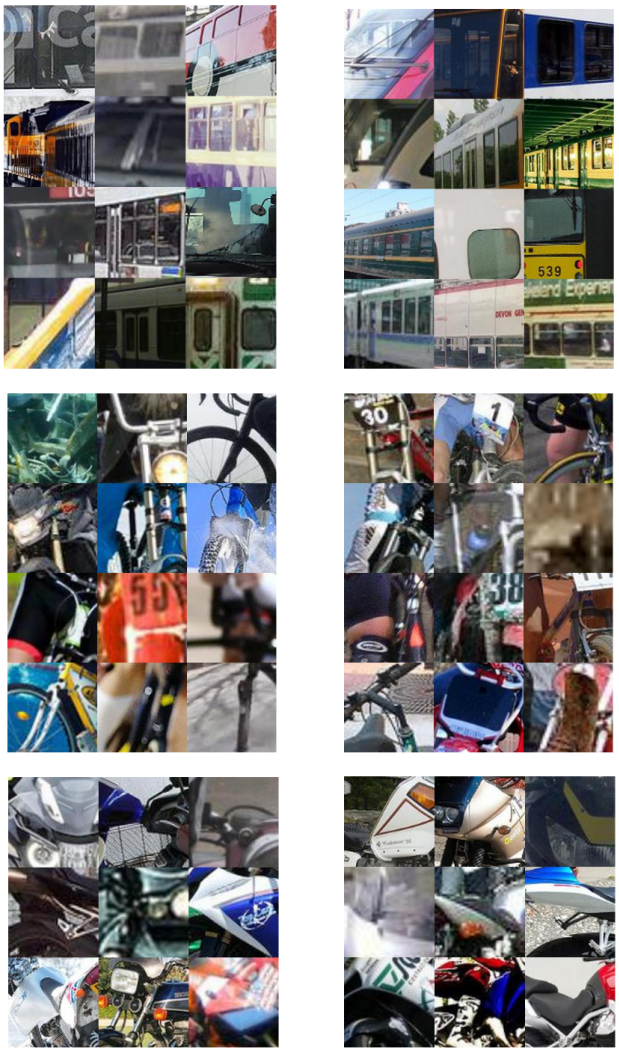

Figure 9. Image patches from OOD-CV training (left) and nuisance test (right) data which have high activations for corresponding vMF vectors from the source (initialised $\Lambda^{\mathcal{R}}$) and transitional vMF dictionaries.

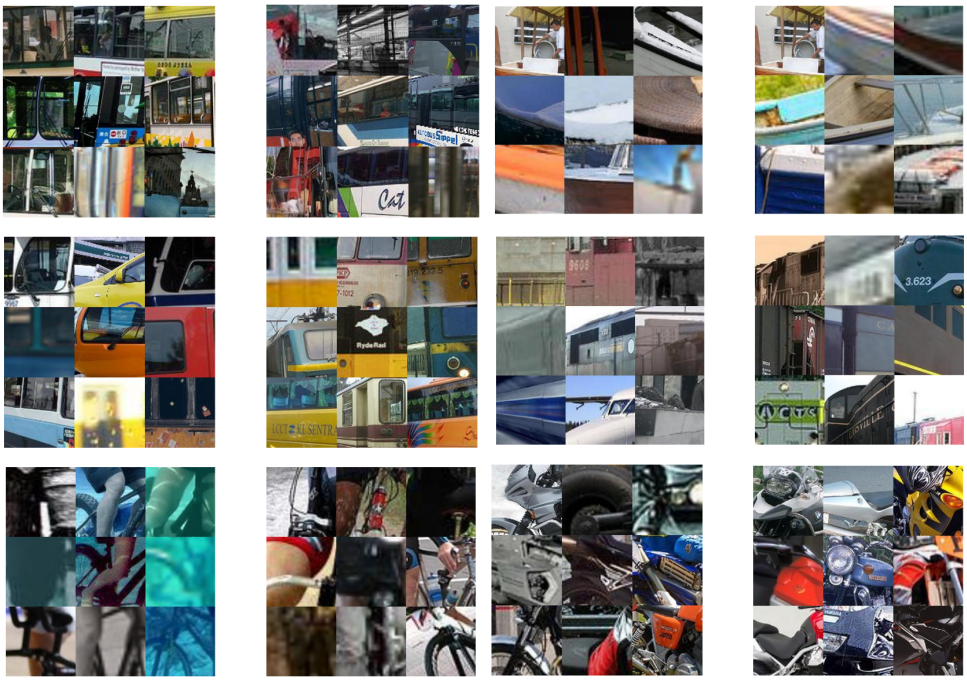

Figure 10. Additional examples of roughly corresponding image patches from OOD-CV training (left) and nuisance test (right).

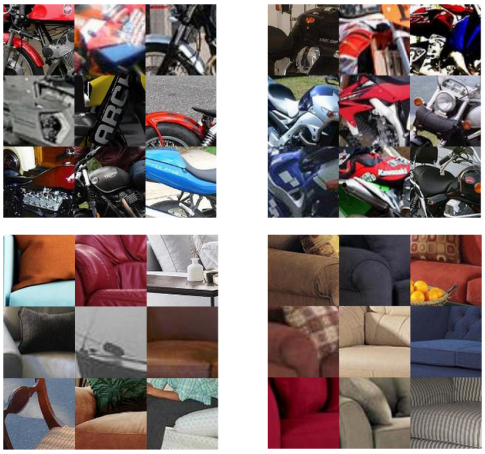

Figure 11. Additional example image patches from OOD-CV training (left) and nuisance test (right) data.

Figure 12. Occluder localization in a *snow* corrupted car image

References

[1] a. Compositional Convolutional Neural Networks. https://github.com/AdamKortylewski/CompositionalNets,. 5

[2] a. Robusta Toolbox. https://github.com/bethgelab/robustness,. 5

[3] Xinyang Chen, Sinan Wang, Mingsheng Long, and Jianmin Wang. Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation. In *Proceedings of the 36th International Conference on Machine Learning*, pages 1081–1090. PMLR, 2019. 4, 7, 8

[4] Jean-Luc Gauvain and Chin-Hui Lee. Lee, c.: Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. ieee trans. speech audio process. 2, 291-298. *Speech and Audio Processing, IEEE Transactions on*, 2:291 – 298, 1994. 5

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 4, 5, 7

[6] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation, 2019. 4, 7

[7] Jiang Junguang, Chen Baixu, Fu Bo, and Long Mingsheng. Transfer-learning-library. https://github.com/thuml/Transfer-Learning-Library, 2020. 7

[8] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. Pare: Part attention regressor for 3d human body estimation, 2021. 4, 5

[9] Adam Kortylewski, Ju He, Qing Liu, and Alan Loddon Yuille. Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8937–8946, 2020. 13

[10] Adam Kortylewski, Qing Liu, Angtian Wang, Yihong Sun, and Alan Yuille. Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion. *International Journal of Computer Vision*, 129(3):736–760, 2021. 4, 5, 7, 8, 9, 10, 13

[11] Adam Kortylewski, Qing Liu, Angtian Wang, Yihong Sun, and Alan Yuille. Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion. *International Journal of Computer Vision*, 2021. 7, 13

[12] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. *Lecture Notes in Computer Science*, 2014. 5

[13] Qing Liu, Adam Kortylewski, Zhishuai Zhang, Zizhang Li, Mengqi Guo, Qihao Liu, Xiaoding Yuan, Jiteng Mu, Weichao Qiu, and Alan Yuille. Learning part segmentation through unsupervised domain adaptation from synthetic vehicles. In *CVPR*, 2022. 4, 5

[14] Mingsheng Long, ZHANGJIE CAO, Jianmin Wang, and Michael I Jordan. Conditional adversarial domain adaptation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2018. 4, 7, 8

[15] Evgenia Rusak, Steffen Schneider, Peter Gehler, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Adapting imagenet-scale models to complex distribution shifts with self-learning, 2021. 4, 5, 7, 8, 9, 10

[16] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018. 4, 7, 8

[17] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation, 2020. 4, 5, 7, 8, 9, 10

[18] Yihong Sun, Adam Kortylewski, and Alan Yuille. Weakly-supervised amodal instance segmentation with compositional priors. *arXiv preprint arXiv:2010.13175*, 2020. 13

[19] Angtian Wang, Yihong Sun, Adam Kortylewski, and Alan L Yuille. Robust object detection under occlusion with context-aware compositionalnets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12645–12654, 2020. 4, 5, 9

[20] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2014. 5

[21] Youshan Zhang. A survey of unsupervised domain adaptation for visual recognition, 2021. 4

[22] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation, 2019. 4, 7

[23] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018. 7

[24] Bingchen Zhao, Shaozuo Yu, Wufei Ma, Mingxin Yu, Shenxiao Mei, Angtian Wang, Ju He, Alan Yuille, and Adam Kortylewski. Ood-cv: A benchmark for robustness to out-of-distribution shifts of individual nuisances in natural images. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 1, 4, 5, 7, 8

[25] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–20, 2022. 4