A Bayesian Approach to OOD Robustness in Image Classification

Prakhar Kaushik
Johns Hopkins University
pkaushi1@jh.edu

Adam Kortylewski
University of Freiburg
akortyle@mpi-inf.mpg.de

Alan Yuille
Johns Hopkins University
ayuille1@jh.edu

Abstract

An important and unsolved problem in computer vision is to ensure that the algorithms are robust to changes in image domains. We address this problem in the scenario where we only have access to images from the target domains. Motivated by the challenges of the OOD-CV [45] benchmark where we encounter real world Out-of-Domain (OOD) nuisances and occlusion, we introduce a novel Bayesian approach to OOD robustness for object classification. Our work extends Compositional Neural Networks (CompNets), which have been shown to be robust to occlusion but degrade badly when tested on OOD data. We exploit the fact that CompNets contain a generative head defined over feature vectors represented by von Mises-Fisher (vMF) kernels, which correspond roughly to object parts, and can be learned without supervision. We obverse that some vMF kernels are similar between different domains, while others are not. This enables us to learn a transitional dictionary of vMF kernels that are intermediate between the source and target domains and train the generative model on this dictionary using the annotations on the source domain, followed by iterative refinement. This approach, termed Unsupervised Generative Transition (UGT), performs very well in OOD scenarios even when occlusion is present. UGT is evaluated on different OOD benchmarks including the OOD-CV dataset, several popular datasets (e.g., ImageNet-C [9]), artificial image corruptions (including adding occluders), and synthetic-to-real domain transfer, and does well in all scenarios.

1. Introduction

In recent years, machine learning algorithms have been extremely successful for tasks like object classification when evaluated on benchmarked datasets like ImageNet. But these successes require that the training and test data (or the source domain and the target domain data) be identically and independently distributed (IID) from some underlying source. However, in practice, it is important to ensure that the algorithms generalize to data that differ from the training data. For example, in real-world applications, an algorithm for car detection may encounter cars with unusual shapes and textures (Fig. 3), which did not occur in the training set.

Existing OOD methods [9–12, 28] have shown success in dealing with robustness issues when evaluated on early robustness datasets, such as Imagenet-C [9], Imagenet-R [11], and Imagenet-A [12], where the domain differences are due to synthetic corruptions, adversarial images, rendered images, and similar factors [45]. But these algorithms performed less well on a newer benchmark, OOD-CV [45], which focuses on systematic analysis of real-world nuisances, e.g. changes in texture, 3D pose, weather, shape, and context. From a related perspective, OOD-CV studies the causal factors that result in the domain gap [4]. In addition, previous works have rarely been evaluated for robustness to occlusion, an important OOD robustness metric.

In this work, we address OOD robustness on OOD-CV, and related datasets, focusing on real-world domain differences and occlusion. We build on a class of Bayesian neural models called Compositional Neural Networks (CompNets), as they have been shown to be robust to partial occlusion [20, 21, 36, 42]. This is achieved by replacing the discriminative head of a CNN with a generative model of the feature vectors based on the objects' spatial geometry. However, CompNets are fully supervised and are not robust to OOD nuisances. In this work, we develop an unsupervised approach, Unsupervised Generative Transition (**UGT**), which generalizes CompNets to OOD scenarios.

UGT relies on intuition that in OOD scenarios, the appearance of object parts is highly variable (due to changes like texture or weather), while the spatial geometry of objects is often fairly similar between domains. We analyze CompNets and modify them to take advantage of the intuition mentioned above. By introducing a *transitional dictionary* of von Mises-Fisher [17] kernels (Fig. 1), which shares the properties of both domains, we can intuitively

This work has been supported by Army Research Laboratory award W911NF2320008 and ONR with N00014-21-1-2812. A Kortylewski acknowledges support via his Emmy Noether Research Group funded by the German Science Foundation (DFG) under Grant No. 468670075.

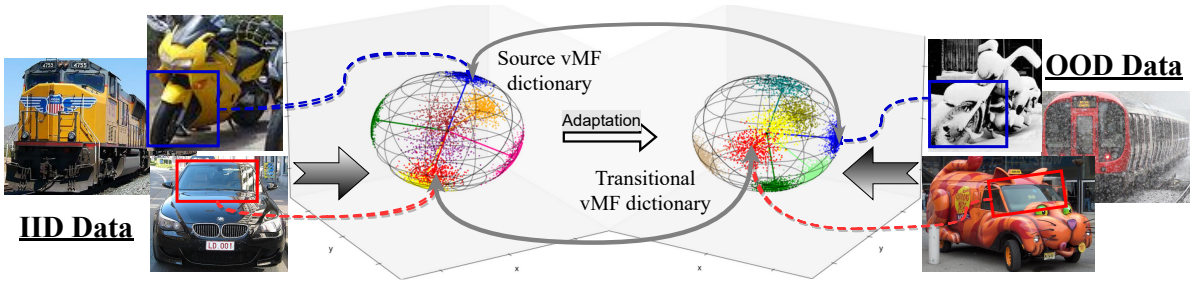

Figure 1. Illustration of the key principle underlying our Bayesian approach. Related work has shown that clusters of feature vectors learned in an unsupervised manner resemble part-like patterns [21, 39]. We observe that some feature clusters (represented here on a vMF manifold) are very similar in both IID and OOD data (illustrated in blue and red boxes), whereas for other feature clusters there is no corresponding equivalent in the other domain. Our Bayesian approach exploits this property by first learning a generative model of feature clusters and their spatial combinations on the IID data and subsequently adapting the model to OOD data via an unsupervised adaptation of the vMF cluster dictionary, while retaining the spatial relations between clusters.

learn the spatial geometry of the source and transfer it to the target domain. UGT leverages the property that the hierarchical structure of generative models like CompNets can be learned in a two-stage manner. **1)** An unsupervised learning stage of a dictionary of neural network features, called *vMF kernels*, using clustering in both source and target domains. The vMF kernels intuitively represent local object part structures. **2)** A supervised learning stage of the spatial relations of the vMF kernels on the source domain.
We primarily evaluate UGT on the OOD-CV benchmark [45]. In addition, to challenge UGT, we add occluders to OOD-CV and create a new dataset called *Occluded-OOD-CV* (Sec. 4.1). We also test UGT on Imagenet-C corruptions and Synthetic-to-Real domain robustness. Our studies show that UGT performs well on all these tasks and significantly outperforms the SOTA baselines.

We make several important contributions in this paper.

1. We model objects by a generative model on feature vectors. Our method, UGT, extends CompNets [21] by decoupling the learning into unsupervised learning of vMF kernels and supervised learning of the spatial geometry enabling us to learn transitional dictionaries.
2. UGT achieves state-of-the-art results on the real-world OOD robustness problem on the OOD-CV dataset [45] and demonstrates exceptional performance on generalizing under the synthetic corruptions of Imagenet-C.
3. UGT also achieves strong results for the Synthetic-to-Real scenario (UDAParts [24] to Pascal3d+) dataset.
4. We introduce the Occluded-OOD-CV dataset by adding occluders to OOD-CV and show that UGT is robust to this compounded problem of occlusion and nuisance.

2. Related Works

OOD robustness can be considered a subset of the larger unsupervised domain adaptation problem and is closely related to domain generalization and transfer learning. Although related to both, our work focuses on OOD robustness. 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 not found in the source dataset.

In the past few years, there has been an increase in the number of works [9–12, 28] that characterize model performance on OOD data and treat this as a measure of robustness. The common idea that underlies most works is to leverage a property of the unlabeled target domain to allow generalization of a model trained on the source domain. There have been successful efforts to use feature statistics to adapt to the new domain; e.g., Sun et al. [35] try to minimize domain shift by aligning the second-order statistics of source and target distributions; Bug et al. [1] employ feature aware normalization with gating elements from Long Short-Term Memory units for normalization among different spatial regions of interest. Some methods employ techniques based on adaptive batch normalization and weight normalisation [32]. Other methods include self-learning using entropy minimization [38], adaptive pseudo-labeling techniques [5, 14, 33, 34] and robust lost functions [6, 44].

Although, current works have been successful at dealing with robustness problems when evaluated on earlier robustness datasets [9, 11, 12] they have been shown to struggle with real world nuisances (OOD-CV [45]) and occlusion [16, 21]. Few generative Bayesian methods such as CompNets [21, 36, 39] have shown their relative robustness to occlusion, but still struggle with other OOD nuisances.

3. Method

We address OOD robustness from a Bayesian perspective which, to the best of our knowledge, is novel. Our starting point is a class of generative models, described in Sec. 3.1,

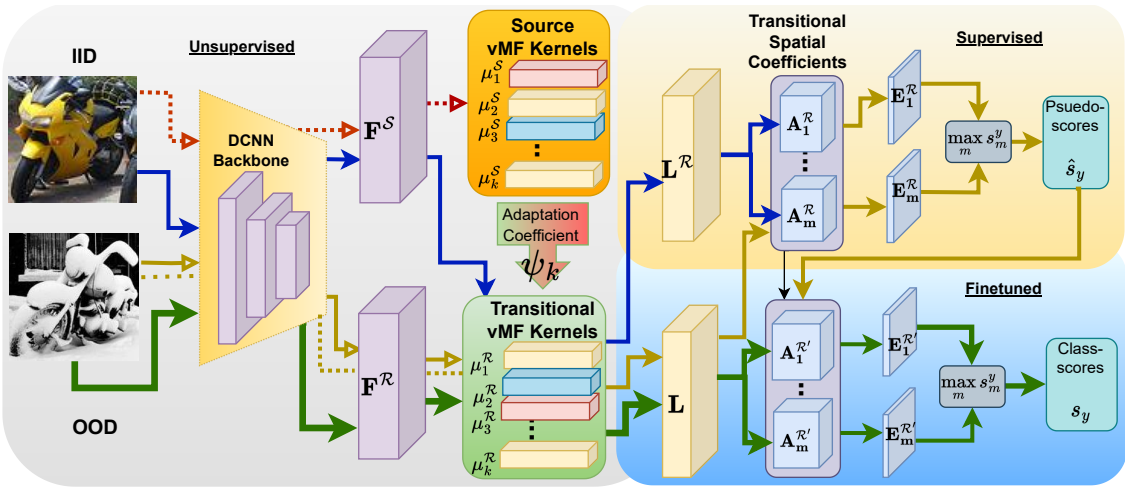

Figure 2. Rough illustration of our Bayesian method. (- - →,- - →) A DCNN backbone is used to extract the source (IID) $F^{\mathcal{S}}$ and target (OOD) features $F^{\mathcal{R}}$. The source feature vectors $F^{\mathcal{S}}$ are then used to learn the source vMF kernels that are then adapted to the transitional vMF kernels using target domain features $F^{\mathcal{R}}$ and the adaptation coefficients ψ_k in an unsupervised manner. (→) Transitional Spatial coefficients ($A^{\mathcal{R}}$) are then learned using the transitional vMF likelihood $L^{\mathcal{R}}$ i.e. non-linear activation applied to a convolution of $F^{\mathcal{S}}$ and transitional kernels using source labels. (→) These spatial coefficients are then finetuned ($A^{\mathcal{R}'}$) using pseudo-scores $\{\hat{s}\}$ generated using the transitional mixture likelihood $E^{\mathcal{R}}$ of target domain features $F^{\mathcal{R}}$. (→) shows the final feedforward pipeline during inference.

which have been shown to be robust to occlusion [21] when not dealing with other OOD nuisances. We describe method motivation in Sec. 3.2 and the technical details in Sec. 3.3.

3.1. Bayesian Neural Architecture

Our base architecture is similar to CompNets [21] and is explained in this section to help readers unfamiliar with them. Our method extends this class of neural models by non-trivially modifying the training methodology to enable OOD robustness along with occlusion robustness.

This class of models differs from conventional Deep Networks by replacing the discriminative head by a generative model of feature vectors. For each object y we learn a generative model $P(F|y)$ for the feature vectors F. This model is formulated as a mixture model $P(F|y) = \sum_m P(F|y,m)$ where the mixture variable m roughly corresponds to the viewpoint of the object. The conditional distributions $P(F|y,m)$ for the features are factorizable in terms of position so that $P(F|y,m) = \prod_{a\in\mathcal{D}} P(f_a|y,m)$, where $a \in \mathcal{D}$ specifies the position in the image. These distributions $P(f_a|y,m)$ are specified in terms of *von Mises-Fisher (vMF) dictionaries*, with parameters $\Lambda = \{\sigma_k, \mu_k\}$ and by *spatial coefficients* with parameters $\mathcal{A} = \{\alpha_{a,k}^{y,m}\}$. We use the following generative probability distribution for the neural features F conditioned on an object y [20, 21]:

$$P(F|y) = \sum_m P(F|y,m) = \sum_m \prod_{a\in\mathcal{D}} P_a(f_a|y,m)P(m), \quad (1)$$

$$P_a(f_a|y,m) = P_a(f_a|\mathcal{A},\Lambda) = \sum_k \alpha_{a,k}^{y,m} P(f_a|\sigma_k,\mu_k), \quad (2)$$

$$P(f|\sigma_k,\mu_k) = \frac{e^{\sigma_k \mu_k^T f}}{Z(\sigma_k)}, ||f|| = 1, ||\mu_k|| = 1, \quad (3)$$

We typically use 4 mixture components in our method and $P(m)$ is an uniform prior over the mixture components. As shown in [21, 39] each vMF kernel can be qualitatively interpreted as a subpart of the object (i.e., all image patches with feature responses close to μ_k look like visually similar object subparts). We use von Mises-Fisher distributions instead of Gaussian distributions because the feature vectors f_a and the means μ_k must have a unit norm [7, 8]. The *spatial coefficients* $\mathcal{A} = \{\alpha_{a,k}^{y,m}\}$ specify the probability that the vMF kernel k occurs at the position a conditioned on the object y and its mixture component m.

Inference. After learning, inference on an image with feature vectors F is performed by a forward pass which estimates which object is more likely to generate the features F of the input image, $\hat{y} = argmax_y P(F|y)$ [21, 36].

Occlusion modeling. To make the model, described above, robust to occlusion (in non-OOD data), an outlier process is added to allow for some of the image features to be generated by the object and others by a separate outlier process [20, 36]. This is formalised by:

$$P(F|y) = \prod_{a\in\mathcal{D}} \sum_m P_a(f_a|y,m)^{z_a} Q(f_a)^{1-z_a} P(m) P(z_a) \quad (4)$$

where $Q(f_a)$ is a vMF distribution for a feature generated by an occluder which can be estimated from non-annotated images [19, 21, 42]. The latent variable $z_a \in \{0,1\}$ indicates whether pixel a is occluded or not occluded ($z_a = \{1,0\}$ respectively) and the prior $P(z_a)$ indicates the prior probability of a pixel being occluded. Note that we could also, in theory, sum over z (we currently take a max).

Training CompNets [21, 36, 42] are trained end-to-end to optimize the model parameters $\Lambda, \mathcal{A}$ using the standard

supervision for object classification (e.g., the mixture components and the vMF kernels are treated as latent variables). In an OOD scenario the image features no longer correspond well with the learned generative model and without labels, we cannot trivially finetune the model. UGT utilizes an insightful training strategy to solve this problem.

3.2. Motivation on Generalizing to OOD Data

UGT builds upon by the aforementioned Bayesian model because it gives a natural way to formulate an occluder process. These models, however, do not do well on OOD data (Sec. 4). To solve this problem in an unsupervised manner requires reformulation of the training process. We motivate our solution for OOD, UGT, in following stages.

Firstly, the vMF kernel dictionaries (i.e., the subparts of the object) can be learnt without supervision and hence can be found on both the source (annotated) and the target (non-annotated) domains. **Secondly**, we observe that some of the vMF kernels are similar between different domains (intuitively some subparts are similar between both domains). **Thirdly**, we can build on this observation to learn a transitional dictionary, which encourages vMF kernels in both domains to be similar if possible, and which works well in both domains. **Fourthly**, we note that the spatial coefficients capture the spatial activity pattern of the vMF kernels and these patterns depend on the spatial structure of the objects and so are mostly invariant to the domain, which suggests that we can learn the spatial coefficient on the source domain (where annotations are available), provided we use the transitional dictionary of vMF kernels, and that these spatial coefficients give a good initial estimate for the spatial coefficients on the target domain (which can be improved by simple pseudo-labeling).

In the **first stage** we learn the vMF dictionaries Λ without supervision by maximum likelihood estimation (MLE) assuming that the feature vectors $\{f_a\}$ of all the images (and at all positions) in each domain are generated by a mixture of von-Mises-Fisher distributions $P(f|\Lambda) = \sum_k \pi_k e^{\sigma_k \mu_k^T f}/Z[\sigma_k]$. This is essentially clustering similar to that used in earlier studies [21, 39]. After the Λ are learnt, if annotations are available (i.e., we know the object y) then we can learn the spatial coefficients $\mathcal{A}$ from the data $\{F_n\}$ in the annotated (source) domain by MLE from the distribution $\sum_m \prod_{a \in D} \sum_k \alpha_{a,k}^{y,m} P(f_a|\sigma_k, \mu_k)$.

In the **second stage**, we compare the vMF dictionaries $(\Lambda^{\mathcal{S}})$ and (Λ^{T}) on the source (S) and target (T) domain respectively. We observe that a subset of the dictionary vectors are similar, as measured by cosine similarity in the vMF feature space (Fig. 1). We **conjecture** that this is because a subset of the vMF kernels, which correspond roughly to object subparts [39], is invariant to the nuisance variables which cause the differences between the domains. For example, for an object like a car or bus, some subparts like wheels and license plates may be very similar between the source and target domains but others may not (Fig. 1). These observations motivate us to learn a *transitional vMF dictionary* $(\Lambda^{\mathcal{R}})$. This dictionary is learnt by learning the dictionary on the target domain but adding a prior (or regularization constraint) that the dictionary elements in both domains are similar. Finally, we learn the spatial coefficients $\mathcal{A}$ on the source domain, but using the transitional dictionary (Sec. 3.3.2). This allows us to utilize object geometry knowledge from the source domain in the target domain. As we show in our experiments and ablation (Sec. 4, Sec. 4.3), this model already works well on the target domain and can be improved by pseudo-labelling techniques.

3.3. Training UGT

Our Bayesian method, **UGT**, involves 3 steps - **1)** primarily, learning transitional dictionary $\Lambda^{\mathcal{R}}$, **2)** learning transitional spatial coefficients $\mathcal{A}^{\mathcal{R}}$ using $f^{\mathcal{S}}$ and $\Lambda^{\mathcal{R}}$, and lastly **3)** finetuning the transitional parameters $(\Lambda^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}})$ using simple pseudo-labelling. Refer to Fig. 2 for a simple illustration.

3.3.1 Learning Transitional Dictionary

We initialize the *transitional* von Mises-Fisher(vMF) dictionary vectors with the learnt source domain vMF dictionary vectors, i.e., $\Lambda^{\mathcal{R}}_{\text{initial}} = \Lambda^{S}$. The source domain vMF dictionaries i.e., $(\Lambda^{\mathcal{S}}(\mu, \sigma))$ are learnt from the features $f^{\mathcal{S}}$ in source domain by MLE as described in Sec. 3.1 using the EM algorithm [39]. We can learn the transitional vMF dictionary parameters $\Lambda^{\mathcal{R}}$ from the target domain feature vectors $f^{\mathcal{R}}$ through a few ways. We can maximize the regularized likelihood shown in Eq. (5) using the EM algorithm used to calculate the source domain parameters. Eq. (5) shows the Bayesian parameterization of our transitional model and can be seen as a penalized or regularized form of maximum likelihood estimation. We penalize the distance between the initialized transitional mean vectors (which are the source parameters) and the learnt ones. This regularization (like others) also helps in avoiding overfitting. Since, we fix σ_k as constant to reduce computation, the normalization term $Z(\sigma)$ reduces to a constant, and we can derive the penalized log-likelihood term as shown in Eq. (6). ψ is a adaptation parameter discussed later.

$$p(f^{\mathcal{R}}|\Lambda^{\mathcal{R}}) = \prod_n \sum_k \alpha_k P(f_a|\sigma_k, \mu_k) \exp\left(-\psi_k \sum_k (||\mu_k - \mu_k^{\mathcal{S}}||)\right) \tag{5}$$

$$l(\Lambda^{\mathcal{R}}) = \sum^{n} \log(\sum^{k} \pi_k \frac{e^{\sigma_k \mu_k^T f_i}}{Z(\sigma_k)}) - \psi_k \sum^{n} \sum^{k} (||\mu_k - \mu_k^{\mathcal{S}}||) \tag{6}$$

$$||f|| = 1, ||\mu_k|| = 1, \sigma = 1 \implies Z(\sigma) = const.$$

The Expectation step for learning the transitional parameters is similar the source version. In the first step, we calculate the summary statistics for the transitional parameters

Algorithm 1 Unsupervised Generative Transition

Input: Set of source domain images $I^{\mathcal{S}} = \{I_1^{\mathcal{S}}, ..., I_n^{\mathcal{S}}\}$, target domain images $I^T = \{I_1^T, ..., I_N^T\}$, source domain labels $y = \{y_1^{\mathcal{S}}, ..., y_n^{\mathcal{S}}\}$, deep network backbone $\Gamma(., \zeta)$, background images $\mathcal{B}_{i=1}^r$
Output: Target domain model parameters $\mathcal{T} = (\mathcal{A}, \Lambda)$, background model β_r
procedure UGT($I^{\mathcal{S}}, I^T, y, \Gamma, \beta_r$)
  $\{F^{\mathcal{S}}\}, \{F^{\mathcal{R}}\} \longleftarrow \Gamma((\{I^{\mathcal{S}}\}, \{I^T\}), \zeta)$ ▷ Extract source & target featuremaps from DCNN backbone
  $\Lambda^{\mathcal{S}}(\mu_k) \longleftarrow$ cluster & MLE($\{F^{\mathcal{S}}\}$) ▷ Initialize source vMF kernels by kmeans & learn using MLE
  $\Lambda^{\mathcal{R}}_{initial}(\mu) \longleftarrow \Lambda^{\mathcal{S}}(\mu_k)$ ▷ Initialise transitional vMF kernels with source vMF kernels
  $\Lambda^{\mathcal{R}}(\mu) \longleftarrow$ MLE($F^T, \Delta(\psi, \Lambda^{\mathcal{S}}, \Lambda^{\mathcal{R}})$) ▷ Learn transitional vMF features using regularized MLE with target domain data (Sec. 3.3.1, Eq. (5)-Eq. (9))
  $\{L^{\mathcal{R}}\} \longleftarrow \sum_k \pi_k e^{\sigma_k \mu_k^T f^{\mathcal{S}}} / Z[\sigma_k](F * \Lambda^{\mathcal{R}}(\mu_k))$ ▷ Compute regularized transitional vMF likelihood with source featuremaps and transitional vMF kernels
  ${\mathcal{A}^{\mathcal{R}}}_{y_s,m} \longleftarrow$ cluster&MLE($\{L^{\mathcal{R}}\}, y_S$) ▷ Calculate spatial coefficients using transitional vMF likelihood and source feature vectors (Sec. 3.3.2)
  $y_{\hat{T}} \longleftarrow argmax_y P(F|\Lambda^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}})$ ▷ Pseudo-label target domain data using transitional model
  ${\mathcal{A}^{\mathcal{R}}}'_{y_{\hat{T},m}} \longleftarrow$ cluster&MLE($\{L^{\mathcal{R}}\}, y_{\hat{T}}$) ▷ Finetune spatial coefficients using pseudolabelled data $y_{\hat{T}}$
  $\mathcal{T} \longleftarrow optimize(\mathcal{L}_{\text{gce}} + \psi_v \mathcal{L} + \psi_\alpha \mathcal{L})$ ▷ Optionally, finetune entire model using $y_{\hat{T}}$ (Eq. (11))
end procedure

using the new data. For posterior probability defined as

$$P(k|f_i, \Lambda) = \frac{\pi_k p(f_i|\mu_k, \sigma_k)}{\sum^K \pi_k p(f_i|\mu_k, \sigma_k)} \tag{7}$$

for the k^{th} mixture and where $p(f|\mu_k, \sigma_k)$ is defined in Eq. (3), we update the mixture parameters in the maximization step in a regularized manner as follows,

$$\hat{\pi}_k = \nu[\psi_k^\pi \frac{1}{n} \sum_{i=1}^{n} P(k|f_i, \Lambda) + (1 - \psi_k^\pi)\pi_k^S] \tag{8}$$

$$\hat{\mu}_k = \psi_k^\mu \mathcal{E}_k + (1 - \psi_k^\mu)\mu_k^S \tag{9}$$

where, $\mathcal{E}_k$ is the first moment or mean of the k^{th} mixture calculated on the new data, ν is a scaling parameter to ensure that $\sum_k \pi_k = 1$ and ψ_k is an adaptation coefficient which is defined for each parameter and mixture. It can be defined in a data dependent manner [29], i.e., $\psi_k^{\mu,\pi} = (\frac{\omega_k}{P(k|f_i,\Lambda)} + 1)^{-1}$ where w_k is an empirically set hyperparameter which controls the adaptation emphasis between source and transitional parameters. Empirically, we observed that the adaptation coefficient is not very sensitive to changes to its value and therefore, we increase it monotonically during the EM iterations. A ψ_k for a specific vMF kernel μ_k at time-step t in $\Lambda^{\mathcal{R}}$ stabilizes if the change in its likelihood component is below a threshold value over the previous EM iteration step t-1 and then ψ_k value. We find that only using the parameter update works well. For simpler datasets, even directly learning the transitional dictionary would suffice.

3.3.2 Learning Transitional Spatial Coefficients

After learning $\Lambda^{\mathcal{R}}$, we use it to estimate the transitional spatial coefficients $(\mathcal{A}^{\mathcal{R}}(\alpha))$ using the labeled source domain features $f^{\mathcal{S}}$ (using MLE). The spatial coefficients represent the expected activation of a calculated vMF kernel μ_k at a position a in the feature map for a specific class y.

$$P_a(f_a|y_s, m; \mathcal{A}^{\mathcal{R}}, \Lambda^R) = \sum_k \alpha_{a,k}^{y_s,m} P(f_a|\Lambda^{\mathcal{R}}(\sigma_k, \mu_k)) \tag{10}$$

We can leverage the learnt transitional vMF kernel dictionary $\Lambda^{\mathcal{R}}$ to learn spatial coefficients $\mathcal{A}^{\mathcal{R}}(\alpha)$ which represent the spatial relationships of the vMF dictionary vectors over the source domain data D_S. As these spatial coefficients $\mathcal{A}^{\mathcal{R}}$ are conditioned on $\Lambda^{\mathcal{R}}$, they also correspond to parts of target domain features even when they are learned using $f^{\mathcal{S}}$, thus creating a transitional model with parameters $(\Lambda^{\mathcal{R}}, \mathcal{A}^{\mathcal{R}})$ that we can use to classify target domain data.

This combination of conditioned transitional vMF dictionary $(\Lambda^{\mathcal{R}})$ and spatial coefficients $(\mathcal{A}^{\mathcal{R}})$ can be leveraged to label a subset of target domain features, especially since we can focus on the subset of transitional vMF kernels $(\Lambda^{\mathcal{R}})$ which are similar to their source counterparts. We can use these pseudo labeled feature vectors $(y_{\hat{T}})$, along with $\Lambda^{\mathcal{R}}$ to finetune the current spatial coefficients $\mathcal{A}^{\mathcal{R}}$ which leads to improved spatial coefficients $\mathcal{A}^{\mathcal{R}'}$.

Finetuning spatial coefficients. Transitional spatial coefficients $(\mathcal{A}^{\mathcal{R}})$ are initialized with the values describing the expected activation of transitional vMF dictionary vectors $\Lambda^{\mathcal{R}}(\mu_k)$ for the source data features $f^{\mathcal{S}}$ at a position a on a feature map f_a. Subsequently, we finetune these spatial coefficients $\mathcal{A}^{\mathcal{R}}$ using a subset of target domain images that present high activations for the robust set of transitional vMF dictionary vectors $\Lambda^{\mathcal{R}}$. Optionally, we can also finetune $\Lambda^{\mathcal{R}}$ by relearning them without any initialization and

regularization constraints. Although our model is trained by partitioning into two parts, it is still fully differentiable and trainable from end to end [20, 36, 42]. We use this model property to finetune the entire model. The loss function (Eq. (11)) consists of a generalized cross entropy [44] term calculated using the model predictions and two regularization parameters for the vMF dictionary and the spatial coefficient parameters. This is to encourage the vMF clusters to be similar to the feature vectors f_a. In Eq. (11), $\zeta_{\{v,\alpha\}}$ represent the trade-off hyperparameters of the regularizing loss terms,

$$\mathcal{L} = \mathcal{L}_{\text{gce}}(y_{pred}, y_{\hat{T}}) + \zeta_v \mathcal{L}(F, \Lambda) + \zeta_\alpha \mathcal{L}(F, \mathcal{A}), \quad (11)$$

For a constant vMF variance σ_k (which also reduces the normalisation term to a constant) and assuming hard assignment of features f_a to vMF dictionary clusters[21],

$$\mathcal{L}(F, \Lambda^{\mathcal{R}}) = -\sum_a \max_k \log p(f_a | \Lambda^{\mathcal{R}}(\mu_k)) \quad (12)$$

$$\mathcal{L}(F, \mathcal{A}^{\mathcal{R}\prime}) = -\sum_a (1 - z_a) \log[\sum_k \alpha_{a,k}^{y_{\hat{T}}, m} p(f_a | \Lambda^{\mathcal{R}}(\mu_k))] \quad (13)$$

Latent variable $z_a \in \{0, 1\}$ is explained in Sec. 3.1.

4. Experiments

Our experiments evaluate *robustness* of vision classification models in an extended out-of-domain setup i.e., generalizing to target domains with individual nuisance factors and partial occlusion. This allows us to thoroughly evaluate the efficacy of current methods which have been shown to perform well on other OOD robustness datasets on OOD-CV[45] (which enables a systematic analysis of nuisances on real-world data), Occluded-OOD-CV (which allows us to evaluate models on a combination of partial occlusion with individual nuisances) and Imagenet-C corruptions (for analysis of synthetic corruptions). Lastly, we also show some initial results on Synthetic to Real OOD robustness using the UDAParts [24] dataset.

4.1. Setup and Data

Datasets. For primary evaluation, we use the OOD-CV [45] dataset. OOD-CV dataset consists of test subcategories which vary from the training data in terms of a main nuisance factor, namely, context, weather, texture, pose and shape. We use $L0$ for the (0%) occlusion level to represent this data setup in Tab. 1 and Supplementary Sec. B.

Occluded-OOD-CV. In addition to OOD-CV, we experiment with a more complex robustness analysis setup involving partial occlusion. In this setup, models that have been adapted in an unsupervised manner to target domains with nuisance factors are then evaluated on data with partial occlusion in addition to the real-world nuisances. For this purpose, we create a dataset named *Occluded-OOD-CV* where we superimpose occluders on the OOD-CV test images objects in order to approximate real-life occlusion. These occluders have been cropped from the MS-COCO dataset, similar to [20] and are superimposed on objects in the OOD-CV test set. There are three levels of partial occlusions - $L1(20-40\%), L2(40-60\%)$ and $L3(60-80\%)$ which allows us to diversely analyze the occlusion robustness of the model (in addition to individual nuisance factors). Fig. 3 shows some example images from our dataset. Previous works [18, 21] have shown that using cropped occluders, as done in Occluded-OOD-CV, is akin to the use of real occluders for classification evaluation. We also use

Figure 3. Occluded-OOD-CV dataset examples. Each object category is identified by its nuisance factor and occlusion percentage

Imagenet-C[9] corruptions in the Pascal3D+ dataset for robustness evaluation with conventionally used synthetic corruptions. We also evaluate models in a synthetic (UDA-Parts [24]) to real data (Pascal3D+ [41]) setup.

In summary, we have 5 different real world nuisance data subcategories (context, weather, texture, pose, shape), at least seven synthetic corruption categories (fog, pixelate, motion blur, etc.), one synthetic source dataset and 4 partial occlusion levels (including no occlusion) for each experiment. We also run experiments on all the combined nuisance subcategories (Tab. 1). So, in total we have 24 sets of data and experiments for our (extended) OOD robustness setup on the OOD-CV dataset alone.

Models. We compare our work with our baseline method CompNets [21], other well known recent works [30, 32] which have been shown to be SOTA on various *robustness* datasets [9, 11, 12] as well as many well-known UDA methods [3, 13, 15, 22, 23, 25–27, 40, 43]. We focus on VGG16 and Resnet-50 backbones as they have been commonly used in most current methods[20, 30, 32, 44].

Training Setup. All models are trained on the source data with corresponding labels. Models can access some unlabeled nuisance (target) data, which could be a single nuisance (OOD-CV, Imagenet-C), combined nuisances (Tab. 1) or real data (when source data are synthetic). Models do not have access to images with partial occlusion at any time, and partially occluded images are only used for inference. We also *avoid* using different types of data aug-

Table 1. OOD-CV Nuisances Top-1 Classification Results. Occlusion levels greater than 0% represent Occluded-OOD-CV dataset.

Method	**Combined**				**Context**				**Weather**			
Occlusion→	**0%**	**20-40%**	**40-60%**	**60-80%**	**0%**	**20-40%**	**40-60%**	**60-80%**	**0%**	**20-40%**	**40-60%**	**60-80%**
CDAN [25]**	.760	.531	.420	.380	.710	.541	.436	.397	.745	.476	.335	.299
BSP [2]**	.753	.506	.401	.351	.610	.511	.419	.385	.730	.391	.266	.254
MDD [43]**	.780	.551	469	.410	.761	.531	.436	.410	.802	.439	.306	.271
MCD [31]**	.772	.556	.461	.403	.798	.523	.426	.374	.810	.447	.336	.286
MCC [15]**	.785	.582	.492	.434	.730	.577	.454	.420	.767	.503	.376	.362
FixBi [27]**	.821	.534	.478	.399	.802	.542	.445	.409	.755	.489	.358	.335
MIC [13]**	.837	.540	.376	.262	.755	.602	.532	.499	.817	**.612**	.496	.427
ToAlign [40]**	.761	.507	.411	.346	.712	.501	.393	.382	.720	.381	.252	.213
CST [23]**	.840	.579	.539	.477	.687	.491	.452	.411	.813	.558	.397	.356
DUA [26]**	.699	.523	.480	.403	.667	.471	.434	.401	.701	.465	.391	.210
DINE [22]**	.835	.600	.493	.443	.867	.515	.418	.397	.798	.423	.290	.261
RPL [30]	.664	.430	.346	.300	.675	.457	.368	.315	.642	.247	.138	.122
BNA [32]	.653	.426	.343	.298	.580	.397	.342	.278	.635	.295	.179	.171
CompNet [21]	.720	.506	.462	.415	.790	.517	.454	.369	.683	.434	.398	.362
UGT (Ours)	**.850**	**.620**	**.570**	**.501**	**.875**	**.624**	**.565**	**.511**	**.856**	.600	**.528**	**.465**
	Texture				**Pose**				**Shape**			
CDAN [25]**	.820	.532	.420	.364	.844	.620	.521	.450	.773	.561	.491	.441
BSP [2]**	.696	.444	.384	.315	.831	.610	.510	.423	.757	.535	.485	.434
MDD [43]**	.895	.518	.427	.400	.870	.611	.534	.469	.836	.541	.459	.386
MCD [31]**	.896	.522	.432	.392	.865	.623	.532	.471	.834	.538	.456	.397
MCC [15]**	.874	.671	.547	.495	.867	.611	.521	.460	.818	.601	.524	.460
FixBi [27]**	.854	.574	.445	.369	.842	.533	.472	.446	.801	.500	.435	.373
MIC [13]**	.821	.706	.631	.576	.799	.613	.509	.455	.807	.608	.565	.467
ToAlign [40]**	.594	.413	.312	.273	.788	.574	.503	.418	.719	.548	.460	.391
CST [23]**	.858	.657	.538	.477	.887	.617	.525	.451	.831	.617	.495	.441
DUA [26]**	.918	.691	.514	.468	.755	.511	.423	.355	.695	.455	.386	.345
DINE [22]**	.911	.572	.432	.401	.885	.618	.543	.448	.838	.520	.426	.360
RPL [30]	.703	.371	.238	.227	.730	.493	.400	.329	.670	.426	.340	.311
BNA [32]	.701	.383	.247	.239	.737	.510	.407	.355	.662	.436	.350	.311
CompNet [21]	.747	.539	.462	.426	.768	.581	.538	.458	.698	.466	.451	.400
UGT (Ours)	**.936**	**.726**	**.665**	**.635**	**.892**	**.632**	**.555**	**.481**	**.852**	**.644**	**.601**	**.567**

** **Pretrained Imagenet Backbone used (Resnet-50) / Pretrained UDA model used**.

Table 2. Imagenet-C Corruptions on Pascal3D+ dataset - Classification Results (Vgg16)

Model	**Elastic Transform**				**Gaussian Blur**				**Snow**			
Occlusion→	**0%**	**20-40%**	**40-60%**	**60-80%**	**0%**	**20-40%**	**40-60%**	**60-80%**	**0%**	**20-40%**	**40-60%**	**60-80%**
RPL [30]	.830	.597	.461	.371	.855	.541	.403	.320	.842	.592	.435	.408
BNA [32]	.793	.601	.498	.400	.833	.618	.484	.300	.767	.627	.542	.454
CompNet [21]	.268	.183	.157	.146	.732	.395	.296	.241	.529	.348	.258	.210
UGT (Ours)	**.872**	**.712**	**.712**	**.494**	**.909**	**.720**	**.613**	**.509**	**.890**	**.742**	**.634**	**.523**
	Motion Blur				**Contrast**				**Frost**			
RPL [30]	.862	.629	.481	.373	.901	.610	.433	.321	.850	.670	.511	.402
BNA [32]	.844	.623	.481	.355	.899	.601	.401	.315	.845	.654	.501	.399
CompNet [21]	.639	.362	.287	.241	.760	.472	.374	.312	.740	.481	.360	.301
UGT (Ours)	**.891**	**.763**	**.673**	**.567**	**.923**	**.701**	**.534**	**.412**	**.911**	**.782**	**.672**	**.561**

Table 3. Ablation analysis for (a) OOD-CV [45] Combined (b) OOD-CV Texture (c) Imagenet-C (Snow) Corruption

Occlusion→	L0	L1	L2	L3	L0	L1	L2	L3	L0	L1	L2	L3
Baseline(B)	.698	.466	.451	.400	.715	.575	.475	.409	.529	.348	.258	.210
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}}$	.816	.598	.524	.498	.785	.660	.559	.515	.781	.671	.582	.480
$+\Lambda^{\mathcal{R}} + \mathcal{A}^{\mathcal{R}'}$	.852	.644	.601	.567	.843	.764	.656	.623	.885	.742	.634	.523

mentations and additional data training to have a fairer comparison amongst all the works. Although, our Bayesian model does not use **pretrained Imagenet backbones** for feature extraction for fairness, a number of our comparative methods [2, 15, 25, 26, 43] perform poorly without one, so we relax this constraint for them. Our method is still capable of surpassing them in terms of classification accuracy. Further details are provided in Supplementary Section C.

4.2. Results

OOD robustness to individual nuisances. Tab. 1 (L0 columns) shows classification results on entire OOD-CV test data (combined nuisances) as well as five individual nuisances. We see that our model achieves state-of-the-art results in all experiments. In Tab. 2, we observe that our model also performs exceedingly well when dealing with synthetic Imagenet-C corruptions. Refer to Supplementary Sec.C2 and Tables 5-11 for additional Imagenet-C results.

Synthetic to Real. Tab. 4 shows our results on both normal and extended OOD robustness scenario in a synthetic to real setup, showing that our unsupervised method can robustly close the gap between its supervised counterpart while outperforming other methods by large margins.

Table 4. Synthetic (UDAParts) [24] to Real (Pascal3D+) [41] dataset - Classification Results on Resnet50

Model	**0%**	**20-40%**	**40-60%**	**60-80%**
RPL [30]	.822	.432	.370	.335
BNA [32]	.950	.684	.484	.356
CompNet [21]	.940	.650	.475	.347
UGT (Ours)	**.992**	**.957**	**.861**	**.753**

Extended OOD robustness under partial Occlusion. In Tab. 1, Tab. 2 and Supplementary Tables 1-3, 5-11, our model outperforms other methods by significant margins in the *extended OOD* scenarios of nuisance parameters with partial occlusion. We observe that the performance of other models which have been adapted to the target domain data drops drastically when encountering partial occlusion along with nuisance factors. This underlines the increased complexity of the extended OOD robustness scenario relative to the vanilla OOD robustness setup and how our Bayesian model is able to perform exceedingly well compared to conventional methods.

4.3. Ablation Analysis

Tab. 3 and Supplementary Sec. D & Tables 12-17 show the extensive results of the ablation study for UGT, underlying how each component contributes to the overall compositional model. We can see that just calculating the transitional vMF kernel dictionary ($\Lambda^{\mathcal{R}}$) and the transitional spatial coefficients $\mathcal{A}^{\mathcal{R}}$ improves the results significantly over the baseline method[21]. Further finetuning the spatial coefficients ($\mathcal{A}^{\mathcal{R}'}$) using pseudo-labelled target domain features boosts the performance. We ablate our hypothesis regarding similar vMF kernels in source and target domains by visualizing image patches that are activated by similar cross-domain kernels (Supplementary Figures 9-11). We also ablate our hypothesis regarding robust spatial geometry by visualizing images activated by the same spatial coefficient in both source and target domains (using source and transitional vMF dictionaries) in Supp. Fig 4 and 7. Analysis of adaptation coefficient is discussed in Supp. Sec. E.

5. Conclusion and Future Work

In this work, we addressed the problem of developing object classification algorithms that are robust to OOD factors such as weather, context and occlusion. We generalize CompNets[21] for OOD robustness by observing that they could be learned in two uncoupled steps: (i) unsupervised learning of a dictionary of vMF kernels (roughly corresponding to the subparts of the object) and (ii) supervised learning of the spatial structure of the objects (intuitively where the subparts occur). This enabled us to: (a) learn a transitional dictionary which captured the feature properties of both domains, and (b) learn the distribution of spatial structure on the source domain and transfer it to the target. This model is very successful and could be improved by simple pseudo-labeling techniques. Our empirical results on the OOD-CV[45], synthetic Imagenet-C corruptions, and the synthetic UDA-Parts dataset display the strong and versatile SOTA performance of our method. In addition, we developed a more challenging dataset Occluded-OOD-CV by introducing occlusion into OOD-CV and show that our Bayesian method, UGT, performed well in this difficult challenge. Our Bayesian approach could be extended to other tasks such as semantic segmentation, exploiting properties of CompNets[36, 37] . We give a qualitative proof of concept in the Supplementary.

References

[1] D. Bug, S. Schneider, A. Grote, E. Oswald, F. Feuerhake, J. Schüler, and D. Merhof. Context-based normalization of histological stains using deep convolutional features. *Lecture Notes in Computer Science*, pages 135–142, 2017. 2

[2] 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. 7, 8

[3] Shuhao Cui, Shuhui Wang, Junbao Zhuo, Liang Li, Qingming Huang, and Qi Tian. Towards discriminability and diversity: Batch nuclear-norm maximization under label insufficient situations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[4] Nathan Drenkow, Numair Sani, Ilya Shpitser, and Mathias Unberath. Robustness in deep learning for computer vision: Mind the gap? *CoRR*, abs/2112.00639, 2021. 1

[5] Aram Galstyan and Paul R Cohen. Empirical comparison of "hard" and "soft" label propagation for relational classification. In *International Conference on Inductive Logic Programming*, pages 98–111. Springer, Berlin, Heidelberg, 2007. 2

[6] Aritra Ghosh, Himanshu Kumar, and P. S. Sastry. Robust loss functions under label noise for deep neural networks, 2017. 2

[7] Siddharth Gopal and Yiming Yang. Von mises-fisher clustering models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 154–162, Bejing, China, 2014. PMLR. 3

[8] Md Hasnat, Julien Bohné, Jonathan Milgram, Stéphane Gentric, Liming Chen, et al. von mises-fisher mixture model-based deep learning: Application to face verification. *arXiv preprint arXiv:1706.04264*, 2017. 3

[9] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations, 2019. 1, 2, 6

[10] Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. Pretrained transformers improve out-of-distribution robustness. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[11] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization, 2021. 1, 2, 6

[12] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples, 2021. 1, 2, 6

[13] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation, 2023. 6, 7

[14] Dong hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. 2

[15] Ying Jin, Ximei Wang, Mingsheng Long, and Jianmin Wang. Minimum class confusion for versatile domain adaptation, 2019. 6, 7, 8

[16] Prakhar Kaushik, Aayush Mishra, Adam Kortylewski, and Alan Yuille. Source-free and image-only unsupervised domain adaptation for category level object pose estimation, 2024. 2

[17] Toru Kitagawa and Jeff Rowley. von mises-fisher distributions and their statistical divergence, 2022. 1

[18] Muhammed Kocabas, Chun-Hao P. Huang, Otmar Hilliges, and Michael J. Black. Pare: Part attention regressor for 3d human body estimation, 2021. 6

[19] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 3

[20] 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. 1, 3, 6

[21] 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. 1, 2, 3, 4, 6, 7, 8

[22] Jian Liang, Dapeng Hu, Jiashi Feng, and Ran He. Dine: Domain adaptation from single and multiple black-box predictors, 2022. 6, 7

[23] Hong Liu, Jianmin Wang, and Mingsheng Long. Cycle self-training for domain adaptation, 2021. 6, 7

[24] 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. 2, 6, 8

[25] 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. 6, 7, 8

[26] M. Jehanzeb Mirza, Jakub Micorek, Horst Possegger, and Horst Bischof. The norm must go on: Dynamic unsupervised domain adaptation by normalization, 2022. 7, 8

[27] Jaemin Na, Heechul Jung, Hyung Jin Chang, and Wonjun Hwang. Fixbi: Bridging domain spaces for unsupervised domain adaptation, 2021. 6, 7

[28] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet?, 2019. 1, 2

[29] Douglas A. Reynolds, Thomas F. Quatieri, and Robert B. Dunn. Speaker verification using adapted gaussian mixture models. *Digital Signal Processing*, 10(1):19–41, 2000. 5

[30] 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. 6, 7, 8

[31] 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. 7

[32] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation, 2020. 2, 6, 7, 8

[33] Ozan Sener, Hyun Oh Song, Ashutosh Saxena, and Silvio Savarese. Learning transferrable representations for unsupervised domain adaptation. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016. 2

[34] Hwanjun Song, Minseok Kim, Dongmin Park, Yooju Shin, and Jae-Gil Lee. Learning from noisy labels with deep neural networks: A survey, 2020. 2

[35] Baochen Sun, Jiashi Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. *Advances in Computer Vision and Pattern Recognition*, pages 153–171, 2017. 2

[36] Yihong Sun, Adam Kortylewski, and Alan Yuille. Weakly-supervised amodal instance segmentation with compositional priors. *arXiv preprint arXiv:2010.13175*, 2020. 1, 2, 3, 6, 8

[37] 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. 8

[38] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization, 2020. 2

[39] Jianyu Wang, Zhishuai Zhang, Cihang Xie, Yuyin Zhou, Vittal Premachandran, Jun Zhu, Lingxi Xie, and Alan Yuille. Visual concepts and compositional voting, 2017. 2, 3, 4

[40] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. Toalign: Task-oriented alignment for unsupervised domain adaptation, 2021. 6, 7

[41] 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. 6, 8

[42] Xiaoding Yuan, Adam Kortylewski, Yihong Sun, and Alan Yuille. Robust instance segmentation through reasoning about multi-object occlusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11141–11150, 2021. 1, 3, 6

[43] Yuchen Zhang, Tianle Liu, Mingsheng Long, and Michael I. Jordan. Bridging theory and algorithm for domain adaptation, 2019. 6, 7, 8

[44] Zhilu Zhang and Mert R. Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels, 2018. 2, 6

[45] 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, 2, 6, 8