Robustness via Cross-Domain Ensembles

Teresa Yeo * Oğuzhan Fatih Kar * Amir Zamir
Swiss Federal Institute of Technology (EPFL)

http://crossdomain-ensembles.epfl.ch/

Abstract

We present a method for making neural network predictions robust to shifts from the training data distribution. The proposed method is based on making predictions via a diverse set of cues (called 'middle domains') and ensembling them into one strong prediction. The premise of the idea is that predictions made via different cues respond differently to a distribution shift, hence one should be able to merge them into one robust final prediction. We perform the merging in a straightforward but principled manner based on the uncertainty associated with each prediction. The evaluations are performed using multiple tasks and datasets (Taskonomy, Replica, ImageNet, CIFAR) under a wide range of adversarial and non-adversarial distribution shifts which demonstrate the proposed method is considerably more robust than its standard learning counterpart, conventional deep ensembles, and several other baselines.

1. Introduction

Neural networks deployed in the real world will encounter data with naturally occurring distortions, e.g. motion blur, or adversarial ones. Such changes make up shifts from the training data distribution. While neural networks are able to learn complex functions in-distribution, their predictions are profoundly unreliable under such shifts [9, 20, 50, 25]. This presents a core challenge that needs to be solved for these models to be useful in the real world.

Suppose we want to learn a mapping from an input domain, e.g. RGB images, to a target domain, e.g. surface normals (see Fig. 1). A common approach is to learn this mapping with a *direct* path, i.e. *RGB* → *surface normals*. Since this path directly operates on the input domain, it is prone to being affected by any slight alterations in the RGB image, e.g. brightness changes. An alternative can be to go through a *middle domain*[1] that is invariant to that change. For example, the surface normals predicted via the *RGB* → *2D edges* → *surface normals* path will be resilient to brightness distortions in the input as the 2D edges domain abstracts that away. However, one does not know which middle-domain to use ahead of time as the distortions that a model may encounter are broad and apriori unknown, and some middle domains can be too lossy for certain downstream predictions. These issues can be mitigated by employing an *ensembling* approach where predictions made via a diverse *set* of middle domains are merged into one strong prediction on-the-fly.

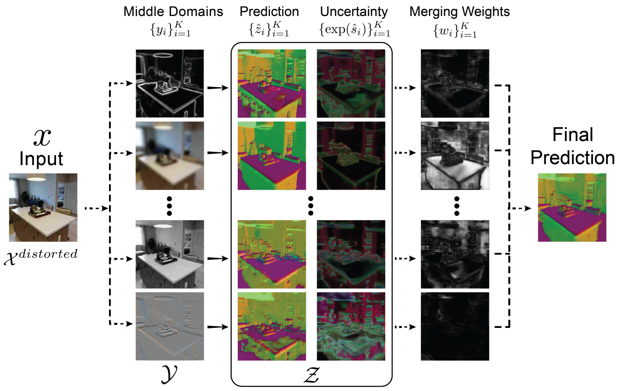


Figure 1: **An overview of the proposed method for creating a robust and diverse ensemble of predictions.** A set of K networks predict a target domain (here surface normals) given an input image that has undergone an unknown distribution shift (here JPEG compression degradation), via K middle domains (e.g. 2D texture edges, low-pass filtering, greyscale image, emboss filtering, etc). The predictions from the K paths are then merged into one final strong prediction using weights that are based on the uncertainty associated with each prediction. This method is shown to be significantly robust against adversarial and non-adversarial distribution shifts for several tasks. In the figure above, solid (–) and dashed (- -) arrows represent learned and analytical functions, respectively.

This paper presents a general approach for the aforementioned process. We first use a set of K middle domains from which we learn to predict the final domain (Fig. 1). Each of the K paths reacts differently to a particular distribution shift due to its inherent biases, so its prediction may or may not degrade severely. Thus, we further estimate the *uncer-*

[1] or equivalently "middle task", as most vision tasks can be viewed as mapping an input onto some other domain.

* Equal contribution.

tainty of each path's prediction which allows us to employ a principled way of combining these predictions into the one final prediction.

Prior knowledge of the relationship between middle domains is not needed as their contribution to the final prediction is guided by their predicted uncertainties in a fully computational manner independent of the definition of the middle domain. In other words, no manual modification or re-design is needed upon a change in these domains. Moreover, the middle domains we adopt can all be *programmatically extracted*. Thus, this framework does not require any additional supervision/labeling than what a dataset already comes with. The proposed method would be equally applicable if the middle domains were also obtained using a learning based approach, e.g. predicting surface normals from the output of another network such as a depth estimator. We show in Sec. 4 that the method performs well insensitive to the choice of middle domains and it generalizes to completely novel non-adversarial and adversarial corruptions.

2. Related Work

This work has connections to a number of topics, including ensembling, uncertainty estimation and calibration, inductive bias learning [4], or works in neuroscience that suggest the brain uses multiple, sometimes partially redundant, cues to perceive [23, 24]. We give an overview of some of them within the constraints of space.

Ensembling allows us to resolve the bias-variance tradeoff which states that errors in a models' prediction can be decomposed into bias, variance, and an irreducible data-dependent noise term [14, 11]. This is done by combining multiple models with low bias and high variance, e.g. bagging [8], or with high bias and low variance, e.g. boosting [8], to have predictions with both low bias *and* variance.

A primary challenge for ensemble methods is to ensure diversity. Sources of diversity include using different initializations [33], hyperparameters [53] or network architectures [58] for the ensemble components, or training the ensemble with additional loss terms [41, 26, 56]. However, under distribution shifts, reduction in performance can stem from an increase in the bias, rather than the variance term [57]. Our set of middle domains yields a more diverse ensemble by design and promotes invariance to different distortions to keep bias low (see Fig. 1).

Estimating uncertainty: Uncertainty in a model's prediction can be decomposed into two sources [7, 27]. *Epistemic* uncertainty accounts for uncertainty in the model parameters, while *aleatoric* uncertainty stems from the noise inherent in the data. There are many proposed methods to estimate the former, such as using dropout [12, 48], stochastic variational inference methods [5, 15, 38, 37, 52, 43], ensembling [33], and consistency energy [59] where a *single* uncalibrated uncertainty estimate is extracted from consistency of different paths. Most of the existing methods in this area solely estimate uncertainty without using it towards improving the predictions. In contrast, our formulation estimates a calibrated uncertainty for each path *and* uses it to produce a stronger prediction.

Improving calibration with auxiliary datasets: Neural networks tend to produce outputs that are miscalibrated, i.e. their estimated uncertainty does not reflect the true likelihood of being correct [16, 31]. In particular, their predictions tend to be *overconfident* for unfamiliar examples. This is usually handled by a calibration step. Similar to [17, 21, 36, 34], we use a separate dataset from the one at test time to train the model to output high sigmas (uncertainties) for unfamiliar cases. Previous papers focus on generalizing uncertainties for classification; in Section 3.1, we show this can be extended to dense regression problems.

Enforcing consistency constraints in the context of cross-task predictions involves ensuring that the output predictions remain the same regardless of the intermediate domain [59, 35, 62, 55]. Particularly in contrast to [59] which uses (non-probabilistic) training-time consistency constraints to improve a network's prediction and does not have any consolidation mechanism, our goal is to robustify the final prediction by *merging the output of multiple prediction paths at the test time*. Our formulation and the *training-time* consistency constraints are complimentary.

Robustness via data augmentation: One approach to addressing robustness involves using data augmentation during training [39, 61, 22, 28, 2]. Such methods usually involve *training with a set of corruptions* to generalize to unseen ones [45]. However, performance gains can be non-uniform, e.g. Gaussian noise augmentation improves performance on other noise corruptions (e.g. impulse, shot noise) but hurts performance on fog and contrast [10]. Instead, our main mechanism uses a large set of middle domains (not corruptions) to be resistant to a wide range of diverse *unseen* corruptions. We do not use any corruptions during training, except to calibrate uncertainty.

Adversarial attacks add imperceptible worst case shifts to the input to fool a model [50, 32, 39]. In contrast to [41, 26, 56], which are ensemble based adversarial robustness methods with an additional loss term to promote diversity, the diversity of our ensembles is a natural consequence of using different middle domains. While our focus is not limited to robustness against adversarial attacks, it yields supportive evaluations against them as well (Sec. 4.1.2).

3. Method

We explain the technical details of our method below.

Notations: Define $\mathcal{X}$ as the RGB domain, $\mathcal{Y} = \{\mathcal{Y}_j\}_{j=1}^K$ as the K intermediate domains, $\mathcal{Z}$ as the desired prediction domain. A single datapoint n from these K domains is denoted as $(x_n, y_{1,n}, \ldots, y_{j,n}, \ldots, y_{K,n}, z_n)$. $\mathcal{F}_{\mathcal{X}\mathcal{Y}}$ is the set of functions that maps the RGB images to their intermedi-

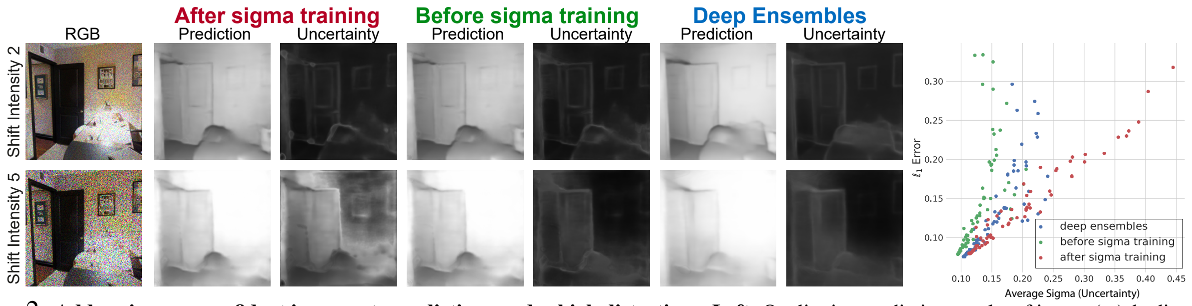


Figure 2: **Addressing overconfident inaccurate predictions under high distortions. Left**: Qualitative prediction results of image (re)shading and their corresponding uncertainty estimates (i.e. sigma) under two intensities of speckle noise distortion. This is shown for a single UNet model before and after *sigma training* (ST) as well as for deep ensembles (standard deviation of predictions in the ensemble). Darker denotes lower uncertainty/sigma. ST was done using Gaussian noise and Gaussian blur distortions. *Using other distortions yields similar performance* (see supplementary). **Right**: Scatter plot of ℓ_1 error versus average sigma. Each point is computed from an average over 16k test images for one of the *unseen* distortions and one of 5 levels of shift intensity. Notice (qualitatively and quantitatively) that when the models without ST produce poor results, their uncertainty does *not* correspondingly increase. Our ST helps the model to have a stronger correlation between its uncertainty estimates and error when *tested on unseen distortions*. This indicates that sigma *after ST* can be an effective signal for merging multiple predictions. Note that the predicted mean ("Prediction") *does not change* with ST.

ate domains, $\mathcal{F}_{\mathcal{X}\mathcal{Y}} = \{f_j : \mathcal{X} \rightarrow \mathcal{Y}_j\}_{j=1}^{K}$, and $\mathcal{F}_{\mathcal{Y}\mathcal{Z}}$ is the set of functions mapping from the intermediate to the target prediction domain, $\mathcal{F}_{\mathcal{Y}\mathcal{Z}} = \{g_j : \mathcal{Y}_j \rightarrow \mathcal{Z}\}_{j=1}^{K}$. Given K predictions of domain $\mathcal{Z}$, they are merged using the function m to get a final single prediction, $m : \{g_j(\mathcal{Y}_j)\}_{j=1}^{K} \rightarrow \mathcal{Z}$.

3.1. Estimating Per-Path Predictions and Uncertainty

We learn the mappings g_j using a neural network. We model the noise in the predictions made by g_j with a Laplace distribution. Thus, for an input sample $y_{j,n}$, the network outputs two sets of parameters $[\hat{z}_{j,n}, \hat{s}_{j,n}] = g_j(y_{j,n})$ where we set $\hat{s}_{j,n} = \log \hat{b}_{j,n}$ for numerical stability and $\hat{b}_{j,n}$ is the scale parameter of the Laplace distribution. We remove the dependence on j for brevity. This leads to the following negative log-likelihood (NLL) loss for g:

$$\mathcal{L}_{g,NLL} = \frac{1}{N} \sum_{n=1}^{N} \exp\left(-\hat{s}_n\right) \|\hat{z}_n - z_n\|_1 + \hat{s}_n, \quad (1)$$

where N is the number of samples, and z_n is the label for the nth sample. This results in an ℓ_1-norm loss on the errors as opposed to an ℓ_2-norm loss with a Gaussian distribution, which has been shown to improve prediction quality [27, 59]. Finally, the *sigma* is given by $\sqrt{2}\exp(\hat{s}_n)$ and it captures the uncertainty in predictions.

Calibration via Sigma training (ST): Uncertainty estimates under distribution shifts are poorly calibrated [40], i.e. there is a tendency to output a poor prediction with high confidence. This can be seen in Fig. 2, "Before sigma training" columns. With a higher noise distortion, the prediction clearly degraded, but the uncertainty estimate did not increase correspondingly. This issue persists even with methods that estimate epistemic uncertainty (Fig. 2, "Deep Ensembles" columns) which are meant to detect these shifts.

To mitigate this, we adopt a two-stage training setup where the network trained on in-distribution data is further trained to output high uncertainty outside the training distribution. We denote this step as *sigma training* (ST). Here, $\hat{s}_n$ is trained to learn its maximum likelihood estimator, with the loss denoted as *sigma calibration* (SC). As the goal of this step is to maximize the likelihood by correcting the sigma $\hat{s}_n$, and not the mean $\hat{z}_n$, under a distortion (*dist*), we add a loss term to ensure that $\hat{z}_n$ does not deviate from its predictions at the start of ST, which we define as $\hat{z}_n^0$. We denote this loss as *mean grounding* (MG). Finally, we include the original NLL from Eq. 1 on undistorted data (*undist*) to prevent forgetting. This results in the following loss formulation:

$$\mathcal{L}_{g,ST} = \mathcal{L}_{g,NLL}^{undist} + \alpha_1 \mathcal{L}_{g,MG}^{dist} + \alpha_2 \mathcal{L}_{g,SC}^{dist}, \quad (2)$$

where α_1, α_2 controls the weighting between the loss terms. For a given $\hat{z}_n^0$, the MG loss is defined as the ℓ_1-norm distance between the current prediction and the one at the start of sigma training, i.e. $\mathcal{L}_{g,MG}^{dist} = \|\hat{z}_n^0 - \hat{z}_n\|_1$. The SC loss guides the scale parameter towards its maximum likelihood estimate, i.e. $\mathcal{L}_{g,SC}^{dist} = \|\exp\left(\hat{s}_n\right) - \arg\min_{\hat{s}_n} \mathcal{L}_{g_j,NLL}^{dist}\|_1 = \|\exp\left(\hat{s}_n\right) - |z_n - \hat{z}_n^0|\|_1$.

Following ST, the network outputs sigmas that are highly correlated with error (Fig. 2, rightmost plot). Given multiple predictions of the same target domain and their sigma estimates, this allows us to use the latter as a signal for merging to get a single strong prediction (Sec. 3.2).

As the objective of ST is to expose the network to inputs with high distortions as opposed to updating the final predicted mean, *any* corruption with high intensity will suffice. The distortions used for ST are not the same distortions as the ones at test time. Please see supplementary Section 2.5 for a detailed study. Furthermore, the experiments (Fig. 2, Fig. 6, Table 1) indicate that sigma clearly generalizes to unseen distortions.

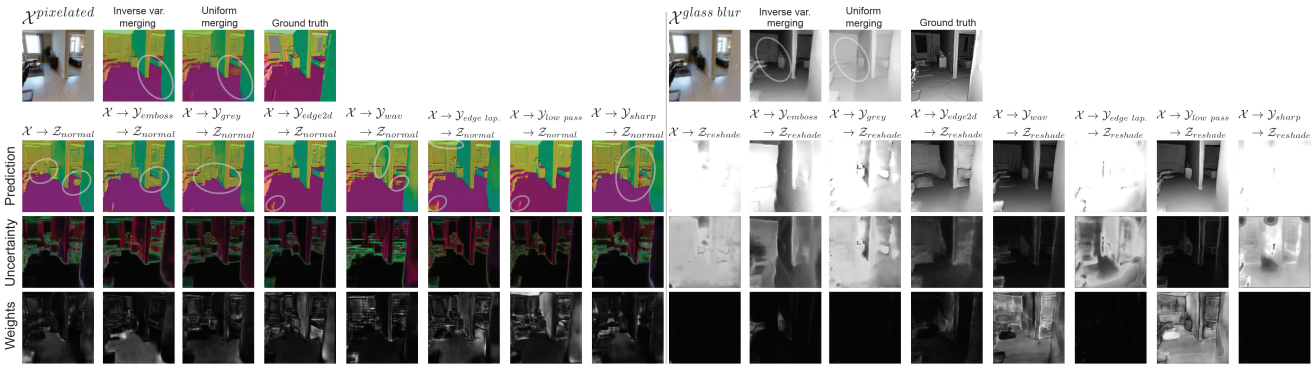

$\mathcal{X}^{pixelated}$
Inverse var. merging
Uniform merging
Ground truth
$\mathcal{X} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{emboss} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{grey} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{edge2d} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{wav} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{edge\ lap.} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{low\ pass} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X} \rightarrow \mathcal{Y}_{sharp} \rightarrow \mathcal{Z}_{normal}$
$\mathcal{X}^{glass\ blur}$
Inverse var. merging
Uniform merging
Ground truth
$\mathcal{X} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{emboss} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{grey} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{edge2d} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{wav} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{edge\ lap.} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{low\ pass} \rightarrow \mathcal{Z}_{reshade}$
$\mathcal{X} \rightarrow \mathcal{Y}_{sharp} \rightarrow \mathcal{Z}_{reshade}$
Prediction
Uncertainty
Weights

Figure 3: **How does the method work?** Each network in each path receives different cues for making a prediction, due to going through different middle domains. **Left**: Given a distorted pixelated query, each path (columns) is affected differently by the distortion, which is reflected in its prediction, uncertainty, and weights (lighter means higher weights/uncertainty). The inverse variance merging uses the weights to assemble a final prediction that is better than each of the individual predictions. (The uncertainties of surface normals look colorful as surface normals domain includes 3 channels, thus there are 3 uncertainty channels.) **Right**: Similarly, for a query with glass blur distortion, the method successfully disregards the degraded predictions and assembles an accurate final prediction. Note that the proposed method (*inverse var. merging*) obtains significantly better results than learning from the RGB directly (leftmost column of each example) which is the most common approach. The quality of the final prediction depends on the following elements: **1.** For each pixel, at least one middle domain is robust against the encountered distortion, and **2.** The uncertainty estimates are well correlated with error, allowing the merger to select regions from the best performing paths. *Uniform merging* does not take into account the uncertainties and consequently lead to worse predictions. The elliptical markers denote sample regions where the merged result is better than all individual predictions.

3.2. Merging Predictions

After obtaining the set of mappings $\mathcal{F}_{\mathcal{XY}}$ and $\mathcal{F}_{\mathcal{YZ}}$ with the method described above, it remains to merge the predictions coming from multiple paths using a merging function m. We employ an analytical approach given by $m(\{g_j(y_{j,n})\}_{j=1}^{K}) = C\sum_{j=1}^{K}\exp(-2\hat{s}_{j,n})\hat{z}_{j,n}$ where C is a normalizing constant defined as $C = (\sum_{j=1}^{K}\exp(-2\hat{s}_{j,n}))^{-1}$. This performs a straightforward weighting of each pixel in each path by the inverse of its variance [18] which can be done with negligible computational cost. We denoted this as *Inverse variance merging* and will show in Section 4.1 that it performs better than other analytical and learning based variants of our method.

The algorithm 1 summarizes our training procedure.

Algorithm 1 Summary of the training procedure of our method

```
Require: Define $f_j \in \mathcal{F}_{\mathcal{XY}}$ and $g_j \in \mathcal{F}_{\mathcal{YZ}}$ $\forall j$.
for $j = 1 : K$ do
    Train $g_j$ using NLL loss in Eq. 1.
    (Optional) Train $g_j$ using consistency constraints [59]. (Sec. 4.1)
    Perform sigma training over $g_j$ with Eq. 2.
end for
Merge the $K$ predictions from the $\mathcal{F}_{\mathcal{YZ}}$ networks using Inv. var. merging (Sec. 3.2).
```

A working example. Figure 3 illustrates our method with an example. For a given image, each path's prediction, uncertainty, and corresponding weights are shown. For the distorted (pixelated) query in the left, each path reacted differently to the distortion, and the final prediction is obtained by combining individual predictions based on their uncertainties. Similar observations can be made for the glass blurred image in the right, where the method learned weights in a way such that the degraded paths are not used in the final prediction. We also show the final prediction from a uniform average of each path. While it is better than simply using the direct path ($\mathcal{X} \rightarrow \mathcal{Z}_{normal}$ or $\mathcal{X} \rightarrow \mathcal{Z}_{reshade}$), using the uncertainty estimates as weights results in a notably more accurate prediction.

There are two key elements to the effectiveness of our method. **I.** With a diverse set of middle domains, it is more likely that one of them will be less affected by distortions and returns an accurate prediction. **II.** The error of the prediction correlates well with its corresponding uncertainty estimates, i.e. the uncertainty is low in the region of the image where the prediction is accurate. This allows us to use these uncertainty estimates as a signal to have a final prediction with parts of the image taken from different paths.

4. Experiments

We demonstrate that the proposed approach leads to robustness against different *distribution shifts*, over different *datasets*, and different *prediction tasks*. For pixel-wise prediction tasks, we train on the Taskonomy dataset [60]. To evaluate the robustness under corruptions, we report performance under Common Corruptions [20] and adversarial perturbations [50, 32, 39]. To evaluate against dataset shifts, we report on Replica [49] and Habitat [47] datasets. For classification, we train on ImageNet [46], CIFAR [29], and evaluate on ImageNet-C and CIFAR-C [20]. Please see the supplementary and project page for more extensive qualitative results.

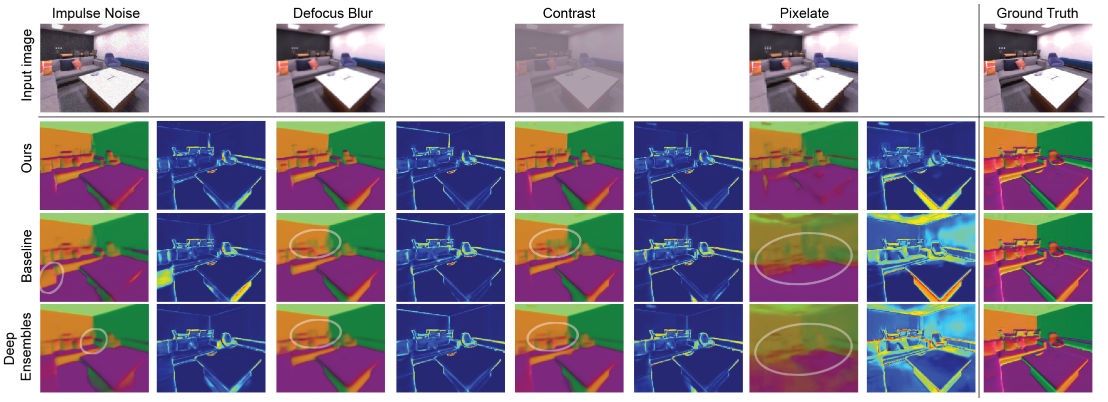


Figure 4: **Qualitative prediction results under 4 distribution shifts from Common Corruptions [20]** shown on a sample image from the Replica[49] dataset with shift intensity 3. Each prediction is followed by its corresponding error map. Our method is resistant to distortions compared to the baselines and provides better accuracy *especially over fine-grained regions and sharpness* (see the white markers). Best seen on screen.

4.1. Evaluations on Pixel-Wise Prediction Tasks

Training dataset: We use Taskonomy [60] as our training dataset which includes 4 million real images of indoor scenes with multiple annotations for each image. We report results for *surface normals*, *depth (zbuffer)*, and *reshading* prediction, as popular target domains.

Middle Domains: From the RGB images we extract *2D edges*, *Laplace edges*, *greyscale*, *embossed*, *low-pass filtered*, *sharpened*, and *wavelet* images as the middle domains (detailed definitions can be found in the supplementary). These middle domains are commonly used for low-level image processing tasks with negligible computation cost [1, 6] and do not need any supervision. The performance was not sensitive to the choice of middle domains as the method consistently outperforms baselines and improves with more middle domains (Sec. 4.1.3, Fig. 7a).

Evaluation datasets: Our goal is to have test data that has a distribution shift from the training data to evaluate the robustness of our method. All the results are reported on the test set of the following datasets:

Taskonomy with Common Corruptions [20]: We apply the Common Corruptions on the test set of Taskonomy. They include all corruptions except outdoor corruptions (snow, frost, fog) and the ones that change the geometry of the scene (elastic transform, motion, and zoom blur). We exclude Gaussian noise and blur from evaluations as they were used for ST, to keep training and testing fully separate. Visualizations of a subset of distortions are shown in Figure 4 and for all severities in the supplementary.

Taskonomy with Adversarial corruptions [50, 32, 39]: We generate adversarial examples using Iterative-Fast Gradient Sign Method (I-FGSM) [32].

Other datasets: Replica [49] consists of 1227 images from high quality 3D reconstructions of indoor scenes. Similar to Taskonomy, we also apply common corruptions on these images. Habitat [47] consists of 1116 images from mesh renderings with a substantial shift from Taskonomy. We test on both datasets without fine-tuning (see supplementary).

Training details: All networks for our method and baselines use the same UNet backbone architecture [44] and were trained with AMSGrad [42]. We used a learning rate of 5×10^{-4}, weight decay of 2×10^{-6}, and batch size of 64. The upsampling blocks of all networks resize the activation maps using bilinear interpolation.

We also augment the network training with “cross-task consistency constraints” (X-TC) [59] for generally better results, but this is not a fundamental requirement (ablation results provided in Sec. 4.1.3). We follow [59] and apply non-probabilistic perceptual losses on the predicted mean.

Baselines: We evaluate the following baselines. They are trained with NLL loss (Eq. 1), i.e. the models output both mean and sigma.

Baseline UNet: It is a single network that maps from RGB to the target domain without going through a middle domain (i.e. direct). This is the main baseline.

Multi-domain baseline: It is a network model with *RGB* image *and* all middle domains as inputs. Since this model is not *forced* to use different middle domains as opposed to the proposed method, it reveals if learning from middle domains needs to be explicit and distributed.

Multi-task baseline: It is a single model that maps from *RGB* to *depth*, *reshading*, and *normals*. This is to reveal if learning additional tasks improved robustness.

Data augmentation baselines: We consider a baseline UNet adversarially trained to defend against I-FGSM attacks with $\epsilon = (0, 16]$. This baseline shows how well adversarial robustness translates to non-adversarial distortions. We also include style augmentation [13] as another baseline, which has been shown to reduce the texture bias that are less robust than shape cues.

Blind guess is a single prediction that captures the overall statistics of the domain, i.e. it returns the best guess of

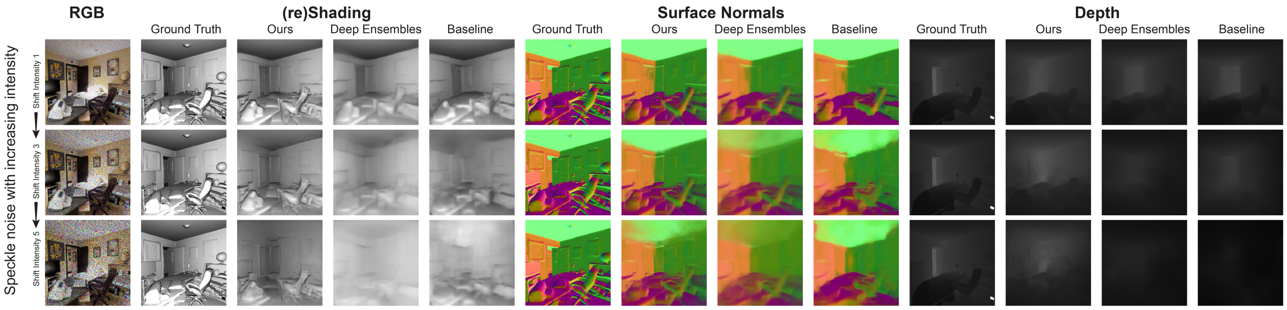


Figure 5: **Qualitative results under distribution shifts** for reshading, surface normals, and depth predictions. Each row shows the predictions from a query image from the Taskonomy test set under increasing speckle noise. Our method degrades less than the other baselines, demonstrating the effectiveness of using different cues to obtain a robust prediction. Notable improvements in the accuracy can be seen *especially in fine-grained regions*.

what the prediction should be independent of the input. Hence, it shows what can be learned from general dataset regularities (further details are in supplementary).

Deep ensembles [33] creates an ensemble by training the same exact networks with different initializations. Although there are recent papers proposing new ways to enforce diversity in ensembles, their improvement in performance against deep ensembles has not been found significant under non-adversarial shifts [51, 53]. Thus, deep ensembles remains the most relevant ensemble baseline. We use the same number of paths, i.e. ensemble components, as in our method. The predictions from each path are weighted equally to attain the final prediction. This baseline reveals if learning from different cues yields diverse predictions that results in a stronger final estimator.

Cross-domain ensemble setups evaluated: We evaluate several variants of our merging method. In all variants, different paths goes through different middle domain to produce a prediction along with one path being the *direct* prediction. They are then merged into the final prediction. We show the proposed analytical merging is superior to others.

Inverse variance merging: Each path's prediction is weighted inversely proportional to its variance, as proposed in Section 3.2.

Uniform merging: A simplified merging where each path is weighted equally, i.e. uncertainty is not used.

Network merging: A neural network is used to merge the predictions. Specifically, we consider a stacking model [54] that learns the final predictions given the outputs from each path and models the final output as a mixture of Laplacians. It has the advantage that the loss is over the entire image, thus, taking into account its spatial structure (see supplementary for details).

4.1.1 Robustness to Common Corruptions

Figures 4 and 5 show the qualitative results of our method against the baselines. Performance under various distortions is demonstrated in Figure 4 for the surface normals predictions of a sample image from Replica dataset. The proposed method consistently outperforms the baselines and provides more accurate predictions especially in fine-grained regions. This is further supported by quantitative results in Figure 6 where the ℓ_1 error over these distortions are notably lower for the proposed method compared to the baselines in all three target domains and shift intensities.

Among the evaluated baselines the data augmentation methods are the most competitive, e.g. adversarial robustness partially transferred to non-adversarial distortions, though *inverse variance merging* performs notably better.

We also observe *inverse variance merging* does much better than *uniform merging* and also better or comparable to *network merging* (Fig. 6) despite being simpler, more lightweight, and interpretable. Moreover, it does not demand fixing the number of paths beforehand (unlike *network merging*), thus the number of paths can be decided by taking computational considerations into account on the fly.

4.1.2 Robustness to Adversarial Attacks

We demonstrate the effectiveness of the proposed method under adversarial attacks. The attacks are generated by I-FGSM. Following [32], we use attack strengths $\epsilon = [2, 4, 8, 16]$, with the number of iterations given by $N = \min(4 + \epsilon, 1.25\epsilon)$. The results are shown in Table 1. Neither our method nor the baselines utilize explicit adversarial defense mechanisms – while deep ensembles perform nearly as poorly as baseline UNet, the proposed method performs significantly better. This indicates that *using middle domains* promotes ensemble diversity in a way that *makes it more challenging to create one attack that fools all paths simultaneously*, hence this approach can be a promising remedy for adversarial attacks as well. Moreover, the proposed method also outperforms *Uniform merging* (see supplementary for the results) which does not use uncertainty estimates during merging. This indicates that the additional uncertainty output did not create an additional avenue for attack that I-FGSM could exploit.

Note that we do not obfuscate gradients by e.g. intentionally making certain operations non-differentiable, or using stochastic transforms [3]. The analytical operations to

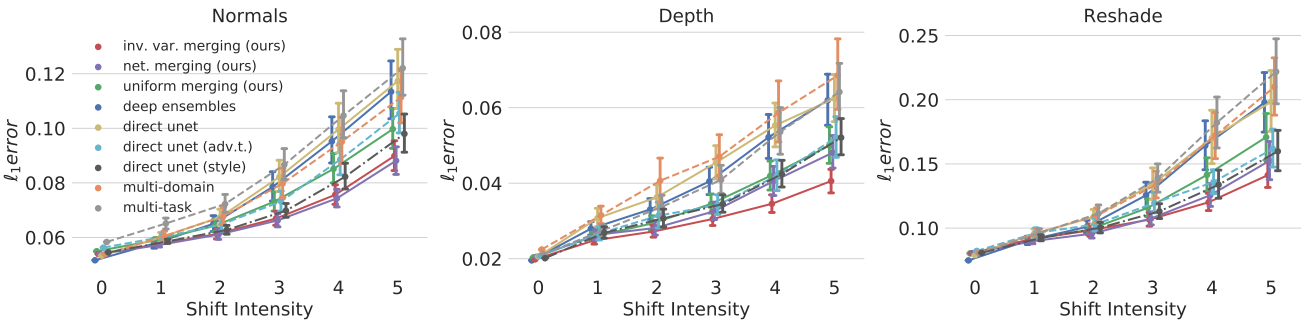


Figure 6: **Quantitative robustness evaluations using Common Corruption distortions applied on Taskonomy test set: Average ℓ_1 errors over 11 *unseen* distortions.** Our main method *inv. var. merging*, and frequently its simplified variant *uniform merging* and *network merging*, are more robust against shifts compared to the baselines. Error bars indicate one 'standard error' from the mean (via bootstrapping). Plots for additional perceptual error metrics and individual distortions are provided in supplementary material.

	Normal				Reshade				Depth			
Method \ ϵ	2	4	8	16	2	4	8	16	2	4	8	16
Baseline UNet	8.23	11.53	13.03	14.37	17.92	22.78	27.26	34.40	5.50	6.76	8.36	9.80
Deep ensembles	**7.49**	11.13	13.36	15.65	15.66	21.95	27.75	34.98	5.45	6.68	8.27	10.52
Inv. var. merging	7.60	**8.89**	**10.40**	**12.77**	**15.56**	**16.55**	**18.93**	**22.01**	**4.94**	**4.99**	**5.93**	**6.75**
Adv. T. (lower bound error)	5.78	5.74	5.45	5.53	9.39	8.98	8.07	8.20	2.23	2.27	2.39	2.74

Table 1: **Robustness against adversarial corruptions.** ℓ_1 errors for surface normals, reshade, and depth under adversarial attacks are reported. (Lower is better. Errors are multiplied by 100 for readability.) The proposed method significantly improves robustness against I-FGSM [32] based attacks *without adversarial training*, compared to the baselines. The last row shows the error for a model that has undergone adversarial training [39] with the *same attacks as those evaluated at test time*, hence it gives a lower bound on the error (see supplementary for additional details).

obtain the middle domains are deterministic and differentiable.

4.1.3 Additional Ablation Studies

Contribution of ST/X-TC: To quantify the contribution of each stage of training to the overall robustness of our setup, we study the performance of our proposed method without sigma training (ST) and/or cross-task consistency constraints (X-TC) in the first row of Figure 7a (and supplementary). Our method, with or without ST or X-TC constraints, still outperformed deep ensembles as almost all bars are below the 0 line.

In Section 2.3 of the supplementary, we compare the effect of equipping deep ensembles with ST and X-TC, and perform uniform and inverse variance merging. Thus, the only difference with our method is the use of middle domains. Our method still outperforms.

Robustness vs number of employed paths: In Figure 7a, we investigate performance as a function of number of paths. Each point shows the average ℓ_1 error of *all possible combinations* for a given number of paths. Although all methods improve as more paths are added, our proposed methods has a much steeper downward trend than deep ensembles and our uniform merging variant, indicating that performance gap increases with more paths.

Sensitivity to choice of middle domains: Figure 7a also shows that the performance of our method is not sensitive to a particular set of middle domains. For a fixed number of paths n, our method outperforms deep ensembles for all possible combinations of n paths on average.

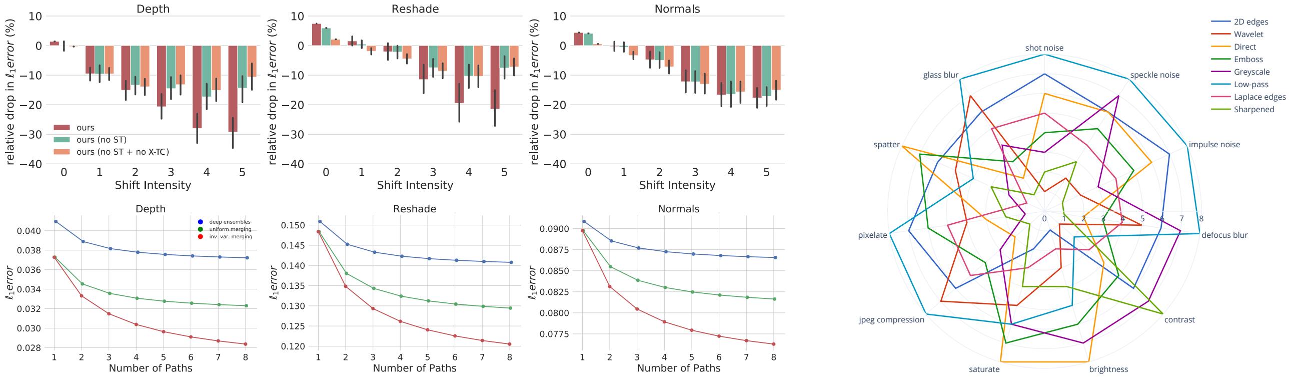
Path importance: We show the importance of each middle domain for the final prediction under each distortion in Fig. 7b. For number of paths $n = 1, \ldots, 8$, we compute the set of best performing paths, i.e. the set of n paths with the lowest ℓ_1 error, denoted by $P_n = \{p_{(i)}\}_{i=1}^{n}$. The n^{th} best performing path is given by $P_n \backslash P_{n-1}$. The plot shows different paths indeed react differently to a given corruption, e.g. noise distortions substantially benefited from *low-pass*, while contrast distortion did not – thus the benefit is not attributed to one or few middle domains under all distortions.

4.1.4 Performance on undistorted data

In order to demonstrate that the robustness of our method on out-of-distribution data did not come at the cost of degraded performance on in-distribution data, we provide quantitative evaluations on the *undistorted* Taskonomy and Replica datasets in supplementary. The results show the performance of our method, *when tested on undistorted data*, is indeed comparable to or better than the methods that are trained to perform well only on undistorted data.

4.2. Evaluation on Classification Tasks

The benefits of the proposed method is not limited to regression or dense pixel-wise tasks. We performed an experiment on ImageNet-C to evaluate the robustness against Common Corruptions (Table 2). Our method and deep ensembles both use 8 paths with identical ResNet-50 [19] network architecture. In addition, in this experiment our method does a simple averaging of the output probabilities from each path, similar to deep ensembles, and no ST or X-TC training was involved. The superior results show the basic value in using a diverse set of middle domains. Similar conclusions were obtained for CIFAR-10-C and CIFAR-



(a) **Effect of sigma and/or consistency training. (Top row)** The plots show the relative change in ℓ_1 error compared to deep ensembles (i.e. negative means outperforming deep ensembles). The proposed method outperforms deep ensembles under distribution shifts even without ST and X-TC (consistency).
Robustness as a function of number of paths. (Bottom row) The plots show the average ℓ_1 error as we increase the number of paths (or ensemble components in the case of deep ensembles). The proposed method (*inv. var. merging*) and its simplified variant (*uniform merging*) consistently outperforms deep ensembles which plateaus much faster.

(b) **Importance of each middle domain for different distortions.** The chart shows the order of the best performing paths for surface normal perdiction for different distortions, with 8 denoting the most important and 1 the least important path. The plot shows, for instance, "noise" distortions benefited most from the *low-pass* middle domain while "contrast" distortion benefited most from the *sharpened* middle domain.

Figure 7: **Ablation studies:** We performed additional studies to gain insights on the effect of sigma and consistency training, increasing the number of paths, and the ordinal effect of each distortion on the middle domains. Similar to Figure 6, in (a) and (b), we apply Common Corruption distortions on Taskonomy data and average the ℓ_1 errors over 11 *unseen* distortions. Error bars indicate one 'standard error' from the mean (via bootstrapping). See supplementary for a more detailed breakdown of each of these studies.

100-C datasets (full results in supplementary).

Method	Clean error	mCE
Baseline ResNet-50	24.37	76.21
Deep ensembles	21.50	70.43
Ours	21.61	67.85

Table 2: **Robustness on ImageNet-C.** Error on clean and distorted data (mean Corruption Error – mCE). Following [20], the mCE is relative to AlexNet [30]. All methods are trained only on clean ImageNet training data. Our method performs noticeably better under distortions compared to deep ensembles and a single model baseline ResNet. See supplementary for a detailed breakdown and additional results on CIFAR.

5. Conclusion and Discussion

We presented a general framework for making robust predictions based on creating a diverse ensemble of various middle domains. Experiments demonstrated that this approach indeed leads to more robust predictions compared to several baselines.

We also showed that our method is not sensitive to the choice of middle domains (Sec. 4.1.3) or the corruptions used for ST (supplementary). Furthermore, even after equipping deep ensembles with ST and consistency training (Sec. 4.1.3, supplementary), our method still outperforms, confirming the effectiveness of using middle domains.

Below we briefly discuss some of the limitations:

Uncertainty under distribution shift: Our method relies on having reasonable uncertainty estimates (i.e. sigma) in presence of distribution shifts. While we observed sigma training to be helpful for this purpose, and also, uniform merging which does not rely on uncertainty estimates to still outperform the baselines, our method will benefit from better uncertainty estimation techniques.

Choice of middle domains: We adopted a fixed set of middle domains, and, as discussed in Sec. 4.1.3, the final performance was not sensitive to the adopted dictionary. However, *learning* or computationally *selecting* such middle domains with the objective of downstream robustness could be a worthwhile future direction.

Multi-modal distributions: We modeled our individual path outputs with single-modal distributions for convenience and considered multi-modal distributions only at merging step. Allowing for multi-modality in each path's output may further help with ambiguous data points.

Computational cost: While the computational complexity of our method and deep ensembles [33] are virtually the same, the methods based on ensembling generally increase the computational complexity as they involve turning one estimator into multiple. Investigating if the models in the ensemble can be compressed would be worthwhile – especially for our method since the diversity in the ensemble is by structure and owed to adopting different middle-domains, rather than stochasticities that often assume independence among models.

Acknowledgement: We thank Alexander Sax for the insightful early experiments and useful discussions.

References

[1] Tinku Acharya and Ajoy K Ray. *Image processing: principles and applications*. John Wiley & Sons, 2005. 5

[2] Arsenii Ashukha, Alexander Lyzhov, Dmitry Molchanov, and Dmitry Vetrov. Pitfalls of in-domain uncertainty estimation and ensembling in deep learning. *arXiv preprint arXiv:2002.06470*, 2020. 2

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018. 6

[4] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000. 2

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural networks. *arXiv preprint arXiv:1505.05424*, 2015. 2

[6] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 5

[7] Armen Der Kiureghian and Ove Ditlevsen. Aleatory or epistemic? does it matter? *Structural Safety*, 31(2):105–112, 2009. 2

[8] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer, 2000. 2

[9] Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2017. 1

[10] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019. 2

[11] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001. 2

[12] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059, 2016. 2

[13] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. 5

[14] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992. 2

[15] Alex Graves. Practical variational inference for neural networks. In *Advances in neural information processing systems*, pages 2348–2356, 2011. 2

[16] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330, 2017. 2

[17] Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, pages 905–914. PMLR, 2020. 2

[18] Joachim Hartung, Guido Knapp, and Bimal K Sinha. *Statistical meta-analysis with applications*, volume 738. John Wiley & Sons, 2011. 4

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7

[20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019. 1, 4, 5, 8

[21] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018. 2

[22] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019. 2

[23] Ian P Howard and Brian J Rogers. *Seeing in depth, Vol. 2: Depth perception.* University of Toronto Press, 2002. 2

[24] Ian P Howard, Brian J Rogers, et al. *Binocular vision and stereopsis*. Oxford University Press, USA, 1995. 2

[25] Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017. 1

[26] Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019. 2

[27] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, pages 5574–5584, 2017. 2, 3

[28] Ildoo Kim, Younghoon Kim, and Sungwoong Kim. Learning loss for test-time augmentation. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 8

[31] Volodymyr Kuleshov, Nathan Fenner, and Stefano Ermon. Accurate uncertainties for deep learning using calibrated regression. In *International Conference on Machine Learning*, pages 2796–2804, 2018. 2

[32] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016. 2, 4, 5, 6, 7

[33] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6402–6413, 2017. 2, 6, 8

[34] Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017. 2

[35] Marius Leordeanu, Mihai Pirvu, Dragos Costea, Alina Marcu, Emil Slusanschi, and Rahul Sukthankar. Semi-supervised learning for multi-task scene understanding by

neural graph consensus. *arXiv preprint arXiv:2010.01086*, 2020. 2

[36] Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017. 2

[37] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716, 2016. 2

[38] Christos Louizos and Max Welling. Multiplicative normalizing flows for variational bayesian neural networks. *arXiv preprint arXiv:1703.01961*, 2017. 2

[39] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2, 4, 5, 7

[40] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, pages 13991–14002, 2019. 3

[41] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. *arXiv preprint arXiv:1901.08846*, 2019. 2

[42] Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019. 5

[43] Carlos Riquelme, George Tucker, and Jasper Snoek. Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*, 2018. 2

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015. 5

[45] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pages 53–69. Springer, 2020. 2

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4

[47] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 4, 5

[48] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 2

[49] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The Replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 4, 5

[50] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1, 2, 4, 5

[51] Yeming Wen, Dustin Tran, and Jimmy Ba. Batchensemble: an alternative approach to efficient ensemble and lifelong learning. In *International Conference on Learning Representations*, 2020. 6

[52] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018. 2

[53] Florian Wenzel, Jasper Snoek, Dustin Tran, and Rodolphe Jenatton. Hyperparameter ensembles for robustness and uncertainty quantification. *arXiv preprint arXiv:2006.13570*, 2020. 2, 6

[54] David H Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. 6

[55] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018. 2

[56] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: Diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[57] Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. *arXiv preprint arXiv:2002.11328*, 2020. 2

[58] Sheheryar Zaidi, Arber Zela, Thomas Elsken, Chris Holmes, Frank Hutter, and Yee Whye Teh. Neural ensemble search for performant and calibrated predictions. *arXiv preprint arXiv:2006.08573*, 2020. 2

[59] Amir Zamir, Alexander Sax, Teresa Yeo, Oğuzhan Kar, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas Guibas. Robust learning through cross-task consistency. *arXiv preprint arXiv:2006.04096*, 2020. 2, 3, 4, 5

[60] Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018. 4, 5

[61] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

[62] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2223–2232, 2017. 2