ROBUSTNESS VIA PROBABILISTIC CROSS-TASK ENSEMBLES

Anonymous authors
Paper under double-blind review

ABSTRACT

We present a method for making predictions using neural networks that, at the test time, is robust against shifts from the training data distribution. The proposed method is based on making *one prediction via different cues* (called middle domains) and ensembling their outputs into one strong prediction. The premise of the idea is that predictions via different cues respond differently to distribution shifts, hence one can merge them into one robust final prediction, if ensembling can be done successfully. We perform the ensembling in a straightforward but principled probabilistic manner. The evaluations are performed using multiple vision dataset under a range of natural and synthetic distribution shifts which demonstrate the proposed method is considerably more robust compared to its standard learning counterpart, conventional ensembles, and several other baselines.

1 INTRODUCTION

Neural networks deployed in the real-world will encounter data with naturally occurring distortions, e.g. blur, brightness changes, etc. Such changes make up shifts from the training data distribution. While neural networks are able to learn complex functions in-distribution, their predictions are deemed unreliable under such shifts (Dodge & Karam (2017); Hendrycks & Dietterich (2019)). This presents a core challenge that needs to be solved for these models to be reliable 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 Figure 1). A common approach is to learn this mapping with a *direct* path, i.e. *RGB* $\rightarrow$ *surface normals*. Since this path directly operates on the input domain, it is prone to being affected by any modest alterations in the RGB image, e.g. brightness changes. An alternative can be to go through a *middle domain* (or middle "task") that is invariant to that change. For example, an *RGB* $\rightarrow$ *2D edges* $\rightarrow$ *surface normals* path will be resilient to the brightness and color changes in the input. By creating an ensemble of prediction made via a diverse set of such middle domains, we can be robust against a wide range of distribution shifts.

This paper presents a general approach for obtaining a single robust prediction from multiple paths. We first use a set of middle domains from which we learn to predict the final domain. Each path reacts differently to a particular distribution shift, thus its prediction may or may not degrade severely. We further estimate the uncertainty of each path's prediction which allows us to adopt 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 are guided by their predicted uncertainties. Moreover, the middle domains we adopt are all self-supervised (as they can be programmatically extracted), thus this framework does not require any additional supervision/labeling than what a dataset already comes with. We demonstrate improved robustness to both natural and synthetic shifts on standard benchmark datasets compared to several baselines.

2 RELATED WORK

This work has connections to a number of topics, including ensembling, uncertainty estimation and calibration, enforcing consistency constraints, and adversarial attacks. We overview some of them within the constraints of space.

Ensembling combines multiple weak learners into a single strong learner e.g. boosting and bagging (Dietterich (2000)). A deep ensemble (Lakshminarayanan et al. (2017)) produces multiple hypotheses by *training the same network with different initializations*. This reliance on the network

initialization or such stochasticities does not necessarily result in effectively *diverse* predictions to lead to a robust final prediction. In contrast, our method *forces* the predictions to use a diverse set of cues. This provides a more diverse ensemble by design, providing a better opportunity for assmebling a final robust prediction in presense of a distribution shift (see Figure 1).

Estimating uncertainty: Uncertainty in a model's prediction can be decomposed into two sources (Der Kiureghian & Ditlevsen (2009); Kendall & Gal (2017)). *Epistemic* uncertainty accounts for uncertainty in the model's parameters, while *aleatoric* uncertainty models the noise inherent in the data. While there are many proposed methods to estimate the former, such as using dropout as approximate Bayesian inference (Gal & Ghahramani (2016)) and ensembling (Lakshminarayanan et al. (2017)), consistency energy (Zamir et al. (2020)) is the most relevant one, where a *single* uncertainty estimate is predicted from different paths based on their cross-task consistency. Here, we estimate the uncertainty for *each* path and use it to produce a single strong estimator.

Calibration: Neural networks tend to produce outputs that are miscalibrated i.e. their predictions do not reflect the true likelihood of being correct (Guo et al. (2017); Kuleshov et al. (2018)). In particular, their predictions tend to be *overconfident* for unfamiliar examples. Similar to (Hafner et al. (2020)), we perform data augmentation to train the model (in two stages) to output high uncertainty outside of the training distribution, with the difference that we do not need to assume a prior distribution for the out of distribution data, but train sigmas (uncertainty estimates) in a supervised way (Sec. 3.1).

Enforcing consistency constraints in the context of multiple paths predictions involves ensuring that the output predictions remains the same regardless of the intermediate domain. It has been shown to lead to a better generalization, transfer performance, and may prevent overfitting to superficial cues (Szegedy et al. (2013); Jo & Bengio (2017)) in the training data (Zamir et al. (2020); Zhu et al. (2017)). In comparison to (Zamir et al. (2020)), we cast the consistency framework into a probabilistic one by modelling input and path dependent uncertainty which allows for a more principled way of composing individual path estimates. Also, unlike (Zamir et al. (2020)), our goal is to robustify the final prediction by merging the output of multiple prediction paths at the test time.

Adversarial attacks adds imperceptible worst case shifts to the input to fool a model (Szegedy et al. (2013); Madry et al. (2017)). While our framework is applicable to any type of distribution shifts, we focus on natural shifts and commonly occurring distortions within the scope of this work.

Robustness via data augmentation: One approach to address robustness involves the use of data augmentation during training. However, performance gains are non-uniform across corruptions (Ford et al. (2019)). While these methods usually involve training with a set of corruptions to generalize to the unseen ones, here we use middle domains to be resistant to different corruptions.

3 METHOD

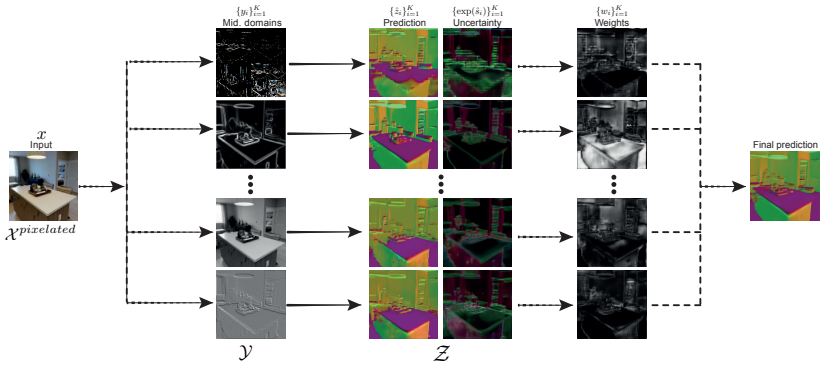


Figure 1: **An overview of our method for creating an ensemble of diverse prediction paths.** A network is trained to go from a *pixelated* (low resolution) RGB image to a target domain, e.g. surface normals, via several middle domains, e.g. wavelet, 2D edges, greyscale, and emboss. We then compute the corresponding weights of these predictions based on their uncertainties. The final prediction is obtained by a weighted average. Solid arrows represent learned mappings and dashed ones represent analytical mappings.

Figure 1 shows an overview of our method with an example on learning a mapping from RGB to surface normals. We obtain surface normals predictions and their corresponding uncertainty estimates for a given input image via several middle domains. These predictions are then combined to obtain the final prediction. Each middle domain reacts differently to distribution shifts, thus the performance of the method can be expected to increase by combining predictions from a diverse set of these domains. For example, greyscale of the input can be used to bring robustness against color changes in the input. While these domains can also be obtained using a learning based approach, e.g. predicting surface normals from the output of another network such as reshading estimator, here we choose to use analytical (non-learning based) ones for simplicity.

In the rest of this section, we elaborate on the technical details of our method.

Notations: Define $\mathcal{X}$ as the RGB domain, $\mathcal{Y} = \{\mathcal{Y}_i\}_{i=1}^{K}$ as the intermediate domains, $\mathcal{Z}$ as the desired prediction domain, where $\mathcal{Z} \cap \mathcal{Y} \neq \emptyset$. A single datapoint from these domains is denoted as $(x, y_1, \ldots, y_K, z)$. $\mathcal{F}_{\mathcal{XY}}$ is the set of functions that maps the RGB images to their intermediate domains, $\mathcal{F}_{\mathcal{XY}} = \{f_i : \mathcal{X} \to \mathcal{Y}_i\}_{i=1}^{K}$, and $\mathcal{F}_{\mathcal{YZ}}$ from the intermediate to the prediction domain, $\mathcal{F}_{\mathcal{YZ}} = \{g_i : \mathcal{Y}_i \to \mathcal{Z}\}_{i=1}^{K}$. Given K predictions of domain $\mathcal{Z}$, they are merged using the function m to get a final single prediction, $m : \mathcal{Z}^K \to \mathcal{Z}$.

3.1 Modelling prediction noise

We model the noise in the predictions with a Laplace distribution. This results in an ℓ_1-norm loss on the errors as opposed to an ℓ_2-norm loss with a Gaussian distribution, as it has been shown to improve prediction quality (Kendall & Gal (2017)). For a mapping function g_j, this leads to the following negative log-likelihood (NLL) formulation:

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

where N is the number of samples, z_n is the label for the nth sample, and $[\hat{z}_n, \hat{s}_n] = g_j(y_n)$ are the model outputs given the input y_n. We predict $\hat{s}_n = \log \hat{b}_n$ for numerical stability where $\hat{b}_n$ is the scale parameter of the Laplace distribution. The *sigma* is obtained by scaling this parameter with $\sqrt{2}$ and it captures the per-pixel uncertainty in predictions.

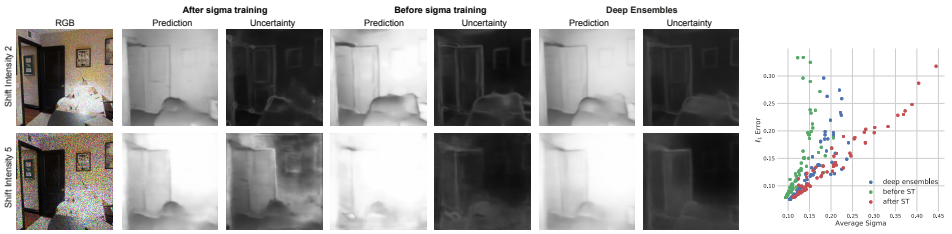


Figure 2: **Overconfident predictions under high distortions.** (Left) Qualitative prediction results of image reshading and uncertainty estimates under speckle noise distortion, for deep ensembles and a single UNet model before and after sigma training. Darker sigma denotes lower uncertainty. (Right) Scatter plot of ℓ_1 error versus average sigma. Each point represents one of the unseen distortions and one of 5 levels of shift intensity. Notice (qualitatively and quantitatively) that the models without sigma training produce poorer results under the larger shift, while their uncertainty does not correspondingly increase. Our proposed method of training sigmas generalize the model to have a stronger correlation with the error compared to the baselines. This indicates that sigma (*after sigma training*) can be an effective signal for merging multiple predictions.

Sigma training: Uncertainty estimates under distribution shifts are poorly calibrated (Ovadia et al. (2019)), i.e. there is a tendency to output a poor prediction with high confidence. This can be seen in Figure 2, in the columns corresponding to "Before sigma training". With a higher noise distortion, the prediction clearly degraded, however, the uncertainty estimate did not increase correspondingly. This issue persists even with methods that estimate epistemic uncertainty, which are meant to detect these shifts (see Figure 2, "Deep Ensembles" columns).

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 caused by a few

distortions from the set of common corruptions (Hendrycks & Dietterich (2019)). We denote this step as *sigma training* (ST). As the goal of this step is to train $\hat{s}_i$, as opposed to $\hat{z}_i$, to generalize under distortions (dist), we have a loss term to ensure that $\hat{z}_n$ does not deviate from its predictions at the start of training, $\hat{z}_{n,0}$, which we denote as *mean grounding* (MG). In addition, $\hat{s}_n$ is trained in a supervised way to learn its maximum likelihood estimator, with the loss denoted as *sigma calibration* (SC). Finally, we include the original NLL term from Equation 1 on undistorted data (undist) to prevent forgetting. This results in the following loss formulation:

$$\mathcal{L}_{f_j,ST} \;=\; \mathcal{L}^{undist}_{f_j,NLL} + \alpha_1 \mathcal{L}^{dist}_{f_j,MG} + \alpha_2 \mathcal{L}^{dist}_{f_j,SC}, \tag{2}$$

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 training, i.e. $\mathcal{L}^{dist}_{g_j,MG} = \|\hat{z}_{n,0} - \hat{z}_n\|_1$. The SC loss guides the scale parameter towards its maximum likelihood estimate, i.e. $\mathcal{L}^{dist}_{g_j,SC} = \|\exp(\hat{s}_n) - \arg\min_{\hat{s}_n} \mathcal{L}^{dist}_{g_j,NLL}\|_1 = \|\exp(\hat{s}_n) - |z_n - \hat{z}_{n,0}|\|_1$.

Following sigma training, the network outputs sigmas that are highly correlated with error (Figure 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.

3.2 Merging predictions of multiple paths

After training the set of mappings $\mathcal{F}_{\mathcal{XY}}$ and $\mathcal{F}_{\mathcal{YZ}}$ with the proposed methods described above, it remains to combine these predictions $\mathcal{Z}^K$ obtained via multiple paths using a merging function m. We consider both analytical and learned models for m. The former can be a straightforward weighting of each pixel in each path by the inverse of its variance (Hartung et al. (2011)). This is denoted as *Inverse variance merging*, and it can be done with negligible computational cost. The latter can be a stacking model (Wolpert (1992)) that *learns* the final predictions given the outputs of each path. It has the advantage that the loss is over the entire image, thus, taking into account its spatial structure. In what follows, we further discuss our approach for this case, denoted as *Network merging*. We observe that both methods perform comparably well, thus we primarily suggest the analytical method due to being simpler, more lightweight, and interpretable.

Multi-modal predictions: Mixture models can capture inherent ambiguity in the data by assuming that there are several possible distributions that could have generated the observed data. Thus, they are essential for ill-posed problems such as ours, e.g. there can be several depth estimates that corresponds to a given RGB image. Given K random variables, their mixing weights $\{\hat{w}_i\}_{i=1}^K$ reflect the uncertainty over which of the K variables generated the observed data. From Sec. 3.1 we estimate the parameters of these K distributions, $\{\hat{z}_i, \hat{s}_i\}_{i=1}^K$, then function m learns to outputs the mixing weights given these set of parameters. Our final distribution is a mixture of Laplacians,

$$h_{Mix}(z) \;=\; \sum_i \hat{w}_i h(z|\hat{z}_i, \hat{s}_i) \tag{3}$$

where $h(z|\hat{z}_i, \hat{s}_i)$ is the probability density function of a Laplace distribution with mean $\hat{z}_i$ and scale $\exp(\hat{s}_i)$. The final loss is the NLL of the multi-modal prediction $\mathcal{L}_m \;=\; \frac{1}{N}\sum_{n=1}^{N} -\log h_{Mix}(z_n)$.

Mean approximation: The mean of a multi-modal distribution is in general not representative of the overall distribution. Instead of directly using the weights predicted by the network to compute the mean, we approximate the weights for a path to be proportional to its mixture probability distribution function evaluated at itself, i.e. $w_i \propto h_{Mix}(\hat{z}_i)$.

3.3 Training with Cross-Task Consistency Loss

The above framework can be further augmented with "cross-task consistency constraints"Zamir et al. (2020), to ensure that predictions from the different paths are in cross-task agreement. While this is not a fundamentally required step for the proposed method, it yields better accuracy especially in fine-grained regions as demonstrated in the Experiments section. Following (Zamir et al. (2020)), we consider a set of *perceptual* loss networks on the outputs of g_j. This corresponds to minimizing an ℓ_1 error between the predictions obtained by the model and those from the ground truth in the perceptual domain. In a probabilistic setting, such as ours, consistency training can be done by minimizing the symmetric KL divergence between the predicted distribution and the distribution

obtained from the ground truth in the perceptual domain:

$$\begin{aligned}\mathcal{L}_{g_j,consistency} \;=\; & KL(h(z|g_{jk}(\hat{z}_j,\hat{s}_j)||h(z|g_{jk}(z_j,-\infty))) \\ & + KL(h(z|g_{jk}(z_j,-\infty))||h(z|g_{jk}(\hat{z}_j,\hat{s}_j)),\end{aligned} \tag{4}$$

where $[\hat{z}_{jk},\hat{s}_{jk}] = g_{jk}(\hat{z}_j,\hat{s}_j)$ is the mapped output of g_j, i.e. $[\hat{z}_j,\hat{s}_j]$, to a perceptual domain $\mathcal{P}_k$. This can be performed for several perceptual domains for more effective cross-task consistency.

As consistency training in our setting requires a mapping of probabilistic input onto perceptual domains, this may limit the usage of existing pre-trained networks with point estimates trained on large datasets, e.g. ResNet (He et al. (2016)) model trained on ImageNet (Deng et al. (2009)). Hence an alternative approach would be to pass only the predicted mean, i.e. $\hat{z}_j$, to the perceptual domain with a point estimate network penalizing ℓ_1 error, while keeping the NLL loss for the direct path g_j to supervise predicted uncertainty.

4 EXPERIMENTS

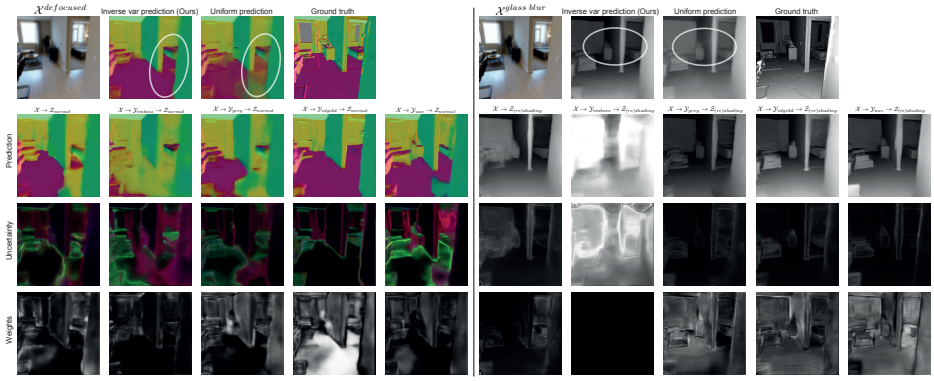


Figure 3: **How does our method work?** Each network receives different information for making a prediction, due to going through different middle domains. Given a defocused image (left), the 2D edges extracted from it was the least affected by the distortion, and returned a prediction with the lowest overall uncertainty, which is reflected in the weights. Similarly, for an image with glass blur (right), the method successfully disregards the degraded output from the emboss path for the final prediction. The quality of the final prediction depends on the following elements: **1.** At least one middle domain is robust against the encountered distortion - the proposed inverse variance merging obtains significantly better results than learning from the RGB directly (leftmost column of each example). **2.** The uncertainty estimates are well correlated with error, allowing us to select regions from the best performing path - a uniform average (first row) of path predictions does not take into account the uncertainties and results in worse predictions.

We demonstrate here that our method is robust to natural and synthetic distribution shifts compared to several baselines over different datasets and target domains.

Training dataset: We use Taskonomy (Zamir et al. (2018)) as our training dataset which includes 4 million real images of indoor scenes with multiple annotations for each image. For the experiments, we employed the *RGB* and the following target domains from the dataset: *surface normals*, *depth (zbuffer)*, and *reshading*. From the RGB images we extracted *2D edges*, *greyscale*, *embossed*, and *wavelet* images as middle domains. All these middle domains are self-supervised as they need no supervision and can be computed programmatically extracted. All the results are reported on the test set.

Evaluation datasets: Our goal is to have test data that has a distribution shift from the training data to evaluate the robustness of our setup. Thus, the following datasets are used:

Common corruptions (Hendrycks & Dietterich (2019)): We apply the most relevant set of corruptions on the test set of Taskonomy. They include all corruptions with the exception of weather, and elastic transform, motion and zoom blur, as they change the geometry of the scene, and the two corruptions that were used for sigma training (Gaussian noise and blur). Visualizations of the distortions used, for all levels of severity are shown in Fig. 7 in the Appendix for a sample test image.

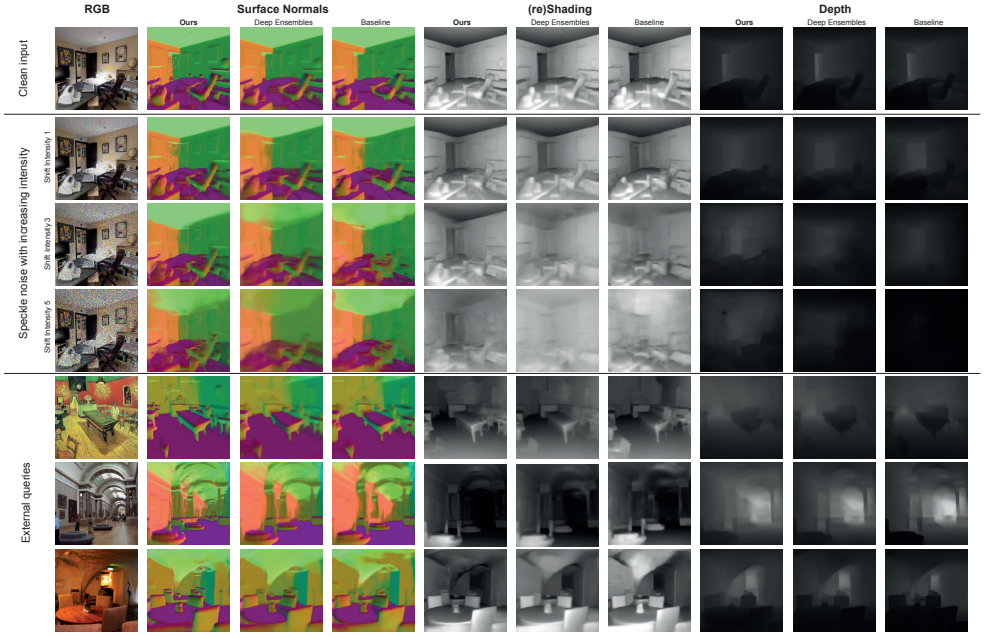


Figure 4: **Qualitative results under synthetic and natural distribution shifts** for normal, reshading, and depth. The first four rows show the predictions from a query image from the Taskonomy test set under no distortion and increasing speckle noise. Our method degrades less than the other baselines, demonstrating the effectiveness of using different cues to obtain a robust prediction. The last three rows shows the results from external queries (Zamir et al. (2020)). Again, our method demonstrates better generalization to images considerably different from the training dataset. Notable improvements in the accuracy can be seen *especially in fine-grained regions*.

Replica (Straub et al. (2019) consists of images from high quality 3D reconstructions of indoor scenes. We test on 1227 images (no training).

Training details All networks use a UNet architecture (Ronneberger et al. (2015)) and were trained with AMSGrad (Reddi et al. (2019)). Networks in $\mathcal{F}_{\mathcal{YZ}}$ used a learning rate of 5×10^{-4} and batch size of 64. For the merging network, a learning rate of 3×10^{-5} and batch size of 32 are used. The upsampling blocks of all networks resizes the activation maps using bilinear interpolation.

For each target domain, we have five paths: one that learns the target domain without going through an intermediate domain (i.e. direct) and the other four with intermediate domains: *greyscale*, *embossed*, *wavelet*, and *2D edges*. Each path has either no or one intermediate domain. Gaussian noise and Gaussian blur distortions were used for the sigma and merging training.

Baselines We evaluate the following baselines.

Baseline UNet: It is a single model that maps from RGB to the target domain, trained with NLL loss (see equation 1), i.e. it outputs both mean and standard deviation, without going through an intermediate domain. This is the main baseline.

Multi-domain baseline UNet: It is a single model with *RGB* image and its middle domains as inputs, trained with an NLL loss. Since this model is not *forced* to use different middle domains as opposed to the proposed method, it would reveal if improvements were be attributed to learning from multiple middle domains.

Blind guess is a single prediction that captures the overall statistics of the domain, i.e. it returns the best guess of what the prediction should be independent of the input. Hence it shows what can be learned from general dataset regularities (Further details are shown in the Appendix A.2).

Deep ensembles (Lakshminarayanan et al. (2017)) trains the same exact networks just with different initializations. We use the same number of paths, i.e. ensemble components, here as in our setup. Each path is weighted equally. This baseline would show if learning from different cues yields diverse predictions that result in a stronger final estimator.

Cross-task ensemble setups evaluated: We evaluate several variants of our method. In all the variants, each path goes through a different middle domain to get the final prediction, with one path being a direct prediction:

Uniform merging: Each path is weighted equally.

Inverse variance merging: Each path's prediction is weighted inversely proportional to its variance.

Network merging: Each path's prediction is weighted inversely proportional to its mixture probability density function as described in Sec 3.2.

We refer to Figure 3 to demonstrate how our method works. For a single distortion, we show each path's prediction, uncertainty, and the corresponding weights. For the defocused image in the left, the $\mathcal{X} \rightarrow \mathcal{Y}_{emboss} \rightarrow \mathcal{Z}_{normal}$ path returned a degraded prediction, which is reflected in its high uncertainty estimates. Similarly, $\mathcal{X} \rightarrow \mathcal{Y}_{edge2d} \rightarrow \mathcal{Z}_{normal}$ is less affected by the distortion and have a lower uncertainty. As a result, the final prediction is weighted more towards it. We also show the final prediction from a uniform average of each path. While it is better than learning from the input image directly, i.e. $\mathcal{X} \rightarrow \mathcal{Z}_{normal}$, using the uncertainty estimates as weights result in a more accurate prediction. Similar observations can be made for the glass blurred image in the right, where the method learned weights in a way that the degraded path is not used in the final prediction.

There are two key elements to the effectiveness of our method, the first being that with a diverse set of middle domains, it is more likely that one of them is less affected by distortions, and returns an accurate prediction. The second is that the error of the prediction well correlates 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.

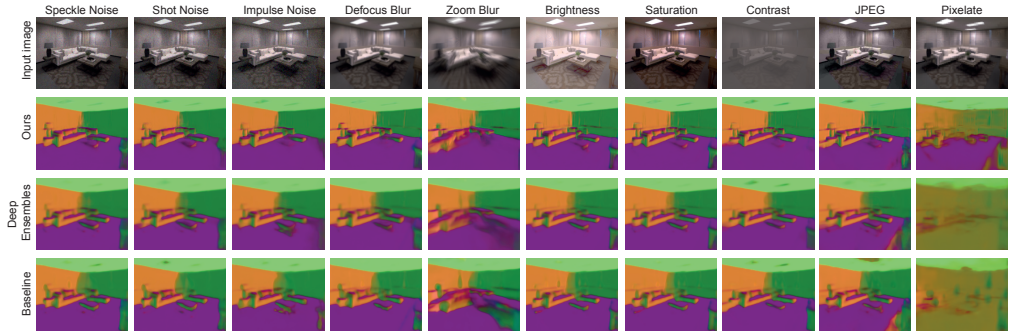


Figure 5: **Benefits of ensembling with multiple middle domains** shown on a sample image from the Replica dataset for 10 distortions, shift intensity 3. Our method is resistant to distortions compared to the baselines and provides better accuracy especially in the fine-grained regions. Best seen on screen.

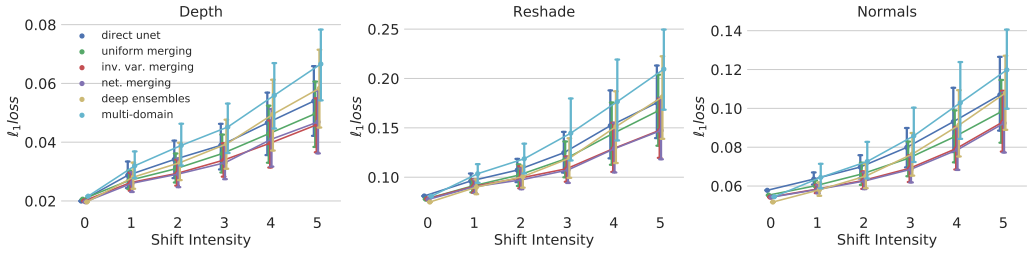


Figure 6: **Quantitative results: Average ℓ_1 loss over 11 unseen distortions.** This *does not* include the 2 distortions used during training and shows the performance for *unseen* distortions. Error bars correspond to the bootstrapped standard error. The losses are computed over two buildings of the Taskonomy test set. The proposed ensembling approaches are more robust against shifts with increasing intensities compared to the baselines.

Figures 4 and 5 show the qualitative results of our method against the baselines. Our method *generalizes better* to *unseen* distortion and query images markedly different from those seen during

training. Performance in various distortions is demonstrated further in Figure 5 for the surface normals predictions of a sample image from Replica dataset. It can be seen that 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 lower for the proposed ensembles compared to the baselines in all three target domains and shift intensities. Losses for individual distortions are shown in Appendix A.1.

Method	Taskonomy									Replica								
	Depth			Reshade			Normal			Depth			Reshade			Normal		
	Perceptual err.			Perceptual err.			Perceptual err.			Perceptual err.			Perceptual err.			Perceptual err.		
	Reshade	Normal	Direct	Depth	Normal	Direct	Reshade	Depth	Direct	Reshade	Normal	Direct	Depth	Normal	Direct	Reshade	Depth	Direct
Blind guess	25.517	21.232	7.951	9.068	19.939	21.639	31.529	5.123	16.169	21.001	22.397	5.307	4.778	16.460	16.986	28.424	3.390	15.012
Baseline UNet	9.227	8.933	2.187	2.829	6.379	6.807	19.376	4.587	4.590	11.133	7.608	2.517	2.252	5.731	8.867	18.211	3.298	4.413
Multi-domain	10.114	9.708	2.390	2.938	6.373	7.067	18.915	4.498	4.685	11.174	8.798	2.089	2.351	5.792	9.201	17.316	3.023	4.727
Deep ensembles	8.941	8.267	2.209	2.773	6.009	6.459	20.141	4.807	4.463	9.690	7.080	1.921	2.041	5.557	8.558	20.015	3.765	4.353
Uniform merging	8.342	7.294	2.398	2.583	5.658	6.776	8.535	2.334	4.799	8.718	5.796	2.034	2.092	5.224	8.825	9.383	2.183	4.436
Ours (Inv. var. merging)	8.453	7.775	2.321	2.487	5.649	6.650	8.370	1.982	4.711	9.494	6.710	2.054	2.209	5.225	8.718	8.768	1.597	4.227
Ours (Network merging)	8.247	7.451	2.222	2.535	5.660	6.716	8.233	1.860	4.699	9.078	6.286	2.015	2.150	5.221	8.782	8.566	1.369	4.199

Table 1: **Quantitative evaluation on *undistorted* Taskonomy and Replica datasets**. Results are reported for depth, reshading, and normal using direct and perceptual error metrics. The perceptual metrics evaluate the target prediction in another domain. ℓ_1 losses are reported, multiplied by 100 for readability. Our proposed method outperforms on the perceptual metrics while being comparable in the direct metrics, showing that *the performance does not decrease on undistorted data while being robust against distribution shifts*.

In Table 1, we report quantitative evaluations on the *undistorted* Taskonomy and Replica datasets for the target domains depth, reshading, and surface normals. The proposed inverse variance and network merging methods yield similar performance in the direct ℓ_1 error with other baselines, while also being more robust against distribution shifts as shown in Fig. 6. This demonstrates *the robustness on out-of-distribution data did not come at the cost of degraded performance for in-distribution data*. Besides the direct metric, we also report perceptual errors to evaluate the same prediction in a different representation space (e.g. depth $\rightarrow$ normal) to give a non-uniform attention to pixel properties (Zamir et al. (2020)). Our cross-task consistent ensembles yield significantly lower perceptual errors compared to the baselines. Note that the network merging results is slightly better than that of inverse variance merging (see Table 1 and Figure 6). As the latter does not require a network to attain the weights, thus, it can be used if there are computational constraints.

5 Conclusion

We presented a general framework for making predictions robust against distribution shifts, based on creating a diverse ensemble of predictions via various (self-supervised) middle domains. Experiments demonstrated that this approach indeed leads to more robust predictions compared to standard learning as well as conventional ensembles. We briefly discuss some of the limitations:

Uncertainty under distribution shift: Our method relies on having reasonable uncertainty estimates in presense of distribution shifts. While we observe sigma training to be helpful for this purpose, the effectiveness of our method expands with better uncertainty estimation techniques.

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

Independent channels: Another assumption made for convenience was that the channels of the multi-channel outputs, e.g. surface normals, are independent. Modelling such covariances may return better uncertainty estimates.

Fixed set of middle domains and sigma training corruptions: In our experiments, we adopted a fixed set of (self-supervised) middle domains. The final performance is expected to improve with including more middle domains, as that makes it more likely to have an invariance that holds up for an unknown distribution shift. Similarly, *learning* such middle domains with the downstream objective of robustness appears to be a worthwhile future direction. Also, the selection of corruptions for sigma training was not optimized. Finding/learning corruptions that transfers better to other unseen corruptions can provide uncertainties with better generalization.

Low-dimensional task: Our experiments are on pixel-wise prediction tasks. Investigation of robustness for categorical tasks, such as classification, requires further studies.

Adversarial robustness: We focused on corruptions with a non-adversarial nature in our experiments. Also, the proposed multiple domain ensembling approach, in theory, could prevent learning surface statistics of the data (Jo & Bengio (2017)) compared to the methods using the input domain only and entirely. These angles require further investigations.

REFERENCES

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. Ieee, 2009.

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

Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, pp. 1–15. Springer, 2000.

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)*, pp. 1–7. IEEE, 2017.

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.

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

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

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

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

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*, pp. 770–778, 2016.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

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

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

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pp. 6402–6413, 2017.

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.

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*, pp. 13991–14002, 2019.

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

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*, pp. 234–241. Springer, 2015.

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.

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.

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

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.

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*, pp. 3712–3722, 2018.

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*, pp. 2223–2232, 2017.

A Appendix

A.1 Loss curves for distortions

Figures 8 show the ℓ_1 errors of our method and several baselines for each distortion, for normal, reshading and depth targets. The proposed inverse variance and network merging approaches consistently outperform baselines.

A.2 Blind guess

This output is computed using the following formula:

$$g^* = \min_g \mathcal{L}_{NLL} \tag{5}$$

The resulting "blind guess" minimizes expected NLL loss on the training dataset. Hence it is a statistically informed guess which does not look at the input for predicting the label. A visualization of these guesses for depth, normal, and reshading are provided in Figure 9.

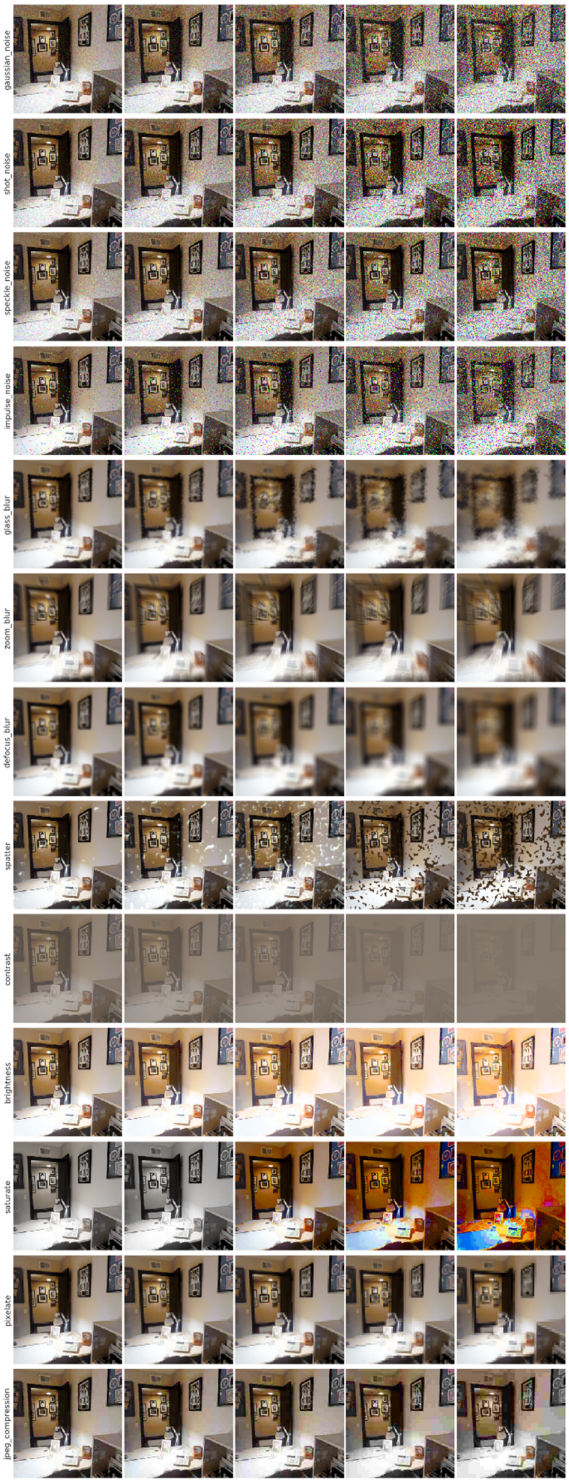
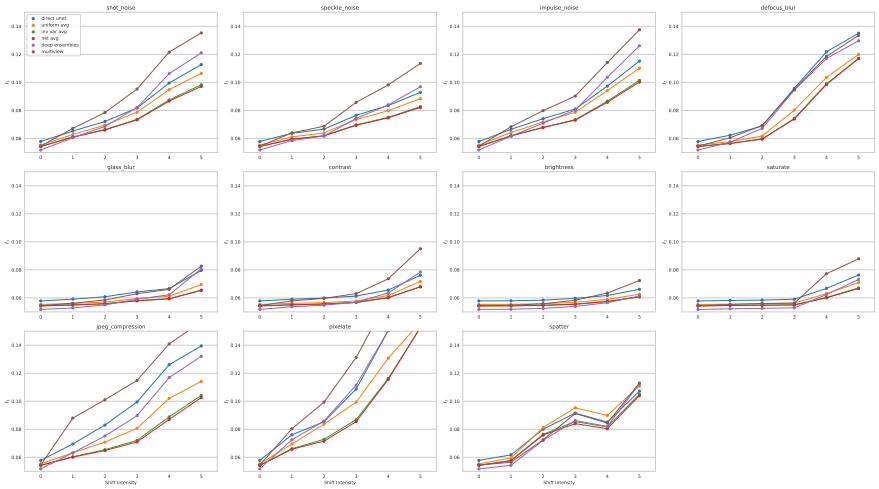
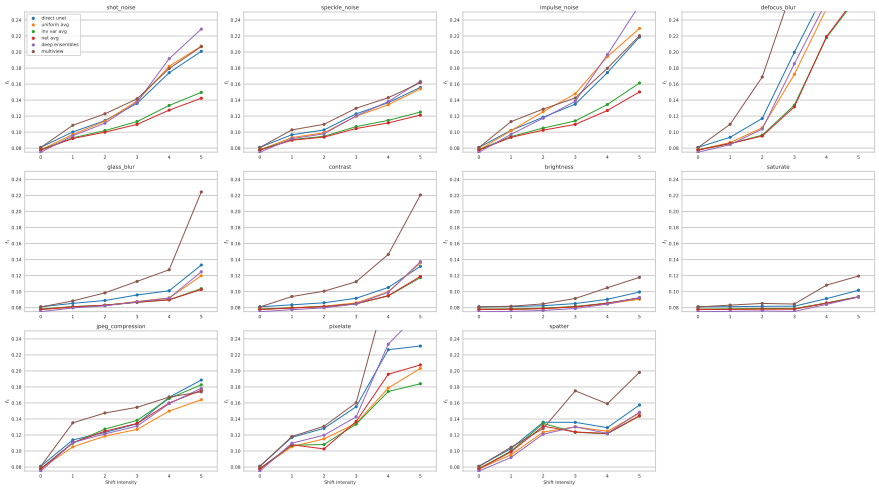


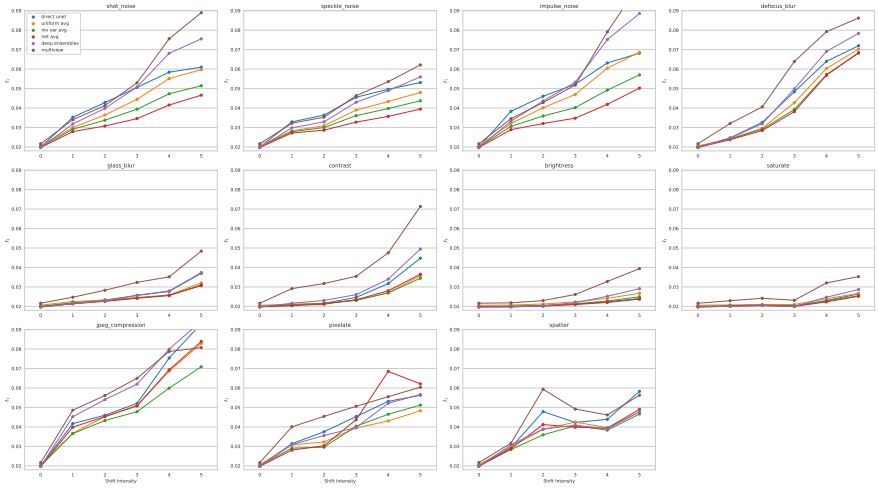
Figure 7: Visuals of common corruptions used for a single image sample.



(a) Normal

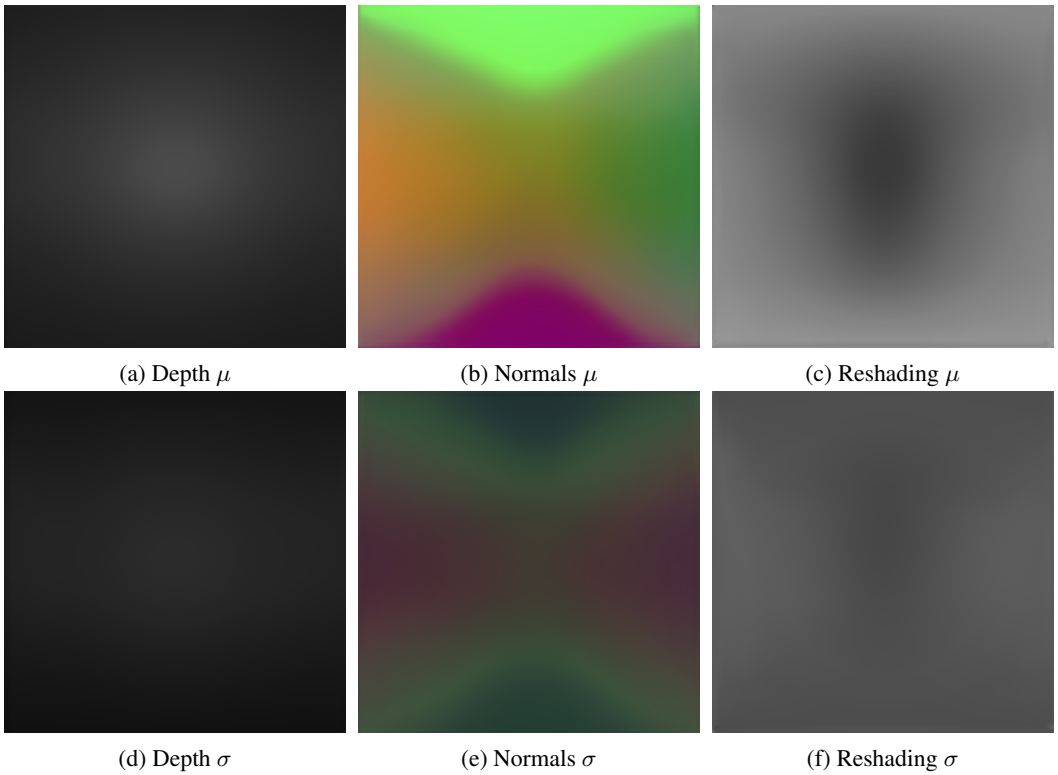


(b) Reshade



(c) Depth

Figure 8: Loss for each distortion for our method against several baselines.



(a) Depth μ (b) Normals μ (c) Reshading μ

(d) Depth σ (e) Normals σ (f) Reshading σ

Figure 9: Statistically informed guesses ("Blind Guess") on the Taskonomy dataset for depth, normal, and reshading tasks. Top row shows the predicted mean, μ, while the bottom row corresponds to the predicted standard deviation, σ, for these tasks. The blind guess predictions minimize the expected NLL loss on the training dataset.